



© 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



**SIMULASI COVERAGE SINYAL WIRELESS LOCAL AREA
NETWORK (WLAN) BERBASIS MACHINE LEARNING
MENGUNAKAN VISUALISASI WEB**

TUGAS AKHIR

Diajukan sebagai salah satu syarat untuk memperoleh gelar

Diploma Tiga

**POLITEKNIK
NEGERI
JAKARTA**

RAJA DANTON'S DAMANIK

2303332077

PROGRAM STUDI TEKNIK TELEKOMUNIKASI

JURUSAN TEKNIK ELEKTRO

POLITEKNIK NEGERI JAKARTA

2026



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

HALAMAN PERNYATAAN ORISINALITAS

Tugas akhir ini adalah hasil karya saya sendiri dan semua sumber baik yang dikutip maupun dirujuk telah saya nyatakan dengan benar.

Nama : Raja Danton's Damanik

NIM : 2303332077

Tanda Tangan :

Tanggal : 27 Juli 2026



**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

**HALAMAN PENGESAHAN
TUGAS AKHIR**

Tugas Akhir diajukan oleh :

Nama : Raja Danton's Damanik
NIM : 2303332077
Program Studi : Teknik Telekomunikasi
Judul Tugas Akhir : Simulasi *Coverage* sinyal *Wireless Local Area Network* (WLAN) Berbasis *Machine Learning* Menggunakan Visualisasi Web

Telah Diuji oleh tim penguji dalam Sidang Tugas Akhir pada hari Selasa tanggal 7 Juli 2026 dan dinyatakan LULUS

Pembimbing : Dr. Yenniwarti Rafsyam, SST., MT.
NIP. 192680627 199303 2 002

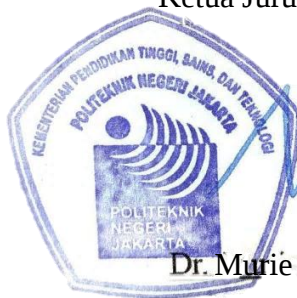
()

**POLITEKNIK
NEGERI
JAKARTA**

Depok, 30 Juli 2026

Disahkan Oleh

Ketua Jurusan Teknik Elektro




Dr. Murie Dwiyaniti, S.T., M.T.

NIP. 1978033 120031 2 200



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

KATA PENGANTAR

Puji syukur saya panjatkan kepada Tuhan Yang Maha Esa, karena atas berkat dan rahmat-Nya, penulis dapat menyelesaikan Tugas Akhir ini. Penulisan Tugas Akhir ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Diploma Tiga.

Penulis menyadari bahwa, tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan tugas akhir ini, sangatlah sulit bagi penulis untuk menyelesaikan tugas akhir ini. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Dr. Yenniwarti Rafsyam, SST., MT., selaku dosen pembimbing yang telah menyediakan waktu, tenaga, dan pikiran untuk mengarahkan penulis dalam penyusunan tugas akhir ini;
2. Orang tua dan keluarga penulis yang telah memberikan bantuan dukungan material dan moral.
3. Teman-teman yang telah banyak membantu penulis dalam menyelesaikan tugas laporan ini.

Akhir kata, penulis berharap Tuhan Yang Maha Esa berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga Tugas Akhir ini membawa manfaat bagi pengembangan ilmu.

Depok, Senin 27 Juli 2026

Penulis



Simulasi Coverage sinyal Wireless Local Area Network (WLAN) Berbasis Machine Learning Menggunakan Visualisasi Web

ABSTRAK

Perencanaan jaringan Wireless Local Area Network (WLAN) di dalam gedung memerlukan pemahaman mengenai pola penyebaran sinyal yang dipengaruhi oleh berbagai faktor fisik seperti jarak, jumlah, material, dan tipe tembok penghalang. Pendekatan konvensional berbasis rumus propagasi Log-Distance Path Loss hanya mempertimbangkan jarak dan redaman tembok dengan asumsi material seragam, sehingga tidak mampu mencerminkan variasi kondisi lapangan secara akurat. Penelitian ini mengembangkan aplikasi SimNet, sebuah simulasi coverage sinyal WLAN berbasis web yang mengintegrasikan model propagasi Log-Distance dengan algoritma Machine Learning Random Forest untuk mengklasifikasikan kualitas sinyal di setiap titik pada denah bangunan. Aplikasi dibangun menggunakan Python dan framework Dash, dengan empat fitur input model yang seluruhnya berasal dari kondisi fisik terukur: jarak (distance), jumlah tembok (wall_count), material tembok (wall_material), dan tipe tembok (wall_full). Model Random Forest dilatih menggunakan 2.800 data pengukuran lapangan dengan distribusi kelas yang seimbang antara kelas Baik, Sedang, dan Buruk. Hasil pengujian menunjukkan bahwa model classifier mencapai akurasi 79,29% dengan F1-Score macro 79,48% pada data uji, serta akurasi Cross-Validation 5-fold sebesar $79,54\% \pm 1,56\%$, sedangkan model regressor mencapai MAE 3,64 dBm dan R^2 sebesar 0,89. Kedekatan antara nilai OOB (77,81%), test accuracy (79,29%), dan CV accuracy (79,54%) mengkonfirmasi bahwa model tidak mengalami overfitting. Aplikasi SimNet menyediakan visualisasi heatmap coverage berbasis prediksi Machine Learning serta fitur perbandingan hasil klasifikasi antara pendekatan Random Forest dan threshold Teori Log-Distance secara langsung di dalam antarmuka web, sehingga dapat digunakan sebagai alat bantu perencanaan penempatan Access Point sebelum implementasi fisik.

Kata kunci: WLAN, coverage sinyal, Random Forest, Log-Distance Path Loss, visualisasi web, Machine Learning

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



MACHINE LEARNING BASED WIRELESS LOCAL AREA NETWORK (WLAN) SIGNAL COVERAGE SIMULATION USING WEB VISUALIZATION

ABSTRAK

Planning a Wireless Local Area Network (WLAN) in an indoor environment requires an understanding of signal propagation patterns influenced by various physical factors, including distance, the number, material, and type of wall obstacles. The conventional approach based on the Log-Distance Path Loss propagation formula only considers distance and wall attenuation under the assumption of uniform material, making it unable to accurately reflect variations in real field conditions. This study develops the SimNet application, a web-based WLAN signal coverage simulation that integrates the Log-Distance propagation model with a Random Forest Machine Learning algorithm to classify signal quality at each point on a building floor plan. The application is built using Python and the Dash framework, with four model input features entirely derived from measurable physical conditions: distance, wall count, wall material, and wall type. The Random Forest model was trained using 2,800 field measurement data points with a balanced class distribution among the Good, Fair, and Poor categories. Evaluation results show that the classifier model achieved an accuracy of 79.29% with a macro F1-Score of 79.48% on the test data, and a Stratified 5-Fold Cross-Validation accuracy of $79.54\% \pm 1.56\%$, while the regressor model achieved a MAE of 3.64 dBm and an R^2 of 0.89. The proximity between the OOB score (77.81%), test accuracy (79.29%), and CV accuracy (79.54%) confirms that the model does not experience overfitting. The SimNet application provides Machine Learning-based coverage heatmap visualization and a direct comparison feature between Random Forest classification and Log-Distance Theory threshold results within the web interface, making it suitable as a tool to assist Access Point placement planning prior to physical implementation.

Keywords: WLAN, signal coverage, Random Forest, Log-Distance Path Loss, web visualization, Machine Learning

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



HALAMAN SAMPUL.....	i
HALAMAN JUDUL	ii
HALAMAN PERNYATAAN ORISINALITAS	iii
HALAMAN PENGESAHAN.....	iv
KATA PENGANTAR.....	v
ABSTRAK	vi
ABSTRAK.....	vii
DAFTAR ISI.....	viii
DAFTAR TABEL	x
DAFTAR GAMBAR.....	xi
DAFTAR LAMPIRAN	xii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.1 Perumusan Masalah.....	2
1.2 Tujuan.....	2
1.3 Luaran.....	3
BAB II TINJAUAN PUSTAKA.....	4
2.1 Wireless Local Area Network (WLAN).....	4
2.2 Propagasi log-distance	4
2.3 Kualitas Jaringan	6
2.4 Wifi 2.4GHz	9
2.5 Firebase.....	10
2.6 Machine Learning.....	11
2.6.1 Random Forest	11
2.6.2 xMetrik Evaluasi Model Machine Learning	12
2.7 Visualisasi Web	13
2.8 Python.....	14
2.8.1 NumPy.....	14
2.8.2 pandas	15
2.8.3 Scikit-learn	15
2.8.4 OpenCV	15
2.8.5 Plotly dan Dash	16
2.8.6 Joblib	16
BAB III PERANCANGAN DAN REALISASI	18
3.1 Perancangan Aplikasi	18
3.1.1 Deskripsi Aplikasi	18
3.1.2 Cara kerja Aplikasi	19
3.1.3 Spesifikasi Aplikasi	21
3.1.4 Diagram Blok	21

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

3.2	Perancangan <i>Machine Learning</i> Algoritma <i>Random Forest</i>	22
3.3	Realisasi Aplikasi	33
3.3.1	Realisasi Aplikasi <i>Website</i>	33
BAB IV PEMBAHASAN.....		57
4.1	Hasil Pembuatan <i>Datasets</i>	57
4.1.1	Deskripsi Pembuatan <i>Datasets</i>	57
4.1.2	Tujuan Pembuatan <i>Datasets</i>	58
4.1.3	Prosedur Pembuatan <i>Datasets</i>	58
4.1.4	Hasil Pembuatan Model <i>Datasets</i>	59
4.2	Hasil <i>Training</i> Model <i>Random Forest</i>	60
4.2.1	Deskripsi Pengujian <i>Training Random Forest</i>	60
4.2.2	Tujuan <i>Training</i> Model	61
4.2.3	Prosedur Pengujian Hasil <i>Training</i> Algoritma <i>Random Forest</i>	61
4.2.4	Data Hasil Pengujian <i>Training</i> Algoritma <i>Random Forest</i>	62
4.3	Pengujian Aplikasi Teori dan <i>Machine Learning</i>	64
4.3.1.	Deskripsi Pada Teori dan <i>Machine Learning</i>	64
4.3.2.	Tujuan Pengujian Teori dan Model <i>machine Learning</i>	65
4.3.3.	Prosedur Pada ML dan Teori.....	65
4.3.4.	Data Hasil Pada ML dan Teori.....	69
4.4	Analisa	70
BAB V PENUTUP.....		73
4.1.	Kesimpulan.....	73
4.2.	Saran	74
DAFTAR PUSTAKA.....		76
DAFTAR RIWAYAT HIDUP		77
LAMPIRAN.....		78



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 TABEL

Tabel 2.1 Tabel n varian.....	5
Tabel 2.2 Nilai Redaman Material Penghalang pada Frekuensi 2,4 GHz	6
Tabel 2.3 Klasifikasi Kualitas Sinyal Berdasarkan RSSI	7
Tabel 2.4 Klasifikasi Kualitas Jaringan Berdasarkan SNR.....	8
Tabel 2.5 Level Interferensi pada Aplikasi SIMNET	8
Tabel 2.6 Klasifikasi Kualitas Berdasarkan Path Loss	9
Tabel 3.1 Spesifikasi Aplikasi	21
Tabel 3.2. Klasifikasi Material Tembok Berdasarkan Rentang Warna HSV	38
Tabel 3.3 Mode Kontrol Interaksi Pengguna pada Panel Kiri	40
Tabel 4. 1. Pemeraan Fitur Input Model	59
Tabel 4. 2.Cuplikan <i>Dataset</i> Hasil Pengolahan	60
Tabel 4. 3.Distribusi Label Kualitas Sinyal pada dataset hasil pengolahan	60
Tabel 4. 4.Cuplikan isi Dataset Training	61
Tabel 4.5. Ringkasan Metrik Evaluasi Hasil Training	62
Tabel 4.6. Ringkasan Metrik Evaluasi Hasil Training.....	63
Tabel 4. 7. Feature Importance Model <i>Random Forest</i>	64
Tabel 4. 8. Hasil penyebaran Algoritma RF	69
Tabel 4. 9. Data hasil penyebaran Teori	69



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 GAMBAR

Gambar 2. 1. Logo Python	14
Gambar 2. 2. Logo Numpy	14
Gambar 2. 3. Logo Pandas	15
Gambar 2. 4. Logo Scikit-learn.....	15
Gambar 2. 5. Logo OpenCV-Python	16
Gambar 2. 6. Logo Joblib.....	17
Gambar 3.1 Gambar Aplikasi SimNet	19
Gambar 3.2 Flowchart aplikasi SimNet	20
Gambar 3.3 Diagram Blok SimNet.....	22
Gambar 3.4 Tampilan Halaman Login Aplikasi SIMNET	34
Gambar 3.5 Tampilan Halaman Register Aplikasi SIMNET	35
Gambar 3.6 Hasil Deteksi Tembok Otomatis pada Denah	39
Gambar 3.7 Tampilan Antarmuka Aplikasi SimNet (Tiga Panel)	42
Gambar 3.8 Fitur Kalibrasi Skala Peta.....	44
Gambar 3.9 Heatmap RSSI pada Denah Ruangan.....	46
Gambar 3.10 Heatmap SNR.....	46
Gambar 3.11 Panel Edit Material Tembok (Kuning = Dipilih)	47
Gambar 3.12 Panel Ubah Semua Tembok Sekaligus.....	48
Gambar 3.13 Fitur Existing AP - Input RSSI Terukur.....	49
Gambar 3.14 Hasil Analisis Existing AP.....	49
Gambar 3.15 Heatmap dengan interferensi noise floor -90 dBm	50
Gambar 3.16 Heatmap dengan interferensi noise floor -80 dBm	51
Gambar 4.1. Cuplikan Data mentah Pengukuran Wifi	58
Gambar 4.2. Confusion Matitrix	63
Gambar 4. 3. Tembok otomatis dan upload denah.....	65
Gambar 4.4. Hasil Tembok Sudah Rapi.....	66
Gambar 4. 5. Mengatur Skala	67
Gambar 4.6. Titik Penempatan Users	67
Gambar 4. 7. Hasil Peletakkan Posisi AP yang Iseal Menurut Model.....	68
Gambar 4. 8. penyebaran Wifi Menggunakan Teori.....	68



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 LAMPIRAN

L- 1. Sketch Code.....	78
------------------------	----



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

Perkembangan teknologi komunikasi nirkabel telah menjadikan *Wireless Local Area Network* (WLAN) sebagai salah satu infrastruktur utama dalam mendukung aktivitas di lingkungan *indoor*. WLAN berperan penting dalam menunjang pembelajaran daring, akses sistem informasi akademik, serta komunikasi data yang menuntut konektivitas stabil dan mobilitas pengguna yang tinggi. Oleh karena itu, kualitas dan cakupan jaringan WLAN dalam *indoor* menjadi faktor krusial dalam menjamin ketersediaan layanan jaringan yang andal dan merata bagi seluruh pengguna (Slameto & Asror, 2023).

Pada lingkungan *indoor*, sinyal WLAN diharapkan mampu menjangkau seluruh ruangan dengan kualitas yang memadai. Namun pada praktiknya, sering ditemukan area tertentu yang tidak mendapatkan layanan sinyal secara optimal atau bahkan tidak terjangkau sama sekali, yang dikenal sebagai *blind spot*. Kondisi ini menyebabkan penurunan kualitas layanan jaringan dan berdampak langsung pada aktivitas pengguna. Kompleksitas struktur bangunan gedung *indoor*, seperti keberadaan tembok, sekat ruangan, serta variasi jarak terhadap *Access Point* (AP), sangat memengaruhi karakteristik propagasi sinyal WLAN. Faktor-faktor seperti redaman material bangunan, kondisi *Line of Sight* (LoS) dan *Non-Line of Sight* (NLoS), serta jarak transmisi menjadi parameter penting dalam menentukan kualitas sinyal yang diterima pengguna (Kisworo, 2025).

Evaluasi kualitas jaringan WLAN umumnya dilakukan menggunakan parameter *Received Signal Strength Indicator* (RSSI) dan *Signal Noise Ratio* (SNR), yang merepresentasikan kekuatan sinyal dan performa laju data. Nilai RSSI yang rendah dan SNR yang buruk mengindikasikan keterbatasan cakupan jaringan serta potensi keberadaan *blind spot*. Namun, metode konvensional dalam menganalisis cakupan sinyal sering kali belum mampu memodelkan kondisi propagasi sinyal *indoor* yang bersifat kompleks dan non-linear secara akurat.

Berdasarkan latar belakang tersebut, Tugas Akhir ini mengembangkan aplikasi simulasi coverage sinyal WLAN indoor berbasis web yang mengintegrasikan model propagasi Log-Distance Path Loss dan algoritma *machine*



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

learning dengan algoritma *Random Forest*. Model propagasi digunakan untuk mensimulasikan distribusi sinyal berdasarkan jarak dan hambatan pada lingkungan indoor, sedangkan algoritma *Random Forest* digunakan untuk mengklasifikasikan kualitas sinyal berdasarkan parameter yang diperoleh dari hasil simulasi dan data pengukuran lapangan. Melalui integrasi kedua metode tersebut, aplikasi diharapkan dapat membantu proses analisis, evaluasi, dan perencanaan jaringan WLAN secara lebih mudah melalui visualisasi cakupan sinyal dan informasi kualitas jaringan pada area yang diamati.

1.1 Perumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, maka perumusan masalah dalam Tugas Akhir ini adalah sebagai berikut:

1. Bagaimana menentukan kualitas cakupan sinyal jaringan WLAN indoor berdasarkan parameter RSSI, SNR, dan path loss menggunakan model propagasi log-distance ?
2. Bagaimana menerapkan algoritma *Machine Learning* untuk mengklasifikasikan kualitas cakupan sinyal jaringan WLAN indoor ke dalam kategori Baik, Sedang, dan Buruk secara adaptif ?
3. Bagaimana memodelkan dan menampilkan cakupan sinyal jaringan WLAN indoor dalam bentuk visualisasi heatmap interaktif berbasis web ?
4. Bagaimana menguji akurasi klasifikasi model *Machine Learning* dan memvalidasi hasil simulasi heatmap dengan membandingkan prediksi terhadap data pengukuran RSSI nyata di lingkungan indoor yang berbeda?

1.2 Tujuan

Tujuan yang diharapkan dari Tugas Akhir ini adalah sebagai berikut:

1. Mampu menyajikan peta cakupan sinyal jaringan WLAN indoor dalam bentuk heatmap interaktif yang menampilkan distribusi RSSI, SNR, dan path loss secara visual.
2. Mampu menerapkan algoritma *Machine Learning* untuk mengklasifikasikan kualitas cakupan sinyal jaringan WLAN indoor ke dalam kategori Baik, Sedang, dan Buruk berdasarkan parameter RSSI, jarak, dan jumlah penghalang.



© 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

3. Mampu membangun visualisasi berbasis web yang menampilkan hasil analisis dan klasifikasi cakupan sinyal jaringan WLAN secara interaktif dan informatif.
4. Mampu menguji akurasi klasifikasi model *Random Forest* dengan membandingkan hasil prediksi terhadap data pengukuran RSSI nyata di lingkungan indoor yang berbeda dari data *training*, serta memvalidasi kelayakan aplikasi SIMNET sebagai alat bantu perencanaan penempatan *Access Point* sebelum implementasi fisik

1.3 Luaran

Luaran yang dihasilkan dari Tugas Akhir ini adalah:

1. Aplakasi simulasi analisa sinyal WLAN berbasis *Machine Learning* menggunakan *Visualisasi Web*
2. Laporan Tugas Akhir
3. Artikel Ilmiah Nasional Terakreditasi

**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

BAB V PENUTUP

4.1. Kesimpulan

Berdasarkan hasil perancangan, implementasi, dan pengujian yang telah dilakukan, dapat diambil beberapa kesimpulan sebagai berikut:

1. Aplikasi SimNet berhasil menyajikan peta cakupan sinyal jaringan WLAN indoor dalam bentuk heatmap interaktif yang menampilkan distribusi RSSI, SNR, dan *path loss* secara visual pada denah gedung yang diunggah pengguna. Heatmap tersedia dalam dua mode yang dapat dipilih secara langsung: mode *Machine Learning* yang menghasilkan klasifikasi kualitas berdasarkan pola data lapangan, dan mode Teori yang menghitung RSSI menggunakan rumus Log-Distance Path Loss. Kedua mode ditampilkan pada antarmuka web yang sama sehingga pengguna dapat membandingkan hasil keduanya secara langsung dan interaktif tanpa memerlukan perangkat lunak tambahan.
2. Algoritma *Machine Learning* berbasis *Random Forest* berhasil diterapkan untuk mengklasifikasikan kualitas cakupan sinyal jaringan WLAN indoor ke dalam tiga kategori: Baik ($RSSI \geq -65 \text{ dBm}$), Sedang ($-80 \text{ dBm} \leq RSSI < -65 \text{ dBm}$), dan Buruk ($RSSI < -80 \text{ dBm}$). Model *RandomForestClassifier* dilatih menggunakan empat parameter kondisi fisik yaitu jarak, jumlah tembok, jenis material tembok, dan tipe tembok. Selain *classifier*, dikembangkan pula *RandomForestRegressor* yang memprediksi nilai RSSI secara kontinu sebagai komplemen dari klasifikasi label. Kedua model berdiri sendiri dan tidak bergantung pada rumus Log-Distance dalam menghasilkan prediksinya, sehingga prediksi murni berasal dari pola yang dipelajari dari data pengukuran lapangan.
3. Visualisasi berbasis web berhasil dibangun menggunakan framework Plotly Dash dengan antarmuka yang interaktif dan informatif. Pengguna dapat mengunggah gambar denah gedung, mendeteksi tembok secara otomatis menggunakan OpenCV, menempatkan *Access Point* dan *Receiver* secara interaktif, mengatur skala denah, memilih channel WiFi, dan melihat hasil simulasi



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

heatmap secara *real-time*. Aplikasi berhasil di-*deploy* pada platform Render sehingga dapat diakses melalui browser dari perangkat mana saja, dengan sistem autentikasi berbasis Firebase Cloud Firestore yang memungkinkan akun pengguna digunakan lintas perangkat tanpa perlu registrasi ulang.

4. Akurasi klasifikasi model *Random Forest* berhasil diuji menggunakan 560 data uji yang tidak pernah digunakan selama pelatihan, mencapai akurasi 79,29%, F1-Score macro 79,48%, dan Cross-Validation 5-fold sebesar $79,54\% \pm 1,56\%$. Nilai OOB (*Out-of-Bag*) sebesar 77,81% yang konsisten dengan *test accuracy* mengkonfirmasi bahwa model tidak mengalami *overfitting* dan mampu menggeneralisasi pola propagasi ke kondisi baru. Pengujian pada denah Gedung G lantai 1 Politeknik Negeri Jakarta dengan skenario satu *Access Point* dan lima titik *Receiver* menunjukkan bahwa model ML dan mode Teori menghasilkan klasifikasi yang sama pada kondisi *Line of Sight*, namun berbeda pada kondisi dengan dua tembok penghalang — di mana model ML menilai kondisi tersebut sebagai Sedang sementara Teori menilainya Baik. Perbedaan ini membuktikan bahwa model ML menangkap faktor kondisi penghalang yang tidak dapat dimodelkan secara deterministik oleh rumus Log-Distance, sekaligus memvalidasi kelayakan aplikasi SimNet sebagai alat bantu perencanaan penempatan *Access Point* sebelum implementasi fisik.

4.2.Saran

Berdasarkan hasil pengembangan aplikasi yang telah dilakukan, terdapat beberapa saran untuk pengembangan lebih lanjut, yaitu:

1. Berdasarkan hasil penelitian dan keterbatasan yang ditemukan selama proses pengembangan dan pengujian aplikasi SimNet, terdapat beberapa saran yang dapat menjadi acuan untuk pengembangan lanjutan.
2. *Dataset training* yang digunakan pada penelitian ini terbatas pada lima titik jarak pengukuran (1, 2, 5, 10, dan 20 meter) dengan variasi kondisi tembok yang tidak seluruhnya mencakup semua kombinasi material dan tipe tembok secara eksplisit. Untuk meningkatkan kemampuan generalisasi model, disarankan pengumpulan data *walk test* dengan interval jarak yang lebih rapat (setiap 1 meter



© 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

dari 1 hingga 20 meter) dan pencatatan jenis material tembok serta tipe tembok secara langsung pada setiap titik pengukuran, sehingga fitur *wall_material* dan *wall_full* dapat memberikan kontribusi yang lebih bervariasi dan informatif bagi model.

3. Pengujian pada penelitian ini dilakukan di satu lokasi yaitu Gedung G lantai 1 Politeknik Negeri Jakarta dengan satu skenario penempatan *Access Point*. Disarankan untuk memperluas cakupan pengujian ke berbagai jenis gedung dengan karakteristik material yang berbeda-beda, termasuk gedung dengan material beton dominan, untuk mengevaluasi kemampuan generalisasi model pada lingkungan indoor yang lebih beragam.
4. Aplikasi SimNet saat ini hanya mendukung simulasi pada pita frekuensi 2,4 GHz dengan tiga channel (Ch 1, 6, dan 11). Mengingat semakin luasnya penggunaan WiFi pada pita frekuensi 5 GHz dan 6 GHz (WiFi 6/6E), disarankan untuk mengembangkan model dan parameter propagasi yang mendukung multi-band, termasuk penyesuaian nilai *path loss exponent* dan redaman material pada frekuensi yang lebih tinggi.
5. Simulasi yang dilakukan pada penelitian ini terbatas pada satu lantai gedung. Untuk mendukung perencanaan jaringan pada gedung bertingkat, disarankan penambahan fitur simulasi multi-lantai yang memperhitungkan *floor penetration loss* sebagai fitur tambahan pada model *Machine Learning*, sehingga aplikasi dapat digunakan untuk perencanaan jaringan WLAN pada gedung dengan lebih dari satu lantai.



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 PUSTAKA

Yunanta, R. E. C., & Susyanto, T. (2024). Analysis of WiFi Reliability at 2.4 GHz and 5 GHz Frequencies in the Environment of STMIK Sinar Nusantara Surakarta. *MALCOM: Indonesian Journal of Machine Learning and Computer Science*.

Slameto, A. A., & Khozinul Asror, M. (2023). Analisis Perbandingan Kinerja Jaringan WLAN 2,4 GHz dan 5 GHz pada Proses Thetering Menggunakan Metode QoS. *Jurnal PROCESSOR*.

Kisworo, D., Utomo, A. P., Saputra, Y. A., Rismayanti, A., & Nulpulaela, L. (2025). *Analisis Performa Jaringan WLAN pada Frekuensi 2.4 GHz dan 5 GHz sebagai Evaluasi Pemilihan Frekuensi Optimal*. *Jurnal Informatika dan Teknik Elektro Terapan*.

Firtiani, H., Maulana, A., Hasbi, M., Septiani, T., & Azzahr, R. (2025). *Analisis Perbandingan Performa Jaringan Wireless 2.4GHz dan 5GHz dalam Penggunaan Sehari-hari*. *Jurnal Komputer, Informasi dan Teknologi*.

Rusdiansah, M., Nurhayati, A. (2020). *PENGUKURAN DAN ANALISIS PERFORMANSI JARINGAN WIRELESS CLIENT PT. RAJAWALI SINERGI GROUP*.

<https://dash.plotly.com/annotations>

DAFTAR RIWAYAT HIDUP



Raja Danton's Damanik lahir di Jakarta , 15 April 2005. Lulus dari SD Swasta Antonius Otista pada tahun 2017, SMP Negeri 243 Jakarta Timur pada tahun 2020, dan SMA Swasta Fons Vitae 1 Matraman pada tahun 2023. Penulis melanjutkan pendidikan Diploma 3 (D3) di Politeknik Negeri Jakarta Depok.

© 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**



LAMPIRAN

L- 1. Sketch Code

app.py

```
# =====
# app.py
# Membuat instance aplikasi Dash + load model Random Forest.
#
# Catatan revisi: model disederhanakan menjadi Random Forest saja
# (sesuai metodologi TA), tidak lagi multi-model (RF/DT/SVM/KNN).
# Perbandingan antar-model tetap ada sebagai studi pembandingan di
# luar aplikasi (lihat uji_keselarasan_ml.py), bukan bagian dari
# sistem yang di-deploy.
# =====

import os
import joblib
import numpy as np
from dash import Dash
import sys
sys.stdout.reconfigure(encoding='utf-8')

BASE_DIR = os.path.dirname(os.path.abspath(__file__))
MODEL_PATH = os.path.join(BASE_DIR, "rf_wlan_model.pkl")
DIST_BUCKETS = [1, 2, 5, 10, 20]

try:
    bundle = joblib.load(MODEL_PATH)
    rf_model = bundle["model"]
    le = bundle["label_encoder"]
    FEATURES = bundle["features"] # ["rssi_teor",
    "distance", "wall_count"]
    ML_LOADED = True
    print(f"[OK] Model Random Forest berhasil dimuat dari
    {MODEL_PATH}.")
except Exception as e:
    rf_model = None
    le = None
    FEATURES = None
    ML_LOADED = False
    print(f"[WARN] Model tidak ditemukan, fallback ke rumus
    teori. ({e})")

def _fallback_threshold(rssi: float) -> str:
    if rssi >= -65:
        return "Baik"
    elif rssi >= -80:
        return "Sedang"
    else:
        return "Buruk"

def predict_kualitas_ml(signal_strength: float, distance_m:
float, wall_count: int,
                                wall_material: int = 2, wall_full: int =
1) -> str:
    """
```

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

```
Prediksi kualitas sinyal menggunakan Random Forest.

signal_strength (rssi_teoris) dipakai HANYA untuk fallback
threshold
saat model tidak tersedia – TIDAK dimasukkan ke RF sebagai
fitur.

RF memprediksi langsung dari kondisi fisik:
distance, wall_count, wall_material, wall_full
sehingga heatmap yang dihasilkan benar-benar dari ML, bukan
Log-Distance.
"""
if not ML_LOADED:
    return _fallback_threshold(signal_strength)

dist_int = min(DIST_BUCKETS, key=lambda d: abs(d -
distance_m))

feature_values = {
    "distance":    dist_int,
    "wall_count":  wall_count,
    "wall_material": wall_material,
    "wall_full":   wall_full,
    # rssi_teoris sengaja tidak dimasukkan
}
try:
    X_new = np.array([[feature_values.get(f, 0) for f in
FEATURES]])
except Exception as e:
    print(f"[WARN] Gagal menyusun fitur: {e}, fallback ke
rumus teori.")
    return _fallback_threshold(signal_strength)

pred = rf_model.predict(X_new)[0]
return le.inverse_transform([pred])[0]

# — Dash App Instance —
app = Dash(
    __name__,
    suppress_callback_exceptions=True,
    title="WLAN Coverage Simulator",
    update_title=None,
)
```

auth.py

```
# =====
# auth.py
# Sistem autentikasi berbasis file JSON lokal.
# Tidak memerlukan Firebase, XAMPP, atau server eksternal.
# Data pengguna disimpan di: users.json (di direktori yang sama)
# =====

import json
import hashlib
import os
import sys
```



© 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

```
sys.stdout.reconfigure(encoding='utf-8')

USERS_FILE = os.path.join(os.path.dirname(__file__),
"users.json")

def _hash(password: str) -> str:
    """Menghasilkan hash SHA-256 dari password."""
    return hashlib.sha256(password.encode()).hexdigest()

# — Akun default (jika users.json belum ada) —
DEFAULT_USERS = {
    "admin": {
        "password": _hash("admin123"),
        "name": "Administrator",
        "role": "admin",
    }
}

def _load_users() -> dict:
    """Membaca data pengguna dari file JSON."""
    if not os.path.exists(USERS_FILE):
        _save_users(DEFAULT_USERS)
        return DEFAULT_USERS
    try:
        with open(USERS_FILE, "r", encoding="utf-8") as f:
            return json.load(f)
    except Exception:
        return DEFAULT_USERS

def _save_users(users: dict) -> None:
    """Menyimpan data pengguna ke file JSON."""
    with open(USERS_FILE, "w", encoding="utf-8") as f:
        json.dump(users, f, indent=2, ensure_ascii=False)

def verify_login(username: str, password: str):
    """
    Memverifikasi login.
    Mengembalikan dict info user jika berhasil, None jika gagal.
    """
    users = _load_users()
    username = username.strip().lower()
    if username in users:
        if users[username]["password"] == _hash(password):
            return {
                "username": username,
                "name": users[username]["name"],
                "role": users[username]["role"],
            }
    return None

def register_user(username: str, password: str, name: str, role:
str = "user") -> bool:
    """
    Mendaftarkan user baru.
    Mengembalikan True jika berhasil, False jika username sudah
    ada.
    """
    users = _load_users()
    username = username.strip().lower()
```



© 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

```

if username in users:
    return False
users[username] = {
    "password": _hash(password),
    "name": name,
    "role": role,
}
_save_users(users)
return True

def change_password(username: str, old_password: str,
new_password: str) -> bool:
    """Mengganti password user. Mengembalikan True jika
    berhasil."""
    users = _load_users()
    username = username.strip().lower()
    if username in users and users[username]["password"] ==
    _hash(old_password):
        users[username]["password"] = _hash(new_password)
        _save_users(users)
        return True
    return False

def list_users() -> list:
    """Mengembalikan daftar semua username (untuk admin)."""
    return list( _load_users().keys())

```

Auth_callbacks.py

```

# =====
# auth_callbacks.py
# Callback untuk: Login, Register, Logout, Save Data,
# Export Gambar, Load Data, Statistik, Bantuan, Modal.
# =====

import json
import base64
import datetime
import io
import os
import csv
import sys
sys.stdout.reconfigure(encoding='utf-8')

import dash
from dash import Input, Output, State, ctx, html, dcc
import plotly.io as pio

from app import app
import state
from auth import verify_login, register_user
from login_layout import login_form, register_form
from navbar import create_navbar

#

```



© 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

```
# 0. Tab switch di halaman Login
#
#

@app.callback(
    Output("auth-tab-content", "children"),
    Input("auth-tabs", "value"),
)
def switch_auth_tab(tab):
    if tab == "login":
        return login_form()
    return register_form()

#

# 1. Login
#

@app.callback(
    Output("session-store", "data"),
    Output("login-page", "style"),
    Output("main-page", "style"),
    Output("navbar-container", "children"),
    Output("auth-message", "children"),
    Output("auth-message", "className"),
    Input("btn-login", "n_clicks"),
    State("login-username", "value"),
    State("login-password", "value"),
    prevent_initial_call=True,
)
def do_login(n_clicks, username, password):
    if not n_clicks:
        raise dash.exceptions.PreventUpdate

    if not username or not password:
        return (
            dash.no_update, dash.no_update, dash.no_update,
            dash.no_update,
            "! Username dan password harus diisi!", "auth-
            message auth-error",
        )

    user = verify_login(username, password)
    if user:
        navbar = create_navbar(user["name"])
        return (
            user,
            {"display": "none"},
            {"display": "block"},
            navbar,
            "",
            "auth-message",
        )
    else:
```



© 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 (
            dash.no_update, dash.no_update, dash.no_update,
            dash.no_update,
            "✖ Username atau password salah!", "auth-message
            auth-error",
        )
    #

# 2. Register
#

@app.callback(
    Output("auth-message", "children", allow_duplicate=True),
    Output("auth-message", "className", allow_duplicate=True),
    Input("btn-register", "n_clicks"),
    State("reg-name", "value"),
    State("reg-username", "value"),
    State("reg-password", "value"),
    prevent_initial_call=True,
)
def do_register(n_clicks, name, username, password):
    if not n_clicks:
        raise dash.exceptions.PreventUpdate

    if not name or not username or not password:
        return " ! Semua kolom harus diisi!", "auth-message
        auth-error"

    if len(password) < 6:
        return " ! Password minimal 6 karakter!", "auth-message
        auth-error"

    success = register_user(username, password, name)
    if success:
        return "✔ Akun berhasil dibuat! Silakan login.", "auth-
        message auth-success"
    else:
        return "✖ Username sudah digunakan, pilih username
        lain.", "auth-message auth-error"
    #

# 3. Logout
#

@app.callback(
    Output("session-store", "data", allow_duplicate=True),
    Output("login-page", "style", allow_duplicate=True),
    Output("main-page", "style", allow_duplicate=True),
    Input("btn-logout", "n_clicks"),
    prevent_initial_call=True,

```



© 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

```

)
def do_logout(n_clicks):
    if not n_clicks:
        raise dash.exceptions.PreventUpdate
    # Bersihkan semua state simulasi
    state.ap_x.clear(); state.ap_y.clear()
    state.ext_ap_x.clear(); state.ext_ap_y.clear()
    state.rx_x.clear(); state.rx_y.clear()
    state.rx_data.clear(); state.walls.clear()
    state.auto_walls.clear(); state.temp_wall.clear()
    state.scale_points.clear()
    state.uploaded_image = None
    return None, {"display": "flex"}, {"display": "none"}

#
#
# 4. Modal: Buka / Tutup
#

@app.callback(
    Output("modal-overlay", "style"),
    Output("modal-content", "children"),
    Input("btn-save-data", "n_clicks"),
    Input("btn-load-data", "n_clicks"),
    Input("btn-stats", "n_clicks"),
    Input("btn-help", "n_clicks"),
    Input("btn-close-modal", "n_clicks"),
    State("graph", "figure"),
    State("session-store", "data"),
    prevent_initial_call=True,
)
def toggle_modal(n_save, n_load, n_stats, n_help, n_close,
figure, session):
    triggered = ctx.triggered_id

    if triggered == "btn-close-modal" or not triggered:
        return {"display": "none"}, ""

    # — Save Data —
    if triggered == "btn-save-data":
        content = _build_save_modal(session)
        return {"display": "flex"}, content

    # — Muat Data —
    if triggered == "btn-load-data":
        content = _build_load_modal()
        return {"display": "flex"}, content

    # — Statistik —
    if triggered == "btn-stats":
        content = _build_stats_modal()
        return {"display": "flex"}, content

    # — Bantuan —
    if triggered == "btn-help":
        content = _build_help_modal()

```



© 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 {"display": "flex"}, content

        return {"display": "none"}, ""

#
#
# 5. Simpan Data → Download JSON
#
#

@app.callback(
    Output("download-data", "data"),
    Input("btn-confirm-save", "n_clicks"),
    State("save-project-name", "value"),
    State("session-store", "data"),
    prevent_initial_call=True,
)
def download_data(n_clicks, project_name, session):
    if not n_clicks:
        raise dash.exceptions.PreventUpdate

    name = project_name.strip() if project_name else
    "proyek_wlan"
    timestamp = datetime.datetime.now().strftime("%Y%m%d_%H%M%S")
    filename = f"{name}_{timestamp}.json"

    data = {
        "meta": {
            "project_name": name,
            "saved_by": session.get("name", "unknown") if session
            else "unknown",
            "saved_at": datetime.datetime.now().isoformat(),
            "version": "2.0",
        },
        # Gambar denah (base64) – agar bisa dimuat kembali
        # sepenuhnya
        "uploaded_image": state.uploaded_image,
        "img_width": state.img_width,
        "img_height": state.img_height,
        # Access Point
        "ap": {"x": state.ap_x, "y": state.ap_y},
        "ext_ap": {
            "x": state.ext_ap_x,
            "y": state.ext_ap_y,
            "measured": getattr(state, "ext_ap_measured", []),
        },
        # Receiver
        "rx": [
            {
                "x": d[0], "y": d[1],
                "rssi": d[2].get("rssi") if
                isinstance(d[2], dict) else None,
                "snr": d[2].get("snr") if
                isinstance(d[2], dict) else None,
                "throughput": d[2].get("throughput") if
                isinstance(d[2], dict) else 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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

        "path_loss": d[2].get("path_loss") if
        isinstance(d[2], dict) else None,
        "kualitas_ml": d[2].get("kualitas_ml","-") if
        isinstance(d[2], dict) else "-",
        "wall_count": d[2].get("wall_count", 0) if
        isinstance(d[2], dict) else 0,
        "jarak": d[3],
        "terhubung": d[4] if len(d) > 4 else None,
    }
    for d in state.rx_data
],
# Tembok manual
"walls": [
    {
        "p1": list(w[0]), "p2": list(w[1]),
        "material": w[2],
        "source": w[3] if len(w) > 3 else "manual",
        "conf": w[4] if len(w) > 4 else 1.0,
    }
    for w in state.walls
],
# Tembok otomatis (hasil deteksi OpenCV)
"auto_walls": [
    {
        "p1": list(w[0]), "p2": list(w[1]),
        "material": w[2],
        "source": w[3] if len(w) > 3 else "color",
        "conf": w[4] if len(w) > 4 else 1.0,
    }
    for w in state.auto_walls
],
"use_auto_walls": state.use_auto_walls,
# Skala
"scale_meter": state.scale_meter,
"scale_points": [list(p) for p in state.scale_points],
# Parameter simulasi
"tx_power": state.tx_power,
"noise_floor": getattr(state, "noise_floor", -90),
"ext_ap_power_src": getattr(state, "ext_ap_power_source",
"teori"),
# Pengukuran manual
"measurements": [
    {"jarak": m[0], "rssi": m[1], "label": m[2]}
    for m in state.wifi_measurements
],
}

return dcc.send_string(
    json.dumps(data, indent=2, ensure_ascii=False),
    filename=filename,
)

#
#
# 6. Export Gambar → Download PNG (Termasuk Tabel Receiver)
#
#

```



© 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

```
@app.callback(
    Output("download-image", "data"),
    Input("btn-export-img", "n_clicks"),
    State("graph", "figure"),
    prevent_initial_call=True,
)
def download_image(n_clicks, figure):
    if not n_clicks or figure is None:
        raise dash.exceptions.PreventUpdate

    fname = "wlan_coverage"
    timestamp = datetime.datetime.now().strftime("%Y%m%d_%H%M%S")
    full_name = f"{fname}_{timestamp}.png"

    try:
        import plotly.graph_objects as go
        fig = go.Figure(figure)
        # Render plotly figure to PNG bytes
        img_bytes = pio.to_image(fig, format="png", width=1400,
            height=900, scale=2)

        # Draw the receiver data table at the bottom using PIL
        from PIL import Image, ImageDraw, ImageFont

        fig_img = Image.open(io.BytesIO(img_bytes))
        fig_w, fig_h = fig_img.size

        # Table dimensions and layout parameters
        padding = 30
        row_height = 40
        header_height = 50
        title_height = 50

        rx_data = state.rx_data
        num_rows = len(rx_data)
        table_h = title_height + header_height + (max(1,
            num_rows) * row_height) + (2 * padding)

        total_w = fig_w
        total_h = fig_h + table_h

        # Create base canvas with white background
        comp_img = Image.new("RGB", (total_w, total_h), "white")
        comp_img.paste(fig_img, (0, 0))

        draw = ImageDraw.Draw(comp_img)

        # Load fonts
        try:
            font_title = ImageFont.truetype("arial.ttf", 24)
            font_header = ImageFont.truetype("arial.ttf", 16)
            font_body = ImageFont.truetype("arial.ttf", 14)
        except IOError:
            font_title = ImageFont.load_default()
            font_header = ImageFont.load_default()
            font_body = ImageFont.load_default()

        y_offset = fig_h + padding
```



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

```
# Title of table
draw.text((padding, y_offset), "DATA RECEIVER HASIL
SIMULASI WLAN", fill="black", font=font_title)
y_offset += title_height

headers = ["No", "Receiver", "Posisi (X, Y)", "Jarak ke
AP", "RSSI", "SNR", "Throughput", "Path Loss"]
usable_w = total_w - (2 * padding)
col_ratios = [0.06, 0.12, 0.16, 0.16, 0.12, 0.12, 0.14,
0.12]
col_widths = [int(usable_w * r) for r in col_ratios]
col_widths[-1] = usable_w - sum(col_widths[:-1])

# Header background
draw.rectangle([padding, y_offset, total_w - padding,
y_offset + header_height], fill="#e4e8f0")

# Header text
x_curr = padding
for idx, h in enumerate(headers):
    draw.text((x_curr + 12, y_offset + 15), h,
fill="black", font=font_header)
    x_curr += col_widths[idx]

y_offset += header_height

if num_rows == 0:
    draw.rectangle([padding, y_offset, total_w - padding,
y_offset + row_height], fill="#f5f7fa")
    draw.text((padding + 15, y_offset + 12), "Tidak ada
data receiver yang ditempatkan.", fill="gray", font=font_body)
    y_offset += row_height
else:
    for r_idx, rx in enumerate(rx_data):
        rx_x, rx_y, params, jarak, *rest = rx
        rx_name = f"Rx {r_idx + 1}"
        pos_str = f"({int(rx_x)}, {int(rx_y)})"
        dist_str = f"{jarak:.2f} m" if jarak is not None
        else "--"

        rssi_val = params.get("rssi")
        snr_val = params.get("snr")
        thr_val = params.get("throughput")
        pl_val = params.get("path_loss")

        rssi_str = f"{rssi_val:.1f} dBm" if rssi_val is
not None else "--"
        snr_str = f"{snr_val:.1f} dB" if snr_val is not
None else "--"
        thr_str = f"{thr_val:.1f} Mbps" if thr_val is not
None else "--"
        pl_str = f"{pl_val:.1f} dB" if pl_val is not None
        else "--"

        row_data = [
            str(r_idx + 1), rx_name, pos_str, dist_str,
            rssi_str, snr_str, thr_str, pl_str
        ]
```



© 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

```

        bg_color = "white" if r_idx % 2 == 0 else
        "#f9fafb"
        draw.rectangle([padding, y_offset, total_w -
padding, y_offset + row_height], fill=bg_color)

        x_curr = padding
        for col_i, text in enumerate(row_data):
            draw.text((x_curr + 12, y_offset + 12), text,
fill="black", font=font_body)
            x_curr += col_widths[col_i]

        y_offset += row_height

    # Outer boundary lines
    draw.rectangle([padding, fig_h + padding + title_height,
total_w - padding, y_offset], outline="#cfd8dc", width=2)

    # Vertical grid lines
    x_curr = padding
    for w in col_widths[:-1]:
        x_curr += w
        draw.line([x_curr, fig_h + padding + title_height,
x_curr, y_offset], fill="#cfd8dc", width=1)

    # Horizontal grid lines
    y_curr = fig_h + padding + title_height + header_height
    for _ in range(max(1, num_rows)):
        draw.line([padding, y_curr, total_w - padding,
y_curr], fill="#cfd8dc", width=1)
        y_curr += row_height

    out_buf = io.BytesIO()
    comp_img.save(out_buf, format="PNG")
    encoded = base64.b64encode(out_buf.getvalue()).decode()

    return {
        "content": encoded,
        "filename": full_name,
        "type": "image/png",
        "base64": True,
    }
except Exception as e:
    import plotly.graph_objects as go
    fig = go.Figure(figure)
    html_str = pio.to_html(fig, full_html=True)
    return dcc.send_string(html_str,
filename="wlan_coverage.html")

#
#
# 7. Muat Data dari Upload JSON
#
#

@app.callback(
    Output("load-status", "children"),

```



© 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

```

Output("graph", "figure", allow_duplicate=True),
Input("upload-json-data", "contents"),
State("upload-json-data", "filename"),
prevent_initial_call=True,
)

def load_project_data(contents, filename):
    if contents is None:
        raise dash.exceptions.PreventUpdate

    from visualization import create_figure

    try:
        _, content_string = contents.split(",")
        decoded = base64.b64decode(content_string).decode("utf-
8")

        data = json.loads(decoded)
        ver = data.get("meta", {}).get("version", "1.0")

        # — Gambar denah —————
        state.uploaded_image = data.get("uploaded_image")
        state.img_width = data.get("img_width", 1000)
        state.img_height = data.get("img_height", 1000)

        # — Access Point —————
        state.ap_x.clear(); state.ap_y.clear()
        state.ap_x.extend(data.get("ap", {}).get("x", []))
        state.ap_y.extend(data.get("ap", {}).get("y", []))

        # — Existing AP —————
        ext = data.get("ext_ap", {})
        state.ext_ap_x.clear(); state.ext_ap_y.clear()
        state.ext_ap_x.extend(ext.get("x", []))
        state.ext_ap_y.extend(ext.get("y", []))
        state.ext_ap_measured = list(ext.get("measured", []))

        # — Receiver —————
        state.rx_data.clear()
        for rx in data.get("rx", []):
            params = {
                "rssi": rx.get("rssi"),
                "snr": rx.get("snr"),
                "throughput": rx.get("throughput"),
                "path_loss": rx.get("path_loss"),
                "kualitas_ml": rx.get("kualitas_ml", "-"),
                "wall_count": rx.get("wall_count", 0),
            }
            state.rx_data.append((
                rx["x"], rx["y"], params,
                rx.get("jarak"), rx.get("terhubung"),
            ))

        # — Tembok manual —————
        state.walls.clear()
        for w in data.get("walls", []):
            state.walls.append((
                tuple(w["p1"]), tuple(w["p2"]),
                w["material"],
                w.get("source", "manual"),
                w.get("conf", 1.0),
            ))

```



© 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

```

    ))

    # — Tembok otomatis —————
    state.auto_walls.clear()
    for w in data.get("auto_walls", []):
        state.auto_walls.append((
            tuple(w["p1"]), tuple(w["p2"]),
            w["material"],
            w.get("source", "color"),
            w.get("conf", 1.0),
        ))
    state.use_auto_walls = data.get("use_auto_walls", True)

    # — Skala —————
    state.scale_meter = data.get("scale_meter", 0.05)
    state.scale_points = [tuple(p) for p in
data.get("scale_points", [])]

    # — Parameter —————
    state.tx_power = data.get("tx_power", 15)
    state.noise_floor = data.get("noise_floor", -90)
    state.ext_ap_power_source = data.get("ext_ap_power_src",
"teori")

    # — Pengukuran manual —————
    state.wifi_measurements.clear()
    for m in data.get("measurements", []):
        state.wifi_measurements.append((m["jarak"],
m["rssi"], m["label"]))

    meta = data.get("meta", {})
    has_img = "Ada" if state.uploaded_image else "Tidak ada"
    n_aw = len(state.auto_walls)

    status = html.Div([
        html.P(f"Berhasil memuat: {filename}",
            style={"color": "green", "fontWeight": "bold"}),
        html.P(f"Proyek: {meta.get('project_name', '-') }
(v{ver}))"),
        html.P(f"Disimpan oleh: {meta.get('saved_by', '-') }"),
        html.Hr(style={"margin": "6px 0"}),
        html.P(f"AP: {len(state.ap_x)} | Rx:
{len(state.rx_data)} "
            f"| Tembok: {len(state.walls)} | Auto:
{n_aw}"),
        html.P(f"Gambar denah: {has_img} | Skala:
{state.scale_meter:.4f} m/px"),
        html.P("Tutup modal – grafik sudah diperbarui
otomatis.",
            style={"color": "#2196F3", "fontSize": "12px", "ma
rginTop": "6px"}),
    ])

    # Langsung render figure dengan gambar yang sudah di-
restore
    return status, create_figure()

except Exception as e:

```



© 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 html.P(f"Gagal memuat file: {str(e)}",
style={"color":"red"}), dash.no_update

#
=====
# Helper: Bangun konten modal
#
=====

def _build_save_modal(session):
    user_name = session.get("name", "Pengguna") if session else
"Pengguna"
    return html.Div([
        html.H3("💾 Simpan Data Proyek", className="modal-
title"),
        html.P(f"Disimpan sebagai: {user_name}",
className="modal-hint"),
        html.Label("Nama Proyek:", className="input-label"),
        dcc.Input(
            id="save-project-name",
            type="text",
            placeholder="cth: gedung_A_lantai2",
            value="proyek_wlan",
            className="ctrl-input",
        ),
        html.Br(), html.Br(),
        html.Div([
            html.P(f"📶 AP: {len(state.ap_x)} titik"),
            html.P(f"📡 Receiver: {len(state.rx_data)} titik"),
            html.P(f"🧱 Tembok: {len(state.walls)} segmen"),
            html.P(f"📶 Pengukuran:
{len(state.wifi_measurements)} data"),
        ], className="save-summary"),
        html.Button("⬇️ Download JSON", id="btn-confirm-save",
n_clicks=0, className="ctrl-btn ctrl-btn-green"),
    ])

def _build_export_modal(figure):
    has_kaleido = False
    try:
        import kaleido
        has_kaleido = True
    except ImportError:
        pass

    warn = ""
    if not has_kaleido:
        warn = html.Div([
            html.P("⚠️ Library kaleido tidak ditemukan.",
style={"color": "orange"}),
            html.P("File akan diekspor sebagai HTML. Untuk ekspor
PNG, jalankan:"),
            html.Code("pip install kaleido",
style={"backgroundColor": "#1e1e1e", "color": "#4fc3f7",
"padding": "4px 8px", "borderRadius": "4px"}),

```



© 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

```

], style={"marginBottom": "12px"})

ext = "PNG" if has_kaleido else "HTML"
return html.Div([
    html.H3(f"📄 Export Gambar ({ext})", className="modal-
title"),
    warn,
    html.Label("Nama File:", className="input-label"),
    dcc.Input(
        id="export-filename",
        type="text",
        placeholder="cth: coverage_gedung_A",
        value="wlan_coverage",
        className="ctrl-input",
    ),
    html.Br(), html.Br(),
    html.P(f"Resolusi: 1400 x 900 px (scale x2) - format
.{ext.lower()}", className="modal-hint"),
    html.Button(f"↓ Download {ext}", id="btn-confirm-export",
n_clicks=0, className="ctrl-btn ctrl-btn-green"),
])

def _build_load_modal():
    return html.Div([
        html.H3("📁 Muat Data Proyek", className="modal-title"),
        html.P("Pilih file JSON hasil simpan dari aplikasi ini
untuk melanjutkan simulasi.", className="modal-hint"),
        html.Div([
            html.Span("📄 Format JSON yang didukung:",
style={"fontWeight": "600", "color": "var(--text-main)"}),
            html.Pre(
                '{\n'
                '  "ap": { "x": [100, 250], "y": [150, 300] },\n'
                '  "rx": [ { "x": 120, "y": 180, "rssi": -55}
],\n'
                '  "walls": [ { "p1": [50, 50], "p2": [50, 200],
"material": "bata" } ]\n'
                '}',
                style={
                    "backgroundColor": "#f4f6f8",
                    "border": "1px solid var(--border)",
                    "borderRadius": "6px",
                    "padding": "8px 12px",
                    "fontSize": "11px",
                    "color": "#37474f",
                    "fontFamily": "monospace",
                    "marginTop": "6px",
                    "whiteSpace": "pre-wrap"
                }
            )
        ]),
        ], style={"marginBottom": "12px"}),
    dcc.Upload(
        id="upload-json-data",
        children=html.Div([
            html.Span("📁 Drag & Drop atau "),
            html.A("Pilih File JSON"),
        ]),
        className="upload-box",

```



© 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

```

        accept=".json",
    ),
    html.Br(),
    html.Div(id="load-status"),
])

def _build_stats_modal():
    if not state.rx_data:
        return html.Div([
            html.H3("🇮🇩 Statistik Simulasi", className="modal-
title"),
            html.P("Belum ada data receiver. Tambahkan receiver
terlebih dahulu.", className="modal-hint"),
        ])

    rssi_vals = [d[2]["rssi"] for d in state.rx_data if
d[2]["rssi"] is not None]
    snr_vals = [d[2]["snr"] for d in state.rx_data if
d[2]["snr"] is not None]
    thr_vals = [d[2]["throughput"] for d in state.rx_data if
d[2]["throughput"] is not None]

    import statistics as st
    def fmt(vals):
        if not vals: return "-"
        return f"{min(vals):.1f} / {st.mean(vals):.1f} /
{max(vals):.1f}"

    # Hitung coverage
    total = len(rssi_vals)
    baik = sum(1 for v in rssi_vals if v >= -65)
    sedang = sum(1 for v in rssi_vals if -80 <= v < -65)
    buruk = sum(1 for v in rssi_vals if -90 <= v < -80)
    blind = sum(1 for v in rssi_vals if v < -90)

    return html.Div([
        html.H3("🇮🇩 Statistik Simulasi", className="modal-
title"),
        html.Table([
            html.Tr([html.Th("Parameter"), html.Th("Min"),
html.Th("Rata-rata"), html.Th("Maks")]),
            html.Tr([html.Td("RSSI
(dBm)", html.Td(f"{min(rssi_vals):.1f}"), html.Td(f"{su
m(rssi_vals)/len(rssi_vals):.1f}"),
html.Td(f"{max(rssi_vals):.1f}")) if rssi_vals else None,
            html.Tr([html.Td("SNR
(dB)", html.Td(f"{min(snr_vals):.1f}"), html.Td(f"{
sum(snr_vals)/len(snr_vals):.1f}"), html.Td(f"{max(snr_vals):.1f
}")) if snr_vals else None,
            html.Tr([html.Td("Throughput (Mbps)",
html.Td(f"{min(thr_vals):.1f}"), html.Td(f"{sum(thr_vals)/len
(thr_vals):.1f}"), html.Td(f"{max(thr_vals):.1f}")) if
thr_vals else None,
        ], className="stats-table"),
        html.Hr(),
        html.H4("Distribusi Kualitas Sinyal"),
        html.Div(className="quality-bars", children=[

```



© 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

```

        html.Div([html.Span("✅
Baik"),      html.Span(f"{baik}/{total}
({baik/total*100:.0f}%)")],      className="quality-row quality-
good"),

        html.Div([html.Span("🟡
Sedang"),    html.Span(f"{sedang}/{total}
({sedang/total*100:.0f}%)")],      className="quality-row quality-
mid"),

        html.Div([html.Span("🔴
Buruk"),      html.Span(f"{buruk}/{total}
({buruk/total*100:.0f}%)")],      className="quality-row quality-
bad"),

        html.Div([html.Span("🔴 Blind Spot"),
html.Span(f"{blind}/{total} ({blind/total*100:.0f}%)")],
className="quality-row quality-blind"),
    ]),
    html.Hr(),
    html.P(f"Total AP: {len(state.ap_x)} | Existing AP:
{len(state.ext_ap_x)} | Total Receiver: {total}"),
    html.P(f"Skala: 1 px = {state.scale_meter:.4f} m | Tx
Power: {state.tx_power} dBm"),
    ])

def _build_help_modal():
    return html.Div([
        html.H3("📖 Panduan Penggunaan", className="modal-title"),
        html.Div(className="help-content", children=[
            html.H4("📄 Cara Memulai"),
            html.Ol([
                html.Li("Upload gambar denah gedung/ruangan
(PNG/JPG)."),
                html.Li("Pilih mode 'Set Skala', klik 2 titik
dengan jarak nyata diketahui, lalu klik 'Hitung Skala'."),
                html.Li("Pilih mode 'Access Point' lalu klik pada
grafik untuk menempatkan AP."),
                html.Li("Pilih mode 'Receiver' lalu klik untuk
melihat kualitas sinyal di titik tersebut."),
            ]),
            html.H4("📄 Mode Kontrol"),
            html.Table([
                html.Tr([html.Th("Mode"), html.Th("Fungsi")]),
                html.Tr([html.Td("Access Point"),
html.Td("Tempatkan AP baru di titik klik")]),
                html.Tr([html.Td("Receiver"), html.Td("Ukur
kualitas sinyal di titik klik")]),
                html.Tr([html.Td("Obstacle"), html.Td("Gambar
tembok/dinding (klik 2 titik)")]),
                html.Tr([html.Td("Set Skala"), html.Td("Kalibrasi
skala jarak (klik 2 titik)")]),
                html.Tr([html.Td("Hapus Titik"), html.Td("Klik
dekat AP/Rx untuk menghapusnya")]),
                html.Tr([html.Td("Hapus Garis"), html.Td("Klik
dekat tembok untuk menghapusnya")]),
                html.Tr([html.Td("Geser Titik"), html.Td("Klik
titik lalu klik lokasi tujuan")]),
                html.Tr([html.Td("Existing AP"), html.Td("Tandai
AP yang sudah ada di lapangan")]),
            ]),

```



© 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

```

], className="stats-table"),
html.H4("📄 Fitur Menu Atas"),
html.Ul([
    html.Li("Simpan Data – Ekspor seluruh data
simulasi ke file JSON"),
    html.Li("Export Gambar – Simpan grafik sebagai
gambar PNG (perlu kaleido)"),
    html.Li("Muat Data – Impor file JSON yang
sebelumnya disimpan"),
    html.Li("Statistik – Ringkasan kualitas sinyal
semua receiver"),
    html.Li("Bantuan – Panduan ini"),
]),
html.H4("📦 Instalasi Library Tambahan"),
html.P("Untuk ekspor PNG, install kaleido:"),
html.Code("pip install kaleido",
style={"backgroundColor": "#1e1e1e", "color": "#4fc3f7",
"padding": "4px 8px", "borderRadius": "4px", "display": "block",
"margin": "6px 0"}),
]),
#
#
# Export CSV Data Receiver
#
@app.callback(
    Output("download-data", "data", allow_duplicate=True),
    Input("btn-export-csv", "n_clicks"),
    prevent_initial_call=True,
)
def export_csv(n_clicks):
    if not n_clicks or not state.rx_data:
        raise dash.exceptions.PreventUpdate
    out = io.StringIO()
    w = csv.writer(out)
    w.writerow(["No", "X (px)", "Y (px)", "RSSI (dBm)", "SNR (dB)",
        "Throughput (Mbps)", "Path Loss (dB)", "Jarak (m)",
        "Jumlah Tembok", "Terhubung ke", "Kualitas ML"])
    for i, d in enumerate(state.rx_data):
        p = d[2] if isinstance(d[2], dict) else {}
        w.writerow([
            i+1, round(d[0],1), round(d[1],1),
            round(p.get("rssi",0),2) if
p.get("rssi") is not None else "-",
            round(p.get("snr",0),2) if
p.get("snr") is not None else "-",
            round(p.get("throughput",0),2) if p.get("throughput")
is not None else "-",
            round(p.get("path_loss",0),2) if
p.get("path_loss") is not None else "-",
            round(d[3],2) if
d[3] is not None else "-",
            p.get("wall_count", "-"),

```



© 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

```
d[4] if len(d) > 4 else "-",
p.get("kualitas_ml", "-"),
])
ts = datetime.datetime.now().strftime("%Y%m%d_%H%M%S")
return dict(content=out.getvalue(),
filename=f"data_receiver_{ts}.csv", type="text/csv")
```

Buka_SimNet.bat

```
@echo off
setlocal enabledelayedexpansion
title SIMNET Launcher

rem =====
rem Buka_SIMNET.bat
rem Letakkan file ini di folder yang SAMA dengan run.py
rem Lalu klik dua kali untuk menjalankan aplikasi SIMNET.
rem (Atau buat shortcut ke file ini dan letakkan di Desktop)
rem =====

set "PROJECT_DIR=%~dp0"
cd /d "%PROJECT_DIR%"

set "URL=http://127.0.0.1:8050/"

rem ---- Cari Python: pakai venv lokal jika ada, kalau tidak
rem pakai Python sistem ----
set "PYTHON_EXE=python"
set "USING_VENV=0"

if exist "%PROJECT_DIR%venv\Scripts\python.exe" (
    set "PYTHON_EXE=%PROJECT_DIR%venv\Scripts\python.exe"
    set "USING_VENV=1"
) else if exist "%PROJECT_DIR%.venv\Scripts\python.exe" (
    set "PYTHON_EXE=%PROJECT_DIR%.venv\Scripts\python.exe"
    set "USING_VENV=1"
)

echo =====
echo SIMNET - WLAN Coverage Simulator
echo Memulai server, mohon tunggu...
echo =====
echo.

if "%USING_VENV%"=="1" (
    if not exist "%PYTHON_EXE%" (
        echo [ERROR] Virtual environment ditemukan tapi
        python.exe tidak ada di:
        echo %PYTHON_EXE%
        pause
        exit /b 1
    )
) else (
    where python >nul 2>&1
    if errorlevel 1 (
        echo [ERROR] Python tidak ditemukan di PATH.
```



© 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

```

    echo Install Python dari
    https://www.python.org/downloads/
    echo dan saat instalasi centang "Add Python to PATH".
    pause
    exit /b 1
)

rem ---- Jalankan server Dash di jendela terpisah ----
start "SIMNET Server - JANGAN DITUTUP selama aplikasi dipakai"
cmd /k "%PYTHON_EXE%" run.py"

echo Menunggu server siap...
set /a ATTEMPTS=0

:WAIT_LOOP
set /a ATTEMPTS+=1
if %ATTEMPTS% GTR 40 (
    echo.
    echo [WARN] Server belum merespons setelah 40 detik.
    echo Cek jendela "SIMNET Server" untuk melihat pesan error.
    pause
    exit /b 1
)

powershell -NoProfile -Command "try { (New-Object
Net.Sockets.TcpClient).Connect('127.0.0.1',8050); exit 0 } catch
{ exit 1 }" >nul 2>&1
if errorlevel 1 (
    timeout /t 1 /nobreak >nul
    goto WAIT_LOOP
)

echo Server siap. Membuka browser...
start "" "%URL%"

exit /b 0

```

Callbacks.py

```

# =====
# callbacks.py -- v3
# Tambahan baru:
#   - Mode 'edit_wall': klik tembok -> pilih material baru
#   - Panel info tembok: tampilkan material, source, confidence
#   - Noise level, Existing AP workflow (dari v2)
# =====

import numpy as np
import pandas as pd
import dash
from dash import Input, Output, State, ctx, html, dcc
from PIL import Image
import io, base64, statistics

from sklearn.metrics import (

```



© 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

```

accuracy_score, precision_score, recall_score, f1_score,
confusion_matrix
)

from app import app, ML_LOADED, predict_kualitas_ml
import state
from functions import (
    get_parameters_with_ml, hitung_jarak, update_all_rx,
    get_kualitas, get_warna_from_label, get_existing_wifi,
    get_connected_ap, wall_material, warna_material,
)
from vision import auto_scan_walls, change_wall_material
from visualization import (create_figure, auto_placement_logic,
                           create_kualitas_comparison_figure,
                           create_test_data_scatter,
                           create_test_overlay_figure,
                           cari_posisi_ap_optimal_ml)
from train_all_models import compute_rssi_teoris

MAT_OPTIONS = [
    {'label': '[Beton] Beton (-15 dB)', 'value': 'beton'},
    {'label': '[Bata] Bata (-12 dB)', 'value': 'bata'},
    {'label': '[Kaca] Kaca (-5 dB)', 'value': 'kaca'},
    {'label': '[Kayu] Kayu (-3 dB)', 'value': 'kayu'},
]

SRC_LABEL = {'color': 'Warna piksel', 'thickness': 'Ketebalan
piksel', 'manual': 'Manual'}
SRC_COLOR = {'color': 'green', 'thickness': 'orange', 'manual':
'#9C27B0'}

# -- 0. Popup panduan -----
@app.callback(
    Output('autoscan-info-popup', 'style'),
    Input('autoscan-info-btn', 'n_clicks'),
    prevent_initial_call=True,
)
def toggle_thickness_popup(n):
    if n % 2 == 1:
        return {'display': 'block', 'padding': '8px', 'border': '1px
solid #aaa',
               'borderRadius': '6px', 'backgroundColor': '#fff9e6',
               'marginTop': '6px'}
    return {'display': 'none'}

# -- 1. Upload Gambar -----
@app.callback(
    Output('graph', 'figure', allow_duplicate=True),
    Output('output', 'children', allow_duplicate=True),
    Input('upload-image', 'contents'),
    State('toggle_autoscan', 'value'),
    State('material', 'value'),
    prevent_initial_call=True,
)
def update_image(contents, toggle_autoscan, material_value):
    if not contents: return dash.no_update, dash.no_update
    state.uploaded_image = contents

```



© 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

```
_, cs = contents.split(',')
img = Image.open(io.BytesIO(base64.b64decode(cs)))
state.img_width, state.img_height = img.size

# Bersihkan semua data dari sesi sebelumnya
for lst in [state.ap_x, state.ap_y, state.ext_ap_x,
state.ext_ap_y,
state.rx_x, state.rx_y, state.rx_data,
state.scale_points,
state.walls, state.auto_walls, state.temp_wall,
state.ext_ap_measured]:
    try: lst.clear()
    except: pass

state.dead_zones = []
state.dead_zone_tmp = None
state.heatmap_locked = False
state.heatmap_locked_grid = None
state.scale_meter = 0.05
state.selected_wall_idx = None
state.selected_wall_source = None

if toggle_autoscan == 'on':
    state.use_auto_walls = True
    auto_scan_walls(contents, material_value)
    # Hitung statistik deteksi
    from collections import Counter
    src_c = Counter(w[3] if len(w)>3 else '?' for w in
state.auto_walls)
    mat_c = Counter(w[2] for w in state.auto_walls)
    msg = (f"Gambar diunggah. {len(state.auto_walls)} tembok
terdeteksi "
f"| Warna:{src_c.get('color',0)}
Ketebalan:{src_c.get('thickness',0)} "
f"| {dict(mat_c)}")
else:
    state.use_auto_walls = False
    msg = "Gambar diunggah tanpa scan otomatis."

return create_figure(), msg

@app.callback(
    Output('graph', 'figure', allow_duplicate=True),
    Output('output', 'children', allow_duplicate=True),
    Input('toggle_autoscan', 'value'),
    State('material', 'value'),
    State('toggle_heatmap', 'value'),
    State('parameter', 'value'),
    prevent_initial_call=True,
)

def toggle_autoscan(toggle, mat, hm, param):
    if not param: param = 'rssi'
    if toggle == 'on':
        state.use_auto_walls = True
        if state.uploaded_image and not state.auto_walls:
            auto_scan_walls(state.uploaded_image, mat)
        update_all_rx()
        return create_figure(hm, param), "Tembok otomatis
diaktifkan.", *NO_PANEL
```



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

```

state.use_auto_walls = False
update_all_rx()
return create_figure(hm, param), "Tembok otomatis
dinonaktifkan.", *NO_PANEL

# -- 2. Skala -----

@app.callback(
    Output('output', 'children', allow_duplicate=True),
    Output('scale-display', 'children'),
    Output('scale-display', 'style'),
    Input('btn-scale', 'n_clicks'),
    State('real-distance', 'value'),
    prevent_initial_call=True,
)
def hitung_skala(_, real_distance):
    if len(state.scale_points) < 2:
        return "Pilih 2 titik skala terlebih dahulu!",
dash.no_update, dash.no_update
    if not real_distance:
        return "Masukkan jarak nyata (meter)!", dash.no_update,
dash.no_update

    (x1,y1), (x2,y2) = state.scale_points
    state.length_points = np.sqrt((x2-x1)**2+(y2-y1)**2)
    state.scale_meter = real_distance / state.length_points

    notif = f"Panjang garis = {state.length_points:.2f} satuan,
Skala = 1:{state.scale_meter:.5f}"

    display_text = [
        html.B("📏 Skala Aktif"),
        html.Br(),
        f"Panjang garis: {state.length_points:.1f} satuan",
        html.Br(),
        f"Jarak nyata: {real_distance} m",
        html.Br(),
        html.B(f"skala = 1 : {state.scale_meter:.4f}"),
    ]
    display_style = {
        "fontSize": "12px", "color": "#dadbdb",
        "marginTop": "6px", "padding": "6px",
        "backgroundColor": "#1d82e0", "borderRadius": "4px",
        "display": "block"
    }
    return notif, display_text, display_style

# -- 3. Auto AP -----

@app.callback(
    Output('graph', 'figure', allow_duplicate=True),
    Output('output', 'children', allow_duplicate=True),
    Input('btn-auto', 'n_clicks'),
    State('auto-ap-count', 'value'),
    State('toggle_heatmap', 'value'),
    State('parameter', 'value'),
    prevent_initial_call=True,
)

```



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

```
def run_auto(_, cnt, hm, param):
    if cnt and cnt >= 1:
        info = auto_placement_logic(cnt)
        update_all_rx()
        return create_figure(hm, param), info
    return dash.no_update

# -- 4. Noise level -----

@app.callback(
    Output('graph', 'figure', allow_duplicate=True),
    Output('output', 'children', allow_duplicate=True),
    Input('noise-level', 'value'),
    State('toggle_heatmap', 'value'),
    State('parameter', 'value'),
    prevent_initial_call=True,
)
def update_noise(noise_level, hm, param):
    if not param: param = 'rssi'
    state.noise_floor = {'none':-90, 'medium':-80, 'high':-
70}.get(noise_level, -90)
    update_all_rx()
    lbl = {'none':"Tidak ada interferensi (-90 dBm)",
          'medium':"Interferensi sedang (-80 dBm)",
          'high':"Interferensi tinggi (-70 dBm)"}
    return create_figure(hm, param), lbl.get(noise_level, "")

# -- 4b. Switch mode kualitas: ML vs Teori -----

@app.callback(
    Output('graph', 'figure', allow_duplicate=True),
    Output('output', 'children', allow_duplicate=True),
    Input('kualitas_mode', 'value'),
    State('toggle_heatmap', 'value'),
    State('parameter', 'value'),
    prevent_initial_call=True,
)
def switch_kualitas_mode(mode, hm, param):
    if not param: param = 'rssi'
    state.kualitas_mode = mode or 'ml'
    label = "Machine Learning" if state.kualitas_mode == 'ml'
    else "Teori (Log-Distance)"
    return create_figure(hm, param), f"Mode tampilan kualitas: {label}"

# -- 4d. Slider radius heatmap ML -----

@app.callback(
    Output('graph', 'figure', allow_duplicate=True),
    Output('output', 'children', allow_duplicate=True),
    Input('heatmap-radius-slider', 'value'),
    State('toggle_heatmap', 'value'),
    State('parameter', 'value'),
    prevent_initial_call=True,
)
def update_heatmap_radius(radius, hm, param):
    if not param: param = 'rssi'
    state.heatmap_radius = float(radius) if radius else 10.0
```



© 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 create_figure(hm, param), f"Radius heatmap ML:
{radius} meter"

# -- 4e. Kunci / buka kunci heatmap ML -----

@app.callback(
    Output('graph',          'figure',    allow_duplicate=True),
    Output('output',         'children',  allow_duplicate=True),
    Output('btn-lock-heatmap', 'children'),
    Output('btn-lock-heatmap', 'style'),
    Output('lock-status-text', 'children'),
    Input('btn-lock-heatmap', 'n_clicks'),
    State('toggle_heatmap', 'value'),
    State('parameter',       'value'),
    prevent_initial_call=True,
)

def toggle_lock_heatmap(n_clicks, hm, param):
    if not param: param = 'rss'

    if state.heatmap_locked:
        state.heatmap_locked      = False
        state.heatmap_locked_grid = None
        label = "🔒 Kunci Heatmap ML"
        style =
{"marginTop": "8px", "width": "100%", "backgroundColor": "#37474F",
  "color": "white", "border": "none", "fontWeight": "b
old", "fontSize": "11px"}
        status = "Heatmap: tidak dikunci"
        pesan = "Kunci heatmap dilepas – heatmap akan diperbarui
setiap ada Rx baru."
    else:
        if state.kualitas_mode == 'ml' and state.rx_data:
            grid_x = np.linspace(0, state.img_width, 60)
            grid_y = np.linspace(0, state.img_height, 60)
            gx, gy = np.meshgrid(grid_x, grid_y)
            from visualization import buat_heatmap_interpolasi_ml
            labels, cukup = buat_heatmap_interpolasi_ml(gx, gy)
            if cukup:
                state.heatmap_locked      = True
                state.heatmap_locked_grid = labels
                label = "🔒 Buka Kunci Heatmap"
                style =
{"marginTop": "8px", "width": "100%", "backgroundColor": "#1565C0",
  "color": "white", "border": "none", "fontWe
ight": "bold", "fontSize": "11px"}
                status = "Heatmap: 🔒 TERKUNCI"
                pesan = "Heatmap ML dikunci – tidak akan berubah
meski ada Rx baru."
            else:
                label = "🔒 Kunci Heatmap ML"
                style =
{"marginTop": "8px", "width": "100%", "backgroundColor": "#37474F",
  "color": "white", "border": "none", "fontWe
ight": "bold", "fontSize": "11px"}
                status = "Heatmap: tidak dikunci"
                pesan = "⚠️ Perlu minimal 3 titik Rx dalam mode
ML untuk mengunci heatmap."
            else:

```



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

```

label = "🔒 Kunci Heatmap ML"
style =
{"marginTop":"8px","width":"100%","backgroundColor":"#37474F",
  "color":"white","border":"none","fontWeight":
  "bold","fontSize":"11px"}
status = "Heatmap: tidak dikunci"
pesan = "⚠️ Aktifkan mode ML dan tambahkan minimal
3 Rx sebelum mengunci."

return create_figure(hm, param), pesan, label, style, status

@app.callback(
    Output('graph', 'figure', allow_duplicate=True),
    Output('output', 'children', allow_duplicate=True),
    Input('btn-optimal-ml', 'n_clicks'),
    State('toggle_heatmap', 'value'),
    State('parameter', 'value'),
    prevent_initial_call=True,
)
def optimal_ap_ml(n_clicks, hm, param):
    if not param: param = 'rss'

    if not ML_LOADED:
        return dash.no_update, "⚠️ Model RF belum dimuat."

    # Kalau belum ada AP, buat satu di tengah denah sebagai titik
    awal
    if not state.ap_x:
        state.ap_x.append(state.img_width / 2)
        state.ap_y.append(state.img_height / 2)
        dibuat_otomatis = True
    else:
        dibuat_otomatis = False

    import time
    t0 = time.time()

    skor_awal, skor_akhir, total, detail, mode_eval =
    cari_posisi_ap_optimal_ml()

    if isinstance(mode_eval, str) and mode_eval not in ("rx",
    "grid"):
        return dash.no_update, f"⚠️ {mode_eval}"

    update_all_rx()
    durasi = time.time() - t0

    baris = [
        f"AP {d['ap_idx']}: {d['pct_sbml']:.0f}% →
    {d['pct_ssdh']:.0f}% Baik"
        for d in detail
    ]

    pct_awal = skor_awal / total * 100
    pct_akhir = skor_akhir / total * 100

    sumber = f"{total} titik Rx yang dipasang" if mode_eval ==
    "rx" \

```



© 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

```

else f"{total} titik grid internal"

prefix = "AP dibuat otomatis dan " if dibuat_otomatis else ""

pesan = (
    f"📶 {prefix}Optimasi selesai ({len(detail)} AP,
{durasi:.1f} detik). "
    f"Evaluasi dari: {sumber}. "
    f"Kualitas Baik: {pct_awal:.0f}% → {pct_akhir:.0f}%. "
    f"| " + " | ".join(baris)
)

return create_figure(hm, param), pesan

# -- 5. EDIT WALL: toggle sumber power existing AP -----

@app.callback(
    Output('ext-ap-measured-input', 'style'),
    Output('ext-ap-rssi-inputs', 'children'),
    Input('ext-ap-power-source', 'value'),
    prevent_initial_call=True,
)
def toggle_ext_src(source):
    state.ext_ap_power_source = source
    if source != 'terukur' or not state.ext_ap_x:
        return {'display':'none'}, []
    inputs = []
    for i in range(len(state.ext_ap_x)):
        stored = state.ext_ap_measured[i] if i <
len(state.ext_ap_measured) else None
        inputs.append(html.Div([
            html.Label(f"Ext AP {i+1}:", className="input-label",
                style={"fontSize":"11px","marginBottom":"2
px"}),
            dcc.Input(id={'type':'ext-rssi-input','index':i},
type='number', max=0,
                placeholder="cth: -55", value=stored,
className="ctrl-input",
                style={"marginBottom":"4px"}),
        ]))
    return {'display':'block'}, inputs

@app.callback(
    Output('output', 'children', allow_duplicate=True),
    Input('btn-save-measured-rssi', 'n_clicks'),
    State({'type':'ext-rssi-input','index':dash.ALL}, 'value'),
    prevent_initial_call=True,
)
def save_measured_rssi(_, vals):
    state.ext_ap_measured = list(vals)
    update_all_rx()
    return f"RSSI terukur disimpan: {sum(1 for v in vals if
v)}/{len(vals)} AP."

# -- 6. Tombol Analisis Existing AP -----

@app.callback(
    Output('existing-ap-result', 'children'),

```



© 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

```

Output('graph', 'figure', allow_duplicate=True),
Output('output', 'children', allow_duplicate=True),
Input('btn-analyze-existing', 'n_clicks'),
State('toggle_heatmap', 'value'),
State('parameter', 'value'),
prevent_initial_call=True,
)

def analyze_existing(_, hm, param):
    if not param: param = 'rssi'
    if not state.ext_ap_x:
        return html.P("Tambahkan Existing AP
dulu.", style={"color": "orange"}), \
        dash.no_update, "Belum ada Existing AP."
    if not state.rx_data:
        return html.P("Tambahkan Receiver
dulu.", style={"color": "orange"}), \
        dash.no_update, "Belum ada Receiver."

    update_all_rx()
    noise_lbl = {-90: "Tidak ada", -80: "Sedang", -
70: "Tinggi"}.get(state.noise_floor, "")
    mode_aktif = getattr(state, 'kualitas_mode', 'ml')
    kunci_kualitas = 'kualitas_ml' if mode_aktif == 'ml' else
'kualitas_teoris'
    rows = []
    rssi_vals, snr_vals = [], []
    kc = {"Baik": 0, "Sedang": 0, "Buruk": 0, "Blind Spot": 0}

    for i, d in enumerate(state.rx_data):
        p = d[2]; rssi=p.get("rssi"); snr=p.get("snr");
kml=p.get(kunci_kualitas, "-")
        if rssi: rssi_vals.append(rssi)
        if snr: snr_vals.append(snr)
        if kml in kc: kc[kml]+=1
        warna = get_warna_from_label(kml)
        rows.append(html.Tr([
            html.Td(f"Rx {i+1}", style={"fontWeight": "bold"}),
            html.Td(f"{rssi:.1f} dBm" if rssi else "-"),
            html.Td(f"{snr:.1f} dB" if snr else "-"),
            html.Td(f"{d[3]:.1f} m" if d[3] else "-"),
            html.Td(kml,
style={"color": warna, "fontWeight": "bold"}),
        ]))

    avg_r = statistics.mean(rssi_vals) if rssi_vals else None
    avg_s = statistics.mean(snr_vals) if snr_vals else None
    total = len(state.rx_data)
    bp = kc['Baik']/total*100 if total else 0

    result = html.Div([
        html.H4("Hasil Analisis Existing
AP", style={"fontSize": "13px", "marginBottom": "6px"}),
        html.Div(f"Existing AP: {len(state.ext_ap_x)} | Sumber:
{state.ext_ap_power_source} "
f"| Interferensi: {noise_lbl} | Mode kualitas: "
f"'ML' if mode_aktif=='ml' else 'Teori'"),
        style={"fontSize": "11px", "color": "#546e7a", "marg
inBottom": "6px"}),
        html.Table(

```



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

```

[html.Tr([html.Th(c) for c in
["Rx","RSSI","SNR","Jarak","Kualitas"]])] + rows,
style={"borderCollapse":"collapse","width":"100%","font
ntSize":"11px"},
),
html.Hr(style={"margin":"8px 0"}),
html.Div([html.B("Rata-rata: "),
html.Span(f"RSSI {avg_r:.1f} dBm | SNR
{avg_s:.1f} dB"
if avg_r else "--")],
style={"fontSize":"11px","marginBottom":"4px"}),
html.Div([html.Span(f"Baik
{kc['Baik']} ",style={"color":"green","fontSize":"11px"}),
html.Span(f"Sedang
{kc['Sedang']} ",style={"color":"orange","fontSize":"11px"}),
html.Span(f"Buruk
{kc['Buruk']} ",style={"color":"red","fontSize":"11px"}),
html.Span(f"Blind Spot {kc['Blind
Spot']} ",style={"color":"gray","fontSize":"11px"})]),
html.P(f"Coverage 'Baik': {bp:.0f}% dari {total} titik",
style={"fontWeight":"bold","fontSize":"12px","marg
inTop":"6px",
"color":"green" if bp>=70 else "orange"}),
])
return result, create_figure(hm,param), f"Analisis selesai:
{total} receiver."

# -- 8. Callback Utama: klik grafik -----

@app.callback(
    Output('graph', 'figure'),
    Output('output', 'children'),
    Output('wall-info-panel', 'children'),
    Output('wall-info-panel', 'style'),
    Input('graph', 'clickData'),
    Input('reset-btn', 'n_clicks'),
    Input('tx_power', 'value'),
    Input('parameter', 'value'),
    State('mode', 'value'),
    State('toggle_heatmap', 'value'),
    State('material', 'value'),
    State('wall-full-selector', 'value'),
    prevent_initial_call=True,
)
def update_graph(clickData, n_reset, tx_power_val, param,
mode, hm, material, wall_full_val):
    NO_PANEL = dash.no_update, dash.no_update

    state.tx_power = tx_power_val or 15
    if not param: param = 'rss'
    tid = ctx.triggered_id

    if tid == 'parameter':
        return create_figure(hm,param), f"Menampilkan
{param.upper()}", *NO_PANEL

    if tid == 'reset-btn':
        # Hapus semua list data

```



© 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

```

        for lst in [state.ap_x, state.ap_y, state.ext_ap_x,
                    state.ext_ap_y,
                    state.rx_x, state.rx_y, state.rx_data,
                    state.scale_points,
                    state.walls, state.auto_walls,
                    state.temp_wall,
                    state.ext_ap_measured]:
            try: lst.clear()
            except: pass

        # Hapus dead zones
        state.dead_zones.clear()
        state.dead_zone_tmp = None

        # Reset heatmap lock
        state.heatmap_locked = False
        state.heatmap_locked_grid = None

        # Hapus gambar denah dan reset dimensi
        state.uploaded_image = None
        state.img_width = 800
        state.img_height = 600
        state.scale_meter = 0.05

        # Reset state lain
        state.selected_wall_idx = None
        state.selected_wall_source = None

        return create_figure(hm, param), "Semua data dihapus.
        Silakan upload denah baru.", *NO_PANEL

    if tid == 'tx_power':
        update_all_rx()
        return create_figure(hm,param), f"Tx Power:
        {state.tx_power} dBm", *NO_PANEL

    if not clickData:
        return create_figure(hm,param), "Klik grafik untuk
        mulai.", *NO_PANEL

    x = clickData['points'][0]['x']
    y = clickData['points'][0]['y']
    info = ""

    if mode == 'edit_wall':
        # Cari tembok terdekat dari titik klik
        def dts(px,py,x1,y1,x2,y2):
            ddx,ddy = x2-x1,y2-y1
            if ddx==ddy==0: return np.sqrt((px-x1)**2+(py-y1)**2)
            tt = max(0,min(1,((px-x1)*ddx+(py-
            y1)*ddy)/(ddx*ddx+ddy*ddy)))
            return np.sqrt((px-x1-tt*ddx)**2+(py-y1-tt*ddy)**2)

        bd, bwi, bsrc = float('inf'), None, None
        for wi,w in enumerate(state.walls):
            d = dts(x,y,w[0][0],w[0][1],w[1][0],w[1][1])
            if d < bd: bd,bwi,bsrc = d,wi,'manual'
        if state.use_auto_walls:
            for wi,w in enumerate(state.auto_walls):

```



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

```

d = dts(x,y,w[0][0],w[0][1],w[1][0],w[1][1])
if d < bd: bd,bwi,bsrc = d,wi,'auto'

if bd > 20 or bwi is None:
    return (create_figure(hm,param),
            "Tidak ada tembok di dekat klik.",
            html.P("Tidak ada tembok di dekat klik.",
                    style={"color":"gray","fontSize":"12px"})),

    {"display":"block","padding":"10px","border":
"1px solid #ccc",
    "borderRadius":"8px","marginTop":"8px"})

wls = state.auto_walls if bsrc=='auto' else state.walls
ww = wls[bwi]
mat = ww[2]
src = ww[3] if len(ww)>3 else 'manual'
conf = ww[4] if len(ww)>4 else 1.0

state.selected_wall_idx = bwi
state.selected_wall_source = bsrc

src_txt = SRC_LABEL.get(src, src)
src_color = SRC_COLOR.get(src, 'gray')
conf_pct = f"{conf*100:.0f}%" if isinstance(conf, float)
else f"{conf}px"

panel_content = html.Div([
    html.B("Tembok dipilih:", style={"fontSize":"12px"}),
    html.Div([
        html.Span("Material: ",
            style={"fontSize":"11px"}),
        html.Span(mat, style={"fontWeight":"bold",
            "color":warna_material(mat)
            ,"fontSize":"11px"}),
    ], style={"marginTop":"4px"}),
    html.Div([
        html.Span("Deteksi via: ",
            style={"fontSize":"11px"}),
        html.Span(src_txt,
            style={"color":src_color,"fontSize":"11px",
            "fontWeight":"bold"}),
        html.Span(f" ({conf_pct})",
            style={"color":"#90a4ae","fontSize":"10px"}),
    ]),
    html.Hr(style={"margin":"6px 0"}),
    html.B("Ganti material:", style={"fontSize":"11px"}),
    dcc.RadioItems(
        id='wall-material-picker',
        options=MAT_OPTIONS,
        value=mat,
        className="radio-group-vertical",
        style={"fontSize":"11px","marginTop":"4px"},
    ),
    html.Button("[OK] Terapkan", id="btn-apply-wall-mat",
n_clicks=0,
        className="ctrl-btn ctrl-btn-green",
        style={"marginTop":"6px","width":"100%","
fontSize":"12px"}),

```



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

```

html.Button("X Batal pilih", id="btn-cancel-wall",
n_clicks=0,
className="ctrl-btn",
style={"marginTop":"4px","width":"100%","
fontSize":"12px"}),
])
panel_style = {
"display":"block","padding":"10px",
"border":"2px solid #FFD700","borderRadius":"8px",
"backgroundColor":"#ffffde7","marginTop":"8px",
}
# Highlight tembok terpilih di peta (warna kuning)
return (create_figure(hm,param),
f"Tembok '{mat}' dipilih. Pilih material baru
lalu klik Terapkan.",
panel_content,
panel_style)

elif mode == 'ap':
state.ap_x.append(x); state.ap_y.append(y)
update_all_rx()
info = f"AP {len(state.ap_x)} ditambahkan."

elif mode == 'ext_ap':
state.ext_ap_x.append(x); state.ext_ap_y.append(y)
state.ext_ap_measured.append(None)
update_all_rx()
info = f"Existing AP {len(state.ext_ap_x)} ditambahkan."

elif mode == 'rx':
params = get_parameters_with_ml(x,y)
jarak = hitung_jarak(x,y)
terhub = get_connected_ap(x,y)
if params['rssi'] is not None:
state.rx_x.append(x); state.rx_y.append(y)
state.rx_data.append((x,y,params,jarak,terhub))
mode_aktif = getattr(state,'kualitas_mode','ml')
kunci_kualitas = 'kualitas_ml' if mode_aktif == 'ml'
else 'kualitas_teoris'
tag = "[ML]" if mode_aktif == 'ml' else "[Teoris]"
info = (f"Rx {len(state.rx_data)}:
RSSI={params['rssi']:.1f} | "
f"SNR={params['snr']:.1f} | {tag}:
{params.get(kunci_kualitas, params['kualitas_ml'])}")
else:
info = "Belum ada AP!"

elif mode == 'scale':
state.scale_points.append((x,y))
if len(state.scale_points) > 2: state.scale_points.pop(0)
info = f"Titik skala {len(state.scale_points)}/2"

elif mode == 'delete':
thr = 20
for lx,ly,lbl in [(state.ap_x,state.ap_y,'AP'),
(state.ext_ap_x,state.ext_ap_y,'Existi
ng AP')]:
for i,(ax,ay) in enumerate(zip(lx,ly)):
if np.sqrt((x-ax)**2+(y-ay)**2) < thr:

```



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

```

        lx.pop(i); ly.pop(i)
        if lbl=='Existing AP' and
i<len(state.ext_ap_measured):
            state.ext_ap_measured.pop(i)
            update_all_rx()
            return create_figure(hm,param), f"{lbl}
dihapus.", *NO_PANEL
        for i,d in enumerate(state.rx_data):
            if np.sqrt((x-d[0])**2+(y-d[1])**2) < thr:
                state.rx_data.pop(i)
                return create_figure(hm,param), "Receiver
dihapus.", *NO_PANEL
                info = "Tidak ada titik terdekat."

        elif mode == 'move':
            thr = 20
            if state.selected_point is None:
                for i,(ax,ay) in
enumerate(zip(state.ap_x,state.ap_y)):
                    if np.sqrt((x-ax)**2+(y-ay)**2)<thr:
                        state.selected_point,state.selected_type =
i,'ap'
                        return create_figure(hm,param),"AP dipilih --
klik tujuan.", *NO_PANEL
                        for i,d in enumerate(state.rx_data):
                            if np.sqrt((x-d[0])**2+(y-d[1])**2)<thr:
                                state.selected_point,state.selected_type =
i,'rx'
                                return create_figure(hm,param),"Rx dipilih --
klik tujuan.", *NO_PANEL
                                for i,(ax,ay) in
enumerate(zip(state.ext_ap_x,state.ext_ap_y)):
                                    if np.sqrt((x-ax)**2+(y-ay)**2)<thr:
                                        state.selected_point,state.selected_type =
i,'ext_ap'
                                        return create_figure(hm,param),"Existing AP
dipilih -- klik tujuan.", *NO_PANEL
                                        return create_figure(hm,param),"Tidak ada titik
terdekat.", *NO_PANEL
                                else:
                                    if state.selected_type=='ap':
                                        state.ap_x[state.selected_point]=x;
                                        state.ap_y[state.selected_point]=y
                                        update_all_rx(); info=f"AP dipindahkan."
                                    elif state.selected_type=='rx':
                                        p=get_parameters_with_ml(x,y);
                                        j=hitung_jarak(x,y)
                                        state.rx_data[state.selected_point]=(x,y,p,j,get_
connected_ap(x,y))
                                        info="Rx dipindahkan."
                                    elif state.selected_type=='ext_ap':
                                        state.ext_ap_x[state.selected_point]=x;
                                        state.ext_ap_y[state.selected_point]=y
                                        update_all_rx(); info="Existing AP dipindahkan."
                                        state.selected_point=state.selected_type=None

        elif mode == 'del_wall':
            def dts(px,py,x1,y1,x2,y2):
                dx,dy=x2-x1,y2-y1

```



© 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

```

        if dx==dy==0: return np.sqrt((px-x1)**2+(py-y1)**2)
        t=max(0,min(1, ((px-x1)*dx+(py-y1)*dy)/(dx*dx+dy*dy)))
        return np.sqrt((px-x1-t*dx)**2+(py-y1-t*dy)**2)
    bd,bs,bi=float('inf'),None,None
    for i,w in enumerate(state.walls):
        d=pts(x,y,w[0][0],w[0][1],w[1][0],w[1][1])
        if d<bd: bd,bs,bi=d,'walls',i
    for i,w in enumerate(state.auto_walls):
        d=pts(x,y,w[0][0],w[0][1],w[1][0],w[1][1])
        if d<bd: bd,bs,bi=d,'auto_walls',i
    if bd<=15 and bs:
        deleted=(state.walls if bs=='walls' else
state.auto_walls).pop(bi)
        update_all_rx(); info=f"Garis {deleted[2]} dihapus."
        else: info="Tidak ada garis terdekat."

    elif mode == 'obs':
        state.temp_wall.append((x,y))
        if len(state.temp_wall)==2:
            wf = int(wall_full_val) if wall_full_val is not None
else 1
            state.walls.append((state.temp_wall[0],state.temp_wal
l[1],material,'manual',1.0,wf))
            state.temp_wall.clear(); update_all_rx()
            info=f"Tembok {material} dibuat."
        else: info="Klik titik kedua untuk menyelesaikan."

    elif mode == 'dead_zone':
        if state.dead_zone_tmp is None:
            state.dead_zone_tmp = (x, y)
            info = "❌ Klik titik kedua (sudut berlawanan) untuk
menyelesaikan Dead Zone."
        else:
            x1, y1 = state.dead_zone_tmp
            state.dead_zones.append(((x1,y1),(x,y)))
            state.dead_zone_tmp = None
            update_all_rx()
            w_m = abs(x-x1) * state.scale_meter
            h_m = abs(y-y1) * state.scale_meter
            info = (f"❌ Dead Zone dibuat ({w_m:.1f}x{h_m:.1f}
m). "
                    f"Redaman tambahan: {35} dB per jalur yang
melewatinya.")

    elif mode == 'del_dead_zone':
        from functions import point_in_rect
        deleted = False
        for i, ((x1,y1),(x2,y2)) in enumerate(state.dead_zones):
            if point_in_rect(x, y, x1, y1, x2, y2):
                state.dead_zones.pop(i)
                update_all_rx()
                info = "🗑 Dead Zone dihapus."
                deleted = True
                break
        if not deleted:
            info = "Klik di dalam area Dead Zone untuk
menghapusnya."

```



© 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 create_figure(hm,param), info, *NO_PANEL

# -- 9. Tabel Receiver -----

@app.callback(
    Output('table-data','children'),
    Input('graph','figure'),
    State('parameter','value'),
)
def update_table(_,param):
    if not param: param='rssi'
    if not state.rx_data: return "Belum ada data receiver."
    titles={'rssi':'RSSI (dBm)','snr':'SNR (dB)',
            'throughput':'Throughput (Mbps)','path_loss':'Path
Loss (dB)'}
    mode_aktif = getattr(state, 'kualitas_mode', 'ml')
    kunci_kualitas = 'kualitas_ml' if mode_aktif == 'ml' else
'kualitas_teoris'
    ml_lbl = "Kualitas (ML)" if mode_aktif == 'ml' else "Kualitas
(Teoris)"
    nl = {-90:"Normal",-80:"Interferensi Sedang",-
70:"Interferensi Tinggi"
        }.get(getattr(state,'noise_floor',-90),"")

    header = html.Tr([html.Th(c) for c in
                        ["No","X","Y",titles.get(param,"Val"),
                        "Jarak (m)","Tembok","Terhubung",ml_lbl]])
    rows=[]
    for i,d in enumerate(state.rx_data):
        p=d[2]; j=d[3]; th=d[4] if len(d)>4 else "-"
        kml=p.get(kunci_kualitas) if isinstance(p,dict) else None
        if not kml: kml=get_kualitas(p.get('rssi',-100),'rssi')
        wc=p.get("wall_count","-") if isinstance(p,dict) else "-"
        val=p.get(param) if isinstance(p,dict) else None
        rows.append(html.Tr([
            html.Td(f"Rx {i+1}"),
            html.Td(f"{d[0]:.0f}"),html.Td(f"{d[1]:.0f}"),
            html.Td(f"{val:.1f}" if val is not None else "-"),
            html.Td(f"{j:.1f}" if j else "-"),
            html.Td(str(wc)),
            html.Td(th or "-",style={"color":"purple" if th and
"Ext" in str(th) else "darkred",
                                "fontWeight":"bold"}),
            html.Td(kml,style={"color":get_warna_from_label(kml),
"fontWeight":"bold"}),
        ]))

    note=html.P(f"Kondisi: {nl} (noise
{getattr(state,'noise_floor',-90)} dBm)",
                style={"fontSize":"11px","color":"orange","margin
":"4px 0 0"}) if nl else None
    return html.Div([html.Table([header]+rows,
                                style={"border":"2px solid
black","borderCollapse":"collapse","width":"100%"}),note])

# -- 10. Scan Wi-Fi -----

@app.callback(
    Output('wifi-info','children'),

```



© 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

```

Input('scan-wifi-btn','n_clicks'),
prevent_initial_call=True,
)
def scan_wifi(n):
    if n:
        info=get_existing_wifi()
        c={'Baik':'green','Sedang':'orange','Buruk':'red'}.get(in
fo['quality'],'black')
        return html.Div([
            html.P(html.B(f"SSID: {info['ssid']}")),
            html.P(f"RSSI: {info['rssi']} dBm"),
            html.P(f"SNR: {info['snr']} dB"),
            html.P(f"Channel: {info['channel']}"),
            html.P(["Kualitas: ",html.Span(info['quality'],
style={"color":c,"fontWeight":"bold"})]),
        ])
    return dash.no_update

# -- 11. Tabel pengukuran manual -----

@app.callback(
    Output('meas-table','children'),
    Output('input-jarak','value'),
    Output('input-rssi','value'),
    Output('input-label','value'),
    Input('btn-add-meas','n_clicks'),
    Input('btn-clear-meas','n_clicks'),
    State('input-jarak','value'),
    State('input-rssi','value'),
    State('input-label','value'),
    prevent_initial_call=True,
)
def meas_table(na,nc,jarak,rssi_v,label):
    if ctx.triggered_id=='btn-clear-meas':
        state.wifi_measurements.clear()
        return html.P("Belum ada
data.",style={'color':'gray'}),None,None,None
    if ctx.triggered_id=='btn-add-meas':
        if jarak is None or rssi_v is None:
            return
dash.no_update,dash.no_update,dash.no_update,dash.no_update
        state.wifi_measurements.append((float(jarak),float(rssi_v
),label.strip() if label else "-"))
        if not state.wifi_measurements:
            return html.P("Belum ada
data.",style={'color':'gray'}),None,None,None
        def rc(v): return 'green' if v>=-65 else 'orange' if v>=-80
else 'red')
        sm=sorted(state.wifi_measurements,key=lambda m:m[0])
        rows=[html.Tr([html.Td(f"{m[0]} m"),
            html.Td(f"{m[1]}
dBm",style={'color':rc(m[1]),'fontWeight':'bold'}),
            html.Td(m[2])]) for m in sm]
        n,f=sm[0],sm[-1]
        sum_div=html.Div([html.Hr(),html.B("Perbandingan:"),
            html.Table([
                html.Tr([html.Td("Terdekat"),html.Td(f"{n[0]} m"),
                    html.Td(html.Span(f"{n[1]}
dBm",style={'color':rc(n[1]),'fontWeight':'bold'}))]),

```



© 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

```

        html.Tr([html.Td("Terjauh"), html.Td(f"{f[0]} m"),
                  html.Td(html.Span(f"{f[1]} dBm", style={'color':rc(f[1]),'fontWeight':'bold'})))]),
        html.Tr([html.Td("Delta"),html.Td(""),
                  html.Td(html.Span(f"{n[1]-f[1]:.1f} dBm", style={'fontWeight':'bold'})))]),
        ], style={'marginTop':'6px','width':'100%'}))
    header_row = html.Tr([html.Th(h) for h in ["Jarak", "RSSI", "Keterangan"]])
    return html.Div([
        html.Table(
            [header_row] + rows,
            style={'borderCollapse': 'collapse', 'width':
'100%'},
        ),
        sum_div,
    ]), None, None, None

# -- 12. Apply & Cancel wall material edit -----

@app.callback(
    Output('graph',          'figure',    allow_duplicate=True),
    Output('output',         'children',  allow_duplicate=True),
    Output('wall-info-panel', 'children',  allow_duplicate=True),
    Output('wall-info-panel', 'style',     allow_duplicate=True),
    Input('btn-apply-wall-mat', 'n_clicks'),
    State('wall-material-picker', 'value'),
    State('toggle_heatmap',      'value'),
    State('parameter',           'value'),
    prevent_initial_call=True,
)
def apply_wall_material(n, new_mat, hm, param):
    if not n: return dash.no_update, dash.no_update,
dash.no_update, dash.no_update
    if not param: param = 'rssi'
    idx = getattr(state, 'selected_wall_idx', None)
    src = getattr(state, 'selected_wall_source', None)
    if idx is None:
        return dash.no_update, "Pilih tembok dulu.",
dash.no_update, dash.no_update

    ok = change_wall_material(idx, new_mat, wall_source=src)
    state.selected_wall_idx = None
    state.selected_wall_source = None
    update_all_rx()

    if ok:
        return (create_figure(hm, param),
                f"Material tembok diubah ke: {new_mat}",
                "", {"display": "none"})
    return dash.no_update, "Gagal mengubah material.", "",
{"display": "none"}

@app.callback(
    Output('graph',          'figure',    allow_duplicate=True),
    Output('wall-info-panel', 'children',  allow_duplicate=True),
    Output('wall-info-panel', 'style',     allow_duplicate=True),
    Input('btn-cancel-wall', 'n_clicks'),
    State('toggle_heatmap',  'value'),

```



© 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

```

State('parameter', 'value'),
prevent_initial_call=True,
)
def cancel_wall_selection(n, hm, param):
    if not n: return dash.no_update, dash.no_update,
dash.no_update
    state.selected_wall_idx = None
    state.selected_wall_source = None
    if not param: param = 'rssi'
    return create_figure(hm, param), "", {"display": "none"}

#
#
# Ubah semua tembok ke satu material / reset otomatis
#

@app.callback(
    Output('graph', 'figure', allow_duplicate=True),
    Output('output', 'children', allow_duplicate=True),
    Input('btn-bulk-material', 'n_clicks'),
    Input('btn-reset-auto-walls', 'n_clicks'),
    State('bulk-material-picker', 'value'),
    State('toggle_heatmap', 'value'),
    State('parameter', 'value'),
    prevent_initial_call=True,
)
def bulk_change_material(n_bulk, n_reset, new_mat, hm, param):
    if not param: param = 'rssi'
    tid = ctx.triggered_id

    if tid == 'btn-reset-auto-walls':
        if state.uploaded_image:
            from vision import auto_scan_walls
            auto_scan_walls(state.uploaded_image)
            update_all_rx()
            msg = f"Tembok otomatis di-reset:
{len(state.auto_walls)} garis terdeteksi ulang."
        else:
            msg = "Belum ada gambar denah yang diunggah."
        return create_figure(hm, param), msg

    if tid == 'btn-bulk-material':
        if not new_mat:
            return dash.no_update, "Pilih material terlebih
dahulu!"
        total = 0
        for i, w in enumerate(state.walls):
            state.walls[i] = (w[0], w[1], new_mat, 'manual', 1.0)
            total += 1
        for i, w in enumerate(state.auto_walls):
            state.auto_walls[i] = (w[0], w[1], new_mat, 'manual',
1.0)
            total += 1
        update_all_rx()

```



© 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

```

        label =
        {'beton':'Beton','bata':'Bata','kaca':'Kaca','kayu':'Kayu'}.get(new_mat, new_mat)
        return create_figure(hm, param), f"Semua {total} tembok diubah ke: {label}."

    return dash.no_update, dash.no_update

#
=====
# BANTUAN – Penjelasan lengkap semua fitur SimNet
#
=====

BANTUAN_KONTEN = [
    ("📁 Upload Gambar Denah",
     "Unggah gambar denah bangunan (PNG/JPG) sebagai bidang kerja simulasi. "
     "Setelah diunggah, gambar menjadi latar peta tempat AP, Rx, dan tembok digambar. "
     "Pastikan gambar jelas dan memiliki skala yang dapat dikenali."),
    ("🧱 Deteksi Tembok Otomatis",
     "Saat ON, aplikasi memindai gambar denah dan mendeteksi garis tembok secara otomatis "
     "berdasarkan warna piksel (prioritas) dan ketebalan garis (fallback). "
     "Material terdeteksi: beton (merah), bata (coklat), kaca (biru), kayu (hijau). "
     "Saat OFF, tembok digambar manual menggunakan mode Obstacle."),
    ("⚙️ Mode Kontrol",
     "Menentukan aksi ketika pengguna mengklik peta. "
     "• Access Point: tempatkan titik AP (sumber sinyal). "
     "• Receiver (Rx): tempatkan titik penerima, langsung dihitung kualitas sinyalnya. "
     "• Obstacle: gambar tembok baru (klik dua titik). "
     "• Set Skala: tentukan jarak nyata (meter) dari dua titik di peta. "
     "• Hapus Titik: klik titik AP/Rx untuk menghapusnya. "
     "• Hapus Garis: klik tembok untuk menghapusnya. "
     "• Edit Material: klik tembok untuk ubah materialnya. "
     "• Geser Titik: klik dan drag AP/Rx ke posisi baru. "
     "• Existing AP: tempatkan AP yang sudah ada di gedung (pakai RSSI terukur)."),
    ("📏 Set Skala",
     "Kalibrasi skala peta agar jarak piksel pada denah sesuai dengan jarak meter nyata. "
     "Cara: pilih mode Set Skala → klik dua titik di peta yang jaraknya diketahui → "
     "masukkan jarak nyata (meter) → klik Hitung Skala. "
     "Skala yang tepat sangat penting agar perhitungan jarak dan RSSI akurat."),

```



© 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

```
( Tampilan Heatmap",
  "Mengaktifkan/menonaktifkan overlay heatmap sinyal di atas
  denah. "
  "• Mode Teori: warna kontinu berdasarkan estimasi RSSI dari
  rumus Log-Distance Path Loss. "
  "• Mode ML: warna diskrit (Baik/Sedang/Buruk) berdasarkan
  interpolasi prediksi "
  "Random Forest dari titik-titik Rx yang sudah dipasang
  (butuh minimal 3 titik Rx). "
  "Hijau = Baik ( $\geq -65$  dBm), Kuning = Sedang ( $-65$  s.d.  $-80$ 
  dBm), Merah = Buruk ( $< -80$  dBm)."),

( Mode Klasifikasi Kualitas",
  "Menentukan sumber label kualitas sinyal yang ditampilkan di
  warna titik Rx dan tabel. "
  "• Machine Learning: prediksi dari model Random Forest
  berdasarkan kondisi fisik. "
  "• Teori: klasifikasi langsung dari threshold RSSI standar
  IEEE 802.11-2020. "
  "Kedua nilai dihitung setiap saat – toggle ini hanya
  mengubah mana yang ditampilkan."),

( Parameter Tampilan",
  "Memilih parameter fisik yang divisualisasikan pada heatmap
  mode Teori. "
  "• RSSI (dBm): kekuatan sinyal yang diterima. "
  "• SNR (dB): rasio sinyal terhadap noise. "
  "• Throughput (Mbps): estimasi kecepatan data. "
  "• Path Loss (dB): total redaman sinyal dari AP ke titik
  penerima."),

( Jenis Obstacle & Tipe Tembok",
  "Menentukan material tembok yang digambar secara manual. "
  "Material menentukan besaran redaman sinyal: "
  "Beton ( $-15$  dB), Bata ( $-12$  dB), Kaca ( $-5$  dB), Kayu ( $-3$  dB).
  "
  "Tipe Tembok: Penuh (redaman penuh) vs Setengah/Partisi
  (redaman 50%). "
  "Sumber: ITU-R P.1238-10."),

( Existing AP – Analisis Lapangan",
  "Fitur untuk mengevaluasi AP yang sudah terpasang di gedung.
  "
  "• Sumber Teori: estimasi menggunakan Tx Power default (15
  dBm). "
  "• Sumber Terukur: masukkan RSSI hasil pengukuran WiFi
  Analyzer di jarak referensi "
  "(1 meter, tanpa penghalang) sebagai kalibrasi. Mode Terukur
  menghasilkan estimasi "
  "yang lebih akurat karena disesuaikan dengan karakteristik
  AP nyata."),

( Posisi AP Optimal (ML)",
  "Fitur optimasi penempatan AP berbasis Machine Learning. "
  "Algoritma mencoba 400 kandidat posisi AP di seluruh denah,
  menghitung prediksi "
```



© 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

```

"Random Forest untuk 625 titik grid di setiap kandidat,
kemudian memindahkan AP "
"ke posisi yang memaksimalkan luas area dengan kualitas
Baik. "
"Proses berlangsung 2-5 detik tergantung kompleksitas denah.
"
"Pastikan mode ML aktif dan sudah ada AP di peta sebelum
menggunakan fitur ini."),

("📍 Pasang Optimal (Log-Distance)",
"Penempatan AP otomatis berbasis rumus Log-Distance Path
Loss. "
"Berbeda dengan Posisi AP Optimal (ML), tombol ini
menghitung titik dengan "
"coverage terlemah dan menempatkan AP tambahan untuk
memperkuatnya. "
"Masukkan jumlah AP yang diinginkan, lalu klik tombol."),

("💾 Simpan Data / 📄 Export CSV",
"Simpan Data: menyimpan seluruh konfigurasi simulasi saat
ini (AP, Rx, tembok, skala) "
"ke database agar dapat dimuat kembali di sesi berikutnya. "
"Export CSV: mengunduh tabel Data Receiver sebagai file .csv
"
"berisi koordinat, RSSI, SNR, throughput, path loss, dan
kualitas sinyal setiap titik Rx."),

("📷 Export PNG",
"Mengunduh tampilan denah simulasi saat ini (beserta
heatmap, AP, Rx, dan tembok) "
"sebagai file gambar PNG beresolusi tinggi. "
"Berguna untuk dokumentasi dan laporan."),

("📊 Statistik",
"Menampilkan ringkasan statistik dari semua titik Rx yang
sudah dipasang: "
"rata-rata RSSI, distribusi kualitas
(Baik/Sedang/Buruk/Blind Spot), "
"rekomendasi channel, dan analisis coverage keseluruhan."),
]

@app.callback(
    Output('modal-content', 'children', allow_duplicate=True),
    Output('modal-overlay', 'style', allow_duplicate=True),
    Output('modal-box', 'style', allow_duplicate=True),
    Input('btn-help', 'n_clicks'),
    prevent_initial_call=True,
)

def tampil_bantuan(n_clicks):
    kartu = []
    for judul, isi in BANTUAN_KONTEN:
        kartu.append(html.Div(style={
            "background": "var(--bg-card)",
            "border": "1px solid var(--border)",
            "borderRadius": "8px",
            "padding": "12px 16px",
            "marginBottom": "10px",
        }, children=[

```



© 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

```

html.Div(judul, style={
    "fontWeight": "600", "fontSize": "13px",
    "color": "var(--primary-d)", "marginBottom":
"5px",
    }),
html.P(isi, style={
    "fontSize": "12px", "color": "var(--text-muted)",
    "margin": "0", "lineHeight": "1.6",
    }),
)))

konten = html.Div([
    html.H3(" ? Panduan Penggunaan SimNet",
        style={"marginBottom": "6px"}),
    html.P("Klik fitur di bawah untuk memahami cara
kerjanya.",
        style={"fontSize": "12px", "color": "var(--text-
hint)",
            "marginBottom": "14px"}),
    html.Div(kartu, style={"maxHeight": "520px", "overflowY":
"auto",
        "paddingRight": "4px"}),
])
return konten, {"display": "flex"}, {"maxWidth": "640px"}

@app.callback(
    Output('modal-overlay', 'style', allow_duplicate=True),
    Input('btn-close-modal', 'n_clicks'),
    prevent_initial_call=True,
)
def tutup_modal(n_clicks):
    return {"display": "none"}

# -----
# -----
# Workflow:
# 1. User upload Excel/CSV data pengukuran lapangan BARU (belum
pernah dipakai
# untuk training) lewat tombol "Upload Test" di navbar.
# 2. User klik "Bandingkan dengan Teori" untuk melihat:
# - Evaluasi model Random Forest yang SUDAH dilatih (tidak
dilatih ulang)
# terhadap data baru ini: accuracy, precision, recall, F1,
confusion matrix
# - Perbandingan dengan pendekatan threshold Teori (Log-
Distance, tanpa ML)
# - Referensi CV historis dari training awal (bukan dihitung
ulang dari
# data baru -- CV adalah teknik saat training, bukan saat
evaluasi data baru)
#
=====
=====

import os as _os

```



© 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

functions.py

```
# =====
# functions.py - v3
# Perubahan: wall tuple kini bisa 3 atau 5 elemen
#           (backward-compatible dengan walls manual lama)
# =====

import numpy as np
import state
import platform, subprocess, re
import sys
sys.stdout.reconfigure(encoding='utf-8')

from app import predict_kualitas_ml, ML_LOADED

PL_D0 = 50
N_LOG = 26

def loss_material(material):
    return {'beton':15, 'bata':12, 'kayu':3, 'kaca':5}.get(material,
0)

def warna_material(material):
    return
{'beton':'black', 'bata':'#c0392b', 'kaca':'cyan', 'kayu':'#8B4513'}
.get(material, 'gray')

def wall_material(w):
    """Ambil material dari tuple wall (3 atau lebih elemen)."""
    return w[2]

def wall_full_flag(w):
    """Ambil flag wall_full dari tuple wall. Default 1 (penuh)
jika tidak ada."""
    return int(w[5]) if len(w) > 5 else 1

def loss_for_wall(w):
    """
    Hitung redaman (dB) satu tembok, memperhitungkan tipe tembok:
    - Tembok penuh (wall_full=1): redaman normal sesuai material
    - Tembok setengah/partisi (wall_full=0): redaman × 0.5
    """
    base = loss_material(wall_material(w))
    faktor = 1.0 if wall_full_flag(w) == 1 else 0.5
    return base * faktor

def line_intersect(p1, p2, p3, p4):
    def ccw(a,b,c):
        return (c[1]-a[1])*(b[0]-a[0]) > (b[1]-a[1])*(c[0]-a[0])
    return ccw(p1,p3,p4) != ccw(p2,p3,p4) and
ccw(p1,p2,p3) != ccw(p1,p2,p4)

# — Dead Zone (ruangan basah/toilet)

DEAD_ZONE_LOSS_DB = 35 # redaman dB per dead zone yang dilewati
(20-40 dB literatur)

def point_in_rect(px, py, x1, y1, x2, y2):
```



© 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

```

"""Cek apakah titik (px,py) berada di dalam persegi
panjang."""
return min(x1,x2) <= px <= max(x1,x2) and min(y1,y2) <= py <=
max(y1,y2)

def line_melalui_dead_zone(ax, ay, bx, by):
    """
    Cek berapa dead zone yang dilalui garis dari (ax,ay) ke
    (bx,by).
    Setiap dead zone dihitung satu kali meski garis memotong
    beberapa sisi.
    """
    zones = getattr(state, 'dead_zones', [])
    count = 0
    for (x1,y1),(x2,y2) in zones:
        xmin, xmax = min(x1,x2), max(x1,x2)
        ymin, ymax = min(y1,y2), max(y1,y2)

        # Empat sisi persegi panjang
        sisi = [
            ((xmin,ymin),(xmax,ymin)), # bawah
            ((xmin,ymax),(xmax,ymax)), # atas
            ((xmin,ymin),(xmin,ymax)), # kiri
            ((xmax,ymin),(xmax,ymax)), # kanan
        ]

        intersect = any(
            line_intersect((ax,ay),(bx,by), s[0], s[1])
            for s in sisi
        )
        # Atau salah satu titik ada di dalam kotak
        inside = (point_in_rect(ax,ay,xmin,ymin,xmax,ymax) or
                  point_in_rect(bx,by,xmin,ymin,xmax,ymax))
        if intersect or inside:
            count += 1
    return count

def dead_zone_penalty_db(ax, ay, bx, by):
    """Total redaman tambahan (dB) akibat dead zone yang
    dilewati."""
    return line_melalui_dead_zone(ax, ay, bx, by) *
    DEAD_ZONE_LOSS_DB

def get_noise_floor():
    return getattr(state, 'noise_floor', -90)

def get_kualitas(val, p_name):
    if p_name == 'rssi':
        if val >= -65: return "Baik"
        elif val >= -80: return "Sedang"
        elif val >= -90: return "Buruk"
        else: return "Blind Spot"
    elif p_name == 'snr':
        if val >= 25: return "Baik"
        elif val >= 10: return "Sedang"
        elif val >= 0: return "Buruk"
        else: return "Blind Spot"
    elif p_name == 'throughput':
        if val >= 50: return "Baik"
        elif val >= 20: return "Sedang"

```



© 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

```

        elif val > 0: return "Buruk"
        else: return "Blind Spot"
    elif p_name == 'path_loss':
        if val <= 80: return "Baik"
        elif val <= 95: return "Sedang"
        elif val <= 105: return "Buruk"
        else: return "Blind Spot"
    return "Unknown"

def get_kualitas_ml(rssi, jarak_m, wall_count,
wall_material_kode=2, wall_full=1):
    if rssi < -90: return "Blind Spot"
    return predict_kualitas_ml(rssi, jarak_m, wall_count,
wall_material_kode, wall_full)

def get_warna(val, p_name):
    k = get_kualitas(val, p_name)
    return {"Baik": "green", "Sedang": "orange", "Buruk": "red", "Blind
Spot": "gray"}.get(k, "black")

def get_warna_from_label(label):
    return {"Baik": "green", "Sedang": "orange", "Buruk": "red", "Blind
Spot": "gray"}.get(label, "black")

def count_walls_between(ax, ay, rx, ry):
    active = state.walls + (state.auto_walls if
state.use_auto_walls else [])
    return sum(1 for w in active if
line_intersect((ax, ay), (rx, ry), w[0], w[1]))

def hitung_rssi(x, y):
    all_ap_x = state.ap_x + state.ext_ap_x
    all_ap_y = state.ap_y + state.ext_ap_y
    if not all_ap_x: return None

    active = state.walls + (state.auto_walls if
state.use_auto_walls else [])
    rssi_list = []
    n_ap = len(state.ap_x)

    for i, (ax, ay) in enumerate(zip(all_ap_x, all_ap_y)):
        d_m = max(np.sqrt((x-ax)**2+(y-ay)**2)*state.scale_meter,
1)

        if i >= n_ap and
getattr(state, 'ext_ap_power_source', 'teori') == 'terukur':
            ei = i - n_ap
            meas = getattr(state, 'ext_ap_measured', [])
            if ei < len(meas) and meas[ei] is not None:
                rssi = meas[ei] - N_LOG*np.log10(d_m)
            else:
                rssi = state.tx_power - PL_D0 -
N_LOG*np.log10(d_m)
            else:
                rssi = state.tx_power - PL_D0 - N_LOG*np.log10(d_m)

    for w in active:
        if line_intersect((ax, ay), (x, y), w[0], w[1]):
            rssi -= loss_for_wall(w)

```



© 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

```
# Dead zone penalty (ruangan basah/toilet)
rssi -= dead_zone_penalty_db(ax, ay, x, y)

rssi_list.append(max(min(rssi,-10),-140))

return max(rssi_list)

def hitung_jarak(x, y):
    all_ap_x = state.ap_x + state.ext_ap_x
    all_ap_y = state.ap_y + state.ext_ap_y
    if not all_ap_x: return None
    return min(np.sqrt((x-ax)**2+(y-ay)**2)*state.scale_meter
               for ax,ay in zip(all_ap_x,all_ap_y))

def get_connected_ap(x, y):
    best_rssi, best_label = None, None
    active = state.walls + (state.auto_walls if
state.use_auto_walls else [])

    for idx, (ax,ay) in enumerate(zip(state.ap_x, state.ap_y)):
        d_m = max(np.sqrt((x-ax)**2+(y-ay)**2)*state.scale_meter,1)
        rssi = state.tx_power - PL_D0 - N_LOG*np.log10(d_m)
        for w in active:
            if line_intersect((ax,ay),(x,y),w[0],w[1]): rssi -=
loss_material(wall_material(w))
            rssi = max(min(rssi,-10),-140)
            if best_rssi is None or rssi > best_rssi:
                best_rssi, best_label = rssi, f"AP {idx+1}"

    for idx, (ax,ay) in enumerate(zip(state.ext_ap_x,
state.ext_ap_y)):
        d_m = max(np.sqrt((x-ax)**2+(y-ay)**2)*state.scale_meter,1)
        rssi = state.tx_power - PL_D0 - N_LOG*np.log10(d_m)
        for w in active:
            if line_intersect((ax,ay),(x,y),w[0],w[1]): rssi -=
loss_material(wall_material(w))
            rssi = max(min(rssi,-10),-140)
            if best_rssi is None or rssi > best_rssi:
                best_rssi, best_label = rssi, f"Ext AP {idx+1}"

    return best_label

def get_parameters(x, y):
    rssi = hitung_rssi(x, y)
    if rssi is None:
        return
    {"rssi":None,"snr":None,"throughput":None,"path_loss":None}
    noise = get_noise_floor()
    snr = rssi - noise
    return
    {"rssi":rssi,"snr":snr,"throughput":max(0,snr*2),"path_loss":stat
e.tx_power-rssi}

def get_parameters_with_ml(x, y):
    rssi = hitung_rssi(x, y)
    jarak = hitung_jarak(x, y)
```



© 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

```

    if rssi is None:
        return
    {"rssi":None,"snr":None,"throughput":None,"path_loss":None,
     "kualitas_ml":"-","kualitas_teoris":"-
    ","wall_count":0}

    all_ap_x = state.ap_x + state.ext_ap_x
    all_ap_y = state.ap_y + state.ext_ap_y
    best_idx, best_r = 0, -999
    for i, (ax,ay) in enumerate(zip(all_ap_x,all_ap_y)):
        d_m = max(np.sqrt((x-ax)**2+(y-
ay)**2)*state.scale_meter,1)
        r = state.tx_power - PL_D0 - N_LOG*np.log10(d_m)
        if r > best_r: best_r, best_idx = r, i

    active = state.walls + (state.auto_walls if
state.use_auto_walls else [])

    # Hitung wall_count, material dominan, dan wall_full dominan
    dari tembok yang dilewati
    walls_between = [w for w in active
                     if line_intersect((all_ap_x[best_idx],
all_ap_y[best_idx]),
                                     (x, y), w[0], w[1])] if
all_ap_x else []
    wc = len(walls_between)

    MAT_CODE = {"kaca": 0, "kayu": 1, "bata": 2, "beton": 3}
    mat_codes = [MAT_CODE.get(wall_material(w), 2) for w in
walls_between]
    full_flags = [wall_full_flag(w) for w in walls_between]

    from collections import Counter
    wm_kode = Counter(mat_codes).most_common(1)[0][0] if
mat_codes else 2
    wall_full_val = Counter(full_flags).most_common(1)[0][0] if
full_flags else 1

    noise = get_noise_floor()
    snr = rssi - noise
    kml = get_kualitas_ml(rssi, jarak if jarak else 1, wc,
wm_kode, wall_full_val)
    kteori = get_kualitas(rssi, 'rssi')

    return {"rssi":rssi,"snr":snr,"throughput":max(0,snr*2),
           "path_loss":state.tx_power-rssi,
           "kualitas_ml":kml,"kualitas_teoris":kteori,"wall_count
":wc}

def update_all_rx():
    new_rx = []
    for d in state.rx_data:
        x,y = d[0],d[1]
        params = get_parameters_with_ml(x,y)
        jarak = hitung_jarak(x,y)
        terhubung = get_connected_ap(x,y)
        new_rx.append((x,y,params,jarak,terhubung))
    state.rx_data = new_rx

```



© 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

```
def get_existing_wifi():
    info = {'ssid': 'Tidak
    terdeteksi', 'rssi': None, 'channel': None, 'snr': None, 'quality': 'Unkn
    own'}
    try:
        if platform.system() == 'Windows':
            out =
            subprocess.check_output(['netsh', 'wlan', 'show', 'interfaces'],
                                    shell=True, text=True)

            ms = re.search(r'SSID\s*:\s*(.+)', out)
            mg = re.search(r'Signal\s*:\s*(\d+)%', out)
            mc = re.search(r'Channel\s*:\s*(\d+)', out)
            if ms: info['ssid'] = ms.group(1).strip()
            if mc: info['channel'] = mc.group(1)
            if mg:
                rssi = (int(mg.group(1))/2) - 100
                info['rssi'] = round(rssi, 1)
                info['snr'] = round(rssi-(-90), 1)
                info['quality'] = 'Baik' if rssi>=-65 else
                ('Sedang' if rssi>=-80 else 'Buruk')
            except Exception as e:
                info['ssid'] = f'Error: {e}'
    return info
```

layout.py

```
# =====
# layout.py
# =====

import sys
sys.stdout.reconfigure(encoding='utf-8')

from dash import dcc, html
from app import app
from visualization import create_figure
from login_layout import login_page
from navbar import create_navbar
from callbacks import hitung_skala

app.layout = html.Div(id="root-container", children=[

    dcc.Store(id="session-store", storage_type="session"),
    dcc.Download(id="download-data"),
    dcc.Download(id="download-image"),

    login_page(),

    html.Div(
        id="main-page",
        style={"display": "none"},
        children=[
            html.Div(id="navbar-container"),
```



© 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

```

        html.Div(id="modal-overlay", className="modal-
overlay", style={"display": "none"}, children=[
            html.Div(id="modal-box", className="modal-box",
children=[
                html.Button("X", id="btn-close-modal",
n_clicks=0, className="modal-close"),
                html.Div(id="modal-content"),
            ]),
        ]),

        html.Div(className="main-content", children=[

            # — PANEL KIRI

            html.Div(className="panel-left", children=[

                html.H4("Upload Denah Bangunan",
className="panel-title"),
                dcc.Upload(
                    id='upload-image',
                    children=html.Div([html.Span("📁 "),
html.Span("Upload Gambar Denah")]),
                    className="upload-box",
                    multiple=False,
                ),

                html.Hr(className="divider"),

                html.H4("Deteksi Tembok Otomatis",
className="panel-title"),
                dcc.RadioItems(
                    id='toggle_autoscan',
                    options=[{'label': 'ON', 'value': 'on'},
{'label': 'OFF', 'value': 'off'}],
                    value='on', inline=True,
                    className="radio-group",
                ),
                html.Span("📖 Panduan Ketebalan",
id='autoscan-info-btn', n_clicks=0, className="info-link"),
                html.Div(
                    id='autoscan-info-popup',
                    className="info-popup",
                    children=[
                        html.Table([
                            html.Tr([html.Th("Material"),
html.Th("Ketebalan"), html.Th("Redaman")]),
                            html.Tr([html.Td("Kaca"), html.T
d("1-3 px"), html.Td("-5 dB")]),
                            html.Tr([html.Td("Kayu"), html.T
d("4-7 px"), html.Td("-3 dB")]),
                            html.Tr([html.Td("Bata"), html.T
d("8-14 px"), html.Td("-12 dB")]),
                            html.Tr([html.Td("Beton"),
html.Td("≥ 15 px"), html.Td("-15 dB")]),
                        ], className="info-table"),
                    ],
                    style={"display": "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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

        html.Hr(className="divider"),

        html.H4("Mode Kontrol", className="panel-
title"),

        dcc.RadioItems(
            id='mode',
            options=[
                {'label': '📶 Access
Point', 'value': 'ap'},
                {'label': '📡
Receiver', 'value': 'rx'},
                {'label': '🚧
Obstacle', 'value': 'obs'},
                {'label': '🚫 Dead
Zone', 'value': 'dead_zone'},
                {'label': '📏 Set
Skala', 'value': 'scale'},
                {'label': '✖ Hapus
Titik', 'value': 'delete'},
                {'label': '✂ Hapus
Garis', 'value': 'del_wall'},
                {'label': '🗑 Hapus Dead
Zone', 'value': 'del_dead_zone'},
                {'label': '✎ Edit Material',
'value': 'edit_wall'},
                {'label': '👉 Geser Titik', 'value':
'move'},
                {'label': '🟦 Existing
AP', 'value': 'ext_ap'},
            ],
            value='ap',
            className="radio-group-vertical",
        ),

        html.Hr(className="divider"),

        html.H4("Set Skala", className="panel-
title"),

        dcc.Input(id="real-distance", type="number",
placeholder="Panjang nyata (meter)", className="ctrl-input"),
        html.Button("🔢 Hitung Skala", id="btn-
scale", n_clicks=0, className="ctrl-btn"),

        html.Div(id='scale-display', style={
            "fontSize": "12px", "color": "#4fc3f7",
            "marginTop": "6px", "padding": "6px",
            "backgroundColor": "#1e2a35",
            "borderRadius": "4px",
            "display": "none"
        }),

        html.Hr(className="divider"),

        html.H4("Tampilan Heatmap", className="panel-
title"),

```



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

```

        dcc.RadioItems(
            id='toggle_heatmap',
            options=[{'label': 'ON', 'value': 'on'},
{'label': 'OFF', 'value': 'off'}],
            value='on', inline=True,
            className="radio-group",
        ),

        html.Hr(className="divider"),

        html.H4("🧠 Mode Klasifikasi Kualitas",
            className="panel-title"),
        dcc.RadioItems(
            id='kualitas_mode',
            options=[
                {'label': '📶 ML Interpolasi (dari
Rx)', 'value': 'ml'},
                {'label': '📶 ML Coverage (dari
AP)', 'value': 'ml_coverage'},
                {'label': '📶 Teori (Log-
Distance)', 'value': 'teori'},
            ],
            value='ml', className="radio-group-
vertical",
        ),
        html.P(
            "ML Interpolasi: warna dari titik Rx. "
            "ML Coverage: warna menyebar dari AP
(RF). "
            "Teori: rumus Log-Distance.",
            style={"fontSize": "11px", "color":
"#78909c", "margin": "4px 0 0"},
        ),
        html.Div(style={"marginTop": "10px"},
            children=[
                html.Label("Radius Heatmap ML:",
                    style={"fontSize": "11px",
"fontWeight": "600"}),
                html.Div(style={"display": "flex",
"alignItems": "center", "gap": "8px"}, children=[
                    dcc.Slider(
                        id='heatmap-radius-slider',
                        min=5, max=30, step=1, value=10,
                        marks={5:"5m", 10:"10m",
20:"20m", 30:"30m"},
                        tooltip={"placement": "bottom",
"always_visible": False},
                        className="radius-slider",
                    ),
                ],
            ),
            html.P(
                "Area di luar radius dari Rx terdekat
tidak diwarnai (tidak diketahui).",
                style={"fontSize": "10px", "color":
"var(--text-hint)", "margin": "2px 0 0"},
            ),

```



© 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

```

    )},

    html.Button(
        "🔒 Kunci Heatmap ML",
        id="btn-lock-heatmap", n_clicks=0,
        className="ctrl-btn",
        style={"marginTop": "8px", "width":
"100%",
                    "backgroundColor": "#37474F",
                    "color": "white",
                    "border": "none", "fontWeight":
"bold", "fontSize": "11px"},
    ),
    html.P(id="lock-status-text",
        children="Heatmap: tidak dikunci",
        style={"fontSize": "10px", "color":
"var(--text-hint)",
                    "margin": "3px 0 0"}),
    html.Hr(className="divider"),
    html.H4("Parameter Tampilan",
        className="panel-title"),
    dcc.Dropdown(
        id='parameter',
        options=[
            {'label': '📶 RSSI
(dBm)', 'value': 'rssi'},
            {'label': '🔊 SNR
(dB)', 'value': 'snr'},
            {'label': '⚡ Throughput (Mbps)',
'value': 'throughput'},
            {'label': '📶 Path Loss
(dB)', 'value': 'path_loss'}],
        value='rssi', className="ctrl-dropdown",
    ),
    html.Hr(className="divider"),
    html.H4("Jenis Obstacle", className="panel-
title"),
    dcc.Dropdown(
        id='material',
        options=[
            {'label': '🏠 Beton (-15 dB)',
'value': 'beton'},
            {'label': '🧱 Bata (-12 dB)',
'value': 'bata'},
            {'label': '🪟 Kaca (-5 dB)',
'value': 'kaca'},
            {'label': '🪵 Kayu (-3 dB)',
'value': 'kayu'}],
        value='bata', className="ctrl-dropdown",
    ),

```



© 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

```

html.Div(style={"marginTop": "8px"},
children=[
    html.Label("Tipe Tembok:", style={
        "fontSize": "11px", "color": "var(--
text-muted)",
        "fontWeight": "600", "marginBottom":
"4px", "display": "block",
    }),
    dcc.RadioItems(
        id='wall-full-selector',
        options=[
            {'label': '  Tembok Penuh
(floor-to-ceiling)', 'value': 1},
            {'label': ' ☐ Tembok Setengah /
Partisi', 'value': 0},
        ],
        value=1,
        labelStyle={"display": "block",
"fontSize": "11px",
"marginBottom": "3px"},
        inputStyle={"marginRight": "6px"},
    ),
    html.P(
        "Tembok setengah: redaman -50% dari
tembok penuh.",
        style={"fontSize": "10px", "color":
"var(--text-hint)",
        "margin": "4px 0 0"},
    ),
    html.Hr(className="divider"),
    html.H4("Kekuatan Sinyal (Tx Power)",
className="panel-title"),
    dcc.Dropdown(
        id='tx_power',
        options=[
            {'label': '  Low (10 dBm)',
'value': 10},
            {'label': '  Medium (15 dBm)',
'value': 15},
            {'label': '  High (20 dBm)',
'value': 20},
        ],
        value=15, className="ctrl-dropdown",
    ),
    html.Hr(className="divider"),
    # — NOISE / INTERFERENSI (BARU)

    html.H4("  Noise & Interferensi",
className="panel-title"),
    html.P("Simulasi gangguan sinyal Wi-Fi
lain:", className="hint-text",

```



© 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

```

        style={"fontSize": "11px", "color":
"#78909c", "margin": "2px 0 6px"}),
        dcc.RadioItems(
            id='noise-level',
            options=[
                {'label': ' ☒ Tidak Ada (noise -90
dBm)', 'value': 'none'},
                {'label': ' ☐ Sedang (noise -80
dBm)', 'value': 'medium'},
                {'label': ' ☐ Tinggi (noise -70
dBm)', 'value': 'high'},
            ],
            value='none',
            className="radio-group-vertical",
        ),

        html.Hr(className="divider"),

        html.H4("Penempatan AP Otomatis",
className="panel-title"),
        html.Div(className="auto-ap-row", children=[
            dcc.Input(id="auto-ap-count",
type="number", placeholder="Jml AP",
min=1, max=10, value=1,
className="ctrl-input-small"),
            html.Button("  Pasang Optimal",
id="btn-auto", n_clicks=0,
className="ctrl-btn ctrl-btn-
green"),
        ],

        html.Hr(className="divider"),
        html.Button("  Reset Semua", id="reset-
className="ctrl-btn ctrl-btn-
red"),

        # Notifikasi – warna putih agar terbaca di
background gelap
float",
        html.Div(id='output', className="notif-

        style={"color": "white"}),
    ),

    # — PANEL TENGAH

    html.Div(className="panel-center", children=[
        dcc.Loading(
            id="loading-graph", type="circle",
color="#4fc3f7",

            children=[
                dcc.Graph(
                    id='graph',
                    figure=create_figure(),
                    style={'height': '78vh', 'width':
'100%'},

                    config={

```



© 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

```

        "toImageButtonOptions":
{"format": "png", "filename": "wlan_coverage",
                                "height": 1000, "width": 1400, "scale": 2},
                                "displayModeBar": True,
                                "modeBarButtonsToAdd":
["drawrect"],
                                },
                                ),
                                ],
                                ),

        html.H4("📄 Data Receiver",
className="panel-title", style={"marginTop": "10px"}),
        html.Div(id='table-data'),
    ]),

    # — PANEL KANAN

    html.Div(className="panel-right", children=[

        # — EXISTING AP WORKFLOW (BARU)

        html.Div(id="existing-ap-panel", children=[
            html.Div(
                id="wall-info-panel",
                style={"display": "none"},
            ),
            html.Hr(className="divider"),
            html.Div(id="wall-info-panel",
style={"display": "none"}),
            html.Hr(className="divider"),

            html.H4("🎨 Ubah Semua Tembok",
className="panel-title"),
            html.P("Ganti material semua tembok
sekaligus:",
                style={"fontSize": "11px", "color": "#789
09c", "margin": "2px 0 6px"}),
            dcc.Dropdown(
                id='bulk-material-picker',
                options=[

                    {'label': '🧱 Semua Beton (-15 dB)',
'value': 'beton'},

                    {'label': '🧱 Semua Bata (-12 dB)',
'value': 'bata'},

                    {'label': '🪟 Semua Kaca (-5 dB)',
'value': 'kaca'},

                    {'label': '🪵 Semua Kayu (-3 dB)',
'value': 'kayu'},

                ],
                placeholder="Pilih material...",
                className="ctrl-dropdown",
                style={"marginBottom": "6px"},
            ),
            html.Div(className="btn-row", children=[

```



© 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

```

        html.Button("✅ Terapkan Semua",
id="btn-bulk-material", n_clicks=0,
                        className="ctrl-btn ctrl-btn-
green", style={"flex":"1"}),
                        html.Button("↩ Reset Otomatis", id="btn-
reset-auto-walls", n_clicks=0,
                        className="ctrl-btn",
style={"flex":"1"}),
                    ]),
                    html.Hr(className="divider"),

                    html.H4("🔵 Existing AP – Analisis
Lapangan", className="panel-title"),

                    # Progress steps
                    html.Div([
                        html.Span("1 ",
style={"marginRight":"4px"}),
                        html.Span("Klik tombol ",
style={"fontSize":"11px","color":"#546e7a"}),
                        html.B("Bantuan",
style={"fontSize":"11px","color":"#2E75B6"}),
                        html.Span(" di menu atas untuk
panduan lengkap penggunaan Existing AP.",
style={"fontSize":"11px","c
olor":"#546e7a"}),
                    ],
style={"backgroundColor":"#E3F2FD","borderRadius":"6px",
padding":"8px
10px","marginBottom":"10px",
display":"flex","alignItems":"
center","flexWrap":"wrap"}),

                    # Sumber kekuatan AP
                    html.Label("Sumber Nilai Kekuatan AP:",
className="input-label"),
                    dcc.RadioItems(
                        id='ext-ap-power-source',
                        options=[
                            {'label': '📶 Teori (Tx Power
default)', 'value': 'teori'},
                            {'label': '📶 Terukur (RSSI dari
lapangan)', 'value': 'terukur'},
                        ],
                        value='teori',
                        className="radio-group-vertical",
                    ),

                    # Input RSSI terukur (muncul jika pilih
'terukur')
                    html.Div(id="ext-ap-measured-input",
style={"display": "none"}, children=[
                        html.Label("RSSI Terukur di Lapangan
(dBm):", className="input-label",
style={"marginTop":
"6px"}),
                        html.P("Masukkan nilai RSSI dari WiFi
Analyzer untuk setiap Existing AP:",

```



© 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

```

        style={"fontSize": "11px",
"color": "#78909c", "margin": "2px 0 4px"}),
        html.Div(id="ext-ap-rssi-inputs"),
        html.Button("💾 Simpan RSSI
Terukur", id="btn-save-measured-rssi",
                    n_clicks=0,
className="ctrl-btn", style={"marginTop": "6px"}),
        ),
        html.Hr(className="divider"),
        # Tombol Mulai Analisis
        html.Button(
            "▶ Mulai Analisis Existing AP",
            id="btn-analyze-existing",
            n_clicks=0,
            className="ctrl-btn ctrl-btn-green",
            style={"width": "100%", "fontWeight":
"bold", "fontSize": "13px"}),
        ),
        html.Div(id="existing-ap-result",
                style={"marginTop": "10px",
"fontSize": "12px"}),
        html.Hr(className="divider"),
        ),
        html.H4("📶 Existing Wi-Fi",
className="panel-title"),
        html.Button("🔍 Scan Existing Wi-Fi",
id="scan-wifi-btn", n_clicks=0,
                    className="ctrl-btn ctrl-btn-
purple", style={"width": "100%"}),
        html.Div(id='wifi-info', className="wifi-
info-box"),
        html.Hr(className="divider"),
        html.H4("📝 Input Data Pengukuran",
className="panel-title"),
        html.P("Masukkan data RSSI dari pengukuran
nyata atau aplikasi WiFi Analyzer:",
                className="hint-text"),
        html.Div(className="meas-form", children=[
            html.Label("Jarak (meter)",
className="input-label"),
            dcc.Input(id='input-jarak',
type='number', placeholder='cth: 10',
                    min=0.1, className="ctrl-
input"),
            html.Label("RSSI (dBm)",
className="input-label"),
            dcc.Input(id='input-rssi', type='number',
placeholder='cth: -65',
                    max=0, className="ctrl-input"),
            html.Label("Keterangan (opsional)",
className="input-label"),

```

login-layout.py

```
# =====
# login_layout.py
# Tampilan halaman Login & Register dengan tema Wi-Fi.
# =====

import sys
sys.stdout.reconfigure(encoding='utf-8')

from dash import dcc, html

def login_page():
    """Layout halaman login."""
    return html.Div(
        id="login-page",
        children=[
            # Overlay animasi background
            html.Div(className="login-bg-overlay"),

            # Card utama
            html.Div(
                className="login-card",
                children=[
                    # Header dengan ikon WiFi (SVG inline via
                    dangerouslySetInnerHTML)
                    html.Div(className="login-header", children=[
                        # Ikon WiFi menggunakan div dengan CSS
                        animation
                        html.Div(className="wifi-icon-wrap",
                                children=[
                                    html.Div(className="wifi-arc wifi-
                                    arc-3"),
                                    html.Div(className="wifi-arc wifi-
                                    arc-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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

html.Div(className="wifi-arc wifi-arc-1"),

        html.Div(className="wifi-dot"),
    ]),
    html.H1("SIMNET", className="login-title"),

        html.P("Simulasi Network",
className="login-subtitle"),
    ]),

    # Tab Login / Register
    dcc.Tabs(
        id="auth-tabs",
        value="login",
        className="auth-tabs",
        children=[
            dcc.Tab(
                label="🔒 Login",
                value="login",
                className="auth-tab",
                selected_className="auth-tab-
selected",
            ),
            dcc.Tab(
                label="📝 Register",
                value="register",
                className="auth-tab",
                selected_className="auth-tab-
selected",
            ),
        ],
    ),

    # Konten tab (diisi oleh callback)
    html.Div(id="auth-tab-content"),

    # Pesan status (error/sukses)
    html.Div(id="auth-message", className="auth-
message"),
],
),
style={"display": "flex"},
)

def login_form():
    """Form untuk tab Login."""
    return html.Div(className="auth-form", children=[
        html.Div(className="input-group", children=[
            html.Label("Username", className="input-label"),
            dcc.Input(
                id="login-username",
                type="text",
                placeholder="Masukkan username...",
                className="auth-input",
                debounce=False,
                n_submit=0,
            ),
        ],
    ),

```



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

```

    ]),
    html.Div(className="input-group", children=[
        html.Label("Password", className="input-label"),
        dcc.Input(
            id="login-password",
            type="password",
            placeholder="Masukkan password...",
            className="auth-input",
            debounce=False,
            n_submit=0,
        ),
    ]),
    html.Button(
        "🚀 Masuk",
        id="btn-login",
        n_clicks=0,
        className="auth-btn auth-btn-primary",
    ),
    html.P(
        "Akun default: admin / admin123",
        style={
            "fontSize": "11px",
            "color": "#78909c",
            "textAlign": "center",
            "marginTop": "10px",
        },
    ),
])

def register_form():
    """Form untuk tab Register."""
    return html.Div(className="auth-form", children=[
        html.Div(className="input-group", children=[
            html.Label("Nama Lengkap", className="input-label"),
            dcc.Input(
                id="reg-name",
                type="text",
                placeholder="Nama lengkap Anda...",
                className="auth-input",
            ),
        ]),
        html.Div(className="input-group", children=[
            html.Label("Username", className="input-label"),
            dcc.Input(
                id="reg-username",
                type="text",
                placeholder="Pilih username...",
                className="auth-input",
            ),
        ]),
        html.Div(className="input-group", children=[
            html.Label("Password", className="input-label"),
            dcc.Input(
                id="reg-password",
                type="password",
                placeholder="Minimal 6 karakter...",
                className="auth-input",
            ),
        ]),
    ])

```



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

```

    )),
    html.Button(
        "✅ Daftar Sekarang",
        id="btn-register",
        n_clicks=0,
        className="auth-btn auth-btn-success",
    ),
)

```

navbar.py

```

import sys
sys.stdout.reconfigure(encoding='utf-8')
from dash import html, dcc

def create_navbar(user_name: str = "Pengguna"):
    return html.Div(
        id="navbar", className="navbar",
        children=[
            html.Div(className="navbar-brand", children=[
                html.Span("🏠",
                    style={"fontSize": "24px", "marginRight": "10px"}),
                html.Span("Simulation Network",
                    className="navbar-title"),
            ]),
            html.Div(className="navbar-menu", children=[
                html.Button("💾 Simpan Data", id="btn-save-data", n_clicks=0, className="nav-btn"),
                html.Button("📄 Export PNG", id="btn-export-img", n_clicks=0, className="nav-btn"),
                html.Button("📄 Export CSV", id="btn-export-csv", n_clicks=0, className="nav-btn",
                    style={"backgroundColor": "#2e7d32", "color": "white", "border": "none", "fontWeight": "bold"}),
                html.Button("📁 Muat Data", id="btn-load-data", n_clicks=0, className="nav-btn"),
                html.Button("📊 Statistik", id="btn-stats", n_clicks=0, className="nav-btn"),
                html.Button("❓ Bantuan", id="btn-help", n_clicks=0, className="nav-btn"),
                html.Div(style={
                    "display": "flex", "alignItems": "center", "gap": "6px",
                    "marginLeft": "12px", "paddingLeft": "12px",
                    "borderLeft": "1px solid rgba(255,255,255,0.15)"}, children=[
                    html.Span("📶 RF", style={
                        "fontSize": "11px", "color": "#ce93d8",
                        "fontWeight": "600", "whiteSpace": "nowrap"},
                    ),
                    html.Button("📍 Posisi AP Optimal",

```



© 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

```

id="btn-optimal-ml", n_clicks=0,
className="nav-btn",
title="Pindahkan AP ke posisi terbaik
berdasarkan prediksi RF",
style={"backgroundColor":"#6A1B9A","color":
"white",
"border":"none","fontWeight":"bold",
"fontSize":"11px"}),
    ),
    html.Div(className="navbar-user", children=[
        html.Span(f"👤 {user_name}", className="navbar-
username"),
        html.Button("🚪 Keluar", id="btn-logout",
n_clicks=0,
className="nav-btn nav-btn-logout"),
    ),
    dcc.Store(id="store-figure-json"),
    1,
)

```

run.py

```

# =====
# run.py
# Entry point aplikasi – jalankan file ini untuk memulai server.
#
# Cara menjalankan:
# python run.py
# =====

from app import app

import layout # noqa: F401 – mendaftarkan app.layout
import callbacks # noqa: F401 – mendaftarkan semua
@app.callback (simulasi)
import auth_callbacks # noqa: F401 – mendaftarkan callback
login/logout/save/export

if __name__ == '__main__':
    app.run(debug=True)

```

state.py

```

# =====
# state.py
# =====

ap_x = []
ap_y = []

```



© 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

```

ext_ap_x = []
ext_ap_y = []
ext_ap_measured = []

rx_x = []
rx_y = []

# walls format: ((x1,y1), (x2,y2), material, source, confidence)
# source: 'color' | 'thickness' | 'manual'
walls = []
temp_wall = []
auto_walls = []
use_auto_walls = True

rx_data = []
wifi_measurements = []
scale_points = []

img_height = 1000
img_width = 1000
scale_meter = 0.05
tx_power = 15
uploaded_image = None

selected_point = None
selected_type = None

# Noise / interferensi
noise_floor = -90

# Existing AP power source
ext_ap_power_source = 'teori'

# Mode tampilan kualitas sinyal: 'ml' atau 'teori'
kualitas_mode = 'ml'

# Radius tampilan heatmap ML (meter)
heatmap_radius = 10.0

# Kunci heatmap ML
heatmap_locked = False
heatmap_locked_grid = None

# Dead Zone - area yang memblokir sinyal (toilet, ruangan basah)
# Format: list of ((x1,y1), (x2,y2)) dalam koordinat piksel
dead_zones = []
dead_zone_tmp = None # simpan klik pertama saat menggambar dead zone

# Data test baru yang diupload user (untuk fitur "Bandingkan dengan Data Test Baru")
# list of dict: {distance_meter, wall_count, rssi_teoris, label_aktual}
test_data = []
test_data_filename = None

# Wall edit mode
selected_wall_idx = 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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
selected_wall_source = None    # 'auto' | 'manual'

# Tipe tembok yang dipilih user saat menggambar obstacle
# 1 = tembok penuh (floor-to-ceiling), 0 = tembok
# setengah/partisi
selected_wall_full    = 1
```

train_rf.py

```
# =====
# train_rf.py
# -----
# Script training model Random Forest untuk SimNet.
#
# Fitur input (4 kondisi fisik murni, TANPA rssi_teoris):
#   distance      : jarak AP-Rx (meter)
#   wall_count    : jumlah tembok
#   wall_material : material tembok (0=kaca, 1=kayu, 2=bata)
#   wall_full     : tipe tembok (1=penuh, 0=setengah/partisi)
#
# Label output:
#   Baik / Sedang / Buruk
#   (ditentukan dari threshold RSSI hasil pengukuran lapangan)
#
# Cara pakai:
#   python train_rf.py nama_dataset.xlsx
#
# Output:
#   rf_wlan_model.pkl  ← model yang dipakai aplikasi SimNet
#   rf_report.txt      ← classification report + CV
#   rf_confusion.png   ← confusion matrix
# =====

import os
import sys
import numpy as np
import pandas as pd
import joblib

import matplotlib
matplotlib.use("Agg")
import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split,
StratifiedKFold, cross_val_score
from sklearn.preprocessing import LabelEncoder
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import (
    accuracy_score, f1_score, precision_score, recall_score,
    classification_report, confusion_matrix,
)

sys.stdout.reconfigure(encoding="utf-8")

# — Konfigurasi —
```



© 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

```

OUTPUT_DIR      = os.path.dirname(os.path.abspath(__file__))
FEATURE_COLUMNS = ["distance", "wall_count", "wall_material",
"wall_full"]
DIST_BUCKETS     = [1, 2, 5, 10, 15, 20]

WALL_MATERIAL_MAP = {0: "kaca", 1: "kayu", 2: "bata", 3: "beton"}
PARTIAL_WALL_FACTOR = 0.5

# — Parameter Random Forest —————
RF_PARAMS = {
    "n_estimators": 200,
    "class_weight": "balanced",
    "random_state": 42,
    "n_jobs": -1,
}

# =====
# 1. LOAD DATASET
# =====
def load_dataset(path):
    print(f"\n[1/4] Membaca dataset: {path}")
    ext = os.path.splitext(path)[1].lower()
    df = pd.read_csv(path) if ext == ".csv" else
pd.read_excel(path)
    print(f"        Kolom tersedia: {list(df.columns)}")

    # Normalisasi nama kolom
    ALIAS = {
        "distance_meter": ["distance", "distance_m", "jarak"],
        "wall_count":     ["tembok", "jumlah_tembok",
"wallcount"],
        "wall_material":  ["material", "jenis_tembok"],
        "wall_full":      ["full_wall", "tipe_wall", "partisi"],
        "label":          ["kualitas", "quality", "kelas"],
    }
    for baku, alias_list in ALIAS.items():
        if baku not in df.columns:
            for alias in alias_list:
                if alias in df.columns:
                    df = df.rename(columns={alias: baku})
                    print(f"        '{alias}' → '{baku}'")
                    break

    # Kolom wajib
    missing = [c for c in ["distance_meter", "wall_count",
"label"] if c not in df.columns]
    if missing:
        raise ValueError(
            f"Kolom wajib tidak ditemukan: {missing}\n"
            f"Kolom di file: {list(df.columns)}"
        )

    # Default kolom opsional
    if "wall_material" not in df.columns:
        df["wall_material"] = 2
        print("        'wall_material' tidak ada → default bata
(2)")
    if "wall_full" not in df.columns:
        df["wall_full"] = 1

```



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

```

print("      'wall_full' tidak ada → default tembok penuh
(1)")

# Bersihkan
df = df.dropna(subset=["distance_meter", "wall_count",
"label"]).reset_index(drop=True)

# Pastikan tipe data
df["distance"]      = df["distance_meter"].astype(int)
df["wall_count"]    = df["wall_count"].astype(int)
df["wall_material"] =
df["wall_material"].fillna(2).astype(int)
df["wall_full"]     = df["wall_full"].fillna(1).astype(int)

print(f"      Jumlah sampel: {len(df)}")
print(f"      Distribusi label:")
for lbl, cnt in df["label"].value_counts().items():
    pct = cnt / len(df) * 100
    print(f"      {lbl:8s}: {cnt:4d} ({pct:.1f}%)")

return df

# =====
# 2. TRAINING
# =====
def train(df):
    print(f"\n[2/4] Training Random Forest...")
    print(f"      Fitur: {FEATURE_COLUMNS}")

    X = df[FEATURE_COLUMNS].values
    le = LabelEncoder()
    y = le.fit_transform(df["label"].astype(str))
    print(f"      Kelas: {list(le.classes_)}")

    X_train, X_test, y_train, y_test = train_test_split(
        X, y, test_size=0.2, stratify=y, random_state=42
    )
    print(f"      Data latih: {len(X_train)} | Data uji:
{len(X_test)}")

    model = RandomForestClassifier(**RF_PARAMS)
    model.fit(X_train, y_train)

    y_pred = model.predict(X_test)

    # Metrik test set
    acc = accuracy_score(y_test, y_pred)
    f1 = f1_score(y_test, y_pred, average="macro",
zero_division=0)
    prec = precision_score(y_test, y_pred, average="macro",
zero_division=0)
    rec = recall_score(y_test, y_pred, average="macro",
zero_division=0)
    cm = confusion_matrix(y_test, y_pred)
    rep = classification_report(y_test, y_pred,
target_names=le.classes_, zero_division=0)

    # Cross-validation

```



© 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

```

cv = StratifiedKFold(n_splits=5, shuffle=True,
random_state=42)
cv_scores = cross_val_score(model, X_train, y_train, cv=cv,
scoring="accuracy")

print(f"\n      === HASIL EVALUASI ===")
print(f"      Akurasi (test set)      : {acc:.4f}")
print(f"      F1-Score (macro)         : {f1:.4f}")
print(f"      Precision (macro)        : {prec:.4f}")
print(f"      Recall (macro)           : {rec:.4f}")
print(f"      CV Accuracy (5-fold)      : {cv_scores.mean():.4f} ±
{cv_scores.std():.4f}")
print(f"\n      Classification Report:")
for line in rep.split("\n"):
    if line.strip():
        print(f"      {line}")

# Feature importance
print(f"\n      Feature Importance:")
for fname, imp in zip(FEATURE_COLUMNS,
model.feature_importances_):
    bar = "█" * int(imp * 40)
    print(f"      {fname:15s}: {imp:.4f} {bar}")

return {
    "model":    model,
    "le":       le,
    "metrics": {
        "accuracy": acc,
        "f1_macro": f1,
        "precision": prec,
        "recall":    rec,
        "cv_mean":   float(cv_scores.mean()),
        "cv_std":    float(cv_scores.std()),
    },
    "report":   rep,
    "cm":       cm,
    "y_test":   le.inverse_transform(y_test),
    "y_pred":   le.inverse_transform(y_pred),
}

# =====
# 3. SIMPAN
# =====
def save(result):
    print(f"\n[3/4] Menyimpan model dan laporan...")
    le = result["le"]
    model = result["model"]
    m = result["metrics"]

    # Bundle .pkl (format yang dipakai app.py SimNet)
    bundle = {
        "model":    model,
        "label_encoder": le,
        "features":  FEATURE_COLUMNS,
        "model_name": "Random Forest",
        "accuracy":  m["accuracy"],
        "f1":        m["f1_macro"],
        "cv_mean":   m["cv_mean"],

```



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

```

}
pkl_path = os.path.join(OUTPUT_DIR, "rf_wlan_model.pkl")
joblib.dump(bundle, pkl_path)
print(f"      Tersimpan: rf_wlan_model.pkl")

# Classification report
rpt_path = os.path.join(OUTPUT_DIR, "rf_report.txt")
with open(rpt_path, "w", encoding="utf-8") as f:
    f.write("=== Random Forest - SimNet ===\n")
    f.write(f"Fitur : {FEATURE_COLUMNS}\n")
    f.write(f"Kelas : {list(le.classes_)}\n\n")
    f.write(f"Akurasi : {m['accuracy']:.4f}\n")
    f.write(f"F1-macro : {m['f1_macro']:.4f}\n")
    f.write(f"Precision: {m['precision']:.4f}\n")
    f.write(f"Recall : {m['recall']:.4f}\n")
    f.write(f"CV Acc : {m['cv_mean']:.4f} ±
{m['cv_std']:.4f}\n\n")
    f.write("Classification Report:\n")
    f.write(result["report"])
    f.write("\nConfusion Matrix (baris=aktual,
kolom=prediksi):\n")
    f.write(f"Urutan kelas: {list(le.classes_)}\n")
    f.write(str(result["cm"]))
print(f"      Tersimpan: rf_report.txt")

# Confusion matrix PNG
fig, ax = plt.subplots(figsize=(6, 5))
cm = result["cm"]
ax.imshow(cm, cmap="Blues")
ax.set_title("Confusion Matrix - Random Forest", fontsize=12)
ax.set_xticks(range(len(le.classes_)))
ax.set_yticks(range(len(le.classes_)))
ax.set_xticklabels(le.classes_, rotation=30)
ax.set_yticklabels(le.classes_)
ax.set_xlabel("Prediksi"); ax.set_ylabel("Aktual")
thresh = cm.max() / 2 if cm.max() > 0 else 0.5
for i in range(cm.shape[0]):
    for j in range(cm.shape[1]):
        ax.text(j, i, str(cm[i, j]), ha="center",
va="center",
color="white" if cm[i, j] > thresh else
"black")
plt.tight_layout()
cm_path = os.path.join(OUTPUT_DIR, "rf_confusion.png")
plt.savefig(cm_path, dpi=150)
plt.close()
print(f"      Tersimpan: rf_confusion.png")

# =====
# 3b. VISUALISASI POHON KEPUTUSAN & FEATURE IMPORTANCE
# =====
def visualisasi_pohon(result):
    """
    Hasilkan dua visualisasi pohon dari Random Forest:
    1. rf_tree_3level.png - pohon 3 level, mudah dibaca di
laporan
    2. rf_tree_full.png - pohon 4 level, lebih detail
    3. rf_feature_importance.png - kontribusi tiap fitur

```



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

```

Dipilih pohon dengan kedalaman mendekati median dari 200
pohon
(paling representatif terhadap ensemble secara keseluruhan).
"""
from sklearn.tree import plot_tree

print(f"\n[3b] Membuat visualisasi pohon keputusan dan
feature importance...")

model      = result["model"]
le         = result["le"]
CLASS_NAMES = list(le.classes_)
m         = result["metrics"]

# — Pilih pohon yang paling representatif —————
depths     = [(i, est.get_depth()) for i, est in
enumerate(model.estimators_)]
med_depth  = int(np.median([d for _, d in depths]))
candidates = [(i, d) for i, d in depths if abs(d - med_depth)
<= 1]
chosen_idx = candidates[len(candidates) // 2][0]
tree       = model.estimators_[chosen_idx]
print(f"      Pohon terpilih: #{chosen_idx} |
kedalaman={tree.get_depth()} | "
      f"node={tree.tree_.node_count}")

plt.rcParams.update({'font.family': 'DejaVu Sans'})

# — Gambar 1: Pohon 3 level (untuk laporan utama) —————
fig1, ax1 = plt.subplots(figsize=(22, 10))
plot_tree(
    tree, max_depth=3, feature_names=FEATURE_COLUMNS,
    class_names=CLASS_NAMES, filled=True, rounded=True,
    impurity=True, proportion=False, fontsize=10, ax=ax1,
precision=2,
)
ax1.set_title(
    f"Visualisasi Pohon Keputusan (3 Level Pertama)\n"
    f"Sampel Pohon #{chosen_idx} dari 200 Pohon Random
Forest\n"
    f"Hijau = Baik | Merah = Buruk | Oranye = Sedang\n"
    f"samples = jumlah data training di node | "
    f"value = [Baik, Buruk, Sedang] | "
    f"gini = tingkat ketidakmurnian node",
    fontsize=11, pad=14,
)
plt.tight_layout()
p1 = os.path.join(OUTPUT_DIR, "rf_tree_3level.png")
fig1.savefig(p1, dpi=180, bbox_inches="tight",
facecolor="white")
plt.close(fig1)
print(f"      Tersimpan: rf_tree_3level.png")

# — Gambar 2: Pohon 4 level (lebih detail) —————
fig2, ax2 = plt.subplots(figsize=(32, 14))
plot_tree(
    tree, max_depth=4, feature_names=FEATURE_COLUMNS,
    class_names=CLASS_NAMES, filled=True, rounded=True,

```



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

```

        impurity=True, proportion=False, fontsize=9, ax=ax2,
        precision=2,
    )
    ax2.set_title(
        f"Visualisasi Pohon Keputusan (4 Level) – Pohon
        #{chosen_idx} dari 200 Pohon RF\n"
        f"Fitur: {' | '.join(FEATURE_COLUMNS)} "
        f"Akurasi model: {m['accuracy']*100:.2f}% | "
        f"F1-macro: {m['f1_macro']*100:.2f}%",
        fontsize=11, pad=14,
    )
    plt.tight_layout()
    p2 = os.path.join(OUTPUT_DIR, "rf_tree_full.png")
    fig2.savefig(p2, dpi=180, bbox_inches="tight",
    facecolor="white")
    plt.close(fig2)
    print(f"    Tersimpan: rf_tree_full.png")

    # — Gambar 3: Feature importance

    fig3, ax3 = plt.subplots(figsize=(8, 5))
    importances = model.feature_importances_
    colors = ["#2196F3", "#4CAF50", "#FF9800", "#9C27B0"]
    bars = ax3.barh(FEATURE_COLUMNS, importances, color=colors,
    edgecolor="none")
    ax3.set_xlabel("Kontribusi terhadap keputusan model",
    fontsize=11)
    ax3.set_title("Feature Importance – Random Forest SimNet",
    fontsize=13, pad=12)
    ax3.set_xlim(0, max(importances) * 1.20)
    for bar, imp in zip(bars, importances):
        ax3.text(imp + 0.005, bar.get_y() + bar.get_height() / 2,
        f"{imp*100:.1f}%", va="center", fontsize=11,
        fontweight="bold")
    ax3.invert_yaxis()
    ax3.grid(axis="x", alpha=0.3)
    ax3.spines[["top", "right"]].set_visible(False)
    plt.tight_layout()
    p3 = os.path.join(OUTPUT_DIR, "rf_feature_importance.png")
    fig3.savefig(p3, dpi=180, bbox_inches="tight",
    facecolor="white")
    plt.close(fig3)
    print(f"    Tersimpan: rf_feature_importance.png")

    return chosen_idx, tree.get_depth()

# =====
# 4. RINGKASAN
# =====
def print_summary(result, chosen_idx=None):
    m = result["metrics"]
    print(f"\n[4/4] === RINGKASAN AKHIR ===")
    print(f"    Akurasi      : {m['accuracy']*100:.2f}%")
    print(f"    F1-Macro     : {m['f1_macro']*100:.2f}%")
    print(f"    CV Accuracy  : {m['cv_mean']*100:.2f}% ±
    {m['cv_std']*100:.2f}%")
    print(f"\n    File output:")
    print(f"    rf_wlan_model.pkl      ← dipakai langsung
    oleh SimNet")

```



© 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

```

    print(f"          rf_report.txt          ← laporan evaluasi
    lengkap")
    print(f"          rf_confusion.png        ← confusion
    matrix")
    print(f"          rf_tree_3level.png       ← visualisasi
    pohon (3 level, untuk laporan)")
    print(f"          rf_tree_full.png        ← visualisasi
    pohon (4 level, detail)")
    print(f"          rf_feature_importance.png ← kontribusi tiap
    fitur")
    if chosen_idx is not None:
        print(f"\n          Pohon yang divisualisasikan:
    #{chosen_idx} dari 200 pohon")
    print(f"\n          Selesai. Model siap dipakai di aplikasi
    SimNet.")

# =====
# MAIN
# =====
if __name__ == "__main__":
    if len(sys.argv) < 2:
        print("Cara pakai: python train_rf.py nama_dataset.xlsx")
        print("          python train_rf.py nama_dataset.csv")
        sys.exit(1)

    path = sys.argv[1]
    if not os.path.exists(path):
        print(f"[ERROR] File tidak ditemukan: {path}")
        sys.exit(1)

    df          = load_dataset(path)
    result      = train(df)
    save(result)
    chosen_idx, _ = visualisasi_pohon(result)
    print_summary(result, chosen_idx)
  
```

vision.py

```

# =====
# vision.py - v3 (warna + ketebalan + manual override)
#
# Strategi deteksi material (prioritas urutan):
#   A. Warna piksel (HSV) → akurat untuk denah SketchUp berwarna
#   B. Ketebalan piksel → fallback untuk denah hitam-putih
#   NMS → hapus garis dobel dari tembok tebal
#
# Manual override dilakukan dari callbacks.py (mode 'edit_wall')
# =====

import cv2
import numpy as np
import base64
import state
import sys
sys.stdout.reconfigure(encoding='utf-8')
  
```



© 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

```
#
# TABEL REFERENSI MATERIAL
#

THICKNESS_GUIDE = {
    'kaca': (1, 3),
    'kayu': (4, 7),
    'bata': (8, 14),
    'beton': (15, 999),
}

# HSV ranges per material (dikalibrasi dari denah SketchUp)
# Format: list of (lower_hsv, upper_hsv)
COLOR_RANGES = {
    'beton': [
        # Hitam / abu sangat gelap (v < 60 apapun hue-nya)
        ((0, 0, 0), (180, 255, 55)),
    ],
    'bata': [
        # Merah HSV range 1 (0-10)
        ((0, 100, 50), (10, 255, 255)),
        # Merah HSV range 2 (165-180)
        ((165, 100, 50), (180, 255, 255)),
    ],
    'kaca': [
        # Cyan / biru-hijau terang (SketchUp kaca)
        ((82, 60, 80), (100, 255, 255)),
        # Biru muda
        ((100, 60, 80), (120, 200, 255)),
    ],
    'kayu': [
        # Coklat / oranye tua
        ((95, 80, 30), (115, 255, 200)),
        # Oranye kecoklatan
        ((8, 80, 60), (20, 255, 200)),
    ],
}

#
# A. DETEKSI MATERIAL DARI WARNA
#

def _build_color_masks(img):
    """
    Buat mask biner per material berdasarkan warna piksel.
    Return dict: {'beton': mask, 'bata': mask, 'kaca': mask,
    'kayu': mask}
    """
    hsv = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)
    masks = {}
    order = ['beton', 'bata', 'kaca', 'kayu'] # prioritas jika
    overlap

    for mat in order:
        combined = np.zeros(img.shape[:2], dtype=np.uint8)
        for (lo, hi) in COLOR_RANGES[mat]:
            combined = cv2.bitwise_or(combined,
```



© 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

```

cv2.inRange(hsv,
np.array(lo), np.array(hi)))
    masks[mat] = combined

    return masks

def classify_material_by_color(img, x1, y1, x2, y2, masks):
    """
    Tentukan material garis berdasarkan warna mayoritas piksel
    di sepanjang segmen garis (sampling setiap 4 piksel).

    Return (material, confidence) dimana confidence = 0.0-1.0.
    Jika tidak ada warna dominan, return (None, 0.0).
    """
    h, w = img.shape[:2]
    length = np.sqrt((x2-x1)**2 + (y2-y1)**2)
    if length < 1:
        return None, 0.0

    n_samples = max(5, int(length / 4))
    ts = np.linspace(0, 1, n_samples)
    xs = np.clip((x1 + ts*(x2-x1)).astype(int), 0, w-1)
    ys = np.clip((y1 + ts*(y2-y1)).astype(int), 0, h-1)

    # Hitung berapa piksel per material
    counts = {}
    for mat, mask in masks.items():
        counts[mat] = int(np.sum(mask[ys, xs] > 0))

    total = sum(counts.values())
    if total == 0:
        return None, 0.0

    best_mat = max(counts, key=counts.get)
    best_conf = counts[best_mat] / n_samples

    # Minimal 25% piksel harus cocok agar dianggap terdeteksi
    if best_conf < 0.25:
        return None, best_conf

    return best_mat, best_conf

# -----
# B. DETEKSI MATERIAL DARI KETEBALAN
# -----

def measure_line_thickness(binary, x1, y1, x2, y2):
    """Sampling tegak lurus di titik tengah untuk estimasi
    ketebalan piksel."""
    mx, my = int((x1+x2)/2), int((y1+y2)/2)
    dx, dy = x2-x1, y2-y1
    length = np.sqrt(dx**2 + dy**2)
    if length < 1:
        return 1

    perp_x = -dy / length
    perp_y = dx / length
    h, w = binary.shape
    count = 0

```



© 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

```

for t in range(-30, 31):
    px = int(round(mx + t * perp_x))
    py = int(round(my + t * perp_y))
    if 0 <= px < w and 0 <= py < h:
        if binary[py, px] > 0:
            count += 1

return max(1, count)

def classify_material_by_thickness(thickness):
    if thickness >= 15: return 'beton'
    elif thickness >= 8: return 'bata'
    elif thickness >= 4: return 'kayu'
    else: return 'kaca'

# -----
# GABUNGAN A + B
# -----

def classify_material(img, binary, x1, y1, x2, y2, masks):
    """
    Prioritas:
    1. Warna (confidence >= 0.25) → paling akurat untuk SketchUp
    2. Ketebalan piksel → fallback untuk denah B&W
    """
    mat_color, conf = classify_material_by_color(img, x1, y1, x2, y2, masks)

    if mat_color is not None:
        return mat_color, 'color', round(conf, 2)

    thickness = measure_line_thickness(binary, x1, y1, x2, y2)
    mat_thick = classify_material_by_thickness(thickness)
    return mat_thick, 'thickness', thickness

# -----
# FILTER GARIS DIMENSI / ANOTASI
# Garis jarak/label pada denah SketchUp berwarna abu-abu terang
# dan biasanya lebih pendek dari tembok nyata.
# -----

def _is_dimension_line(img, x1, y1, x2, y2, min_wall_length):
    """
    Return True jika garis kemungkinan garis dimensi (bukan tembok).
    Kriteria: terlalu pendek ATAU berwarna abu-abu terang / putih.
    """
    length = np.sqrt((x2-x1)**2 + (y2-y1)**2)
    if length < min_wall_length * 0.6:
        return True
  
```



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

```

h_img, w_img = img.shape[:2]
n = max(5, int(length / 6))
ts = np.linspace(0, 1, n)
xs = np.clip((x1 + ts*(x2-x1)).astype(int), 0, w_img-1)
ys = np.clip((y1 + ts*(y2-y1)).astype(int), 0, h_img-1)

pixels = img[ys, xs].astype(float)
avg_b, avg_g, avg_r = pixels[:,0].mean(), pixels[:,1].mean(),
pixels[:,2].mean()

# Abu-abu terang: semua channel hampir sama dan cukup terang
if abs(avg_r-avg_g) < 20 and abs(avg_g-avg_b) < 20 and avg_r
> 150:
    return True
# Hampir putih (background)
if avg_r > 200 and avg_g > 200 and avg_b > 200:
    return True

return False

# -----
# NMS - HAPUS GARIS DOBEL
# -----

def _line_angle(x1, y1, x2, y2):
    return np.degrees(np.arctan2(abs(y2-y1), abs(x2-x1))) % 180

def _perp_dist(mx, my, x1, y1, x2, y2):
    dx, dy = x2-x1, y2-y1
    L = np.sqrt(dx**2 + dy**2)
    if L < 1e-6:
        return np.sqrt((mx-x1)**2 + (my-y1)**2)
    return abs(dy*mx - dx*my + x2*y1 - y2*x1) / L

def _nms_lines(lines, angle_tol=12.0, dist_tol=12.0):
    """Merge garis paralel yang jaraknya < dist_tol. Simpan yang
    terpanjang."""
    if not lines:
        return []

    segs = list(lines)
    used = [False] * len(segs)
    order = sorted(range(len(segs)),
                    key=lambda i: np.sqrt((segs[i][2]-
segs[i][0])**2 +
                                         (segs[i][3]-
segs[i][1])**2),
                    reverse=True)

    result = []

    for idx in order:
        if used[idx]: continue
        x1,y1,x2,y2 = segs[idx]
        ang_i = _line_angle(x1,y1,x2,y2)
        mx_i = (x1+x2)/2
        my_i = (y1+y2)/2
        used[idx] = True

        for jdx in order:

```



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

```

        if used[jdx] or jdx == idx: continue
        x3,y3,x4,y4 = segs[jdx]
        ang_j = _line_angle(x3,y3,x4,y4)
        d_ang = abs(ang_i - ang_j)
        if d_ang > 90: d_ang = 180 - d_ang
        if d_ang > angle_tol: continue
        mx_j = (x3+x4)/2; my_j = (y3+y4)/2
        if _perp_dist(mx_j, my_j, x1,y1,x2,y2) < dist_tol:
            used[jdx] = True

    result.append((x1,y1,x2,y2))

return result

# _____
# PREPROCESSING ADAPTIF
# _____

def _preprocess(img):
    """Binary + edges adaptif sesuai karakteristik gambar."""
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    hsv = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)

    # Mask semua warna tembok yang dikenal
    all_color = np.zeros(img.shape[:2], dtype=np.uint8)
    for mat, ranges in COLOR_RANGES.items():
        for (lo, hi) in ranges:
            all_color = cv2.bitwise_or(all_color,
                                       cv2.inRange(hsv,
np.array(lo), np.array(hi)))

    # Binary dari gray (invert: gelap jadi putih)
    _, binary_inv = cv2.threshold(gray, 100, 255,
cv2.THRESH_BINARY_INV)

    # Adaptive threshold untuk gambar terang
    binary_adapt = cv2.adaptiveThreshold(
        gray, 255,
        cv2.ADAPTIVE_THRESH_GAUSSIAN_C,
        cv2.THRESH_BINARY_INV,
        blockSize=15, C=8
    )

    # Pilih strategi: color-heavy vs grayscale
    h_img, w_img = img.shape[:2]
    color_ratio = np.sum(all_color > 0) / (h_img * w_img)
    binary = cv2.bitwise_or(all_color, binary_inv) if
color_ratio > 0.005 \
        else binary_adapt

    # Bersihkan noise
    kernel = cv2.getStructuringElement(cv2.MORPH_RECT, (2,2))
    binary = cv2.morphologyEx(binary, cv2.MORPH_OPEN, kernel)
    binary = cv2.morphologyEx(binary, cv2.MORPH_CLOSE, kernel)

    # Edges multi-threshold
    med = np.median(gray)
    lo = max(20, int(med * 0.4))
    hi = min(250, int(med * 1.2))

```



© 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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

edges = cv2.bitwise_or(
    cv2.Canny(gray, 10, 10, apertureSize=3),
    cv2.Canny(binary, 30, 120, apertureSize=3),
)
edges = cv2.dilate(edges, np.ones((2,2), np.uint8),
iterations=1)

return binary, edges

# -----
# FUNGSI UTAMA
# -----

def auto_scan_walls(contents, material_type='bata'):
    """
    Deteksi tembok otomatis dari gambar denah.
    Material ditentukan dari warna piksel (A) atau ketebalan (B).
    Hasil disimpan ke state.auto_walls sebagai list:
    ((x1,y1_plotly), (x2,y2_plotly), material, source,
confidence)
    """
    _, cs = contents.split(',')
    nparr = np.frombuffer(base64.b64decode(cs), np.uint8)
    img = cv2.imdecode(nparr, cv2.IMREAD_COLOR)
    if img is None: return

    h_img, w_img = img.shape[:2]
    min_len = max(25, int(min(h_img, w_img) * 0.03))
    dist_tol = max(8, int(min(h_img, w_img) * 0.015))

    binary, edges = _preprocess(img)
    masks = _build_color_masks(img)

    # Multi-pass Hough
    raw = []
    for thr, gap in [(80, 10), (40, 15)]:
        batch = cv2.HoughLinesP(edges, 1, np.pi/180,
                                threshold=thr,
minLineLength=min_len, maxLineGap=gap)
        if batch is not None:
            raw.extend(tuple(l[0]) for l in batch)

    # Fallback jika kosong
    if not raw:
        batch = cv2.HoughLinesP(edges, 1, np.pi/180,
                                threshold=20,
minLineLength=max(15, min_len//2),
                                maxLineGap=20)
        if batch is not None:
            raw.extend(tuple(l[0]) for l in batch)

    # NMS
    cleaned = _nms_lines(raw, angle_tol=12.0, dist_tol=dist_tol)

    # Simpan ke state - filter garis dimensi/jarak terlebih
dahulu
    state.auto_walls.clear()
    n_color = 0; n_filtered = 0
    for (x1,y1,x2,y2) in cleaned:

```



© 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

```

        if _is_dimension_line(img, x1, y1, x2, y2, min_len):
            n_filtered += 1
            continue
        mat, src, conf = classify_material(img, binary,
x1,y1,x2,y2, masks)
        if src == 'color': n_color += 1
        y1p = h_img - y1
        y2p = h_img - y2
        state.auto_walls.append(((x1, y1p), (x2, y2p), mat, src,
round(conf,2)))

    print(f"[vision] raw={len(raw)} -> NMS={len(cleaned)} "
          f"-> filtered={n_filtered} ->
final={len(state.auto_walls)} "
          f"| color={n_color} thickness={len(state.auto_walls)-
n_color} "
          f"| dist_tol={dist_tol}px")

def change_wall_material(wall_idx, new_material,
wall_source='auto'):
    """
    Override material sebuah tembok (manual edit dari UI).
    wall_source: 'auto' atau 'manual'
    """
    walls = state.auto_walls if wall_source == 'auto' else
state.walls
    if 0 <= wall_idx < len(walls):
        w = walls[wall_idx]
        # Pertahankan koordinat, update material + tandai manual
        p1, p2 = w[0], w[1]
        walls[wall_idx] = (p1, p2, new_material, 'manual', 1.0)
        return True
    return False

```

visualization.py

```

# =====
# visualization.py -- v3
# Perubahan: wall tuple 3 atau 5 elemen, ambil material via w[2]
# =====

import numpy as np
import pandas as pd
import plotly.graph_objects as go
from PIL import Image
import io, base64

import state
from functions import loss_material, warna_material,
wall_material, wall_full_flag, loss_for_wall, line_intersect

LABEL_COLOR = {"Baik": "#4CAF50", "Sedang": "#FF9800", "Buruk":
"#F44336"}

```



© 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

```
def create_test_data_scatter(jarak, rssi_teor, label_aktual,
pred_benar):
    """
    Scatter plot dataset test baru: Jarak vs RSSI Teori,
    warna=label aktual,
    bentuk titik=benar/salah prediksi RF.
    """
    fig = go.Figure()
    jarak = np.asarray(jarak, dtype=float)
    rssi_teor = np.asarray(rssi_teor, dtype=float)
    label_aktual = np.asarray(label_aktual)
    pred_benar = np.asarray(pred_benar, dtype=bool)

    rng = np.random.default_rng(42)
    jarak_jitter = jarak + rng.uniform(-0.3, 0.3,
size=len(jarak)) * np.maximum(jarak, 1) * 0.06

    for lbl in ["Baik", "Sedang", "Buruk"]:
        for benar, simbol, nama in [(True, "circle", "benar"),
(False, "x", "salah")]:
            mask = (label_aktual == lbl) & (pred_benar == benar)
            if not mask.any():
                continue
            fig.add_trace(go.Scatter(
x=jarak_jitter[mask], y=rssi_teor[mask],
mode='markers',
marker=dict(symbol=simbol, size=10 if simbol ==
"circle" else 11,
color=LABEL_COLOR[lbl],
line=dict(width=1.5, color="white")
if simbol == "circle"
else dict(width=2,
color=LABEL_COLOR[lbl])),
name=f"{lbl} (RF {nama})",
))

    fig.update_layout(
title="Distribusi Data Test Baru: Jarak vs RSSI Teori",
xaxis_title="Jarak (meter)", yaxis_title="RSSI Teori
(dBm)",
paper_bgcolor='#ffffff', plot_bgcolor='#f5f7fa',
legend=dict(orientation="h", y=-0.22,
font=dict(size=10)),
margin=dict(t=40, b=90), height=380,
)
    return fig

def create_test_overlay_figure(df_test, pred_rf):
    """
    Overlay data test di atas peta denah yang sedang aktif.

    Untuk setiap kombinasi unik (distance, wall_count) di dataset
    test:
    - Tempatkan titik pada jarak proporsional dari AP terkuat
    (konversi meter -> piksel)
    - Disebarkan secara radial di berbagai sudut (wall_count
    lebih banyak -> sudut lebih jauh)
    - Warna = label aktual mayoritas di kombinasi itu
    - Ukuran = proporsional jumlah data di kombinasi itu
```



© 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

```

- Tanda X = RF majority-salah di kombinasi itu; bulat =
majority-benar
- Caption menyebut bahwa posisi adalah proyeksi (bukan
koordinat ukur asli)
"""
fig = go.Figure()

# — Latar belakang: gambar denah

if state.uploaded_image is not None:
    _, cs = state.uploaded_image.split(',')
    img = Image.open(io.BytesIO(base64.b64decode(cs)))
    fig.add_layout_image(dict(
        source=img, x=0, y=state.img_height,
        sizex=state.img_width, sizey=state.img_height,
        xref="x", yref="y", sizing="stretch", layer="below",
    ))

# — Heatmap RSSI sebagai konteks (reuse fungsi yang sudah
ada)
grid_x = np.linspace(0, state.img_width, 60)
grid_y = np.linspace(0, state.img_height, 60)
gx, gy = np.meshgrid(grid_x, grid_y)
if state.ap_x or state.ext_ap_x:
    rssi_grid = hitung_rssi_grid(gx, gy, 'rssi')
    rssi_mask = rssi_grid.copy()
    rssi_grid[rssi_mask < -90] = np.nan
    fig.add_trace(go.Contour(
        x=grid_x, y=grid_y, z=rssi_grid,
        zmin=-90, zmax=-35,
        colorscale=[[0,"red"],[0.182,"red"],[0.182,"yellow"],
                    [0.455,"yellow"],[0.455,"green"],[1,"gree
n"]],
        contours=dict(showlines=False), opacity=0.35,
        showscale=False, hoverinfo='skip',
    ))

# — Tembok

active = state.walls + (state.auto_walls if
state.use_auto_walls else [])
for w in active:
    (x1, y1), (x2, y2) = w[0], w[1]
    fig.add_trace(go.Scatter(
        x=[x1, x2], y=[y1, y2], mode='lines',
        line=dict(color='black', width=2),
        showlegend=False, hoverinfo='skip',
    ))

# — AP

all_ap_x = state.ap_x + state.ext_ap_x
all_ap_y = state.ap_y + state.ext_ap_y
if all_ap_x:
    fig.add_trace(go.Scatter(
        x=all_ap_x, y=all_ap_y, mode='markers',
        marker=dict(size=14, color='red', symbol='circle',
                    line=dict(color='white', width=2)),

```



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

```

        name="AP", showlegend=True,
    ))

    # — Tidak ada AP → kembalikan figure kosong

    if not all_ap_x:
        fig.update_layout(
            title="Overlay Data Test (belum ada AP aktif)",
            xaxis=dict(range=[0, state.img_width],
                visible=False),
            yaxis=dict(range=[0, state.img_height],
                visible=False, scaleanchor="x"),
            height=420, paper_bgcolor='#ffffff',
            plot_bgcolor='#f5f7fa',
            margin=dict(t=36, b=10, l=10, r=10),
        )
        return fig

    # — Hitung posisi overlay per kombinasi (distance, wall_count) —
    # Pilih AP terkuat sebagai pusat referensi
    ap_ref_x = all_ap_x[0]
    ap_ref_y = all_ap_y[0]

    df = pd.DataFrame(df_test)
    label_aktual = np.asarray(df['label'].astype(str))
    pred = np.asarray(pred_rf)
    jarak_m = np.asarray(df['distance'].astype(float))
    wc = np.asarray(df['wall_count'].astype(int))

    kombis = sorted(set(zip(jarak_m.tolist(), wc.tolist())))
    n_kombi = len(kombis)

    # Sudut awal per wall_count (pisahkan supaya tidak tumpuk)
    wc_unik = sorted(set(wc.tolist()))
    sudut_base = {w: i * (2 * np.pi / max(len(wc_unik), 1)) for
        i, w in enumerate(wc_unik)}

    # Untuk setiap kombinasi unik (distance, wall_count), cari sudut yang unik
    sudut_per_kombi = {}
    wc_counter = {w: 0 for w in wc_unik}
    n_per_wc = {}
    for d, w in kombis:
        n_per_wc[w] = n_per_wc.get(w, 0) + 1
    for d, w in kombis:
        idx = wc_counter[w]
        total = n_per_wc[w]
        sudut = sudut_base[w] + idx * (2 * np.pi / max(total, 1))
        * 0.9
        sudut_per_kombi[(d, w)] = sudut
        wc_counter[w] += 1

    for (d_val, wc_val), sudut in sudut_per_kombi.items():
        mask = (jarak_m == d_val) & (wc == wc_val)
        if not mask.any():
            continue

    lbl_group = label_aktual[mask]

```



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

```

pred_group = pred[mask]
n = mask.sum()

# Label mayoritas aktual (untuk hover info)
unik, counts = np.unique(lbl_group, return_counts=True)
lbl_mayor = unik[np.argmax(counts)]

# Prediksi RF mayoritas per grup -> warna
unik_rf, counts_rf = np.unique(pred_group,
return_counts=True)
pred_mayor = unik_rf[np.argmax(counts_rf)]

# RF majority benar atau salah
acc_group = (pred_group == lbl_group).mean()
rf_benar_mayor = acc_group >= 0.5

# Konversi meter -> piksel
px = d_val / state.scale_meter if state.scale_meter > 0
else d_val * 20
ox = ap_ref_x + px * np.cos(sudut)
oy = ap_ref_y + px * np.sin(sudut)

n_counts = [(jarak_m == dm) & (wc == wm) for dm, wm in
sudut_per_kombi]
n_max = max(int(mc.sum()) for mc in n_counts) if n_counts
else n
size = int(14 + min((n / max(n_max, 1)) * 22, 22))

simbol = "circle" if rf_benar_mayor else "x"
warna = LABEL_COLOR.get(pred_mayor, "#9E9E9E") # warna
dari PREDIKSI RF

fig.add_trace(go.Scatter(
x=[ox], y=[oy], mode='markers+text',
marker=dict(symbol=simbol, size=size, color=warna,
opacity=0.85,
line=dict(width=2, color='white')),
text=[f"{int(d_val)}m/w{int(wc_val)}<br>n={n}"],
textposition="top center",
textfont=dict(size=9, color="#263238"),
name=f"RF:{pred_mayor}
({int(d_val)}m,{int(wc_val)}tmb)",
showlegend=False,
hovertemplate=
f"<b>{int(d_val)} m, {int(wc_val)}
tembok</b><br>"
f"n data: {n}<br>"
f"Prediksi RF mayoritas: <b>{pred_mayor}</b><br>"
f"Label aktual mayoritas: {lbl_mayor}<br>"
f"Akurasi RF di grup ini:
{acc_group*100:.0f}%<br>"
f"<i>Posisi adalah proyeksi dari AP, bukan
koordinat ukur asli</i>"
"<extra></extra>"
),
))

fig.update_layout(
title="Overlay Proyeksi Data Test di Atas Peta Aktif",

```



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

```

        xaxis=dict(range=[0, state.img_width], visible=False),
        yaxis=dict(range=[0, state.img_height], visible=False,
scaleanchor="x"),
        paper_bgcolor='#ffffff', plot_bgcolor='#f5f7fa',
        margin=dict(t=36, b=10, l=10, r=10), height=420,
        legend=dict(orientation="h", y=-0.05),
    )
    return fig

def get_grid_rssi_for_ap(ax, ay, gx, gy, current_walls,
base_power=None):
    """
    base_power: dBm pada referensi d0=1m yang dipakai sebagai
    basis perhitungan.
    - None -> AP simulasi: pakai state.tx_power - PL_D0
    (rumus teori penuh)
    - angka -> Existing AP dengan RSSI terukur: dipakai
    sebagai basis langsung
    (konsisten dengan hitung_rssi() di
    functions.py)
    """
    d_pixel = np.sqrt((gx-ax)**2+(gy-ay)**2)
    d_meter = np.maximum(d_pixel*state.scale_meter, 1)

    if base_power is not None:
        rssi = base_power - 26*np.log10(d_meter)
    else:
        rssi = state.tx_power - 50 - 26*np.log10(d_meter)

    for w in current_walls:
        (wx1,wy1),(wx2,wy2) = w[0], w[1]
        A_x,A_y = ax,ay; B_x,B_y = gx,gy
        C_x,C_y = wx1,wy1; D_x,D_y = wx2,wy2

        ccw_ACD = (D_y-A_y)*(C_x-A_x) > (C_y-A_y)*(D_x-A_x)
        ccw_BCD = (D_y-B_y)*(C_x-B_x) > (C_y-B_y)*(D_x-B_x)
        ccw_ABC = (C_y-A_y)*(B_x-A_x) > (B_y-A_y)*(C_x-A_x)
        ccw_ABD = (D_y-A_y)*(B_x-A_x) > (B_y-A_y)*(D_x-A_x)

        intersect = (ccw_ACD != ccw_BCD) & (ccw_ABC != ccw_ABD)
        loss = loss_for_wall(w)

        if isinstance(intersect, np.ndarray): rssi[intersect] -=
loss
        elif intersect: rssi -= loss

    return np.clip(rssi, -140, -10)

def get_grid_wallcount_for_ap(ax, ay, gx, gy, current_walls):
    """Hitung jumlah tembok yang terlewati dari satu AP ke setiap
    titik grid (gx, gy)."""
    count = np.zeros(gx.shape, dtype=int)
    for w in current_walls:
        (wx1,wy1),(wx2,wy2) = w[0], w[1]
        A_x,A_y = ax,ay; B_x,B_y = gx,gy
        C_x,C_y = wx1,wy1; D_x,D_y = wx2,wy2

        ccw_ACD = (D_y-A_y)*(C_x-A_x) > (C_y-A_y)*(D_x-A_x)

```



© 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

```

ccw_BCD = (D_y-B_y)*(C_x-B_x) > (C_y-B_y)*(D_x-B_x)
ccw_ABC = (C_y-A_y)*(B_x-A_x) > (B_y-A_y)*(C_x-A_x)
ccw_ABD = (D_y-A_y)*(B_x-A_x) > (B_y-A_y)*(C_x-A_x)
intersect = (ccw_ACD != ccw_BCD) & (ccw_ABC != ccw_ABD)

    if isinstance(intersect, np.ndarray): count[intersect] +=
1
        elif intersect: count += 1
    return count

def hitung_kualitas_grid(gx, gy, mode='ml'):
    """
    Hitung label kualitas (Baik/Sedang/Buruk) untuk setiap titik
    grid,
    berdasarkan AP yang sedang ditempatkan di simulasi saat ini.

    mode: 'ml' -> klasifikasi Random Forest (fitur:
    rssi_teori, distance, wall_count)
    'teori' -> threshold RSSI langsung (IEEE 802.11-2020),
    tanpa ML

    Return: array string (Baik/Sedang/Buruk), shape sama dengan
    gx/gy. '-' jika belum ada AP.
    """
    all_ap_x = state.ap_x + state.ext_ap_x
    all_ap_y = state.ap_y + state.ext_ap_y
    if not all_ap_x:
        return np.full(gx.shape, "-", dtype=object)

    active = state.walls + (state.auto_walls if
    state.use_auto_walls else [])
    n_ap = len(state.ap_x)

    rssi_layers, wc_layers, d_layers = [], [], []
    for i, (ax, ay) in enumerate(zip(all_ap_x, all_ap_y)):
        base_power = None
        if i >= n_ap and getattr(state, 'ext_ap_power_source',
        'teori') == 'terukur':
            ei = i - n_ap
            meas = getattr(state, 'ext_ap_measured', [])
            if ei < len(meas) and meas[ei] is not None:
                base_power = meas[ei]
        rssi_layers.append(get_grid_rssi_for_ap(ax, ay, gx, gy,
        active, base_power=base_power))
        wc_layers.append(get_grid_wallcount_for_ap(ax, ay, gx,
        gy, active))
        d_pixel = np.sqrt((gx-ax)**2 + (gy-ay)**2)
        d_layers.append(np.maximum(d_pixel * state.scale_meter,
        1))

    rssi_stack = np.stack(rssi_layers, axis=0) # (n_ap, H,
    W)
    best_idx = np.argmax(rssi_stack, axis=0) # (H, W) -
    - AP terkuat per titik
    best_rssi = np.take_along_axis(rssi_stack, best_idx[None, :,
    :], axis=0)[0]
    wc_stack = np.stack(wc_layers, axis=0)
    best_wc = np.take_along_axis(wc_stack, best_idx[None, :,
    :], axis=0)[0]

```



© 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

```

d_stack = np.stack(d_layers, axis=0)
best_d = np.take_along_axis(d_stack, best_idx[None, :, :], axis=0)[0]

flat_rssi = best_rssi.flatten()
flat_wc = best_wc.flatten()
flat_d = best_d.flatten()

if mode == 'teori':
    labels_flat = np.where(
        flat_rssi >= -65, "Baik",
        np.where(flat_rssi >= -80, "Sedang", "Buruk")
    )
else:
    from app import rf_model, le, FEATURES, ML_LOADED, DIST_BUCKETS
    if not ML_LOADED or rf_model is None:
        labels_flat = np.where(
            flat_rssi >= -65, "Baik",
            np.where(flat_rssi >= -80, "Sedang", "Buruk")
        )
    else:
        buckets = np.array(DIST_BUCKETS)
        idx_bucket = np.argmin(np.abs(flat_d[:, None] - buckets[None, :]), axis=1)
        dist_bucketed = buckets[idx_bucket]

        MAT_CODE = {"kaca": 0, "kayu": 1, "bata": 2, "beton": 3}
        mat_codes_sim = [MAT_CODE.get(wall_material(w), 2) for w in active]
        from collections import Counter
        wm_dom = Counter(mat_codes_sim).most_common(1)[0][0]
        if mat_codes_sim else 2

        # RF memakai fitur fisik MURNI - rssi_teorI tidak dimasukkan
        # supaya heatmap benar-benar dari ML, bukan dari Log-Distance
        feature_values = {
            "distance": dist_bucketed,
            "wall_count": flat_wc,
            "wall_material": np.full(len(flat_rssi), wm_dom, dtype=int),
            "wall_full": np.ones(len(flat_rssi), dtype=int),
        }
        X_grid = np.column_stack([feature_values.get(f, np.zeros(len(flat_rssi))) for f in FEATURES])
        pred_int = rf_model.predict(X_grid)
        labels_flat = le.inverse_transform(pred_int)

    return labels_flat.reshape(gx.shape)

KUALITAS_NUM = {"Baik": 2, "Sedang": 1, "Buruk": 0}
KUALITAS_COLORSCALE = [[0, "#F44336", [0.5, "#F44336", [0.5, "#FF9800", [0.75, "#FF9800", [0.75, "#4CAF50", [1, "#4CAF50"]]]]]]]

```



© 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

```
def create_kualitas_comparison_figure(mode, judul):
    """Figure ringkas: denah + tembok + AP + Rx + heatmap
    klasifikasi diskrit (Teori/RF)."""
    fig = go.Figure()

    if state.uploaded_image is not None:
        _, cs = state.uploaded_image.split(',')
        img = Image.open(io.BytesIO(base64.b64decode(cs)))
        fig.add_layout_image(dict(
            source=img, x=0, y=state.img_height,
            sizex=state.img_width, sizey=state.img_height,
            xref="x", yref="y", sizing="stretch", layer="below",
        ))

        grid_x = np.linspace(0, state.img_width, 60)
        grid_y = np.linspace(0, state.img_height, 60)
        gx, gy = np.meshgrid(grid_x, grid_y)

        labels = hitung_kualitas_grid(gx, gy, mode=mode)
        z = np.vectorize(lambda l: KUALITAS_NUM.get(l, -1))(labels).astype(float)
        z[z < 0] = np.nan

        fig.add_trace(go.Heatmap(
            x=grid_x, y=grid_y, z=z, zmin=0, zmax=2,
            colorscale=KUALITAS_COLORSCALE, showscale=False,
            opacity=0.65,
        ))

        active = state.walls + (state.auto_walls if
            state.use_auto_walls else [])
        for w in active:
            (x1,y1),(x2,y2) = w[0], w[1]
            fig.add_trace(go.Scatter(x=[x1,x2], y=[y1,y2],
            mode='lines',
            line=dict(color='black', width=2), showlegend=False,
            hoverinfo='skip'))

            all_ap_x = state.ap_x + state.ext_ap_x
            all_ap_y = state.ap_y + state.ext_ap_y
            if all_ap_x:
                fig.add_trace(go.Scatter(x=all_ap_x, y=all_ap_y,
                mode='markers',
                marker=dict(size=11, color='red', symbol='circle',
                    line=dict(color='white', width=1.5)),
                name="AP", showlegend=False))

        if state.rx_data:
            rx_x = [d[0] for d in state.rx_data]
            rx_y = [d[1] for d in state.rx_data]
            rx_txt = [f"Rx {i+1}" for i in range(len(state.rx_data))]
            fig.add_trace(go.Scatter(x=rx_x, y=rx_y,
            mode='markers+text',
            marker=dict(size=9, color='white', symbol='circle',
                line=dict(color='#1565C0', width=2)),
            text=rx_txt, textposition="top center",
            textfont=dict(size=9),
            showlegend=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 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

```

fig.update_layout(
    title=dict(text=judul, font=dict(size=13)),
    xaxis=dict(range=[0, state.img_width], visible=False),
    yaxis=dict(range=[0, state.img_height], visible=False,
scaleanchor="x"),
    paper_bgcolor='#ffffff', plot_bgcolor='#f5f7fa',
    margin=dict(t=36, b=10, l=10, r=10),
    height=320,
)
return fig

# — Konversi label kualitas ↔ angka untuk interpolasi

LABEL_NUM = {"Baik": 2.0, "Sedang": 1.0, "Buruk": 0.0, "Blind
Spot": -1.0}
NUM_RANGES = [(1.5, "Baik"), (0.5, "Sedang"), (-0.5, "Buruk")]

def _label_to_num(label):
    return LABEL_NUM.get(label, 0.0)

def _num_to_label(val):
    for threshold, lbl in NUM_RANGES:
        if val >= threshold:
            return lbl
    return "Buruk"

def buat_heatmap_interpolasi_ml(gx, gy):
    """
    Heatmap coverage berbasis interpolasi prediksi RF dengan
    wall-awareness
    dan radius filter.

    PILIHAN A – Radius filter:
    Titik grid yang jaraknya lebih dari RADIUS_METER meter dari
    Rx terdekat
    dikembalikan sebagai None (tidak ditampilkan / transparan di
    peta).
    Ini mencegah ekstrapolasi ke area yang tidak ada pengukurannya
    sama sekali,
    sehingga heatmap hanya menampilkan area yang memang ada
    datanya.

    RADIUS_METER bisa diubah di state.heatmap_radius (default 10
    meter).
    """
    if not state.rx_data:
        return None, False

    pts_x, pts_y, pts_val = [], [], []
    for d in state.rx_data:
        params = d[2]
        if not isinstance(params, dict):
            continue
        kml = params.get("kualitas_ml", "-")
        if kml in ("-", None):
            continue
        pts_x.append(float(d[0]))
        pts_y.append(float(d[1]))

```



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

```

pts_val.append(_label_to_num(kml))

if len(pts_x) < 3:
    return None, False

from functions import line_intersect
active = state.walls + (state.auto_walls if
state.use_auto_walls else [])

pts_x    = np.array(pts_x)
pts_y    = np.array(pts_y)
pts_val  = np.array(pts_val)

gx_flat  = gx.flatten()
gy_flat  = gy.flatten()
n_grid   = len(gx_flat)
n_pts    = len(pts_x)

# Radius filter - konversi meter ke piksel
RADIUS_METER = getattr(state, 'heatmap_radius', 10.0)
scale = state.scale_meter if state.scale_meter > 0 else 0.05
RADIUS_PX = RADIUS_METER / scale # meter → piksel

# Hitung jarak dari setiap titik grid ke Rx terdekat (dalam
piksel)
# Shape: (n_grid, n_pts)
dx = gx_flat[:, None] - pts_x[None, :]
dy = gy_flat[:, None] - pts_y[None, :]
dist_to_rx = np.sqrt(dx**2 + dy**2) #
(n_grid, n_pts)
dist_nearest = dist_to_rx.min(axis=1) #
(n_grid,)

# Mask: True = terlalu jauh dari semua Rx → tidak ditampilkan
mask_jauh = dist_nearest > RADIUS_PX

WALL_PENALTY = 2.0
z_flat = np.full(n_grid, np.nan) # default NaN = transparan

for gi in range(n_grid):
    if mask_jauh[gi]:
        continue # skip - di luar radius semua Rx, biarkan
NaN

gx_i, gy_i = gx_flat[gi], gy_flat[gi]
weights = np.zeros(n_pts)

for pi in range(n_pts):
    px, py = pts_x[pi], pts_y[pi]
    dist = np.sqrt((gx_i - px)**2 + (gy_i - py)**2)
    if dist < 1e-6:
        weights = np.zeros(n_pts)
        weights[pi] = 1.0
        break

n_walls = 0
for w in active:
    try:

```



© 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

```

        if line_intersect((gx_i, gy_i), (px, py),
w[0], w[1]):
            n_walls += 1
        except Exception:
            pass

        idw = 1.0 / max(dist, 1.0) ** 2
        penalty = np.exp(-WALL_PENALTY * n_walls)
        weights[pi] = idw * penalty

    total_w = weights.sum()
    if total_w > 0:
        z_flat[gi] = np.dot(weights, pts_val) / total_w

z = z_flat.reshape(gx.shape)

# Konversi angka ke label - NaN → None (tidak ditampilkan)
def to_label(val):
    if np.isnan(val):
        return None
    return _num_to_label(val)

labels_grid = np.vectorize(to_label, otypes=[object])(z)
return labels_grid, True

def buat_heatmap_ml_coverage(gx, gy):
    """
    Heatmap ML Coverage: prediksi RF langsung dari AP ke setiap
    titik grid.

    Berbeda dari buat_heatmap_interpolasi_ml (yang interpolasi
    dari Rx),
    fungsi ini menghitung fitur kondisi fisik dari AP ke setiap
    titik grid
    lalu langsung menjalankan prediksi RF - persis seperti Teori
    tapi
    menggunakan model Random Forest, bukan formula.

    Hasilnya: heatmap menyebar dari AP, tidak memerlukan titik
    Rx.
    Bila ada beberapa AP, setiap titik grid mendapat prediksi
    dari AP
    yang paling mungkin memberikan sinyal Baik (probability
    tertinggi).
    """
    from app import rf_model, le, FEATURES, ML_LOADED,
DIST_BUCKETS
    if not ML_LOADED or not (state.ap_x or state.ext_ap_x):
        return None

    gx_flat = gx.flatten()
    gy_flat = gy.flatten()
    n_grid = len(gx_flat)

    buckets = np.array(DIST_BUCKETS)
    active = state.walls + (state.auto_walls if
state.use_auto_walls else [])

    MAT_CODE = {"kaca": 0, "kayu": 1, "bata": 2, "beton": 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 mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

from collections import Counter
from functions import line_melalui_dead_zone
mat_codes = [MAT_CODE.get(wall_material(w), 2) for w in
active]
wm_dom = Counter(mat_codes).most_common(1)[0][0] if mat_codes
else 2
wf_dom = 1

# Gabung semua AP (new + existing)
all_ap_x = list(state.ap_x) + list(state.ext_ap_x)
all_ap_y = list(state.ap_y) + list(state.ext_ap_y)

idx_baik = list(le.classes_).index("Baik")

# Untuk setiap grid point, ambil probabilitas Baik tertinggi
dari semua AP
best_prob_baik = np.full(n_grid, -1.0)
best_pred_idx = np.zeros(n_grid, dtype=int)

for ax, ay in zip(all_ap_x, all_ap_y):
    d_px = np.sqrt((gx_flat - ax)**2 + (gy_flat - ay)**2)
    d_meter = np.maximum(d_px * state.scale_meter, 1)
    idx_b = np.argmin(np.abs(d_meter[:, None] -
buckets[None, :]), axis=1)
    dist_b = buckets[idx_b]

# Hitung wall_count per titik grid
wc = np.zeros(n_grid, dtype=int)
for w in active:
    for i in range(n_grid):
        try:
            if
line_intersect((ax, ay), (gx_flat[i], gy_flat[i]), w[0], w[1]):
                wc[i] += 1
            except Exception:
                pass

X = np.column_stack([
    dist_b if f == "distance" else
    wc if f == "wall_count" else
    np.full(n_grid, wm_dom) if f == "wall_material" else
    np.full(n_grid, wf_dom)
    for f in FEATURES
])

proba = rf_model.predict_proba(X) # (n_grid, n_classes)
prob_baik = proba[:, idx_baik]

# Perbarui prediksi untuk titik yang dapat sinyal lebih
baik dari AP ini
update_mask = prob_baik > best_prob_baik
best_prob_baik[update_mask] = prob_baik[update_mask]
best_pred_idx[update_mask] =
np.argmax(proba[update_mask], axis=1)

# Paksa Buruk untuk titik yang jalurnya melewati dead zone
for i in range(n_grid):
    for ax, ay in zip(all_ap_x, all_ap_y):

```



© 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

```

        if line_melalui_dead_zone(ax, ay, gx_flat[i],
gy_flat[i]) > 0:
            best_pred_idx[i] =
list(le.classes_).index("Buruk")
            break

labels_flat = le.inverse_transform(best_pred_idx)
return labels_flat.reshape(gx.shape)

def hitung_rssi_grid(gx, gy, param name):
    all_ap_x = state.ap_x + state.ext_ap_x
    all_ap_y = state.ap_y + state.ext_ap_y
    if not all_ap_x:
        return np.full_like(gx, -100, dtype=float)

    active = state.walls + (state.auto_walls if
state.use_auto_walls else [])
    n_ap = len(state.ap_x)

    rssi_layers = []
    for i, (ax, ay) in enumerate(zip(all_ap_x, all_ap_y)):
        base_power = None
        if i >= n_ap and getattr(state, 'ext_ap_power_source',
'teori') == 'terukur':
            ei = i - n_ap
            meas = getattr(state, 'ext_ap_measured', [])
            if ei < len(meas) and meas[ei] is not None:
                base_power = meas[ei]
            rssi_layers.append(get_grid_rssi_for_ap(ax, ay, gx, gy,
active, base_power=base_power))
        max_rssi = np.max(rssi_layers, axis=0)

    if param_name == 'rssi': return max_rssi
    noise = getattr(state, 'noise_floor', -90)
    snr = max_rssi - noise
    if param_name == 'snr': return snr
    if param_name == 'throughput': return np.maximum(0, snr*2)
    if param_name == 'path_loss': return state.tx_power -
max_rssi
    return max_rssi

def create_figure(toggle_heatmap='on', param_name='rssi'):
    fig = go.Figure()

    if state.uploaded_image is not None:
        _, cs = state.uploaded_image.split(',')
        img = Image.open(io.BytesIO(base64.b64decode(cs)))
        fig.add_layout_image(dict(
            source=img, x=0, y=state.img_height,
            sizex=state.img_width, sizey=state.img_height,
            xref="x", yref="y", sizing="stretch", layer="below",
        ))
        fig.update_xaxes(range=[0, state.img_width],
constrain='domain')
        fig.update_yaxes(range=[0, state.img_height],
scaleanchor="x")

    grid_x = np.linspace(0, state.img_width, 80)
    grid_y = np.linspace(0, state.img_height, 80)

```



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

```

gx, gy = np.meshgrid(grid_x, grid_y)

# Heatmap
if len(state.ap_x)+len(state.ext_ap_x) > 0 and toggle_heatmap
== 'on':

    mode_aktif = getattr(state, 'kualitas_mode', 'ml')

    # — Mode ML Interpolasi: dari titik Rx

    if mode_aktif == 'ml':
        if getattr(state, 'heatmap_locked', False) and
state.heatmap_locked_grid is not None:
            labels_interp, cukup = state.heatmap_locked_grid,
True
        else:
            labels_interp, cukup =
buat_heatmap_interpolasi_ml(gx, gy)
            state.heatmap_locked_grid = labels_interp if
cukup else None

    # — Mode ML Coverage: prediksi langsung dari AP

    elif mode_aktif == 'ml_coverage':
        labels_cov = buat_heatmap_ml_coverage(gx, gy)
        if labels_cov is not None:
            z_cov = np.vectorize(
                lambda l: KUALITAS_NUM.get(l, np.nan)
            )(labels_cov).astype(float)
            fig.add_trace(go.Heatmap(
                x=grid_x, y=grid_y, z=z_cov,
                zmin=0, zmax=2,
                colorscale=KUALITAS_COLORSCALE,
                showscale=False, opacity=0.65,
                hoverinfo='skip',
            ))
            judul_mode = "ML Coverage (prediksi RF dari AP)"
        else:
            judul_mode = "ML Coverage (tempatkan AP terlebih
dahulu)"
            labels_interp, cukup = None, True # skip fallback
ke Teori

    else:
        labels_interp, cukup = None, False

    if mode_aktif == 'ml' and cukup:
        # Heatmap dari interpolasi prediksi RF
        z = np.vectorize(lambda l: KUALITAS_NUM.get(l,
np.nan) if l is not None else
np.nan)(labels_interp).astype(float)
        fig.add_trace(go.Heatmap(
            x=grid_x, y=grid_y, z=z,
            zmin=0, zmax=2,
            colorscale=KUALITAS_COLORSCALE,
            showscale=False, opacity=0.65,
            hoverinfo='skip',
        ))
        judul_mode = "ML (interpolasi dari titik Rx)"

```



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

```

else:
    # Heatmap Log-Distance (mode teori atau belum cukup
    titik Rx)
    z = hitung_rssi_grid(gx, gy, param_name)
    titles = {'rssi': 'RSSI (dBm)', 'snr': 'SNR (dB)',
              'throughput': 'Throughput
    (Mbps)', 'path_loss': 'Path Loss (dB)'}
    rssi_mask = hitung_rssi_grid(gx, gy, 'rssi')
    z[rssi_mask < -90] = np.nan

    if param_name == 'rssi':
        zmin, zmax = -90, -35
        cscale =
    [[0, "red"], [0.182, "red"], [0.182, "yellow"], [0.455, "yellow"],
      [0.455, "green"], [1, "green"]]
    elif param_name == 'snr':
        zmin, zmax = -5, 40
        cscale =
    [[0, "red"], [0.333, "red"], [0.333, "yellow"], [0.666, "yellow"],
      [0.666, "green"], [1, "green"]]
    elif param_name == 'throughput':
        zmin, zmax = -10, 80
        cscale =
    [[0, "red"], [0.333, "red"], [0.333, "yellow"], [0.666, "yellow"],
      [0.666, "green"], [1, "green"]]
    elif param_name == 'path_loss':
        zmin, zmax = 65, 110
        cscale =
    [[0, "green"], [0.333, "green"], [0.333, "yellow"], [0.666, "yellow"],
      [0.666, "red"], [1, "red"]]
    else:
        zmin, zmax = -100, 0
        cscale = [[0, "red"], [0.5, "yellow"], [1, "green"]]

    fig.add_trace(go.Contour(
        x=grid_x, y=grid_y, z=z,
        zmin=zmin, zmax=zmax, colorscale=cscale,
        contours=dict(showlines=False), opacity=0.6,
        colorbar=dict(title=titles.get(param_name, 'Value'
    )),
    ))

    if mode_aktif == 'ml' and not cukup:
        judul_mode = "ML (tambahkan min. 3 titik Rx untuk
    heatmap ML)"
    else:
        judul_mode = "Teori (Log-Distance)"

    # Skala
    if len(state.scale_points) == 2:
        (x1, y1), (x2, y2) = state.scale_points
        fig.add_trace(go.Scatter(x=[x1, x2], y=[y1, y2], mode='lines+
    text',
        line=dict(color='blue', width=3), name="Skala",
        text=["", "Skala"], textposition="top center"))

    # Lingkaran AP

```



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

```

for ax,ay in zip(state.ap_x+state.ext_ap_x,
state.ap_y+state.ext_ap_y):
    fig.add_shape(type="circle",x0=ax-100,y0=ay-
100,x1=ax+100,y1=ay+100,
                    line=dict(color="green"),opacity=0.3)

# Click layer
fig.add_trace(go.Scatter(x=gx.flatten(),y=gy.flatten(),mode='
markers',
                        marker=dict(size=5,opacity=0),hoverinfo='none',showlegend
=False))

# AP
fig.add_trace(go.Scatter(x=state.ap_x,y=state.ap_y,mode='mark
ers+text',
                        marker=dict(size=10,color='red'),
                        text=[f"AP {i+1}" for i in range(len(state.ap_x))],
                        textposition="top center",name="Access Point"))

# Existing AP
if state.ext_ap_x:
    fig.add_trace(go.Scatter(x=state.ext_ap_x,y=state.ext_ap_
y,mode='markers+text',
                        marker=dict(size=10,color='purple',symbol='star'),
                        text=[f"Ext AP {i+1}" for i in
range(len(state.ext_ap_x))],
                        textposition="top center",name="Existing AP"))

# Rx -- warna marker = kualitas ML (bukan heatmap)
# Ini memastikan konsistensi antara tabel dan visual di peta
KUALITAS_COLOR = {
    "Baik":      "#4CAF50",    # hijau
    "Sedang":    "#FF9800",    # oranye
    "Buruk":     "#F44336",    # merah
    "Blind Spot": "#9E9E9E",    # abu
    "-":         "#2196F3",    # biru (default, belum ada AP)
}

if state.rx_data:
    rx_x_list, rx_y_list = [], []
    rx_colors, rx_symbols = [], []
    rx_hover, rx_labels = [], []

    mode_aktif = getattr(state, 'kualitas_mode', 'ml')
    kunci_kualitas = 'kualitas_ml' if mode_aktif in
('ml','ml_coverage') else 'kualitas_teor'
    label_mode = "ML" if mode_aktif in ('ml','ml_coverage')
else "Teori"

    for i, d in enumerate(state.rx_data):
        params = d[2]
        rssi = params.get("rssi") if isinstance(params,
dict) else None
        snr = params.get("snr") if isinstance(params,
dict) else None
        # Fallback ke kualitas_ml kalau key kualitas_teor
belum ada (data lama)

```



© 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

```

        kml = params.get(kunci_kualitas,
        params.get("kualitas_ml", "-")) if isinstance(params, dict) else
        "-"

        jarak = d[3]
        terhubung = d[4] if len(d) > 4 else "-"

        rx_x_list.append(d[0])
        rx_y_list.append(d[1])
        rx_colors.append(KUALITAS_COLOR.get(kml, "#2196F3"))
        rx_labels.append(f"Rx {i+1}")

        hover = (
            f"<b>Rx {i+1}</b><br>"
            f"RSSI: {rssi:.1f} dBm<br>" if rssi else "RSSI: -
<br>"
        )
        hover += (
            f"SNR: {snr:.1f} dB<br>" if snr else "SNR: -
<br>"
        )
        hover += (
            f"Jarak: {jarak:.1f} m<br>" if jarak else "Jarak:
-<br>"
        )
        hover += f"Terhubung: {terhubung}<br>Kualitas
({label_mode}): <b>{kml}</b>"
        rx_hover.append(hover)

# Lingkaran luar berwarna kualitas (ring tebal)
fig.add_trace(go.Scatter(
    x=rx_x_list, y=rx_y_list,
    mode='markers',
    marker=dict(
        size=20,
        color=rx_colors,
        opacity=0.35,
        line=dict(width=0),
    ),
    hoverinfo='skip',
    showlegend=False,
))

# Titik inti + label
fig.add_trace(go.Scatter(
    x=rx_x_list, y=rx_y_list,
    mode='markers+text',
    marker=dict(
        size=12,
        color=rx_colors,
        symbol='circle',
        line=dict(color='white', width=2),
    ),
    text=rx_labels,
    textposition="top center",
    hovertext=rx_hover,
    hoverinfo='text',
    name="Receiver",
))

```



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

```
# -- Tembok: warna berbeda jika sedang dipilih untuk edit --
active = state.walls + (state.auto_walls if
state.use_auto_walls else [])
sel_idx = getattr(state, 'selected_wall_idx', None)
sel_source = getattr(state, 'selected_wall_source', None)

# — Dead Zones (ruangan basah/toilet)

for dz in getattr(state, 'dead_zones', []):
    (dx1, dy1), (dx2, dy2) = dz
    # Kotak fill merah transparan dengan border merah gelap
    fig.add_shape(type="rect",
                  x0=dx1, y0=dy1, x1=dx2, y1=dy2,
                  fillcolor="rgba(183,28,28,0.25)",
                  line=dict(color="rgba(183,28,28,0.9)", width=2,
dash="dashdot"),
                  layer="above",
                  )
    # Label "⊘ Dead Zone" di tengah kotak
    cx, cy = (dx1+dx2)/2, (dy1+dy2)/2
    fig.add_annotation(x=cx, y=cy, text="⊘ Dead Zone",
                      showarrow=False,
                      font=dict(size=11, color="rgba(183,28,28,0.9)",
family="sans-serif"),
                      bgcolor="rgba(255,255,255,0.6)",
                      borderpad=3,
                      )

for wi, w in enumerate(state.walls):
    (x1, y1), (x2, y2) = w[0], w[1]
    mat = wall_material(w)
    src = w[3] if len(w) > 3 else 'manual'
    wf = int(w[5]) if len(w) > 5 else 1 # wall_full
    flag
    is_sel = (sel_source == 'manual' and wi == sel_idx)
    color = '#FFD700' if is_sel else warna_material(mat)
    width = 5 if is_sel else 4
    dash_style = 'solid' if wf == 1 else 'dot' # setengah =
    putus-putus
    label_src = '[M]' if src == 'manual' else ''
    label_wf = '' if wf == 1 else '[%]'
    fig.add_trace(go.Scatter(x=[x1, x2], y=[y1, y2], mode='lines'
,
                      line=dict(color=color, width=width, dash=dash_style),
                      name=f"{mat}{label_src}{label_wf}", showlegend=False))

if state.use_auto_walls:
    for wi, w in enumerate(state.auto_walls):
        (x1, y1), (x2, y2) = w[0], w[1]
        mat = wall_material(w)
        src = w[3] if len(w) > 3 else 'color'
        conf = w[4] if len(w) > 4 else 1.0
        is_sel = (sel_source == 'auto' and wi == sel_idx)
        color = '#FFD700' if is_sel else warna_material(mat)
        width = 5 if is_sel else 4
        # Badge: C=color detected, T=thickness, M>manual
        badge =
        {'color': 'C', 'thickness': 'T', 'manual': 'M'}.get(src, '?')
```



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

```

fig.add_trace(go.Scatter(x=[x1,x2],y=[y1,y2],mode='lines',
                        line=dict(color=color,width=width,
                                dash='dash' if src=='thickness' else
                                'solid'),
                        name=f"{mat} [{badge}]",showlegend=False,
                        hovertext=f"{mat} | {src} | conf={conf}",
                        hoverinfo='text'))

mode_aktif_label = getattr(state, 'kualitas_mode', 'ml')
if len(state.ap_x)+len(state.ext_ap_x) > 0 and toggle_heatmap == 'on':
    pass # judul_mode sudah di-set di dalam blok if di atas
    elif mode_aktif_label == 'ml_coverage':
        judul_mode = "ML Coverage (tambahkan AP terlebih dahulu)"
    elif mode_aktif_label == 'ml':
        judul_mode = "ML (tambahkan AP terlebih dahulu)"
    else:
        judul_mode = "Teori (Log-Distance)"

fig.update_layout(
    title=f"Klik grafik untuk menambahkan titik | Mode: {judul_mode}",
    xaxis=dict(range=[0,state.img_width],gridcolor='#cfd8dc',
zerolinecolor='#cfd8dc'),
    yaxis=dict(range=[0,state.img_height],gridcolor='#cfd8dc',
zerolinecolor='#cfd8dc'),
    paper_bgcolor='#ffffff', plot_bgcolor='#f5f7fa',
    font=dict(color='#263238'),
)
return fig

def auto_placement_logic(num_ap):
    state.ap_x.clear(); state.ap_y.clear()
    grid_res = 20
    search_x = np.linspace(50, state.img_width-50, grid_res)
    search_y = np.linspace(50, state.img_height-50, grid_res)
    gx,gy = np.meshgrid(search_x, search_y)

    state.ap_x.append(state.img_width/2)
    state.ap_y.append(state.img_height/2)

    for _ in range(num_ap-1):
        rg = hitung_rssi_grid(gx,gy,'rssi')
        mi = np.unravel_index(np.argmin(rg), rg.shape)
        state.ap_x.append(gx[mi]); state.ap_y.append(gy[mi])

    final = hitung_rssi_grid(gx,gy,'rssi')
    worst = np.min(final)
    if worst < -65:
        rec = state.tx_power + (-65 - worst)
        return (f"AP {num_ap} diatur optimal. Titik terlemah: {worst:.1f} dBm. "
                f"Rekomendasi power: {rec:.1f} dBm.")
    return (f"AP {num_ap} diatur optimal. Sinyal terbaik (min {worst:.1f} dBm).")

def cari_posisi_ap_optimal_ml():
    """

```



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

Optimasi posisi semua AP secara BERURUTAN berbasis prediksi *Random Forest*.

Titik Evaluasi:

- Kalau ada Rx yang dipasang user → pakai koordinat Rx sebagai titik evaluasi
(AP berpindah ke posisi terbaik untuk melayani pengguna-pengguna itu)
- Kalau tidak ada Rx → fallback ke 625 titik grid internal yang tersebar merata

Algoritma Sequential per AP:

1. Posisi AP lain dibiarkan tetap
2. Coba 400 kandidat posisi untuk AP ini
3. Skor = jumlah titik evaluasi yang diprediksi RF sebagai Baik,
dengan mengambil sinyal TERBAIK dari semua AP (perangkat selalu

terkoneksi ke AP terkuat)

4. Pindahkan AP ke kandidat dengan skor tertinggi
5. Lanjut ke AP berikutnya

"""

```
from app import rf_model, le, FEATURES, ML_LOADED,
DIST_BUCKETS
```

```
if not ML_LOADED or rf_model is None:
```

```
    return 0, 0, 0, [], "Model RF belum dimuat"
```

```
if not state.ap_x:
```

```
    return 0, 0, 0, [], "Belum ada AP yang ditempatkan"
```

```
active = state.walls + (state.auto_walls if
state.use_auto_walls else [])
```

```
buckets = np.array(DIST_BUCKETS)
```

```
MAT_CODE = {"kaca": 0, "kayu": 1, "bata": 2, "beton": 3}
```

```
from collections import Counter
```

```
mat_codes_all = [MAT_CODE.get(wall_material(w), 2) for w in
active]
```

```
wm_dom = Counter(mat_codes_all).most_common(1)[0][0] if
mat_codes_all else 2
```

```
wf_dom = 1
```

```
# — Tentukan titik evaluasi
```

```
if state.rx_data:
```

```
    # Pakai posisi Rx yang dipasang user
```

```
    flat_ex = np.array([float(d[0]) for d in state.rx_data])
```

```
    flat_ey = np.array([float(d[1]) for d in state.rx_data])
```

```
    mode_eval = "rx"
```

```
else:
```

```
    # Fallback: grid internal 25x25
```

```
    N_EVAL = 25
```

```
    eval_x = np.linspace(30, state.img_width - 30, N_EVAL)
```

```
    eval_y = np.linspace(30, state.img_height - 30, N_EVAL)
```

```
    gx_e, gy_e = np.meshgrid(eval_x, eval_y)
```

```
    flat_ex = gx_e.flatten()
```

```
    flat_ey = gy_e.flatten()
```

```
    mode_eval = "grid"
```



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

```

total = len(flat_ex)

# Grid kandidat posisi AP
N_CAND = 20
cand_x = np.linspace(50, state.img_width - 50, N_CAND)
cand_y = np.linspace(50, state.img_height - 50, N_CAND)

def prediksi_satu_ap(ax, ay):
    """Probabilitas Baik dari satu AP ke semua titik
    evaluasi."""
    d_px = np.sqrt((flat_ex - ax)**2 + (flat_ey - ay)**2)
    d_meter = np.maximum(d_px * state.scale_meter, 1)
    idx_b = np.argmin(np.abs(d_meter[:,None] -
    buckets[None,:]), axis=1)
    dist_b = buckets[idx_b]

    wc = np.zeros(total, dtype=int)
    for w in active:
        for i in range(total):
            try:
                if
                line_intersect((ax,ay),(flat_ex[i],flat_ey[i]),w[0],w[1]):
                    wc[i] += 1
            except Exception:
                pass

    X = np.column_stack([
        dist_b          if f == "distance"      else
        wc              if f == "wall_count"    else
        np.full(total,wm_dom) if f == "wall_material" else
        np.full(total,wf_dom)
        for f in FEATURES
    ])
    proba = rf_model.predict_proba(X)
    idx_baik = list(le.classes_).index("Baik")
    return proba[:, idx_baik]

def hitung_skor_semua_ap(ap_xs, ap_ys):
    """Skor Baik terbaik dari semua AP per titik evaluasi."""
    if not ap_xs:
        return 0
    prob_best = np.zeros(total)
    for ax, ay in zip(ap_xs, ap_ys):
        prob_best = np.maximum(prob_best,
        prediksi_satu_ap(ax, ay))
    return int((prob_best >= 0.5).sum())

# — Skor awal

skor_awal = hitung_skor_semua_ap(state.ap_x, state.ap_y)
detail = []

# — Optimasi berurutan per AP

for i in range(len(state.ap_x)):
    skor_sebelum_i = hitung_skor_semua_ap(state.ap_x,
    state.ap_y)
    best_s = -1
    best_cx, best_cy = state.ap_x[i], state.ap_y[i]

```



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

```

        for cy in cand_y:
            for cx in cand_x:
                ap_xs_t = state.ap_x[:i] + [cx] +
state.ap_x[i+1:]
                ap_ys_t = state.ap_y[:i] + [cy] +
state.ap_y[i+1:]
                s = hitung_skor_semua_ap(ap_xs_t, ap_ys_t)
                if s > best_s:
                    best_s, best_cx, best_cy = s, cx, cy

            state.ap_x[i] = best_cx
            state.ap_y[i] = best_cy
            skor_sesudah_i = hitung_skor_semua_ap(state.ap_x,
state.ap_y)

            detail.append({
                "ap_idx" : i + 1,
                "sebelum" : skor_sebelum_i,
                "sesudah" : skor_sesudah_i,
                "pct_sbilm": skor_sebelum_i / total * 100,
                "pct_ssdh": skor_sesudah_i / total * 100,
            })

        skor_akhir = hitung_skor_semua_ap(state.ap_x, state.ap_y)
        return skor_awal, skor_akhir, total, detail, mode_eval
    from app import rf_model, le, FEATURES, ML_LOADED,
DIST_BUCKETS

    if not ML_LOADED or rf_model is None:
        return 0, 0, 0, [], "Model RF belum dimuat"
    if not state.ap_x:
        return 0, 0, 0, [], "Belum ada AP yang ditempatkan"

    active = state.walls + (state.auto_walls if
state.use_auto_walls else [])
    buckets = np.array(DIST_BUCKETS)

    MAT_CODE = {"kaca": 0, "kayu": 1, "bata": 2, "beton": 3}
    from collections import Counter
    mat_codes_all = [MAT_CODE.get(wall_material(w), 2) for w in
active]
    wm_dom = Counter(mat_codes_all).most_common(1)[0][0] if
mat_codes_all else 2
    wf_dom = 1

    # Grid evaluasi tetap untuk semua iterasi
    N_EVAL = 25
    eval_x = np.linspace(30, state.img_width - 30, N_EVAL)
    eval_y = np.linspace(30, state.img_height - 30, N_EVAL)
    gx_e, gy_e = np.meshgrid(eval_x, eval_y)
    flat_ex = gx_e.flatten()
    flat_ey = gy_e.flatten()
    total = len(flat_ex)

    N_CAND = 20
    cand_x = np.linspace(50, state.img_width - 50, N_CAND)
    cand_y = np.linspace(50, state.img_height - 50, N_CAND)

```



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

```
def prediksi_satu_ap(ax, ay):
    """Hitung prediksi RF dari satu AP ke semua titik
    evaluasi."""
    d_px = np.sqrt((flat_ex - ax)**2 + (flat_ey - ay)**2)
    d_meter = np.maximum(d_px * state.scale_meter, 1)
    idx_b = np.argmin(np.abs(d_meter[:,None] -
    buckets[None,:]), axis=1)
    dist_b = buckets[idx_b]

    wc = np.zeros(total, dtype=int)
    for w in active:
        for i in range(total):
            try:
                if
                line_intersect((ax,ay),(flat_ex[i],flat_ey[i]),w[0],w[1]):
                    wc[i] += 1
            except Exception:
                pass

    X = np.column_stack([
        dist_b if f == "distance" else
        wc if f == "wall_count" else
        np.full(total, wm_dom) if f == "wall_material" else
        np.full(total, wf_dom)
        for f in FEATURES
    ])
    # Kembalikan probabilitas Baik tiap titik (untuk
    perbandingan multi-AP)
    proba = rf_model.predict_proba(X)
    idx_baik = list(le.classes_).index("Baik")
    return proba[:, idx_baik] # shape (total,)

def hitung_skor_semua_ap(ap_xs, ap_ys):
    """
    Hitung skor Baik dengan mempertimbangkan semua AP.
    Setiap titik grid diambil dari AP yang memberikan
    probabilitas Baik tertinggi.
    """
    if not ap_xs:
        return 0
    prob_best = np.zeros(total)
    for ax, ay in zip(ap_xs, ap_ys):
        p = prediksi_satu_ap(ax, ay)
        prob_best = np.maximum(prob_best, p)
    return int((prob_best >= 0.5).sum())

# — Skor awal sebelum optimasi

skor_awal = hitung_skor_semua_ap(state.ap_x, state.ap_y)
detail = []

# — Optimasi berurutan per AP

for i in range(len(state.ap_x)):
    skor_sebelum_i = hitung_skor_semua_ap(state.ap_x,
    state.ap_y)
    best_s = -1
    best_cx, best_cy = state.ap_x[i], state.ap_y[i]
```



© 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

```

for cy in cand_y:
    for cx in cand_x:
        # Coba posisi kandidat untuk AP-i, AP lain tetap
        ap_xs_trial = state.ap_x[:i] + [cx] +
state.ap_x[i+1:]
        ap_ys_trial = state.ap_y[:i] + [cy] +
state.ap_y[i+1:]
        s = hitung_skor_semua_ap(ap_xs_trial,
ap_ys_trial)
        if s > best_s:
            best_s, best_cx, best_cy = s, cx, cy

# Pindahkan AP-i ke posisi terbaik
state.ap_x[i] = best_cx
state.ap_y[i] = best_cy
skor_sesudah_i = hitung_skor_semua_ap(state.ap_x,
state.ap_y)

detail.append({
    "ap_idx" : i + 1,
    "sebelum" : skor_sebelum_i,
    "sesudah" : skor_sesudah_i,
    "pct_sbilm" : skor_sebelum_i / total * 100,
    "pct_ssdh" : skor_sesudah_i / total * 100,
})

skor_akhir = hitung_skor_semua_ap(state.ap_x, state.ap_y)
return skor_awal, skor_akhir, total, detail, None

```

style.css

```

/* — CSS Variables (Light & Simple Theme) — */
:root {
    --primary: #0288d1;
    --primary-l: #0288d1;
    --primary-d: #01579b;
    --accent: #00b0ff;
    --bg-dark: #f5f7fa; /* Latar belakang terang */
    --bg-mid: #e4e8f0; /* Latar panel atas/nav */
    --bg-panel: #ffffff; /* Panel putih */
    --bg-card: #ffffff; /* Card putih */
    --border: #cfd8dc; /* Border abu terang */
    --text-main: #263238; /* Teks gelap/terlihat */
    --text-muted: #546e7a; /* Teks abu sedang */
    --text-hint: #78909c; /* Teks petunjuk */
    --green: #2e7d32;
    --red: #c62828;
    --purple: #651fff;
    --orange: #ef6c00;
    --radius: 8px;
    --shadow: 0 2px 10px rgba(0,0,0,0.05);
    --navbar-h: 54px;
}

/* — Reset — */
*, *::before, *::after {
    box-sizing: border-box;

```



© 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

```

margin: 0;
padding: 0;
}

body {
  font-family: 'Inter', sans-serif;
  background: var(--bg-dark);
  color: var(--text-main);
  font-size: 13px;
  min-height: 100vh;
}

/* =====
LOGIN PAGE
===== */

#login-page {
  min-height: 100vh;
  display: flex;
  align-items: center;
  justify-content: center;
  position: relative;
  overflow: hidden;
  background: radial-gradient(circle at 30% 30%, #eef2f7 0%,
#d9e2ec 100%);
}

/* Animasi lingkaran sinyal Wi-Fi di background */
.login-bg-overlay {
  position: absolute;
  inset: 0;
  pointer-events: none;
  overflow: hidden;
}

.login-bg-overlay::before,
.login-bg-overlay::after {
  content: '';
  position: absolute;
  border-radius: 50%;
  border: 1.5px solid rgba(2, 136, 209, 0.15);
  animation: pulseRing 4s ease-out infinite;
}

.login-bg-overlay::before {
  width: 600px; height: 600px;
  left: -100px; top: 50%;
  transform: translateY(-50%);
  animation-delay: 0s;
}

.login-bg-overlay::after {
  width: 800px; height: 800px;
  left: -200px; top: 50%;
  transform: translateY(-50%);
  animation-delay: 1.5s;
}

@keyframes pulseRing {

```



© 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

```

0% { opacity: 0.6; transform: translateY(-50%) scale(1); }
100% { opacity: 0; transform: translateY(-50%) scale(1.4); }
}

.login-card {
  position: relative;
  z-index: 10;
  background: #ffffff;
  border: 1px solid var(--border);
  border-radius: 12px;
  padding: 40px 36px;
  width: 420px;
  box-shadow: 0 8px 30px rgba(0,0,0,0.08);
}

.login-header {
  text-align: center;
  margin-bottom: 28px;
}

.login-title {
  font-size: 20px;
  font-weight: 700;
  color: var(--text-main);
  margin-top: 12px;
  letter-spacing: -0.3px;
}

.login-subtitle {
  font-size: 12px;
  color: var(--text-muted);
  margin-top: 4px;
}

/* — CSS-only WiFi icon — */
.wifi-icon-wrap {
  position: relative;
  width: 80px;
  height: 60px;
  margin: 0 auto 14px;
  display: flex;
  align-items: flex-end;
  justify-content: center;
}

.wifi-arc {
  position: absolute;
  border-radius: 50%;
  border: 6px solid transparent;
  border-top-color: var(--primary-1);
  bottom: 8px;
  left: 50%;
  transform: translateX(-50%) rotate(-45deg);
  opacity: 0;
  animation: wifiPulse 2.4s ease-in-out infinite;
}

.wifi-arc-1 {
  width: 28px; height: 28px;

```



© 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

```
border-color: #0288d1 transparent transparent transparent;
animation-delay: 0s;
}

.wifi-arc-2 {
width: 50px; height: 50px;
border-color: #29b6f6 transparent transparent transparent;
animation-delay: 0.3s;
}

.wifi-arc-3 {
width: 72px; height: 72px;
border-color: #4fc3f7 transparent transparent transparent;
animation-delay: 0.6s;
}

.wifi-dot {
position: absolute;
bottom: 4px;
left: 50%;
transform: translateX(-50%);
width: 9px; height: 9px;
border-radius: 50%;
background: #01579b;
animation: wifiDot 2.4s ease-in-out infinite;
}

@keyframes wifiPulse {
0% { opacity: 0; }
30% { opacity: 1; }
70% { opacity: 1; }
100% { opacity: 0; }
}

@keyframes wifiDot {
0%, 100% { background: #01579b; }
50% { background: #4fc3f7; box-shadow: 0 0 8px #4fc3f7; }
}

/* Auth Tabs */
.auth-tabs .tab {
background: transparent !important;
border: none !important;
color: var(--text-muted) !important;
padding: 10px 20px !important;
font-family: 'Inter', sans-serif !important;
font-size: 13px !important;
cursor: pointer;
border-bottom: 2px solid transparent !important;
transition: all 0.2s;
}

.auth-tabs .tab--selected,
.auth-tab-selected {
color: var(--primary-1) !important;
border-bottom: 2px solid var(--primary-1) !important;
background: transparent !important;
}
```



© 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

```
.auth-tabs .tab-container {
  border-bottom: 1px solid var(--border) !important;
  margin-bottom: 20px !important;
}

/* Form */
.auth-form {
  display: flex;
  flex-direction: column;
  gap: 14px;
  padding: 10px 0;
}

.input-group {
  display: flex;
  flex-direction: column;
  gap: 5px;
}

.input-label {
  font-size: 11px;
  font-weight: 600;
  color: var(--text-muted);
  text-transform: uppercase;
  letter-spacing: 0.5px;
}

.auth-input {
  background: #ffffff !important;
  border: 1px solid var(--border) !important;
  border-radius: 8px !important;
  padding: 10px 14px !important;
  color: var(--text-main) !important;
  font-family: 'Inter', sans-serif !important;
  font-size: 13px !important;
  width: 100% !important;
  transition: border-color 0.2s, box-shadow 0.2s;
  outline: none !important;
}

.auth-input:focus {
  border-color: var(--primary) !important;
  box-shadow: 0 0 0 3px rgba(2, 136, 209, 0.15) !important;
}

.auth-btn {
  border: none;
  border-radius: 8px;
  padding: 12px 20px;
  font-family: 'Inter', sans-serif;
  font-size: 14px;
  font-weight: 600;
  cursor: pointer;
  transition: all 0.2s;
  width: 100%;
}

.auth-btn-primary {
```



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

```

    background: linear-gradient(135deg, var(--primary) 0%, var(--
primary-d) 100%);
    color: white;
    box-shadow: 0 4px 15px rgba(2, 136, 209, 0.3);
  }

.auth-btn-primary:hover {
    transform: translateY(-1px);
    box-shadow: 0 6px 20px rgba(2, 136, 209, 0.45);
}

.auth-btn-success {
    background: linear-gradient(135deg, var(--green) 0%, #00796b
100%);
    color: white;
    box-shadow: 0 4px 15px rgba(38, 166, 154, 0.3);
}

.auth-btn-success:hover {
    transform: translateY(-1px);
    box-shadow: 0 6px 20px rgba(38, 166, 154, 0.45);
}

.auth-message {
    margin-top: 12px;
    padding: 8px 14px;
    border-radius: 8px;
    font-size: 12px;
    font-weight: 500;
    text-align: center;
    min-height: 14px;
}

.auth-error {
    background: rgba(239, 83, 80, 0.1);
    border: 1px solid rgba(239, 83, 80, 0.3);
    color: #ef9a9a;
}

.auth-success {
    background: rgba(38, 166, 154, 0.1);
    border: 1px solid rgba(38, 166, 154, 0.3);
    color: #80cbc4;
}

/*
NAVBAR
*/

.navbar {
    height: var(--navbar-h);
    background: linear-gradient(90deg, var(--bg-mid) 0%, var(--bg-
panel) 100%);
    border-bottom: 1px solid var(--border);
    display: flex;
    align-items: center;
    padding: 0 16px;
    gap: 12px;
    box-shadow: 0 2px 12px rgba(0,0,0,0.06);

```



© 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

```

position: sticky;
top: 0;
z-index: 1000;
}

.navbar-brand {
  display: flex;
  align-items: center;
  white-space: nowrap;
  min-width: 200px;
}

.navbar-title {
  font-size: 15px;
  font-weight: 700;
  color: var(--primary-1);
  letter-spacing: -0.2px;
}

.navbar-menu {
  display: flex;
  gap: 6px;
  flex: 1;
  justify-content: center;
  flex-wrap: wrap;
}

.navbar-user {
  display: flex;
  align-items: center;
  gap: 10px;
  min-width: 160px;
  justify-content: flex-end;
}

.navbar-username {
  font-size: 12px;
  color: var(--text-muted);
  white-space: nowrap;
}

.nav-btn {
  background: rgba(255,255,255,0.06);
  border: 1px solid var(--border);
  border-radius: 7px;
  padding: 6px 12px;
  color: var(--text-main);
  font-family: 'Inter', sans-serif;
  font-size: 12px;
  cursor: pointer;
  transition: all 0.18s;
  white-space: nowrap;
}

.nav-btn:hover {
  background: rgba(79, 195, 247, 0.12);
  border-color: var(--primary-1);
  color: var(--primary-1);
}

```



© 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

```
.nav-btn-logout {
  background: rgba(239, 83, 80, 0.1);
  border-color: rgba(239, 83, 80, 0.35);
  color: #ef9a9a;
}

.nav-btn-logout:hover {
  background: rgba(239, 83, 80, 0.2);
  border-color: var(--red);
  color: #ff8a80;
}

/* =====
MAIN PAGE — THREE PANEL LAYOUT
===== */

.main-content {
  display: flex;
  height: calc(100vh - var(--navbar-h));
  overflow: hidden;
}

/* — Panel Kiri ————— */
.panel-left {
  width: 210px;
  min-width: 210px;
  background: var(--bg-panel);
  border-right: 1px solid var(--border);
  padding: 14px 12px;
  overflow-y: auto;
  overflow-x: hidden;
  scrollbar-width: thin;
  scrollbar-color: var(--border) transparent;
}

.panel-left::-webkit-scrollbar {
  width: 4px;
}

.panel-left::-webkit-scrollbar-thumb {
  background: var(--border);
  border-radius: 2px;
}

/* — Panel Tengah ————— */
.panel-center {
  flex: 1;
  padding: 12px;
  overflow-y: auto;
  background: var(--bg-dark);
}

/* — Panel Kanan ————— */
.panel-right {
  width: 220px;
  min-width: 220px;
  background: var(--bg-panel);
}
```



© 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

```
border-left: 1px solid var(--border);
padding: 14px 12px;
overflow-y: auto;
scrollbar-width: thin;
scrollbar-color: var(--border) transparent;
}

.panel-right::-webkit-scrollbar {
  width: 4px;
}

.panel-right::-webkit-scrollbar-thumb {
  background: var(--border);
  border-radius: 2px;
}

/* — Panel Title — */
.panel-title {
  font-size: 11px !important;
  font-weight: 700 !important;
  color: var(--primary-1) !important;
  text-transform: uppercase;
  letter-spacing: 0.6px;
  margin-bottom: 8px !important;
  display: block;
}

h4.panel-title { /* override Dash default h4 */
  font-size: 11px;
  font-weight: 700;
  color: var(--primary-1);
  text-transform: uppercase;
  letter-spacing: 0.6px;
  margin-bottom: 8px;
}

.divider {
  border: none;
  border-top: 1px solid var(--border);
  margin: 10px 0;
}

/* — Upload Box — */
.upload-box {
  border: 1.5px dashed var(--border) !important;
  border-radius: 8px !important;
  padding: 12px 8px !important;
  text-align: center !important;
  cursor: pointer !important;
  background: rgba(255,255,255,0.03) !important;
  color: var(--text-muted) !important;
  font-size: 12px !important;
  transition: all 0.2s !important;
}

.upload-box:hover {
  border-color: var(--primary-1) !important;
  background: rgba(79, 195, 247, 0.06) !important;
  color: var(--primary-1) !important;
}
```



© 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

```

}

/* — Radio Groups ————— */
.radio-group {
  display: flex;
  gap: 10px;
  flex-wrap: wrap;
}

.radio-group-vertical {
  display: flex;
  flex-direction: column;
  gap: 5px;
}

/* Style for Dash RadioItems labels */
.radio-group label,
.radio-group-vertical label {
  color: var(--text-main) !important;
  font-size: 12px !important;
  font-family: 'Inter', sans-serif !important;
  cursor: pointer;
  display: flex;
  align-items: center;
  gap: 5px;
}

/* — Inputs & Dropdowns ————— */
.ctrl-input {
  width: 100% !important;
  background: #ffffff !important;
  border: 1px solid var(--border) !important;
  border-radius: 6px !important;
  padding: 7px 10px !important;
  color: var(--text-main) !important;
  font-family: 'Inter', sans-serif !important;
  font-size: 12px !important;
  margin-bottom: 6px;
  outline: none !important;
}

.ctrl-input:focus {
  border-color: var(--primary) !important;
}

.ctrl-input-small {
  width: 65px !important;
  background: #ffffff !important;
  border: 1px solid var(--border) !important;
  border-radius: 6px !important;
  padding: 7px 8px !important;
  color: var(--text-main) !important;
  font-family: 'Inter', sans-serif !important;
  font-size: 12px !important;
  outline: none !important;
}

/* Dropdown overrides for light theme */
.ctrl-dropdown .Select-control,

```



© 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

```
.ctrl-dropdown .VirtualizedSelectOption {
  background: #ffffff !important;
  border: 1px solid var(--border) !important;
  color: var(--text-main) !important;
  border-radius: 6px !important;
  font-size: 12px !important;
}

.Select-menu-outer {
  background: var(--bg-card) !important;
  border: 1px solid var(--border) !important;
  z-index: 9999 !important;
}

.VirtualizedSelectOption,
.Select-option {
  background: var(--bg-card) !important;
  color: var(--text-main) !important;
  font-size: 12px !important;
}

.VirtualizedSelectOption.is-focused,
.Select-option.is-focused {
  background: rgba(2, 136, 209, 0.08) !important;
}

.Select-value-label {
  color: var(--text-main) !important;
  font-size: 12px !important;
}

.Select-placeholder {
  color: var(--text-hint) !important;
  font-size: 12px !important;
}

/* — Buttons (Kontrol) — */
.ctrl-btn {
  border: none;
  border-radius: 7px;
  padding: 8px 12px;
  font-family: 'Inter', sans-serif;
  font-size: 12px;
  font-weight: 600;
  cursor: pointer;
  transition: all 0.18s;
  background: rgba(255,255,255,0.08);
  color: var(--text-main);
  border: 1px solid var(--border);
  width: 100%;
  margin-bottom: 4px;
}

.ctrl-btn:hover {
  background: rgba(255,255,255,0.14);
  border-color: var(--primary-1);
}

.ctrl-btn-green {
```



© 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

```

background: rgba(38, 166, 154, 0.15) !important;
border-color: rgba(38, 166, 154, 0.35) !important;
color: #80cbc4 !important;
}

.ctrl-btn-green:hover {
  background: rgba(38, 166, 154, 0.25) !important;
  color: #b2dfdb !important;
}

.ctrl-btn-red {
  background: rgba(239, 83, 80, 0.1) !important;
  border-color: rgba(239, 83, 80, 0.3) !important;
  color: #ef9a9a !important;
}

.ctrl-btn-red:hover {
  background: rgba(239, 83, 80, 0.2) !important;
  color: #ff8a80 !important;
}

.ctrl-btn-purple {
  background: rgba(124, 77, 255, 0.12) !important;
  border-color: rgba(124, 77, 255, 0.3) !important;
  color: #b39ddb !important;
}

.ctrl-btn-purple:hover {
  background: rgba(124, 77, 255, 0.22) !important;
  color: #d1c4e9 !important;
}

.auto-ap-row {
  display: flex;
  align-items: center;
  gap: 8px;
}

.auto-ap-row .ctrl-btn {
  flex: 1;
  margin-bottom: 0;
}

.btn-row {
  display: flex;
  gap: 6px;
  margin-top: 4px;
}

.btn-row .ctrl-btn {
  flex: 1;
  margin-bottom: 0;
}

/* — Info Link & Popup — */
.info-link {
  cursor: pointer;
  color: #9c27b0;
  font-size: 11px;

```



© 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

```

user-select: none;
}

.info-link:hover {
  text-decoration: underline;
}

.info-popup {
  padding: 8px;
  border: 1px solid var(--border);
  border-radius: 7px;
  background: var(--bg-card);
  margin-top: 6px;
}

.info-table {
  border-collapse: collapse;
  font-size: 11px;
  width: 100%;
  color: var(--text-main);
}

.info-table th, .info-table td {
  padding: 4px 6px;
  border-bottom: 1px solid var(--border);
}

.info-table th {
  color: var(--primary-1);
  font-weight: 600;
}

/* — Notifikasi floating — */
.notif-float {
  position: fixed;
  top: calc(var(--navbar-h) + 12px);
  left: 50%;
  transform: translateX(-50%);
  z-index: 9999;
  padding: 10px 24px;
  background: rgba(1, 87, 155, 0.9);
  backdrop-filter: blur(8px);
  color: white !important;
  font-weight: 600;
  border-radius: 8px;
  border: 1px solid rgba(79, 195, 247, 0.3);
  box-shadow: 0 4px 20px rgba(0, 0, 0, 0.4);
  text-align: center;
  font-size: 13px;
  pointer-events: none;
  transition: opacity 0.3s;
}

/* — Wi-Fi Info Box — */
.wifi-info-box {
  padding: 10px;
  border: 1px solid var(--border);
  border-radius: 8px;
  background: rgba(255, 255, 255, 0.03);

```



© 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

```

font-size: 12px;
margin-top: 8px;
color: var(--text-main);
}

/* — Hint Text ————— */
.hint-text {
  font-size: 11px;
  color: var(--text-hint);
  margin-bottom: 8px;
}

/* — Measurement Form ————— */
.meas-form {
  display: flex;
  flex-direction: column;
  gap: 4px;
}

.meas-table {
  font-size: 11px;
  margin-top: 8px;
  color: var(--text-main);
}

.meas-table table {
  width: 100%;
  border-collapse: collapse;
}

.meas-table th, .meas-table td {
  padding: 4px 6px;
  border-bottom: 1px solid var(--border);
}

.meas-table th {
  color: var(--primary-1);
  font-weight: 600;
}

/* — Data Receiver Table ————— */
#table-data table {
  border: 1px solid var(--border) !important;
  border-collapse: collapse !important;
  width: 100% !important;
  font-size: 12px !important;
  color: var(--text-main) !important;
  background: var(--bg-panel) !important;
}

#table-data th, #table-data td {
  border: 1px solid var(--border) !important;
  padding: 6px 10px !important;
}

#table-data th {
  background: var(--bg-card) !important;
  color: var(--primary-1) !important;
  font-weight: 600 !important;

```



© 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

```

}

#table-data tr:hover {
  background: rgba(79, 195, 247, 0.05) !important;
}

/* =====
MODAL
===== */

.modal-overlay {
  position: fixed;
  inset: 0;
  background: rgba(0, 0, 0, 0.6);
  backdrop-filter: blur(4px);
  z-index: 5000;
  display: flex;
  align-items: center;
  justify-content: center;
}

.modal-box {
  background: var(--bg-card);
  border: 1px solid var(--border);
  border-radius: 16px;
  padding: 32px;
  max-width: 540px;
  width: 90%;
  position: relative;
  max-height: 85vh;
  overflow-y: auto;
  box-shadow: 0 24px 60px rgba(0,0,0,0.5);
  scrollbar-width: thin;
  scrollbar-color: var(--border) transparent;
}

.modal-close {
  position: absolute;
  top: 14px;
  right: 16px;
  background: transparent;
  border: none;
  color: var(--text-muted);
  font-size: 18px;
  cursor: pointer;
  transition: color 0.2s;
}

.modal-close:hover {
  color: var(--red);
}

.modal-title {
  font-size: 17px !important;
  font-weight: 700 !important;
  color: var(--text-main) !important;
  margin-bottom: 16px !important;
}

```



© 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

```
.modal-hint {
  font-size: 12px;
  color: var(--text-muted);
  margin-bottom: 12px;
}

/* — Save Summary — */
.save-summary {
  background: rgba(79, 195, 247, 0.06);
  border: 1px solid rgba(79, 195, 247, 0.15);
  border-radius: 8px;
  padding: 12px;
  margin-bottom: 16px;
  font-size: 12px;
  color: var(--text-muted);
}

.save-summary p { margin-bottom: 4px; }

/* — Stats Table — */
.stats-table {
  width: 100%;
  border-collapse: collapse;
  font-size: 12px;
  color: var(--text-main);
  margin-bottom: 12px;
}

.stats-table th, .stats-table td {
  padding: 8px 12px;
  border-bottom: 1px solid var(--border);
  text-align: left;
}

.stats-table th {
  background: rgba(255, 255, 255, 0.04);
  color: var(--primary-1);
  font-weight: 600;
}

/* — Quality Bars — */
.quality-bars {
  display: flex;
  flex-direction: column;
  gap: 6px;
  margin: 12px 0;
}

.quality-row {
  display: flex;
  justify-content: space-between;
  align-items: center;
  padding: 7px 14px;
  border-radius: 7px;
  font-size: 13px;
  font-weight: 500;
}
```



© 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

```
.quality-good { background: rgba(38, 166, 154, 0.15); color:
#80cbc4; border: 1px solid rgba(38, 166, 154, 0.25); }
.quality-mid { background: rgba(255, 167, 38, 0.12); color:
#ffcc80; border: 1px solid rgba(255, 167, 38, 0.25); }
.quality-bad { background: rgba(239, 83, 80, 0.12); color:
#ef9a9a; border: 1px solid rgba(239, 83, 80, 0.25); }
.quality-blind { background: rgba(120, 144, 156, 0.1); color:
#90a4ae; border: 1px solid rgba(120, 144, 156, 0.2); }

/* — Help Content — */
.help-content h4 {
  color: var(--primary-1);
  font-size: 13px;
  font-weight: 600;
  margin: 16px 0 8px;
}

.help-content p, .help-content li {
  font-size: 12px;
  color: var(--text-muted);
  margin-bottom: 4px;
  line-height: 1.6;
}

.help-content ol, .help-content ul {
  padding-left: 18px;
  margin-bottom: 8px;
}

/*
=====
DASH OVERRIDES
=====
*/

/* Loading spinner */
._dash-loading-callback {
  background: transparent !important;
}

/* Dash Tab Container */
.dash-tab {
  background: transparent !important;
  border: none !important;
  color: var(--text-muted) !important;
  font-family: 'Inter', sans-serif !important;
}

.tab--selected {
  color: var(--primary-1) !important;
  background: transparent !important;
  border-bottom: 2px solid var(--primary-1) !important;
}

/* Graph background */
.js-plotly-plot .plotly .bg {
  fill: var(--bg-dark) !important;
}

/* Scrollbars globally */
::-webkit-scrollbar {
```



© 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

```

width: 6px;
height: 6px;
}

::-webkit-scrollbar-thumb {
  background: var(--border);
  border-radius: 3px;
}

::-webkit-scrollbar-track {
  background: transparent;
}

/* =====
TAB UTAMA (Simulasi vs Hasil Pengujian Model)
===== */
.main-tabs {
  border-bottom: none !important;
}
.main-tab {
  font-weight: 600;
  padding: 10px 18px !important;
  color: var(--text-muted) !important;
  border: none !important;
  border-bottom: 3px solid transparent !important;
}
.main-tab-selected {
  color: var(--primary-1) !important;
  border-bottom: 3px solid var(--primary-1) !important;
}

/* =====
PANEL HASIL PENGUJIAN MODEL
===== */
.pengujian-panel {
  background: var(--bg-dark);
  padding: 24px 32px;
  max-width: 1200px;
  margin: 0 auto;
}
.pengujian-header {
  color: var(--text-main);
  margin-bottom: 4px;
}
.pengujian-subtext {
  font-size: 12px;
  color: var(--text-hint);
  margin-bottom: 14px;
}

.fitur-card-row {
  display: flex;
  gap: 14px;
  flex-wrap: wrap;
  margin-bottom: 8px;
}
.fitur-card {

```



© 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

```

flex: 1;
min-width: 220px;
background: var(--bg-card);
border: 1px solid var(--border);
border-radius: 8px;
padding: 14px 16px;
}
.fitur-card-title {
  font-weight: 700;
  color: var(--primary-d);
  font-size: 14px;
  margin-bottom: 2px;
}
.fitur-card-unit {
  font-size: 11px;
  color: var(--text-hint);
  margin-bottom: 6px;
}
.fitur-card-desc {
  font-size: 12px;
  color: var(--text-muted);
  margin: 0;
  line-height: 1.5;
}

.pengujian-table {
  width: 100%;
  border-collapse: collapse;
  margin-bottom: 14px;
  font-size: 13px;
}
.pengujian-table th {
  background: var(--bg-mid);
  color: var(--text-main);
  text-align: left;
  padding: 8px 10px;
  border: 1px solid var(--border);
}
.pengujian-table td {
  padding: 6px 10px;
  border: 1px solid var(--border);
  color: var(--text-main);
}
.pengujian-table-scroll {
  max-height: 420px;
  overflow-y: auto;
  border: 1px solid var(--border);
  border-radius: 6px;
}

.cm-row {
  display: flex;
  gap: 16px;
  flex-wrap: wrap;
}

/* Modal hasil perbandingan RF vs Teori (data test baru) */
#modal-box {
  transition: max-width 0.2s ease;

```

}



© 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

