



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

**PERANCANGAN *REUSABLE PACKAGING* TUMBLER
BERBASIS *KANSEI ENGINEERING* BERDASARKAN
PERSEPSI LINTAS GENERASI**



PRODI TEKNOLOGI INDUSTRI CETAK KEMASAN

JURUSAN TEKNIK GRAFIKA DAN PENERBITAN

POLITEKNIK NEGERI JAKARTA

2026



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

**PERANCANGAN *REUSABLE PACKAGING* TUMBLER
BERBASIS *KANSEI ENGINEERING* BERDASARKAN
PERSEPSI LINTAS GENERASI**



JURUSAN TEKNIK GRAFIKA DAN PENERBITAN

POLITEKNIK NEGERI JAKARTA

2026

LEMBAR PERSETUJUAN

PERANCANGAN *REUSABLE PACKAGING* TUMBLER BERBASIS *KANSEI ENGINEERING* BERDASARKAN PERSEPSI LINTAS GENERASI



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Disetujui,
Depok, 14 Juli 2026

Pembimbing

Novi Purnama Sari, S.T.P., M.Si.
NIP. 198911212019032018

Ketua Program Studi

Muryeti, S.Si., M.Si.
NIP. 197308111999032001

**POLITEKNIK
NEGERI
JAKARTA**



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

LEMBAR PENGESAHAN

PERANCANGAN REUSABLE *PACKAGING TUMBLER* BERBASIS *KANSEI ENGINEERING* BERDASARKAN PERSEPSI LINTAS GENERASI

Disahkan pada,
Depok, 14 Juli 2026

Penguji I

Penguji II

Dr. Zulkarnain, S.T., M.Eng
NIP. 198405292012121002

Adita Evalina Fitria Utami, S.T., M.T.
NIP. 199403102024062001

Ketua Program Studi

Muryeti, S.Si, M.Si.
NIP. 197308111999032001

Ketua Jurusan

Dr. Zulkarnain, S.T., M.Eng
NIP. 198405292012121002

PERNYATAAN ORISINALITAS

Saya yang bertanda tangan di bawah ini menyatakan dengan sebenarnya bahwa semua pernyataan dalam skripsi saya ini dengan judul **PERANCANGAN *REUSABLE PACKAGING* TUMBLER BERBASIS *KANSEI ENGINEERING* BERDASARKAN PERSEPSI LINTAS GENERASI** merupakan hasil studi pustaka, penelitian lapangan dan tugas karya akhir saya sendiri, di bawah bimbingan Dosen Pembimbing yang telah ditetapkan oleh pihak Jurusan Teknik Grafika dan Penerbitan Politeknik Negeri Jakarta.

Skripsi ini belum pernah diajukan sebagai syarat kelulusan pada program manapun di perguruan tinggi lain. Semua informasi, data dan hasil analisis maupun pengolahan yang digunakan, telah dinyatakan sumbernya dengan jelas dan dapat diperiksa kebenarannya.

Depok, 14 Juli 2026



Abdillah Nur Fatah

2206411069



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

**POLITEKNIK
NEGERI
JAKARTA**



RINGKASAN

Penelitian ini menerapkan pendekatan *Machine Learning* dan *Kansei Engineering* (KE) untuk merancang botol tumbler berdasarkan lintas generasi dengan menerjemahkan persepsi emosional konsumen menjadi parameter desain fisik yang objektif dan terukur guna mengatasi bias subjektivitas kuesioner konvensional. Perbandingan preferensi Generasi Milenial dan Generasi Z menjadi landasan utama, mengingat kedua segmen tersebut mendominasi pasar tumbler namun memiliki karakteristik dan nilai keberlanjutan yang berbeda. Tahap awal dilakukan ekstraksi kata *Kansei* dari 17.325 ulasan konsumen dari 40 sampel tumbler *e-commerce* menggunakan *Text mining* atau *Natural Language Processing* (NLP). Setelah *preprocessing* dan penyaringan ulasan dengan minimal dua kata *Kansei*, diperoleh 2.504 ulasan dan 34 kata *Kansei* terpilih melalui pembobotan TF-IDF. Metode *Self-organizing Map* (SOM) kemudian memetakan 34 kata *Kansei* ke dalam tujuh dimensi desain, lalu mensegmentasikan 2.504 ulasan menjadi 25 segmen pengguna. Hasilnya, tiga konsep prioritas terpilih: Premium dengan 28% segmen, Durabilitas dan Ergonomis dengan masing-masing 20% segmen. Penentuan elemen desain selanjutnya memanfaatkan 13 faktor morfologi (X1–X13) dari 40 sampel yang telah didapatkan, kemudian diolah dengan *Random Forest* (RF) dan dievaluasi melalui *Leave-One-Out Cross-Validation* (LOOCV) pada enam kombinasi model (dua generasi $\times$ tiga konsep). Struktur dinding (X8) konsisten menjadi faktor dengan importance tertinggi, diikuti bahan material (X7) dan *finishing* (X11). Berdasarkan nilai kinerja terendah pada ketiga metrik (MSE, RMSE, MAE), konsep Ergonomis terpilih sebagai model terbaik pada kedua generasi. Hasil akhir berupa visualisasi 3D *mockup*, yaitu: desain Generasi Milenial bergaya *Minimalist* (tutup ulir polos, *Carry half circle handle*, *stainless steel*, *double-wall*, warna *duo-tone* krem-hitam), sedangkan desain Generasi Z bergaya *Cute* (*Quencher 3-way lid*, *Carry half hexagon handle* dengan *Strap/lanyard*, warna *three-tone* oranye-kuning-biru keunguan) yang merepresentasikan karakter afektif masing-masing generasi.

Kata kunci: *Kansei Engineering*, *Natural Language Processing*, *Self-organizing Map*, *Support Vector Regression*, Tumbler

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

SUMMARY

This research applies a Machine Learning and Kansei Engineering (KE) to design cross-generational tumbler bottles by translating consumers' emotional perceptions into objective, measurable design parameters, addressing the subjectivity bias of conventional questionnaires. Comparing Millennial and Gen Z preferences serves as the main focus, as both segments dominate the tumbler market but differ in characteristics and sustainability values. Kansei words were first extracted from 17,325 consumer reviews across 40 e-commerce tumbler samples using Text mining/NLP. After preprocessing and filtering reviews with at least two Kansei words, 2,504 reviews and 34 Kansei words were selected via TF-IDF weighting. Self-organizing Map (SOM) then mapped these words into seven design dimensions and segmented the reviews into 25 user segments, yielding three priority concepts: Premium (28% of segments), and Durability and Ergonomic (20% each). Design elements were then determined using 13 morphological factors (X1–X13) from the 40 samples, processed with Random Forest (RF) and evaluated via LOOCV across six model combinations (two generations × three concepts). Wall structure (X8) consistently showed the highest importance, followed by material (X7) and finishing (X11). Based on the lowest MSE, RMSE, and MAE values, the Ergonomic concept performed best for both generations. The final output comprises 3D mockups: a Minimalist design for Millennials (plain screw lid, half-circle handle, stainless steel, double-wall, cream-black duo-tone) and a Cute design for Gen Z (3-way quencher lid, hexagonal handle with Strap, orange-yellow-purplish blue tri-tone), each reflecting the affective character of its generation.

Keyword: Kansei Engineering, Natural Language Processing, Self-organizing Map, Support Vector Regression, Tumbler



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

KATA PENGANTAR

Puji syukur kehadiran Tuhan Yang Maha Esa atas rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi ini dengan judul "PERANCANGAN REUSABLE PACKAGING TUMBLER BERBASIS KANSEI *ENGINEERING* BERDASARKAN PERSEPSI LINTAS GENERASI" ini dapat diselesaikan tepat pada waktunya. Skripsi ini disusun untuk memenuhi syarat dalam menyelesaikan pendidikan dan memperoleh gelar Sarjana Terapan (D4) pada Program Studi Teknologi Industri Cetak Kemasan, Jurusan Teknik Grafika dan Penerbitan, Politeknik Negeri Jakarta. Penulis menyadari bahwa skripsi ini tidak dapat selesai tanpa bimbingan dari pihak lain. Oleh karena itu, penulis ingin mengucapkan terima kasih kepada:

1. Dr. Syamsurizal, S.E., M.M. selaku Direktur Politeknik Negeri Jakarta.
2. Dr. Zulkarnain, S.T., M.Eng. M.M. selaku Ketua Jurusan Teknik Grafika Penerbitan.
3. Muryeti, S.Si., M.Si. selaku Ketua Program Studi Teknologi Industri Cetak Kemasan.
4. Novi Purnama Sari, S.TP, M.Si., selaku dosen pembimbing pengembangan kemasan yang telah membimbing, memberi arahan, dan memberi saran dalam penyusunan skripsi ini.
5. Kepada dosen-dosen TICK lainnya yang turut membantu dan memberikan ilmu pembelajaran selama 4 tahun.
6. Kedua orang tua, adik, dan keluarga penulis yang selalu memberi doa, dukungan, serta semangat dalam penyusunan skripsi ini hingga selesai.
7. *Expert panelist* yang telah meluangkan waktunya untuk wawancara dan berdiskusi dalam penelitian ini.
8. Teman Yamin Lovers yang telah saling mendukung dan berjuang bersama sehingga penelitian skripsi ini dapat selesai dengan baik dan tepat waktu.
9. Teman mahasiswa/i TICK 22 C atas segala bantuan yang diberikan selama proses perkuliahan dan berjuang hingga tahap akhir ini.
10. Seluruh responden yang telah memberikan waktu luang untuk mengisi seluruh kuesioner penelitian ini.



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Penulis menyadari bahwa penelitian skripsi ini masih terdapat kekurangan dan jauh dari kata sempurna. Oleh karena itu, segala kritik dan saran dari semua pihak sangat diharapkan demi penyempurnaan di masa mendatang. Semoga penelitian skripsi ini dapat memberikan kontribusi positif dan manfaat bagi pengembangan ilmu pengetahuan, khususnya dalam bidang pengembangan kemasan.

Depok, 14 Juli 2026

Abdillah Nur Fatah





DAFTAR ISI

LEMBAR PERSETUJUAN	i
LEMBAR PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
RINGKASAN	iv
SUMMARY	v
KATA PENGANTAR	vi
DAFTAR ISI	viii
DAFTAR GAMBAR	x
DAFTAR TABEL	xii
DAFTAR LAMPIRAN	xiii
BAB I PENDAHULUAN	12
1.1 Latar Belakang	12
1.2 Rumusan Masalah	16
1.3 Tujuan Penelitian	17
1.4 Manfaat Penelitian	17
1.5 Ruang Lingkup Penelitian	18
BAB II TINJAUAN PUSTAKA	19
2.1 <i>State of The Art</i>	19
2.2 Tumbler	21
2.3 Kemasan	22
2.4 <i>Kansei Engineering</i>	23
2.5 <i>Machine Learning</i>	25
2.6 <i>Natural Language Processing</i>	26
2.7 Self-organizing Map	28
2.8 Random Forest	32
BAB III METODOLOGI PENELITIAN	35
3.1 Rancangan Penelitian	35
3.1.1 Alat Penelitian	36
3.2 Metode Pengumpulan Data	36
3.3 Prosedur Analisis Data	38

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

3.3.1 Identifikasi Masalah dan Studi Literatur	39
3.3.2 Pengumpulan Sampel-sampel Tumbler	40
3.3.3 Pengumpulan Ulasan-ulasan Konsumen	40
3.3.4 Pengolahan Kata <i>Kansei</i> dengan <i>Natural Language Processing</i> (NLP)	41
3.3.5 Analisis Konsep Desain dengan <i>Self-organizing Map</i> (SOM)	41
3.3.6 Analisis Morfologi Elemen-elemen Desain Tumbler	42
3.3.7 Analisis Konsep pada Elemen-elemen Desain Tumbler	42
3.3.8 Pengolahan Elemen-elemen Desain Tumbler dengan <i>Random Forest</i> (RF)	43
3.3.9 Perancangan Desain dan <i>Mockup</i> 3D Tumbler	43
BAB IV HASIL DAN PEMBAHASAN	44
4.1 Identifikasi Masalah dan Studi Literatur	44
4.2 Pengumpulan Sampel-sampel Tumbler	45
4.3 Pengumpulan Ulasan-ulasan Konsumen	46
4.4 Pengolahan Kata <i>Kansei</i> dengan <i>Natural Language Processing</i> (NLP)	47
4.5 Analisis Konsep Desain dengan <i>Self-organizing Map</i> (SOM)	58
4.5.1 Pengelompokan Kata <i>Kansei</i> menjadi Dimensi Desain (Stage 1 – SOM)	58
4.5.2 Segmentasi Ulasan menjadi Segmen Pengguna (Stage 2 – SOM)	62
4.6 Analisis Morfologi Elemen-elemen Desain Tumbler	69
4.7 Pengukuran Evaluasi Konsep Kemasan Berdasarkan Sampel Kemasan	74
4.8 Analisis Metode Elemen-elemen Desain Tumbler dengan <i>Random Forest</i>	75
4.9 Pembuatan Desain Tumbler dan 3D <i>Mock Up</i>	102
BAB V SIMPULAN DAN SARAN	108
5.1 Simpulan	108
5.2 Saran	110
DAFTAR PUSTAKA	112
LAMPIRAN	126
DAFTAR RIWAYAT HIDUP	181



DAFTAR GAMBAR

Gambar 3.1. Studi <i>Framework</i> Penelitian.....	35
Gambar 3.2. Diagram Alur Penelitian.....	39
Gambar 4.1. Tampilan <i>Software</i> WebHarvy©.....	47
Gambar 4.2. Peta <i>U-Matrix</i> 6x6 Stage 1	60
Gambar 4.3. Kurva <i>Silhouette</i> Stage 1	60
Gambar 4.4. Peta SOM 6x6 pada Stage 1	61
Gambar 4.5. Peta <i>U-Matrix</i> 17x16 Stage 2	63
Gambar 4.6. Kurva <i>Silhouette</i> Stage 2	64
Gambar 4.7. Peta SOM 16x17 pada Stage 2	65
Gambar 4.8. Diagram <i>Pie-chart</i> Dominan Segmen Konseptual Desain.....	69
Gambar 4.9. Kuesioner Semantic Differential II Likert	74
Gambar 4.10. 6 Distribusi Skor Kansei pada 2 Generasi dan 3 Konsep.....	76
Gambar 4.11. <i>Heatmap</i> Korelasi Antar 13 Faktor Morfologi	78
Gambar 4.12. <i>Feature Importance</i> LOOCV Gen Milenial Konsep Premium	79
Gambar 4.13. (a) Grafik Actual vs <i>OOF-Predicted</i> Gen Milenial Konsep Premium dan (b) Grafik Residual Plot (LOOCV) Gen Milenial Konsep Premium.....	79
Gambar 4.14. <i>Feature Importance</i> LOOCV Gen Milenial Konsep Durabilitas ...	82
Gambar 4.15. (a) Grafik Actual vs <i>OOF-Predicted</i> Gen Milenial Konsep Durabilitas dan (b) Grafik Residual Plot (LOOCV) Gen Milenial Konsep Durabilitas	83
Gambar 4.16. <i>Feature Importance</i> LOOCV Gen Milenial Konsep Ergonomis ...	86
Gambar 4.17. (a) Grafik Actual vs <i>OOF-Predicted</i> Gen Milenial Konsep Ergonomis dan (b) Grafik Residual Plot (LOOCV) Gen Milenial Konsep Ergonomis	86
Gambar 4.18. <i>Feature Importance</i> LOOCV Gen Z Konsep Premium	89
Gambar 4.19. (a) Grafik Actual vs <i>OOF-Predicted</i> Gen Z Konsep Premium dan (b) Grafik Residual Plot (LOOCV) Gen Z Konsep Premium.....	89
Gambar 4.20. <i>Feature Importance</i> LOOCV Gen Z Konsep Durabilitas	92
Gambar 4.21. (a) Grafik Actual vs <i>OOF-Predicted</i> Gen Z Konsep Durabilitas dan (b) Grafik Residual Plot (LOOCV) Gen Z Konsep Durabilitas.....	93

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Gambar 4.22. <i>Feature Importance</i> LOOCV Gen Z Konsep Ergonomis	96
Gambar 4.23. (a) Grafik Actual vs <i>OOF-Predicted</i> Gen Z Konsep Ergonomis dan (b) Grafik Residual Plot (LOOCV) Gen Z Konsep Ergonomis	96
Gambar 4.24. Metrik Kinerja Model (RMSE, MSE, MAE) terhadap 2 Generasi dan 3 Konsep.....	99
Gambar 4.25. <i>Mindmap</i> Generasi Milenial.....	103
Gambar 4.26. <i>Moodboard</i> Generasi Milenial	104
Gambar 4.27. Hasil <i>Mockup</i> 3D Generasi Milenial.....	104
Gambar 4.28. <i>Mindmap</i> Generasi Z	106
Gambar 4.29. <i>Moodboard</i> Generasi Z	106
Gambar 4.30. Hasil <i>Mockup</i> 3D Generasi Z	107





Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

DAFTAR TABEL

Tabel 3.1. Alat Penelitian	36
Tabel 3.2. Metode Pengumpulan Data	37
Tabel 4.1. Sampel-sampel Tumbler.....	45
Tabel 4.2. Hasil <i>Case Folding</i> dan <i>Cleaning</i>	48
Tabel 4.3. Hasil <i>Tokenization</i>	49
Tabel 4.4. Hasil <i>Normalization</i>	50
Tabel 4.5. Hasil <i>Phrase Matching</i>	51
Tabel 4.6. Hasil <i>Stopword Removal</i>	51
Tabel 4.7. Hasil <i>Stemming</i>	52
Tabel 4.8. Hasil <i>POS Tagging</i>	53
Tabel 4.9. Hasil Pembobotan TF-IDF	54
Tabel 4.10. Hasil Frekuensi Analisis Sentimen.....	55
Tabel 4.11. Hasil Analisis <i>Sentiment Score</i> dan <i>Polarity</i>	56
Tabel 4.12. 7 Dimensi Desain <i>Kansei</i> Stage 1	62
Tabel 4.13. 25 Segmen Kluster Desain <i>Kansei</i> Stage 2	65
Tabel 4.14. Morfologi Sampel Tumbler.....	70
Tabel 4.15. Rincian <i>Breakdown</i> Morfologi Sampel Tumbler	72
Tabel 4.16. Hasil Metrik Kinerja Model (RMSE, MSE, dan MAE) terhadap 2 Generasi dan 3 Konsep.....	99
Tabel 4.17. Hasil <i>Kansei</i> Prediction pada Konsep Desain Gen Milenial dan Gen Z	101



DAFTAR LAMPIRAN

Lampiran 1. Hasil <i>Web Scraping Text</i>	126
Lampiran 2. <i>Source Code Text Mining</i> atau NLP	130
Lampiran 3. Hasil kata <i>Kansei</i> dari TF-IDF	141
Lampiran 4. Hasil <i>Document-term Matrix</i>	143
Lampiran 5. Hasil <i>Metadata</i> Min 2 Kata <i>Kansei</i>	147
Lampiran 6. <i>Source Code Self-organizing Map</i> Stage 1	150
Lampiran 7. <i>Source Code Self-organizing Map</i> Stage 2	155
Lampiran 8. Identitas <i>Expert Panelist</i> dalam Penentuan Konsep Desain	159
Lampiran 9. Bukti Diskusi Bersama <i>Expert Panelist</i>	160
Lampiran 10. Contoh Kuesioner <i>Semantic Differential</i> (SD) II Skala <i>Likert</i>	161
Lampiran 11. Hasil Kuesioner <i>Semantic Differential</i> (SD) II Skala <i>Likert</i>	162
Lampiran 12. Dataset Input <i>Random Forest</i>	170
Lampiran 13. <i>Source Code Random Forest</i>	172
Lampiran 14. Logbook Kegiatan Bimbingan Materi	179

POLITEKNIK
NEGERI
JAKARTA

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

BAB I PENDAHULUAN

1.1 Latar Belakang

Peningkatan konsumsi minuman dalam kemasan botol plastik sekali pakai telah berkontribusi signifikan terhadap peningkatan timbunan sampah plastik secara global. *United Nations Environment Programme* melaporkan bahwa produksi plastik dunia mencapai lebih dari 400 juta ton per tahun dan sekitar 36% di antaranya digunakan untuk kemasan sekali pakai [1]. Dominasi sampah plastik ini sangat dipengaruhi oleh tingginya permintaan terhadap produk Air Minum Dalam Kemasan (AMDK) yang umumnya menggunakan botol *Polyethylene Terephthalate* (PET) [2]. Penggunaan botol plastik sekali pakai yang bersifat praktis dan ekonomis mendorong masyarakat untuk terus mengandalkan produk tersebut dalam aktivitas sehari-hari [3]. Data Badan Pusat Statistik (BPS) tahun 2021 juga mencatat bahwa timbunan sampah nasional mencapai lebih dari 30 juta ton per tahun dengan komposisi sampah plastik berada pada kisaran 17–19% dari total timbunan, dan sebagian besar berasal dari kemasan sekali pakai [4]. Kondisi tersebut menunjukkan bahwa penggunaan botol plastik sekali pakai berkontribusi secara nyata terhadap peningkatan volume sampah yang sulit terurai dan berpotensi mencemari lingkungan dalam jangka panjang. Oleh karena itu, diperlukan upaya strategis untuk mengurangi ketergantungan terhadap kemasan botol plastik sekali pakai melalui pengembangan dan pemanfaatan produk alternatif yang lebih berkelanjutan.

Salah satu alternatif untuk mengurangi kemasan botol plastik sekali pakai adalah menggunakan botol tumbler. Tumbler termasuk *reuseable packaging* karena berfungsi sebagai peralihan kemasan sekali pakai menjadi wadah yang memfasilitasi proses pengisian kembali, baik dilakukan secara mandiri oleh konsumen maupun melalui dukungan infrastruktur produk tambahan yang tersedia di pasar. Menurut *International Organization for Standardization* melalui standar ISO 18603:2013 mendefinisikan kemasan guna ulang sebagai kemasan yang dirancang, diproduksi, dan dipasarkan untuk digunakan kembali dalam sejumlah siklus penggunaan sesuai dengan tujuan awalnya [5]. Menurut laporan *Ellen MacArthur Foundation* dalam dokumen "*Reuse: Rethinking Packaging*" sebagai



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

kemasan reusable, tumbler mendukung berbagai model bisnis seperti *refill at home* (isi ulang di rumah) maupun *refill on the go* (isi ulang di lokasi publik) yang memungkinkan wadah tersebut berfungsi menggantikan botol plastik sekali pakai secara terus-menerus sepanjang siklus hidupnya [6]. Sehingga, penggunaan botol tumbler sebagai pengganti plastik sekali pakai menjadi solusi relevan di tengah meningkatnya tuntutan konsumen akan produk yang personal, berkualitas, dan ramah lingkungan.

Penggunaan tumbler kini memiliki peminat kalangan generasi, terutama kalangan generasi Milenial dan Gen-Z. Fenomena ini muncul seiring dengan meningkatnya kesadaran lingkungan dan perubahan pola konsumsi masyarakat urban dalam satu dekade terakhir. Kondisi tersebut mendorong munculnya gerakan pengurangan plastik sekali pakai yang secara aktif dipromosikan melalui kampanye digital, komunitas lingkungan, dan kebijakan pembatasan kantong plastik di berbagai daerah [7]. Setiap individu yang memiliki kesadaran lingkungan lebih tinggi cenderung lebih memilih untuk membeli tumbler ramah lingkungan dan lebih mungkin mempercayai klaim lingkungan serta kualitas produk-produk tersebut [8]. Perbandingan lintas generasi yang berfokus antara generasi Milenial dan Gen Z juga memiliki peranan penting karena kedua generasi ini mendominasi pangsa pasar. Generasi Milenial umumnya menggunakan tumbler dari nilai fungsional produk dengan tanggung jawab lingkungan jangka panjang secara pragmatis dan berorientasi pada nilai investasi [9], sedangkan Gen-Z menggunakan tumbler sebagai bentuk partisipasi proaktif dalam gerakan sosial, seperti isu-isu lingkungan, sosial, serta memiliki pandangan yang lebih terbuka secara global dan representasi gaya hidup kekinian [10]. Oleh karena itu, perancangan desain botol tumbler membutuhkan sebuah pendekatan komprehensif yang mampu menangkap dan menstrukturkan perbedaan persepsi emosional lintas generasi tersebut secara presisi, sehingga desainer memiliki landasan yang objektif dalam merancang spesifikasi produk.

Proses perancangan awal botol tumbler berfokus pada pengembangan konsep untuk membangun respons emosional dan persepsi positif konsumen. Metode *Kansei Engineering* dapat mengeksplorasi preferensi serta aspek emosional pelanggan secara mendalam [11]. Pendekatan tersebut bertujuan menerjemahkan



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

persepsi afektif pengguna ke dalam atribut desain yang terstruktur dan terukur terhadap objek atau produk [12]. Metode *Kansei Engineering* umumnya menggunakan kuesioner, survei, wawancara, dan diskusi kelompok terarah (*focus group discussion*) untuk mengidentifikasi, mengukur, serta mengevaluasi respons emosional konsumen terhadap suatu produk [13]. Sehingga desain yang dihasilkan dapat disesuaikan secara persepsi afektif dengan karakteristik generasi Milenial dan Gen-Z.

Penelitian ini mengadopsi pendekatan *Kansei Engineering* dengan memanfaatkan data ulasan pelanggan sebagai sumber utama identifikasi respons emosional. *Kansei Engineering* memiliki keunggulan dibandingkan metode sejenis karena kemampuannya menerjemahkan kebutuhan emosional konsumen ke dalam parameter desain yang terukur. *Kansei Engineering* berfungsi untuk mengubah parameter kualitatif dari preferensi emosional konsumen yang diekspresikan melalui pengumpulan kata sifat (kata *Kansei*) menjadi sekumpulan data kuantitatif yang terukur untuk acuan elemen fisik produk [14]. Berbeda dengan teknik konvensional yang bergantung pada kuesioner atau wawancara terstruktur, penelitian ini menggunakan data ulasan daring yang tersedia pada berbagai *platform e-commerce* berdasarkan *Text mining* berdasarkan metode *Natural Language Processing* (NLP). Metode *Natural Language Processing* (NLP) dalam *domain text mining* digunakan untuk mengidentifikasi kata *Kansei* secara otomatis dari ribuan ulasan pelanggan [15]. *Text mining* sendiri merupakan sub-bidang *data mining* yang berfungsi mengeksplorasi dan menganalisis kumpulan teks atau *dataset* dalam skala besar [16]. Oleh karena itu, penggunaan metode *text mining* dalam penelitian ini sangat penting karena metode tersebut mendemonstrasikan kapabilitas algoritma yang tangguh untuk memproses volume teks tidak terstruktur secara masif dan otomatis dan memitigasi anomali data atau teks konsumen menjadi lebih presisi, terukur, dan sepenuhnya digerakkan oleh data (*data-driven*) [17].

Proses mendapatkan *Kansei words* melalui *text mining* menggunakan *Natural Language Processing* (NLP) yang telah dilakukan, tahap selanjutnya adalah menentukan konsep desain menggunakan metode *Self-organizing Map* (SOM). SOM berfungsi sebagai model *Machine Learning neural-network based* yang mampu memetakan data emosional kompleks dan berdimensi tinggi ke dalam



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

representasi visual dua dimensi yang lebih sederhana [18]. Keunggulan utama SOM terletak pada kemampuan untuk melakukan pengelompokan (*clustering*) secara otomatis serta mengidentifikasi hubungan antar-variabel afektif yang serupa dalam satu unit pemetaan [19]. Integrasi ini memungkinkan ekstraksi kebutuhan yang terkandung oleh pengguna secara langsung dari data tekstual berskala besar, sehingga meniadakan ketergantungan pada kuesioner konvensional yang sering kali terbatas oleh bias subjektivitas dan jumlah partisipan. Hal ini memungkinkan desainer untuk melihat pola persepsi pengguna secara objektif, efisien, dan mendalam, sehingga tercipta landasan yang kuat dalam menentukan atribut desain yang paling responsif terhadap emosi spesifik dari generasi Milenial dan Gen-Z.

Konsep yang telah diperoleh dengan metode SOM, kemudian ditentukan elemen-elemen desain yang sesuai dengan menggunakan metode *Random Forest* (RF). RF dirancang untuk menyelesaikan permasalahan regresi melalui prinsip *ensemble learning* berbasis agregasi pohon keputusan dan mekanisme *bootStrap aggregating*, sehingga mampu menghasilkan model dengan generalisasi yang baik, tahan terhadap *overfitting*, dan tidak memerlukan asumsi distribusi tertentu meskipun jumlah data relatif terbatas dan hubungan antarvariabel bersifat nonlinier. Karakteristik RF menjadi krusial karena relasi antara persepsi emosional pengguna dan elemen morfologis produk umumnya kompleks, ambigu, melibatkan interaksi antar-elemen, serta tidak mengikuti pola linear sederhana. Selain itu, evaluasi model dilakukan menggunakan *Leave-One-Out Cross-Validation* (LOOCV) sebagai strategi pengujian data latih dan data uji, karena LOOCV memprediksi seluruh sampel secara *out-of-fold* sehingga memberikan estimasi generalisasi yang lebih objektif dan stabil pada *dataset* berukuran kecil dibandingkan skema *single-split* yang mudah menghasilkan nilai evaluasi tidak stabil. Penelitian [20] menunjukkan bahwa RF efektif dalam membangun model prediksi nilai evaluasi *Kansei* sekaligus mengidentifikasi elemen desain dominan melalui *metrik Feature Importance* yang dapat langsung ditafsirkan oleh desainer. Studi [21] memperlihatkan bahwa penerapan RF dalam sistem perancangan berbasis *data mining* menghasilkan prediksi elemen desain yang lebih akurat ketika dikombinasikan dengan segmentasi pengguna. Kemudian studi [22] juga menemukan bahwa RF mampu mempertahankan performa pemodelan respons



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

afektif yang konsisten meskipun beroperasi pada ruang fitur berdimensi tinggi yang mencerminkan kompleksitas morfologi desain nyata. Berdasarkan temuan tersebut, RF memiliki keunggulan dalam memodelkan hubungan nonlinier, mengurangi risiko *overfitting* melalui mekanisme randomisasi, menghasilkan interpretasi kontribusi tiap elemen desain secara transparan, serta fleksibel terhadap data kategorikal tanpa transformasi ordinal yang keliru, sehingga sangat tepat digunakan untuk menentukan dan mengoptimalkan elemen-elemen desain dalam *Kansei Engineering* secara sistematis dan berbasis data.

Penelitian ini diharapkan dapat memberikan kontribusi bagi industri tumbler dalam memahami preferensi desain dan rekomendasi elemen desain yang tepat sasaran. Rekomendasi desain tersebut mampu merepresentasikan identitas karakteristik generasi Milenial dan Gen-Z sebagai segmen konsumen dominan di pasar saat ini. Metode *Natural Language Processing* (NLP) digunakan untuk mengekstraksi representasi kata *Kansei* dari data *text mining* ulasan konsumen. Selanjutnya, metode *Self-organizing Map* (SOM) dan *Random Forest* (RF) diaplikasikan untuk mengklasifikasikan preferensi konsep desain dan menentukan elemen desain tumbler secara objektif.

1.2 Rumusan Masalah

Berdasarkan permasalahan yang terjadi dapat dirumuskan beberapa masalah pada penelitian ini sebagai berikut:

1. Apa saja kata *Kansei* yang didapatkan melalui *text mining* yang sesuai dengan referensi generasi Milenial dan Gen-Z terhadap botol tumbler yang diperoleh dari pengolahan kata *Kansei* menggunakan metode *Natural Language Processing* (NLP)?
2. Bagaimana hasil konsep perancangan botol tumbler yang sesuai dengan referensi generasi Milenial dan Gen-Z menggunakan metode *Self-organizing Map* (SOM)?
3. Bagaimana hasil elemen desain perancangan botol tumbler yang sesuai dengan referensi generasi Milenial dan Gen-Z menggunakan metode *Random Forest* (RF)?



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

4. Bagaimana hasil bentuk 3D *mockup* desain dari setiap elemen terpilih perancangan botol tumbler yang sesuai dengan referensi generasi Milenial dan Gen-Z?

1.3 Tujuan Penelitian

Berdasarkan masalah yang telah dirumuskan di atas, maka tujuan dari penelitian ini adalah sebagai berikut:

1. Menentukan kata *Kansei* melalui *text mining* dengan implementasi metode *Natural Language Processing* (NLP).
2. Menentukan hasil konsep desain terpilih menggunakan metode *Self-organizing Map* (SOM).
3. Menganalisis elemen-elemen desain botol tumbler berdasarkan konsep desain dan sampel tumbler dengan implementasi metode *Random Forest* (RF).
4. Membuat hasil rancangan konsep desain terpilih dalam bentuk 3D *mockup* tumbler untuk setiap elemen terpilih berdasarkan preferensi generasi Milenial dan Gen-Z menggunakan metode *Kansei Engineering*.

1.4 Manfaat Penelitian

Manfaat yang didapatkan berdasarkan tujuan dari penelitian ini adalah sebagai berikut:

1. Bagi konsumen, desain botol tumbler yang dikembangkan berdasarkan preferensi generasi Milenial dan generasi Z tidak hanya berfungsi sebagai wadah minuman, tetapi juga sebagai representasi gaya hidup, nilai keberlanjutan, dan identitas personal.
2. Bagi produsen, hasil penelitian ini memberikan dasar pengambilan keputusan berbasis data dalam merancang dan mengembangkan botol tumbler yang sesuai dengan preferensi pasar Milenial dan Gen-Z. Dengan demikian, produsen dapat merumuskan strategi desain yang lebih terarah untuk meningkatkan daya saing dan minat beli di pasar yang didominasi generasi muda.
3. Manfaat penelitian ini memperluas pemahaman mengenai integrasi metode NLP, SOM, dan RF dalam kerangka *Kansei Engineering* untuk perancangan



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

ulang desain produk berbasis preferensi generasional. Pendekatan ini menawarkan kontribusi metodologis baru dalam menghubungkan data tekstual konsumen dengan parameter desain yang terukur, khususnya pada pengembangan botol tumbler yang selaras dengan karakter dan ekspektasi emosional Generasi Milenial dan Gen-Z.

1.5 Ruang Lingkup Penelitian

Berdasarkan rumusan masalah di atas, ruang lingkup diperlukan agar penelitian ini memiliki batasan terhadap masalah penelitian. Ruang lingkup yang ditentukan pada penelitian sebagai berikut:

1. Penelitian hanya dilakukan terhadap botol tumbler berdasarkan preferensi generasi Milenial dan Gen-Z.
2. Penelitian hanya menggunakan metode NLP, SOM, dan RF.
3. Hasil penelitian berupa rancangan botol tumbler dalam 3D *mockup* dan tidak melibatkan data terkait aspek produksi maupun kondisi finansial dari objek yang diteliti.
4. Hasil 3D *mockup* rancangan botol tumbler tidak dilanjutkan ke tahap evaluasi.

**POLITEKNIK
NEGERI
JAKARTA**

BAB V SIMPULAN DAN SARAN

5.1 Simpulan

Berdasarkan hasil analisis data dalam perancangan konsep desain produk tumbler berbasis preferensi konsumen generasi Milenial dan Gen Z dengan metode *Kansei Engineering*, diperoleh kesimpulan sebagai berikut:

1. Penentuan kata *Kansei* dilakukan melalui *text mining* dengan implementasi metode *Natural Language Processing* (NLP) terhadap 17.325 ulasan konsumen yang dikumpulkan dari 40 sampel produk tumbler pada *platform e-commerce* Tokopedia. Pipeline NLP yang dibangun mencakup tahap awal yaitu *preprocessing*, *case folding*, *data cleaning*, *tokenization*, *normalization*, *phrase matching*, *stopword removal*, dan *stemming* melalui *library* Sastrawi atau NLTK. Selanjutnya, peneliti menerapkan *Part-of-Speech* (POS) *Tagging*, serta pembobotan TF-IDF dan analisis sentimen berbasis leksikon. Setelah dilakukan penyaringan terhadap ulasan yang memuat minimal dua kata *Kansei*, diperoleh 2.504 ulasan yang relevan untuk dianalisis lebih lanjut. Hasil perhitungan TF-IDF menghasilkan 34 kata *Kansei* bermuatan positif. Kata-kata *Kansei* tersebut merepresentasikan respons afektif konsumen terhadap atribut produk tumbler yang menjadi dasar bagi tahap perancangan konsep desain selanjutnya.
2. Penentuan konsep desain terpilih dilakukan menggunakan metode *Self-organizing Map* (SOM) melalui pipeline dua tahap. Pada tahap pertama, 34 kata *Kansei* dikelompokkan menjadi tujuh dimensi desain, yaitu Minimalis, Premium, Fungsional, Kualitas, Durabilitas, Ergonomis, dan Kenyamanan, menggunakan peta heksagonal berukuran 6×6 dengan nilai *Quantization Error* (QE) sebesar 0,3658, *Topographic Error* (TE) sebesar 0,0294, dan *Silhouette Score* sebesar 0,0914. Pada tahap kedua, 2.504 ulasan konsumen disegmentasikan ke dalam 25 segmen menggunakan peta berukuran 17×16 dengan nilai QE sebesar 0,4774, TE sebesar 0,0152, dan *Silhouette Score* sebesar 0,3726. Hasil segmentasi menunjukkan bahwa dimensi desain yang paling dominan dalam preferensi





Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

konsumen adalah Premium (28%), diikuti oleh Durabilitas (20%) dan Ergonomis (20%). Ketiga dimensi dominan tersebut menjadi dasar penentuan konsep desain terpilih, yang mengarahkan rancangan produk tumbler pada karakter.

3. Analisis elemen-elemen desain botol tumbler berdasarkan tiga belas faktor morfologi (X1–X13) yang diperoleh dari 40 sampel tumbler, dengan mengimplementasikan metode *Random Forest* yang dievaluasi menggunakan skema *Leave-One-Out Cross-Validation* (LOOCV) pada enam kombinasi model yang terbentuk dari dua segmen generasi (Milennial dan Gen Z) serta tiga konsep *Kansei* (Premium, Durabilitas, dan Ergonomis), yang menghasilkan nilai kinerja model berupa MSE, RMSE, dan MAE secara berturut-turut sebesar 0,0662, 0,2573, dan 0,2063 pada Gen Milennial–Premium; 0,0393, 0,1983, dan 0,157 pada Gen Milennial–Durabilitas; 0,026, 0,1611, dan 0,1329 pada Gen Milennial–Ergonomis; 0,1645, 0,4056, dan 0,3104 pada Gen Z–Premium; 0,1416, 0,3763, dan 0,3245 pada Gen Z–Durabilitas; serta 0,0983, 0,3136, dan 0,2315 pada Gen Z–Ergonomis, sehingga konsep Ergonomis terpilih sebagai konsep dengan akurasi prediksi absolut terbaik pada kedua generasi berdasarkan nilai kesalahan terendah, dan selanjutnya menghasilkan kombinasi elemen desain terpilih yang meliputi tipe tutup ulir polos (X1.1) untuk Gen Milennial dan *Quencher 3-way lid* (X1.6) untuk Gen Z, sistem minum *Wide mouth* langsung (X2.5) pada kedua generasi, sistem pegangan *Carry half circle handle* (X3.4) dan *Carry half hexagon handle* (X3.6), bentuk badan atas lurus/silinder (X4.1) pada kedua generasi, bentuk badan tengah silinder lurus (X5.1) dan *Hydro Flask classic* (X5.4), bentuk badan bawah lurus rata (X6.1) pada kedua generasi, bahan material *stainless steel* (X7.1) pada kedua generasi, struktur dinding double-wall (X8.1) pada kedua generasi, skema warna dua-nada (X9.2) dan tiga-nada (X9.3), fitur tanpa aksesoris (X10.6) dan *Strap/lanyard* (X10.2), *finishing glossy* (X11.2) pada kedua generasi, volume kecil 250–500 ml (X12.1) dan sedang 501–800 ml (X12.2), serta gaya desain *Minimalist* (X13.2) dan *cute* (X13.8) yang masing-masing merepresentasikan preferensi Generasi Milennial dan Generasi Z, dengan nilai skor *Kansei* aktual dan hasil



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

prediksi model sebesar 5,589 dan 5,865 pada Gen Milenial serta 4,885 dan 5,555 pada Gen Z.

4. Hasil rancangan konsep desain terpilih dalam bentuk 3D *mockup* tumbler berdasarkan elemen-elemen desain pada konsep Ergonomis yang dihasilkan dari metode *Random Forest*, yang kemudian diterjemahkan ke dalam *mindmap*, *moodboard*, dan 3D *mockup* untuk merepresentasikan preferensi masing-masing segmen generasi melalui pendekatan *Kansei Engineering*, yakni rancangan tumbler bergaya *Minimalist* dengan tutup ulir polos, pegangan *half circle handle*, dan skema warna *duo-tone* krem-hitam yang mencerminkan karakter praktis dan fungsional Generasi Milenial, serta rancangan tumbler bergaya *cute* dengan tutup *Quencher 3-way lid*, pegangan *half hexagon handle* dilengkapi *Strap*, dan skema warna *three-tone* oranye-kuning-biru keunguan yang mencerminkan karakter ekspresif dan *playful* Generasi Z.

5.2 Saran

Berdasarkan hasil dari penelitian ini, terdapat beberapa saran yang dapat dipertimbangkan untuk pengembangan penelitian yang sama selanjutnya, di antaranya:

1. Menerapkan metode pendukung atau pendekatan analisis tambahan dalam proses penentuan kata *Kansei* maupun konsep desain.
2. Menentukan hasil konsep yang bisa tidak menggunakan kuesioner dan wawancara dan menggunakan ulasan konsumen dari data bobot *text mining* yang efisien dan efektif.
3. Mengevaluasi hasil desain menggunakan metode lanjutan seperti *Fuzzy*, *Deep Learning*, *Genetic Algorithm*, *Support Vector Regression*, dan metode lainnya
4. Memperhitungkan aspek biaya dan proses produksi dalam upaya merealisasikan konsep desain tumbler yang diusulkan, agar rancangan yang dihasilkan tidak hanya sesuai dengan preferensi afektif konsumen tetapi juga layak untuk diproduksi secara massal.

© Hak Cipta milik Politeknik Negeri Jakarta



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

5. Memperluas sumber data ulasan konsumen dengan menyertakan beberapa *platform e-commerce* lain maupun media sosial, agar respons afektif yang diperoleh lebih representatif dan beragam.





Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

DAFTAR PUSTAKA

- [1] *Drowning in Plastics – Marine Litter and Plastic Waste Vital Graphics*. United Nations Environment Programme, 2021.
- [2] T. Istirokhatun, S. E. Siahaan, R. R. Gonzales, H. Susanto, A. Malik, and I. Filardli, “Jurnal Presipitasi An Overview of the Utilization of PET Plastic Bottle Waste for Membrane Fabrication,” vol. 22, no. 1, pp. 15–37, 2025.
- [3] M. Indrawaty and L. H. Maulana, “Pro-Environmental Behavior of Indonesian Household Consumers Toward Single-Use Plastic Packaging Application of the Extended Theory of Planned Behavior,” vol. 13, no. 5, pp. 4231–4244, 2025, doi: 10.37641/jimkes.v13i5.4266.
- [4] B. P. Statistik, *Statistik Lingkungan Hidup Indonesia 2021 Energi dan Lingkungan*. 2021.
- [5] ISO, “ISO 18603:2013(en) Packaging and the environment — Reuse.”
- [6] R. F. Ellen MacArthur Foundation, Annette Lendal, Sara Lindeblad Wingstrand, Rob Opsomer, Sander Defruyt, Mats Linder, Leela Dilkes-Hoofman, Sarah Churchill-Slough, Matthew Barber, Piers Young, Ian Banks, *Reuse Rethinking Packaging*. 2019.
- [7] Ingrid and R. F. Boer, “The Effectiveness of Offline and Online Campaigns towards Eco-Friendly Attitude and Intention in Reducing the Use of Plastic Cups,” vol. 13, no. 1, pp. 1–18, 2024, doi: 10.21776/ub.apmba.2024.013.01.1.
- [8] M.-L. Tseng, R. Y. Sujanto, M. Iranmanesh, K. Tan, and A. S. F. Chiu, “Sustainable packaged food and beverage consumption transition in Indonesia: Persuasive communication to affect consumer behavior,” *Resour. Conserv. Recycl.*, vol. 161, p. 104933, 2020, doi: <https://doi.org/10.1016/j.resconrec.2020.104933>.
- [9] N. Made, N. Bunga, and S. Dewi, “SUSTAINABLE LIVING GENERASI



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

MILENIAL DALAM MENANGGULANGI SAMPAH (Sustainable Living Milenial Generation in Treating Waste),” no. April, pp. 32–41, 2023.

- [10] N. N. Fitri, E. Wulandari, M. A. Budiman, F. Pertanian, and U. Padjadjaran, “Mimbar Agribisnis : Jurnal Pemikiran Masyarakat Ilmiah Berwawasan Agribisnis Minat Beli Generasi Z terhadap Tumbler dan Sedotan Ramah Lingkungan Generation Z ’ s Purchase Intention towards Tumbler and Straw Eco-friendly,” vol. 10, no. 1, pp. 702–716, 2024.
- [11] D. P. Restuputri, T. R. Indriani, and I. Masudin, “Cogent Business & Management The effect of logistic service quality on customer satisfaction and loyalty using kansei engineering during the COVID-19 pandemic The effect of logistic service quality on customer satisfaction and loyalty using kansei enginee,” *Cogent Bus. Manag.*, vol. 8, no. 1, 2021, doi: 10.1080/23311975.2021.1906492.
- [12] H. Quan, S. Li, and J. Hu, “Product innovation design based on deep learning and Kansei engineering,” *Appl. Sci.*, vol. 8, no. 12, pp. 1–17, 2018, doi: 10.3390/app8122397.
- [13] D. Weiss, “Development of Methods to Quantify the Affective Response to Food Product Consumption Based on Kansei Engineering and their Application in New Product Development Development of methods to quantify the affective response to food product consumption based ,” 2024.
- [14] Ó. López, C. Murillo, and A. González, “Systematic literature reviews in kansei engineering for product design—a comparative study from 1995 to 2020†,” *Sensors*, vol. 21, no. 19, pp. 1–19, 2021, doi: 10.3390/s21196532.
- [15] W. M. Wang, Z. Li, Z. G. Tian, J. W. Wang, and M. N. Cheng, “This is the Pre-Published Version .,” 2018.
- [16] H. Hassani, C. Beneki, S. Unger, and M. T. Mazinani, “Text Mining in Big Data Analytics,” pp. 1–34, 2020, doi: 10.3390/bdcc4010001.



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

- [17] M. Chiu and K. Lin, “Advanced Engineering Informatics Utilizing text mining and Kansei Engineering to support data-driven design automation at conceptual design stage,” *Adv. Eng. Informatics*, vol. 38, no. 101, pp. 826–839, 2018, doi: 10.1016/j.aei.2018.11.002.
- [18] G. Drakopoulos, I. Giannoukou, P. Mylonas, and S. Sioutas, “Self organizing maps for cultural content delivery,” *Neural Comput. Appl.*, vol. 34, no. 22, pp. 19547–19564, 2022, doi: 10.1007/s00521-022-07376-1.
- [19] F. Giudicepietro *et al.*, “Clustering of Experimental Seismo-Acoustic Events Using Self-Organizing Map (SOM),” vol. 8, no. January, pp. 1–14, 2021, doi: 10.3389/feart.2020.581742.
- [20] W. S. Pasaribu, S. Sormin, and Z. Hazimah, “Descriptive study of tumblr user behavior to support green market development,” vol. 13, no. 3, pp. 435–445, 2025.
- [21] D. S. Wulandari and U. Suhud, “The Role of Tumbler Products In Supporting Sustainable Consumption and Production in Generation Z,” vol. 2, no. 1, pp. 1–10, 2025.
- [22] M. M. J. L. D. J. Assuncao, M. Mulyadi, I. Komunikasi, and U. P. Nasional, “PEMANFAATAN TUMBLER SEBAGAI UPAYA PENGURANGAN SAMPAH PLASTIK DI LINGKUNGAN SEKOLAH DASAR NEGERI 1 UBUNG,” vol. 6, no. 2, pp. 37–48, 2025.
- [23] A. Yamada and S. Hashimoto, “A Text Mining Approach for Automatic Modeling of Kansei Evaluation from Review Texts,” pp. 319–328, 2018.
- [24] M. Hartono, D. N. Prayogo, and I. M. Ronyastra, “Theoretical Issues in Ergonomics Science Kansei engineering with online review mining methodology for robust service design,” *Theor. Issues Ergon. Sci.*, vol. 25, no. 4, pp. 495–520, 2024, doi: 10.1080/1463922X.2023.2261995.
- [25] X. Kang and Z. Zhao, “A study on kansei attraction of products ’ online reviews



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

- by using text mining and kano model,” vol. 18, no. 2, pp. 1–12, 2024, doi: 10.1299/jamdsm.2024jamdsm0010.
- [26] W. Kim, T. Ko, I. Rhiu, and M. H. Yun, “Mining affective experience for a kansei design study on a recliner,” *Appl. Ergon.*, vol. 74, no. January 2018, pp. 145–153, 2019, doi: 10.1016/j.apergo.2018.08.014.
- [27] S. Han, B. D. Williamson, and Y. Fong, “Improving random forest predictions in small datasets from two - phase sampling designs,” *BMC Med. Inform. Decis. Mak.*, vol. 3, pp. 1–9, 2021, doi: 10.1186/s12911-021-01688-3.
- [28] J. Prosise, *Applied machine learning and AI for engineers*. 2023.
- [29] T. S. Lestari, D. Agustin, and N. Sirodj, “Klasifikasi Penipuan Transaksi Kartu Kredit Menggunakan Metode Random Forest,” pp. 160–167.
- [30] N. Syam and R. Kaul, “Random Forest, Bagging, and Boosting of Decision Trees,” *Machine Learning and Artificial Intelligence in Marketing and Sales: Essential Reference for Practitioners and Data Scientists*. Emerald Publishing Limited, p. 0, Mar. 10, 2021. doi: 10.1108/978-1-80043-880-420211006.
- [31] D. A. Devyatkin and O. G. Grigoriev, “Random Kernel Forests,” vol. 10, no. July, 2022.
- [32] A. Agarwal and B. Yu, “MDI +: A Flexible Random Forest-Based *Feature Importance* Framework,” pp. 1–83, 2023.
- [33] N. Indana, A. Diwa, and R. A. Dwi, “Level of Student Awareness in Using Tumbler Water Bottles in an Effort to Reduce the Use of Plastic Bottles,” vol. 10, no. 12, pp. 369–381, 2020.
- [34] H. Kopnina, “Towards Ecological Management: Identifying Barriers and Opportunities in Transition from Linear to Circular Economy,” *Philos. Manag.*, vol. 20, no. 1, pp. 5–19, 2021, doi: 10.1007/s40926-019-00108-x.
- [35] M. L. Astolfi *et al.*, “Reusable Water Bottles : Release of Inorganic Elements ,



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Phthalates , and Bisphenol A in a ‘ Real Use ’ Simulation Experiment,” 2021.

- [36] E. Dusauw, M. Mangantar, and M. H. C. Pandowo, “the Impact of Sensory Marketing on Consumer Behavior At Starbucks Manado,” *J. EMBA J. Ris. Ekon. Manajemen, Bisnis dan Akunt.*, vol. 11, no. 1, pp. 138–146, 2023, doi: 10.35794/emba.v11i1.45579.
- [37] Susilawati, N. Fajrina, and R. Dwi Pramesti, “Peran Elemen Visual Sebagai Strategi Komunikasi Pemasaran Pada Kemasan Produk,” *J. Comput. Sci. Vis. Commun. Des.*, vol. 8, no. 2, pp. 322–332, 2023, doi: 10.55732/jikdiskomvis.v8i2.903.
- [38] R. Iskandar and A. Gustiawan, “Jurnal Desain Komunikasi Kreatif Perancangan Film Dokumenter Lapik Pandan,” vol. 5, no. 2, pp. 119–127, 2023, doi: 10.35134/judikatif.v4i2.1.
- [39] H. C. Chang, K. L. Huang, H. Y. Chen, and C. I. Huang, “Evaluation of packaging form regarding consumers’ sentimental response to bottled beverage containers,” *Appl. Syst. Innov.*, vol. 1, no. 2, pp. 1–13, 2018, doi: 10.3390/asi1020016.
- [40] E. Anwari, E. Meilani, and O. Prasetyowati, “Perancangan Grafis Kemasan Makanan Burayot Sebagai Oleh-Oleh Khas Garut,” *DeKaVe*, vol. 10, no. 2, pp. 12–24, 2018, doi: 10.24821/dkv.v10i2.1990.
- [41] B. Li and Y. Li, “The International Journal of Organizational Innovation Vol 9 Num 3 January 2017 - Section B.,” *Int. J. Organ. Innov.*, vol. 9, no. 3, pp. 1–138, 2017, [Online]. Available: [http://www.ijoi-online.org/attachments/article/51/Final Issue January 2017 - Section B.pdf#page=16](http://www.ijoi-online.org/attachments/article/51/Final%20Issue%20January%202017%20-%20Section%20B.pdf#page=16)
- [42] P. L. Identification, “Analisis Kepatuhan Regulasi Keamanan Pangan Produk Olahan Perikanan Melalui Identifikasi Label Kemasan,” vol. 15, no. 2, pp. 207–220, 2024.



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

- [43] M. Tenggara, T. Maturbongs, M. M. Tanihatu, S. Ronald, and J. Saleky, "T. Maturbongs, M. M. Tanihatu, and S. R. J. Saleky, 'Desain Kemasan dan Label Produk Enbal pada UMKM Nen Te Idar Desa Ngilngof, Maluku Tenggara,' *Jurnal Administrasi Terapan*, vol. 3, no. 1, pp. 168–178, 2024," vol. 3, no. 1, pp. 168–178, 2024.
- [44] R. Istianah, "Pengembangan Desain Kemasan Produk Makanan Merek Krispy Yammy Babeh," *J. Seni dan Desain Serta Pembelajarannya*, vol. 4, no. 1, pp. 33–44, 2022, [Online]. Available: <https://ejournal.upi.edu/index.php/irama/article/download/49133/19959>
- [45] E. T. Maziriri, "Green packaging and green advertising as precursors of competitive advantage and business performance among manufacturing small and medium enterprises in South Africa," *Cogent Bus. Manag.*, vol. 7, no. 1, 2020, doi: 10.1080/23311975.2020.1719586.
- [46] L. K. Ncube, A. U. Ude, E. N. Ogunmuyiwa, R. Zulkifli, and I. N. Beas, "An overview of plasticwaste generation and management in food packaging industries," *Recycling*, vol. 6, no. 1, pp. 1–25, 2021, doi: 10.3390/recycling6010012.
- [47] A. Roihan, P. A. Sunarya, and A. S. Rafika, "Pemanfaatan Machine Learning dalam Berbagai Bidang: Review paper," *IJCIT (Indonesian J. Comput. Inf. Technol.)*, vol. 5, no. 1, pp. 75–82, 2020, doi: 10.31294/ijcit.v5i1.7951.
- [48] J. C. Li, M. Namvar, G. P. Im, and S. Akhlaghpour, "Machine Learning Based Decision-Making: A Sensemaking Perspective," *Australas. J. Inf. Syst.*, vol. 28, pp. 1–22, 2024, doi: 10.3127/ajis.v28.4781.
- [49] D. Chopra and R. Khurana, *Introduction to Machine Learning with Python*. 2023. doi: 10.2174/97898151244221230101.
- [50] K. Kumar, M. H. Syed, S. Bhargava, L. L. Mohakud, M. Serajuddin, and M. Lourens, "Natural Language Processing: Bridging the Gap between Human



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

- Language and Machine Understanding,” in *2024 International Conference on Trends in Quantum Computing and Emerging Business Technologies*, 2024, pp. 1–6. doi: 10.1109/TQCEBT59414.2024.10545250.
- [51] M. N. O. Sadiku, P. A. Adekunle, and J. O. Sadiku, “Applications of Natural Language Processing,” vol. 10, no. 3, 2026.
- [52] A. Rahali, “End-to-End Transformer-Based Models in Textual-Based NLP,” pp. 54–110, 2023.
- [53] Y. Kang, Z. Cai, C.-W. Tan, Q. Huang, and H. Liu, “Natural Language Processing (NLP) in Management Research A Literature Review Natural Language Processing (NLP) in Management Research : A Literature Review,” 2020, doi: 10.1080/23270012.2020.1756939.
- [54] I. Tenney, D. Das, and E. Pavlick, “BERT Rediscovered the Classical NLP Pipeline,” pp. 4593–4601, 2019.
- [55] N. Ahmed, M. A. Khuhro, and M. A. Dootio, “International Journal of Applied Engineering & Technology,” *Rom. Sci. Publ. Ins*, vol. 6, no. 1, pp. 353–360, 2024, [Online]. Available: <https://romanpub.com/resources/ijaet20v5-4-2023-93.pdf>
- [56] C. C. Aggarwal, “Machine Learning for Text: An Introduction BT - Machine Learning for Text,” C. C. Aggarwal, Ed., Cham: Springer International Publishing, 2018, pp. 1–16. doi: 10.1007/978-3-319-73531-3_1.
- [57] A. Algarni and A. Abdulmohsen, “TF-TDA : A Novel Supervised Term Weighting Scheme for,” 2023, doi: <https://doi.org/10.3390/electronics12071632>.
- [58] A. R. Lubis, M. K. M. Nasution, O. S. Sitompul, and E. M. Zamzami, “The effect of the TF-IDF algorithm in times series in forecasting word on social media,” vol. 22, no. 2, pp. 976–984, 2021, doi: 10.11591/ijeecs.v22.i2.pp976-984.



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

- [59] M. Liang and T. Niu, "ScienceDirect ScienceDirect Research on Text Classification Techniques Based on Improved TF- Research on Text Classification Techniques Based on Improved TF- IDF Algorithm and LSTM Inputs IDF Algorithm and LSTM Inputs," *Procedia Comput. Sci.*, vol. 208, pp. 460–470, 2022, doi: 10.1016/j.procs.2022.10.064.
- [60] Z. Jiang, B. Gao, Y. He, Y. Han, P. Doyle, and Q. Zhu, "Text Classification Using Novel Term Weighting Scheme-Based Improved TF-IDF for Internet Media Reports," vol. 2021, no. ii, 2021, doi: 10.1155/2021/6619088.
- [61] M. Umadevi, "DOCUMENT COMPARISON BASED ON TF-IDF METRIC," pp. 1546–1550, 2020.
- [62] R. Das and T. D. Singh, "Multimodal Sentiment Analysis: A Survey of Methods, Trends, and Challenges," *ACM Comput. Surv.*, vol. 55, no. 13, 2023, doi: 10.1145/3586075.
- [63] N. Borisova, E. Karashtranova, and I. Atanasova, "The advances in natural language processing technology and its impact on modern society," *Int. J. Electr. Comput. Eng.*, vol. 15, no. 2, p. 2325, 2025, doi: 10.11591/ijece.v15i2.pp2325-2333.
- [64] Samson Damilola Fabiyi, "A Review of Unsupervised Artificial Neural Networks with Applications," *Int. J. Comput. Appl.*, vol. 181, no. 40, pp. 22–26, 2019, doi: 10.5120/ijca2019918425.
- [65] E. Sarkar, I. Biktasheva, and R. Laraki, "Artificial Neural Networks: Kohonen Self-Organising Maps," University of Liverpool, 2018.
- [66] J. Cong, C. Chen, X. Meng, Z. Xiang, and L. Dong, "Advanced Engineering Informatics Conceptual design of a user-centric smart product-service system using self-organizing map," *Adv. Eng. Informatics*, vol. 55, no. January, p. 101857, 2023, doi: 10.1016/j.aei.2022.101857.
- [67] M. J. Friedel, "Estimation and scaling of hydrostratigraphic units: application of



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

- unsupervised machine learning and multivariate statistical techniques to hydrogeophysical data,” *Hydrogeol. J.*, vol. 24, no. 8, pp. 2103–2122, 2016, doi: 10.1007/s10040-016-1452-5.
- [68] M. Deetz, “Interna tio na l Journa l of M anage me nt Scie nce a nd Bus ine s s Adminis t ra t io n K-Means Clustering of Self-Organizing Maps : An Empirical Study on the Information Content of Self-Classification of Hedge Fund Managers 2 . Methodological Framework,” vol. 5, no. 3, pp. 43–57, 2019, doi: 10.18775/ijmsba.1849-5664-5419.2014.53.1006.
- [69] S. Adhurya and Y. S. Park, “A novel method for predicting ecological interactions with an unsupervised machine learning algorithm,” *Methods Ecol. Evol.*, vol. 15, no. 7, pp. 1247–1260, 2024, doi: 10.1111/2041-210X.14358.
- [70] L. Anh Tu, “Improving Feature Map Quality of SOM Based on Adjusting the Neighborhood Function,” *Sustain. Urban Plan. Des.*, vol. 14, no. 9, pp. 746–757, 2020, doi: 10.5772/intechopen.89233.
- [71] D. Huysmans *et al.*, “Unsupervised learning for mental stress detection exploration of self-organizing maps,” *BIOSIGNALS 2018 - 11th Int. Conf. Bio-Inspired Syst. Signal Process. Proceedings; Part 11th Int. Jt. Conf. Biomed. Eng. Syst. Technol. BIOSTEC 2018*, vol. 4, pp. 26–35, 2018, doi: 10.5220/0006541100260035.
- [72] M. Mohammadagha, “Hyperparameter Optimization Strategies for Tree-Based Machine Learning Models Prediction: A Comparative Study of AdaBoost, Decision Trees, and Random Forest,” *SSRN Electron. J.*, 2025, doi: 10.2139/ssrn.5226457.
- [73] Q. Jia *et al.*, “A machine learning model for predicting consumer preferences of melon flavor and sensory attributes,” *Appl. Food Res.*, vol. 6, no. 1, p. 102000, 2026, doi: 10.1016/j.afres.2026.102000.
- [74] M. S. Efendi, Sarwido, and A. K. Zyen, “Penerapan Algoritma Random Forest



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Untuk Prediksi Penjualan Dan Sistem Persediaan Produk,” *RESOLUSI Rekayasa Tek. Inform. dan Inf.*, vol. 5, no. 1, p. 20, 2024, doi: 10.30865/resolusi.v5i1.2149.

- [75] P. Probst and A. L. Boulesteix, “To tune or not to tune the number of trees in random forest,” *J. Mach. Learn. Res.*, vol. 18, pp. 1–8, 2018.
- [76] G. W. Cha, H. J. Moon, and Y. C. Kim, “Comparison of random forest and gradient boosting machine models for predicting demolition waste based on small datasets and categorical variables,” *Int. J. Environ. Res. Public Health*, vol. 18, no. 16, 2021, doi: 10.3390/ijerph18168530.
- [77] J. Cheng, J. C. M. Dekkers, and R. L. Fernando, “Cross-validation of best linear unbiased predictions of breeding values using an efficient leave-one-out strategy,” *J. Anim. Breed. Genet.*, vol. 138, no. 5, pp. 519–527, 2021, doi: 10.1111/jbg.12545.
- [78] G. I. Austin, I. Pe’er, and T. Korem, “Distributional bias compromises leave-one-out cross-validation,” *Sci. Adv.*, vol. 11, no. 48, pp. 1–10, 2025, doi: 10.1126/sciadv.adx6976.
- [79] R. Hidayat *et al.*, “Implementasi Algoritma Random Forest Regression Untuk Memprediksi Penjualan Produksi di Supermarket,” *Simkom*, vol. 10, no. 1, pp. 101–109, 2025, doi: 10.51717/simkom.v10i1.703.
- [80] Sari Novi Purnama, *Perencanaan Dan Pengembangan Kemasan Kansei Engineering*. Depok: PNJ Press, 2019.
- [81] A. R. Lubis, Y. Y. Lase, D. A. Rahman, and D. Witarsyah, “Improving Spell Checker Performance for Bahasa Indonesia Using Text Preprocessing Techniques with Deep Learning Models,” *Ing. des Syst. d’Information*, vol. 28, no. 5, pp. 1335–1342, 2023, doi: 10.18280/isi.280522.
- [82] T. H. Saputro and A. Hermawan, “The Accuracy Improvement of Text Mining Classification on Hospital Review through The Alteration in The Preprocessing



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

- Stage,” *Int. J. Comput. Inf. Technol.*, vol. 10, no. 4, pp. 140–146, 2021, doi: 10.24203/ijcit.v10i4.138.
- [83] D. Lande and O. Dmytrenko, “Using part-of-speech tagging for building networks of terms in legal sphere,” *CEUR Workshop Proc.*, vol. 2870, pp. 87–97, 2021.
- [84] N. P. Sari, *Perencanaan Dan Pengembangan Kemasan Kansei Engineering*. Depok: PNJ Press, 2019.
- [85] I. Kusmaryono, D. Wijayanti, and H. R. Maharani, “Number of Response Options, Reliability, Validity, and Potential Bias in the Use of the Likert Scale Education and Social Science Research: A Literature Review,” *Int. J. Educ. Methodol.*, vol. 8, no. 4, pp. 625–637, 2022, doi: 10.12973/ijem.8.4.625.
- [86] Sarwinanti and L. P. Sari, “Faktor-Faktor Yang Berhubungan Dengan Kejadian Anemia Gravidarum,” *Univ. Res. Colloquium*, vol. 5, no. 2, pp. 13–24, 2020, [Online]. Available: <https://repository.urecol.org/index.php/proceeding/article/view/1093/1064>
- [87] L. Salma *et al.*, “Visual Kemasan Perhiasan Kawat Tembaga Pernik Elok Sebagai Representasi Produk,” vol. 02, no. 01, pp. 36–46, 2024.
- [88] Databoks and C. M. Annur, “Proporsi Jumlah Transaksi E-commerce Berdasarkan Kelompok Usia (Januari-Desember 2022).” [Online]. Available: <https://databoks.katadata.co.id/teknologi-telekomunikasi/statistik/e3db959da18b8e6/transaksi-e-commerce-konsumen-usia-tua-terus-meningkat-ketimbang-gen-z-dan-milenial>
- [89] “DIGITAL 2025 INDONESIA THE ESSENTIAL GUIDE TO DIGITAL TRENDS,” 2025. [Online]. Available: <https://wearesocial.com/id/blog/2025/02/digital-2025/>
- [90] Plastiks, “Why Younger Generation Demands Sustainable Businesses?” [Online]. Available: <https://www.plastiks.io/blog/younger-generations->



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

demand-sustainable-businesses

- [91] M. Nagamachi, "History of Kansei Engineering and Application of Artificial Intelligence BT - Advances in Affective and Pleasurable Design," W. Chung and C. S. Shin, Eds., Cham: Springer International Publishing, 2018, pp. 357–368.
- [92] E. T. Pardede, K. D. Tania, and M. Afrina, "Knowledge Discovery: Analisis Sentimen dan Emosi WhatsApp Business dengan Machine learning dan Deep Learning," *J. Nas. Teknol. dan Sist. Inf.*, vol. 11, no. 3, pp. 310–318, 2026, doi: 10.25077/teknosi.v11i3.2025.310-318.
- [93] E. Apriani, I. F. Hanif, and F. Oktavianalisti, "Sentiment Analysis of Using TikTok as a Learning Media Using the Naïve Bayes Classifiers Algorithm Analisis Sentimen Penggunaan TikTok Sebagai Media Pembelajaran Menggunakan Algoritma Naïve Bayes Classifier," vol. 4, no. July, pp. 1160–1168, 2024.
- [94] A. Kamal and R. Astri, "Eksplorasi Sentimen Pengguna pada Aplikasi E-Commerce dengan Deep Learning," *J. Teknol. Dan Sist. Inf. Bisnis*, vol. 7, no. 3, pp. 435–441, 2025, doi: 10.47233/jteksis.v7i3.2010.
- [95] W. Bush, J. Callan, and Z. Liu, "Convolutional Neural Networks for So - Matching N-Grams in Ad-hoc Search," pp. 126–134, 2018.
- [96] A. Sinaga and S. P. Nainggolan, "Analisis Perbandingan Akurasi Dan Waktu Proses Algoritma Stemming Arifin-Setiono Dan Nazief-Adriani Pada Dokumen Teks Bahasa Indonesia," *Sebatik*, vol. 27, no. 1, pp. 63–69, 2023, doi: 10.46984/sebatik.v27i1.2072.
- [97] B. Bakiyev, "Method for Determining the Similarity of Text Documents for the Kazakh language, Taking Into Account Synonyms: Extension to TF-IDF," *SIST 2022 - 2022 Int. Conf. Smart Inf. Syst. Technol. Proc.*, pp. 28–30, 2022, doi: 10.1109/SIST54437.2022.9945747.



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

- [98] N. Anisa, “Penerapan Lexicon Based Untuk Analisis Sentimen pada Game PUBG dengan Ekstraksi Fitur TF-IDF,” *J. Pract. Comput. Sci.*, vol. 5, no. 1, pp. 35–41, 2025, doi: 10.37366/jpcs.v5i1.5826.
- [99] L. O. A. Oyinkanola, O. A. Aremu, J. A. Fajemiroye, and S. O. Makinde, “On the physical significance and di-electric response of Castor oil processed in Nigeria as transformer insulating fluid,” vol. VIII, no. 2454, pp. 60–66, 2023, doi: 10.51584/IJRIAS.
- [100] S. Jatmika, M. T. Indriastuti, and G. A. Wafdulloh, “Implementasi Teks Mining Untuk Klasifikasi Buku Berdasarkan Dewey Decimal Clasification (Ddc) Di Perpustakaan Stmik Asia Malang Berbasis Vektor Space Model,” *POSITIF J. Sist. dan Teknol. Inf.*, vol. 4, no. 2, p. 103, 2018, doi: 10.31961/positif.v4i2.567.
- [101] R. A. Puspasari, C. Amelia, B. P. Pangestu, and N. P. Sari, “Perancangan Konsep Desain Kemasan Makanan Kucing Menggunakan Metode Kansei Engineering,” *Performa Media Ilm. Tek. Ind.*, vol. 22, no. 2, p. 143, 2023, doi: 10.20961/performa.22.2.80744.
- [102] N. V. Leuwol, S. Gaspersz, M. S. Tupamahu, and W. Wonmaly, “Karakteristik Kepemimpinan Ideal di Era Generasi Milenial,” vol. 05, no. 02, pp. 4292–4302, 2023.
- [103] T. P. Hidayat, “PERAN PENTING PALET WARNA DAN SEMIOTIKA DALAM,” vol. 5, no. 1, pp. 34–47, 2022.
- [104] M. A.-G. pd. Aqrabawi, “Chromatic Identity in Digital Luxury : The Impact of Colour Psychology on Generation Z and Millennial Female Consumers .,” 2025.
- [105] F. Oktavia, F. Ekonomi, D. A. N. Bisnis, U. I. Negeri, and R. I. Lampung, “DIGITAL DAN PERILAKU KONSUMTIF DI MARKETPLACE SHOPEE DALAM PERSPEKTIF EKONOMI ISLAM (Studi Pada Gen Z di Bandar Lampung) PERSPEKTIF EKONOMI ISLAM (Studi Pada Gen Z di Bandar Lampung),” 2023.



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

- [106] W. Swasty, L. A. Utami, and B. I. Ronggowarsito, "Identifying the Meaning of Colors as Design Information for Design Tools : A Systematic Literature Review," vol. 16, no. 2, pp. 205–228, 2024, doi: 10.5614/j.vad.2024.16.2.5und.sci.2016.48.
- [107] A. R. Pramestiningrum, "KOMUNIKASI VISUAL: FEEDS INSTAGRAM @YUREZALINA DALAM MEMBANGUN SELF-IMAGE," *RELASI J. Penelit. Komun.*, vol. 02, no. 03, pp. 28–38, 2022.





© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

LAMPIRAN

Lampiran 1. Hasil *Web Scraping Text*

Reviews Sampel	Name Person	Rating	Review
Reviews Sampel 1.xlsx	H***i	bintang 5	Respons, persiapan barang oleh dan durasi pengiriman semua cepat bahkan lebih cepat daripada deskripsi. Produk keren banget, bagian luar terbuat dr <i>stainless steel</i> beberapa lapis, sedangkan bagian dalam terbuat dr keramik yang dijamin dapat menahan panas maupun dingin dalam waktu cukup lama. Secara volume 385 ml cukup besar, secara size juga cukup besar serta kokoh dan enak dlm genggam tangan. Secara desain juga keren banget lah..
Reviews Sampel 1.xlsx	H***h	bintang 5	Material: Lapisan keramiknya rapi, tidak berbau, aman untuk panas dan dingin Characteristics: Simple dan praktis. Mudah dibersihkan Design: Minimalis dengan bukaan atas yang lebar, lapisan karer anti bocor. Size: Sesuai untuk dibawa2. Ada tenggangnya tinggal cantol2
Reviews Sampel 1.xlsx	H***y	bintang 5	Produk diterima dengan baik dan lengkap, desain cukup bagus dan rapi. Cocok untuk jadi tumbler kopi, dan gak mudah berbau kopi. Coating ceramic rapi dan berfungsi untuk minuman acid
Reviews Sampel 1.xlsx	N***z	bintang 5	Bahan bagus berkualitas, desain keren bngit aq suka, ukuran pas untuk buat kopi dan wedang rempah2.
Reviews Sampel 1.xlsx	V***f	bintang 5	keren nih mug bener-bener terbuat dari keramik dalamnya, warnanya suka, desainnya apik dengan detail yang menarik, belum dicoba apakah awet utk air panas mudah-mudahan real, thx all
Reviews Sampel 1.xlsx	W***i	bintang 5	Desain: Mulus hitam cool Ukuran: 13oz 380ml Bahan: Keramik, stanlis tapi tutup nya tembus panas Kuat kokoh ga tembus panas tapi tutup nya tembus panas hati2 Kalau untuk dingin bener2 tahan lama
Reviews Sampel 1.xlsx	I***a	bintang 5	bagus & kokoh, keramik coating inner & stainless stell 304 outer, warna birunya cakepp.. sampai tanpa lecet sedikitpun meskipun packagingnya cuma pake bubblewrap selapis.
Reviews Sampel 1.xlsx	K***s	bintang 5	Mug kopi Civago nya bagus bgt. Bahan: Tebal dan keramik bersih Desain: Elegan dan trendy, cocok dipakai kapanpun
Reviews Sampel 1.xlsx	E***i	bintang 5	Udah lama ngincer yang ungu tapi pas kemaren lagi PO, ternyata PO nya ga nyampe 30 hari, cuma semingguan langsung dikirim. Seneng banget, kualitasnya cakep banget, dipake buat ngopi udah pasti enak karna ga ninggalin rasa besi kaya pake stainless, lapisan keramik putihnya di dalam ngebantu bgt. warnanya jangan ditanya, inceran dari lama ungunya gradasi cantik sekali, plus dapet tali jadi enak buat ditentang kemana mana. mantap lah pokoknya, jadi pengen beli warna lain. thank you so much civago ????
Reviews Sampel 1.xlsx	K***x	bintang 5	MasyaAllah Tabarakallah suka bgt sama mug nya.bqkalan jd andalan. Bahannya tebal, dilapis keramik dibagian dalam, jadi gk bau besi, mudah di bersihkan. Pokoknya rekomendasi.best quality .Trims seller
Reviews Sampel 1.xlsx	S***t	bintang 5	Bahannya bagus, ketahanannya sesuai klaimnya aku isi kopi di jam 9.11 besoknya di jam 7.48 masih ada es batunya, padahal udah aku isi ulang sama air biasa terus terus dinginnya awet suka, jadi mau beli yg model lainnya, pokoknya gk bakal nyesel
Reviews Sampel 1.xlsx	H***i	bintang 5	Material: Kokoh dan halus Design: Mewah dengan warna krem coklat yang kalem Size: Pas untuk bekal di meja kantor Characteristics: Betul betul mampu menahan dingin cukup lama
Reviews Sampel 1.xlsx	G***s	bintang 5	Material: Food grade, premium Characteristics: Keren Design: Mewah Size: Cukup, tidak terlalu besar/kecil Kualitas tumbler premium, warna dan design keren, mudah dibawa ke mana-mana dengan lanyard.
Reviews Sampel 1.xlsx	F***u	bintang 5	Material: Food grade, premium Characteristics: Keren Design: Mewah Size: Cukup, tidak terlalu besar/kecil Kualitas tumbler premium, warna dan design keren, mudah dibawa ke mana-mana dengan lanyard.
Reviews Sampel 1.xlsx	U***z	bintang 5	Bahan: Tebal and bagus Karakteristik: Bisa tahan ckp lama buat panas atau dingin Desain: Elegan finising jg rapih halus Ukuran: Cukup buat kopi and minum simple
Reviews Sampel 1.xlsx	X***c	bintang 5	Bahan: Sesuai deskripsi, es batu tahan 10 jam (pengalaman pribadi, buat es kopi jam 9 pagi. Jam 7 malam masih ada sisa es batu) Desain: OK Ukuran: Pas Rekomen utk yg masih bingung cari tumbler dgn es kopi dgn harga dibawah 120rb, dgn kwalitas OK. Karakteristik: Design menarik, pas digenggam
Reviews Sampel 1.xlsx	T***h	bintang 5	Bahan: Barangnya sangat Bagus & bahannusangat kuat ad kemarin di dalam nya Karakteristik: Mencerminkan sebuah tempat minum yg elegan Desain: Desain sangat mewah & elegan. Ukuran: Ukuran sangat pas untuk di bawa bepergian.
Reviews Sampel 1.xlsx	A***p	bintang 5	Bagus dengan harga yang ramah dompet dapat Tumbler dg build quality yg OK. Sudah ceramic coating pula.. dan warna hitamnya terkesan elegan dan tangguh. Nyimpen es dr jam 9 pagi, hanya cair sedikit sampe jam 4 sore.
Reviews Sampel 1.xlsx	Z***e	bintang 5	Barang bagus berkualitas dengan harga yang terjangkau. Seller sangat responsif dan informatif. Barang tidak lama diprosesnya. Barang langsung cepat dikirim. Barang dipacking dengan cukup



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

			Baik dan rapih . Terimakasih seller, shopee dan kurir. Next order disni lagi ya. Bahan: Bagus kombinasi Karakteristik: Awet Desain: Keren Ukuran: Pas
Reviews Sampel 1.xlsx	L***f	bintang 5	Bahan: Kuat dan kokoh Desain: Mewah Ukuran: Sesuai Mungkin karna masih baru jadi ada bau plastik apa keramik? Entahlah, bisa jadi
Reviews Sampel 1.xlsx	B***m	bintang 5	Bahan: Rapih n kokoh Karakteristik: Sangat elegan Desain: Sangat bagus Ukuran: Pas buat keseharian Semoga awet cat nya ga cepet lecet
Reviews Sampel 1.xlsx	M***u	bintang 5	Barang diterima dalam kondisi baik, tidak cacat, packing bagus memakai bubble wrap sehingga produk aman, delivery cepat. Material: Sesuai informasi produk Characteristics: Bagus Design: Menarik Size: Sesuai
Reviews Sampel 1.xlsx	E***m	bintang 5	Bahan: Kokoh Desain: Cakep Ukuran: Ok Karakteristik: Hot & cold Alhamdulillah Paket landing dengan selamat. Asli cakep banget, warnanya jg cakep. Packing dan pengirimannya sangat ok. Seller jg responnya ok. Terima kasih civago, tiktokshop, dan j&t
Reviews Sampel 1.xlsx	Y***r	bintang 5	Pengiriman cepat, tumbler yg pertama ada kerusakan dibagian kuping /cantolan (mngkin saat pengiriman) tapi resellernya baik direpson dengan baik dikirim lagi ganti yg kuping yg rusak Dan ini tumbler gantinya baguss dalamnya cramic suka cocok untuk tmpat kopi bagi yg suka kopi Rekomendasi utk kalian
Reviews Sampel 1.xlsx	B***r	bintang 5	Pesanan sudah sampai, sangat pas dengan barangnya desain sangat bagus dan tidak ada cacat dan pakingannya juga bagus dan rapih, ini kedua kali beli tetap puas dengan barangnya, terimakasih??
Reviews Sampel 1.xlsx	J***x	bintang 5	Material is verybgood, ceramic. Characteristics: I love this, manly, humble in black, solid black Design: Very eye cathing Size: Fit for me. For my esspresso or americano
Reviews Sampel 1.xlsx	N***a	bintang 5	Terimakasih seller dan kurir yg baik sudah diterima dan sudah dipakai mugnya, dibawa bekel air hangat dari pagi jam 6 sampai siang msh hangat, tidakbbau, warna juga soft , keren suka ??
Reviews Sampel 1.xlsx	R***g	bintang 5	Bahan: <i>Stainless steel</i> 304 food grade Karakteristik: Foodgrade Desain: Mug Ukuran: 385ml Barang telah sampai. Pengiriman cepat, respon baik. Pengiriman menggunakan bubble rap serta rapih, Barang dikirimkan sesuai pesanan. Rekomendasi bagi yg mencari tumbler minum atau untuk coffee. Terima kasih
Reviews Sampel 1.xlsx	M***l	bintang 5	kualitas material baik, kuat, tahan dingin gampang dicuci, tidak berbau habis dicuci, warna keren
Reviews Sampel 1.xlsx	Josh	bintang 5	sudah di test, di bawa bawa aman gk tumpah, naik motor naik mobil semua aman... Kopi juga rasanya gk berubah, beneran gk berubah... bahkan pernah simpen kopi sampe Basi, pas di cuci rasa Normal lagi... gk ada berubah kek Stainless
Reviews Sampel 1.xlsx	E***s	bintang 5	Material: material yg mewah dengan harga terjangkau Design: simple tapi elegan Size: pas buat hot/cold black coffee or etc Thanks civago
Reviews Sampel 1.xlsx	C***m	bintang 5	Bahan: Bagus Karakteristik: Baik Desain: Bagus Ukuran: Sesuai Terima kasih seller barang yang di pesan sudah sampai dengan baik dan sesuai, pengiriman sesuai estimasi.
Reviews Sampel 1.xlsx	V***r	bintang 5	Bahan: Bahan ny ok bagus, Karakteristik: Ok juga bgs Desain: Bgs bgt desain nya tumbler nya aku suka bgt Ukuran: Sedang lah buat ngopi mah..
Reviews Sampel 1.xlsx	W***v	bintang 5	Bahan: Muantab Karakteristik: Keren Desain: Ganteng Ukuran: Pas buat dirumah dan di bawa G kecewa CO ini.. Bakal jd favorit dirumah dan di luar rumah
Reviews Sampel 1.xlsx	Y***e	bintang 5	warna ungunya cakep deh.. ga terlalu berat saat dalam keadaan kosong.. model tutup anti bocor.. ketahanan panas ga lama, paling 2-3 jam hanya masih terasa hangat..
Reviews Sampel 1.xlsx	Voidy	bintang 5	bahannya lumayan bagus kok, untuk pemakaian sehari2 cukup lah. walaupun entah kenapa transaksi pertama gagal, akhirnya kebeli pas transaksi kedua
Reviews Sampel 1.xlsx	B***z	bintang 5	Mug nya cantik, mewah, elegan. Semoga tahan panasnya. Next mau pesan tumbler lagi. Terimakasih reseller.
Reviews Sampel 1.xlsx	P***m	bintang 5	Tumblernya cantik,,warna nya soft banget,,datangnya cepet ,,packingan oke
Reviews Sampel 1.xlsx	X***g	bintang 5	Bahan: Best Karakteristik: Bold Ukuran: Compact Desain: Elegant, Simple
Reviews Sampel 1.xlsx	Q***h	bintang 5	Bahan: Bagus Karakteristik: Simpel Desain: Bagus Ukuran: Pas Barang sesuai dengan pesanan Makasih
Reviews Sampel 1.xlsx	B***z	bintang 5	Mug nya cantik, mewah, elegan. Semoga tahan panasnya. Next mau pesan tumbler lagi. Terimakasih reseller.
Reviews Sampel 1.xlsx	K***u	bintang 5	Barang diterima dengan baik, lengkap, dapet harga lumayan agak murah dibanding sebelumnya, ini civago ke dua ku, sayangnya pengiriman terlalu lama
Reviews Sampel 1.xlsx	Q***h	bintang 5	Bahan: Bagus Karakteristik: Simpel Desain: Bagus Ukuran: Pas Barang sesuai dengan pesanan Makasih
Reviews Sampel 1.xlsx	J***q	bintang 5	Bahan: Bagus Banget Karakteristik: Futuristik dan Elegan Desain: Modern Ukuran: Pas Banget di tangan
Reviews Sampel 1.xlsx	A***e	bintang 5	Material: Belum dipakai, tapi sprtnya bagus banget, bisa digunakan u/ berbagai minuman. Characteristics: Cool Design: Cakep beneerrr Size: Pas untuk dibawa ke kantor atau sekolah...
Reviews Sampel 1.xlsx	P***z	bintang 5	Bahan: Kualitas barang ok, dalamnya kramik Desain: Desainnya keren dengan desain tali kualitas ok Ukuran: Pas buat minum kopi Thanks kak semoga awet, terima kasih smoga berkah jualanx
Reviews Sampel 1.xlsx	R***y	bintang 5	Terbaik dan Premium Bahan: Mulus Karakteristik: Elegan Desain: Menarik Ukuran: Standar yg tidak terlalu kecil



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Reviews Sampel 1.xlsx	M***q	bintang 5	Bahan: Bagus Karakteristik: Tahan dingin dan agak kurang tahan panas nya, nggak sesuai yang di iklankan Desain: Agak unik dan menarik Ukuran: Pas untuk 1 kopi dingin, dan kalau mau dibandingkan, = 1 botol teh pucuk
Reviews Sampel 1.xlsx	K***p	bintang 5	Bahan: Tebal,lapis keramik Karakteristik: Sangat kuat Desain: Pas dan unik Ukuran: Cukup simpel
Reviews Sampel 1.xlsx	Y***d	bintang 5	Material: Bagus tapi untuk tutupnya agak kurang rapih detailnya Design: Keren Size: Cukup, tidak terlalu kecil dan tidak terlalu besar
...			
Reviews Sampel 40.xlsx	J***k	bintang 5	packingan rapi, makasih yaaa
Reviews Sampel 40.xlsx	C***l	bintang 5	Bagus dan nyampe dengan aman
Reviews Sampel 40.xlsx	L***i	bintang 5	Benr bener masih tahan panas.
Reviews Sampel 40.xlsx	J***v	bintang 5	Bangut tahan lama panasnya
Reviews Sampel 40.xlsx	Z***w	bintang 5	Ssip lumayan bagus lh ??
Reviews Sampel 40.xlsx	*****	bintang 5	Good sukaaakk syekaliiii
Reviews Sampel 40.xlsx	Sunil	bintang 5	Good quality product.
Reviews Sampel 40.xlsx	Y***r	bintang 3	packing asal saja, kurang aman, nggak pake bubble wrap
Reviews Sampel 40.xlsx	W***u	bintang 5	Sukak.. soalnya unik??
Reviews Sampel 40.xlsx	*****	bintang 5	lucu banget warna nya
Reviews Sampel 40.xlsx	Z***e	bintang 5	Produk sangat memuaskan
Reviews Sampel 40.xlsx	N***h	bintang 5	Baguss bgtttt
Reviews Sampel 40.xlsx	S***g	bintang 4	Barang bagus sesuai pesanan
Reviews Sampel 40.xlsx	N***n	bintang 5	Lucu botol nya
Reviews Sampel 40.xlsx	N***i	bintang 5	mantapppp...
Reviews Sampel 40.xlsx	E***y	bintang 5	Top banget
Reviews Sampel 40.xlsx	P***i	bintang 5	Oke punya dah
Reviews Sampel 40.xlsx	Q***h	bintang 5	Bahan: bgs
Reviews Sampel 40.xlsx	T***S	bintang 5	bagus bgt
Reviews Sampel 40.xlsx	N***i	bintang 5	Mantap
Reviews Sampel 40.xlsx	F***y	bintang 5	Mantap
Reviews Sampel 40.xlsx	n***.	bintang 5	bagus pol
Reviews Sampel 40.xlsx	T***S	bintang 5	Kerennnn
Reviews Sampel 40.xlsx	a***m	bintang 5	amazing
Reviews Sampel 40.xlsx	u***n	bintang 5	baguuss
Reviews Sampel 40.xlsx	b***s	bintang 5	Keren
Reviews Sampel 40.xlsx	T***S	bintang 5	Cakep
Reviews Sampel 40.xlsx	F***r	bintang 5	bagus



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Reviews Sampel 40.xlsx	M***e	bintang 5	Bagus
Reviews Sampel 40.xlsx	*****	bintang 5	Bguss
Reviews Sampel 40.xlsx	P***T	bintang 1	order hitam datengnya beda
Reviews Sampel 40.xlsx	B***d	bintang 5	Good
Reviews Sampel 40.xlsx	R****	bintang 5	Good
Reviews Sampel 40.xlsx	T***S	bintang 5	Oke
Reviews Sampel 40.xlsx	T***S	bintang 5	Good
Reviews Sampel 40.xlsx	j***3	bintang 5	Nice
Reviews Sampel 40.xlsx	V***q	bintang 5	Mantap
Reviews Sampel 40.xlsx	p****	bintang 5	Top
Reviews Sampel 40.xlsx	D***g	bintang 5	oke
Reviews Sampel 40.xlsx	V***t	bintang 5	Sipp
Reviews Sampel 40.xlsx	S***u	bintang 5	Bagus
Reviews Sampel 40.xlsx	R***s	bintang 5	ok
Reviews Sampel 40.xlsx	k***2	bintang 5	Ok
Reviews Sampel 40.xlsx	T***S	bintang 5	Ok
Reviews Sampel 40.xlsx	r***a	bintang 1	Ko ga dingin yaa, padahal saya udh beli yg ke 2. Ini ori ga sih?
Reviews Sampel 40.xlsx	A***n	bintang 5	Pengiriman cepat dan barangnya bagus semoga awet
Reviews Sampel 40.xlsx	U***f	bintang 5	Brg sesuai foto ,tp kurang cocok buat minuman dingin soalnya mulut nya kecil susah masuk es batu
Reviews Sampel 40.xlsx	KADEK	bintang 5	kualitas barangnya bagus worth it lah ??
Reviews Sampel 40.xlsx	Y***e	bintang 5	Bagus tahan panas 12 jam
Reviews Sampel 40.xlsx	A***v	bintang 1	Maaf ini paket udah di Terima,tapi paket nya kotak nya sudah terbuka dan rusak seperti udah di buka sendiri entah siapa yg membuka tolong di lacak siapa pelaku nya.



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Lampiran 2. Source Code Text Mining atau NLP

```
import re
import argparse
import pandas as pd

# -----
# TAHAP 1 : LOAD & CLEANING
# -----

def rating_to_int(raw):
    """'bintang 5' -> 5; kembalikan None bila tak ada angka."""
    m = re.search(r"(\d)", str(raw))
    return int(m.group(1)) if m else None

# pola dikompilasi sekali (efisiensi)
_URL_RE = re.compile(r"http\S+|www\.\S+")
_HTML_RE = re.compile(r"<[^>+>")
_KEEP_RE = re.compile(r"[^a-z\s]") # setelah lowercase: sisakan huruf &
spasi
_SPACE_RE = re.compile(r"(\s+)")

def clean_text(text):
    """Case folding + buang URL/HTML/angka/tanda baca + rapikan spasi."""
    text = str(text).lower()
    text = _URL_RE.sub("", text)
    text = _HTML_RE.sub("", text)
    text = _KEEP_RE.sub("", text) # angka & tanda baca ikut terbuang
    text = _SPACE_RE.sub(" ", text).strip()
    return text

def stage1_load_clean(path, sep, encoding, review_col, rating_col):
    """Muat CSV, buang baris kosong, ubah rating ke int, bersihkan teks."""
    df = pd.read_csv(path, sep=sep, encoding=encoding)
    df = df.dropna(subset=[review_col]).reset_index(drop=True)

    df["rating_int"] = [rating_to_int(r) for r in df[rating_col]]
    df["clean"] = [clean_text(t) for t in df[review_col]]

    # buang baris yang jadi kosong setelah cleaning
    df = df[df["clean"].str.len() > 0].reset_index(drop=True)
    return df

# -----
# TAHAP 2 : NORMALISASI & PHRASE MATCHING (longest-match n-gram)
# -----
#
# AFFECTIVE_PHRASES = {
#     # trigram
#     "sesuai dengan deskripsi": "sesuai_deskripsi",
#     "tidak mudah bocor": "anti_bocor",
#     "tidak gampang bocor": "anti_bocor",
#     # bigram
#     "sesuai deskripsi": "sesuai_deskripsi",
#     "tahan panas": "tahan_panas",
#     "tahan dingin": "tahan_dingin",
#     "tahan lama": "tahan_lama",
#     "awet banget": "tahan_lama",
#     "anti bocor": "anti_bocor",
#     "tidak bocor": "anti_bocor",
#     "tidak tumpah": "anti_tumpah",
#     "anti tumpah": "anti_tumpah",
#     "tidak berbau": "tidak_berbau",
#     "tidak bau": "tidak_berbau",
#     "tidak berisik": "tidak_berisik",
#     "mudah dibawa": "mudah_dibawa",
#     "mudah digenggam": "mudah_digenggam",
#     "mudah dibersihkan": "mudah_dibersihkan",
#     "bahan tebal": "bahan_tebal",
#     "bahan kokoh": "bahan_kokoh",
#     "bahan premium": "bahan_premium",
#     "kualitas bagus": "kualitas_bagus",
#     "kualitas baik": "kualitas_bagus",
#     "harga terjangkau": "harga_terjangkau",
#     "harga murah": "harga_terjangkau",
#     "ramah lingkungan": "ramah_lingkungan",
#     # --- NEGASI -> ANTONIM POSITIF (kata dasar negatif, dinegasikan jadi baik)
#     "tidak bau": "anti_bau",
#     "tidak berbau": "anti_bau",
#     "tidak bocor": "anti_bocor",
#     "tidak mudah bocor": "anti_bocor",
#     "tidak tumpah": "anti_tumpah",
#     "tidak berembun": "anti_embun",
#     "tidak ngembun": "anti_embun",
#     "tidak embun": "anti_embun",
#     "tidak licin": "anti_licin",
#     "tidak berat": "ringan",
#     "tidak menyesal": "puas",
#     "tidak mengecewakan": "puas",
#     "tidak kecewa": "puas",
#     "tidak mengecewa": "puas",
#     "tanpa cacat": "mulus",
#     "tidak cacat": "mulus",
#     "tidak penyok": "mulus",
#     "tidak murahan": "berkualitas",
#     "tidak kaleng": "berkualitas",
#     # --- NEGASI -> TOKEN NEGATIF (kata dasar positif, dinegasikan jadi buruk) -
#     "tidak sesuai": "tidak_sesuai",
#     "kurang sesuai": "tidak_sesuai",
#     "tidak bagus": "tidak_bagus",
#     "kurang bagus": "tidak_bagus",
#     "tidak baik": "tidak_bagus",
#     "tidak rapi": "tidak_rapi",
#     "kurang rapi": "tidak_rapi",
#     "tidak awet": "tidak_awet",
#     "tidak kuat": "tidak_kokoh",
#     "tidak kokoh": "tidak_kokoh",
#     "tidak memuaskan": "tidak_puas",
#     "tidak puas": "tidak_puas",
#     # Negasi + kata sifat POSITIF (tak terdaftar eksplisit) -> token "tidak_kata".
#     # Jaring pengaman umum agar negasi tak hilang; hanya untuk kata
#     # bermuatan Kansei.
#     NEGATION_WORDS = {"tidak", "kurang", "bukan", "tanpa", "tak"}
#     NEGATABLE_POSITIF = {
#         "bagus", "baik", "rapi", "kuat", "kokoh", "sesuai", "awet", "tebal",
#         "mulus", "halus", "puas", "mantap", "layak", "pas", "cocok", "rapat",
#         "memuaskan", "presisi", "oke", "wangi", "ringan", "bersih", "aman",
#     }
#     # Sinonim level UNIGRAM -> bentuk Kansei kanonik (disatukan maknanya).
#     UNIGRAM_SYNONYM = {
#         "kuat": "kokoh",
#         "kekar": "kokoh",
#         "sturdy": "kokoh",
#         "kukuh": "kokoh",
#         "elok": "cantik",
#         "ayu": "cantik",
#         "menawan": "cantik",
#         "imut": "lucu",
#         "gemas": "lucu",
#         "adem": "dingin",
#         "sejuk": "dingin",
#     }
#     # panjang n-gram maksimum yang dicoba (trigram)
#     _MAX_NGRAM = 3
#     # -----
#     # NORMALISASI : koreksi singkatan & kata tidak baku -> bentuk baku (EYD)
#     # -----
#     def load_normalized_word_dict(path):
#         """Muat dataset mapping term (slang/tidak_baku : bentuk_baku)."""
#         import csv as _csv
#         d = {}
#         with open(path, encoding="utf-8") as fh:
#             for row in _csv.DictReader(fh):
#                 s, baku = row["slang"].strip(), row["baku"].strip()
#                 if s and baku:
#                     d[s] = baku
#         return d
#     def normalized_term(document, normalized_word_dict):
#         """Seragamkan tiap term dgn kamus; token frasa underScore dilewati."""
```



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
return [
    normalized_word_dict[term] if term in normalized_word_dict else term
    for term in document
]
```

```
def match_affective_phrases(tokens):
```

```
    """Phrase matching longest-match: trigram -> bigram -> unigram.
```

```
    Memindai token kiri-ke-kanan. Pada tiap posisi, coba cocokkan frasa
    TERPANJANG dulu (3 kata), lalu 2 kata; bila cocok -> 1 token underScore
    dan
    lompat kata-kata itu. Bila tak ada, ambil unigram (dengan pemetaan
    sinonim
```

```
    ke bentuk Kansei kanonik, mis. 'kuat' -> 'kokoh')."""
    result = []
```

```
    i, n = 0, len(tokens)
```

```
    while i < n:
```

```
        matched = False
```

```
        # coba n-gram terpanjang dulu (longest-match): 3, lalu 2
```

```
        for size in range(_MAX_NGRAM, 1, -1):
```

```
            if i + size <= n:
```

```
                gram = " ".join(tokens[i:i + size])
```

```
                if gram in AFFECTIVE_PHRASES:
```

```
                    result.append(AFFECTIVE_PHRASES[gram])
```

```
                    i += size
```

```
                    matched = True
```

```
                    break
```

```
            if matched:
```

```
                continue
```

```
        # jaring pengaman negasi umum: negasi + kata sifat positif ->
```

```
        tidak_kata>
```

```
        if (tokens[i] in NEGATION_WORDS and i + 1 < n
```

```
            and tokens[i + 1] in NEGATIVE_POSITIVE):
```

```
            result.append("tidak_" + tokens[i + 1])
```

```
            i += 2
```

```
            continue
```

```
        # unigram: petakan sinonim -> kanonik bila ada, selain itu apa adanya
```

```
        result.append(UNIGRAM_SYNONYM.get(tokens[i], tokens[i]))
```

```
        i += 1
```

```
    return result
```

```
def stage2_normalize_phrasematch(clean_text_str, normalized_word_dict):
```

```
    """String bersih -> tokenisasi -> normalisasi term (EYD) -> phrase
```

```
    matching."""
```

```
    tokens = clean_text_str.split() # tokenisasi
```

```
    tokens = normalized_term(tokens, normalized_word_dict) # EYD (gak
```

```
    >tidak dst)
```

```
    tokens = match_affective_phrases(tokens) # n-gram longest-match
```

```
    return tokens
```

```
# -----
```

```
# TAHAP 3 : STOPWORD REMOVAL (tanpa whitelist)
```

```
# -----
```

```
from Sastrawi.StopWordRemover.StopWordRemoverFactory import
```

```
StopWordRemoverFactory
```

```
# more_stopword : tambahan di luar bawaan Sastrawi, khusus data ulasan
```

```
ini.
```

```
# Berisi kata spam/tidak jelas, penanda field berbahasa Inggris yang ikut
```

```
# terbawa dari struktur ulasan, sapaan/filler toko, dan partikel obrolan.
```

```
MORE_STOPWORD = {
```

```
    # penanda field bawaan struktur ulasan (Material, Characteristics: dst)
```

```
    "material", "characteristics", "characteristic", "simple", "size",
```

```
    "color", "type", "item", "product", "review", "reviews", "sampel",
```

```
    # sapaan / filler toko & penutup ulasan
```

```
    "sis", "gan", "kak", "min", "admin", "seller", "shop", "toko",
```

```
    "terima", "kasih", "makasih", "thanks", "thank", "you", "ya", "yaa",
```

```
    "yah", "deh", "sih", "nih", "kok", "kan", "dong", "lah", "loh", "lho",
```

```
    "aja", "ajah", "saja", "juga", "banget", "bgt", "pokoknya", "intinya",
```

```
    # kata obrolan minim makna
```

```
    "nya", "ny", "yg", "dgn", "krn", "jd", "jadi", "trus", "terus",
```

```
    "buat", "bikin", "punya", "udah", "sdh", "blm", "belum",
```

```
    "moga", "semoga", "mohon", "tolong", "guna", "pakai", "pake",
```

```
    # spam / tak jelas
```

```
    "wkwk", "wkwwk", "haha", "hehe", "hihi", "xixi", "hmm", "hmmm",
```

```
    "oke", "ok", "okeh", "sip", "siap", "nan", "pun", "se", "ku", "mu",
```

```
    # kata Inggris umum yang terbawa (bukan Kansei berbahasa Indonesia)
```

```
    "good", "the", "and", "best", "next", "fast", "free", "top", "no",
```

```
    "but", "nice", "full", "very", "buy", "all", "is", "wow", "far",
```

```
    "its", "not", "for", "time", "pic", "pict", "up", "out", "over",
```

```
    "star", "hot", "cool", "love", "cute", "soft", "live", "chat",
    "plus", "set", "body", "navy", "ice", "seal", "lock", "slip", "wrap",
    # huruf tunggal / token sangat pendek tak bermakna
    "n", "i", "a", "u", "e", "an",
```

```
    # kata fungsional yang lolos (bukan Kansei, minim makna deskriptif)
```

```
    "kalau", "kalo", "mau", "kali", "mana", "apa", "biar", "dulu",
```

```
    "sini", "cara", "tadi", "kira", "sama", "satu", "atas", "coba",
```

```
    "baru", "hari", "pagi", "sore", "awal", "jauh", "luar", "dalam",
```

```
    "nama", "ribu", "masa", "soal", "asal", "kena", "ikut", "cari",
```

```
    # filler & interjeksi obrolan
```

```
    "sus", "sok", "po", "pol", "pol", "tes", "test", "amin", "guys",
```

```
    "ber", "gada", "ah", "ih", "eh", "huhu", "wah", "gue", "tuh", "iya",
```

```
    "mah", "si", "mas", "bang", "joss", "luv", "sia", "pero", "foto",
```

```
    "idd", "oz",
```

```
    # noise frekuensi-rendah: elongasi & typo (varian salah tulis kata lain)
```

```
    "puolll", "barangyaaa", "mantabbb", "mantab", "puasss", "bagussssss",
```

```
    "bguss", "cakepppp", "teball", "youu", "topp", "nex", "tik", "tok",
```

```
    "gud", "recomen", "rekomend", "allhamdullilah", "ledingding", "kasi",
```

```
    "poto", "looknya", "thumblernya", "kurirny", "kulitas", "matrial",
```

```
    "stailless", "stanlies", "stainless", "stainlis", "kramik", "standart",
```

```
    "vacum", "gliter", "dek", "sat", "rap", "euy", "req", "babang", "rama",
```

```
    "diraguin", "kilas", "puoll", "mantaap", "okeey", "okeee",
```

```
    # brand / nama tempat / nama diri yang lolos
```

```
    "tanjung", "pinang", "java", "stanley", "neko", "aqua", "cosmic",
```

```
    "leopard", "army", "owner", "stanlee", "stenlis",
```

```
    # kata Inggris non-afektif (bukan deskriptor Kansei)
```

```
    "made", "value", "food", "grade", "service", "reel", "dent", "grip",
```

```
    "reguler", "standard", "non", "keychain", "size", "type", "item",
```

```
    # fragmen / bentuk / objek non-Kansei
```

```
    "drat", "segitiga", "metal", "umum", "dewasa", "pribadi", "utama",
```

```
    # noise pasca-stem (muncul setelah stemming, tak tertangkap stopwords)
```

```
    "kilas", "kasi", "wrapnya", "sama", "baru", "pasti", "nongkrong",
```

```
    "finishing", "tekstur", "jinjing", "serius", "murmer", "thumbler",
```

```
    "tumbler", "mksih", "wajar", "khusus",
```

```
}
```

```
_STOPWORDS = set(StopWordRemoverFactory().get_stop_words()) |
MORE_STOPWORD
```

```
def remove_stopwords(tokens):
```

```
    """Buang token yang ada di daftar stopwords (Sastrawi + more_stopword).
```

```
    Token frasa (underScore) aman karena tak sama dengan entri stopwords."""
```

```
    return [t for t in tokens if t not in _STOPWORDS]
```

```
# -----
```

```
# TAHAP 4 : STEMMING (Sastrawi)
```

```
# -----
```

```
#
```

```
import os
```

```
import json
```

```
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
```

```
_STEMMER = StemmerFactory().create_stemmer()
```

```
def build_term_dict(unique_terms, cache_path=None):
```

```
    """Bangun term_dict {term: bentuk_dasar} untuk SEMUA term unik di
```

```
    korpus.
```

```
- Stemming hanya dilakukan sekali per term unik (bukan tiap kemunculan),
```

```
  karena Sastrawi lambat; hasilnya di-cache ke disk.
```

```
- Token frasa ber-underScore dipetakan ke DIRINYA SENDIRI (tidak distem),
```

```
  agar tetap utuh sebagai kata Kansei dan agar term_dict[term] tak
```

```
  KeyError.
```

```
    """
```

```
    cache = {}
```

```
    if cache_path and os.path.exists(cache_path):
```

```
        try:
```

```
            with open(cache_path, encoding="utf-8") as fh:
```

```
                cache = json.load(fh)
```

```
        except Exception:
```

```
            cache = {}
```

```
    todo = [w for w in unique_terms if w not in cache]
```

```
    if todo:
```

```
        for w in todo:
```

```
            if "_" in w: # token frasa: jangan dipotong imbuhan
```

```
                cache[w] = w
```

```
            else:
```

```
                stem = _STEMMER.stem(w)
```

```
                # koreksi akar non-baku Sastrawi -> bentuk dasar EYD
```



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

cache[w] = STEM_CORRECTION.get(stem, stem)
if cache_path:
    with open(cache_path, "w", encoding="utf-8") as fh:
        json.dump(cache, fh, ensure_ascii=False)
return cache

def get_stemmed_term(document, term_dict):
    """Kembalikan tiap term ke bentuk dasar via term_dict (pola sesuai
    metodologi).
    term_dict dijamin memuat semua term dokumen, jadi lookup langsung
    aman."""
    return [term_dict[term] for term in document]

STEM_CORRECTION = {
    # peluluhan p/t/k/s yang terpotong keliru
    "muas": "puas", # memuaskan/kepuasan -> (Sastrawi: muas) -> puas
    "nyimpan": "simpan",
    "nyuci": "cuci",
    "ngambil": "ambil",
    "nyari": "cari",
    "nyoba": "coba",
    "ngirim": "kirim",
    "mukul": "pukul",
    "nukar": "tukar",
    # varian salah tulis yang lolos ke stem
    "penyot": "penyok",
    "penyokk": "penyok",
    "pengirimanya": "kirim",
    "pengiriman": "kirim",
    "kualits": "kualitas",
    "kualias": "kualitas",
    "dingn": "dingin",
    "dinginn": "dingin",
    "panass": "panas",
    "baguss": "bagus",
    "cantikk": "cantik",
}

# -----
# TAHAP 5 : POS TAGGING (Part-of-Speech) untuk bahasa Indonesia
# -----
ADJ_OVERRIDE = {
    "elegant", "kokoh", "mewah", "awet", "aman", "adem", "sejuk", "nyaman",
    "keren", "simpler", "minimalis", "estetik", "estetis", "solid", "presisi",
    "rapat", "premium", "halus", "kekinian", "klasik", "kuat", "cantik",
    "lucu", "unik", "bagus", "tebal", "tipis", "murah", "mahal", "cepat",
    "lambat", "dingin", "panas", "hangat", "ringan", "berat", "besar",
    "kecil", "rapi", "bersih", "wangi", "harum", "indah", "imut", "gemas",
    "praktis", "modern", "berkualitas", "memuaskan", "mantap", "puas",
    "licin", "mulus", "ergonomis", "fungsional", "ringkas", "cakep",
    "oke", "stylish", "kekar", "kukuh", "anggun", "menawan", "apik",
    "pas", "cocok", "sesuai", "lengkap", "penyok", "cacat", "lecet",
}

# kelas kata yang dipertahankan sebagai kandidat Kansei (kata sifat + frasa)
KANSEI_POS_KEEP = {"ADJ"}

# afiks untuk rekonstruksi bentuk berimbuhan saat lookup ke kamus UD
_POS_PREF = ["", "me", "mem", "men", "meng", "meny", "ber", "be", "di",
    "ter", "pe", "pen", "peng", "per"]
_POS_SUF = ["", "kan", "i", "an", "nya"]
_POS_SKIP = {"PROPN", "PUNCT", "SYM", "X", "NUM"}

def build_ud_pos_dict(conllu_dir, out_path):
    """Bangun kamus kata->UPOS dari korpus UD (CoNLL-U). Disimpan ke
    out_path."""
    import glob
    from collections import Counter, defaultdict
    wt = defaultdict(Counter)
    for path in glob.glob(os.path.join(conllu_dir, "*.conllu")):
        for line in open(path, encoding="utf-8"):
            line = line.strip()
            if not line or line.startswith("#"):
                continue
            cols = line.split("\t")
            if len(cols) < 4 or "-" in cols[0] or "." in cols[0]:
                continue
            wt[cols[1].lower()][cols[3]] += 1
    pos_dict = {w: c.most_common(1)[0][0] for w, c in wt.items()}

    with open(out_path, "w", encoding="utf-8") as fh:
        json.dump(pos_dict, fh, ensure_ascii=False)
        return pos_dict

def _emb_backoff(stem, pos_dict, w2v):
    """OOV: pinjam tag mayoritas dari tetangga terdekat di ruang
    embedding."""
    from collections import Counter
    if w2v is None or stem not in w2v.wv:
        return None
    tags = Counter()
    for nb, _ in w2v.wv.most_similar(stem, topn=15):
        t = _dict_lookup(nb, pos_dict)
        if t:
            tags[t] += 1
    return tags.most_common(1)[0][0] if tags else None

def pos_tag_word(stem, pos_dict, w2v):
    """Tag satu kata: gazetteer ADJ -> kamus -> embedding -> PROPN
    fallback."""
    if "_" in stem: # token frasa afektif = kata sifat
        return "ADJ"
    if stem in ADJ_OVERRIDE: # gazetteer domain (prioritas)
        return "ADJ"
    t = _dict_lookup(stem, pos_dict) # dictionary (UD) + rekonstruksi
    imbuhan
    if t:
        return t
    t = _emb_backoff(stem, pos_dict, w2v) # word embedding (Word2Vec)
    if t:
        return t
    t = _dict_lookup(stem, pos_dict, allow_skip=True) # fallback akhir (mis.
    PROPN)
    return t or "X"

def build_pos_map(unique_terms, pos_dict, w2v, cache_path=None):
    """Bangun & cache peta kata->UPOS (tipe-level) utk seluruh term unik."""
    pos_map = {}
    if cache_path and os.path.exists(cache_path):
        with open(cache_path, encoding="utf-8") as fh:
            pos_map = json.load(fh)
        todo = [w for w in unique_terms if w not in pos_map]
        if todo:
            for w in todo:
                pos_map[w] = pos_tag_word(w, pos_dict, w2v)
    if cache_path:
        with open(cache_path, "w", encoding="utf-8") as fh:
            json.dump(pos_map, fh, ensure_ascii=False)
    # gazetteer override selalu diterapkan (juga utk entri lama di cache)
    for w in ADJ_OVERRIDE:
        if w in pos_map:
            pos_map[w] = "ADJ"
    return pos_map

def pos_tag_tokens(tokens, pos_map):
    """Beri label UPOS tiap token (lookup tipe-level)."""
    return [pos_map.get(t, "X") for t in tokens]

def filter_by_pos(tokens, tags, keep=KANSEI_POS_KEEP):
    """Pertahankan hanya token dgn POS pada set keep (default: ADJ)."""
    return [t for t, p in zip(tokens, tags) if p in keep]

# -----
# TAHAP 6 : FILTERING (buang kata generik/non-Kansei yang lolos POS)

```

```

with open(out_path, "w", encoding="utf-8") as fh:
    json.dump(pos_dict, fh, ensure_ascii=False)
    return pos_dict

def _dict_lookup(stem, pos_dict, allow_skip=False):
    """Cari tag via kamus UD; bila stem tak ada, REKONSTRUKSI imbuhan."""
    from collections import Counter
    if stem in pos_dict and (pos_dict[stem] not in _POS_SKIP or allow_skip):
        return pos_dict[stem]
    tags = Counter()
    for p in _POS_PREF:
        for s in _POS_SUF:
            c = p + stem + s
            if c in pos_dict and pos_dict[c] not in _POS_SKIP:
                tags[pos_dict[c]] += 1
    return tags.most_common(1)[0][0] if tags else None

def _emb_backoff(stem, pos_dict, w2v):
    """OOV: pinjam tag mayoritas dari tetangga terdekat di ruang
    embedding."""
    from collections import Counter
    if w2v is None or stem not in w2v.wv:
        return None
    tags = Counter()
    for nb, _ in w2v.wv.most_similar(stem, topn=15):
        t = _dict_lookup(nb, pos_dict)
        if t:
            tags[t] += 1
    return tags.most_common(1)[0][0] if tags else None

def pos_tag_word(stem, pos_dict, w2v):
    """Tag satu kata: gazetteer ADJ -> kamus -> embedding -> PROPN
    fallback."""
    if "_" in stem: # token frasa afektif = kata sifat
        return "ADJ"
    if stem in ADJ_OVERRIDE: # gazetteer domain (prioritas)
        return "ADJ"
    t = _dict_lookup(stem, pos_dict) # dictionary (UD) + rekonstruksi
    imbuhan
    if t:
        return t
    t = _emb_backoff(stem, pos_dict, w2v) # word embedding (Word2Vec)
    if t:
        return t
    t = _dict_lookup(stem, pos_dict, allow_skip=True) # fallback akhir (mis.
    PROPN)
    return t or "X"

def build_pos_map(unique_terms, pos_dict, w2v, cache_path=None):
    """Bangun & cache peta kata->UPOS (tipe-level) utk seluruh term unik."""
    pos_map = {}
    if cache_path and os.path.exists(cache_path):
        with open(cache_path, encoding="utf-8") as fh:
            pos_map = json.load(fh)
        todo = [w for w in unique_terms if w not in pos_map]
        if todo:
            for w in todo:
                pos_map[w] = pos_tag_word(w, pos_dict, w2v)
    if cache_path:
        with open(cache_path, "w", encoding="utf-8") as fh:
            json.dump(pos_map, fh, ensure_ascii=False)
    # gazetteer override selalu diterapkan (juga utk entri lama di cache)
    for w in ADJ_OVERRIDE:
        if w in pos_map:
            pos_map[w] = "ADJ"
    return pos_map

def pos_tag_tokens(tokens, pos_map):
    """Beri label UPOS tiap token (lookup tipe-level)."""
    return [pos_map.get(t, "X") for t in tokens]

def filter_by_pos(tokens, tags, keep=KANSEI_POS_KEEP):
    """Pertahankan hanya token dgn POS pada set keep (default: ADJ)."""
    return [t for t, p in zip(tokens, tags) if p in keep]

# -----
# TAHAP 6 : FILTERING (buang kata generik/non-Kansei yang lolos POS)

```



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
# -----
#
GENERIC_BLACKLIST = {
    # kata benda logistik / objek / bahan
    "packing", "packagingnya", "kurir", "kurirnya", "design", "plastik",
    "material", "abang", "kardus", "dus", "bubblewrap", "kurr", "ekspedisi",
    "paket", "resi", "gosend", "grab", "jne", "kurr",
    # sapaan / interjeksi / penutup
    "selamat", "sukses", "trimakasih", "terimakasih", "makasih",
    "alhamdulillah",
    "amin", "sip",
    # adverbial / kata fungsi / intensifier yang lolos
    "langsung", "overall", "super", "anti", "habis", "pertama", "minus",
    "nyata", "benar", "salah", "lumayan", "kira", "agak", "cukup", "sekali",
    "banget", "sangat", "lebih", "paling", "terlalu",
    # kata kerja generik yg lolos
    "kirim", "beli", "pesan", "datang", "terima", "pakai", "coba", "bawa",
    # kata Inggris yang lolos sebagai ADJ (bukan Kansei berbahasa Indonesia)
    "high", "well", "strong", "eye", "blink", "good", "nice", "best",
    "cool", "soft", "hard", "fast", "slow", "big", "small", "new", "old",
    "love", "like", "happy", "real", "fix", "ori", "fragile", "safe",
    # noise pasca-stem (muncul setelah stemming, tak tertangkap stopword
    # pra-stem)
    "kilas", "kasi", "wrapnya", "sama", "baru", "pasti", "nongkrong",
    "finishing", "tekstur", "jinjing", "serius", "murmer", "thumber",
    "tumbler", "mkasih", "wajar", "khusus", "utama", "umum", "dewasa",
    "pribadi", "drat", "segitiga", "metal", "kilasnya",
    # store / logistik / promo & noise sisa yang lolos ke kandidat
    "store", "delivery", "reseller", "dpet", "makas", "maksih", "pak",
    "seler", "gratis", "laris", "bpa", "betul", "kaya", "lain", "mirip",
    "makasih", "masyaallah", "sabar", "sopan", "jujur", "teliti", "pak",
}

# nama warna -> atribut objektif, bukan deskriptor afektif (opsional dibuat)
COLOR_WORDS = {
    "pink", "hitam", "emas", "biru", "putih", "ungu", "merah", "hijau",
    "kuning", "abu", "coklat", "cokelat", "navy", "tosca", "pastel", "cream",
    "krem", "silver", "gold", "maroon", "fanta", "mocca", "peach", "lavender",
    "ijo", "merah", "kuning", "hijau",
}

def filter_Kansei(tokens, drop_colors=True, min_len=3):
    """Buang kata generik/non-Kansei. Token frasa underScore selalu
    disimpan."""
    out = []
    for t in tokens:
        if "_" in t: # frasa afektif: selalu simpan
            out.append(t)
            continue
        if t in GENERIC_BLACKLIST:
            continue
        if drop_colors and t in COLOR_WORDS:
            continue
        if len(t) < min_len: # buang token sangat pendek
            continue
        out.append(t)
    return out

# -----
# TAHAP 7 : TF-IDF (bobot kepentingan tiap kata Kansei)
# -----
#

import math

def compute_tfidf_manual(list_of_token_lists, min_df=5, max_df=0.5):
    """Hitung TF-IDF per-kata secara MANUAL. Kembalikan daftar baris
    ringkasan.

    - min_df : kata harus muncul di >= sekian dokumen (buang kata super
    langka).
    - max_df : kata di > proporsi dokumen ini diabaikan (terlalu umum/tak
    informatif).
    """
    from collections import Counter
    N = len(list_of_token_lists)
    df = Counter() # document frequency
    freq_total = Counter() # total kemunculan (jumlah semua TF)
    for toks in list_of_token_lists:
        seen = set()
        for t in toks:
            freq_total[t] += 1
            seen.add(t)
        for t in seen:
            df[t] += 1

    max_df_count = max(df * N
    rows = []
    for t, df_t in df.items():
        if df_t < min_df or df_t > max_df_count:
            continue # filter kata terlalu langka / terlalu umum
        idf = math.log10(N / df_t) # IDF = log10(N / df)
        tf = freq_total[t] / df_t # TF rata-rata per dokumen pemuat
        tfidf = tf * idf # TF-IDF = TF x IDF
        rows.append({
            "kata": t,
            "is_frasa": int("_" in t),
            "freq_total": freq_total[t], # = jumlah seluruh TF
            "df": df_t, # jumlah dokumen pemuat
            "N": N, # jumlah dokumen
            "tf": round(tf, 6), # freq_total / df
            "idf": round(idf, 6), # log10(N / df)
            "tfidf": round(tfidf, 6), # tf * idf (dapat dicek manual)
        })
    rows.sort(key=lambda r: r["tfidf"], reverse=True)
    return rows, N

# -----
# TAHAP 8 : ANALISIS SENTIMEN (Lexicon-based)
# -----
#

# positif
"bagus": 2, "baik": 2, "keren": 2, "mantap": 3, "cantik": 2, "lucu": 2,
"elegan": 2, "mewah": 2, "unik": 1, "puas": 3, "suka": 2, "senang": 2,
"awet": 2, "kokoh": 2, "kuat": 2, "tebal": 1, "aman": 2, "nyaman": 2,
"rapi": 2, "bersih": 1, "praktis": 2, "ringan": 1, "cepat": 2,
"murah": 2, "terjangkau": 2, "berkualitas": 3, "premium": 2, "halus": 1,
"mulus": 2, "presisi": 2, "estetik": 2, "estetis": 2, "simpler": 1,
"minimalis": 1, "modern": 1, "ergonomis": 2, "cakep": 2, "apik": 2,
"imut": 2, "gemas": 2, "wangi": 1, "harum": 1, "indah": 2, "stylish": 2,
"sleek": 1, "glossy": 1, "sporty": 1, "futuristik": 1, "gagah": 1,
"mungil": 1, "anggun": 2, "menawan": 2, "istimewa": 2, "memuaskan": 3,
"ramah": 1, "kedap": 1, "rapat": 1, "segar": 1, "lengkap": 1, "tepat": 1,
"kekinian": 1, "tahan": 1, "lega": 1, "muat": 1, "pas": 1, "cocok": 2,
"sesuai": 2, "asli": 1, "orisinal": 1, "padat": 1, "bahagia": 2,

# negatif
"penyok": -3, "cacat": -3, "rusak": -3, "lecek": -2, "kecewa": -3,
"bocor": -3, "tumpah": -2, "retak": -3, "pecah": -3, "jelek": -2,
"tipis": -1, "rapuh": -2, "ringkih": -2, "lemah": -2, "longgar": -1,
"licin": -1, "bau": -2, "aneh": -1, "sulit": -2, "ribet": -2,
"lama": -1, "lambat": -2, "mahal": -1, "rawan": -2, "rentan": -2,
"berisik": -2, "sempit": -1, "panas": 0, "dingin": 0,

# skor sentimen TOKEN FRASA (hasil phrase matching / negasi)
PHRASE_SENTIMENT = {
    "sesuai_deskripsi": 2, "tahan_panas": 2, "tahan_dingin": 2,
    "tahan_lama": 3, "anti_bocor": 3, "anti_tumpah": 2, "anti_embun": 2,
    "anti_licin": 2, "anti_bau": 2, "bahan_tebal": 2, "bahan_kokoh": 2,
    "bahan_premium": 2, "kualitas_bagus": 3, "harga_terjangkau": 2,
    "mudah_dibawa": 2, "mudah_digenggam": 2, "mudah_dibersihkan": 2,
    "ramah_lingkungan": 1, "tidak_berisik": 2,
    # token negasi bermakna negatif
    "tidak_sesuai": -3, "tidak_bagus": -2, "tidak_rapi": -2, "tidak_awet": -2,
    "tidak_kokoh": -2, "tidak_puas": -3, "tidak_presisi": -1, "tidak_rapat": -1,
    "tidak_aman": -2, "tidak_pas": -1, "tidak_cocok": -2, "tidak_tebal": -1,
    "tidak_mulus": -2, "tidak_baik": -2, "tidak_bersih": -1,
}

def build_sentiment_lexicon():
    """Gabungkan leksikon domain + frasa menjadi satu peta kata->skor."""
    lex = dict(EMBEDDED_SENTIMENT)
    lex.update(PHRASE_SENTIMENT)
    return lex

# kata sentimen umum tambahan utk skor DOKUMEN (bukan kandidat
Kansei,
# tapi membantu menilai polaritas kalimat ulasan mentah)
DOC_SENTIMENT_EXTRA = {
    "suka": 2, "recommended": 2, "rekomendasi": 2, "oke": 1, "ok": 1,
    "memuaskan": 3, "mantap": 3, "mantul": 3, "top": 2, "joss": 2,
}
```



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
"jelek": -2, "buruk": -3, "mengecewakan": -3, "kecewa": -3,
"menyesal": -3, "parah": -3, "gagal": -3, "rugi": -2, "zank": -3,
"komplain": -2, "lambat": -2, "lemot": -2, "telat": -2,
```

```
def build_doc_lexicon():
    """Leksikon utk skor sentimen DOKUMEN (domain + frasa + kata sentimen
    umum)."""
    lex = build_sentiment_lexicon()
    lex.update(DOC_SENTIMENT_EXTRA)
    return lex
```

```
def Score_document_sentiment(raw_text, doc_lexicon, tokenizer):
    """Skor sentimen satu ulasan = JUMLAH skor leksikon tiap kata (lexicon-
    based).
```

```
    toks = tokenizer(str(raw_text)).lower()
    Score = 0
    for w in toks:
        Score += doc_lexicon.get(w, 0)
    if Score > 0:
        label = "positif"
    elif Score < 0:
        label = "negatif"
    else:
        label = "netral"
    return Score, label
```

```
def lexicon_Score(term, lexicon):
    """Skor sentimen kata dari leksikon (kriteria: tulis sentimen bila
    ditemukan).
    Bila tak ada, kembalikan 0 (netral)."""
    if term in lexicon:
        return lexicon[term]
    return 0
```

```
def compute_word_sentiment(list_of_token_lists, ratings, lexicon,
    corpus_mean,
    pos_min=4, neg_max=2):
    """Hitung sentimen tiap kata Kansei: skor leksikon + sentimen dokumen +
    frekuensi per kelas rating (positif/negatif).
```

```
    Mengikuti kriteria bag-of-words: telusuri tiap dokumen, naikkan
    akumulator
    tiap kata. Kelas dokumen dari rating: positif bila rating >= pos_min,
    negatif bila rating <= neg_max, sisanya netral. Hasil per kata:
    freq_total, df, mean_rating, rating_polarity, skor_Leksikon,
    freq_pos, freq_neg, polarity_ratio, sentiment (skor numerik), sentimen
    (label).
```

```
    from collections import Counter, defaultdict
    freq = Counter()
    rating_sum = defaultdict(float) # akumulator sentimen dokumen (rating)
    df = Counter()
    freq_pos = Counter() # kemunculan di ulasan positif (rating tinggi)
    freq_neg = Counter() # kemunculan di ulasan negatif (rating rendah)
    for toks, r in zip(list_of_token_lists, ratings):
        seen = set()
        for t in toks:
            freq[t] += 1 # naikkan nilai (bag of words)
            if r is not None:
                rating_sum[t] += r # akumulasi sentimen dokumen
                if r >= pos_min:
                    freq_pos[t] += 1
                elif r <= neg_max:
                    freq_neg[t] += 1
        seen.add(t)
    for t in seen:
        df[t] += 1
```

```
    result = {}
    for t in freq:
        # mean_rating berbasis seluruh kemunculan kata
        mean_rating = (rating_sum[t] / freq[t]) if freq[t] > 0 else corpus_mean
        rating_polarity = mean_rating - corpus_mean
        lex = lexicon_Score(t, lexicon)
        # polarity_ratio = proporsi kemunculan positif dari (pos + neg)
        denom = freq_pos[t] + freq_neg[t]
        polarity_ratio = (freq_pos[t] / denom) if denom > 0 else 1.0
```

```
# skor sentiment numerik: [skor leksikon] diarahkan oleh polaritas
dokumen
# (positif bila dominan di ulasan baik, negatif bila dominan di ulasan
```

```
buruk)
base = abs(lex) if lex != 0 else 1.0
sentiment_num = base if polarity_ratio >= 0.5 else -base
# label akhir
if lex < 0 or rating_polarity <= -0.30:
    label = "negatif"
elif lex > 0 or rating_polarity >= 0.02:
    label = "positif"
else:
    label = "netral"
result[t] = {
    "freq": freq[t],
    "df": df[t],
    "mean_rating": round(mean_rating, 4),
    "rating_polarity": round(rating_polarity, 4),
    "skor_leksikon": lex,
    "freq_pos": freq_pos[t],
    "freq_neg": freq_neg[t],
    "polarity_ratio": round(polarity_ratio, 4),
    "sentiment": round(sentiment_num, 4),
    "sentimen": label,
}
```

```
return result
```

```
# -----
# TAHAP 9 : TABEL KANSEI FINAL + PENGELOMPOKAN KATEGORI
# -----
```

```
KANSEI_CATEGORY = {
    # ===== KATA KANSEI SEJATI : EMOTIONAL FEELING =====
    "cantik": "emotional", "elegant": "emotional", "mewah": "emotional",
    "lucu": "emotional", "keren": "emotional", "unik": "emotional",
    "imut": "emotional", "gemas": "emotional", "anggun": "emotional",
    "menawan": "emotional", "cakep": "emotional", "apik": "emotional",
    "kece": "emotional", "ciamik": "emotional", "manis": "emotional",
    "gagah": "emotional", "stylish": "emotional", "sleek": "emotional",
    "sporty": "emotional", "futuristik": "emotional", "indah": "emotional",
    "estetik": "emotional", "estetis": "emotional", "kekinian": "emotional",
    "puas": "emotional", "senang": "emotional", "suka": "emotional",
    "bahagia": "emotional", "nyaman": "emotional", "mantap": "emotional",
    "bagus": "emotional", "istimewa": "emotional", "spesial": "emotional",
    "favorit": "emotional", "kecewa": "emotional", "sedih": "emotional",
    "khawatir": "emotional", "tidak_bagus": "emotional",
    "tidak_baik": "emotional", "tidak_puas": "emotional",
    # ===== ORGANOLEPTIK : persepsi indrawi =====
    "dingin": "organoleptik", "panas": "organoleptik", "hangat": "organoleptik",
    "wangi": "organoleptik", "harum": "organoleptik", "bau": "organoleptik",
    "enak": "organoleptik", "segar": "organoleptik", "cair": "organoleptik",
    "basah": "organoleptik", "halus": "organoleptik", "kasar": "organoleptik",
    "licin": "organoleptik", "pekat": "organoleptik", "didih": "organoleptik",
    "terang": "organoleptik", "gelap": "organoleptik", "tajam": "organoleptik",
    "anti_bau": "organoleptik", "anti_embun": "organoleptik",
    # ===== STRUKTUR : bentuk & konstruksi fisik =====
    "kokoh": "struktur", "kuat": "struktur", "tebal": "struktur",
    "tipis": "struktur", "rapat": "struktur", "kedap": "struktur",
    "ringan": "struktur", "berat": "struktur", "padat": "struktur",
    "besar": "struktur", "kecil": "struktur", "mungil": "struktur",
    "lebar": "struktur", "longgar": "struktur", "sempit": "struktur",
    "ramping": "struktur", "panjang": "struktur", "tinggi": "struktur",
    "elastis": "struktur", "rapuh": "struktur", "ringkih": "struktur",
    "keras": "struktur", "solid": "struktur", "mulus": "struktur",
    "penyok": "struktur", "cacat": "struktur", "rusak": "struktur",
    "lecek": "struktur", "retak": "struktur", "pecah": "struktur",
    "rembes": "struktur", "miring": "struktur", "rentan": "struktur",
    "rawan": "struktur", "bahan_tebal": "struktur", "bahan_kokoh": "struktur",
    "tidak_kokoh": "struktur", "tidak_tebal": "struktur",
    "tidak_rapat": "struktur", "tidak_mulus": "struktur",
    # ===== FITUR : fungsi/kemampuan produk =====
    "aman": "fitur", "praktis": "fitur", "mudah": "fitur",
    "ergonomis": "fitur", "fungsional": "fitur", "ramah": "fitur",
    "ramah_lingkungan": "fitur", "awet": "fitur", "andal": "fitur",
    "presisi": "fitur", "tenteng": "fitur", "maksimal": "fitur",
    "bersih": "fitur", "muat": "fitur", "lega": "fitur",
    "anti_bocor": "fitur", "anti_tumpah": "fitur", "anti_licin": "fitur",
    "tahan_panas": "fitur", "tahan_dingin": "fitur", "tahan_lama": "fitur",
    "mudah_dibawa": "fitur", "mudah_digenggam": "fitur",
    "mudah_dibersihkan": "fitur", "tidak_berisik": "fitur",
    "tidak_aman": "fitur", "tidak_awet": "fitur", "tidak_presisi": "fitur",
    # ===== ELEMEN DESAIN : tampilan/rupa OBJEKTIF =====
```



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

"minimalis": "elemen_desain", "simpler": "elemen_desain",
"modern": "elemen_desain", "transparan": "elemen_desain",
"glossy": "elemen_desain", "insight": "berkualitas": "insight", "premium": "insight",
"murah": "insight", "mahal": "insight", "terjangkau": "insight",
"ekonomis": "insight", "eksklusif": "insight", "sepadan": "insight",
"tidak_rapi": "elemen_desain",
# ===== INSIGHT : nilai/harga/ekspektasi & kesesuaian =====
"bagus_kualitas": "insight", "berkualitas": "insight", "premium": "insight",
"murah": "insight", "mahal": "insight", "terjangkau": "insight",
"ekonomis": "insight", "eksklusif": "insight", "sepadan": "insight",
"asli": "insight", "orisinal": "insight", "sesuai": "insight",
"cocok": "insight", "pas": "insight", "tepat": "insight",
"lengkap": "insight", "jelas": "insight", "cepat": "insight",
"lama": "insight", "lambat": "insight", "sulit": "insight",
"aneh": "insight", "ribet": "insight",
"harga_terjangkau": "insight", "sesuai_deskripsi": "insight",
"kualitas_bagus": "insight", "bahan_premium": "insight",
"tidak_sesuai": "insight", "tidak_pas": "insight", "tidak_cocok": "insight",
# ===== tambahan pemetaan (sebelumnya 'lainnya') =====
"baik": "insight", "oke": "insight", "standar": "insight",
"normal": "insight", "penting": "insight", "minim": "insight",
"utuh": "struktur", "rata": "struktur", "kosong": "struktur",
"penuh": "struktur", "gede": "elemen_desain",
"kotor": "organoleptik", "sehat": "fitur", "bebas": "fitur",
"susah": "fitur", "telat": "insight", "parah": "emotional",
"sakit": "emotional", "netral": "lainnya",
}

```

```

def categorize_Kansei(term):
    """Kelompokkan kata Kansei; default 'lainnya' bila tak terpetakan."""
    return KANSEI_CATEGORY.get(term, "lainnya")

```

```

# kata Kansei sejati = kata sifat emotional feeling (kategori 'emotional')
EMOTIONAL_FEELING = {w for w, c in KANSEI_CATEGORY.items() if c == "emotional"}

```

```

def is_emotional_feeling(word):
    """True bila kata termasuk kata sifat emotional feeling (Kansei sejati)."""
    return word in EMOTIONAL_FEELING

```

```

# -----
# OUTPUT SIAP-SOM : vektor fitur kata & matriks doc-term & metadata
# -----
#

```

```

def build_Kansei_feature_table(ranking, sentimen_map):
    """Tabel fitur per kata Kansei utk SOM tahap-1, plus kategori & jenis."""
    rows = []
    for r in ranking:
        w = r["kata"]
        s = sentimen_map.get(w, {})
        rows.append({
            "kata": w,
            "is_frasa": r["is_frasa"],
            "jenis": "emotional_feeling" if is_emotional_feeling(w)
            else "karakteristik_desain",
            "kategori": categorize_Kansei(w),
            # ---- vektor fitur SOM tahap-1 ----
            "tfidf": r["tfidf"],
            "skor_sentimen": s.get("skor_leksikon", 0),
            "frekuensi": r["freq_total"],
            "rating_polarity": s.get("rating_polarity", 0.0),
            # ---- atribut pendukung ----
            "df": r["df"],
            "idf": r["idf"],
            "mean_rating": s.get("mean_rating", ""),
            "sentimen": s.get("sentimen", "netral"),
        })
    return rows

```

```

def build_doc_term_matrix(list_of_token_lists, ratings, doc_sentiments,
                           Kansei_vocab, min_Kansei=2):
    """Matriks doc-term (responden x kata Kansei) utk SOM tahap-2.

    Hanya menyertakan responden dgn jumlah kata Kansei UNIK >=
    min_Kansei
    (default 2 => 'lebih dari 1'). Mengembalikan (baris_matriks, metadata,
    vocab).
    """

```

```

vocab = sorted(Kansei_vocab)
vidx = {w: i for i, w in enumerate(vocab)}
matrix_rows = []
meta_rows = []
for doc_id, (toks, rating, dsent) in enumerate(
    zip(list_of_token_lists, ratings, doc_sentiments)):
    Kansei_in_doc = [t for t in toks if t in vidx]
    uniq = set(Kansei_in_doc)
    if len(uniq) < min_Kansei: # syarat: > 1 kata Kansei
        continue
    vec = [0] * len(vocab)
    for t in Kansei_in_doc:
        vec[vidx[t]] += 1 # hitung kemunculan (bag of words)
    matrix_rows.append(vec)
    meta_rows.append({
        "doc_id": doc_id,
        "rating": rating,
        "skor_sentimen_doc": dsent[0],
        "sentimen_doc": dsent[1],
        "jumlah_kata_Kansei": len(Kansei_in_doc),
        "kata_Kansei_unik": len(uniq),
    })
return matrix_rows, meta_rows, vocab

```

```

# -----
# MAIN
# -----

def main():
    p = argparse.ArgumentParser(description="Pipeline Kansei - bertahap")
    p.add_argument("--input", required=True)
    p.add_argument("--sep", default=",")
    p.add_argument("--encoding", default="utf-8-sig")
    p.add_argument("--review-col", default="Review")
    p.add_argument("--rating-col", default="Rating")
    p.add_argument("--output-dir", default="/outputs")
    p.add_argument("--norm-dict", default="normalisasi_dict.csv",
        help="dataset mapping term tidak baku -> baku")
    p.add_argument("--min-token-freq", type=int, default=3,
        help="hanya stem kata dgn frekuensi >= nilai ini (gate)")
    p.add_argument("--ud-dir", default="ud_id",
        help="folder korpus UD Indonesian (*.conllu) utk dictionary POS")
    p.add_argument("--w2v-model", default="w2v_tumbler.model",
        help="model Word2Vec utk backoff POS kata OOV")
    p.add_argument("--min-df", type=int, default=5,
        help="TF-IDF: kata harus muncul di >= sekian dokumen")
    p.add_argument("--max-df", type=float, default=0.5,
        help="TF-IDF: abaikan kata di > proporsi dokumen ini")
    args = p.parse_args()

    print("[1] Load & Cleaning ...")
    df = stage1_load_clean(args.input, args.sep, args.encoding,
        args.review_col, args.rating_col)
    print(f" {len(df):,} ulasan siap (setelah cleaning).")
    print()

    print("[2] Normalisasi (koreksi singkatan/tidak baku -> EYD) & Tokenisasi ...")
    normalized_word_dict = load_normalized_word_dict(args.norm_dict)
    print(f" kamus normalisasi (normalized_word_dict): "
        f" {len(normalized_word_dict):,} entri "
        f" (publik colloquial-lexicon + tambahan domain)")

    # jalankan: tokenisasi -> normalisasi term -> phrase matching (n-gram)
    from collections import Counter
    changed = Counter()
    tokens_list = []
    for ct in df["clean"]:
        raw = ct.split()
        norm = normalized_term(raw, normalized_word_dict)
        for a, b in zip(raw, norm):
            if a != b:
                changed[f"{a} -> {b}"] += 1
        tokens_list.append(match_affective_phrases(norm))
    df["tokens"] = tokens_list

    print(f" total token dinormalisasi (EYD): {sum(changed.values()):,}")
    print(" contoh koreksi normalisasi tersering:")
    for pair, c in changed.most_common(6):
        print(f" {pair:24s} {c:6d}")
    print()

```



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

print("[2b] Phrase matching (longest-match: trigram -> bigram -> unigram)
...")
# hitung token frasa multi-kata (ber-underScore) hasil bi/trigram
phrase_counter = Counter()
for toks in df["tokens"]:
    for t in toks:
        if "_" in t:
            phrase_counter[t] += 1

print(f" total token frasa multi-kata terdeteksi: "
      f"{sum(phrase_counter.values());}")
print(f" rincian frasa afektif:")
for tok, c in phrase_counter.most_common():
    print(f" {tok:20s} {c:5d}")
print()

# ukur jumlah token SEBELUM stopword removal
tok_before = sum(len(t) for t in df["tokens"])

print("[3] Stopword removal (Sastrawi + more_stopword, tanpa whitelist)
...")
removed = Counter()
new_tokens = []
for toks in df["tokens"]:
    kept = []
    for t in toks:
        if (t not in _STOPWORDS) or ("_" in t):
            kept.append(t)
        else:
            removed[t] += 1
    new_tokens.append(kept)
df["tokens"] = new_tokens

tok_after = sum(len(t) for t in df["tokens"])
print(f" token sebelum : {tok_before};")
print(f" token sesudah : {tok_after};")
print(f" {tok_before - tok_after};")
print(f" {100*(tok_before-tok_after)/tok_before:.1f}% terbuang")
print()
print(f" 15 stopwords paling banyak dibuang:")
for w, c in removed.most_common(15):
    print(f" {w:10s} {c:6d}")
print()
print(f" Contoh token akhir (frasa underScore tetap utuh):")
for toks in df["tokens"]:
    if any("_" in t for t in toks):
        print(f" {toks[:14]}")
        break
print()

# -- TAHAP 4 : STEMMING -----
os.makedirs(args.output_dir, exist_ok=True)
print("[4] Stemming Sastrawi (kata -> bentuk dasar) ...")

# kumpulkan SEMUA term unik (termasuk token frasa) -> bangun term_dict
unique_terms = set()
for toks in df["tokens"]:
    unique_terms.update(toks)
print(f" term unik di korpus : {len(unique_terms);}")

cache_path = os.path.join(args.output_dir, "stem_cache.json")
term_dict = build_term_dict(unique_terms, cache_path)
print(f" term_dict (cache) : {len(term_dict);} entri")

# terapkan get_stemmed_term ke tiap dokumen
tok_before_stem = sum(len(t) for t in df["tokens"])
df["tokens"] = [get_stemmed_term(toks, term_dict) for toks in
df["tokens"]]

# ukur penyatuan: berapa kata unik menyusun
unique_after = set()
for toks in df["tokens"]:
    unique_after.update(toks)
print(f" kata unik: {len(unique_terms);} -> {len(unique_after);}")
print(f" (disatukan {len(unique_terms) - len(unique_after);} bentuk)")
print()

# demonstrasi penyatuan variasi morfologis
print(f" Contoh pemotongan imbuhan (kata -> bentuk dasar):")
contoh = ["keramiknya", "pengiriman", "warnanya", "dibersihkan",
"kebocoran", "memuaskan", "berbau", "pesanan", "kemasan"]
shown = 0
for w in contoh:
    if w in term_dict and term_dict[w] != w:
        print(f" {w:14s} -> {term_dict[w]}")
        shown += 1
if shown == 0:
    print(f" (contoh tak ada di korpus aktif)")
    print()
print(f" Contoh token akhir setelah stemming:")
for toks in df["tokens"]:
    if any("_" in t for t in toks):
        print(f" {toks[:14]}")
        break
print()

# -- TAHAP 5 : POS TAGGING -----
print("[5] POS Tagging Indonesia (dictionary UD + word embedding) ...")

# (1) DICTIONARY: kamus kata->UPOS dari korpus UD Indonesian-GSD
pos_dict_path = os.path.join(args.output_dir, "ud_pos_dict.json")
if os.path.exists(pos_dict_path):
    pos_dict = json.load(open(pos_dict_path, encoding="utf-8"))
elif os.path.isdir(args.ud_dir):
    pos_dict = build_ud_pos_dict(args.ud_dir, pos_dict_path)
else:
    pos_dict = {}
print(f" dictionary UD (kata->UPOS): {len(pos_dict);} entri")

# (2) WORD EMBEDDING: Word2Vec terlatih di korpus (backoff OOV)
w2v = None
if args.w2v_model and os.path.exists(args.w2v_model):
    try:
        from gensim.models import Word2Vec
        w2v = Word2Vec.load(args.w2v_model)
        print(f" word embedding Word2Vec : vocab {len(w2v.wv);}")
    except Exception as e:
        print(f" (word embedding dilewati: {e})")

# bangun peta POS tipe-level utk semua term unik (di-cache)
unique_terms = set()
for toks in df["tokens"]:
    unique_terms.update(toks)
pos_map = build_pos_map(unique_terms, pos_dict, w2v,
os.path.join(args.output_dir, "pos_corpus.json"))
print(f" token ditag (tipe-level) : {len(pos_map);}")

# beri label POS tiap token, lalu pertahankan ADJ (kandidat Kansei)
from collections import Counter
pos_dist = Counter()
tok_before_pos = 0
kept_tokens = []
for toks in df["tokens"]:
    tags = pos_tag_tokens(toks, pos_map)
    for p in tags:
        pos_dist[p] += 1
    tok_before_pos += len(toks)
    kept_tokens.append(filter_by_pos(toks, tags))
df["tokens_full"] = df["tokens"] # token lengkap (sebelum filter POS)
df["tokens"] = kept_tokens # token terfilter (ADJ)

tok_after_pos = sum(len(t) for t in df["tokens"])
print(f" distribusi UPOS seluruh token (NN~NOUN, VB~VERB, JJ~ADJ,
RB~ADV):")
for p, c in pos_dist.most_common():
    mark = "<" if p in KANSEI_POS_KEEP else ""
    print(f" {p:7s} {c:7d}{mark}")
    print(f" token: {tok_before_pos;} -> {tok_after_pos;}")
    print(f" (disaring ke ADJ, "
f"{100*(tok_before_pos-tok_after_pos)/tok_before_pos:.1f}%
dibuang)")
print()
print(f" Contoh kata Kansei (ADJ) hasil POS tagging:")
seen = Counter()
for toks in df["tokens"]:
    for t in toks:
        seen[t] += 1
for w, c in seen.most_common(10):
    print(f" {w:18s} {c:5d}")
print()

# -- TAHAP 6 : FILTERING -----
print("[6] Filtering (buang kata generik/non-Kansei yang lolos POS) ...")
tok_before_f = sum(len(t) for t in df["tokens"])
removed_f = Counter()
new_toks = []
for toks in df["tokens"]:
    kept = filter_Kansei(toks)

```



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

for t in toks:
    if t not in kept and "_" not in t:
        removed_f[t] += 1
    new_toks.append(kept)
df["tokens"] = new_toks
tok_after_f = sum(len(t) for t in df["tokens"])

print(f" token: {tok_before_f,} -> {tok_after_f,} "
      f"f({tok_before_f - tok_after_f,}) non-Kansei dibuang:")
print(" contoh kata non-Kansei yang dibuang:")
for w, c in removed_f.most_common(12):
    print(f" {w:16s} {c:5d}")
print()
print(" >> Kandidat kata Kansei final (ADJ + frasa, sudah bersih:-)")
seen2 = Counter()
for toks in df["tokens"]:
    for t in toks:
        seen2[t] += 1
for w, c in seen2.most_common(15):
    tag = "[frasa]" if "_" in w else "[adj]"
    print(f" {tag} {w:18s} {c:5d}")
print()

# -- TAHAP 7 : TF-IDF (MANUAL) -----
print("[7] TF-IDF manual (bobot kepentingan kata Kansei = TF x IDF) ...")
ranking, N = compute_tfidf_manual(
    df["tokens"].tolist(), min_df=args.min_df, max_df=args.max_df)
print(f" dokumen (ulasan) N : {N,}")
print(f" kata Kansei lolos min_df={args.min_df}/max_df={args.max_df}:"
      f"f({len(ranking),})")

# simpan ranking TF-IDF (semua komponen agar bisa diverifikasi manual)
import csv as _csv
rank_path = os.path.join(args.output_dir, "07_tfidf_ranking.csv")
with open(rank_path, "w", encoding="utf-8", newline="") as fh:
    w = _csv.DictWriter(fh, fieldnames=["kata", "is_frasa", "freq_total",
                                       "df", "N", "tf", "idf", "tfidf"])
    w.writeheader()
    w.writerows(ranking)

print()
print(" 20 kata Kansei berbobot TF-IDF tertinggi:")
print(f" { 'kata':18s } { 'freq':>6s } { 'df':>5s } { 'TF':>7s } "
      f"f('IDF':>7s) {'TF-IDF':>8s}")
for r in ranking[:20]:
    tag = "" if r["is_frasa"] else ""
    print(f" {tag} {r['kata']:18s} {r['freq_total']:6d} {r['df']:5d} "
          f"f({r['tf']:7.3f} {r['idf']:7.4f} {r['tfidf']:8.4f})")
print()

# contoh verifikasi manual pada satu kata
ex = ranking[len(ranking) // 2] # ambil satu kata dari tengah ranking
print(" Verifikasi manual (silakan hitung sendiri:-)")
print(f" kata = {ex['kata']}")
print(f" freq_total = {ex['freq_total']} (jumlah seluruh TF)")
print(f" df = {ex['df']} (dokumen pemuat)")
print(f" N = {ex['N']}")
print(f" TF = freq_total / df = {ex['freq_total']} / {ex['df']} "
      f"f={ex['tf']:.6f}")
print(f" IDF = log10(N / df) = log10({ex['N']} / {ex['df']}) "
      f"f={ex['idf']:.6f}")
print(f" TF-IDF = TF x IDF = {ex['tf']:.6f} x "
      f"f({ex['idf']:.6f}) = {ex['tfidf']:.6f}")
print(f" (tabel mencatat tfidf = {ex['tfidf']:.6f})")
print()
print(f" -> ranking lengkap disimpan: {rank_path}")
print()

# -- TAHAP 8 : ANALISIS SENTIMEN (lexicon-based) -----
print("[8] Analisis sentimen lexicon-based (skor leksikon + sentimen
dokumen) ...")
lexicon = build_sentiment_lexicon()
corpus_mean = df["rating_int"].dropna().mean()
print(f" leksikon domain + frasa : {len(lexicon),} entri")
print(f" rata-rata rating korpus : {corpus_mean:.3f}")

sentimen = compute_word_sentiment(
    df["tokens"].tolist(), df["rating_int"].tolist(), lexicon, corpus_mean)

# gabungkan ke ranking TF-IDF -> tabel kata Kansei + sentimen
for r in ranking:
    s = sentimen.get(r["kata"], {})
    r["skor_leksikon"] = s.get("skor_leksikon", 0)
    r["mean_rating"] = s.get("mean_rating", round(corpus_mean, 4))

r["rating_polarity"] = s.get("rating_polarity", 0.0)
r["sentimen"] = s.get("sentimen", "netral")
r["freq_pos"] = s.get("freq_pos", 0)
r["freq_neg"] = s.get("freq_neg", 0)
r["polarity_ratio"] = s.get("polarity_ratio", 1.0)
r["sentiment"] = s.get("sentiment", 0.0)
r["df_ratio"] = round(r["df"] / r["N"], 4)
r["tipe_frasa"] = "multi-kata" if r["is_frasa"] else "1-kata"

sent_path = os.path.join(args.output_dir, "08_Kansei_sentimen.csv")
with open(sent_path, "w", encoding="utf-8", newline="") as fh:
    w = _csv.DictWriter(fh, fieldnames=[
        "kata", "is_frasa", "freq_total", "df", "idf", "tfidf",
        "skor_leksikon", "mean_rating", "rating_polarity", "sentimen"])
    w.writeheader()
    for r in ranking:
        w.writerow({k: r[k] for k in w.fieldnames})

from collections import Counter
dist = Counter(r["sentimen"] for r in ranking)
print(f" distribusi sentimen ({len(ranking)} kata Kansei): "
      f"f(positif={dist['positif']}, negatif={dist['negatif']}, "
      f"f(netral={dist['netral']}))")
print()
print(" Contoh skor sentimen kata Kansei:")
print(f" { 'kata':18s } { 'leksikon':>9s } { 'mean_rate':>10s } "
      f"f({ 'rat_pol':>8s } { 'sentimen':>9s })")
show = ["bagus", "cantik", "awet", "kokoh", "elegan", "anti_bocor",
        "tahan_panas", "penyok", "cacat", "tidak_sesuai", "tidak_bagus",
        "panas", "dingin"]
for w in show:
    if w in sentimen:
        s = sentimen[w]
        print(f" {w:18s} {s['skor_leksikon']:9d} "
              f"f({s['mean_rating']:10.3f} {s['rating_polarity']:8.3f} "
              f"f({s['sentimen']:>9s})")
print()
print(f" -> tabel sentimen disimpan: {sent_path}")
print()

# -- TAHAP 8b : SENTIMEN DOKUMEN/ULASAN (NLTK + leksikon) -----
print("[8b] Sentimen per-ulasan (NLTK word_tokenize + leksikon) ...")
doc_lexicon = build_doc_lexicon()
try:
    import nltk
    try:
        nltk.data.find("tokenizers/punkt_tab")
    except LookupError:
        nltk.download("punkt_tab", quiet=True)
        nltk.download("punkt", quiet=True)
    from nltk.tokenize import word_tokenize
    tokenizer = word_tokenize
    print(f" tokenizer: NLTK word_tokenize | leksikon: {len(doc_lexicon)}
entri")
except Exception as e:
    tokenizer = lambda s: s.split()
    print(f" (NLTK tdk tersedia, pakai split: {e})")

doc_sentiments = [
    Score_document_sentiment(t, doc_lexicon, tokenizer)
    for t in df[args.review_col]]
df["doc_sent_Score"] = [s[0] for s in doc_sentiments]
df["doc_sent_label"] = [s[1] for s in doc_sentiments]

ddist = Counter(df["doc_sent_label"])
print(f" distribusi sentimen ulasan ({len(df),}): "
      f"f(positif={ddist['positif']}, negatif={ddist['negatif']}, "
      f"f(netral={ddist['netral']},)")
# validasi silang dgn rating
import numpy as _np
pos_rate = df.loc[df["doc_sent_label"] == "positif", "rating_int"].mean()
neg_rate = df.loc[df["doc_sent_label"] == "negatif", "rating_int"].mean()
print(f" rata rating: ulasan positif={pos_rate:.2f}, negatif={neg_rate:.2f} "
      f"f(validasi: positif harus > negatif)")
print()

# -- TAHAP 9 : TABEL KANSEI FINAL + KATEGORI -----
print("[9] Tabel Kansei final + pengelompokan kategori ...")
for r in ranking:
    r["kategori"] = categorize_Kansei(r["kata"])
# skor Kansei = bobot kepentingan (tfidf) x prevalensi (log frekuensi)
r["Kansei_Score"] = round(r["tfidf"] * math.log10(1 + r["freq_total"]), 4)

```



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
# urutkan: kategori (emotional/Kansei sejati dulu), lalu frekuensi
cat_order = ["emotional", "organoleptik", "struktur", "fitur",
             "elemen_desain", "insight", "lainnya"]
ranking.sort(key=lambda r: (cat_order.index(r["kategori"]),
                             -r["freq_total"]))

final_path = os.path.join(args.output_dir, "09_Kansei_final.csv")
with open(final_path, "w", encoding="utf-8", newline="") as fh:
    w = _csv.DictWriter(fh, fieldnames=[
        "kata", "kategori", "is_frasa", "freq_total", "df", "idf", "tfidf",
        "Kansei_Score", "skor_leksikon", "mean_rating", "rating_polarity",
        "sentimen"])
    w.writeheader()
    for r in ranking:
        w.writerow({k: r[k] for k in w.fieldnames})

# tabel format ringkas (kolom sesuai contoh: word, tfidf, df, df_ratio, idf,
# tipe_frasa, sentiment, freq_pos, freq_neg, polarity_ratio, freq_total)
ringkas_path = os.path.join(args.output_dir, "Kansei_final.csv")
with open(ringkas_path, "w", encoding="utf-8", newline="") as fh:
    w = _csv.DictWriter(fh, fieldnames=[
        "word", "tfidf", "df", "df_ratio", "idf", "tipe_frasa",
        "sentiment", "freq_pos", "freq_neg", "polarity_ratio", "freq_total"])
    w.writeheader()
    for r in ranking:
        w.writerow({
            "word": r["kata"],
            "tfidf": r["tfidf"],
            "df": r["df"],
            "df_ratio": r["df_ratio"],
            "idf": r["idf"],
            "tipe_frasa": r["tipe_frasa"],
            "sentiment": r["sentimen"],
            "freq_pos": r["freq_pos"],
            "freq_neg": r["freq_neg"],
            "polarity_ratio": r["polarity_ratio"],
            "freq_total": r["freq_total"],
        })

# ringkasan per kategori
from collections import Counter, defaultdict
by_cat = defaultdict(list)
for r in ranking:
    by_cat[r["kategori"]].append(r)
print(f" total kata Kansei final : {len(ranking)}")
print()
for cat in cat_order:
    items = by_cat[cat]
    if not items:
        continue
    sd = Counter(r["sentimen"] for r in items)
    print(f" {{cat.upper()}} ({{len(items)}} kata | "
          f"{{sd['positif']}}-{{sd['negatif']}}~{{sd['netral']}})")
    for r in items[:8]:
        tag = "" if r["is_frasa"] else " "
        print(f" (tag) {r['kata']}:18s freq={{r['freq_total']}}>5d "
              f"{{r['sentimen']}}>7s (Kansei_Score={{r['Kansei_Score']}}:3f)")
    print()
print(f" >> Tabel Kansei final disimpan: {final_path}")
print(f" (kolom: kata, kategori, is_frasa, freq_total, df, idf, tfidf, "
      f"Kansei_Score, skor_leksikon, mean_rating, rating_polarity, sentimen)")
print()

# -- TAHAP 9b : PISAH emotional_feeling vs karakteristik_desain -----
print(f"[9b] Pemisahan Kansei sejati (emotional feeling) vs karakteristik
desain ...")
feat_rows = build_Kansei_feature_table(ranking, sentimen)
n_emo = sum(1 for r in feat_rows if r["jenis"] == "emotional_feeling")
n_des = len(feat_rows) - n_emo
print(f" emotional_feeling (Kansei sejati): {n_emo} kata")
print(f" karakteristik_desain : {n_des} kata")

# tabel fitur siap-SOM tahap-1 (vektor: tfidf, sentimen, frekuensi,
rating_pol)
feat_path = os.path.join(args.output_dir, "10_Kansei_som_features.csv")
with open(feat_path, "w", encoding="utf-8", newline="") as fh:
    w = _csv.DictWriter(fh, fieldnames=[
        "kata", "is_frasa", "jenis", "kategori", "tfidf", "skor_sentimen",
        "frekuensi", "rating_polarity", "df", "idf", "mean_rating",
        "sentimen"])
    w.writeheader()
    w.writerows(feat_rows)
print(f" >> SOM tahap-1 (vektor fitur kata): {feat_path}")

print()

# -- TAHAP 9c : MATRIKS DOC-TERM + METADATA (SOM tahap-2) -----
print(f"[9c] Matriks doc-term + metadata (responden dgn >1 kata Kansei)
...")
Kansei_vocab = {r["kata"] for r in ranking}
matrix_rows, meta_rows, vocab = build_doc_term_matrix(
    df["tokens"].tolist(), df["rating_int"].tolist(),
    doc_sentiments, Kansei_vocab, min_Kansei=2)
print(f" responden lolos (>1 kata Kansei): {len(matrix_rows):,} "
      f"{{dari {len(df):,} ulasan}")
print(f" dimensi matriks: {len(matrix_rows):,} x {len(vocab):,}")

# simpan matriks doc-term (baris=responden, kolom=kata Kansei)
dtm_path = os.path.join(args.output_dir, "11_doc_term_matrix.csv")
with open(dtm_path, "w", encoding="utf-8", newline="") as fh:
    w = _csv.writer(fh)
    w.writerow(["doc_id"] + vocab)
    for meta, vec in zip(meta_rows, matrix_rows):
        w.writerow([meta["doc_id"]] + vec)

# simpan metadata responden
meta_path = os.path.join(args.output_dir, "11_doc_metadata.csv")
with open(meta_path, "w", encoding="utf-8", newline="") as fh:
    w = _csv.DictWriter(fh, fieldnames=[
        "doc_id", "rating", "skor_sentimen_doc", "sentimen_doc",
        "jumlah_kata_Kansei", "kata_Kansei_unik"])
    w.writeheader()
    w.writerows(meta_rows)
print(f" >> SOM tahap-2 (matriks doc-term): {dtm_path}")
print(f" >> SOM tahap-2 (metadata responden): {meta_path}")
print()
print(f" Ringkasan output siap-SOM:")
print(f" - SOM tahap-1 (dimensi atribut tumbler): klaster
{len(feat_rows):,} "
      f"{{kata Kansei}} pakai vektor [tfidf, sentimen, frekuensi, rating_polarity]")
print(f" - SOM tahap-2 (segmen user): klaster {len(matrix_rows):,} "
      f"{{responden}} pakai matriks doc-term {len(vocab):,} kata Kansei +
metadata")

if __name__ == "__main__":
    main()

import argparse
import os
import re
import sys
import time
import importlib.util

import numpy as np
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer

# -----
# Import fungsi tokenisasi dari Kansei_pipeline.py
# -----
def load_pipeline(path):
    spec = importlib.util.spec_from_file_location("kp", path)
    kp = importlib.util.module_from_spec(spec)
    spec.loader.exec_module(kp)
    return kp

# -----
# Parser rating "bintang N"
# -----
def parse_bintang_rating(raw):
    """bintang 5' / '5' / ' ' -> 5 / 5 / None."""
    if raw is None:
        return None
    s = str(raw).strip().lower()
    if s in ("", "nan", "none", "null"):
        return None
    m = re.search(r"bintang\s*(\d)", s)
    if m:
        n = int(m.group(1))
        return n if 1 <= n <= 5 else None
    try:
        n = int(float(s))
        return n if 1 <= n <= 5 else None
    except ValueError:
        return None
```



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
def rating_to_label(rating):
    """rating -> label sentiment: >=4 positif(1), <=2 negatif(0), 3/None ambigu(1)."""
    if rating is None:
        return -1
    if rating >= 4:
        return 1
    if rating <= 2:
        return 0
    return -1

# -----
# Replicate tokenisasi pipeline sampai token Kansei
# -----
def replicate_tokens(reviews, kp, norm_dict, stem_cache_path):
    """clean -> normalisasi -> phrase match (TANPA sinonim) -> stopword -> stem."""
    # matikan pemetaan sinonim unigram agar kuat/gemas tidak tergabung
    kp.UNIGRAM_SYNONYM = {}

    print(" [a] cleaning ...")
    cleaned = [kp.clean_text(t) for t in reviews]

    print(" [b] normalisasi + phrase matching (tanpa sinonim ...)")
    phrased = [kp.match_affective_phrases(kp.normalized_term(c.split()), norm_dict)]
    for c in cleaned:
        print(" [c] stopwords removal ...")
        sw = [kp.remove_stopwords(t) for t in phrased]

    print(" [d] stemming (Sastrawi + cache) ...")
    uniq = set()
    for toks in sw:
        uniq.update(toks)
    term_dict = kp.build_term_dict(uniq, stem_cache_path) # pakai/lengkapi cache
    stemmed = [kp.get_stemmed_term(t, term_dict) for t in sw]
    return stemmed

def main():
    p = argparse.ArgumentParser(description="Matriks dokumen-term untuk SOM")
    p.add_argument("--input", required=True, help="dataset ulasan mentah (CSV)")
    p.add_argument("--kansei-file", default="Kansei_final_2.csv", help="file kata Kansei final (kolom 'word')")
    p.add_argument("--output-dir", default="outputs")
    p.add_argument("--pipeline", default="Kansei_pipeline.py", help="path Kansei_pipeline.py (sumber fungsi tokenisasi)")
    p.add_argument("--norm-dict", default="normalisasi_dict.csv")
    p.add_argument("--stem-cache", default="outputs/stem_cache.json")
    p.add_argument("--encoding", default="latin-1")
    p.add_argument("--sep", default=";")
    p.add_argument("--kansei-sep", default=";")
    p.add_argument("--review-col", default="Review")
    p.add_argument("--rating-col", default="Rating")
    p.add_argument("--min-kansei-per-doc", type=int, default=1)
    args = p.parse_args()

    os.makedirs(args.output_dir, exist_ok=True)
    t0 = time.time()
    kp = load_pipeline(args.pipeline)

    # -- 1. load ulasan -----
    print(f"\n[1/7] Load ulasan: {args.input} (encoding={args.encoding})")
    df = pd.read_csv(args.input, sep=args.sep, encoding=args.encoding, on_bad_lines="skip")
    df.columns = [c.strip() for c in df.columns]
    df = df.dropna(subset=[args.review_col]).reset_index(drop=True)
    df = df[df[args.review_col].astype(str).str.strip() != ""]
    df.reset_index(drop=True)
    print(f" {len(df):,} ulasan siap")
    reviews = df[args.review_col].astype(str).tolist()
    ratings_raw = (df[args.rating_col].tolist()
                  if args.rating_col in df.columns else [""] * len(df))

    # -- 2. load vocabulary Kansei (34 kata) -----
    print(f"\n[2/7] Load vocabulary Kansei: {args.Kansei_file}")
    kdf = pd.read_csv(args.Kansei_file, sep=args.Kansei_sep)
    if "word" not in kdf.columns:
        print(f"ERROR: kolom 'word' tak ada. Kolom: {kdf.columns.tolist()}")
        sys.exit(1)
```

```
vocab = kdf["word"].astype(str).tolist()
print(f" {len(vocab):,} kata Kansei: {vocab[:6]} ...")

# -- 3. replicate tokenisasi -----
print(f"\n[3/7] Replicate tokenisasi pipeline ...")
norm_dict = kp.load_normalized_word_dict(args.norm_dict)
token_lists = replicate_tokens(reviews, kp, norm_dict, args.stem_cache)
n_nonempty = sum(1 for t in token_lists if any(w in vocab for w in t))
print(f" {n_nonempty:,}/{len(token_lists):,} ulasan punya >=1 kata Kansei")

f"({100*n_nonempty/len(token_lists):.1f}%)")

# -- 4. TF-IDF vocabulary terkunci -----
print(f"\n[4/7] TF-IDF (vocab dikunci ke {len(vocab)} kata) ...")
docs = [" ".join(toks) for toks in token_lists]
vec = TfidfVectorizer(
    vocabulary=vocab,
    ngram_range=(1, 1),
    sublinear_tf=True, # TF log-scaled
    token_pattern=r"(?u)\b\w+\b", # akomodasi token underScore
    norm="l2", # normalisasi per dokumen (utk SOM)
    lowercase=False,
)
X = vec.fit_transform(docs)
print(f" matriks: {X.shape[0]:,} ulasan x {X.shape[1]:,} kata | "
      f"nnz={X.nnz:,} | sparsity={1 - X.nnz/(X.shape[0]*X.shape[1]):.2f}")

# -- 5. statistik per dokumen -----
print(f"\n[5/7] Statistik per dokumen ...")
jumlah_Kansei = np.asarray((X > 0).sum(axis=1)).ravel()
sum_tfidf = np.asarray(X.sum(axis=1)).ravel()
rating_int = [parse_bintang_rating(r) for r in ratings_raw]
label_sent = [rating_to_label(r) for r in rating_int]

# -- 6. filter opsional -----
if args.min_Kansei_per_doc > 1:
    print(f"\n[6/7] Filter ulasan >= {args.min_Kansei_per_doc} kata Kansei ...")
    keep = jumlah_Kansei >= args.min_Kansei_per_doc
    nb = X.shape[0]
    X = X[keep]
    jumlah_Kansei = jumlah_Kansei[keep]
    sum_tfidf = sum_tfidf[keep]
    rating_int = [rating_int[i] for i in range(len(rating_int)) if keep[i]]
    label_sent = [label_sent[i] for i in range(len(label_sent)) if keep[i]]
    reviews_f = [reviews[i] for i in range(len(reviews)) if keep[i]]
    original_idx = np.where(keep)[0]
    print(f" {nb:,} -> {X.shape[0]:,} ulasan ({100*X.shape[0]/nb:.1f}%)")
else:
    print(f"\n[6/7] Tanpa filter min_Kansei_per_doc")
    reviews_f = reviews
    original_idx = np.arange(X.shape[0])

# -- 7. export -----
print(f"\n[7/7] Export matriks + metadata ...")
dt = pd.DataFrame(X.toarray(), columns=vocab)
dt.insert(0, "doc_idx", original_idx)
out_mat = os.path.join(args.output_dir, "10_doc_term_matrix.csv")
dt.to_csv(out_mat, index=False)
print(f" [OK] {out_mat} ({os.path.getsize(out_mat)/1024/1024:.1f} MB)")

meta = pd.DataFrame({
    "doc_idx": original_idx,
    "rating": rating_int,
    "label_sentiment": label_sent,
    "jumlah_kata_Kansei": jumlah_Kansei,
    "total_tfidf": np.round(sum_tfidf, 6),
    "review_preview": [str(r)[:120].replace("\n", " ") for r in reviews_f],
})
out_meta = os.path.join(args.output_dir, "10_metadata.csv")
meta.to_csv(out_meta, index=False)
print(f" [OK] {out_meta}")

# ringkasan
print(f"\n[done] {time.time()-t0:.1f} detik")
print(f"=" * 60)
print(f" ulasan final (matriks) : {X.shape[0]:,}")
print(f" vocabulary Kansei : {len(vocab):,} kata")
print(f" sparsity : {1 - X.nnz/(X.shape[0]*X.shape[1]):.2f}")
print(f"=" * 60)
print(f"distribusi jumlah kata Kansei per ulasan:")
print(meta["jumlah_kata_Kansei"].describe().to_string())
print(f"ndistribusi label sentiment (1=pos, 0=neg, -1=ambigu/netral):")
```



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
print(meta["label_sentiment"].value_counts().sort_index().to_string())
# df per kata (berapa ulasan memuat tiap kata)
df_per_word = (X > 0).sum(axis=0)
df_per_word = np.asarray(df_per_word).ravel()
print("\ndocument frequency tiap kata Kansei (df):")
order = np.argsort(-df_per_word)
```

```
for i in order:
    print(f" {vocab[i]:20s} {int(df_per_word[i]:5d}")

if __name__ == "__main__":
    main()
```





© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Lampiran 3. Hasil kata *Kansei* dari TF-IDF

kata	is_frasa	freq_total	df	idf
tidak presisi	1	5	5	3.537995
tidak kokoh	1	6	6	3.458814
tidak rapat	1	6	6	3.458814
tidak baik	1	7	7	3.391867
mudah digenggam	1	7	7	3.391867
apik	0	8	8	3.333875
licin	0	8	8	3.333875
mudah dibersihkan	1	15	14	3.090837
ergonomis	0	9	9	3.282723
tidak pas	1	9	9	3.282723
kuat	0	10	10	3.236965
stylish	0	11	11	3.195573
tidak awet	1	12	12	3.157784
tidak tebal	1	14	14	3.090837
tidak aman	1	14	14	3.090837
gemas	0	15	15	3.060874
bahan premium	1	15	15	3.060874
lambat	0	23	22	2.894543
fungsional	0	17	17	3.006516
anti licin	1	23	23	2.875237
oke	0	25	25	2.839025
mudah dibawa	1	29	29	2.774567
halus	0	30	30	2.759844
tidak bagus	1	35	35	2.692897
modern	0	50	48	2.555724
praktis	0	38	38	2.657182
rapat	0	49	48	2.555724
tidak rapi	1	46	46	2.574207
solid	0	54	53	2.512689
premium	0	78	74	2.367734
nyaman	0	59	59	2.466113
berat	0	84	80	2.333875
tidak sesuai	1	77	75	2.361904
mahal	0	80	78	2.344871
estetik	0	80	79	2.339338
hangat	0	125	115	2.176267
lengkap	0	87	86	2.302467
tipis	0	108	104	2.219932
anti bau	1	87	87	2.297446



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

anti embun	1	112	111	2.191642
anti tumpah	1	112	111	2.191642
simpel	0	118	118	2.165083
bersih	0	150	146	2.072612
mewah	0	160	154	2.049445
bahan kokoh	1	129	129	2.126376
ringan	0	169	162	2.02745
mulus	0	190	180	1.981693
kecil	0	193	187	1.965124
cacat	0	196	192	1.953664
harga terjangkau	1	177	177	1.988992
unik	0	220	212	1.910629
murah	0	235	225	1.884783
anti bocor	1	187	187	1.965124
lecet	0	266	256	1.828725
bahan tebal	1	258	256	1.828725
penyok	0	504	462	1.572323
tahan dingin	1	374	374	1.664094
tahan panas	1	398	393	1.642573
cocok	0	424	419	1.614751
cakep	0	518	494	1.543238
elegan	0	487	472	1.563023
tebal	0	467	457	1.577049
kualitas bagus	1	426	426	1.607556
tahan lama	1	482	474	1.561187
sesuai deskripsi	1	537	532	1.511054
rapi	0	626	619	1.445275
kokoh	0	906	827	1.31946
puas	0	793	771	1.349911
panas	0	1233	1100	1.195573
cantik	0	987	934	1.266618
lucu	0	1395	1206	1.155618
keren	0	1066	998	1.237835
besar	0	1098	1047	1.217019
mantap	0	1417	1331	1.112787
dingin	0	1639	1505	1.059429
aman	0	1677	1597	1.03366
pas	0	1604	1549	1.046914
awet	0	1750	1698	1.007028
cepat	0	2117	2038	0.927761
sesuai	0	3959	3528	0.689437

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Lampiran 4. Hasil *Document-term Matrix*

doc_idx	mudah_digenggam	mudah_dibersihkan	ergonomis	kuat	stylish	gemus	fungsional	anti_licin	mudah_dibawa	modern	praktis	rapat	premium	nyaman	estetik	simpel	bersih	megah	ringan	multis	unik	anti_bocor	tahan_dingin	tahan_panas	elegan	tebal	kuualitas_bagus	tahan_lama	kokoh	cantik	lucu	keren	besar	awet
0	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.43 94	0.00 00	0.00 00	0.63 91	0.63 12	0.00 00	
1	0.00 00	0.66 71	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.58 79	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.45 76	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00
4	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.75 74	0.00 00	0.65 30	
5	0.00 00	0.00 00	0.00 00	0.49 70	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.56 81	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.46 95	0.45 81	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	
7	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.78 18	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.62 35	0.00 00	0.00 00	0.00 00	0.00 00	0.61 73	0.00 00	0.00 00	0.00 00	0.00 00	
11	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.45 39	0.00 00	0.00 00	0.67 28	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.35 26	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.40 24	0.23 47	0.00 00	0.00 00		
12	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.45 39	0.00 00	0.00 00	0.67 28	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.35 26	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.40 24	0.23 47	0.00 00	0.00 00		
13	0.00 00	0.00 00	0.00 00	0.44 98	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.52 84	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.72 01	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.75 74	0.00 00	0.65 30	
16	0.00 00	0.00 00	0.00 00	0.55 64	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.65 37	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.63 61	0.00 00	0.00 00	0.00 00	0.51 01	0.00 00	0.00 00	0.00 00	0.00 00		
18	0.00 00	0.00 00	0.00 00	0.61 85	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.77 90	0.00 00	0.00 00	0.00 00	0.00 00	0.61 44	0.00 00	0.62 70	0.00 00	0.00 00	0.00 00	0.00 00	0.48 94	0.00 00	0.00 00		
20	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.68 69	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.55 28	0.00 00	0.00 00	0.00 00	0.00 00	0.47 17	0.00 00	0.00 00	0.00 00		
28	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.83 08	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.68 69	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.55 65	0.00 00	0.00 00	0.00 00	0.00 00	0.47 17	0.00 00	0.00 00	0.00 00		
30	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.67 88	0.00 00	0.00 00	0.00 00	0.55 61	0.00 00	0.47 95		
36	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.66 59	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.57 48	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.47 55	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00		
40	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.62 18	0.00 00	0.55 66	0.55 09	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	
43	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.74 23	0.00 00	0.00 00	0.00 00	0.67 01	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00		
45	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.67 01	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.71 15	0.70 27	0.00 00	0.00 00	
46	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.74 23	0.00 00	0.00 00	0.00 00	0.00 00	0.67 01	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	
47	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.62 18	0.00 00	0.55 66	0.55 09	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	
48	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	
49	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00	0.00 00																		



1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

144

© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang mengumumkan atau memperbanyak karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

[illegible]

- Hak Cipta :**
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
 2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

© Hak Cipta milik Politeknik Negeri Jakarta

[illegible]



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta:

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Lampiran 5. Hasil *Metadata* Min 2 Kata *Kansei*

doc_idx	rating	label_sen timent	jumlah _kata_ Kansei	total_tfidf	review_preview
0	5	1	3	1.709753	Respons, persiapan barang oleh dan durasi pengiriman semua cepat bahkan lebih cepat daripada deskripsi. Produk keren ban
1	5	1	3	1.712565	Material: Lapisan keramiknya rapi, tidak berbau, aman untuk panas dan dingin Characteristics: Simple dan praktis. Mudah
4	5	1	2	1.410356	keren nih mug bener-bener terbuat dari keramik dalamnya, warnanya suka, desainnya apik dengan detail yang menarik, belum
5	5	1	4	1.992676	Desain: Mulus hitam cool Ukuran: 13oz 380ml Bahan: Keramik, stanlis tapi tutup nya tembus panas kuat kokoh ga tembus pan
7	5	1	2	1.405318	Mug kopi Civago nya bagus bgt. Bahan: Tebal dan keramik bersih Desain: Elegan dan trendy, cocok dipakai kapanpun
11	5	1	2	1.404023	Material: Kokoh dan halus Design: Mewah dengan warna krem coklat yang kalem Size: Pas untuk bekal di meja kantor Charact
12	5	1	5	2.116377	Material: Food grade, premium Characteristics: Keren Design: Mewah Size: Cukup, tidak terlalu besar/kecil Kualitas tumbl
13	5	1	5	2.116377	Material: Food grade, premium Characteristics: Keren Design: Mewah Size: Cukup, tidak terlalu besar/kecil Kualitas tumbl
16	5	1	3	1.698237	Bahan: Barangnya sangat Bagus & bahannusangat kuat ad kemarin di dalam nya Karakteristik: Mencerminkan sebuah tempat min
18	5	1	2	1.410356	Barang bagus berkualitas dengan harga yang terjangkau. Seller sangat responsif dan informatif. Barang tidak lama diprose
19	5	1	3	1.723024	Bahan: Kuat dan kokoh Desain: Mewah Ukuran: Sesuai Mungkin karna masih baru jadi ada bau plastik apa keramik? Entahlah,
20	5	1	3	1.715412	Bahan: Rapih n kokoh Karakteristik: Sangat elegan Desain: Sangat bagus Ukuran: Pas buat keseharian Semoga awet cat nya g
28	5	1	3	1.722761	kualitas material baik, kuat, tahan dingin gampang dicuci, tidak berbau habis dicuci, warna keren
30	5	1	2	1.406017	Material: material yg mewah dengan harga terjangkau Design: simple tapi elegan Size: pas buat hot/cold black coffee or e
36	5	1	3	1.711476	Mug nya cantik, mewah, elegan. Semoga tahan panasnya. Next mau pesan tumbler lagi. Terimakasih reseller.
40	5	1	3	1.711476	Mug nya cantik, mewah, elegan. Semoga tahan panasnya. Next mau pesan tumbler lagi. Terimakasih reseller.
43	5	1	2	1.387364	Bahan: Bagus Banget Karakteristik: Futuristik dan Elegan Desain: Modern Ukuran: Pas Banget di tangan
45	5	1	3	1.714451	Bahan: Kualitas barang ok, dalamnya kramik Desain: Desainnya keren dengan desain tali kualitas ok Ukuran: Pas buat minum
46	5	1	3	1.71628	Terbaik dan Premium Bahan: Mulus Karakteristik: Elegan Desain: Menarik Ukuran: Standar yg tidak terlalu kecil
47	5	1	3	1.729366	Bahan: Bagus Karakteristik: Tahan dingin dan agak kurang tahan panas nya, nggak sesuai yang di ikinkan Desain: Agak uni
48	5	1	2	1.412366	Bahan: Tebal,lapis keramik Karakteristik: Sangat kuat Desain: Pas dan unik Ukuran: Cukup simpel
49	5	1	2	1.414186	Material: Bagus tapi untuk tutupnya agak kurang rapih detailnya Design: Keren Size: Cukup, tidak terlalu kecil dan tidak
50	5	1	2	1.412366	Bahan: Tebal,lapis keramik Karakteristik: Sangat kuat Desain: Pas dan unik Ukuran: Cukup simpel
51	5	1	3	1.729366	Bahan: Bagus Karakteristik: Tahan dingin dan agak kurang tahan panas nya, nggak sesuai yang di ikinkan Desain: Agak uni
56	5	1	2	1.393728	kualitas ok, pengiriman cepat, blm di coba panas sama dinginnya, semoga awet
63	5	1	2	1.366222	Bahan: Tebel Karakteristik: Digenggam nyaman banget Desain: Keren banget kekinian Ukuran: Paslah
66	5	1	2	1.410169	Keren banget modelnya, bahan material kokoh dan cukup berat pas di genggam.. puas bangetttt makasihhhh yaaaa
69	5	1	2	1.391331	Bahan: Keramik Desain: Unik Ukuran: Sedang Paketnya sudah tiba, barang sesuai pesanan, pengiriman cepat, packing rapih &
73	5	1	2	1.413643	Bahan: Staines Keramik Karakteristik: Kuat Desain: Cool Ukuran: 380ml Tahan lama tanpa merubah rasa.
77	5	1	3	1.724441	Bahan: Bagus, materialnya kuat Karakteristik: Mewah Desain: Mantap dan elegan, terlihat minimalis Ukuran: Pas dan cocok
82	5	1	3	1.642873	Bahan: Kuat Karakteristik: Stylish Desain: Keren Ukuran: Pas Sesuai
86	5	1	2	1.409803	Ukuran: 385 ml Desain: Sangat cantik, elegant, dan manis.
88	5	1	2	1.393544	Sangat bagus Desain: Keren dan elegan Ukuran: Pas Karakteristik: Sangat keren



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

93	5	1	2	1.409803	Bahan: Bagus Karakteristik: Cantik Desain: Elegan Ukuran: Sesuai
101	5	1	2	1.409803	Material: Bagus Design: Cantik Warna: Pink elegan gak norak. Sedang di uji coba beneran gak seperti iklannya. Tapi overa
102	5	1	3	1.728344	keren.. bahannya bagus, belum dicoba tapi... semoga awet... alhamdulillah dapet setengah harga...promo gede2 an.. thx to
107	5	1	2	1.410975	sangst bagus sekali, tidak besar dan tidak kecil. awet batu es
126	5	1	2	1.402339	Bahan: Tebal dilapisi keramik Karakteristik: Elegan Desain: Sempel dan menarik Ukuran: Standart
129	5	1	2	1.408664	Bahan: Stainless di lapisi keramik tahan panas Desain: Mewah estetic Ukuran: Kecil pas di bawa kemana-mana
131	5	1	3	1.593866	Bahan: Ada lapisan keramik Karakteristik: Anti tumpah dan lumayan tahan dingin Desain: Elegan dan aesthetic Ukuran: Pas u
142	5	1	2	1.407982	Ukurannya besar dan panasnya tahan lama Hanya bagian kepalanya kalau dimasukkan kopi panas tutupnya ikutan panas juga
143	5	1	2	1.409333	Warna gradasinya cantik.. Cukup berfungsi membuat air dingin awet, begitu juga panas.. Hanya saja bila sedang tdk diguna
146	5	1	2	1.396245	Kualitas materialnya cukup baik, bikin panas & natangnya rata. Ukuran sesuai spek dan Wajib punya. biar awet jangan samp
166	5	1	2	1.395394	Baguss bangetttttt, dusnya itu tebal gk gampang penyok, semoga awet juga tumblernya. Pesanan jg udah sesuai. Thanks min
169	5	1	2	1.394422	Mug kopi dgn warna Pink yg cantik dan benar tidak bocor / rembes, trimakasih
175	5	1	2	1.413484	Bahan: Sesuai yg di butuhkan Karakteristik: Sangat baik Desain: Bagus banget Ukuran: Sempel dan estetik
176	5	1	2	1.404775	Bahan: Sangat bagus Karakteristik: Bagus ,kuat Desain: Keren dan estetik Ukuran: Sesuai dengan keperluan
177	5	1	3	1.710427	Sesuai espektasi.. Kualitas bagus tahan dingin udah 10 jam masih awet dinginnya, cuma di tutupnya berembun
181	4	1	2	1.39055	barangnya bagus, tahan dingin dan panas. Semoga awet.
182	4	1	2	1.406078	Bahan: Sangat bagus di harga terjangkau, keramik tahan panas dan dingin. Desain: Keren, dan modis
...					
16579	5	1	2	1.391331	Bahan: Bagus Karakteristik: Unik Desain: Lucu Ukuran: Pas
16580	5	1	2	1.396245	Tumblernya bagus. Air panas dan air dingin awet tahan lama.
16600	5	1	2	1.414082	bagus kualitas baik suka modelnya tahan dingin/panas cukup lama
16603	5	1	2	1.404775	model keren, bahan baik, ukuran pas gak kegedean, cocok utk ngantor, kuat panas dan dingin
16610	2	0	2	1.391872	kualiatas produk menurun tidak seperti pembelian pertama itu bahan nya tebal. untuk yg sekarang bahan lebih tipis dan mu
16652	5	1	2	1.410144	Ringan tp kuat
16688	5	1	2	1.410356	pembelian yg kesekian.... suka sekaoi dgn warna dan model2nya.... ?????? awet dinginnya... keren
16689	5	1	2	1.397291	Sudah Buy 2 X>>> Bagus + Kuat + nyaman ada Pegangan >>> cukup buat minum....
16697	5	1	2	1.410356	keren banget bahan luarnya matte dalemnya udah stainless semoga awet yaa
16700	5	1	2	1.410356	keren banget bahan luarnya matte dalemnya udah stainless semoga awet yaa
16701	5	1	2	1.395394	barang sampai dengan baik, bahannya matte, bagus, tebal, semoga awet
16714	5	1	2	1.40881	bagus tebal tidak bocor, tutup untuk minumnya mudah terbuka, tidak recommended taroh dalam tas anak sekolah
16731	5	1	2	1.41379	Pengiriman super cepat, pengemasan bagus. Botolnya bagus kuat tebal, botol minumnya mendarat sempurna tidak ada cocol b
16736	5	1	2	1.41401	kualitas baik, warna sesuai gambar, ukuran sesuai, bahan kuat
16748	5	1	2	1.393728	pesanan sesuai, barang bagus, tidak ada cacat, kualitas baik dengan harga promo...??semoga awet yah.. thanks seller????
16760	5	1	2	1.414186	Design: bagus poll Size: lumayan besar Characteristics: warna nya bagus keren
16791	5	1	2	1.380512	Bahan: Bahan tebal dan cukup solid Desain: Sempel tapi tetap best Ukuran: Sangat pass untuk dipakai sehari hari tidak ke
16798	5	1	2	1.406145	bagus, lumayan tebal, bagus lucu sekali
16799	5	1	3	1.664248	Bahan: Plastik tebal Karakteristik: Tebal Desain: Bagus unik keren Ukuran: 750ml



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta:

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

16801	5	1	3	1.724455	Bagusss bgt tebal dan kokoh jg tidak mudah bocor jg pokoknya sukaaaa bgt dengan produk ecentio ini toppp bgt deh
16806	5	1	2	1.414042	Bahan: Bagus Desain: Cantik Ukuran: Lumayan besar Bagussssss bangeeett
16812	5	1	2	1.411386	bagus sesua deskripsi dan dpt harga promo moga kuat dan ga bocor..klo begitu ya pasti order lg krn adknya jg pgn
16814	5	1	2	1.413661	Bahan: Plastik Karakteristik: Bening, kuat Desain: Elegan Ukuran: 500ml
16834	5	1	4	1.99304	Ketebalannya sedang. Kualitas bagus. Tutupnya kokoh tidak bocor
16835	5	1	4	1.981031	terbaik warnanya cantik , elegan mewah,tahan air panas dan Kokoh.
16841	5	1	2	1.414153	kualitas barang bagus, ukuran tidak terlalu besar, masih ok buat dibawa sekolah, ngantor bentuk lucu, recommend buat tok
16853	5	1	2	1.395394	bagus barangnya..lmyan tebal juga.. semoga awet Incar trus usahanya ya??
16873	5	1	3	1.698009	Meskipun plastik botol tdk terlalu tebal yg pnting tidak bocor smoga awet
16904	5	1	2	1.372354	Barang sudah diterima dengan kondisi baik, sudah di tes ternyata anti bocor karena ada seal di tutupnya, semoga awet.
16905	5	1	3	1.72652	kualitas bagus tahan pecah bahan ringan tidak bocor
16925	5	1	2	1.413045	bahan bagus, material kokoh walaupun dari plastik kuat. Selalu langganan beli disini karena kualitas cukup puas, dan har
16937	5	1	3	1.725824	keren... warnanya cantik, bahannya tebal packing aman... terima kasih seller...
16948	5	1	2	1.393728	dapat harga murmer, kualitas oke. semoga awet yaaa. terimakasih seller.
16950	5	1	3	1.726614	kualitasnya bagus, tebal juga, warnanya cantik ??
16952	5	1	3	1.723373	bahannya tebal lucu ga terlalu gede ga terlalu kecil jdi flexible banget
17026	5	1	2	1.387683	botolnya lumayan gede tapi ringan ??
17174	5	1	2	1.410975	botol minumnya besar ya semoga awet tidak mudah pecah.
17181	5	1	3	1.710427	so far ga kecewa sih sama kualitasnya, oke bgtt. tahan dingin dan semoga awet yah.
17184	5	1	3	1.72528	Material: Kokoh Design: Clasic tapi keren Asli gercep bgt seller nya. Packing rapih. Suka sama design & materialnya. War
17186	5	1	2	1.409387	Kali pertama belanja di sini. Penjualnya komunikatif & ramah banget. Ekspedisi kilat. Meski tanpa bungkus gelembung, syu
17194	5	1	3	1.720234	bagus aku suka. desain nya simpel dan elegan. tahan dingin nya juga oke.
17195	5	1	2	1.406078	Barang bagus tahan panas nya sampai 2 hr keren????
17196	5	1	2	1.393728	kualitas bagus, tahan buat nyimpen dingin sampe 20 jam lah masih awet.
17201	5	1	2	1.368444	bagus pol ringan lgi , pengirimannya cpt banget lgii semoga awet ya
17204	5	1	2	1.382035	Dari bodynya kokoh banget, next coba es apakah awet ? Semoga saya es batunya awet, seawet cintaku padanya eaaaa
17207	5	1	2	1.407404	Barang sudah sampai warnanya cantik, semoga tahan panas / dingin.
17208	5	1	2	1.410356	Gak kaleng kaleng kualitasnya sama yg merk sebelah, mudah2an awet sampe lama, keren dan bagus, terima kasih
17229	5	1	2	1.39082	Gede banget ternyata kegedean juga buat sekolah haha. Tapi bagus diisi air panas gak tembus sama sekali keluar dan gak b
17235	5	1	3	1.713974	Turnlarnya bagus sekali, warnanya juga menarik. Kelihatan mewah dan premium..kokoh juga. Mkasih tiktok makasih juga tyes
17250	5	1	2	1.391773	Ngga nyesel beli disini, tahan panas lebih dari 12 jam, semoga awet



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta:

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Lampiran 6. Source Code Self-organizing Map Stage 1

```

from __future__ import annotations

import argparse
import json
import os
import csv
from pathlib import Path

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.patches as mpatches
from matplotlib.patches import RegularPolygon
from minisom import MiniSom
from sklearn.cluster import KMeans
from sklearn.metrics import Silhouette_Score # HANYA SILHOUETTE, TIDAK
ADA DBI
from sklearn.preprocessing import normalize

# KONSTANTA
SEED = 42
GRID_X = 6 # 6x6 = 36 neuron (untuk ~34 kata Kansei)
GRID_Y = 6
TOPOLOGY = 'hexagonal'
NEIGHBORHOOD = 'gaussian'
K_RANGE = list(range(2, 10)) # k = 2..9
N_INIT_KMEANS = 50
MIN_CLUSTER_SIZE = 2

# Tuning grid
TUNE_SIGMA_LIST = [1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2.0]
TUNE_LR_LIST = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0]
TUNE_ITER_SHORT = 10000
TUNE_TOP_N = 5
ITERATIONS_FULL = 18000 # >= 500 x 36 neuron = 18,000

# Default (mode --no-tune)
SIGMA_INIT = 1.5
LEARNING_RATE = 0.5
ITERATIONS = 18000

# Nama file input (default sesuai data terbaru)
DTM_FILE = '10_doc_term_matrix_min2.csv'
KANSEI_FILE = 'Kansei_final_2.csv'

INPUT_DIR = Path(os.environ.get('SOM_INPUT_DIR', ''))
OUTPUT_DIR = Path(os.environ.get('SOM_OUTPUT_DIR', ''))

# WARNA DIMENSI
DIM_COLORS = [
    '#E8A598', '#7FCAC3', '#F5B97B', '#B7D88F',
    '#FCE883', '#C8B4E8', '#FFB6DD', '#9FC4E8',
    '#D4E0B9', '#F2A65A',
]

# UTILITAS HEX
def hex_center(i: int, j: int, r: float = 1.0):
    dx = np.sqrt(3) * r
    dy = 1.5 * r
    x = j * dx + (i % 2) * (dx / 2)
    y = i * dy
    return x, y

def set_hex_axes(ax, grid_x, grid_y, r=1.0, padding=1.2):
    all_x = [hex_center(i, j, r)[0] for i in range(grid_x) for j in range(grid_y)]
    all_y = [hex_center(i, j, r)[1] for i in range(grid_x) for j in range(grid_y)]
    ax.set_xlim(min(all_x) - padding * r, max(all_x) + padding * r)
    ax.set_ylim(min(all_y) - padding * r, max(all_y) + padding * r)
    ax.set_aspect('equal')
    ax.axis('off')

# LOAD DATA
def detect_sep(path: Path) -> str:
    with open(path, encoding='utf-8', errors='replace') as fh:
        header = fh.readline()
    return ';' if header.count(';') >= header.count(',') else ','

def read_csv_robust(path: Path, sep: str | None = None) -> pd.DataFrame:
    if sep is None:
        sep = detect_sep(path)
    df = pd.read_csv(path, sep=sep, quoting=csv.QUOTE_NONE, dtype=str)

cols = list(df.columns)
if cols[0].startswith(''): cols[0] = cols[0].lstrip('')
if cols[-1].endswith(''): cols[-1] = cols[-1].rstrip('')
df.columns = cols
first, last = df.columns[0], df.columns[-1]
df[first] = df[first].astype(str).str.strip('')
df[last] = df[last].astype(str).str.strip('')
for c in df.columns:
    coerced = pd.to_numeric(df[c], errors='coerce')
    if coerced.notna().all():
        if (coerced == coerced.astype(np.int64)).all():
            df[c] = coerced.astype(np.int64)
        else:
            df[c] = coerced.astype(float)
    return df

def load_data(input_dir: Path, dtm_file: str, Kansei_file: str):
    dtm = read_csv_robust(input_dir / dtm_file)
    Kansei = read_csv_robust(input_dir / Kansei_file)
    for c in dtm.columns[1:]:
        dtm[c] = pd.to_numeric(dtm[c], errors='coerce').fillna(0.0)
    feature_cols = [c for c in dtm.columns if c != 'doc_idx']
    n_Kansei = len(Kansei) if 'word' in Kansei.columns else len(feature_cols)
    Xnum = dtm[feature_cols].values.astype(float)
    n_ber_Kansei = int((Xnum > 0).any(axis=1).sum())
    n_unik = dtm['doc_idx'].nunique() if 'doc_idx' in dtm.columns else
    dtm.shape[0]
    print(f"[INFO] DTM: {dtm.shape[0]} baris x {len(feature_cols)} kata "
          f"f'(vocab Kansei: {n_Kansei})")
    print(f"[INFO] doc_idx unik = {n_unik}")
    + f"(ADA {dtm.shape[0] - n_unik} duplikat)" if n_unik != dtm.shape[0]
    else ""
    + f"f' | ulasan ber-Kansei (>=1 kata) = {n_ber_Kansei}"
    if 'word' in Kansei.columns and set(feature_cols) !=
    set(Kansei['word']).tolist():
        print(f"[WARN] Kata di matriks dan di file Kansei tidak identik "
              "(profil dimensi tetap dibuat dgn join 'word').")
    return dtm, Kansei, feature_cols

def build_word_vectors(dtm: pd.DataFrame, feature_cols: list) -> np.ndarray:
    """Ko-okurens: C = X^T X; word_vec[i] = C[i, :] dinormalisasi L2."""
    X = dtm[feature_cols].values
    co_oc = X.T @ X
    wv = normalize(co_oc, norm='l2', axis=1)
    print(f"[INFO] Ko-okurens matrix: {wv.shape}")
    return wv

# SOM
def make_som(sigma: float, lr: float, input_len: int) -> MiniSom:
    return MiniSom(
        x=GRID_X, y=GRID_Y, input_len=input_len,
        sigma=sigma, learning_rate=lr,
        neighborhood_function=NEIGHBORHOOD,
        topology=TOPOLOGY,
        activation_distance='euclidean',
        random_seed=SEED,
    )

def train_som(X: np.ndarray, sigma: float = SIGMA_INIT,
              lr: float = LEARNING_RATE, iterations: int = ITERATIONS):
    som = make_som(sigma, lr, X.shape[1])
    som.pca_weights_init(X)
    print(f"[INFO] Training SOM {GRID_X}x{GRID_Y} hex "
          f"f'(sigma={sigma}, lr={lr}, {iterations}, {iter}...)")
    som.train_random(X, iterations, verbose=False)
    qe = som.quantization_error(X)
    te = som.topographic_error(X)
    print(f"[INFO] QE = {qe:.4f} TE = {te:.4f}")
    return som, qe, te

# K-MEANS VALIDATION (TANPA DBI)
def kmeans_validation(prototypes: np.ndarray, k_range: list) ->
pd.DataFrame:
    results = []
    for k in k_range:
        km = KMeans(n_clusters=k, n_init=N_INIT_KMEANS,
                    random_state=SEED)
        labels = km.fit_predict(prototypes)
        sil = Silhouette_Score(prototypes, labels)
        results.append({'k': k, 'Silhouette': sil})
    print(f"[INFO] k={k:2d} Silhouette={sil:.4f}")

```



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta:

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

return pd.DataFrame(results)

def _word_cluster_sizes(som, X, prototypes, k):
    km = KMeans(n_clusters=k, n_init=N_INIT_KMEANS,
                random_state=SEED)
    labels = km.fit_predict(prototypes)
    bmus = np.array([som.winner(x) for x in X])
    bflat = bmus[:, 0] * GRID_Y + bmus[:, 1]
    wlabels = labels[bflat]
    return np.bincount(wlabels, minlength=k)

def choose_optimal_k(val_df, som, X, prototypes,
                    min_cluster_size=MIN_CLUSTER_SIZE,
                    max_outlier_clusters=1,
                    max_cluster_proportion=0.5):
    k_sil_raw = int(val_df.loc[val_df['Silhouette'].idxmax(), 'k'])
    total = None
    valid_ks = []
    for _, row in val_df.iterrows():
        k = int(row['k'])
        sizes = _word_cluster_sizes(som, X, prototypes, k)
        if total is None:
            total = int(sizes.sum())
        n_small = int(np.sum(sizes < min_cluster_size))
        max_prop = sizes.max() / total
        reasons = []
        if n_small > max_outlier_clusters:
            reasons.append(f'{n_small} outlier > {max_outlier_clusters}')
        if max_prop > max_cluster_proportion:
            reasons.append(f'max cluster {max_prop:.0%} > {max_cluster_proportion:.0%}')
        status = "OK" if not reasons else f'skip ({', '.join(reasons)})'
        print(f'[INFO] k={k} sizes={sizes.tolist()} max={max_prop:.0%} {status}')
        if not reasons:
            valid_ks.append(k)
    if not valid_ks:
        chosen = k_sil_raw
        note = f'Fallback Silhouette raw k={chosen}'
    else:
        valid_df = val_df[val_df['k'].isin(valid_ks)]
        chosen = int(valid_df.loc[valid_df['Silhouette'].idxmax(), 'k'])
        sizes_ch = _word_cluster_sizes(som, X, prototypes, chosen)
        note = f'k valid: {valid_ks}. Dipilih k={chosen} (Silhouette max di valid set). '
        f'Sizes: {sizes_ch.tolist()}. k_Silhouette_raw={k_sil_raw}.'
    print(f'[INFO] {note}')
    return chosen, k_sil_raw, note

# TUNING 2-FASE (3 metrik, TANPA DBI)
def tune_phase1(X: np.ndarray) -> pd.DataFrame:
    total = len(TUNE_SIGMA_LIST) * len(TUNE_LR_LIST)
    print(f'[TUNING] Fase 1: {total} kombinasi x {TUNE_ITER_SHORT:} iter...')
    results = []
    for sig in TUNE_SIGMA_LIST:
        for lr in TUNE_LR_LIST:
            som = _make_som(sig, lr, X.shape[1])
            som.pca_weights_init(X)
            som.train_random(X, TUNE_ITER_SHORT, verbose=False)
            qe = som.quantization_error(X)
            te = som.topographic_error(X)
            results.append({'sigma': sig, 'lr': lr, 'QE_fase1': qe, 'TE_fase1': te})
    return pd.DataFrame(results)

def select_top_candidates(df_phase1: pd.DataFrame, top_n: int =
TUNE_TOP_N) -> pd.DataFrame:
    df = df_phase1.copy()
    df['rank_QE'] = df['QE_fase1'].rank(method='min')
    df['rank_TE'] = df['TE_fase1'].rank(method='min')
    df['rank_sum'] = df['rank_QE'] + df['rank_TE']
    top = df.sort_values('rank_sum').head(top_n).reset_index(drop=True)
    print(f'[TUNING] Top-{top_n} kandidat (rank QE+TE):')
    for _, r in top.iterrows():
        print(f'    sigma={r['sigma']} lr={r['lr']} '
              f'QE={r['QE_fase1']:.4f} TE={r['TE_fase1']:.4f}')
    return top

def tune_phase2(X: np.ndarray, candidates: pd.DataFrame) -> pd.DataFrame:
    print(f'[TUNING] Fase 2: {len(candidates)} kandidat x {len(K_RANGE)} k '
          f'x {ITERATIONS_FULL:} iter penuh...')
    rows = []
    for _, cand in candidates.iterrows():
        sig, lr = cand['sigma'], cand['lr']
        som = _make_som(sig, lr, X.shape[1])
        som.pca_weights_init(X)
        som.train_random(X, ITERATIONS_FULL, verbose=False)
        qe = som.quantization_error(X)
        te = som.topographic_error(X)
        prototypes = som.get_weights().reshape(-1, X.shape[1])
        for k in K_RANGE:
            km = KMeans(n_clusters=k, n_init=N_INIT_KMEANS,
                        random_state=SEED)
            labels = km.fit_predict(prototypes)
            sil = Silhouette_Score(prototypes, labels)
            rows.append({'sigma': sig, 'lr': lr, 'k': k,
                        'QE': qe, 'TE': te, 'Silhouette': sil})
    return pd.DataFrame(rows)

def _composite_rank(df: pd.DataFrame) -> pd.DataFrame:
    df = df.copy()
    df['rank_QE'] = df['QE'].rank(method='min')
    df['rank_TE'] = df['TE'].rank(method='min')
    df['rank_Sil'] = df['Silhouette'].rank(method='min', ascending=False)
    df['composite_rank'] = df['rank_QE'] + df['rank_TE'] + df['rank_Sil']
    return df.sort_values('composite_rank').reset_index(drop=True)

def select_best_params(df_phase2: pd.DataFrame) -> dict:
    df_ranked = _composite_rank(df_phase2)
    best = df_ranked.iloc[0]
    return {
        'sigma': float(best['sigma']),
        'lr': float(best['lr']),
        'k': int(best['k']),
        'QE': float(best['QE']),
        'TE': float(best['TE']),
        'Silhouette': float(best['Silhouette']),
        'composite_rank': float(best['composite_rank']),
    }

# CLUSTERING FINAL
def final_clustering(prototypes, k):
    km = KMeans(n_clusters=k, n_init=N_INIT_KMEANS, random_state=SEED)
    return km, km.fit_predict(prototypes)

def assign_words(som, neuron_clusters):
    bmus = np.array([som.winner(x) for x in X])
    bflat = bmus[:, 0] * GRID_Y + bmus[:, 1]
    return bmus, neuron_clusters[bflat]

# PROFIL DIMENSI
DIMENSION_NAME_HINTS = {
    'durabilitas': ['awet', 'tahan lama', 'kuat', 'kokoh', 'tebal'],
    'estetika': ['cantik', 'lucu', 'unik', 'gemas', 'keren', 'stylish'],
    'premium': ['elegant', 'mewah', 'premium', 'bahan premium',
                'estetik', 'modern'],
    'kualitas_inti': ['kualitas bagus', 'mulus', 'bersih', 'cacat'],
    'fungsi_thermal': ['tahan panas', 'tahan dingin'],
    'fungsi_seal': ['anti bocor', 'rapat'],
    'kapasitas': ['besar', 'ringan', 'tebal'],
    'ergonomi': ['ergonomis', 'nyaman', 'mudah dibawa',
                'mudah digenggam',
                'anti licin', 'ringan'],
    'kemudahan_pakai': ['praktis', 'fungsional', 'mudah dibersihkan', 'simpler'],
    'nilai_harga': ['harga terjangkau'],
}

def suggest_dimension_name(words):
    word_set = set(words)
    Scores = [(len(word_set & set(hints)), name)
               for name, hints in DIMENSION_NAME_HINTS.items()
               if len(word_set & set(hints)) > 0]
    return sorted(Scores, reverse=True)[0][1] if Scores else 'lainnya'

def build_dimension_profile(Kansei, word_list, word_clusters, stat_cols):
    df_assign = pd.DataFrame({'word': word_list, 'dimensi_id': word_clusters +
1})
    df_full = df_assign.merge(Kansei, on='word', how='left')
    rows = []
    for dim_id in sorted(df_full['dimensi_id'].unique()):
        sub = df_full[df_full['dimensi_id'] == dim_id]
        words = sub['word'].tolist()
        row = {'dimensi_id': int(dim_id),
              'dimensi_nama_suggest': suggest_dimension_name(words),
              'n_kata': len(sub),
              'kata_anggota': ', '.join(words)}
        for c in stat_cols:
            row[f'mean_{c}'] = float(sub[c].mean())
        rows.append(row)

```



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta:

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
return pd.DataFrame(rows), df_full

def count_reviews_per_dimension(dtm, feature_cols, word_list,
word_clusters):
    """Sebaran ULASAN ke tiap dimensi (cluster kata Kansei).

    Perbaikan (vs versi lama):
    * Penyebut = jumlah ULASAN BER-KANSEI (memuat >=1 kata Kansei final),
    BUKAN seluruh baris matriks. Baris tanpa satu pun kata Kansei tidak
    dapat diatribusikan ke dimensi manapun, jadi tidak ikut dihitung
    (ini adalah penyebab total 4911 -> seharusnya jumlah ulasan ber-Kansei).
    * Baris ganda (doc_idx duplikat) di-dedup lebih dulu.
    * Penugasan EKSKLUSIF: tiap ulasan diberikan ke SATU dimensi dominan
    (agregat skor TF-IDF kata anggota tertinggi) -> total persentase = 100%
    (memperbaiki bug persentase yang menjumlah > 100% karena tumpang
    tindih).
    * Kolom '*_singgung' = cakupan non-eksklusif (boleh tumpang tindih),
    hanya sebagai info tambahan, tidak dijumlahkan menjadi 100%.

    Catatan: bila terjadi seri skor, dimensi dengan indeks terkecil dipilih
    (np.argmax bersifat deterministik).
    """
    work = dtm.copy()
    n_raw = len(work)
    if 'doc_idx' in work.columns:
        work = work.drop_duplicates(subset='doc_idx')
    n_dup = n_raw - len(work)

    X = work[feature_cols].values.astype(float)
    n_rows = X.shape[0]

    col_idx = {w: i for i, w in enumerate(feature_cols)}
    dim_ids = sorted(set((word_clusters + 1).tolist()))

    # Kumpulan kata anggota + indeks kolomnya untuk tiap dimensi
    dim_members, dim_colidx = {}, {}
    for d in dim_ids:
        members = [word_list[i] for i in range(len(word_list))
                    if word_clusters[i] + 1 == d]
        dim_members[d] = members
        dim_colidx[d] = [col_idx[w] for w in members if w in col_idx]

    # Skor agregat (jumlah TF-IDF kata anggota) tiap ulasan x dimensi
    dim_scores = np.zeros((n_rows, len(dim_ids)))
    for di, d in enumerate(dim_ids):
        if dim_colidx[d]:
            dim_scores[:, di] = X[:, dim_colidx[d]].sum(axis=1)

    has_Kansei = dim_scores.sum(axis=1) > 0 # ulasan dgn >=1 kata
    Kansei
    n_review_Kansei = int(has_Kansei.sum()) # <- penyebut (mis.
    2504)
    dominant = np.argmax(dim_scores, axis=1) # dimensi dominan per
    ulasan
    base = n_review_Kansei if n_review_Kansei else 1

    rows = []
    for di, d in enumerate(dim_ids):
        excl_mask = has_Kansei & (dominant == di) # eksklusif: dominan = d
        n_excl = int(excl_mask.sum())
        if dim_colidx[d]: # singgung: memuat >=1 kata d
            n_touch = int((X[:, dim_colidx[d]] > 0).any(axis=1).sum())
        else:
            n_touch = 0
        rows.append({
            'dimensi_id': int(d),
            'n_kata': len(dim_members[d]),
            'kata_anggota': '; '.join(dim_members[d]),
            'n_ulasan': n_excl,
            'pct_ulasan': round(100.0 * n_excl / base, 2),
            'n_ulasan_singgung': n_touch,
            'pct_singgung': round(100.0 * n_touch / base, 2),
        })
    df = pd.DataFrame(rows)

    print(f"[INFO] Ulasan: {n_raw} baris matriks"
    + f"({n} -> {len(work)}) unik (buang {n_dup} duplikat doc_idx)" if n_dup else
    "")
    + f"; ber-Kansei (>=1 kata) = {n_review_Kansei}"
    print(f"[INFO] Penugasan eksklusif (dimensi dominan): "
    f"sum n_ulasan = {int(df['n_ulasan'].sum())} (= {n_review_Kansei}), "
    f"sum pct = {df['pct_ulasan'].sum():.1f}%"
    return df, n_review_Kansei
```

```
# VISUALISASI: SILHOUETTE CURVE
def plot_Silhouette_curve(val_df, k_sil, k_chosen, out_path):
    fig, ax = plt.subplots(figsize=(9, 5))
    ax.plot(val_df['k'], val_df['Silhouette'], 'o-',
            color='#A23B72', lw=2.5, markersize=9, label='Silhouette')
    ax.axvline(x=k_sil, color='red', linestyle='--', alpha=0.6,
              label=f'Silhouette max raw: k={k_sil}')
    ax.axvline(x=k_chosen, color='green', linestyle=':', lw=2,
              label=f'Dipilih: k={k_chosen}')
    best_sil = val_df.loc[val_df['Silhouette'].idxmax()]
    ax.annotate(f"max={best_sil['Silhouette']:.4f}",
               xy=(best_sil['k'], best_sil['Silhouette']),
               xytext=(8, 10), textcoords='offset points',
               color='red', fontsize=10)
    ax.set_xlabel('Jumlah Cluster (k)', fontsize=12)
    ax.set_ylabel('Silhouette Coefficient', fontsize=12)
    ax.set_title('Kurva Silhouette - Stage 1: Dimensi Kansei Optimal',
                fontsize=11)
    ax.set_xticks(val_df['k'])
    ax.grid(True, alpha=0.3)
    ax.legend(fontsize=10)
    plt.tight_layout()
    plt.savefig(out_path, dpi=120, bbox_inches='tight')
    plt.close()
    print(f"[OUT] {out_path.name}")

# VISUALISASI: HEX MAP STAGE 1
def plot_hex_map_stage1(som, word_list, X, word_clusters, k, out_path):
    r = 1.0
    bmus = np.array([som.winner(x) for x in X])
    neuron_dim = {}
    neuron_words = {}
    for idx, (bi, bj) in enumerate(bmus):
        neuron_dim[(bi, bj)] = int(word_clusters[idx])
        neuron_words.setdefault((bi, bj), []).append(word_list[idx])
    fig, ax = plt.subplots(figsize=(15, 13))
    for i in range(GRID_X):
        for j in range(GRID_Y):
            x, y = hex_center(i, j, r)
            dim = neuron_dim.get((i, j), -1)
            fc = DIM_COLORS[dim % len(DIM_COLORS)] if dim >= 0 else '#F0F0F0'
            ec = '#888888' if dim < 0 else 'white'
            hp = RegularPolygon(
                (x, y), numVertices=6, radius=r * 0.96,
                orientation=0,
                facecolor=fc, edgecolor=ec, linewidth=1.8, alpha=0.92,
            )
            ax.add_patch(hp)
            if (i, j) in neuron_words:
                wds = neuron_words[(i, j)]
                lines = [w.replace('_', '\n') for w in wds[:3]]
                if len(wds) > 3:
                    lines.append(f'+{len(wds)-3}')
                text = '\n'.join(lines)
                fsz = 6.5 if len(wds) <= 2 else 5.5
                ax.text(x, y, text, ha='center', va='center',
                       fontsize=fsz, fontweight='bold', color='#1a1a1a',
                       multialignment='center', linespacing=1.2)
            _set_hex_axes(ax, GRID_X, GRID_Y, r, padding=1.0)
            handles = [mpatches.Patch(facecolor=DIM_COLORS[i % len(DIM_COLORS)],
                                     label=f'Dimensi {i+1}') for i in range(k)]
            ax.legend(handles=handles, loc='center left',
                    bbox_to_anchor=(1.01, 0.5), fontsize=10,
                    title='Dimensi Kansei', title_fontsize=11)
            ax.set_title(f'Stage 1: Peta SOM Hexagonal {GRID_X}x{GRID_Y}x{n}'
                        f'({k} Dimensi Desain Kansei - Ko-okurens ({len(word_list)} kata)',
                        fontsize=12, pad=12)
            plt.tight_layout()
            plt.savefig(out_path, dpi=130, bbox_inches='tight')
            plt.close()
            print(f"[OUT] {out_path.name}")

# VISUALISASI: HEX U-MATRIX STAGE 1
def plot_hex_umatrix_stage1(som, out_path):
    umat = som.distance_map()
    r = 1.0
    cmap = plt.cm.Blues
    vmin, vmax = umat.min(), umat.max()
    fig, ax = plt.subplots(figsize=(10, 9))
    for i in range(GRID_X):
        for j in range(GRID_Y):
            x, y = hex_center(i, j, r)
            val = umat[i, j]
            norm = (val - vmin) / (vmax - vmin + 1e-9)
```



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta:

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

fc = cmap(norm)
hp = RegularPolygon(
    (x, y), numVertices=6, radius=r * 0.96,
    orientation=0,
    facecolor=fc, edgecolor='white', linewidth=1.2,
)
ax.add_patch(hp)
_set_hex_axes(ax, GRID_X, GRID_Y, r, padding=0.8)
sm = plt.cm.ScalarMappable(cmap=cmap,
                           norm=plt.Normalize(vmin=vmin, vmax=vmax))
sm.set_array([])
plt.colorbar(sm, ax=ax, shrink=0.75, pad=0.02,
             label='Jarak ke neuron tetangga (naik = batas cluster)')
ax.set_title(f'U-Matrix Stage 1 - SOM {GRID_X}x{GRID_Y} Hexagonal',
             fontsize=11, pad=10)
plt.tight_layout()
plt.savefig(out_path, dpi=130, bbox_inches='tight')
plt.close()
print(f'[OUT] {out_path.name}')

# VISUALISASI: TUNING HEATMAP
def plot_tuning_Heatmap(df_phase2: pd.DataFrame, out_path: Path):
    df_ranked = _composite_rank(df_phase2)
    k_best = int(df_ranked.iloc[0]['k'])
    df_k = df_phase2[df_phase2['k'] == k_best].copy()
    sigmas = sorted(df_k['sigma'].unique())
    lrs = sorted(df_k['lr'].unique())
    metrics = [
        ('QE', 'Blues_r', True, 'QE turun lebih baik'),
        ('TE', 'Oranges_r', True, 'TE turun lebih baik'),
        ('Silhouette', 'Greens', False, 'Silhouette naik lebih baik'),
    ]
    fig, axes = plt.subplots(1, 3, figsize=(16, 5))
    for ax, (metric, cmap, low_good, subtitle) in zip(axes, metrics):
        mat = np.zeros((len(sigmas), len(lrs)))
        for si, sig in enumerate(sigmas):
            for li, lr in enumerate(lrs):
                row = df_k[(df_k['sigma'] == sig) & (df_k['lr'] == lr)]
                if not row.empty:
                    mat[si, li] = row[metric].values[0]
        im = ax.imshow(mat, cmap=cmap, aspect='auto')
        ax.set_xticks(range(len(lrs))); ax.set_xticklabels(lrs, rotation=45)
        ax.set_yticks(range(len(sigmas))); ax.set_yticklabels(sigmas)
        ax.set_xlabel('Learning Rate'); ax.set_ylabel('Sigma')
        ax.set_title(f'{metric}\n({subtitle})', fontsize=11)
        fig.colorbar(im, ax=ax, fraction=0.046, pad=0.04)
        best_row = df_k.loc[df_k[metric].idxmin()] if low_good else
        df_k[metric].idxmax()
        bi = sigmas.index(best_row['sigma'])
        li = lrs.index(best_row['lr'])
        ax.add_patch(plt.Rectangle((li - 0.5, bi - 0.5), 1, 1,
                                   fill=False, edgecolor='gold', lw=3))
        fig.suptitle(f'Hyperparameter Tuning Stage 1 - k={k_best} terpilih',
                    fontsize=12, y=1.03)
    plt.tight_layout()
    plt.savefig(out_path, dpi=120, bbox_inches='tight')
    plt.close()
    print(f'[OUT] {out_path.name}')

# MAIN
def main():
    parser = argparse.ArgumentParser(
        description='Stage 1: Dimensi Kansei - SOM Hexagonal, tanpa DBI')
    parser.add_argument('--input-dir', type=Path, default=INPUT_DIR)
    parser.add_argument('--output-dir', type=Path, default=OUTPUT_DIR)
    parser.add_argument('--dtm-file', type=str, default=DTM_FILE,
                        help='nama file matriks dokumen-term')
    parser.add_argument('--kansei-file', type=str, default=KANSEI_FILE,
                        help='nama file kata Kansei final (kolom word)')
    parser.add_argument('-k', type=int, default=None,
                        help='Override jumlah cluster')
    parser.add_argument('--no-tune', action='store_true',
                        help='Gunakan SIGMA_INIT + LEARNING_RATE langsung')
    args = parser.parse_args()

    args.output_dir.mkdir(parents=True, exist_ok=True)
    np.random.seed(SEED)

    print('-' * 78)
    print(f' STAGE 1 - SOM HEXAGONAL {GRID_X}x{GRID_Y} - DIMENSI KANSEI (TANPA DBI)')
    print('-' * 78)
    print(f'Grid: {GRID_X}x{GRID_Y} | TOPOLOGY | Seed: {SEED}')
    if args.no_tune:

```

```

        print(f'[Fixed]
        sigma={SIGMA_INIT} lr={LEARNING_RATE} iter={ITERATIONS:}')
    else:
        print(f'[Tuning] sigma x lr grid | Fase1 iter={TUNE_ITER_SHORT:} | '
              f'Top-N={TUNE_TOP_N} | Fase2 iter={ITERATIONS_FULL:}')
    print(f'k range: {K_RANGE}')
    print('-' * 78)

    dtm, Kansei, feature_cols = load_data(args.input_dir, args.dtm_file,
                                           args.kansei_file)
    X = build_word_vectors(dtm, feature_cols)

    k_from_tuning = None
    if args.no_tune:
        som, qe, te = train_som(X, SIGMA_INIT, LEARNING_RATE, ITERATIONS)
        best_sigma, best_lr, iter_used = SIGMA_INIT, LEARNING_RATE,
        ITERATIONS
    else:
        print('\nTUNING FASE 1 - Screening sigma x lr')
        df_phase1 = tune_phase1(X)
        top_cands = select_top_candidates(df_phase1, TUNE_TOP_N)
        print('\nTUNING FASE 2 - Top-N iter penuh + Silhouette per k')
        df_phase2 = tune_phase2(X, top_cands)
        df_tuning = _composite_rank(df_phase2)
        best = select_best_params(df_phase2)
        best_sigma, best_lr, k_from_tuning = best['sigma'], best['lr'], best['k']
        iter_used = ITERATIONS_FULL
        print(f'[TUNING] Terbaik: sigma={best_sigma} lr={best_lr}
        k={k_from_tuning}')
        fQE=(best["QE"]:.4f) TE=(best["TE"]:.4f) Sil=(best["Silhouette"]:.4f)
        df_tuning.to_csv(args.output_dir / 'stage1_tuning_results.csv',
                        index=False, sep=';')
        print(f'[OUT] stage1_tuning_results.csv ({len(df_tuning)} kombinasi)')
        plot_tuning_Heatmap(df_phase2, args.output_dir /
                            'stage1_tuning_Heatmap.png')
        if args.k is None:
            args.k = k_from_tuning
        print(f'\nSOM FINAL - sigma={best_sigma} lr={best_lr}
        iter={ITERATIONS_FULL:}')
        som, qe, te = train_som(X, best_sigma, best_lr, ITERATIONS_FULL)

        prototypes = som.get_weights().reshape(-1, X.shape[1])
        print('-' * 78)
        print('K-Means validation (Silhouette saja, TANPA DBI):')
        val_df = kmeans_validation(prototypes, K_RANGE)
        print(val_df.to_string(index=False))

        print('-' * 78)
        k_chosen_auto, k_sil, note = choose_optimal_k(val_df, som, X, prototypes)
        k_chosen = args.k if args.k is not None else k_chosen_auto
        if not args.no_tune and k_from_tuning is not None:
            print(f'[INFO] k tuning={k_from_tuning} k
            filter={k_chosen_auto} final={k_chosen}')

        print('-' * 78)
        km, neuron_clusters = final_clustering(prototypes, k_chosen)
        bmsu, word_clusters = assign_words(som, X, neuron_clusters)

        stat_cols = ['tfidf', 'df', 'df_ratio', 'sentiment', 'polarity_ratio', 'freq_total']
        profile_df, df_full = build_dimension_profile(Kansei, feature_cols,
                                                    word_clusters, stat_cols)

        # JUMLAH ULASAN per cluster/dimensi
        review_count_df, n_review_total = count_reviews_per_dimension(
            dtm, feature_cols, feature_cols, word_clusters)
        profile_df = profile_df.merge(
            review_count_df[['dimensi_id', 'n_ulasan', 'pct_ulasan',
                            'n_ulasan_singgung', 'pct_singgung']],
            on='dimensi_id', how='left')

        print(f'Profil {k_chosen} dimensi Kansei')
        f(total {n_review_total:}, ulasan ber-Kansei: '
        f'penguasaan eksklusif ke dimensi dominan, sum pct = 100%:)'
        for _, row in profile_df.iterrows():
            print(f' Dim {row["dimensi_id"]} ({row["dimensi_nama_suggest"]}, "
            f' {row["n_kata"]} kata): {int(row["n_ulasan"])} ulasan dominan "
            f' {row["pct_ulasan"]}% (singgung {int(row["n_ulasan_singgung"])})) "'
            f'> {row["kata_anggota"]}')

        print('-' * 78)
        df_full.sort_values(['dimensi_id', 'word']).to_csv(
            args.output_dir / 'stage1_dimensi_Kansei.csv', index=False)
        print(f'[OUT] stage1_dimensi_Kansei.csv ({len(df_full)} kata)')

```



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
profile_df.to_csv(args.output_dir / 'stage1_dimensi_profil.csv',
index=False)
print(f"[OUT] stage1_dimensi_profil.csv ({k_chosen} dimensi)")
```

```
review_count_df.to_csv(
args.output_dir / 'stage1_jumlah_ulasan_cluster.csv', index=False)
print(f"[OUT] stage1_jumlah_ulasan_cluster.csv "
f"({ulasan_per_dimensi; total {n_review_total};) ulasan ber-Kansei)")
```

```
val_df.to_csv(args.output_dir / 'stage1_validation_metrics.csv',
index=False)
print(f"[OUT] stage1_validation_metrics.csv")
```

```
plot_Silhouette_curve(val_df, k_sil, k_chosen,
args.output_dir / 'stage1_Silhouette_curve.png')
plot_hex_map_stage1(som, feature_cols, X, word_clusters, k_chosen,
args.output_dir / 'stage1_hex_map.png')
plot_hex_umatrix_stage1(som, args.output_dir /
'stage1_hex_umatrix.png')
```

```
meta = {
'stage': 1,
'method': 'SOM Hexagonal + K-Means on word co-occurrence vectors',
'topology': TOPOLOGY,
'tuning_mode': 'fixed' if args.no_tune else 'composite_rank_QE_TE_Sil',
'evaluation': 'QE + TE + Silhouette (no DBI)',
'n_kata': int(len(feature_cols)),
'grid': [GRID_X, GRID_Y],
'sigma': best_sigma if not args.no_tune else SIGMA_INIT,
'learning_rate': best_lr if not args.no_tune else LEARNING_RATE,
```

```
'iterations': iter_used,
'qe': float(qe), 'te': float(te),
'k_chosen': int(k_chosen),
'k_optimal_Silhouette_raw': int(k_sil),
'note': note,
'n_review_total': int(n_review_total),
'dimension_names': {int(r['dimensi_id']): r['dimensi_nama_suggest']
for _, r in profile_df.iterrows()},
'dimension_n_ulasan': {int(r['dimensi_id']): int(r['n_ulasan'])
for _, r in profile_df.iterrows()},
'word_to_dimension': {str(r['word']): int(r['dimensi_id'])
for _, r in df_full.iterrows()},
}
```

```
with open(args.output_dir / 'stage1_dimensi_meta.json', 'w') as f:
json.dump(meta, f, indent=2, ensure_ascii=False)
print(f"[OUT] stage1_dimensi_meta.json")
```

```
print('=' * 78)
print('STAGE 1 SELESAI.')
sigma_out = best_sigma if not args.no_tune else SIGMA_INIT
lr_out = best_lr if not args.no_tune else LEARNING_RATE
print(f" sigma={sigma_out} lr={lr_out} iter={iter_used}")
print(f" QE={qe:.4f} TE={te:.4f} Silhouette_max={val_df['Silhouette'].max()
:.4f}")
print(f" k={k_chosen} | Output: {args.output_dir.resolve()}")
print('=' * 78)
```

```
if __name__ == '__main__':
main()
```

**POLITEKNIK
NEGERI
JAKARTA**



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Lampiran 7. Source Code Self-organizing Map Stage 2

```
import os
import csv
import math
import argparse
import numpy as np
import pandas as pd
import matplotlib

matplotlib.use("Agg")
import matplotlib.pyplot as plt
from matplotlib.patches import RegularPolygon
from matplotlib.collections import PatchCollection
import matplotlib.cm as cm
from matplotlib.colors import Normalize, ListedColormap

from scipy.spatial.distance import cdist

from minisom import MiniSom
from sklearn.cluster import AgglomerativeClustering
from sklearn.metrics import Silhouette_Score

# -----
# KONFIGURASI
# -----
SEED = 42
MAP_X = 17 # jumlah kolom neuron (lebar)
MAP_Y = 16 # jumlah baris neuron (tinggi) -> 17x16 = 272 neuron
(~5*sqrt(N))
K_MIN, K_MAX = 2, 30 # rentang jumlah segmen yg diuji
N_TOP_PLANES = 12 # jumlah kata utk component planes
N_TOP_KANSEI = 5 # jumlah kata Kansei teratas per segmen (kolom
top_kata_Kansei)
TOPOLOGY = "hexagonal"
DIST = "euclidean"

# Grid hyperparameter SOM (dicari kombinasi dgn QE & TE terbaik)
GRID_SIGMA = [2.0, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9, 3.0]
GRID_LR = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0]
GRID_ITERS = [25000]

np.random.seed(SEED)

#
=====
# 1. PEMBACAAN DATA
#
=====
def baca_doc_term_matrix(path):
    """Baca matriks TF-IDF (delimiter ';', kadang ber-quote). Return (X, doc_idx,
    words)."""
    words, doc_idx, rows = [], [], []
    with open(path, encoding="utf-8-sig") as f:
        words = f.readline().strip().split(";")[1:]
        for line in f:
            line = line.strip().strip('"')
            if not line:
                continue
            p = line.split(",")
            doc_idx.append(int(float(p[0])))
            rows.append([float(v) for v in p[1:]])
    X = np.asarray(rows, dtype=float)
    return X, np.asarray(doc_idx), words

def baca_dimensi(path):
    """
    Baca stage1_dimensi_profil.csv.
    Return:
    dim_order : list id dimensi terurut [1..7]
    dim_nama : {id: nama}
    dim_words : {id: [kata,...]} (sudah dibersihkan -> underScore)
    """
    dim_order, dim_nama, dim_words = [], {}, {}
    with open(path, encoding="utf-8-sig") as f:
        for row in csv.DictReader(f, delimiter=";"):
            did = int(row["dimensi_id"])
            dim_order.append(did)
            dim_nama[did] = row["dimensi_nama_suggest"].strip()
            members = [w.strip().replace(" ", "_") for w in
            row["kata_anggota"].split(",")]

            dim_words[did] = [w for w in members if w]

def baca_Kansei_final(path):
    """Baca statistik kata (opsional, dipakai utk urutan/skala). Return dict word-
    >tfidf."""
    out = {}
    try:
        with open(path, encoding="utf-8-sig") as f:
            for row in csv.DictReader(f, delimiter=";"):
                out[row["word"].strip()] = float(row.get("tfidf", 0) or 0)
    except Exception:
        pass
    return out

def baca_metadata(path, doc_idx):
    """Baca metadata; align ke doc_idx yg dipakai SOM. Return DataFrame
    (boleh kosong)."""
    try:
        md = pd.read_csv(path, sep=";", encoding="utf-8-sig")
        md = md.set_index("doc_idx").reindex(doc_idx).reset_index()
        return md
    except Exception:
        return pd.DataFrame({"doc_idx": doc_idx})

#
=====
# 2. PELATIHAN SOM + GRID SEARCH (QE & TE TERBAIK)
#
=====
def latih_som(X, sigma, lr, iters):
    som = MiniSom(MAP_X, MAP_Y, X.shape[1],
                  sigma=sigma, learning_rate=lr,
                  neighborhood_function="gaussian",
                  topology=TOPOLOGY, activation_distance=DIST,
                  random_seed=SEED)
    som.pca_weights_init(X)
    som.train(X, iters, verbose=False)
    return som

def grid_search_som(X):
    """Cari kombinasi (sigma, lr, iters) dgn skor QE+TE ternormalisasi
    terendah."""
    print("\n[Grid search SOM] mencari QE & TE terbaik ...")
    hasil = []
    for it in GRID_ITERS:
        for sg in GRID_SIGMA:
            for lr in GRID_LR:
                som = latih_som(X, sg, lr, it)
                qe = som.quantization_error(X)
                te = som.topographic_error(X)
                hasil.append(dict(sigma=sg, lr=lr, iters=it, qe=qe, te=te, som=som))
    print("sigma={sg:<4} lr={lr:<4} iters={it:<6} ->
    QE={qe:<4f} TE={te:<4f}")

    qe_arr = np.array([h["qe"] for h in hasil])
    te_arr = np.array([h["te"] for h in hasil])

    def norm(a):
        rng = a.max() - a.min()
        return np.zeros_like(a) if rng == 0 else (a - a.min()) / rng

    skor = norm(qe_arr) + 2.0 * norm(te_arr) # TE diberi bobot lebih besar:
    # peta yg topologinya rapi (TE rendah) menghasilkan segmen yg lebih
    kontigu.
    best = hasil[int(np.argmax(skor))]
    print(f"\n[SOM terbaik] sigma={best['sigma']} lr={best['lr']}
    iters={best['iters']} "
    f" QE={best['qe']:.4f} TE={best['te']:.4f}")
    return best

#
=====
# 3. SEGMENTASI: CLUSTER CODEBOOK BERBASIS TOPOLOGI + PILIH k VIA
SILHOUETTE
#
=====
```



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta:

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
def matriks_adjacency():
    """
    Bangun matriks ketetanggaan hexagonal (272x272).
    Dua neuron bertetangga jika jarak pusat hex-nya ~1 (6 tetangga di interior).
    Dipakai sebagai 'connectivity' agar cluster terbentuk KONTIGU (tidak
    terpecah).
    """
    xs, ys = hex_centers()
    C = np.column_stack([xs.ravel(), ys.ravel()])
    D = cdist(C, C)
    A = (D > 1e-6) & (D < 1.05)
    return A

def pilih_jumlah_segmen(som):
    """
    Agglomerative (Ward) atas codebook SOM DENGAN constraint
    ketetanggaan hexagonal.

    KENAPA bukan KMeans biasa?
    KMeans mengelompokkan vektor bobot tanpa tahu posisinya di grid 2D,
    sehingga
    satu segmen sering PECAH jadi banyak bercak terpisah (peta jadi
    'berantakan').
    Dengan connectivity = ketetanggaan hexagonal, tiap segmen DIJAMIN
    satu wilayah
    menyatu -> hasil peta rapi & kontigu seperti referensi.
    """
    W = som.get_weights().reshape(MAP_X * MAP_Y, -1)
    A = matriks_adjacency()
    ks, sils, models = [], [], {}
    print("\n[Segmentasi] uji jumlah cluster (Silhouette, agglomerative +
    connectivity hex):")
    for k in range(K_MIN, K_MAX + 1):
        ac = AgglomerativeClustering(n_clusters=k, connectivity=A,
        linkage="ward").fit(W)
        s = Silhouette_Score(W, ac.labels_)
        ks.append(k); sils.append(s); models[k] = ac
        print(f"  k={k}<2) Silhouette={s:.4f}")
        best_k = ks[int(np.argmax(sils))]
        print(f"[k terpilih] k={best_k} (Silhouette={max(sils):.4f})")
        return ks, sils, best_k, models[best_k], W

#
# =====
# 4. PROFIL DIMENSI PER SEGMENT -> som_review_dimensi_mapping.csv
#
# =====
def profil_dimensi(X, words, seg_review, dim_order, dim_nama, dim_words,
n_top_Kansei=N_TOP_KANSEI):
    """
    Untuk tiap review: agregasi TF-IDF kata -> 7 dimensi, lalu normalisasi
    sum=1
    (vektor komposisi) -> dipakai untuk kolom numerik D1..D7 (komposisi TF-
    IDF).

    Kolom 'dimensi_dominan' dan 'top_kata_Kansei' dihitung pada BASIS YANG
    SAMA,
    yakni atribusi per-ulasan: tiap ulasan diwakili satu kata terkuat (argmax
    TF-IDF), dan dimensi ulasan = dimensi pemilik kata itu. Persentase
    keduanya
    sama-sama "porsi ulasan", sehingga setara & konsisten:
    - dimensi_dominan = dimensi dgn jumlah ulasan terbanyak
    - top_kata_Kansei = kata terbanyak menjadi terkuat, format kata [Dx] (n,
    %)
    - % sebuah dimensi = jumlah % kata-kata anggotanya di top_kata_Kansei
    Total cacah antar-kata = n, sehingga tidak pernah melebihi jumlah ulasan.
    """
    widx = {w: i for i, w in enumerate(words)}
    dim_idx = {d: [widx[w] for w in dim_words[d] if w in widx] for d in
    dim_order}
    # peta balik: indeks kata -> id dimensi (untuk menandai dimensi tiap kata
    teratas)
    word2dim = {}
    for d in dim_order:
        for wi in dim_idx[d]:
            word2dim[word2dim[wi]] = d

    # peta balik: indeks kata -> id dimensi pemiliknya (untuk penanda [D#] pd
    top_kata_Kansei)
    word2dim = {}
    for d in dim_order:
        for wi in dim_idx[d]:

word2dim[wi] = d

        # matriks dimensi per review (n_review x 7)
        Dmat = np.zeros((X.shape[0], len(dim_order)))
        for j, d in enumerate(dim_order):
            Dmat[:, j] = X[:, dim_idx[d]].sum(axis=1)
        rowsum = Dmat.sum(axis=1, keepdims=True)
        rowsum[rowsum == 0] = 1.0
        Dcomp = Dmat / rowsum # komposisi per review (sum=1)

    seg_ids = sorted(np.unique(seg_review))
    baris = []
    for s in seg_ids:
        mask = seg_review == s
        n = int(mask.sum())
        prof = Dcomp[mask].mean(axis=0) # komposisi TF-IDF -> kolom
        D1..D7

    # -----
    # ATRIBUSI PER-ULASAN (basis tunggal utk dimensi_dominan &
    top_kata):
    # - tiap ulasan diwakili oleh SATU kata terkuat (argmax TF-IDF)
    # - dimensi ulasan = dimensi pemilik kata terkuat tsb
    # Dengan basis ini, persentase dimensi_dominan & top_kata_Kansei
    # sama-sama dihitung sbg "porsi ulasan" sehingga setara/dapat
    # dibandingkan; total cacah antar-kata = n (tidak berlebih).
    # -----
    Xseg = X[mask]
    has_word = Xseg.max(axis=1) > 0 # ulasan yg memuat >=1 kata
    Kansei
    dom_word = Xseg.argmax(axis=1) # indeks kata terkuat tiap
    ulasan
    cnt = np.bincount(dom_word[has_word], minlength=len(words)) #
    cacah ulasan / kata

    # dimensi dominan = dimensi dgn jumlah ulasan terbanyak (basis: %
    ulasan).
    # Catatan: porsi sebuah dimensi = jumlah cacah kata-kata anggotanya,
    # sehingga % dimensi_dominan = jumlah % kata-kata bertanda [D#]
    # yg sama pada top_kata_Kansei -> konsisten satu sama lain.
    dim_vote = np.array([cnt[dim_idx[d]].sum() for d in dim_order])
    ord_d = np.argsort(dim_vote)[::-1]
    d1 = dim_order[ord_d[0]]
    d2 = dim_order[ord_d[1]]
    dom = f"D{d1} {dim_nama[d1]}
    ((dim_vote[ord_d[0]]/max(n,1)*100:.0f)%"
    dom2 = f"D{d2} {dim_nama[d2]}
    ((dim_vote[ord_d[1]]/max(n,1)*100:.0f)%"
    f"D{d2} {dim_nama[d2]}
    ((dim_vote[ord_d[1]]/max(n,1)*100:.0f)%"

    # top kata Kansei (partisi, bertanda dimensi) — basis sama dgn di atas
    topw = np.argsort(cnt)[1:-1][n_top_Kansei]
    top_kata = "; ".join(
        f"{words[w]} [D{word2dim.get(w, '?')}] ({int(cnt[w])},
    cnt[w]/max(n,1)*100:.0f)%"
        for w in topw if cnt[w] > 0
    )
    baris.append((s, n, prof, dom, dom2, top_kata))
    return dim_order, dim_nama, baris

def tulis_mapping_csv(path, dim_order, dim_nama, baris):
    """Tulis dgn header & format mengikuti
    som_review_dimensi_mapping.csv,
    ditambah kolom top_kata_Kansei (jumlah review per kata Kansei tiap
    segmen)."""
    # Nama kolom dimensi diambil LANGSUNG dari dim_nama (sumber:
    stage1_dimensi_profil.csv)
    # agar 100% konsisten dgn kolom 'dimensi_dominan'.
    # (Sebelumnya dipakai dict hardcoded yg keliru: D1=Portabilitas,
    D3=Kualitas,
    # D4=Durabilitas, D5=Fungsional, D6=Ketahanan Termal — tidak sesuai
    dimensi
    # sebenarnya: D1=Minimalis, D3=Fungsional, D4=Kualitas, D5=Durabilitas,
    D6=Ergonomis.)
    cols = ["segmen", "n_review"]
    cols += [f"D{d}_{dim_nama[d]}" for d in dim_order]
    cols += ["top_kata_Kansei", "dimensi_dominan"]

    # urutkan & beri nomor segmen 1..k mengikuti urutan kemunculan
    baris_sorted = sorted(baris, key=lambda b: b[0])
    with open(path, "w", newline="", encoding="utf-8") as f:
        w = csv.writer(f)
        w.writerow(cols)
```



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta:

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

for new_id, (_, n, prof, dom, dom2, top_kata) in enumerate(baris_sorted,
start=1):
    row = ([new_id, n] + [prof[j] for j in range(len(dim_order))]
          + [top_kata, dom])
    w.writerow(row)
    print(f"[CSV] ditulis -> {path}")

def tulis_validation_csv(path_curve, path_metrik, ks, sils, best_k, best,
n_review):
    """
    Tulis 2 file metrik validasi:
    - som_review_validation.csv : kurva k vs Silhouette (+ penanda terpilih)
    - som_review_metrik_final.csv : QE, TE, Silhouette terbaik, & parameter
    SOM
    """
    with open(path_curve, "w", newline="", encoding="utf-8") as f:
        w = csv.writer(f)
        w.writerow(["k", "Silhouette", "terpilih"])
        for k, s in zip(ks, sils):
            w.writerow([k, round(s, 6), "" if k == best_k else ""])
        print(f"[CSV] ditulis -> {path_curve}")

    with open(path_metrik, "w", newline="", encoding="utf-8") as f:
        w = csv.writer(f)
        w.writerow(["metrik", "nilai", "keterangan"])
        w.writerow(["QE", round(best["qe"], 6), "Quantization Error (makin kecil makin baik)"])
        w.writerow(["TE", round(best["te"], 6), "Topographic Error (makin kecil makin baik)"])
        w.writerow(["Silhouette_terbaik", round(max(sils), 6), "skor pada k terpilih"])
        w.writerow(["k_terpilih", best_k, "jumlah segmen terpilih"])
        w.writerow(["sigma", best["sigma"], "lebar neighborhood SOM"])
        w.writerow(["learning_rate", best["lr"], "laju belajar SOM"])
        w.writerow(["iterasi", best["iters"], "jumlah iterasi training"])
        w.writerow(["map_size", f"{MAP_Y}x{MAP_X}", f"{MAP_X*MAP_Y} neuron hexagonal"])
        w.writerow(["n_review", n_review, "jumlah review yg disegmentasi"])
        w.writerow(["seed", SEED, "random seed (reproducible)"])
        print(f"[CSV] ditulis -> {path_metrik}")

#
=====
# 5. UTILITAS PLOT HEXAGONAL
#
=====
def hex_centers():
    """Pusat hexagon (pointy-top) utk grid MAP_X x MAP_Y dgn offset baris ganjil."""
    xs = np.zeros((MAP_X, MAP_Y))
    ys = np.zeros((MAP_X, MAP_Y))
    for i in range(MAP_X):
        for j in range(MAP_Y):
            xs[i, j] = i + 0.5 * (j % 2)
            ys[i, j] = j * (np.sqrt(3) / 2.0)
    return xs, ys

def draw_hexmap(ax, values, cmap, norm=None, labels=None,
label_color="black",
edgecolor="white", lw=1.0, radius=0.58, label_fs=7):
    """Gambar peta hexagonal. values & labels berbentuk (MAP_X, MAP_Y)."""
    xs, ys = hex_centers()
    patches, facecolors = [], []
    cmap_obj = cm.get_cmap(cmap) if isinstance(cmap, str) else cmap
    if norm is None:
        finite = values[np.isfinite(values)]
        norm = Normalize(vmin=float(np.nanmin(finite)),
vmax=float(np.nanmax(finite)))
    for i in range(MAP_X):
        for j in range(MAP_Y):
            hexa = RegularPolygon((xs[i, j], ys[i, j]), numVertices=6, radius=radius,
orientation=0, edgecolor=edgecolor, linewidth=lw)
            patches.append(hexa)
            facecolors.append(cmap_obj(norm(values[i, j])))
            if labels is not None and labels[i, j]:
                ax.text(xs[i, j], ys[i, j], labels[i, j], ha="center", va="center",
fontsize=label_fs, color=label_color)
    pc = PatchCollection(patches, match_original=True)
    pc.set_facecolor(facecolors)
    pc.set_edgecolor(edgecolor)
    pc.set_linewidth(lw)
    ax.add_collection(pc)

```

```

ax.set_xlim(-1, MAP_X + 0.5)
ax.set_ylim(-1, ys.max() + 1)
ax.set_aspect("equal")
ax.axis("off")
return norm

# ----- 5a. Kurva validasi Silhouette -----
def plot_validation(ks, sils, best_k, out):
    fig, ax = plt.subplots(figsize=(11, 6.5))
    ax.plot(ks, sils, "-o", color="#1f77b4", lw=2.2, ms=8, label="Silhouette {↑ lebih baik}")
    ax.fill_between(ks, sils, alpha=0.12, color="#1f77b4")
    for k, s in zip(ks, sils):
        ax.annotate(f"{s:.3f}", (k, s), textcoords="offset points", xytext=(0, 8),
ha="center", fontsize=8)
    smax = max(sils)
    ax.axvline(best_k, color="crimson", ls="--", lw=2, label=f"k terpilih = {best_k}")
    ax.annotate(f"max = {smax:.4f}", (best_k, smax), color="crimson",
xytext=(6, 0), textcoords="offset points", va="center", fontsize=10)
    ax.set_title("Kurva Validasi Silhouette — Stage 2: Segmentasi User\n"
"(Semakin tinggi = cluster semakin kompak & terpisah; tanpa DBI)")
    ax.set_xlabel("Jumlah Cluster (k)")
    ax.set_ylabel("Silhouette Coefficient")
    ax.set_xticks(ks)
    ax.grid(alpha=0.3)
    ax.legend(loc="upper left")
    fig.tight_layout()
    fig.savefig(out, dpi=130)
    plt.close(fig)
    print(f"[PNG] {out}")

# ----- 5b. Peta cluster (segmen dominan per neuron) -----
def plot_clusters(som, X, neuron_seg, k, n_review, out):
    # palet PASTEL (tab10 dilembutkan ke arah putih) + tepi PUTIH -> meniru referensi
    base = cm.get_cmap("tab10")

    def pastel(rgba, alpha=0.55):
        r, g, b, _ = rgba
        return (1 - alpha + alpha * r, 1 - alpha + alpha * g, 1 - alpha + alpha * b, 1.0)

    seg_colors = [pastel(base(i % 10)) for i in range(k)]
    empty_color = (0.94, 0.94, 0.94, 1.0) # neuron kosong -> abu sangat muda

    win_count = np.zeros((MAP_X, MAP_Y))
    for x in X:
        wi, wj = som.winner(x)
        win_count[wi, wj] += 1

    xs, ys = hex_centers()
    fig, ax = plt.subplots(figsize=(16, 12))
    patches, facecolors = [], []
    for i in range(MAP_X):
        for j in range(MAP_Y):
            seg = neuron_seg[i, j]
            if win_count[i, j] == 0:
                fc = empty_color; lbl = ""
            else:
                fc = seg_colors[seg]; lbl = f"S{seg+1}"
            patches.append(RegularPolygon((xs[i, j], ys[i, j]), numVertices=6,
radius=0.58,
orientation=0))
            facecolors.append(fc)
            if lbl:
                ax.text(xs[i, j], ys[i, j], lbl, ha="center", va="center", fontsize=7,
color="0.2")
    pc = PatchCollection(patches, match_original=True)
    pc.set_facecolor(facecolors); pc.set_edgecolor("white");
    pc.set_linewidth(2.0)
    ax.add_collection(pc)
    ax.set_xlim(-1, MAP_X + 0.5); ax.set_ylim(-1, ys.max() + 1)
    ax.set_aspect("equal"); ax.axis("off")
    ax.set_title(f"Stage 2: Peta SOM Hexagonal {MAP_Y}x{MAP_X} — {k} Segmen User\n"
f"{n_review} Review Tumbler (warna = segmen dominan per neuron)".replace(" ", ""))
    handles = [plt.Line2D([0], [0], marker="h", ls="", ms=12,
markerfacecolor=seg_colors[i], markeredgecolor="0.6",
label=f"Segmen {i+1}") for i in range(k)]
    ax.legend(handles=handles, title="Segmen User", loc="center left",
bbox_to_anchor=(1.01, 0.5), frameon=False)

```



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta:

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

fig.tight_layout()
fig.savefig(out, dpi=120)
plt.close(fig)
print(f"[PNG] {out}")

# ----- 5c. Densitas review per neuron -----
def plot_density(som, X, out):
    win_count = np.zeros((MAP_X, MAP_Y))
    for x in X:
        wi, wj = som.winner(x)
        win_count[wi, wj] += 1
    fig, ax = plt.subplots(figsize=(16, 12))
    norm = Normalize(vmin=0, vmax=win_count.max())
    labels = np.empty((MAP_X, MAP_Y), dtype=object)
    for i in range(MAP_X):
        for j in range(MAP_Y):
            labels[i, j] = str(int(win_count[i, j])) if win_count[i, j] > 0 else ""
    draw_hexmap(ax, win_count, "Blues", norm=norm, labels=labels,
                label_color="0.25", edgecolor="white", lw=1.0, label_fs=6)
    ax.set_title("Densitas Review — SOM %dx%d Hexagonal\n"
                "(Warna merah = banyak review menumpuk di neuron ini)" %
                (MAP_X, MAP_Y))
    sm = cm.ScalarMappable(cmap="Blues", norm=norm); sm.set_array([])
    cb = fig.colorbar(sm, ax=ax, fraction=0.025, pad=0.02)
    cb.set_label("Jumlah review per neuron")
    fig.tight_layout()
    fig.savefig(out, dpi=120)
    plt.close(fig)
    print(f"[PNG] {out}")

# ----- 5d. Component planes Top-N kata -----
def plot_components(som, words, X, n_top, out):
    mass = X.sum(axis=0)
    top = np.argsort(mass)[::-1][:n_top]
    W = som.get_weights() # (MAP_X, MAP_Y, n_features)
    ncol = 4
    nrow = int(np.ceil(n_top / ncol))
    fig, axes = plt.subplots(nrow, ncol, figsize=(4.6 * ncol, 3.8 * nrow))
    axes = np.atleast_2d(axes)
    for ax in axes.ravel():
        ax.axis("off")
    for idx, w in enumerate(top):
        ax = axes[idx // ncol, idx % ncol]
        plane = W[:, :, w]
        norm = Normalize(vmin=plane.min(), vmax=plane.max())
        draw_hexmap(ax, plane, "Blues", norm=norm, edgecolor="white",
                    lw=0.4)
        ax.set_title(words[w].replace("_", "\n"), fontsize=10)
        sm = cm.ScalarMappable(cmap="Blues", norm=norm); sm.set_array([])
        fig.colorbar(sm, ax=ax, fraction=0.046, pad=0.04)
    fig.suptitle("Component Planes: Top %d Kata Kansei\n"
                "(Warna terang = kata dominan di area tersebut)" % n_top,
                fontsize=13)
    fig.tight_layout(rect=[0, 0, 1, 0.96])
    fig.savefig(out, dpi=110)
    plt.close(fig)
    print(f"[PNG] {out}")

# ----- 5e. U-Matrix -----
def plot_umatrix(som, out):
    um = som.distance_map() # (MAP_X, MAP_Y) sudah ternormalisasi 0..1
    fig, ax = plt.subplots(figsize=(16, 12))
    norm = Normalize(vmin=um.min(), vmax=um.max())
    draw_hexmap(ax, um, "Blues", norm=norm, edgecolor="white", lw=1.5,
                radius=0.60)
    ax.set_title("U-Matrix Stage 2 — SOM %dx%d Hexagonal\n"
                "(Warna gelap = batas antar segmen user)" % (MAP_X, MAP_Y))
    sm = cm.ScalarMappable(cmap="Blues", norm=norm); sm.set_array([])
    cb = fig.colorbar(sm, ax=ax, fraction=0.025, pad=0.02)
    cb.set_label("Jarak ke neuron tetangga (↑ = batas cluster)")
    fig.tight_layout()
    fig.savefig(out, dpi=120)
    plt.close(fig)
    print(f"[PNG] {out}")

#
=====
# MAIN
#
=====

def main():
    ap = argparse.ArgumentParser(description="SOM segmentasi review
    berbasis dimensi Kansei")
    ap.add_argument("--dtm", default="D:/PNJ/TICK/Semester
    8/Skripsi/Progress Skripsi/Running/SOM/SOM
    Pipeline/output_stage2/10_doc_term_matrix_min2.csv")
    ap.add_argument("--dim", default="D:/PNJ/TICK/Semester
    8/Skripsi/Progress Skripsi/Running/SOM/SOM
    Pipeline/output_stage2/stage1_dimensi_profil.csv")
    ap.add_argument("--kansei", default="D:/PNJ/TICK/Semester
    8/Skripsi/Progress Skripsi/Running/SOM/SOM
    Pipeline/output_stage2/Kansei_final_2.csv")
    ap.add_argument("--meta", default="D:/PNJ/TICK/Semester
    8/Skripsi/Progress Skripsi/Running/SOM/SOM
    Pipeline/output_stage2/10_metadata_min.csv")
    ap.add_argument("--outdir", default="D:/PNJ/TICK/Semester
    8/Skripsi/Progress Skripsi/Running/SOM/SOM
    Pipeline/output_stage2/output_som_final")
    args = ap.parse_args()
    os.makedirs(args.outdir, exist_ok=True)

    print("\n" * 70)
    print("STAGE 2 — SEGMENTASI REVIEW (SOM HEXAGONAL + DIMENSI
    KANSEI)")
    print("\n" * 70)

    # 1) data
    X, doc_idx, words = baca_doc_term_matrix(args.dtm)
    dim_order, dim_nama, dim_words = baca_dimensi(args.dim)
    _ = baca_kansei_final(args.kansei)
    _ = baca_metadata(args.meta, doc_idx)
    print(f"[Data] review = {X.shape[0]:,} | fitur kata = {X.shape[1]} | dimensi =
    {len(dim_order):,}")

    # 2) SOM terbaik (QE & TE)
    best = grid_search_som(X)
    som = best["som"]

    # 3) jumlah segmen via Silhouette (agglomerative + connectivity hex ->
    kontigu)
    ks, sils, best_k, ac, W = pilih_jumlah_segmen(som)
    neuron_seg = ac.labels_.reshape(MAP_X, MAP_Y)

    # 4) label segmen tiap review (via cluster neuron pemenang / BMU)
    seg_review = np.array([ac.labels_[som.winner(x)][0] * MAP_Y +
    som.winner(x)[1] for x in X])

    # 5) profil dimensi + top kata Kansei -> mapping CSV
    do, dn, baris = profil_dimensi(X, words, seg_review, dim_order, dim_nama,
    dim_words)
    tulis_mapping_csv(os.path.join(args.outdir,
    "som_review_dimensi_mapping.csv"), do, dn, baris)

    # 5b) metrik validasi -> CSV
    tulis_validation_csv(
    os.path.join(args.outdir, "som_review_validation.csv"),
    os.path.join(args.outdir, "som_review_metrik_final.csv"),
    ks, sils, best_k, best, X.shape[0])

    # 6) gambar
    plot_validation(ks, sils, best_k, os.path.join(args.outdir,
    "som_review_validation.png"))
    plot_clusters(som, X, neuron_seg, best_k, X.shape[0],
    os.path.join(args.outdir, "som_review_clusters.png"))
    plot_density(som, X, os.path.join(args.outdir, "som_review_density.png"))
    plot_components(som, words, X, N_TOP_PLANES,
    os.path.join(args.outdir, "som_review_komponen_top.png"))
    plot_umatrix(som, os.path.join(args.outdir, "som_review_umatrix.png"))

    # ringkasan metrik
    print("\n" + "=" * 70)
    print("RINGKASAN METRIK FINAL")
    print("=" * 70)
    print(f"QE (Quantization Error) : {best['qe']:.4f}")
    print(f"TE (Topographic Error) : {best['te']:.4f}")
    print(f"SIL (Silhouette terbaik) : {max(sils):.4f} @ k={best_k}")
    print(f"Map : {MAP_Y}x{MAP_X} hexagonal ({MAP_X*MAP_Y} neuron)")
    print(f"Param: sigma={best['sigma']} lr={best['lr']} iters={best['iters']}")
    seed=(SEED)
    print(f"Output -> {os.path.abspath(args.outdir)}")

if __name__ == "__main__":
    main()

```



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Lampiran 8. Identitas *Expert Panelist* dalam Penentuan Konsep Desain

<i>Expert Panelist 1</i>	
Nama	Darsono
Umur	> 40 tahun
Pekerjaan	Karyawan Swasta (<i>Packaging Designer Specialist</i> PT Garudafood)
Penghasilan	>Rp 5.000.000

<i>Expert Panelist 2</i>	
Nama	Sulistyono
Umur	> 40 tahun
Pekerjaan	Dosen Komputer Grafis TGP PNJ
Penghasilan	>Rp 5.000.000

<i>Expert Panelist 3</i>	
Nama	Wanda Kusuma Putri
Umur	30 - 40 tahun
Pekerjaan	Dosen Desain Grafis PNJ
Penghasilan	>Rp 5.000.000

<i>Expert Panelist 4</i>	
Nama	Dina Martin
Umur	> 40 tahun
Pekerjaan	Dosen Universitas Indraprasta PGRI
Penghasilan	>Rp 5.000.000

POLITEKNIK
NEGERI
JAKARTA



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Lampiran 9. Bukti Diskusi Bersama *Expert Panelist*



1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Sampel 1 *

	1	2	3	4	5	6	7	
Non-premium	☐	☐	☐	☐	☐	☐	☐	Premium

Sampel 1 *

	1	2	3	4	5	6	7	
Non-durabilitas	☐	☐	☐	☐	☐	☐	☐	Durabilitas

Sampel 1 *

	1	2	3	4	5	6	7	
Non-ergonomis	☐	☐	☐	☐	☐	☐	☐	Ergonomis

Sampel 2 *

	1	2	3	4	5	6	7	
Non-premium	☐	☐	☐	☐	☐	☐	☐	Premium

Sampel 2 *

	1	2	3	4	5	6	7	
Non-durabilitas	☐	☐	☐	☐	☐	☐	☐	Durabilitas

Sampel 2 *

	1	2	3	4	5	6	7	
Non-ergonomis	☐	☐	☐	☐	☐	☐	☐	Ergonomis

Sampel 40 *

	1	2	3	4	5	6	7	
Non-premium	☐	☐	☐	☐	☐	☐	☐	Premium

Sampel 40 *

	1	2	3	4	5	6	7	
Non-durabilitas	☐	☐	☐	☐	☐	☐	☐	Durabilitas

Sampel 40 *

	1	2	3	4	5	6	7	
Non-ergonomis	☐	☐	☐	☐	☐	☐	☐	Ergonomis



© Hak Cipta milik Politeknik Negeri Jakarta

Lampiran 11. Hasil Kuesioner *Semantic Differential* (SD) II Skala Likert

Gen Milenial – Konsep Premium

Sampel	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
1	6	6	5	6	5	4	5	6	6	5	7	6	5	6	4	3	2	6	5	5	3	6	7	6	6	6	7	6	6	7	6
2	7	7	7	4	7	5	5	6	7	4	1	4	6	5	3	7	5	5	7	5	4	6	6	6	7	6	7	7	7	5	7
3	5	3	7	4	6	4	5	7	6	6	7	7	7	7	6	5	4	6	7	4	5	7	6	7	7	6	6	7	6	7	6
4	5	6	5	5	7	4	4	5	6	6	1	5	7	5	6	4	3	7	5	6	3	7	5	6	7	7	6	6	7	7	7
5	5	5	7	5	5	6	5	7	7	6	7	4	7	6	3	5	5	6	7	6	6	7	6	5	7	6	6	6	6	7	5
6	4	7	7	7	6	5	5	5	7	6	7	7	5	5	3	6	2	6	6	5	4	7	6	5	7	7	7	5	7	5	6
7	4	7	7	4	7	4	6	6	6	6	1	4	6	5	5	3	7	6	5	6	4	7	6	5	6	6	6	7	5	6	6
8	6	7	5	4	7	3	6	5	6	5	7	7	6	4	5	4	3	5	5	6	5	6	6	6	6	6	6	5	6	6	6
9	5	7	5	6	6	5	5	5	6	6	1	6	7	6	3	4	3	6	4	5	4	7	5	5	7	6	6	6	6	7	6
10	5	7	5	6	5	4	4	6	6	6	1	6	7	5	6	7	6	5	5	4	5	7	6	5	6	6	6	6	5	7	6
11	5	7	6	7	6	5	5	6	6	5	1	5	5	4	7	6	5	5	4	4	4	7	6	5	6	6	6	6	5	7	6
12	4	7	5	6	7	5	4	6	1	6	1	6	6	7	4	7	3	4	7	5	5	6	6	7	6	6	6	6	5	5	6
13	5	4	7	6	7	6	4	6	5	6	7	6	7	4	3	3	6	5	5	5	4	6	6	6	6	6	5	6	5	6	7
14	5	7	7	6	7	5	4	6	6	6	7	7	6	3	4	6	7	7	5	5	5	5	6	5	7	6	6	6	7	6	6
15	6	4	5	7	6	4	4	5	6	5	1	7	7	6	5	4	4	5	5	6	6	6	6	5	6	6	7	6	5	7	5
16	5	6	6	6	7	6	5	6	6	5	7	6	7	4	7	7	4	4	7	5	4	7	6	5	7	6	7	6	6	7	7
17	5	7	5	7	7	6	4	5	6	5	1	7	7	3	4	7	5	6	5	4	5	6	6	5	6	5	7	6	6	6	5
18	3	7	5	5	7	6	5	5	6	6	7	6	6	6	7	2	6	5	5	5	4	7	5	6	7	6	6	6	7	6	6
19	5	6	7	5	7	6	4	6	1	5	1	5	7	5	4	4	7	7	7	6	5	6	6	6	6	7	6	6	6	5	6
20	3	7	6	7	7	6	4	5	6	6	1	7	6	4	7	5	7	7	5	6	5	6	7	5	6	6	6	6	6	7	6
21	5	6	7	5	5	6	4	7	5	5	7	5	7	5	6	4	6	7	7	6	4	6	5	6	6	6	7	6	6	6	7
22	4	7	6	7	7	5	4	7	1	5	1	7	7	4	6	4	6	6	5	5	6	5	6	7	6	6	6	7	6	6	5
23	5	7	5	5	5	5	5	5	6	4	7	5	6	6	5	4	4	7	6	6	5	5	6	5	7	5	6	6	6	6	7
24	4	7	5	7	7	4	5	6	1	4	7	7	7	4	4	3	4	5	4	5	5	5	6	6	7	6	5	6	6	6	7
25	6	5	6	5	7	5	4	6	4	5	7	7	6	7	7	4	6	6	5	6	5	6	6	6	6	7	6	6	6	7	6
26	4	7	7	5	7	6	4	6	6	5	7	6	6	5	3	4	4	5	4	7	6	5	7	5	6	6	6	6	6	6	7
27	5	4	6	4	6	5	5	6	6	6	1	6	6	5	6	2	5	5	5	5	4	7	6	5	5	6	6	6	7	6	6
28	5	5	5	4	5	4	5	6	6	4	1	7	7	7	6	4	6	7	2	6	4	7	6	6	6	6	6	7	6	5	5
29	4	6	5	5	7	4	5	6	5	5	1	7	7	7	5	2	5	7	5	6	6	7	6	6	7	6	6	6	6	6	7
30	5	5	6	5	7	5	5	5	5	5	7	7	7	4	5	3	7	7	4	5	4	6	7	5	6	7	6	6	6	6	6
31	4	6	4	5	6	3	6	5	6	6	1	7	7	5	6	2	4	6	6	5	5	5	6	6	6	6	6	5	7	7	6
32	6	2	3	3	5	3	3	4	1	5	1	6	7	3	7	3	4	5	3	4	5	6	6	6	7	7	5	7	6	6	6

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

33	4	1	3	4	5	3	3	4	1	4	1	6	7	3	3	4	6	3	2	6	3	5	7	5	5	6	6	6	5	6	
34	4	1	1	4	5	3	3	4	1	4	1	6	7	3	4	3	4	2	3	3	6	6	5	7	6	6	5	6	6	6	7
35	3	2	1	4	5	4	3	3	4	5	1	6	7	3	4	5	7	2	3	4	3	5	6	6	6	6	6	6	6	6	6
36	5	1	3	5	5	3	4	5	4	5	1	5	6	5	4	4	5	4	4	5	5	6	6	6	6	6	7	6	6	7	7
37	4	6	5	5	5	4	4	5	6	5	7	5	7	5	3	7	6	5	3	5	5	6	5	7	6	7	6	6	6	6	5
38	5	7	5	6	6	6	4	5	6	6	7	6	7	5	5	5	5	5	5	5	3	7	5	6	6	6	6	6	6	6	5
39	5	6	5	5	6	4	3	6	1	5	1	4	7	5	6	5	5	6	5	3	6	7	6	6	6	7	6	6	6	6	7
40	5	7	5	6	7	3	5	5	6	3	7	6	7	6	7	6	6	5	7	6	6	7	5	5	6	6	6	6	6	6	6

Gen Milenial – Konsep Durabilitas

Sampel	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
1	5	7	5	6	4	5	5	5	6	5	7	5	7	5	3	3	4	5	6	4	3	7	6	7	7	7	7	7	6	7	6
2	7	6	6	4	7	5	4	6	5	4	1	7	7	5	3	7	5	5	6	5	3	7	5	7	6	7	6	6	6	6	5
3	3	5	7	4	6	4	4	7	6	6	7	6	6	6	6	3	4	7	5	4	5	7	6	6	7	6	6	6	6	6	7
4	5	7	5	5	7	4	5	5	6	6	1	5	7	5	4	4	5	4	5	3	4	7	6	6	7	6	7	7	6	6	6
5	4	7	7	5	5	4	4	7	6	6	7	7	5	6	4	7	4	5	4	4	5	5	6	7	6	6	7	7	6	6	6
6	4	6	7	7	6	5	4	5	6	6	7	7	7	4	3	7	3	5	5	3	4	6	7	7	6	6	6	6	6	6	6
7	5	6	6	4	7	4	5	6	6	6	7	5	7	4	6	3	6	7	5	5	4	6	6	5	7	7	7	6	6	7	7
8	5	6	7	4	7	3	6	5	6	5	1	6	7	3	4	5	3	6	5	5	5	7	6	6	7	6	6	6	7	7	7
9	6	7	7	6	6	5	6	5	6	6	7	4	7	6	2	3	4	5	7	4	6	7	5	6	7	6	6	7	6	6	6
10	5	6	5	6	5	4	5	5	6	6	1	7	6	5	6	6	5	6	6	4	4	7	5	7	7	6	7	7	6	6	6
11	5	6	6	7	6	5	5	6	6	5	1	5	5	3	4	6	3	5	5	5	4	6	6	6	7	6	7	6	6	6	6
12	6	7	5	6	7	5	4	6	2	6	7	6	7	6	5	6	5	5	5	4	5	6	5	7	6	6	7	7	6	6	7
13	5	5	7	6	7	6	4	6	4	6	7	7	7	4	3	4	5	5	6	3	4	6	6	6	6	6	7	6	6	6	6
14	5	7	7	6	7	5	5	6	6	6	7	5	7	4	3	6	4	6	7	5	6	5	5	6	7	7	7	7	6	7	6
15	5	3	5	7	6	4	5	5	5	5	1	6	6	4	6	5	4	4	5	6	6	7	6	6	7	6	7	7	7	5	6
16	5	7	6	6	7	6	4	6	6	5	7	6	6	4	4	5	4	5	4	5	2	5	6	6	7	7	6	6	6	7	6
17	4	6	5	7	7	6	3	5	6	5	7	7	7	5	4	6	5	5	5	5	5	7	7	6	7	7	7	6	7	7	6
18	5	7	5	5	7	6	5	5	6	6	7	6	7	6	6	4	4	5	5	3	6	6	6	6	6	6	6	7	6	6	6
19	5	7	6	5	7	6	5	6	1	5	1	5	7	3	4	5	4	6	5	4	4	7	6	7	7	6	6	6	7	7	7
20	4	7	6	7	7	5	5	5	6	6	1	5	7	4	6	5	5	6	6	6	4	6	5	6	7	7	7	7	6	6	7
21	4	7	7	5	4	5	4	6	5	5	7	7	7	4	4	5	5	6	5	5	5	6	6	6	7	6	7	7	6	6	5
22	6	7	6	7	7	5	4	6	1	5	4	7	7	4	5	3	5	5	5	4	5	5	5	6	6	7	6	6	7	7	6
23	5	7	7	5	5	4	5	5	6	4	7	7	5	6	5	4	5	6	6	5	3	6	6	7	6	6	6	7	7	7	6
24	5	6	6	7	7	4	4	6	4	4	1	7	7	4	5	4	5	5	4	5	4	5	6	6	7	6	6	6	7	6	6



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

25	5	5	6	5	7	5	5	6	6	5	7	5	7	7	5	4	5	7	5	6	5	5	6	7	7	7	6	7	6	6	7
26	5	6	7	5	7	4	5	6	6	5	7	6	7	4	5	4	5	5	4	7	6	7	6	7	7	6	7	6	6	6	6
27	5	6	7	4	6	4	5	6	6	6	1	7	7	6	5	3	4	7	4	5	6	7	7	7	7	6	7	7	6	6	6
28	5	4	5	4	5	4	5	6	6	4	1	7	6	7	6	4	6	5	4	5	4	7	5	7	6	6	7	6	7	6	7
29	5	7	5	5	7	4	4	6	6	5	7	6	7	7	5	4	5	7	5	6	5	6	6	7	6	7	6	7	6	6	6
30	5	6	6	5	7	5	4	5	6	5	7	6	7	3	5	5	5	4	4	5	6	5	5	6	6	6	7	6	7	7	5
31	5	7	6	5	6	3	6	5	6	6	7	7	6	4	4	2	5	6	6	3	4	6	7	7	7	7	7	6	5	7	6
32	4	2	5	3	5	3	3	4	4	5	7	6	7	2	7	3	5	4	4	5	5	6	7	7	7	6	6	7	6	6	6
33	4	2	4	4	5	3	3	4	4	4	7	6	7	3	3	6	6	4	3	5	2	5	6	6	7	7	6	6	6	5	7
34	5	1	4	4	5	4	3	4	4	4	7	7	6	4	4	4	5	4	4	4	4	5	7	6	6	6	6	6	7	7	6
35	5	1	3	4	5	4	3	3	6	5	1	5	7	4	5	4	6	6	4	4	5	6	6	6	5	6	6	7	6	6	7
36	5	1	4	5	5	3	4	5	6	5	1	5	7	3	4	6	5	6	4	5	4	4	5	7	6	7	7	6	6	7	6
37	4	7	6	5	5	3	4	5	6	5	7	4	7	3	3	4	5	4	4	5	6	6	6	6	6	6	6	6	6	6	5
38	5	6	6	6	6	5	4	5	6	6	7	7	7	4	5	4	5	5	3	6	4	6	6	7	6	6	6	7	7	7	6
39	4	7	5	5	6	4	3	6	6	5	1	5	7	3	5	4	4	3	5	4	5	6	7	6	7	7	7	7	6	7	6
40	4	7	6	6	7	3	4	5	6	3	7	5	6	6	7	4	6	4	6	6	6	7	7	6	6	7	7	5	6	7	5

Gen Milenial – Konsep Ergonomis

Sampel	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
1	5	5	5	6	5	4	5	6	6	3	3	5	7	5	4	3	5	5	5	4	5	7	6	7	6	7	7	7	7	7	7
2	5	7	6	4	7	4	5	6	5	4	1	6	6	6	3	6	3	4	5	4	4	6	7	5	6	6	7	6	7	7	6
3	4	7	6	4	6	4	5	6	4	6	7	5	7	6	5	4	4	7	7	5	4	6	6	7	7	7	6	7	7	7	7
4	5	6	5	5	7	4	5	5	4	6	1	4	7	6	6	4	4	5	6	5	4	6	6	7	6	7	6	6	7	6	6
5	5	7	7	5	5	6	5	6	4	6	4	6	5	5	5	7	4	5	5	4	6	7	6	6	7	7	6	7	7	6	6
6	5	7	7	7	6	5	4	5	7	6	1	6	7	3	3	6	4	5	5	4	5	7	7	7	7	7	7	7	7	7	6
7	5	7	6	4	7	4	5	6	7	6	7	4	7	4	7	3	4	7	6	4	5	6	5	5	7	6	6	7	6	7	6
8	6	7	5	4	7	3	4	5	5	5	1	7	7	4	6	4	3	5	7	5	5	6	7	7	6	7	7	6	7	6	7
9	6	7	6	6	6	5	6	5	7	6	7	6	6	6	3	4	3	5	5	3	4	6	7	7	7	6	7	7	5	7	6
10	5	7	5	6	5	4	4	7	5	6	1	7	5	4	6	5	3	7	7	6	5	6	6	7	6	7	7	6	6	6	6
11	5	7	6	7	6	5	4	6	7	5	1	7	7	4	6	6	3	6	6	3	4	6	6	6	7	6	6	7	7	6	7
12	4	7	5	6	7	5	4	6	2	6	7	7	7	6	6	6	4	5	6	4	4	6	5	6	7	6	6	7	6	7	7
13	5	6	7	6	7	6	5	6	4	6	1	6	7	3	3	5	3	5	5	5	4	7	6	6	7	6	7	7	6	6	6
14	5	7	7	6	7	5	5	6	5	6	7	6	6	4	4	4	4	6	6	6	5	5	6	6	6	7	6	6	6	6	7
15	5	2	5	7	6	4	5	5	5	5	1	7	6	5	6	5	4	7	7	5	5	5	6	6	6	7	6	7	6	6	7
16	5	7	6	6	7	6	5	6	5	5	4	6	7	3	5	2	5	6	6	5	5	7	5	7	7	6	7	7	7	7	7



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

17	5	7	5	7	7	6	4	5	6	5	7	7	7	4	5	5	5	6	7	4	4	6	6	7	7	6	7	7	7	5	7
18	5	7	5	5	7	6	4	5	7	6	7	5	6	5	6	6	5	4	6	5	4	6	6	6	7	6	7	7	6	6	7
19	4	7	6	5	7	5	5	5	4	5	7	7	6	5	4	4	4	7	6	4	5	6	6	7	6	7	7	7	7	7	7
20	3	7	6	7	7	4	4	5	6	6	1	5	6	4	5	6	5	6	6	6	5	7	6	7	6	6	6	6	7	6	6
21	5	7	7	5	5	5	4	6	4	5	1	6	6	4	5	5	5	6	6	6	5	6	7	7	7	7	7	6	7	7	7
22	5	7	6	7	7	4	5	6	4	5	7	5	7	4	6	4	4	7	5	5	5	5	7	7	7	7	7	7	6	7	7
23	5	7	6	5	5	5	5	5	6	4	1	6	5	6	5	6	5	6	6	7	5	5	7	6	7	6	7	6	6	6	7
24	4	7	5	7	7	4	4	5	4	4	1	6	7	3	5	5	5	5	5	5	5	6	5	7	7	6	6	7	7	7	7
25	4	6	6	5	7	5	5	6	5	5	7	5	7	6	5	4	4	6	7	5	5	6	6	6	7	6	7	7	7	6	6
26	5	7	7	5	7	6	5	6	5	5	7	6	7	4	6	3	5	5	4	6	5	6	5	7	6	7	5	7	7	7	6
27	5	6	6	4	6	5	6	6	6	6	1	5	7	5	5	5	5	7	4	5	6	7	7	7	7	7	6	6	7	7	7
28	5	3	5	4	5	4	6	6	5	4	1	6	7	7	6	3	5	6	5	5	5	7	6	7	7	7	6	6	6	7	6
29	5	7	5	5	7	5	5	6	6	5	7	5	6	7	7	6	5	6	5	6	6	5	6	6	7	6	7	7	7	7	6
30	5	7	7	5	7	5	5	5	5	5	1	7	7	3	6	5	5	6	6	7	6	6	7	7	7	6	6	7	6	7	5
31	6	7	5	5	6	3	6	5	7	6	1	7	7	4	7	6	5	6	5	7	4	5	5	6	7	7	7	7	7	7	7
32	3	1	6	3	5	3	3	7	3	5	7	6	7	2	6	5	6	5	6	5	6	5	6	6	6	7	6	6	7	7	7
33	5	1	5	4	5	3	3	7	1	4	7	7	7	2	5	6	7	5	4	5	3	5	6	6	6	7	7	7	7	7	6
34	5	1	5	4	5	4	3	7	3	4	7	7	7	4	5	5	6	6	5	5	4	6	6	7	7	7	7	7	6	7	6
35	4	1	4	4	5	3	4	6	6	5	1	7	7	3	6	5	5	7	5	5	4	5	7	7	6	7	7	7	7	7	6
36	5	1	4	5	5	3	4	5	5	5	1	5	6	3	4	4	6	7	5	4	4	6	6	7	7	6	7	7	7	7	6
37	5	7	6	5	5	3	4	5	7	5	1	6	6	2	4	4	4	5	4	6	6	7	7	6	7	7	7	7	7	7	6
38	5	7	6	6	6	4	4	5	6	6	1	5	6	3	4	3	6	5	4	6	5	7	7	6	7	6	6	6	6	7	7
39	4	6	6	5	6	4	3	5	6	5	1	5	6	3	5	6	5	3	4	5	6	6	7	7	6	6	7	7	7	6	6
40	4	7	6	6	7	3	4	5	5	3	7	6	6	7	7	3	6	3	5	6	5	7	7	6	7	6	6	6	7	7	6

Gen Z – Konsep Premium

Sampel	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
1	4	7	5	6	2	6	6	4	4	5	6	5	4	5	5	5	6	5	6	3	3	5	7	4	6	5	6	5	5	6	6
2	4	7	3	6	4	7	5	5	6	6	5	6	4	6	5	6	6	2	7	6	3	5	7	4	7	5	4	7	3	4	3
3	5	7	5	7	4	5	1	5	6	6	7	6	3	6	6	5	6	3	5	5	6	7	7	4	6	7	4	6	6	7	6
4	3	7	6	6	5	7	1	5	4	5	6	4	5	5	6	3	6	3	4	4	3	4	7	3	6	2	4	3	6	7	5
5	6	7	6	6	6	6	3	5	5	5	6	5	4	6	7	7	7	3	5	5	7	6	7	4	7	5	5	6	7	7	5
6	6	7	6	7	5	7	2	6	3	6	6	6	5	6	4	2	5	5	5	5	3	5	7	4	7	6	4	6	5	4	7
7	6	7	6	6	6	6	1	5	5	6	4	7	4	6	3	3	6	6	5	5	1	4	7	3	6	3	5	3	6	5	6
8	5	7	5	6	2	6	1	4	4	5	6	6	5	5	3	2	5	2	6	6	6	5	7	4	6	2	4	6	5	7	3



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

9	7	7	4	6	2	6	1	5	5	6	4	6	4	6	5	3	6	3	5	6	1	5	7	5	6	5	6	4	6	4	3
10	5	5	3	6	5	7	4	4	5	5	6	3	4	4	3	1	5	3	3	6	5	6	7	4	5	5	4	5	5	3	5
11	6	7	5	6	3	6	5	5	4	4	4	2	3	5	5	4	6	3	3	4	5	7	7	4	5	3	5	4	4	3	2
12	6	4	4	6	3	7	5	5	3	5	3	7	3	5	2	4	4	5	5	6	1	6	7	5	6	5	4	3	6	5	7
13	6	7	6	6	5	6	6	6	3	5	5	7	4	6	5	6	6	3	6	5	6	7	7	5	7	7	4	7	7	6	6
14	6	7	6	5	2	5	5	5	4	5	6	3	5	5	6	5	6	6	6	7	6	6	7	4	6	3	7	4	7	6	7
15	5	7	3	6	6	7	5	4	4	5	6	5	4	5	3	2	5	1	3	4	6	4	7	4	5	3	2	4	5	3	6
16	7	7	5	7	4	6	3	5	4	6	7	7	4	6	6	5	7	2	3	5	5	5	7	5	5	5	6	3	6	4	7
17	6	7	6	6	4	7	3	5	3	5	3	7	4	5	3	3	5	6	4	5	1	6	7	5	6	3	5	6	5	6	6
18	5	7	5	6	2	6	4	5	4	6	6	6	5	5	3	2	5	2	7	7	5	6	7	5	7	5	6	5	4	6	6
19	5	5	5	5	3	6	5	5	3	6	3	7	4	5	6	6	6	3	6	6	6	7	7	4	7	6	4	6	7	7	5
20	6	7	6	5	3	6	6	5	4	6	4	5	4	5	2	3	6	5	4	5	1	6	7	4	6	2	6	5	5	7	7
21	6	7	4	6	3	7	4	5	4	6	7	3	4	5	2	1	6	6	7	3	3	6	7	4	7	7	7	2	7	6	1
22	7	5	4	6	6	6	6	5	3	6	4	5	5	5	5	3	6	6	4	4	4	7	7	5	5	3	6	5	7	3	5
23	7	7	6	6	2	6	6	5	4	6	3	6	5	5	5	2	6	6	5	4	2	5	7	4	7	3	5	4	5	2	6
24	7	7	6	6	4	5	6	5	3	6	3	6	4	5	7	4	6	6	5	6	5	7	6	4	6	5	6	2	7	7	7
25	5	7	4	6	1	6	5	5	3	5	6	2	4	5	2	6	5	2	6	5	3	4	6	5	7	6	4	2	6	7	2
26	6	7	5	7	2	7	3	5	4	5	6	6	4	6	4	6	6	2	6	5	5	4	6	4	7	7	3	4	6	6	5
27	3	5	2	7	3	6	7	4	3	7	5	3	3	4	1	2	7	2	7	5	3	6	6	4	5	7	5	6	5	4	3
28	6	7	3	6	5	6	6	5	3	6	4	3	4	4	3	2	5	2	6	1	5	6	7	5	5	3	6	3	4	1	5
29	6	7	5	6	2	7	4	5	2	6	3	5	4	5	5	6	6	3	5	4	1	5	6	4	6	5	7	2	6	5	6
30	5	7	4	6	1	7	7	6	3	5	5	4	4	5	5	5	6	3	6	4	5	6	7	4	7	6	4	7	5	7	3
31	7	7	6	6	2	5	6	4	2	5	7	5	4	4	1	1	4	7	6	6	5	7	6	5	5	2	4	5	6	5	1
32	1	1	2	3	5	3	1	4	1	4	1	1	1	3	1	1	3	3	3	1	1	6	2	4	2	1	2	1	3	2	1
33	1	1	2	3	5	2	1	4	1	4	1	1	1	3	1	2	3	2	4	1	1	5	2	4	2	1	5	1	2	1	3
34	1	1	3	3	6	3	1	3	1	4	1	1	1	2	1	1	3	2	3	1	1	4	2	5	1	1	1	1	4	1	2
35	3	1	1	3	1	3	3	4	3	5	6	3	1	3	1	1	4	2	5	6	3	6	1	5	5	2	6	1	3	1	5
36	5	1	4	5	4	3	5	4	2	6	6	6	2	3	1	2	5	2	5	5	5	4	2	5	5	3	1	3	4	2	5
37	5	7	4	6	4	6	6	5	2	6	6	5	3	4	2	6	6	3	7	4	3	7	2	5	5	5	4	2	6	3	6
38	5	7	4	6	3	6	4	5	3	5	6	6	4	5	5	6	6	3	6	4	5	5	1	5	6	3	3	4	7	5	7
39	5	1	3	6	4	5	5	5	2	6	2	6	3	3	2	3	6	3	5	3	3	4	2	4	5	5	6	2	5	2	3
40	7	7	5	6	4	6	3	5	3	6	6	5	4	5	5	4	5	3	6	4	3	4	3	5	6	3	6	2	7	6	7



Gen Z – Konsep Durabilitas

Sampel	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
1	7	7	6	6	2	6	6	5	5	4	6	5	6	5	6	4	6	6	5	4	2	6	7	3	7	2	5	5	5	5	5
2	5	5	3	6	5	7	5	5	5	6	4	5	3	5	6	7	6	3	5	6	6	4	5	4	6	6	6	6	3	5	2
3	7	7	5	7	2	6	6	5	5	6	6	6	3	6	6	6	7	7	6	6	6	6	7	4	7	5	7	4	6	7	7
4	6	7	5	5	3	6	7	5	3	5	5	6	4	5	5	2	6	6	7	6	5	5	7	3	5	5	5	5	5	5	6
5	7	7	4	6	6	6	6	5	5	6	6	5	4	6	6	7	7	3	6	6	6	7	6	3	7	6	6	5	6	6	6
6	6	7	4	7	5	7	6	5	4	6	6	7	5	7	5	2	6	7	3	6	5	6	6	3	7	5	6	4	5	5	7
7	7	7	6	7	7	6	7	5	5	6	6	7	4	7	5	5	6	5	6	6	5	5	7	4	7	2	3	3	6	4	7
8	4	6	3	7	5	7	7	4	4	5	6	5	3	5	5	3	6	6	3	7	1	7	7	4	5	3	5	6	6	5	6
9	6	7	5	7	6	4	7	5	4	5	6	7	4	7	5	3	7	6	5	7	6	6	7	5	6	6	5	6	5	5	6
10	6	4	5	7	4	7	6	5	5	5	6	4	3	4	3	2	6	6	2	5	2	5	7	4	4	6	6	6	3	4	3
11	7	6	3	6	5	7	4	7	4	5	5	3	3	5	5	4	7	6	2	5	2	6	6	5	6	2	4	6	2	4	5
12	5	3	3	6	6	7	6	5	3	5	4	7	4	6	6	3	6	6	3	7	5	5	7	5	4	2	6	5	5	4	6
13	7	5	6	7	3	6	7	5	4	6	5	6	4	7	5	6	7	6	6	6	5	5	6	4	7	6	5	4	6	5	6
14	7	7	6	6	4	6	6	5	4	5	6	5	4	6	5	5	7	6	7	6	6	6	6	5	6	5	5	5	7	5	7
15	5	5	6	6	6	7	6	5	3	5	6	4	4	5	4	2	7	3	2	6	2	5	6	4	4	5	3	4	4	4	6
16	6	7	6	6	3	6	7	5	4	6	6	5	3	6	6	5	7	5	4	7	5	5	7	4	6	6	3	4	6	5	7
17	7	6	6	6	2	6	6	5	3	6	5	7	4	5	5	3	6	6	5	7	6	7	6	5	6	5	5	5	4	5	6
18	7	7	6	6	4	7	7	5	5	6	5	7	4	5	5	3	6	3	6	6	4	5	6	5	6	5	5	6	3	5	5
19	7	4	6	6	5	7	6	5	3	6	4	7	5	5	6	5	6	5	5	7	6	6	6	5	6	5	5	6	6	5	3
20	7	7	6	6	3	7	6	5	4	6	4	7	4	6	5	4	7	5	3	6	5	5	6	4	6	3	7	5	4	6	6
21	7	7	6	5	4	6	5	5	5	5	6	6	4	5	5	2	7	3	6	4	5	5	6	4	5	6	5	2	6	5	5
22	7	4	3	5	1	6	6	5	3	6	4	7	4	5	5	3	7	3	6	5	3	6	5	4	5	5	6	7	6	4	7
23	6	7	6	6	3	7	7	5	3	6	4	7	5	5	6	2	6	6	4	5	5	5	6	5	5	5	5	6	5	4	6
24	5	4	5	6	2	6	3	4	3	6	3	7	2	5	6	5	5	3	2	5	3	6	5	5	7	5	7	4	6	6	5
25	7	7	5	7	4	6	6	5	4	5	5	5	4	5	5	6	7	6	4	6	5	5	6	5	7	5	5	6	6	5	6
26	7	7	6	6	4	7	3	5	4	6	5	6	4	6	6	5	7	6	5	6	6	6	5	5	7	6	6	3	6	5	6
27	6	4	5	6	5	7	6	5	3	6	5	6	5	4	5	2	7	2	6	6	5	7	4	5	6	5	6	3	5	5	5
28	7	5	5	6	3	6	3	4	3	6	4	3	2	4	5	3	6	3	3	2	3	7	3	4	4	5	4	4	3	3	6
29	7	7	6	7	2	6	7	5	2	6	3	7	3	5	5	5	6	3	4	5	5	5	4	5	6	3	2	5	6	6	6
30	6	7	6	6	4	7	3	4	3	5	4	2	4	5	6	5	6	3	5	6	3	6	5	5	7	5	5	4	5	7	3
31	5	7	5	6	3	6	7	5	3	5	6	6	3	4	5	2	5	4	3	7	5	6	4	5	3	3	7	3	6	6	6
32	2	7	6	4	1	5	5	7	1	6	2	3	1	4	3	3	5	3	2	3	1	5	3	4	4	2	3	5	4	3	5
33	2	7	6	5	4	3	6	6	1	6	2	3	1	4	3	3	4	3	3	4	5	6	4	4	5	2	6	7	1	3	5

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

34	2	7	4	3	2	5	6	5	1	6	2	5	1	3	3	3	4	2	4	3	5	4	4	4	2	2	1	3	4	2	5
35	2	3	3	3	3	3	2	5	2	5	2	5	1	3	3	3	4	2	3	7	3	5	3	4	4	3	5	4	2	1	6
36	3	6	6	5	4	3	1	4	2	6	3	5	1	3	3	4	6	2	2	4	2	5	2	5	3	3	3	6	4	2	6
37	6	7	6	7	4	6	6	4	2	6	6	5	3	5	5	5	7	1	4	4	5	6	3	5	5	3	5	3	5	2	6
38	6	7	6	7	1	7	7	5	3	6	6	6	3	5	6	6	7	2	5	5	5	5	3	5	6	5	3	3	7	4	6
39	5	6	6	6	4	5	2	5	1	6	2	6	2	3	3	4	6	1	3	4	2	5	2	5	6	6	6	1	4	2	5
40	7	7	4	6	3	7	7	4	2	6	6	6	4	5	5	4	7	5	7	6	5	5	2	4	3	3	4	5	7	6	6

Gen Z – Konsep Ergonomis

Sampel	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
1	2	7	3	6	6	6	5	4	4	5	3	6	5	4	6	6	6	2	4	3	2	6	4	3	6	5	4	6	6	6	5
2	6	5	3	6	3	6	2	4	5	4	3	3	5	4	6	5	6	1	6	7	2	6	6	4	6	3	5	5	4	4	5
3	7	7	5	6	5	7	6	6	5	6	6	6	3	5	7	4	7	3	7	7	7	7	5	5	5	7	7	7	5	5	6
4	4	5	4	6	3	5	3	4	3	5	5	5	4	4	7	3	6	5	6	5	4	5	3	4	7	1	4	4	5	5	3
5	7	7	5	6	4	6	4	5	5	6	6	3	4	6	6	6	7	6	5	4	7	6	6	4	6	5	6	4	7	5	5
6	7	7	2	7	6	7	5	4	3	6	4	5	3	5	3	2	6	6	5	6	4	5	6	4	6	6	5	6	4	5	6
7	6	7	4	6	6	6	4	4	4	5	5	7	3	4	3	3	6	6	5	7	3	7	6	4	6	3	6	4	6	6	5
8	5	5	5	6	6	6	5	4	4	6	6	2	4	5	4	2	5	6	4	6	1	6	3	5	5	2	3	7	6	6	5
9	6	7	6	7	5	7	4	4	5	5	5	7	3	4	5	3	6	6	6	7	3	3	6	5	6	7	4	4	4	5	6
10	5	4	6	6	2	6	4	4	5	6	6	5	3	4	3	2	5	7	4	7	4	6	5	5	5	7	7	4	4	4	3
11	7	6	5	6	6	6	5	4	5	6	5	5	3	5	6	4	6	3	5	4	4	7	7	4	5	3	5	5	2	4	6
12	5	5	6	6	3	6	5	4	3	5	4	7	4	5	5	5	5	7	5	6	3	6	7	4	4	3	4	5	6	6	5
13	7	7	7	7	6	7	7	5	3	6	4	3	2	5	6	4	7	3	7	2	7	6	6	5	6	7	7	5	6	5	7
14	7	7	5	6	4	6	7	5	4	5	6	6	2	6	6	6	7	3	6	7	6	5	6	5	5	6	2	4	6	5	6
15	6	5	6	6	5	7	5	4	4	5	5	3	4	5	5	3	6	6	4	4	3	5	5	5	6	3	5	6	5	4	5
16	7	7	5	5	1	6	5	5	4	6	4	1	3	6	7	3	7	3	5	4	5	6	6	4	5	5	4	5	5	5	6
17	7	7	3	6	5	6	4	5	3	6	4	6	4	5	6	3	6	2	6	7	4	7	7	5	6	3	6	4	3	5	5
18	7	7	3	6	2	6	5	5	5	5	5	7	5	5	5	3	7	6	6	7	5	6	7	4	6	6	7	4	2	5	5
19	5	7	6	6	6	6	6	5	4	6	5	7	3	5	6	5	6	6	7	6	6	7	6	5	5	6	6	7	6	5	5
20	7	7	6	6	6	6	5	4	4	6	3	5	3	5	5	5	7	5	3	7	4	5	6	5	6	2	5	7	4	6	6
21	5	7	6	6	6	7	7	5	4	4	4	3	3	5	5	3	7	6	7	2	7	6	7	5	5	7	5	4	7	6	5
22	7	4	6	6	1	7	6	4	4	6	4	7	4	5	6	2	7	3	2	3	3	7	2	5	6	3	7	5	6	4	5
23	7	7	5	6	1	6	7	4	3	6	5	7	5	5	7	4	6	2	2	5	4	6	3	5	5	3	5	6	6	4	5
24	5	4	6	6	7	7	5	4	2	6	5	7	4	4	7	3	6	5	1	3	3	6	4	4	6	6	7	6	5	6	5
25	6	7	3	6	6	6	6	5	4	5	5	5	3	5	5	5	6	3	5	7	5	4	4	4	6	6	6	5	5	5	5



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

26	6	7	3	6	4	7	6	5	4	6	5	6	3	5	7	5	7	2	7	5	6	5	3	5	6	6	4	5	5	5	5
27	7	4	5	7	5	7	6	5	4	6	5	6	4	4	6	1	7	3	6	7	5	5	2	5	6	6	4	4	4	5	5
28	6	5	6	7	5	6	6	4	3	6	5	3	3	4	6	2	6	7	2	4	3	6	2	4	5	3	7	5	2	2	5
29	7	7	3	7	7	7	5	5	2	6	4	6	3	5	6	5	6	6	6	7	4	6	1	4	7	5	7	5	5	5	5
30	7	7	6	7	5	6	7	5	2	5	4	2	2	5	7	4	7	7	6	2	4	5	3	4	6	6	7	6	4	7	5
31	6	7	4	6	5	6	4	4	3	5	5	6	3	4	4	4	5	3	2	7	2	7	2	5	5	2	7	5	5	6	2
32	4	5	7	5	7	5	7	5	1	6	2	2	1	3	3	3	3	6	4	5	5	6	1	4	6	3	2	3	5	2	3
33	2	6	6	5	1	3	6	5	1	6	2	5	1	3	3	3	4	7	4	5	4	5	2	5	6	3	4	1	3	2	6
34	2	6	6	4	7	5	6	5	1	6	2	5	1	2	5	3	4	7	3	5	4	5	2	5	6	3	1	3	5	1	5
35	6	4	7	3	5	3	4	5	2	5	3	5	1	3	3	4	4	6	2	6	3	6	2	5	5	1	7	1	2	1	5
36	5	6	7	5	5	3	6	4	1	6	3	5	2	4	3	2	5	4	4	6	3	5	3	5	5	5	2	4	3	1	6
37	5	7	4	6	6	6	4	4	2	6	4	4	2	4	5	5	6	3	3	1	5	7	4	4	5	5	5	3	6	3	5
38	7	7	4	6	6	7	5	5	2	6	5	6	3	4	6	5	7	5	5	4	5	6	3	5	6	5	6	4	6	5	5
39	7	6	3	6	2	4	5	5	1	6	4	6	3	3	3	4	6	4	5	7	5	4	2	4	6	7	4	6	6	2	5
40	7	7	6	7	6	6	5	4	3	5	5	7	4	4	4	5	6	5	5	7	4	4	4	4	5	2	4	6	5	5	6

**POLITEKNIK
NEGERI
JAKARTA**

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

© Hak Cipta milik Politeknik Negeri Jakarta

Lampiran 12. *Dataset Input Random Forest*

Sampel _Code	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	GenMile nial_Pre mium_M ean	GenMile nial_Dura bilitas_M ean	GenMile nial_Ergo nomis_M ean	GenZ_Pr emium_ Mean	GenZ_Du rabilitas_ Mean	GenZ_Er gonomis_ Mean
1	3	1	1	2	4	3	2	1	1	2	1	1	2	5.4516	5.5484	5.4516	5.0645	5.1290	4.7097
2	6	2	1	1	2	2	2	1	4	3	2	2	4	5.6452	5.4516	5.2903	5.0968	5.0000	4.5161
3	3	1	6	1	4	3	1	1	1	6	1	3	2	5.9032	5.6129	5.8065	5.4516	5.8387	5.8387
4	4	3	4	1	4	3	1	1	1	6	1	3	3	5.4839	5.3548	5.3871	4.6774	5.1613	4.4194
5	4	3	4	1	4	3	1	1	2	6	1	3	3	5.8387	5.6452	5.7097	5.6774	5.7419	5.4516
6	4	3	3	2	1	1	1	1	1	6	1	2	2	5.7097	5.5806	5.7097	5.2258	5.4839	5.0323
7	1	5	1	1	1	1	1	1	1	1	1	2	2	5.4516	5.7419	5.6774	4.9032	5.5806	5.0645
8	3	1	1	2	2	2	1	1	2	6	1	2	2	5.4839	5.4516	5.5161	4.7097	5.0323	4.6774
9	2	5	1	1	1	1	1	1	1	1	1	3	2	5.3548	5.7097	5.7097	4.8065	5.6774	5.1935
10	5	1	1	3	1	1	1	1	2	6	1	2	2	5.5161	5.5806	5.5806	4.5484	4.6774	4.7742
11	5	1	3	3	1	3	1	1	2	4	1	3	4	5.4194	5.3226	5.6452	4.4839	4.7419	4.9677
12	2	5	2	2	1	1	4	1	1	1	2	2	2	5.3226	5.7419	5.7097	4.7419	5.0000	4.9677
13	6	2	7	1	4	2	1	1	1	6	1	3	8	5.4839	5.5484	5.4516	5.7419	5.5806	5.5484
14	3	1	6	1	4	2	1	1	1	6	1	3	3	5.8387	5.9032	5.7419	5.4194	5.6774	5.3871
15	5	1	1	3	1	1	1	1	2	6	1	2	2	5.3871	5.3871	5.4516	4.4839	4.6452	4.8387
16	7	4	4	3	1	1	3	1	1	6	1	3	8	5.9355	5.5484	5.7742	5.2903	5.4194	4.8387
17	1	5	4	2	1	1	1	1	1	6	2	1	2	5.4516	5.8710	5.9032	4.9355	5.3548	5.0323
18	1	5	4	1	1	3	1	1	1	6	1	1	2	5.6774	5.7097	5.8065	5.1613	5.3226	5.2903
19	3	1	6	1	4	2	1	1	1	6	1	3	4	5.4839	5.3871	5.7742	5.3548	5.4516	5.7097
20	4	3	1	2	1	1	1	1	1	6	2	2	2	5.7419	5.7097	5.5806	4.9355	5.3226	5.1935
21	6	2	7	1	4	1	1	1	3	6	1	3	6	5.8065	5.6129	5.6774	4.9355	5.0645	5.3548
22	4	3	5	2	1	1	4	1	1	6	2	2	2	5.4839	5.4516	5.8710	5.0968	4.9355	4.7419
23	3	1	4	2	2	2	1	1	1	6	2	3	4	5.5484	5.6774	5.6129	4.9032	5.3226	4.9032



- Hak Cipta :**
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
 2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

© Hak Cipta milik Politeknik Negeri Jakarta

24	3	1	1	1	2	2	1	1	2	2	1	2	5	5.3226	5.3226	5.4194	5.4516	4.7097	5.0000
25	5	3	4	3	1	1	1	1	3	6	1	3	6	5.8387	5.8710	5.7742	4.5806	5.4839	5.0968
26	3	1	4	1	4	2	1	1	2	6	1	3	4	5.6129	5.8065	5.7742	5.1290	5.5484	5.1935
27	4	3	4	3	1	1	1	1	5	6	1	3	7	5.2581	5.6774	5.7742	4.5161	5.0645	5.0323
28	3	1	1	1	2	2	2	1	1	6	1	1	3	5.3548	5.3871	5.4194	4.4194	4.1613	4.5161
29	1	5	4	2	1	1	1	1	1	6	2	3	2	5.6129	5.8387	6.0000	4.8065	4.9677	5.2903
30	6	2	7	1	4	1	1	1	1	6	1	3	4	5.6129	5.5161	5.7742	5.1290	4.9032	5.1613
31	1	5	1	1	1	3	1	1	2	6	1	1	2	5.3226	5.6129	5.8065	4.7097	4.8710	4.5484
32	3	1	1	3	3	2	5	2	1	6	3	3	1	4.6774	5.0645	5.2581	2.1935	3.6129	4.0000
33	1	5	3	3	3	2	5	2	1	6	3	3	1	4.2903	4.8387	5.1613	2.2581	4.0000	3.8387
34	1	5	1	3	3	2	5	2	1	6	3	3	1	4.2581	4.9355	5.4194	2.0968	3.4516	4.0323
35	3	1	1	1	1	3	5	2	6	6	3	3	2	4.4516	4.8710	5.2581	3.1290	3.3548	3.8387
36	5	1	4	3	2	2	5	2	6	5	3	3	2	4.8710	4.9677	5.0645	3.7097	3.6774	4.1290
37	1	5	4	2	1	3	1	1	1	6	1	3	3	5.3871	5.1935	5.4194	4.6774	4.7419	4.4839
38	1	5	1	1	1	3	1	1	1	6	1	2	2	5.5806	5.6774	5.4194	4.8387	5.0968	5.1935
39	3	1	1	1	2	2	5	2	1	2	3	2	4	5.2258	5.2581	5.2903	3.8387	4.0000	4.5484
40	1	5	1	2	1	3	1	1	1	6	1	2	2	5.8065	5.7097	5.6774	4.8710	5.0968	5.0645



Lampiran 13. Source Code Random Forest

BAGIAN 1: Import & Konfigurasi Lingkungan

```
import os

from collections import defaultdict

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import matplotlib as mpl

import seaborn as sns

import shap

import warnings

from sklearn.ensemble import RandomForestRegressor

from sklearn.model_selection import (train_test_split,
RandomizedSearchCV,

cross_val_score, LeaveOneOut, cross_val_predict)

from sklearn.metrics import mean_squared_error, mean_absolute_error,
r2_score

from scipy.stats import randint, uniform

try:

    display # noqa: F821

except NameError:

    def display(obj):

        try: print(obj.to_string())

        except AttributeError: print(obj)

plt.style.use('seaborn-v0_8-whitegrid')

plt.rcParams['figure.dpi'] = 120

plt.rcParams['axes.titlesize'] = 14

plt.rcParams['axes.labelsize'] = 11

plt.rcParams['font.size'] = 10

plt.rcParams['figure.figsize'] = (12, 5)

warnings.filterwarnings('ignore')

OUTPUT_DIR = 'output_random_forest'

DIR_GAMBAR = os.path.join(OUTPUT_DIR, 'gambar')

DIR_TABEL = os.path.join(OUTPUT_DIR, 'tabel')

os.makedirs(DIR_GAMBAR, exist_ok=True)

os.makedirs(DIR_TABEL, exist_ok=True)

def simpan_gambar(nama_file, dpi=150):

    path = os.path.join(DIR_GAMBAR, nama_file)
```

```
os.makedirs(os.path.dirname(path), exist_ok=True)
```

```
plt.savefig(path, dpi=dpi, bbox_inches='tight')
```

```
print(f' [gambar tersimpan] {path}')
```

```
plt.show()
```

```
def simpan_tabel(df_out, nama_file, index=False):
```

```
    path = os.path.join(DIR_TABEL, nama_file)
```

```
    df_out.to_csv(path, index=index)
```

```
    print(f' [tabel tersimpan] {path}')
```

```
print(f'Folder output siap : {OUTPUT_DIR}/ (gambar/ & tabel/)')
```

```
# -----
```

```
# KAMUS MORFOLOGI TUMBLER
```

```
# -----
```

```
FEATURE_NAMES = {
```

```
    'X1': 'Tipe Tutup', 'X2': 'Sistem Minum', 'X3': 'Sistem Pegangan',
```

```
    'X4': 'Bentuk Badan Atas', 'X5': 'Bentuk Badan Tengah', 'X6': 'Bentuk  
Badan Bawah',
```

```
    'X7': 'Bahan Material', 'X8': 'Struktur Dinding', 'X9': 'Skema Warna',
```

```
    'X10': 'Fitur', 'X11': 'Finishing', 'X12': 'Volume', 'X13': 'Gaya Desain',
```

```
}
```

```
CATEGORY_MAP = {
```

```
    'X1': {1: 'Tutup ulir polos', 2: 'Ulir + carabiner', 3: 'Ulir + spout flip top',
```

```
    4: 'Flip-up handle straw lid (Iceflow)', 5: 'Push-button straw lid',
```

```
    6: 'Quencher 3-way lid', 7: 'Flip-up straw lid + Twist/Ulir',
```

```
    'X2': {1: 'Sip opening (sip langsung)', 2: 'Sedotan eksternal (lepas)', 3:  
'Spout flip',
```

```
    4: 'Spout flip + Wide mouth langsung', 5: 'Wide mouth langsung',
```

```
    'X3': {1: 'Tanpa pegangan', 2: 'Carabiner clip', 3: 'Carry circle handle',
```

```
    4: 'Carry half circle handle', 5: 'Carry half rectangle handle',
```

```
    6: 'Carry half hexagon handle', 7: 'Big side handle (Quencher)'},
```

```
    'X4': {1: 'Lurus / silinder', 2: 'Menyempit ke leher (shouldered)', 3:  
'Membulat / flask (rounded dome)'},
```

```
    'X5': {1: 'Silinder lurus', 2: 'Tapered / tirus', 3: 'Slim', 4: 'Hydra Flask  
classic (bahu lebar)'},
```

```
    'X6': {1: 'Lurus rata', 2: 'Tirus mengecil (tapered narrow)', 3: 'Curvy /  
membulat'},
```

```
    'X7': {1: 'Stainless steel', 2: 'Stainless steel + Keramik', 3: 'Baja', 4:  
'Aluminum', 5: 'PP / Polypropilen'},
```

```
    'X8': {1: 'Double-wall', 2: 'Single-wall'},
```

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
'X9': {1: 'Monokrom solid', 2: 'Dua-nada (two-tone)', 3: 'Tiga-nada (three-tone)',
      4: 'Gradient / ombre', 5: 'Pola / character print', 6: 'Transparan'},
'X10': {1: 'Carabiner', 2: 'Strap / lanyard', 3: 'Charm / pendant',
        4: 'Silicone band tengah', 5: 'Filter', 6: 'Tanpa aksesoris'},
'X11': {1: 'Matte / doff', 2: 'Glossy', 3: 'Transparan / clear'},
'X12': {1: 'Kecil (250-500ml)', 2: 'Sedang (501-800ml)', 3: 'Besar (801-1500ml)'},
'X13': {1: 'Classic', 2: 'Minimalist', 3: 'Minimalist + Modern', 4: 'Modern',
        5: 'Modern + Premium', 6: 'Playful', 7: 'Playful + Ilustratif', 8: 'Cute'},
}

X_CODES = [f'X{i}' for i in range(1, 14)]

FEATURE_LABELS = [FEATURE_NAMES[x] for x in X_CODES]

LABEL_TO_CODE = {v: k for k, v in FEATURE_NAMES.items()}

GENERATIONS = ['GenMilenial', 'GenZ']

KANSEI = ['Premium', 'Durabilitas', 'Ergonomis']

TARGET_COLS = [f'{g}_{k}_Mean' for g in GENERATIONS for k in KANSEI]

GEN_COLOR = {'GenMilenial': '#003049', 'GenZ': '#d62828'}

GEN_PALETTE = {'GenMilenial': 'mako_r', 'GenZ': 'rocket_r'}

print(f'NumPy {np.__version__} | Pandas {pd.__version__} | Seaborn {sns.__version__}')

def faktor_induk(nama_kolom_dummy):
    """Petakan kolom one-hot ke faktor morfologi induknya (salah satu dari 13 faktor)."""
    kandidat = [fl for fl in FEATURE_LABELS
                 if nama_kolom_dummy == fl or
                 nama_kolom_dummy.startswith(fl + '_')]
    return max(kandidat, key=len) if kandidat else nama_kolom_dummy

def agregasi_ke_faktor(seri_dummy):
    """Jumlahkan nilai tiap level one-hot ke 13 faktor; kembalikan Series 'X# - Nama' terurut X1..X13."""
    induk = [faktor_induk(c) for c in seri_dummy.index]
    agg = seri_dummy.groupby(induk).sum().reindex(FEATURE_LABELS).fillna(0.0)
    agg.index = [f'{LABEL_TO_CODE[fl]} - {fl}' for fl in agg.index]
    return agg

path = 'Morfologi_Sampel_Tumbler_GenX_dan_GenZ.xlsx'
```

```
df = pd.read_excel(path, sheet_name='Sheet1')

df.columns = [c.strip() for c in df.columns]

df.columns = [c.replace('GenX', 'GenMilenial') for c in df.columns] # sinkronkan label kolom target Gen X -> Gen Milenial

print(f'Dataset Tumbler : {df.shape[0]} sampel, {df.shape[1]} kolom')

print(f'Variabel morfologi : {len(X_CODES)} (X1-X13)')

print(f'Variabel target : {len(TARGET_COLS)} (2 generasi x 3 konsep Kansei)')

df_decoded = df.copy()

for x in X_CODES:
    df_decoded[FEATURE_NAMES[x]] = df[x].map(CATEGORY_MAP[x])

for g in GENERATIONS:
    df_decoded[f'{g}_Preferensi'] = df_decoded[[f'{g}_{k}_Mean' for k in KANSEI]].mean(axis=1)

print('Contoh hasil decode (Sampel 1-3):')
display(df_decoded[['Sampel_Code'] + FEATURE_LABELS].head(3))

simpan_tabel(df_decoded[['Sampel_Code'] + FEATURE_LABELS],
             'morfologi_terdecode.csv')

display(df[['Sampel_Code'] + X_CODES].head())
display(df[['Sampel_Code'] + TARGET_COLS].head())

desc_target = df[TARGET_COLS].describe()
display(desc_target)
simpan_tabel(desc_target, 'deskriptif_skor_Kansei.csv', index=True)

fig, axes = plt.subplots(2, 3, figsize=(16, 9))
for ax, g in zip(axes[0], axes[1]), GENERATIONS:
    for j, k in enumerate(KANSEI):
        sns.histplot(df[f'{g}_{k}_Mean'], kde=True, color=GEN_COLOR[g],
                     bins=12, ax=ax[j])
        ax[j].set_title(f'{g} - {k}', fontweight='bold')
        ax[j].set_xlabel('Skor (skala 1-7)')

plt.suptitle('Distribusi Skor Kansei per Generasi & Konsep',
             fontweight='bold', fontsize=15)

plt.tight_layout()

simpan_gambar('01_distribusi_skor_Kansei.png')

melted = df[TARGET_COLS].melt(var_name='Target', value_name='Skor')

melted['Generasi'] = melted['Target'].str.split('_').str[0]

melted['Kansei'] = melted['Target'].str.split('_').str[1]

plt.figure(figsize=(12, 6))
```



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
sns.boxplot(data=melted, x='Kansei', y='Skor', hue='Generasi',
            palette=[GEN_COLOR['GenMilenial'], GEN_COLOR['GenZ']])

plt.title('Perbandingan Skor Kansei: Gen Milenial vs Gen Z',
          fontweight='bold')

plt.ylabel('Skor (skala 1-7)')

plt.tight_layout()

simpan_gambar('02_boxplot_Kansei_milenial_vs_genz.png')

def eda_satu_generasi(g):
    """EDA morfologi khusus satu generasi: Frekuensi + Proporsi + Pengaruh
    ke preferensi generasi tsb."""

    pref_col = f'{g}_Preferensi'

    pal = GEN_PALETTE[g]

    tabel_g = []

    print('#' * 80)

    print(f' EDA MORFOLOGI — {g}')

    print('#' * 80)

    for x in X_CODES:
        fl = FEATURE_NAMES[x]

        counts = df_decoded[fl].value_counts()

        order = counts.index

        total = int(counts.sum())

        fig, axes = plt.subplots(1, 3, figsize=(21, 5))

        sns.countplot(data=df_decoded, x=fl, order=order, palette=pal,
                      edgecolor='0.2', ax=axes[0])

        axes[0].set_title(f'Frekuensi: {fl} ({g})', fontweight='bold')

        axes[0].set_xlabel(""); axes[0].set_ylabel('Jumlah')

        axes[0].tick_params(axis='x', rotation=40)

        for lbl in axes[0].get_xticklabels(): lbl.set_ha('right')

        axes[1].pie(counts, labels=counts.index, autopct='%1.1f%%',
                   colors=sns.color_palette(pal, len(counts)), startangle=140,
                   textprops={'fontsize': 8})

        axes[1].set_title(f'Proporsi: {fl} ({g})', fontweight='bold')

        sns.boxplot(data=df_decoded, x=fl, y=pref_col, order=order,
                    palette=pal, ax=axes[2])

        axes[2].set_title(f'{fl} vs Preferensi ({g})', fontweight='bold')

        axes[2].set_xlabel(""); axes[2].set_ylabel(f'Skor preferensi {g}')

        axes[2].tick_params(axis='x', rotation=40)

        for lbl in axes[2].get_xticklabels(): lbl.set_ha('right')

        plt.tight_layout()

        simpan_gambar(f'eda_{g.lower()}_eda_{x}_fl.replace(" ",
        "_")._g.png')
```

```
prefm = df_decoded.groupby(fl)[pref_col].mean().reindex(order)

tab = pd.DataFrame({
    'Kode': x, 'Faktor': fl, 'Generasi': g, 'Kategori': list(order),
    'Frekuensi': counts.reindex(order).values.astype(int),
    'Proporsi_%': (counts.reindex(order).values / total * 100).round(1),
    f'Skor_Preferensi_{g}': prefm.values.round(3),
})

print(f' {x} - {fl} (n={total})')

display(tab[['Kategori', 'Frekuensi', 'Proporsi_%',
             f'Skor_Preferensi_{g}']])

print(f' Dominan: "{order[0]}" ({int(counts.iloc[0])),
      {counts.iloc[0]/total*100:.1f}% ) | '

      f'Preferensi {g} tertinggi: "{prefm.idxmax()}" ({prefm.max():.2f}); '

      f'terendah: "{prefm.idxmin()}" ({prefm.min():.2f})\n')

tabel_g.append(tab.rename(columns={f'Skor_Preferensi_{g}':
'Skor_Preferensi'}))

out = pd.concat(tabel_g, ignore_index=True)

simpan_tabel(out, f'eda_frekuensi_proporsi_{g}.csv')

return out

eda_milenial = eda_satu_generasi('GenMilenial')

eda_genz = eda_satu_generasi('GenZ')

plt.figure(figsize=(11, 9))

corr_src = df[X_CODES].copy()

corr_src.columns = [f'{x} - {FEATURE_NAMES[x]}' for x in X_CODES]

corr_faktor = corr_src.corr()

sns.heatmap(corr_faktor, cmap='mako', annot=True, fmt='.2f',
            square=True,
            linewidths=0.4, cbar_kws={'label': 'Pearson r'})

plt.title('Heatmap Korelasi Antar-13 Faktor Morfologi (X1-X13)',
          fontweight='bold', fontsize=13)

plt.tight_layout()

simpan_gambar('03_Heatmap_korelasi_13_faktor.png')

simpan_tabel(corr_faktor.round(3), 'korelasi_antar_faktor_morfologi.csv',
            index=True)

plt.figure(figsize=(12, 8))

corr_xy = df[X_CODES + TARGET_COLS].corr().loc[X_CODES,
TARGET_COLS]

corr_xy.index = [f'{x} - {FEATURE_NAMES[x]}' for x in X_CODES]

sns.heatmap(corr_xy, cmap='coolwarm', center=0, annot=True, fmt='.2f',
            linewidths=0.5, cbar_kws={'label': 'Pearson r'})
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
plt.title('Korelasi 13 Faktor Morfologi vs Skor Kansei', fontweight='bold')

plt.tight_layout()

simpan_gambar('04_korelasi_morfologi_vs_Kansei.png')

simpan_tabel(corr_xy.round(3), 'korelasi_morfologi_vs_Kansei.csv',
index=True)

X_all = pd.get_dummies(df_decoded[FEATURE_LABELS])

print(f'Dimensi matriks fitur (One-Hot): {X_all.shape}')

param_dist = {

    'n_estimators': [100, 200, 300, 500],

    'max_depth': [None, 5, 10, 15, 20],

    'min_samples_split': [2, 5, 10],

    'min_samples_leaf': [1, 2, 4],

}

results = {}

def build_rf_model(X, y, label, color, slug, show_shap=True):

    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

    RandomForestRegressor(n_estimators=100,
random_state=42).fit(X_train, y_train)

    rf_tuned =
RandomizedSearchCV(RandomForestRegressor(random_state=42),

        param_distributions=param_dist, n_iter=10, cv=3,

        random_state=42, scoring='r2', n_jobs=-1)

    rf_tuned.fit(X_train, y_train)

    best_rf = rf_tuned.best_estimator_

    pred = best_rf.predict(X_test); pred_train = best_rf.predict(X_train)

    mse = mean_squared_error(y_test, pred); rmse = np.sqrt(mse)

    mae = mean_absolute_error(y_test, pred)

    r2 = r2_Score(y_test, pred); r2_train = r2_Score(y_train, pred_train); gap
= abs(r2_train - r2) * 100

    print('=' * 56); print(f' MODEL: {label}'); print('=' * 56)

    print(f'Best Params : {rf_tuned.best_params_}')

    print(f'MSE : {mse:.4f} RMSE : {rmse:.4f} MAE : {mae:.4f} R2 :
{r2:.4f}')

    print(f'R2 train={r2_train:.4f} | test={r2:.4f} | gap={gap:.2f}%')

    f -> ["Stabil" if gap < 10 else "Perlu diwaspadai"]

    fi_faktor = agregasi_ke_faktor(pd.Series(best_rf.feature_importances_,
index=X.columns)).sort_values()

    plt.figure(figsize=(10, 7))
```

```
fi_faktor.plot(kind='barh', color=color, edgecolor='black', alpha=0.85)

    plt.title(f'Feature Importance per 13 Faktor Morfologi - {label}',
fontweight='bold')

    plt.xlabel('Importance Score (agregat per faktor)')

    plt.tight_layout();
simpan_gambar(f'split/{slug}_feature_importance.png')

residuals = y_test.values - pred

fig, ax = plt.subplots(1, 2, figsize=(15, 5))

ax[0].scatter(y_test, pred, color=color, alpha=0.7, edgecolors='black')

mn, mx = min(y_test.min(), pred.min()), max(y_test.max(), pred.max())

ax[0].plot([mn, mx], [mn, mx], 'r--', label='Perfect prediction')

ax[0].set_title(f'Actual vs Predicted - {label}', fontweight='bold')

ax[0].set_xlabel('Actual'); ax[0].set_ylabel('Predicted'); ax[0].legend()

ax[1].scatter(pred, residuals, color=color, alpha=0.7, edgecolors='black')

ax[1].axhline(0, color='red', linestyle='--')

ax[1].set_title(f'Residual Plot - {label}', fontweight='bold')

ax[1].set_xlabel('Predicted'); ax[1].set_ylabel('Residual')

plt.tight_layout(); simpan_gambar(f'split/{slug}_diagnostik.png')

if show_shap:

    try:

        sv = shap.TreeExplainer(best_rf).shap_values(X_test)

        shap_faktor =
agregasi_ke_faktor(pd.Series(np.abs(sv).mean(axis=0),
index=X_test.columns)).sort_values()

        plt.figure(figsize=(10, 7))

        shap_faktor.plot(kind='barh', color=color, edgecolor='black',
alpha=0.85)

        plt.title(f'SHAP Feature Contribution per 13 Faktor - {label}',
fontweight='bold')

        plt.xlabel('mean(|SHAP value|) - agregat per faktor')

        plt.tight_layout(); simpan_gambar(f'split/{slug}_shap.png')

    except Exception as e:

        print(f'SHAP dilewati: {e}')

    return {'model': best_rf, 'mse': mse, 'rmse': rmse, 'mae': mae, 'r2': r2,
'gap': gap,

        'fi_faktor': fi_faktor, 'best_params': rf_tuned.best_params_}

for g in GENERATIONS:

    for k in KANSEI:

        col = f'{g}_{k}_Mean'

        results[col] = build_rf_model(X_all, df[col], f'{g} - {k}', GEN_COLOR[g],
slug=col)
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
results_loocv = {}
```

```
def build_rf_model_loocv(X, y, label, color, slug, nested=False,
show_shap=True):
```

```
    loo = LeaveOneOut()

    if nested:

        oof_pred = np.zeros(len(y)); yv = y.values

        for tr, te in loo.split(X):

            s =
RandomizedSearchCV(RandomForestRegressor(random_state=42),
param_distributions=param_dist,

                        n_iter=10, cv=3, random_state=42, scoring='r2',
n_jobs=-1)

            s.fit(X.iloc[tr], yv[tr]); oof_pred[te] =
s.best_estimator_.predict(X.iloc[te])

            best_params = '(nested: berbeda per fold)'

            final_model = RandomForestRegressor(random_state=42).fit(X, y)

        else:
            s =
RandomizedSearchCV(RandomForestRegressor(random_state=42),
param_distributions=param_dist,

                        n_iter=10, cv=3, random_state=42, scoring='r2',
n_jobs=-1)

            s.fit(X, y); best_params = s.best_params_

            model = RandomForestRegressor(random_state=42, **best_params)

            oof_pred = cross_val_predict(model, X, y, cv=loo, n_jobs=-1)

            final_model = model.fit(X, y)

    mse = mean_squared_error(y, oof_pred); rmse = np.sqrt(mse)

    mae = mean_absolute_error(y, oof_pred); r2 = r2_Score(y, oof_pred)

    print('=' * 56); print(' MODEL (LOOCV): {label}'); print('=' * 56)

    print(f'Best Params : {best_params}')

    print(f'MSE : {mse:.4f} RMSE : {rmse:.4f} MAE : {mae:.4f} R2 (OOF) :
{r2:.4f}')

    fi_faktor =
agregasi_ke_faktor(pd.Series(final_model.feature_importances_,
index=X.columns)).sort_values()

    plt.figure(figsize=(10, 7))

    fi_faktor.plot(kind='barh', color=color, edgecolor='black', alpha=0.85)

    plt.title(f'Feature Importance LOOCV per 13 Faktor - {label}',
fontWeight='bold')

    plt.xlabel('Importance Score (agregat per faktor)')

    plt.tight_layout();
simpan_gambar(f'loocv/{slug}_feature_importance.png')

    residuals = y.values - oof_pred

    fig, ax = plt.subplots(1, 2, figsize=(15, 5))
```

```
ax[0].scatter(y, oof_pred, color=color, alpha=0.7, edgecolors='black')
```

```
mn, mx = min(y.min(), oof_pred.min()), max(y.max(), oof_pred.max())
```

```
ax[0].plot([mn, mx], [mn, mx], 'r--', label='Perfect prediction')
```

```
ax[0].set_title(f'Actual vs OOF-Predicted - {label}', fontWeight='bold')
```

```
ax[0].set_xlabel('Actual'); ax[0].set_ylabel('OOF Predicted');
ax[0].legend()
```

```
ax[1].scatter(oof_pred, residuals, color=color, alpha=0.7,
edgecolors='black')
```

```
ax[1].axhline(0, color='red', linestyle='--')
```

```
ax[1].set_title(f'Residual Plot (LOOCV) - {label}', fontWeight='bold')
```

```
ax[1].set_xlabel('OOF Predicted'); ax[1].set_ylabel('Residual')
```

```
plt.tight_layout(); simpan_gambar(f'loocv/{slug}_diagnostik.png')
```

```
if show_shap:
```

```
    try:
```

```
        sv = shap.TreeExplainer(final_model).shap_values(X)
```

```
        shap_faktor =
agregasi_ke_faktor(pd.Series(np.abs(sv).mean(axis=0),
index=X.columns)).sort_values()
```

```
        plt.figure(figsize=(10, 7))
```

```
        shap_faktor.plot(kind='barh', color=color, edgecolor='black',
alpha=0.85)
```

```
        plt.title(f'SHAP Feature Contribution per 13 Faktor (LOOCV) -
{label}', fontWeight='bold')
```

```
        plt.xlabel('mean(|SHAP value|) - agregat per faktor')
```

```
        plt.tight_layout(); simpan_gambar(f'loocv/{slug}_shap.png')
```

```
    except Exception as e:
```

```
        print(f'SHAP dilewati: {e}')
```

```
    return {'model': final_model, 'mse': mse, 'rmse': rmse, 'mae': mae, 'r2':
r2,
```

```
        'fi_faktor': fi_faktor, 'best_params': best_params}
```

```
USE_NESTED_LOOCV = False
```

```
for g in GENERATIONS:
```

```
    for k in KANSEI:
```

```
        col = f'{g}_{k}_Mean'
```

```
        results_loocv[col] = build_rf_model_loocv(
```

```
            X_all, df[col], f'{g} - {k}', GEN_COLOR[g], slug=col,
nested=USE_NESTED_LOOCV)
```

```
rows_eval = []
```

```
for col in results:
```

```
    g, k, _ = col.split('_'); rs, rl = results[col], results_loocv[col]
```

```
    rows_eval.append((f'Generasi: {g}, 'KANSEI': {k},
```

```
        'R2_split': round(rs['r2'], 4), 'R2_LOOCV': round(rl['r2'], 4),
```



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
'MSE_split': round(rs['mse'], 4), 'MSE_LOOCV':
round(rl['mse'], 4),

'RMSE_split': round(rs['rmse'], 4), 'RMSE_LOOCV':
round(rl['rmse'], 4),

'MAE_split': round(rs['mae'], 4), 'MAE_LOOCV':
round(rl['mae'], 4)))

eval_compare = pd.DataFrame(rows_eval)

display(eval_compare); simpan_tabel(eval_compare,
'perbandingan_split_vs_loocv.csv')

plt.figure(figsize=(13, 6)); idx = np.arange(len(eval_compare)); w = 0.4

lbs = eval_compare['Generasi'] + '\n' + eval_compare['Kansei']

plt.bar(idx - w/2, eval_compare['R2_split'], width=w, label='Single-Split
(n_test=8)', color='#8d99ae', edgecolor='black')

plt.bar(idx + w/2, eval_compare['R2_LOOCV'], width=w, label='LOOCV
(n=40 OOF)', color='#2a9d8f', edgecolor='black')

plt.axhline(0, color='red', linestyle='--', linewidth=1)

plt.xticks(idx, lbs); plt.ylabel('R2 Score')

plt.title('Perbandingan R2: Single-Split vs LOOCV', fontweight='bold');
plt.legend()

plt.tight_layout();
simpan_gambar('05_perbandingan_r2_split_vs_loocv.png')

comp = pd.DataFrame({'Generasi': c.split('_')[0], 'Kansei': c.split('_')[1],

'MSE': rl['mse'], 'RMSE': rl['rmse'], 'MAE': rl['mae'], 'R2':
rl['r2']})

for c, rl in results_loocv.items()))

display(comp.round(4)); simpan_tabel(comp.round(4),
'ringkasan_metrik_rf_loocv.csv')

fig, axes = plt.subplots(1, 3, figsize=(21, 6))

lbs = comp['Generasi'] + '\n' + comp['Kansei']

for i, metric in enumerate(['MSE', 'RMSE', 'MAE']):

    axes[i].bar(lbs, comp[metric], color='#003049', edgecolor='black',
alpha=0.85)

    axes[i].set_title(f'{metric} per Target - RF LOOCV (turun = lebih baik)',
fontweight='bold')

    axes[i].set_ylabel(metric)

plt.tight_layout();
simpan_gambar('06_rf_mse_rmse_mae_per_target.png')

records = []

for col, res in results_loocv.items():

    g, k, _ = col.split('_')

    top3 = res['fi_faktor'].sort_values(ascending=False).head(3)

    print(f'\n{g} - {k} (R2 LOOCV={res["r2"]:.3f}, RMSE={res["rmse"]:.3f})')

    for rank, (feat, imp) in enumerate(top3.items(), 1):

        print(f' {rank}. {feat:<35} {imp:.4f}')

    records.append({'Generasi': g, 'Kansei': k, 'Rank': rank, 'Faktor': feat,
```

```
'Importance': round(float(imp), 4)))

simpan_tabel(pd.DataFrame(records), 'faktor_desain_dominan.csv')

faktor_to_dummies = defaultdict(list)

for c in X_all.columns:

    faktor_to_dummies[faktor_induk(c)].append(c)

faktor_to_dummies = {fl: faktor_to_dummies[fl] for fl in
FEATURE_LABELS}

def nama_elemen(dummy_col, fl):

    return dummy_col[len(fl) + 1:] if dummy_col.startswith(fl + '_') else
dummy_col

def pilih_elemen_optimal(model, X):

    """Partial dependence: pilih elemen ber-skor prediksi tertinggi untuk
tiap faktor.

    Mengembalikan (tabel elemen terpilih, skor konfigurasi optimal
gabungan)."""

    cols = list(X.columns)

    base = X.values.astype(float)

    col_idx = {c: i for i, c in enumerate(cols)}

    baris = []

    x_star = np.zeros((1, X.shape[1]))

    for fl, dummies in faktor_to_dummies.items():

        idxs = [col_idx[d] for d in dummies]

        best_d, best_s = None, -1e18

        for d in dummies:

            Xm = base.copy(); Xm[:, idxs] = 0.0; Xm[:, col_idx[d]] = 1.0

            s = float(model.predict(pd.DataFrame(Xm, columns=cols)).mean())

            if s > best_s:

                best_s, best_d = s, d

        baris.append({'Kode': LABEL_TO_CODE[fl], 'Faktor': fl,

'Elemen_Terpilih': nama_elemen(best_d, fl), 'Skor_PD':
round(best_s, 3)})

        x_star[0, col_idx[best_d]] = 1.0

    skor_optimal = float(model.predict(pd.DataFrame(x_star,
columns=cols))[0])

    return pd.DataFrame(baris), skor_optimal

elemen_records, ringkasan_opt = [], []

for g in GENERATIONS:

    for k in KANSEI:

        col = f'{g}_{k}_Mean'
```



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
tab, skor_opt = pilih_elemen_optimal(results_loocv[col]['model'],
X_all)
```

```
tab.insert(0, 'Konsep', k); tab.insert(0, 'Generasi', g)
```

```
print('=' * 72)
```

```
print(f' ELEMEN DESAIN TERPILIH — {g} x {k} '
```

```
f'(skor optimal prediksi={skor_opt:.3f}; rerata
aktual={df[col].mean():.3f}))'
```

```
print('=' * 72)
```

```
display(tab[['Kode', 'Faktor', 'Elemen_Terpilih', 'Skor_PD']])
```

```
elemen_records.append(tab)
```

```
ringkasan_opt.append({'Generasi': g, 'Konsep': k,
```

```
'Skor_Optimal_Prediksi': round(skor_opt, 3),
```

```
'Rerata_Aktual': round(float(df[col].mean()), 3)})
```

```
elemen_all = pd.concat(elemen_records, ignore_index=True)
```

```
simpan_tabel(elemen_all, 'elemen_desain_terpilih_per_target.csv')
```

```
ringkasan_df = pd.DataFrame(ringkasan_opt)
```

```
print('\nRINGKASAN SKOR KONFIGURASI OPTIMAL (prediksi RF) vs RERATA
AKTUAL:')
```

```
display(ringkasan_df)
```

```
simpan_tabel(ringkasan_df, 'skor_konfigurasi_optimal.csv')
```

```
urutan_faktor = [f'{x} - {FEATURE_NAMES[x]}' for x in X_CODES]
```

```
for g in GENERATIONS:
```

```
sub = elemen_all[elemen_all['Generasi'] == g].copy()
```

```
sub['FaktorLb'] = sub['Kode'] + ' - ' + sub['Faktor']
```

```
piv = (sub.pivot_table(index='FaktorLb', columns='Konsep',
```

```
values='Elemen_Terpilih', aggfunc='first')
```

```
.reindex(urutan_faktor)[KANSEI])
```

```
print('=' * 90)
```

```
print(f' ELEMEN DESAIN TERPILIH — {g} (13 Faktor Morfologi x 3
Konsep Kasei)')
```

```
print('=' * 90)
```

```
display(piv)
```

```
simpan_tabel(piv, f'elemen_desain_terpilih_{g}.csv', index=True)
```

```
plt.figure(figsize=(13, 6))
```

```
idx = np.arange(len(ringkasan_df)); w = 0.4
```

```
lbs = ringkasan_df['Generasi'] + '\n' + ringkasan_df['Konsep']
```

```
plt.bar(idx - w/2, ringkasan_df['Rerata_Aktual'], width=w, label='Rerata
Aktual',
```

```
color='#8d99ae', edgecolor='black')
```

```
plt.bar(idx + w/2, ringkasan_df['Skor_Optimal_Prediksi'], width=w,
label='Skor Optimal (prediksi RF)',
```

```
color='#2a9d8f', edgecolor='black')
```

```
plt.xticks(idx, lbs); plt.ylabel('Skor Kasei (skala 1-7)')
```

```
plt.title('Skor Konfigurasi Desain Optimal vs Rerata Aktual per Target',
fontweight='bold')
```

```
plt.legend(); plt.tight_layout()
```

```
simpan_gambar('07_skor_konfigurasi_optimal.png')
```

POLITEKNIK
NEGERI
JAKARTA



Hak Cipta :

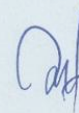
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Lampiran 14. *Logbook* Kegiatan Bimbingan Materi

LOGBOOK

KEGIATAN BIMBINGAN SKRIPSI

Nama : Abdillah Nur Fatah
 NIM : 2206411069
 Judul Penelitian : PERANCANGAN REUSABLE PACKAGING TUMBLER
 BERBASIS KANSEI ENGINEERING BERDASARKAN PRESEPSI
 LINTAS GENERASI
 Nama Pembimbing : Novi Purnama Sari, S.T.P., M.Si.

TANGGAL	CATATAN PEMBIMBING	PARAF
5 Maret 2026	Penggantian penentuan objek penelitian skripsi dan timeline skripsi	
11 Maret 2026	Asistensi sampel tumbler dan proposal	
17 April 2026	Penulisan Bab I, Bab II, Bab III	
30 April 2026	Revisi penulisan Bab I, Bab II, dan Bab III	



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

5 Mei 2026	Asistensi kata Kansei	
21 Mei 2026	Revisi penulisan Bab I, Bab II, Bab III, dan Bab IV	
5 Juni 2026	Revisi hasil konsep desain	
19 Juni 2026	Asistensi morfologi sampel, hasil Semantic Differential, dan hasil konsep desain	
29 Juni 2026	Asistensi hasil elemen desain	
1 Juli 2026	Asistensi hasil perancangan desain tumbler	

DAFTAR RIWAYAT HIDUP



Penulis bernama Abdillah Nur Fatah adalah anak pertama dari pasangan Kukuh Nofriyanto dan Lutfi Lestari. Lahir di Jakarta, 9 Juni 2005. Penulis menyelesaikan pendidikannya di SDN Ciracas 11 Pagi lulus pada tahun 2016. Kemudian, melanjutkan pendidikan di SMPN 9 Jakarta lulus pada tahun 2019. Dan menempuh pendidikan menengah atas di SMAN 58 Jakarta dengan jurusan Matematika Ilmu Pengetahuan Alam (MIPA) lulus tahun 2022. Pada tahun yang sama, penulis diterima di Politeknik Negeri Jakarta (PNJ), Jurusan Teknik Grafika dan Penerbitan, Program Studi Teknologi Industri Cetak Kemasan melalui jalur Ujian Mandiri.

Selama masa kuliah, penulis aktif dalam mengikuti berbagai kegiatan, seperti mengikuti panitia event *Craft 2024* di bidang Logistik. Kemudian, penulis juga mengikuti beberapa perlombaan desain kemasan seperti *Packindo Star 2024* dan *Inpack 2025*. Selain itu, penulis juga berpartisipasi dalam program pengabdian UMKM dan penelitian dosen dalam bidang *Kansei Engineering* yang berfokus pada pengembangan kemasan keripik tempe dan dodol belimbing berbasis persepsi emosional konsumen. Penulis juga memiliki pengalaman magang di PT. *Garudafood Putra Putri Jaya, Tbk* pada tahun 2025 hingga 2026 sebagai *Packaging Designer*.

© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

