© 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

EMBEDDED CONVOLUTIONAL NEURAL NETWORK (CNN) UNTUK KLASIFIKASI KONDISI CNC ROUTER BERDASARKAN KONSUMSI DAYA LISTRIK

Diajukan untuk memenuhi sebagian persyaratan memperoleh gelar Magister Terapan dalam Bidang Rekayasa Kontrol Industri

IRFAN MAHRUBI
2409511003

PROGRAM STUDI MAGISTER TERAPAN

TEKNIK ELEKTRO

PASCASARJANA POLITEKNIK NEGERI JAKARTA

DEPOK

2026

© Hak Cipta milik Politeknik Negeri Jakarta

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

HALAMAN PERNYATAAN BEBAS PLAGIARISME

Saya yang bertanda tangan di bawah ini menyatakan dengan sebenarnya bahwa tesis ini saya susun tanpa tindakan plagiarisme sesuai dengan peraturan yang berlaku di Politeknik Negeri Jakarta.

Jika di kemudian hari ternyata saya melakukan tindakan plagiarisme, saya akan bertanggung jawab sepenuhnya dan menerima sanksi yang diajukan oleh Politeknik Negeri Jakarta kepada saya.

Depok, 27 Juli 2026

Irfan Mahrubi

NIM. 2409511003

© 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

HALAMAN PERNYATAAN ORISINALITAS

Dengan ini saya menyatakan bahwa tesis yang saya susun ini adalah hasil karya sendiri, dan semua sumber baik yang saya kutip maupun saya rujuk telah saya nyatakan benar.

Nama : Irfan Mahrubi
NIM : 2409511003
Tanda Tangan :

Tanggal : 27 Juli 2026

© Hak Cipta milik Politeknik Negeri Jakarta

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

HALAMAN PENGESAHAN

Tesis ini yang telah diajukan oleh :

Nama : Irfan Mahrubi
NIM : 2409511003
Program Studi : Magister Terapan Teknik Elektro
Judul : *EMBEDDED CONVOLUTIONAL NEURAL NETWORK* (CNN) UNTUK KLASIFIKASI KONDISI CNC ROUTER BERDASARKAN KONSUMSI DAYA LISTRIK

Telah diuji oleh Tim Penguji dalam Sidang Tesis pada hari selasa 21 juli tahun 2026 dan dinyatakan LULUS untuk memperoleh derajat gelar Magister Terapan pada Program Studi Magister Terapan Teknik Elektro Politeknik Negeri Jakarta

Pembimbing I : Nana Sutarna, S.T., M.T., Ph.D. ( )

Pembimbing II : Dr. Prihatin Oktivasari, S.Si., M.Si. ( )

Penguji I : Dr. Isdawimah, S.T., M.T. ( )

Penguji II : Mera Kartika Delimayanti, S.Si., M.T., Ph.D. ( )

Penguji III : Dr. Murie Dwiyaniti, S.T., M.T. ( )

Depok, 27 Juli 2026
Diketahui oleh:
Ketua Pascasarjana Politeknik Negeri Jakarta

Prof. Dr. Tatun Hayatun Nufus, M.Si.
NIP. 196604161995122001

© 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

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI
TESIS UNTUK KEPENTINGAN AKADEMIK

Sebagai sivitas akademik Politeknik Negeri Jakarta, saya yang bertanda tangan dibawah ini

Nama : Irfan Mahrubi
NIM : 2409511003
Program Studi : Magister Terapan Teknik Elektro
Jenis Karya : Tesis

Demi pengembangan ilmu pengetahuan, saya menyetujui untuk memberikan kepada Politeknik Negeri Jakarta Hak Bebas Royalti Nonekslusif (*Non-exclusive Royalti – Free Right*) atas karya ilmiah saya berjudul : EMBEDDED CONVOLUTIONAL NEURAL NETWORK (CNN) UNTUK KLASIFIKASI KONDISI CNC ROUTER BERDASARKAN KONSUMSI DAYA LISTRIK
Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Nonekslusif ini Politeknik Negeri Jakarta berhak menyimpan, mengalihmediakan/mengalihinformasikan, mengelola dalam bentuk pangkalan data (*data base*), merawat dan mempublikasikan tesis saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.
Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok
Pada tanggal : 27 Juli 2026
Yang menyatakan

Irfan Mahrubi
2409511003

© 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

ABSTRAK

Nama : Irfan Mahrubi
Program Studi : Magister Terapan Teknik Elektro
Judul Tesis : EMBEDDED CONVOLUTIONAL NEURAL NETWORK (CNN) UNTUK KLASIFIKASI KONDISI CNC ROUTER BERDASARKAN KONSUMSI DAYA LISTRIK

Penggunaan mesin CNC Router secara intensif dapat menimbulkan perubahan pola konsumsi daya listrik yang mengindikasikan kondisi operasi tidak normal, namun klasifikasi kondisi umumnya masih konvensional atau bergantung komputasi terpusat yang kurang sesuai untuk perangkat sumber daya terbatas. Penelitian ini merancang dan merealisasikan sistem klasifikasi kondisi berbasis *Embedded* CNN dengan ESP32, menggunakan tujuh parameter kelistrikan dari sensor PZEM-004T sebagai masukan model. Model 1D-CNN dengan 14.946 parameter dilatih menggunakan skema klasifikasi biner berdasarkan lima kondisi mesin (normal, tumpul, *overwork*, stuck, patah), lalu dikuantisasi menjadi INT8 berukuran 25,39 KB (turun 61% dari 65,4 KB). Model yang berjalan langsung pada *firmware* ESP32 mencapai akurasi 90,60% pada data uji. Eksperimen pembanding menunjukkan skema biner secara empiris mengungguli skema *multiclass* lima kelas (90,68% berbanding 88,03% akurasi), memperkuat justifikasi pemilihan skema biner. Sistem juga terintegrasi dengan autentikasi RFID, LCD, RTC, relay, dan MQTT ke *dashboard*. Hasil ini menunjukkan sistem klasifikasi kondisi berbasis *Embedded* CNN dapat bekerja secara *real-time*, efisien, dan sesuai untuk perangkat dengan keterbatasan memori dan daya komputasi.

Kata kunci: *CNC Router*, Konsumsi Daya Listrik, Klasifikasi Kondisi Mesin, Embedded Convolutional Neural Network (CNN), ESP32, TensorFlow Lite for Microcontrollers, *TinyML*.

© 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

KATA PENGANTAR

Puji syukur kehadirat Allah ﷻ yang telah melimpahkan rahmat, serta karunia-Nya yang begitu besar sehingga penulis dapat menyelesaikan tesis ini yang berjudul "*EMBEDDED CONVOLUTIONAL NEURAL NETWORK* (CNN) UNTUK KLASIFIKASI KONDISI CNC *ROUTER* BERDASARKAN KONSUMSI DAYA LISTRIK" tepat waktu. Sholawat dan salam semoga selalu tercurah limpahkan kepada baginda nabi Muhammad ﷺ kepada keluarganya, kepada para sahabatnya, serta kita selaku umatnya.

Tujuan dalam penyusunan tesis ini adalah untuk memenuhi kebutuhan syarat kelulusan Program Studi Magister Terapan Teknik Elektro Politeknik Negeri Jakarta. Dalam penyusunan tesis ini, penulis memperoleh banyak bantuan, dukungan, bimbingan, saran, motivasi, serta doa dari berbagai pihak. Oleh karena itu, penulis menyampaikan ucapan terima kasih kepada :

1. Kapada istri saya tercinta, Nopa Sari yang selalu mengingatkan untuk melanjutkan S2 dan selalu mendukung selama kuliah, serta anak saya Hanin Dhiya Mahrubi yang selalu memberikan kebahagiaan dan warna disetiap harinya.
2. Ibu Prof. Dr. Tatun Hayatun Nufus, M.Si. selaku Ketua Pascasarjana Politeknik Negeri Jakarta.
3. Bapak Nana Sutarna, S.T., M.T., Ph.D. selaku Ketua Program Studi Magister Terapan Teknik Elektro Politeknik Negeri Jakarta yang sekaligus pembimbing pertama yang senantiasa membimbing dan membantu serta memudahkan pelaksanaan proses penyusunan tesis ini. Semoga Allah senantiasa memberikan kelimpahan riski dan kesehatan kepada bapak.
4. Ibu Dr. Prihatin Oktivasari, S.Si., M.Si. selaku pembimbing kedua yang senantiasa membimbing dan membantu serta memudahkan pelaksanaan proses penyusunan tesis ini. Semoga Allah senantiasa memberikan kelimpahan riski dan kesehatan kepada ibu.
5. Ibu Ani Kristiani, M.Pd. selaku kepala SMKN 39 Jakarta, Bapak Ibu Wakil Kepala Sekolah , serta rekan-rekan guru di Jurusan Teknik Elektronika baik di

konsentrasi Audio Video, Elektronika Industri, dan Elektronika Komunikasi SMKN 39 Jakarta yang selalu memotivasi dan mendukung selama kuliah.

6. Kepada alumni Teknik Audio-Video SMKN 39 Jakarta, Andhika Darmawan, Muhammad Syafiq Aziz, dan Fajar Sidik Febriansyah yang selalu membantu dan mendukung.
7. Seluruh keluarga angkatan MTTE 2024 khususnya Mahmudin selaku ketua angkatan yang selalu membantu dan mengingatkan dalam perkuliahan. Panjiana, dan Inzie Laziera yang selalu ada dan membantu penulis serta seluruh teman. Semoga Allah senantiasa memberikan kita semua kesuksesan dan kemudahan.
8. Kepada seluruh pihak pihak yang telah membantu atas terselesaikannya laporan tesis ini yang tidak bisa penulis sebutkan satu persatu.

Penulis menyadari bahwa laporan tesis ini masih memiliki kekurangan. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan untuk perbaikan di masa mendatang. Penulis memohon maaf atas segala kekurangan dalam penyusunan laporan ini dan berharap karya ini dapat memberikan manfaat bagi penulis maupun pembaca.

Depok, 27 Juli 2026

Penulis

© 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

© 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

DAFTAR ISI

HALAMAN JUDUL i
HALAMAN PERNYATAAN BEBAS PLAGIARISME ii
HALAMAN PERNYATAAN ORISINALITAS iii
HALAMAN PENGESAHAN iv
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TESIS UNTUK KEPENTINGAN AKADEMIK v
ABSTRAK vi
KATA PENGANTAR vii
DAFTAR ISI ix
DAFTAR GAMBAR xi
DAFTAR TABEL xiii
BAB I PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Rumusan Masalah 3
1.3 Tujuan Penelitian 3
1.4 Manfaat Penelitian 4
1.5 Batasan Penelitian 4
1.6 State of the Art 5
BAB II TINJAUAN PUSTAKA 7
2.1 CNC *Router* 3018 7
2.2 *Teaching Factory* (TeFa) 10
2.3 *Monitoring* Energi Listrik Berbasis IoT 11
2.4 Mikrokontroler ESP32 11
2.5 Autentikasi dan Kontrol Akses Berbasis RFID 12
2.6 Convolutional Neural Network (CNN) 12
2.7 Parameter Listrik sebagai Indikator Kondisi CNC Router 14
2.8 Implementasi TinyML 14
2.9 Validasi Model CNN 15
2.10 Hyperparameter Model Deep Learning 15

© 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

BAB III METODOLOGI PENELITIAN 16
3.1 Kerangka Konseptual 16
3.2 Desain Penelitian 18
3.2.1 Tahapan Penelitian 19
3.3 Perancangan Sistem Akuisisi Data 21
3.3.1 Perancangan Hardware 21
3.3.2 Perancangan Software 22
3.3.3 Integrasi Sistem 22
3.4 Metode Pengambilan Data 23
3.4.1 Akuisisi dan Penyimpanan Data 24
3.4.2 Parameter yang Diukur 24
3.4.3 Rumus Perhitungan Parameter Kelistrikan 25
3.4.4 Proses Akuisisi Data 27
3.4.5 Validasi Sensor PZEM-004T 27
3.4.6 Penyimpanan Data 28
3.5 Pengolahan Data dan Penyusunan Dataset 28
3.5.1 Preprocessing Data 28
3.6 Perancangan Model CNN Berbasis TinyML 30
3.6.1 Arsitektur Model CNN 30
3.6.2 Proses Pelatihan Model 31
3.6.3 Evaluasi Model 31
3.6.4 Hyperparameter 33
3.6.5 Implementasi TinyML pada ESP32 33
3.7 Metodologi Eksperimen Rasio Pembagian Dataset 34
BAB IV HASIL DAN PEMBAHASAN 35
4.1 Validasi Sensor PZEM-004T terhadap Multimeter 35
4.2 Verifikasi Perhitungan Parameter Kelistrikan 37
4.3 Hasil Pengujian RFID 38
4.4 Hasil Pengujian SSR 39
4.5 Hasil Pengujian Akuisisi dan Integrasi *Real-Time* 40
4.6 Hasil Akuisisi dan Karakteristik Data 42
4.7 Hasil Analisis Statistik Pemilihan Fitur 46

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

4.8 Hasil Perancangan Dataset dan Pembagian Data 48
4.9 Hasil Pelatihan Model CNN 49
4.10 Analisis Overfitting dan Underfitting pada Model Final 49
4.11 Hasil Pengujian Model pada Data Uji 50
4.12 Verifikasi Perhitungan Metrik Evaluasi 52
4.13 Perbandingan Kinerja Model Klasifikasi Biner dan Multiclass 52
4.14 Hasil Eksperimen Rasio Pembagian Dataset 54
4.15 Hasil Kuantisasi Model untuk Implementasi Embedded 56
4.16 Realisasi Implementasi pada ESP32 57
4.17 Pembahasan 58
BAB V KESIMPULAN DAN SARAN 61
5.1 Kesimpulan 61
5.2 Saran 62
DAFTAR PUSTAKA 64
LAMPIRAN 69

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

DAFTAR GAMBAR

Gambar 2.1. Skema penampang motor stepper dua fasa pada sumbu X/Y/Z CNC Router 3018 8
Gambar 2.2. Rangkaian ekivalen motor DC (spindle) CNC Router 3018 9
Gambar 2.3. Ilustrasi operasi konvolusi 1D pada deret waktu parameter kelistrikan 14
Gambar 3.1. Kerangka Konseptual Penelitian 17
Gambar 3.2. Posisi Random Forest dan CNN dalam kerangka konseptual penelitian 18
Gambar 3.3. Diagram Blok Tahapan Penelitian 20
Gambar 3.4. Diagram Blok Sistem Akuisisi 21
Gambar 3.5. Segitiga daya listrik (hubungan P, Q, dan S) 26
Gambar 3.6. Arsitektur 1D-CNN untuk klasifikasi kondisi normal/tidak normal pada CNC Router 31
Gambar 3.7. Diagram Blok Sistem saat Inferensi *Real-Time* 34
Gambar 4.1. Pengukuran Tegangan menggunakan *Multimeter* 36
Gambar 4.2. Pengukuran Arus menggunakan Multimeter 37
Gambar 4.3. Ujicoba RFID 39
Gambar 4.4. SSR dalam Kondisi ON 40
Gambar 4.5. SSR dalam kondisi Off 40
Gambar 4.6. Distribusi jumlah sampel data bersih per kondisi mesin 43
Gambar 4.7. Pola sinyal daya aktif dan arus pada 60 sampel pertama (~1 menit) dari satu sesi representatif tiap kondisi 43
Gambar 4.8. Pola sinyal arus dan daya aktif per kondisi mesin (Normal, Tumpul, *Overwork*, *Stuck*, Patah) 44
Gambar 4.9. Arus puncak dan daya aktif (puncak vs rata-rata) per kondisi mesin 45
Gambar 4.10. Kontribusi (*feature importance*) tiap parameter kelistrikan 46
Gambar 4.11. Kurva loss dan akurasi training vs validasi pada model final 49
Gambar 4.12. *Confusion matrix* model final (float32) pada data uji 51

Gambar 4.13. Perbandingan confusion matrix model biner dan model multiclass pada data uji yang sama 53
Gambar 4.14. Perbandingan metrik evaluasi (akurasi, *precision*, *recall*, F1-*score*) model biner dan multiclass.....53
Gambar 4.15. Perbandingan ukuran model sebelum (float32) dan sesudah kuantisasi (INT8) 56
Gambar 4.16. Realisasi rangkaian sistem embedded CNN terintegrasi pada CNC Router.....58

© 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

© 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

DAFTAR TABEL

Tabel 3.1 Ringkasan Hyperparameter Model CNN 33
Tabel 4.1 Hasil validasi sensor PZEM-004T terhadap *multimeter* 35
Tabel 4.2 Hasil pengujian sistem autentikasi RFID 38
Tabel 4.3 Hasil pengujian akuisisi dan integrasi data *real-time* 41
Tabel 4.4 Statistik latensi round-trip hasil analisis *Wireshark* 42
Tabel 4.5 Statistik arus dan daya per kondisi mesin 45
Tabel 4.6 Kontribusi total tiap parameter kelistrikan terhadap klasifikasi normal/tidak normal 46
Tabel 4.7 Pembagian dataset (70% latih / 20% validasi / 10% uji) 49
Tabel 4.8 Ringkasan hasil pelatihan model (skema 70/20/10) 50
Tabel 4.9 Laporan klasifikasi model CNN (float32) pada data uji 50
Tabel 4.10 Perbandingan kinerja model CNN klasifikasi biner dan *multiclass* .52
Tabel 4.11 Hasil eksperimen rasio pembagian dataset 55

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

BAB I
PENDAHULUAN

1.1. Latar Belakang

Teaching Factory (TeFa) di SMK bidang elektronika merupakan lingkungan pembelajaran berbasis industri yang melibatkan penggunaan mesin-mesin produksi seperti CNC *Router*, *Reflow Oven* PCB, *Laser Cutting*, serta mesin SMT *Pick and Place*. Mesin seperti CNC *Router* sering digunakan untuk proses pembuatan jalur PCB dan dalam praktiknya mesin tersebut digunakan secara bergantian oleh banyak pengguna dengan tingkat kompetensi yang beragam. Kondisi ini menimbulkan permasalahan berupa tidak adanya pemantauan kondisi mesin secara berkelanjutan, keterbatasan data historis penggunaan mesin, serta belum tersedianya mekanisme identifikasi dini terhadap anomali operasi mesin yang berpotensi menyebabkan downtime dan kerusakan peralatan. Pada sisi operasional, konsumsi energi listrik mesin juga belum tercatat secara rinci per mesin sehingga menyulitkan analisis efisiensi dan biaya produksi.

Beberapa penelitian menunjukkan bahwa sistem *monitoring* energi berbasis *Internet of Things* (IoT) mampu menyediakan data konsumsi energi listrik secara *real-time* dan mendukung pengelolaan energi yang lebih efisien [1],[2]. Pada sisi perangkat, kombinasi sensor kelistrikan dengan mikrokontroler seperti ESP32 terbukti efektif untuk memonitor parameter listrik dengan akurasi memadai dan biaya implementasi yang rendah [3],[4]. Selain itu, teknologi *Radio Frequency Identification* (RFID) banyak digunakan sebagai sistem kontrol akses dan pencatatan penggunaan mesin di lingkungan industri [5].

Dalam konteks klasifikasi kondisi mesin CNC *Router*, pendekatan berbasis *machine learning* dan *deep learning* telah banyak dikembangkan menggunakan parameter getaran, emisi akustik, maupun sinyal kelistrikan. Namun, sebagian besar penelitian tersebut masih mengandalkan komputasi terpusat atau *cloud*, serta model dengan kompleksitas tinggi yang kurang sesuai untuk diimplementasikan pada perangkat dengan sumber daya terbatas [6]. Perkembangan komputasi *edge* mendorong munculnya pendekatan *Tiny Machine Learning* (TinyML), yaitu

© 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

pendekatan implementasi model *machine learning* ringan yang ditanam pada perangkat mikrokontroler, sehingga proses inferensi dapat dilakukan secara *real-time* dengan latensi rendah dan konsumsi daya rendah [7].

Pemilihan parameter listrik sebagai dasar klasifikasi kondisi pada penelitian ini didasarkan pada kesesuaiannya dengan pendekatan TinyML dengan model *Convolutional Neural Network (*CNN). Sensor getaran dan akustik umumnya memerlukan frekuensi sampling tinggi, proses ekstraksi fitur kompleks, dan model *deep learning* yang relatif berat. Selain itu, sensor tersebut sensitif terhadap gangguan lingkungan dan memerlukan instalasi mekanis yang presisi. Sebaliknya, parameter listrik seperti arus, tegangan, daya, dan energi dapat diperoleh langsung dari sistem kelistrikan mesin tanpa modifikasi mekanis. Pola konsumsi daya dan arus telah terbukti mencerminkan kondisi operasi dan beban mekanik mesin, sehingga relevan digunakan untuk klasifikasi kondisi mesin. Dari sisi komputasi, data parameter kelistrikan bersifat lebih sederhana dan stabil, sehingga lebih sesuai untuk diproses menggunakan model *machine learning* ringan yang dijalankan secara *embedded* pada mikrokontroler seperti ESP32 [7],[8].

Pemrosesan data secara terpusat memerlukan koneksi jaringan yang stabil, menimbulkan latensi pengiriman data, serta membutuhkan biaya infrastruktur *server* yang tidak selalu tersedia pada lingkungan *Teaching Factory* di SMK. Pendekatan *embedded* memungkinkan proses inferensi dilakukan langsung pada perangkat ESP32 tanpa bergantung pada konektivitas internet, sehingga klasifikasi kondisi dapat berlangsung secara *real-time* dengan latensi rendah, konsumsi daya yang lebih hemat, serta menjaga privasi data operasional pengunaan CNC karena data tidak perlu dikirim ke pihak ketiga [9],[10].

Meskipun model CNN dan pendekatan TinyML telah banyak diterapkan pada berbagai aplikasi, namun evaluasi pada mesin CNC *Router* berbasis parameter listrik di SMK masih terbatas. Selain itu, belum terdapat penelitian yang secara spesifik mengkaji bagaimana TinyML dapat diintegrasikan dengan sistem akuisisi daya listrik dan klasifikasi kondisi mesin berdasarkan parameter kelistrikan.

Berdasarkan gap tersebut, penelitian ini mengusulkan *Embedded* CNN untuk klasifikasi kondisi berdasarkan konsumsi daya listrik pada CNC *Router*. Fokus utama penelitian ini adalah pada perancangan, implementasi, dan evaluasi model

© 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

CNN dengan pendekatan TinyML yang dijalankan secara *embedded* pada ESP32 untuk klasifikasi kondisi berdasarkan konsumsi daya listrik CNC *Router* secara *real-time*. Sistem akuisisi berbasis sensor PZEM-004T dan autentikasi pengguna berbasis RFID berperan sebagai komponen pendukung penyedia data dan pengelolaan operasional. Dengan *embedded* CNN menggunakan pendekatan TinyML, sistem diharapkan mampu melakukan klasifikasi kondisi yang responsif, hemat sumber daya, dan sesuai dengan keterbatasan infrastruktur *Teaching Factory*, sekaligus meningkatkan keandalan mesin dan kualitas pembelajaran berbasis industri di SMK.

1.2. Rumusan Masalah

Masalah utama yang akan diselesaikan dalam penelitian ini adalah :

1. Bagaimana merancang sistem akuisisi data konsumsi daya listrik pada CNC *Router*?
2. Bagaimana merancang dan mengimplementasikan model *Embedded* CNN pada ESP32 untuk melakukan klasifikasi kondisi normal dan anomali konsumsi daya listrik CNC *Router* secara *real-time*?
3. Bagaimana kinerja model *Embedded* CNN dalam melakukan klasifikasi kondisi normal dan anomali konsumsi daya listrik CNC *Router*?

1.3. Tujuan Penelitian

Tujuan umum penelitian ini adalah mengembangkan sistem klasifikasi kondisi berdasarkan konsumsi daya listrik pada mesin CNC *Router* berbasis *Embedded* CNN yang diimplementasikan pada ESP32 dengan memanfaatkan parameter konsumsi daya listrik untuk mengklasifikasi kondisi operasi mesin.

Adapun tujuan khusus yang ingin dicapai dalam penelitian ini adalah :

1. Merancang dan membangun sistem akuisisi data konsumsi daya listrik pada CNC *Router* sebagai sumber data masukan yang representatif bagi model CNN.
2. Merancang dan mengimplementasikan model *Embedded* CNN pada ESP32 untuk melakukan klasifikasi kondisi berdasarkan konsumsi daya listrik CNC *Router* secara *real-time*.
3. Mengevaluasi kinerja model *Embedded* CNN dalam mengklasifikasi kondisi normal dan anomali konsumsi daya listrik mesin CNC *Router* berdasarkan hasil

© 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

pengujian sistem.

1.4. Manfaat Penelitian

Manfaat penelitian ini dapat ditinjau dari dua aspek, yaitu manfaat teoritis dan manfaat praktis.

Manfaat Teoritis

1. Memberikan kontribusi terhadap pengembangan ilmu pengetahuan di bidang sistem *embedded*, *monitoring* energi listrik, dan penerapan CNN untuk klasifikasi kondisi pada mesin industri.
2. Menambah kajian ilmiah mengenai pemanfaatan parameter konsumsi daya listrik sebagai indikator kondisi operasi CNC *Router* melalui pendekatan *deep learning* berbasis CNN pada perangkat dengan keterbatasan sumber daya komputasi.
3. Menjadi referensi bagi penelitian selanjutnya yang berkaitan dengan klasifikasi kondisi mesin berbasis data energi listrik serta implementasi model *deep learning* pada perangkat *embedded* seperti ESP32.

Manfaat Praktis

1. Menghasilkan sistem akuisisi data konsumsi daya listrik pada CNC *Router* yang dapat digunakan sebagai dasar analisis kondisi operasi mesin.
2. Menyediakan sistem klasifikasi kondisi berdasarkan konsumsi daya listrik berbasis *Embedded* CNN yang dapat diimplementasikan secara *real-time* pada perangkat ESP32.
3. Mendukung peningkatan keandalan operasional CNC *Router* melalui pemanfaatan hasil klasifikasi kondisi sebagai dasar pengambilan keputusan dalam pemeliharaan dan pengawasan kondisi mesin.

1.5. Batasan Penelitian

Agar penelitian terarah dan sesuai dengan ruang lingkup yang telah ditetapkan, maka batasan penelitian ini adalah sebagai berikut:

1. Penelitian difokuskan pada mesin CNC Router sebagai objek studi, dan tidak mencakup mesin produksi lainnya.
2. Parameter yang digunakan sebagai dasar klasifikasi kondisi terbatas pada

© 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

parameter konsumsi daya listrik yang diperoleh melalui sensor PZEM-004T, meliputi tegangan, arus, daya aktif, daya reaktif, daya semu, energi, dan faktor daya.

3. Klasifikasi kondisi mesin pada penelitian ini dibatasi hanya pada dua kelas keluaran, yaitu kondisi normal dan kondisi tidak normal (anomali). Kelima skenario pengambilan data (normal, mata bor tumpul, overwork, *stuck*, dan patah) tetap direkam secara terpisah untuk keperluan analisis pola, namun untuk kebutuhan pelatihan dan pengujian model CNN, seluruh kondisi selain normal dikelompokkan ke dalam satu kelas anomali. Pembatasan ini dilakukan agar model lebih fokus pada tugas deteksi biner (normal/tidak normal) yang relevan untuk kebutuhan pemantauan dini di lapangan [11].
4. Data yang digunakan merupakan data hasil akuisisi langsung dari mesin CNC Router selama kondisi operasi normal dan kondisi anomali yang diamati dalam periode penelitian dengan pengkondisian tertentu.
5. Model yang digunakan untuk klasifikasi kondisi adalah *Embedded* CNN dengan arsitektur ringan (*lightweight*), yang dirancang agar dapat dijalankan pada perangkat ESP32 dengan keterbatasan memori dan daya komputasi.
6. Implementasi sistem dibatasi pada proses inferensi model CNN secara *embedded* pada ESP32, sedangkan proses pelatihan (*training*) model dilakukan pada perangkat komputasi terpisah.
7. Evaluasi kinerja model difokuskan pada kemampuan sistem dalam mengklasifikasi kondisi normal dan anomali berdasarkan hasil pengujian model, tanpa membahas aspek optimasi lanjutan seperti kompresi model tingkat lanjut atau perbandingan dengan algoritma lain.

1.6. State of the Art

Penelitian deteksi anomali di lingkungan industri terus berkembang seiring integrasi IoT dan *machine learning* pada sistem manufaktur cerdas. Studi oleh Villegas-Ch dkk. menunjukkan bahwa penerapan *machine learning* pada data sensor mampu meningkatkan akurasi deteksi anomali dan menurunkan tingkat kesalahan dibandingkan metode konvensional [31]. Meski demikian, pendekatan yang digunakan masih bertumpu pada komputasi terpusat berbasis *server*, sehingga

© 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

ketergantungan terhadap jaringan dan infrastruktur tetap tinggi.

Perkembangan *edge computing* kemudian mendorong lahirnya pendekatan TinyML yang memungkinkan proses inferensi dilakukan langsung pada perangkat dengan sumber daya terbatas. Penelitian Antonini dkk. membuktikan bahwa sistem deteksi anomali berbasis TinyML dapat dijalankan pada ESP32 dengan kebutuhan memori relatif kecil dan waktu respons yang cepat [32]. Namun, penelitian tersebut berfokus pada parameter getaran dan menggunakan pendekatan unsupervised learning, sehingga belum mengkaji konsumsi daya listrik sebagai indikator utama kondisi mesin.

Di sisi lain, Mao dkk. dalam penelitiannya menunjukkan bahwa model *deep learning* berbasis rekonstruksi deret waktu mampu menghasilkan performa deteksi yang sangat tinggi pada data konsumsi energi [33]. Walaupun demikian, implementasinya masih dilakukan pada sistem komputasi berkapasitas besar. Berdasarkan celah tersebut, penelitian ini mengusulkan pengembangan *Embedded* CNN berbasis TinyML untuk klasifikasi kondisi berdasarkan konsumsi daya listrik mesin CNC Router pada ESP32, dengan integrasi sensor PZEM-004T dan model ringan berbasis *TensorFlow Lite for Microcontrollers* agar proses inferensi dapat dilakukan secara lokal, *real-time*, dan sesuai dengan keterbatasan perangkat *embedded*.

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

BAB V
KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan hasil perancangan, implementasi, pengujian, dan pembahasan yang telah dilakukan, dapat disimpulkan hal-hal berikut yang menjawab rumusan masalah penelitian ini diantaranya :

1. Sistem akuisisi data konsumsi daya listrik CNC *Router* berbasis ESP32 dan sensor PZEM-004T berhasil dirancang dan digunakan untuk mengumpulkan data tujuh parameter kelistrikan yaitu tegangan, arus, daya aktif, daya reaktif, daya semu, faktor daya, dan frekuensi secara kontinu yang menghasilkan *dataset* bersih sebanyak 39.997 baris data dari 104 sesi rekaman meliputi lima kondisi operasi normal, tumpul, *overwork*, *stuck*, dan patah yang terbukti representatif sebagai sumber data pelatihan model klasifikasi kondisi CNC *Router*.
2. Model *Embedded* 1D-CNN berhasil dirancang, dilatih, dikuantisasi, dan diimplementasikan menggunakan pendekatan TinyML pada ESP32. Model dengan 14.946 parameter berhasil dikuantisasi menjadi INT8 berukuran 25,39 KB yang mana jauh lebih kecil dibandingkan versi float32 dengan ukuran 65,4 KB, pengurangan ukuran sekitar 61% ini jauh di bawah kapasitas memori ESP32, mengonfirmasi kelayakan model untuk dijalankan pada perangkat dengan keterbatasan memori dan daya komputasi. Model final ini telah diintegrasikan ke dalam *firmware* ESP32 yang lengkap, mencakup autentikasi RFID, tampilan LCD, pencatatan waktu RTC, kontrol SSR, dan publikasi data MQTT.
3. Model Embedded CNN mengklasifikasi kondisi normal dan tidak normal dengan akurasi 90,60% pada 1.245 window data uji independen, dievaluasi langsung dari model final (model_data.h) pada firmware ESP32, menggunakan skema pembagian data 70% latih / 20% validasi / 10% uji yang terpilih sebagai skema terbaik dari empat skenario yang diuji (Tabel 4.11). Kinerja kelas tidak normal (precision 0,878, recall 0,825) dan kelas normal (precision 0,919, recall 0,945) jauh lebih seimbang dibanding model awal, berkat perbaikan volume 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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

dan koreksi metodologi pembagian dataset dari skema kronologis menjadi acak berstrata. Selisih akurasi latih-validasi sebesar 5,34 poin persen menunjukkan overfitting ringan yang telah dibatasi oleh mekanisme early stopping dan restorasi bobot terbaik.

5.2. Saran

Berdasarkan keterbatasan yang telah diidentifikasi, beberapa saran diajukan untuk pengembangan dan penelitian lanjutan sebagai berikut :

1. Memperluas cakupan validasi sensor PZEM-004T dengan menambah jumlah titik pengukuran (idealnya minimal sepuluh titik) dan memperpanjang durasi pengujian akuisisi dan integrasi *real-time* hingga minimal tiga puluh menit pengoperasian berkelanjutan untuk memperoleh hasil validasi yang lebih representatif dan meyakinkan.
2. Pengukuran latensi inferensi, konsumsi memori (SRAM/flash), dan konsumsi daya secara langsung pada perangkat keras ESP32 perlu dilakukan untuk memvalidasi kelayakan implementasi secara empiris, mengingat penelitian ini baru mengukur latensi pada komputer pengembangan (*host* CPU) yang tidak merepresentasikan kondisi sesungguhnya pada mikrokontroler.
3. Penambahan volume data telah dilakukan pada penelitian ini (dari 34 menjadi 104 sesi) dan terbukti meningkatkan kinerja model secara signifikan. Untuk penelitian lanjutan, disarankan merancang akuisisi data secara terkontrol dengan mengacak urutan perekaman kondisi (bukan mengikuti siklus tetap) dan mencatat kondisi jala-jala listrik (tegangan referensi) secara terpisah, untuk mengeliminasi risiko variabel perancu (confounding) seperti yang teridentifikasi pada analisis feature importance tegangan dalam penelitian ini. Variasi kondisi lingkungan (suhu, kelembapan) yang lebih beragam serta perekaman pada rentang waktu yang lebih panjang juga akan meningkatkan generalisasi model lebih lanjut.
4. Validasi menggunakan kondisi kegagalan mesin yang terjadi secara alami (bukan dikondisikan secara sengaja) perlu dilakukan untuk menguji kemampuan model dalam mengenali pola degradasi mesin yang sesungguhnya terjadi di lapangan dalam jangka waktu yang lebih panjang.
5. Penelitian lanjutan dapat mengeksplorasi teknik penanganan ketidakseimbangan data yang lebih lanjut (misalnya data augmentation atau teknik oversampling

© 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

pada level *window*), maupun arsitektur model alternatif yang lebih ringan, untuk meningkatkan kinerja identifikasi khususnya pada kelas tidak normal tanpa menambah kompleksitas komputasi secara signifikan.

6. Uji coba implementasi pada jenis mesin CNC lain atau mesin produksi lain di lingkungan Teaching Factory dapat dilakukan untuk menguji ketertransferan (transferability) pendekatan yang diusulkan pada penelitian ini di luar konteks CNC Router 3018.

© 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

DAFTAR PUSTAKA

[1] H. O. Garcés, J. Godoy, G. Riffo, N. F. Sepúlveda, E. Espinosa, and M. A. Ahmed, "Development of an IoT-Enabled Smart Electricity Meter for Real-Time Energy Monitoring and Efficiency," Electron., vol. 14, no. 6, pp. 1–36, 2025, doi: 10.3390/electronics14061173.

[2] A. A. Mirani, A. Awasthi, N. O'Mahony, and J. Walsh, "Industrial IoT-Based Energy Monitoring System: Using Data Processing at Edge," Internet of Things, vol. 5, no. 4, pp. 608–633, 2024, doi: 10.3390/iot5040027.

[3] J. Varela-Aldás, S. Silva, and G. Palacios-Navarro, "IoT-Based Alternating Current Electrical Parameters Monitoring System," Energies, vol. 15, no. 18, 2022, doi: 10.3390/en15186637.

[4] D. Hercog, T. Lerher, M. Truntič, and O. Težak, "Design and Implementation of ESP32- Based IoT Devices," Sensors, vol. 23, no. 15, 2023, doi: 10.3390/s23156739.

[5] Y. Duroc, "From Identification to Sensing: RFID Is One of the Key Technologies in the IoT Field," Sensors, vol. 22, no. 19, 2022, doi: 10.3390/s22197523.

[6] R. Çekik, "Deep Learning for Anomaly Detection in CNC Machine Vibration Data : A RoughLSTM-Based Approach," pp. 1–29, 2025.

[7] A. V. Pălăcean, D. C. Trancă, R. V. Rughiniș, and D. Rosner, "Smart Internet of Things Power Meter for Industrial and Domestic Applications," Appl. Sci., vol. 14, no. 17, 2024, doi: 10.3390/app14177621.

[8] P. Nunes, E. Rocha, and J. P. Santos, "Using Intelligent Edge Devices for Predictive Maintenance on Injection Molds †," Appl. Sci., vol. 13, no. 12, 2023, doi: 10.3390/app13127131.

[9] S. Heydari dan Q. H. Mahmoud, "Tiny Machine Learning and On-Device Inference: A Survey of Applications, Challenges, and Future Directions," Sensors, vol. 25, no. 10, 3191, 2025.

[10] V. Selvaraj dan S. Min, "Real-time fault identification system for a retrofitted ultra-precision CNC machine from equipment's power consumption data: a case study of an implementation," Int. J. Precis. Eng. Manuf.-Green Technol., vol. 10, pp. 925-

© 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

941, 2023, doi: 10.1007/s40684-022-00497-x.

[11] A. Athar, M. A. I. Mozumder, S. Ali, dan H.-C. Kim, "Deep learning-based anomaly detection using one-dimensional convolutional neural networks (1D CNN) in machine centers (MCT) and computer numerical control (CNC) machines," PeerJ Comput. Sci., vol. 10, e2389, 2024, doi: 10.7717/peerj-cs.2389.

[12] D. S. Rifai, A. Ardiansyah, A. I. B. Yahya, dan R. Fitriyani, “The Implementation of Teaching Factory in Vocational Education: A Systematic Review,” Varidika: Jurnal Ilmu Pendidikan, vol. 38, no. 1, 2026.

[13] A. D. Widiatna, P. P. Utami, dan I. Kemal, “Implementation and Management of the Teaching Factory Learning Model at Vocational Schools,” Jurnal Penelitian Pendidikan IPA, vol. 11, no. 5, pp. 341–351, 2025.

[14] I. Imran, Marji, H. Suswanto, dan B. P. Adhikari, “The Influence of Teaching Factory (TEFA) Implementation and Work Readiness on Vocational High School Students’ Future Job Perspectives,” Jurnal Pendidikan Vokasi, 2024.

[15] SainSmart Genmitsu CNC Router 3018-PRO DIY Kit, SainSmart.com. https://www.sainsmart.com/products/sainsmart-genmitsu-cnc-router-3018-pro-diy-kit?srsltid=AfmBOoo7aUPZAlYFgbpACk-x1xd3rnIH1f6SruWOT1GdTHEJWTnbzRbp

[16] S. S. Pawar, T. C. Bera, dan K. S. Sangwan, "Modelling of spindle energy consumption in CNC milling," Procedia CIRP, vol. 105, pp. 192-197, 2022, doi: 10.1016/j.procir.2022.02.032.

[17] L. Zhou, F. Li, Y. Wang, L. Wang, dan G. Wang, "A new empirical standby power and auxiliary power model of CNC machine tools," Int. J. Adv. Manuf. Technol., vol. 120, pp. 3995-4010, 2022, doi: 10.1007/s00170-021-08274-x.

[18] Rojek, I.; Mikołajewski, D.;Mrozi´nski, A.; Macko, M.; Bednarek,T.; Tyburek, K. Internet of Things Applications for Energy Management in Buildings Using Artificial Intelligence—A Case Study. Energies 2025, 18, 1706.

[19] K. W. Kayohana, R. Asnawi, dan M. A. Amaria, “The Effectiveness of IoT-Based Learning Media for Supporting the Renewable Energy Engineering Expertise Program in SMK,” Jurnal Pendidikan Teknologi dan Kejuruan, vol. 29, no. 2, pp. 170–188, 2023.

[20] Y.-H. Chang, F.-C. Wu, dan H.-W. Lin, “Design and Implementation of ESP32-

© Hak Cipta milik Politeknik Negeri Jakarta

Based Edge Computing for Object Detection," Sensors, vol. 25, no. 6, art. 1656, 2025.

[21] L. Dutta, A. Debnath, and S. Biswas, "TinyML Meets IoT: A Comprehensive Survey," Journal of Network and Computer Applications, vol. 190, 102623, 2021. https://doi.org/10.1016/j.jnca.2021.102623

[22] C. Muñoz-Ausecha, J. Ruiz-Rosero, dan G. Ramirez-Gonzalez, "RFID Applications and Security Review," Computation, vol. 9, no. 6, art. 69, 2021.

[23] M. Krichen et al., "Convolutional Neural Networks: A Survey," Computers, vol. 12, no. 8, Art. 151, 2023

[24] S. Kiranyaz, T. Ince, dan M. Gabbouj, "1D Convolutional Neural Networks and Applications: A Survey," Mechanical Systems and Signal Processing, vol. 151, 2021

[25] P. Cocca et al., "Anomaly detection using electrical signature analysis and machine learning: application to a CNC mill," IFAC PapersOnLine, vol. 58, no. 8, pp. 569–574, 2024

[26] Elhanashi, A.; Dini, P.; Saponara, S.; Zheng, Q. Advancements in TinyML: Applications, Limitations, and Impact on IoT Devices. Electronics 2024, 13, 3562.

[27] J. Alcala, J. A. Juarez, V. Cardenas, S. Charre-Ibarra, J. Gonzalez-Rivera, dan J. Gudino-Lau, "Balancing Cost and Precision: An Experimental Evaluation of Sensors for Monitoring in Electrical Generation Systems," Sensors, vol. 25, no. 22, art. 7052, 2025, doi: 10.3390/s25227052.

[28] I. Raouf, H. Lee, dan H. S. Kim, "Mechanical fault detection based on machine learning for robotic RV reducer using electrical current signature analysis: a data-driven approach," J. Comput. Des. Eng., vol. 9, no. 2, pp. 417-433, 2022, doi: 10.1093/jcde/qwac015.

[29] S. Kim, K. Choi, H.-S. Choi, B. Lee, dan S. Yoon, "Towards a Rigorous Evaluation of Time-Series Anomaly Detection," Proc. AAAI Conf. Artif. Intell., vol. 36, no. 7, pp. 7194-7201, 2022, doi: 10.1609/aaai.v36i7.20680.

[30] I. L. Orasan, C. Seiculescu, dan C. D. Caleanu, "Benchmarking TensorFlow Lite Quantization Algorithms for Deep Neural Networks," in Proc. 2022 IEEE 16th Int. Symp. Applied Computational Intelligence and Informatics (SACI), 2022, pp. 221-226, doi: 10.1109/SACI55618.2022.9919465.

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

[31] W. Villegas-Ch, J. Govea, dan A. Jaramillo-Alcazar, "Tamper Detection in Industrial Sensors: An Approach Based on Anomaly Detection," Sensors, vol. 23, no. 21, art. 8908, 2023, doi: 10.3390/s23218908.

[32] M. Antonini, M. Pincheira, M. Vecchio, dan F. Antonelli, "An Adaptable and Unsupervised TinyML Anomaly Detection System for Extreme Industrial Environments," Sensors, vol. 23, no. 4, art. 2344, 2023, doi: 10.3390/s23042344.

[33] Z. Mao, B. Zhou, J. Huang, D. Liu, dan Q. Yang, "Research on Anomaly Detection Model for Power Consumption Data Based on Time-Series Reconstruction," Energies, vol. 17, no. 19, art. 4810, 2024, doi: 10.3390/en17194810.

[34] L. Christiansen, T. E. Hvidsten, J. H. Kristensen, J. Gebhardt, K. Mahmood, T. Otto, A. H. Lassen, T. D. Brunoe, C. Schou, dan E. S. Laursen, "A Framework for Developing Educational Industry 4.0 Activities and Study Materials," Educ. Sci., vol. 12, no. 10, art. 659, 2022, doi: 10.3390/educsci12100659.

[35] H. Al Joumaa, L. Al-Shrouf, dan M. Jelali, "Time Series-to-Image Encoding for Classification Using Convolutional Neural Networks: A Novel and Robust Approach," Mach. Learn. Knowl. Extr., vol. 7, no. 4, art. 155, 2025, doi: 10.3390/make7040155.

[36] M. A. K. Raiaan, S. Sakib, N. M. Fahad, A. Al Mamun, M. A. Rahman, S. Shatabda, dan M. S. H. Mukta, "A systematic review of hyperparameter optimization techniques in Convolutional Neural Networks," Decis. Anal. J., vol. 11, art. 100470, 2024, doi: 10.1016/j.dajour.2024.100470.

[37] G. Li, C. Wang, D. Zhang, dan G. Yang, "An Improved Feature Selection Method Based on Random Forest Algorithm for Wind Turbine Condition Monitoring," Sensors, vol. 21, no. 16, art. 5654, 2021, doi: 10.3390/s21165654.

[38] A. B. Montejano Leija, E. Ruiz Beltran, J. L. Orozco Mora, dan J. O. Valdes Valadez, "Performance of Machine Learning Algorithms in Fault Diagnosis for Manufacturing Systems: A Comparative Analysis," Processes, vol. 13, no. 6, art. 1624, 2025, doi: 10.3390/pr13061624.

[39] F. Cappelli, G. Castronuovo, S. Grimaldi, dan V. Telesca, "Random Forest and Feature Importance Measures for Discriminating the Most Influential Environmental Factors in Predicting Cardiovascular and Respiratory Diseases," Int. J. Environ. Res. Public Health, vol. 21, no. 7, art. 867, 2024, doi: 10.3390/ijerph21070867.

[40] J. Ali dan M. H. Zafar, "Improved End-to-end service assurance and mathematical modeling of message queuing telemetry transport protocol based massively deployed fully functional devices in smart cities," Alexandria Engineering Journal, vol. 72, pp. 657-672,

© 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

2023, doi: 10.1016/j.aej.2023.04.014.

© Hak Cipta milik Politeknik Negeri Jakarta

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

LAMPIRAN

A. Program Embedded CNN 2 Class

```
## 1. Install & import library
!pip install -q tensorflow scikit-learn pandas numpy matplotlib seaborn

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import tensorflow as tf
from tensorflow.keras import layers, models, callbacks
from sklearn.ensemble import RandomForestClassifier
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import accuracy_score, classification_report,
confusion_matrix
from sklearn.utils.class_weight import compute_class_weight

print("TensorFlow version:", tf.__version__)

## 2. Upload data mentah
from google.colab import files
uploaded = files.upload()  # pilih data_5label.csv
raw_csv_path = list(uploaded.keys())[0]
print("File terupload:", raw_csv_path)

## 3. Eksplorasi awal data mentah
raw = pd.read_csv(raw_csv_path, sep=';')
print("Jumlah baris:", len(raw))
print("Kolom:", list(raw.columns))
print("\nDistribusi label:")
print(raw['kondisi_mesin'].value_counts())
print("\nMissing value per kolom:")
print(raw.isna().sum())
raw.head()

raw['waktu'] = pd.to_datetime(raw['waktu'])
fig, ax = plt.subplots(1, 2, figsize=(12,4))
ax[0].plot(raw['energi_use'].values)
ax[0].set_title('energi_use (KUMULATIF, selalu naik)')
ax[1].plot(raw['biaya'].values)
ax[1].set_title('biaya (KUMULATIF, selalu naik)')
plt.tight_layout()
plt.show()
print("Lihat plot di atas -> kedua kolom ini naik terus dari awal sampai akhir
```

© 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

```
rekaman,")
print("tidak turun/reset walau kondisi mesin berganti-ganti. Ini bukti nyata
kenapa")
print("kolom ini HARUS dibuang dari fitur model.")

## 4. Cleaning data
n0 = len(raw)

# (a) Buang sensor glitch
glitch_mask = (raw['frekuensi'] < 49.5) | (raw['frekuensi'] > 50.5)
print(f"(a) Baris glitch sensor dibuang: {glitch_mask.sum()}")
df = raw[~glitch_mask].reset_index(drop=True)

# (b) Buang kolom yang tidak boleh jadi fitur
df = df.drop(columns=['id', 'nama_pengguna', 'energi_use', 'biaya'])

# (c) Bentuk blok kontinu per kondisi, buang blok yang terlalu pendek
change = (df['kondisi_mesin'] != df['kondisi_mesin'].shift()).cumsum()
df['block_id_temp'] = change
block_len = df.groupby('block_id_temp')['kondisi_mesin'].transform('count')

MIN_BLOCK_LEN = 60
short_mask = block_len < MIN_BLOCK_LEN
print(f"(c) Baris dibuang krn blok pendek (<{MIN_BLOCK_LEN} sampel
kontinu): {short_mask.sum()}")
df_clean = df[~short_mask].reset_index(drop=True)

# Re-index block_id supaya berurutan rapi
new_ids = {old:i+1 for i, old in enumerate(df_clean['block_id_temp'].unique())}
df_clean['block_id'] = df_clean['block_id_temp'].map(new_ids)
df_clean = df_clean.drop(columns=['block_id_temp'])

print(f"\nBaris awal: {n0}  ->  Baris akhir: {len(df_clean)}
({len(df_clean)/n0*100:.1f}% data tersisa)")
print("\nDistribusi label setelah cleaning:")
print(df_clean['kondisi_mesin'].value_counts())
print("\nJumlah blok/sesi kontinu per label:")
print(df_clean.groupby('kondisi_mesin')['block_id'].nunique())

df_clean.to_csv('data_5label_cleaned.csv', sep=';', index=False)
print("\nTersimpan sebagai data_5label_cleaned.csv")

## 5. Analisis statistik: fitur mana yang paling berpengaruh?
FEAT_COLS =
['tegangan','arus','daya_aktif','daya_reaktif','daya_apparent','faktor_daya','frekuensi
']
WINDOW = 30
STRIDE = 5
```

© Hak Cipta milik Politeknik Negeri Jakarta

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

```
rows = []
for bid, g in df_clean.groupby('block_id'):
    g = g.sort_values('waktu')
    arr = g[FEAT_COLS].values
    label = g['kondisi_mesin'].iloc[0]
    n = len(arr)
    if n < WINDOW:
        continue
    for start in range(0, n - WINDOW + 1, STRIDE):
        w = arr[start:start+WINDOW]
        feat = {}
        for i, c in enumerate(FEAT_COLS):
            col = w[:, i]
            feat[f'{c}_mean'] = col.mean()
            feat[f'{c}_std'] = col.std()
            feat[f'{c}_min'] = col.min()
            feat[f'{c}_max'] = col.max()
            feat[f'{c}_slope'] = np.polyfit(np.arange(WINDOW), col, 1)[0]
        feat['label'] = label
        feat['label_bin'] = 'Normal' if label == 'Normal(2)' else 'Abnormal'
        rows.append(feat)

wf = pd.DataFrame(rows)
feat_cols_all = [c for c in wf.columns if c not in ['label', 'label_bin']]
print(f"Jumlah window untuk analisis: {len(wf)}")

# Random Forest untuk target BINARY (Normal vs Abnormal)
rf2 = RandomForestClassifier(n_estimators=300, random_state=42,
class_weight='balanced', n_jobs=-1)
rf2.fit(wf[feat_cols_all], wf['label_bin'])
imp2 = pd.Series(rf2.feature_importances_,
index=feat_cols_all).sort_values(ascending=False)

def base_feat(name):
    for suf in ['_mean','_std','_min','_max','_slope']:
        if name.endswith(suf):
            return name[:-len(suf)]
    return name

agg2 = imp2.groupby(base_feat).sum().sort_values(ascending=False)
print("=== Kontribusi tiap FITUR DASAR (dijumlahkan dari
mean/std/min/max/slope) ===")
print(agg2)

plt.figure(figsize=(8,4))
agg2.plot(kind='barh')
plt.title('Feature Importance per Fitur Dasar (Normal vs Abnormal)')
```

© 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

```
plt.xlabel('Total importance')
plt.gca().invert_yaxis()
plt.tight_layout()
plt.show()

print("\nTop 10 fitur turunan paling berpengaruh (paling detail):")
print(imp2.head(10))

## 6. Bangun window per sesi (block) + split anti-data-leakage untuk training
CNN
def add_derived_features(df):
    df = df.copy()
    for col in ['arus', 'daya_aktif', 'faktor_daya']:
        df[f'{col}_delta'] = df.groupby('block_id')[col].diff().fillna(0.0)
        df[f'{col}_rstd'] = (
            df.groupby('block_id')[col]
              .rolling(window=5, min_periods=1).std()
              .reset_index(level=0, drop=True)
              .fillna(0.0)
        )
    return df

def split_blocks_by_label(df, val_blocks_per_label=1, test_blocks_per_label=1):
    block_info =
df.groupby('block_id').agg(label=('kondisi_mesin','first')).reset_index()
    train_ids, val_ids, test_ids = [], [], []
    for label in block_info['label'].unique():
        ids = sorted(block_info.loc[block_info['label']==label, 'block_id'].tolist())
        n_test = min(test_blocks_per_label, max(1, len(ids)//5))
        n_val = min(val_blocks_per_label, max(1, len(ids)//5))
        test_ids += ids[-n_test:]
        val_ids += ids[-n_test-n_val:-n_test]
        train_ids += ids[:-n_test-n_val]
    return set(train_ids), set(val_ids), set(test_ids)

def make_windows(df, feat_cols, window_size=30, stride=5, block_ids=None):
    X, y, groups = [], [], []
    for bid, g in df.groupby('block_id'):
        if block_ids is not None and bid not in block_ids:
            continue
        g = g.sort_values('waktu')
        arr = g[feat_cols].values
        label = g['kondisi_mesin'].iloc[0]
        n = len(arr)
        if n < window_size:
            continue
        for start in range(0, n - window_size + 1, stride):
            X.append(arr[start:start+window_size])
```

© 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

```
            y.append(label)
            groups.append(bid)
    return np.array(X, dtype=np.float32), np.array(y), np.array(groups)

print("Fungsi bantu siap.")

SEED = 7  # dipilih via validasi (dicoba 10 seed, ini yang val_accuracy terbaik)
np.random.seed(SEED)
tf.random.set_seed(SEED)

df_derived = add_derived_features(df_clean)
train_ids, val_ids, test_ids = split_blocks_by_label(df_clean)
print(f"Blok -> train={len(train_ids)}  val={len(val_ids)}  test={len(test_ids)}")

DERIVED_FEATS = FEAT_COLS + ['arus_delta', 'daya_aktif_delta',
'faktor_daya_delta',
                             'arus_rstd', 'daya_aktif_rstd', 'faktor_daya_rstd']

def prep(block_ids):
    X, y, _ = make_windows(df_derived, DERIVED_FEATS, WINDOW,
STRIDE, block_ids)
    y_bin = np.where(y == 'Normal(2)', 'Normal', 'Abnormal')
    return X, y_bin

Xtr, ytr = prep(train_ids)
Xva, yva = prep(val_ids)
Xte, yte = prep(test_ids)

mean = Xtr.reshape(-1, Xtr.shape[-1]).mean(axis=0)
std = Xtr.reshape(-1, Xtr.shape[-1]).std(axis=0) + 1e-6

def scale(X):
    return (X - mean) / std

Xtr_s, Xva_s, Xte_s = scale(Xtr), scale(Xva), scale(Xte)

le = LabelEncoder()
le.fit(['Abnormal', 'Normal'])  # index 0 = Abnormal, 1 = Normal (fixed order)
ytr_e, yva_e, yte_e = le.transform(ytr), le.transform(yva), le.transform(yte)

cw = compute_class_weight('balanced', classes=np.unique(ytr_e), y=ytr_e)
class_weight = {i: w for i, w in enumerate(cw)}
print("Kelas:", list(le.classes_), " | class_weight:", class_weight)
print(f"Jumlah window -> train={len(Xtr_s)}  val={len(Xva_s)}
test={len(Xte_s)}")
```

7. Bangun & training model CNN 1D (kecil, ramah untuk ESP32)

© 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

```
n_feat = Xtr_s.shape[-1]

model = models.Sequential([
    layers.Input(shape=(WINDOW, n_feat), name='sensor_window'),
    layers.Conv1D(32, 5, activation='relu', padding='same'),
    layers.BatchNormalization(),
    layers.MaxPooling1D(2),
    layers.Conv1D(64, 5, activation='relu', padding='same'),
    layers.BatchNormalization(),
    layers.GlobalAveragePooling1D(),
    layers.Dense(32, activation='relu'),
    layers.Dropout(0.3),
    layers.Dense(2, activation='softmax', name='output'),
], name='cnc_binary_lite')

model.compile(optimizer=tf.keras.optimizers.Adam(1e-3),
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])
model.summary()

es = callbacks.EarlyStopping(monitor='val_accuracy', patience=20,
restore_best_weights=True)
rlr = callbacks.ReduceLROnPlateau(monitor='val_loss', factor=0.5, patience=6)

history = model.fit(Xtr_s, ytr_e, validation_data=(Xva_s, yva_e),
                    epochs=200, batch_size=32, class_weight=class_weight,
                    callbacks=[es, rlr], verbose=1)

## 8. Evaluasi jujur (test set = sesi rekaman yang tidak pernah dilihat model)
pred = np.argmax(model.predict(Xte_s, verbose=0), axis=1)
acc = accuracy_score(yte_e, pred)
print(f"Akurasi test: {acc:.4f}\n")
print(classification_report(yte_e, pred, target_names=le.classes_, digits=3))
print("Confusion matrix (baris=aktual, kolom=prediksi):")
cm = confusion_matrix(yte_e, pred)
print(cm)

plt.figure(figsize=(4,4))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=le.classes_,
yticklabels=le.classes_)
plt.xlabel('Prediksi'); plt.ylabel('Aktual'); plt.title('Confusion Matrix')
plt.show()

## 9. Export ke TensorFlow Lite (float32 & int8 quantized untuk ESP32)
converter = tf.lite.TFLiteConverter.from_keras_model(model)
tflite_float = converter.convert()
with open('model_binary.tflite', 'wb') as f:
    f.write(tflite_float)
```

© Hak Cipta milik Politeknik Negeri Jakarta

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

```
print(f"Ukuran model float32: {len(tflite_float)/1024:.1f} KB")

def representative_dataset():
    idx = np.random.choice(len(Xtr_s), size=min(200, len(Xtr_s)), replace=False)
    for i in idx:
        yield [Xtr_s[i:i+1].astype(np.float32)]

converter = tf.lite.TFLiteConverter.from_keras_model(model)
converter.optimizations = [tf.lite.Optimize.DEFAULT]
converter.representative_dataset = representative_dataset
converter.target_spec.supported_ops = [tf.lite.OpsSet.TFLITE_BUILTINS_INT8]
converter.inference_input_type = tf.int8
converter.inference_output_type = tf.int8
tflite_int8 = converter.convert()
with open('model_binary_int8.tflite', 'wb') as f:
    f.write(tflite_int8)
print(f"Ukuran model int8   : {len(tflite_int8)/1024:.1f} KB")

interpreter = tf.lite.Interpreter(model_content=tflite_int8)
interpreter.allocate_tensors()
in_detail = interpreter.get_input_details()[0]
out_detail = interpreter.get_output_details()[0]
in_scale, in_zp = in_detail['quantization']
out_scale, out_zp = out_detail['quantization']
print(f"Quantization -> input: scale={in_scale:.6f} zp={in_zp} | output:
scale={out_scale:.6f} zp={out_zp}")

preds_int8 = []
for i in range(len(Xte_s)):
    x = Xte_s[i:i+1].astype(np.float32)
    x_q = np.round(x / in_scale + in_zp).astype(np.int8)
    interpreter.set_tensor(in_detail['index'], x_q)
    interpreter.invoke()
    out_q = interpreter.get_tensor(out_detail['index'])
    out_f = (out_q.astype(np.float32) - out_zp) * out_scale
    preds_int8.append(np.argmax(out_f))

acc_int8 = accuracy_score(yte_e, preds_int8)
print(f"Akurasi setelah quantization int8: {acc_int8:.4f}  (float32: {acc:.4f})")

## 10. Generate `model_data.h` dan `scaler_params.h` untuk Arduino/ESP32
def to_c_array(data, var_name):
    lines = [f"// Auto-generated dari model_binary_int8.tflite - JANGAN edit
manual",
             f"#pragma once",
             f"alignas(8) const unsigned char {var_name}[] = {{"]
    for i in range(0, len(data), 12):
        chunk = data[i:i+12]
```

© 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

```
        lines.append("  " + ", ".join(f"0x{b:02x}" for b in chunk) + ",")
    lines.append("};")
    lines.append(f"const unsigned int {var_name}_len = {len(data)};")
    return "\n".join(lines)

with open('model_data.h', 'w') as f:
    f.write(to_c_array(tflite_int8, "g_cnc_model_int8"))

feat_names = DERIVED_FEATS
lines = [
    "// Auto-generated - parameter normalisasi & quantization model",
    "#pragma once",
    "",
    f"#define WINDOW_SIZE {WINDOW}",
    f"#define N_FEATURES {n_feat}",
    "",
    "// Urutan fitur HARUS sama persis dengan urutan ini saat firmware menyusun
window:",
    "// " + ", ".join(feat_names),
    "",
    "const float FEATURE_MEAN[N_FEATURES] = {" + ", ".join(f"{v:.6f}f" for
v in mean) + "};",
    "const float FEATURE_STD[N_FEATURES]  = {" + ", ".join(f"{v:.6f}f" for v
in std) + "};",
    "",
    f"const float INPUT_SCALE = {in_scale:.8f}f;",
    f"const int   INPUT_ZERO_POINT = {in_zp};",
    f"const float OUTPUT_SCALE = {out_scale:.8f}f;",
    f"const int   OUTPUT_ZERO_POINT = {out_zp};",
    "",
    "// Index kelas output model: 0 = Abnormal, 1 = Normal",
    'const char* CLASS_NAMES[2] = {"Abnormal", "Normal"};',
]
with open('scaler_params.h', 'w') as f:
    f.write("\n".join(lines))

print("File siap: model_data.h, scaler_params.h, model_binary_int8.tflite,
model_binary.tflite")
print("Juga tersedia: data_5label_cleaned.csv (hasil cleaning)")

## 11. Download semua file hasil ke komputer kamu
from google.colab import files

for fname in ['data_5label_cleaned.csv', 'model_data.h', 'scaler_params.h',
              'model_binary_int8.tflite', 'model_binary.tflite']:
    files.download(fname)
```

© 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

B. Program Embedded CNN Multi Class

```
## 1. Install & import library
!pip install -q tensorflow scikit-learn pandas numpy matplotlib seaborn

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import tensorflow as tf
from tensorflow.keras import layers, models, callbacks
from sklearn.ensemble import RandomForestClassifier
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import accuracy_score, classification_report,
confusion_matrix
from sklearn.utils.class_weight import compute_class_weight

print("TensorFlow version:", tf.__version__)

## 2. Upload data mentah
Upload file `data_5label.csv` (data mentah asli, BELUM dibersihkan).

from google.colab import files
uploaded = files.upload()  # pilih data_5label.csv
raw_csv_path = list(uploaded.keys())[0]
print("File terupload:", raw_csv_path)

## 3. Eksplorasi awal data mentah
raw = pd.read_csv(raw_csv_path, sep=';')
print("Jumlah baris:", len(raw))
print("Kolom:", list(raw.columns))
print("\nDistribusi label:")
print(raw['kondisi_mesin'].value_counts())
print("\nMissing value per kolom:")
print(raw.isna().sum())
raw.head()

raw['waktu'] = pd.to_datetime(raw['waktu'])
fig, ax = plt.subplots(1, 2, figsize=(12,4))
ax[0].plot(raw['energi_use'].values)
ax[0].set_title('energi_use (KUMULATIF, selalu naik)')
ax[1].plot(raw['biaya'].values)
ax[1].set_title('biaya (KUMULATIF, selalu naik)')
plt.tight_layout()
plt.show()
print("Lihat plot di atas -> kedua kolom ini naik terus dari awal sampai akhir
rekaman,")
print("tidak turun/reset walau kondisi mesin berganti-ganti. Ini bukti nyata
kenapa")
```

© Hak Cipta milik Politeknik Negeri Jakarta

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

```
print("kolom ini HARUS dibuang dari fitur model.")

## 4. Cleaning data
n0 = len(raw)

# (a) Buang sensor glitch
glitch_mask = (raw['frekuensi'] < 49.5) | (raw['frekuensi'] > 50.5)
print(f"(a) Baris glitch sensor dibuang: {glitch_mask.sum()}")
df = raw[~glitch_mask].reset_index(drop=True)

# (b) Buang kolom yang tidak boleh jadi fitur
df = df.drop(columns=['id', 'nama_pengguna', 'energi_use', 'biaya'])

# (c) Bentuk blok kontinu per kondisi, buang blok yang terlalu pendek
change = (df['kondisi_mesin'] != df['kondisi_mesin'].shift()).cumsum()
df['block_id_temp'] = change
block_len = df.groupby('block_id_temp')['kondisi_mesin'].transform('count')

MIN_BLOCK_LEN = 60
short_mask = block_len < MIN_BLOCK_LEN
print(f"(c) Baris dibuang krn blok pendek (<{MIN_BLOCK_LEN} sampel
kontinu): {short_mask.sum()}")
df_clean = df[~short_mask].reset_index(drop=True)

# Re-index block_id supaya berurutan rapi
new_ids = {old:i+1 for i, old in enumerate(df_clean['block_id_temp'].unique())}
df_clean['block_id'] = df_clean['block_id_temp'].map(new_ids)
df_clean = df_clean.drop(columns=['block_id_temp'])

print(f"\nBaris awal: {n0}  ->  Baris akhir: {len(df_clean)}
({len(df_clean)/n0*100:.1f}% data tersisa)")
print("\nDistribusi label setelah cleaning:")
print(df_clean['kondisi_mesin'].value_counts())
print("\nJumlah blok/sesi kontinu per label:")
print(df_clean.groupby('kondisi_mesin')['block_id'].nunique())

df_clean.to_csv('data_5label_cleaned.csv', sep=';', index=False)
print("\nTersimpan sebagai data_5label_cleaned.csv")

## 5. Analisis statistik: fitur mana yang paling berpengaruh?
FEAT_COLS =
['tegangan','arus','daya_aktif','daya_reaktif','daya_apparent','faktor_daya','frekuensi
']
WINDOW = 30
STRIDE = 5

rows = []
for bid, g in df_clean.groupby('block_id'):
```

© 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

```
    g = g.sort_values('waktu')
    arr = g[FEAT_COLS].values
    label = g['kondisi_mesin'].iloc[0]
    n = len(arr)
    if n < WINDOW:
        continue
    for start in range(0, n - WINDOW + 1, STRIDE):
        w = arr[start:start+WINDOW]
        feat = {}
        for i, c in enumerate(FEAT_COLS):
            col = w[:, i]
            feat[f'{c}_mean'] = col.mean()
            feat[f'{c}_std'] = col.std()
            feat[f'{c}_min'] = col.min()
            feat[f'{c}_max'] = col.max()
            feat[f'{c}_slope'] = np.polyfit(np.arange(WINDOW), col, 1)[0]
        feat['label'] = label
        feat['label_bin'] = 'Normal' if label == 'Normal(2)' else 'Abnormal'
        rows.append(feat)

wf = pd.DataFrame(rows)
feat_cols_all = [c for c in wf.columns if c not in ['label', 'label_bin']]
print(f"Jumlah window untuk analisis: {len(wf)}")

# Random Forest untuk target MULTICLASS (Normal / tumpul / overwork /
stuck / patah)
rf5 = RandomForestClassifier(n_estimators=300, random_state=42,
class_weight='balanced', n_jobs=-1)
rf5.fit(wf[feat_cols_all], wf['label'])
imp5 = pd.Series(rf5.feature_importances_,
index=feat_cols_all).sort_values(ascending=False)

def base_feat(name):
    for suf in ['_mean','_std','_min','_max','_slope']:
        if name.endswith(suf):
            return name[:-len(suf)]
    return name

agg5 = imp5.groupby(base_feat).sum().sort_values(ascending=False)
print("=== Kontribusi tiap FITUR DASAR (dijumlahkan dari
mean/std/min/max/slope) - MULTICLASS ===")
print(agg5)

plt.figure(figsize=(8,4))
agg5.plot(kind='barh')
plt.title('Feature Importance per Fitur Dasar (5 kelas kondisi mesin)')
plt.xlabel('Total importance')
plt.gca().invert_yaxis()
```

© 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

```
plt.tight_layout()
plt.show()

print("\nTop 10 fitur turunan paling berpengaruh (paling detail):")
print(imp5.head(10))

## 6. Bangun window per sesi (block) + split anti-data-leakage untuk training
CNN
def add_derived_features(df):
    """Tambahkan fitur dinamis (delta & rolling std) per block, tanpa bocor lintas
blok."""
    df = df.copy()
    for col in ['arus', 'daya_aktif', 'faktor_daya']:
        df[f'{col}_delta'] = df.groupby('block_id')[col].diff().fillna(0.0)
        df[f'{col}_rstd'] = (
            df.groupby('block_id')[col]
              .rolling(window=5, min_periods=1).std()
              .reset_index(level=0, drop=True)
              .fillna(0.0)
        )
    return df

def split_blocks_by_label(df, val_blocks_per_label=1, test_blocks_per_label=1):
    block_info =
df.groupby('block_id').agg(label=('kondisi_mesin','first')).reset_index()
    train_ids, val_ids, test_ids = [], [], []
    for label in block_info['label'].unique():
        ids = sorted(block_info.loc[block_info['label']==label, 'block_id'].tolist())
        n_test = min(test_blocks_per_label, max(1, len(ids)//5))
        n_val = min(val_blocks_per_label, max(1, len(ids)//5))
        test_ids += ids[-n_test:]
        val_ids += ids[-n_test-n_val:-n_test]
        train_ids += ids[:-n_test-n_val]
    return set(train_ids), set(val_ids), set(test_ids)

def make_windows(df, feat_cols, window_size=30, stride=5, block_ids=None):
    X, y, groups = [], [], []
    for bid, g in df.groupby('block_id'):
        if block_ids is not None and bid not in block_ids:
            continue
        g = g.sort_values('waktu')
        arr = g[feat_cols].values
        label = g['kondisi_mesin'].iloc[0]
        n = len(arr)
        if n < window_size:
            continue
        for start in range(0, n - window_size + 1, stride):
            X.append(arr[start:start+window_size])
```

© 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

```
            y.append(label)
            groups.append(bid)
    return np.array(X, dtype=np.float32), np.array(y), np.array(groups)

print("Fungsi bantu siap.")

SEED = 7  # SAMA dengan versi binary, supaya perbandingan adil
np.random.seed(SEED)
tf.random.set_seed(SEED)

df_derived = add_derived_features(df_clean)
train_ids, val_ids, test_ids = split_blocks_by_label(df_clean)
print(f"Blok -> train={len(train_ids)}  val={len(val_ids)}  test={len(test_ids)}")

DERIVED_FEATS = FEAT_COLS + ['arus_delta', 'daya_aktif_delta',
'faktor_daya_delta',
                             'arus_rstd', 'daya_aktif_rstd', 'faktor_daya_rstd']

def prep(block_ids):
    X, y, _ = make_windows(df_derived, DERIVED_FEATS, WINDOW,
STRIDE, block_ids)
    return X, y  # y = label asli 5 kelas, TIDAK dikonversi ke Normal/Abnormal

Xtr, ytr = prep(train_ids)
Xva, yva = prep(val_ids)
Xte, yte = prep(test_ids)

mean = Xtr.reshape(-1, Xtr.shape[-1]).mean(axis=0)
std = Xtr.reshape(-1, Xtr.shape[-1]).std(axis=0) + 1e-6

def scale(X):
    return (X - mean) / std

Xtr_s, Xva_s, Xte_s = scale(Xtr), scale(Xva), scale(Xte)

le = LabelEncoder()
le.fit(sorted(df_clean['kondisi_mesin'].unique()))  # 5 kelas, urutan alfabetis tetap
ytr_e, yva_e, yte_e = le.transform(ytr), le.transform(yva), le.transform(yte)

cw = compute_class_weight('balanced', classes=np.unique(ytr_e), y=ytr_e)
class_weight = {i: w for i, w in enumerate(cw)}
print("Kelas:", list(le.classes_), " | class_weight:", class_weight)
print(f"Jumlah window -> train={len(Xtr_s)}  val={len(Xva_s)}
test={len(Xte_s)}")

## 7. Bangun & training model CNN 1D (kecil, ramah untuk ESP32)
n_feat = Xtr_s.shape[-1]
n_classes = len(le.classes_)  # 5 kelas
```

© 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

```
model = models.Sequential([
    layers.Input(shape=(WINDOW, n_feat), name='sensor_window'),
    layers.Conv1D(32, 5, activation='relu', padding='same'),
    layers.BatchNormalization(),
    layers.MaxPooling1D(2),
    layers.Conv1D(64, 5, activation='relu', padding='same'),
    layers.BatchNormalization(),
    layers.GlobalAveragePooling1D(),
    layers.Dense(32, activation='relu'),
    layers.Dropout(0.3),
    layers.Dense(n_classes, activation='softmax', name='output'),
], name='cnc_multiclass_lite')

model.compile(optimizer=tf.keras.optimizers.Adam(1e-3),
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])
model.summary()

es = callbacks.EarlyStopping(monitor='val_accuracy', patience=20,
restore_best_weights=True)
rlr = callbacks.ReduceLROnPlateau(monitor='val_loss', factor=0.5, patience=6)

history = model.fit(Xtr_s, ytr_e, validation_data=(Xva_s, yva_e),
                    epochs=200, batch_size=32, class_weight=class_weight,
                    callbacks=[es, rlr], verbose=1)

## 8. Evaluasi jujur (test set = sesi rekaman yang tidak pernah dilihat model)
pred = np.argmax(model.predict(Xte_s, verbose=0), axis=1)
acc = accuracy_score(yte_e, pred)
print(f"Akurasi test: {acc:.4f}\n")
print(classification_report(yte_e, pred, target_names=le.classes_, digits=3))
print("Confusion matrix (baris=aktual, kolom=prediksi):")
cm = confusion_matrix(yte_e, pred)
print(cm)

plt.figure(figsize=(4,4))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=le.classes_,
yticklabels=le.classes_)
plt.xlabel('Prediksi'); plt.ylabel('Aktual'); plt.title('Confusion Matrix')
plt.show()

## 9. Export ke TensorFlow Lite (float32 & int8 quantized untuk ESP32)
converter = tf.lite.TFLiteConverter.from_keras_model(model)
tflite_float = converter.convert()
with open('model_multiclass.tflite', 'wb') as f:
    f.write(tflite_float)
print(f"Ukuran model float32: {len(tflite_float)/1024:.1f} KB")
```

© Hak Cipta milik Politeknik Negeri Jakarta

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

```
def representative_dataset():
    idx = np.random.choice(len(Xtr_s), size=min(200, len(Xtr_s)), replace=False)
    for i in idx:
        yield [Xtr_s[i:i+1].astype(np.float32)]

converter = tf.lite.TFLiteConverter.from_keras_model(model)
converter.optimizations = [tf.lite.Optimize.DEFAULT]
converter.representative_dataset = representative_dataset
converter.target_spec.supported_ops = [tf.lite.OpsSet.TFLITE_BUILTINS_INT8]
converter.inference_input_type = tf.int8
converter.inference_output_type = tf.int8
tflite_int8 = converter.convert()
with open('model_multiclass_int8.tflite', 'wb') as f:
    f.write(tflite_int8)
print(f"Ukuran model int8   : {len(tflite_int8)/1024:.1f} KB")

interpreter = tf.lite.Interpreter(model_content=tflite_int8)
interpreter.allocate_tensors()
in_detail = interpreter.get_input_details()[0]
out_detail = interpreter.get_output_details()[0]
in_scale, in_zp = in_detail['quantization']
out_scale, out_zp = out_detail['quantization']
print(f"Quantization -> input: scale={in_scale:.6f} zp={in_zp} | output:
scale={out_scale:.6f} zp={out_zp}")

preds_int8 = []
for i in range(len(Xte_s)):
    x = Xte_s[i:i+1].astype(np.float32)
    x_q = np.round(x / in_scale + in_zp).astype(np.int8)
    interpreter.set_tensor(in_detail['index'], x_q)
    interpreter.invoke()
    out_q = interpreter.get_tensor(out_detail['index'])
    out_f = (out_q.astype(np.float32) - out_zp) * out_scale
    preds_int8.append(np.argmax(out_f))

acc_int8 = accuracy_score(yte_e, preds_int8)
print(f"Akurasi setelah quantization int8: {acc_int8:.4f}  (float32: {acc:.4f})")

## 10. Generate `model_data.h` dan `scaler_params.h` untuk Arduino/ESP32
def to_c_array(data, var_name):
    lines = [f"// Auto-generated dari model_multiclass_int8.tflite - JANGAN edit
manual",
             f"#pragma once",
             f"alignas(8) const unsigned char {var_name}[] = {{"]
    for i in range(0, len(data), 12):
        chunk = data[i:i+12]
        lines.append("  " + ", ".join(f"0x{b:02x}" for b in chunk) + ",")
```

© 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

```
    lines.append("};")
    lines.append(f"const unsigned int {var_name}_len = {len(data)};")
    return "\n".join(lines)

with open('model_data.h', 'w') as f:
    f.write(to_c_array(tflite_int8, "g_cnc_model_multiclass_int8"))

feat_names = DERIVED_FEATS
class_names_c = ", ".join(f'"{c}"' for c in le.classes_)
lines = [
    "// Auto-generated - parameter normalisasi & quantization model
(MULTICLASS)",
    "#pragma once",
    "",
    f"#define WINDOW_SIZE {WINDOW}",
    f"#define N_FEATURES {n_feat}",
    f"#define N_CLASSES {n_classes}",
    "",
    "// Urutan fitur HARUS sama persis dengan urutan ini saat firmware menyusun
window:",
    "// " + ", ".join(feat_names),
    "",
    "const float FEATURE_MEAN[N_FEATURES] = {" + ", ".join(f"{v:.6f}f" for
v in mean) + "};",
    "const float FEATURE_STD[N_FEATURES]  = {" + ", ".join(f"{v:.6f}f" for v
in std) + "};",
    "",
    f"const float INPUT_SCALE = {in_scale:.8f}f;",
    f"const int   INPUT_ZERO_POINT = {in_zp};",
    f"const float OUTPUT_SCALE = {out_scale:.8f}f;",
    f"const int   OUTPUT_ZERO_POINT = {out_zp};",
    "",
    "// Index kelas output model, urutan sesuai LabelEncoder (alfabetis):",
    "// " + ", ".join(f"{i}={c}" for i, c in enumerate(le.classes_)),
    f'const char* CLASS_NAMES[N_CLASSES] = {{{class_names_c}}};',
]
with open('scaler_params.h', 'w') as f:
    f.write("\n".join(lines))

print("File siap: model_data.h, scaler_params.h, model_multiclass_int8.tflite,
model_multiclass.tflite")
print("Juga tersedia: data_5label_cleaned.csv (hasil cleaning)")

## 11. Download semua file hasil ke komputer kamu
from google.colab import files

for fname in ['data_5label_cleaned.csv', 'model_data.h', 'scaler_params.h',
              'model_multiclass_int8.tflite', 'model_multiclass.tflite']:
```

© 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

```
files.download(fname)
```

C. Program ESP32

```
// Library Sistem Monitoring
//=============================================================
============================
#include <WiFi.h>
#include <WiFiMulti.h>          // Library untuk multi Wi-Fi
#include <MQTTClient.h>
#include <PZEM004Tv30.h>
#include <LiquidCrystal_I2C.h>
#include <RTClib.h>
#include <Adafruit_PN532.h>
#include <Wire.h>
//=============================================================
============================

// Library CNN
//=============================================================
============================
// PENTING: include TFLite Micro PERSIS 3 baris ini saja (sesuai library yang
sudah
// terbukti jalan di environment kamu). JANGAN tambahkan
<TensorFlowLite_ESP32.h>
// atau "tensorflow/lite/micro/micro_log.h" -> itu penyebab error compile
sebelumnya
// (file header itu tidak ada di versi library kamu).
#include "tensorflow/lite/micro/micro_mutable_op_resolver.h"
#include "tensorflow/lite/micro/micro_interpreter.h"
#include "tensorflow/lite/schema/schema_generated.h"

#include "model_data.h"       // model CNN binary hasil training (array int8)
#include "scaler_params.h"    // WINDOW_SIZE, N_FEATURES, mean/std,
quant params
//=============================================================
============================

// Variabel Sistem Monitoring
//=============================================================
============================
// Variabel Wifi
WiFiMulti wifiMulti;
//-----------------------------------------------------------------------------
// const char WIFI_SSID[] = "xxxx";  // CHANGE WIFI SSID
// const char WIFI_PASSWORD[] = "xxxx";  // CHANGE WIFI PASSWORD
//-----------------------------------------------------------------------------

// Variabel MQTT
```

© 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

```
//---------------------------------------------------------------------------
const char MQTT_BROKER_ADRRESS[] = "xxxx";  // CHANGE TO MQTT BROKER'S ADDRESS
const int MQTT_PORT = 1883;
const char MQTT_CLIENT_ID[] = "xxxx";  // CHANGE IT AS YOU DESIRE
const char MQTT_USERNAME[] = "";  // CHANGE IT IF REQUIRED, empty if not required
const char MQTT_PASSWORD[] = "";  // CHANGE IT IF REQUIRED, empty if not required

// The MQTT topics that ESP32 should publish/subscribe
//---------------------------------------------------------------------------
const char PUBLISH_TOPIC[] = "xxxx";    // CHANGE IT AS YOU DESIRE
const char SUBSCRIBE_TOPIC[] = "xxxx";  // CHANGE IT AS YOU DESIRE

const int PUBLISH_INTERVAL = 1000;  // in milliseconds

WiFiClient network;
MQTTClient mqtt = MQTTClient(256);

unsigned long lastPublishTime = 0;
String msgStr = "";  // MQTT message buffer
//---------------------------------------------------------------------------

//Variabel NFC PN532
//---------------------------------------------------------------------------
#define SDA_PIN 7
#define SCL_PIN 6
Adafruit_PN532 nfc(SDA_PIN, SCL_PIN);

String id_allowed [] = {"335dcf11", "2ea5fb03", "042e6671", "566c4797", "ffb0e91f"};
String nama_kartu;
String idcard;
//---------------------------------------------------------------------------

// //Variabel PZEM
// //---------------------------------------------------------------------------
// #if defined(ESP32)
// PZEM004Tv30 pzem(Serial2, 16, 17);
// #else
// PZEM004Tv30 pzem(Serial2);
// #endif
// //---------------------------------------------------------------------------

// Variabel PZEM (FIX ARSITEKTUR UART1 ESP32-S3)
//---------------------------------------------------------------------------
#define PIN_RX_ESP 18  // Terhubung ke TX PZEM
```

© 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

```
#define PIN_TX_ESP 17  // Terhubung ke RX PZEM

HardwareSerial SerialPZEM(1);
PZEM004Tv30 pzem(SerialPZEM, PIN_RX_ESP, PIN_TX_ESP);
//---------------------------------------------------------------------------
//Variabel Button dan Relay
//---------------------------------------------------------------------------
#define pinButton1 42
#define pinButton2 41
#define pinRelay1 8
//---------------------------------------------------------------------------
unsigned long lastDebounceTime1 = 0;
unsigned long lastDebounceTime2 = 0;
const unsigned long debounceDelay = 300; // Jeda buta 300ms anti EMI noise
//---------------------------------------------------------------------------
//Variabel LCD 20x4
//---------------------------------------------------------------------------
LiquidCrystal_I2C lcd(0x27, 20, 4); // I2C address 0x27, 20 column and 4 rows
uint32_t millisAutoScroll = 0;
const int INTERVAL_SCROLL_LCD = 5000; // Auto-scroll layar setiap 5 detik
//---------------------------------------------------------------------------

//Variabel RTC
//---------------------------------------------------------------------------
RTC_DS3231 rtc;
//---------------------------------------------------------------------------

//Variabel RFID
//---------------------------------------------------------------------------
uint8_t success;
uint8_t uid[] = { 0, 0, 0, 0, 0, 0, 0 };  // Buffer to store the returned UID
uint8_t uidLength;
uint32_t versiondata;
//---------------------------------------------------------------------------

// Variabel Parameter Kelistrikan Mesin dll
//---------------------------------------------------------------------------
float voltage1, current1, power1, energy1, frequency1, pf1, va1, VAR1;
float biaya = 0;
float total_energy_Wh = 0;

int year, month, day, hour, minute, second;
int hourUseStart, minuteUseStart, secondUseStart;
int hourUse, minuteUse, secondUse;
uint16_t total_waktu_detik = 0;

bool button1 = 0;
bool k1 = 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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
bool button2 = 0;
bool k2 = 0;
uint8_t menuLCD = 0;
bool kunciRFID = 1;
bool kunciRelay = 0;
uint8_t secondHitung = 0;
uint32_t millisLama = 0;
uint32_t millisLamaPZEM = 0;
//---------------------------------------------------------------------------
// End Variabel Sistem Monitoring
//===========================================================
============================

// Variabel CNN
//===========================================================
============
//---------------------------------------------------------------------------
#define N_RAW_FEAT 7   // tegangan, arus, daya_aktif, daya_reaktif,
daya_apparent, faktor_daya, frekuensi

float rawBuffer[WINDOW_SIZE][N_RAW_FEAT];  // ring buffer nilai mentah
int bufCount = 0;
int bufHead = 0;

int consecutiveAbnormal = 0;
const int CONSECUTIVE_ABNORMAL_TO_ALERT = 3;  // anti false-alarm:
3x berturut baru dianggap anomali
const unsigned long CNN_SAMPLE_INTERVAL_MS = 1000;  // HARUS 1
detik, sesuai asumsi data training
unsigned long lastCnnSample = 0;

// Tensor arena
//---------------------------------------------------------------------------
constexpr int kTensorArenaSize = 45 * 1024;
uint8_t tensor_arena[kTensorArenaSize];

const tflite::Model* model;
tflite::MicroInterpreter* interpreter;
TfLiteTensor* input_tensor;
TfLiteTensor* output_tensor;
//---------------------------------------------------------------------------
// String hasil_cnn = "BUFFERING";  // Default awal sebelum 30 sampel
terkumpul
float prob_normal = 0.00;
float prob_abnormal = 0.00;
String hasil_cnn = "IDLE";

// forward declarations
```

© 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

```
void fillModelInput();
float rollingStd(int t, int colIdx);
void runCnnInference();
//--------------------------------------------------------------------------------
// End Variabel CNN
//================================================================
============================

//MAIN SETUP
//================================================================
============================
void setup() {
  // SETUP MONITORING
  //--------------------------------------------------------------------------------
  pinMode(pinButton1, INPUT_PULLUP);
  pinMode(pinButton2, INPUT_PULLUP);
  pinMode(pinRelay1, OUTPUT);
  digitalWrite(pinRelay1, 0);
  Serial.begin(115200);
  Wire.begin(SDA_PIN, SCL_PIN);

  // KOREKSI KUNCI PZEM: Reset jalur UART1 & buka komunikasi secara
eksplisit
  pinMode(PIN_RX_ESP, INPUT_PULLUP);
  pinMode(PIN_TX_ESP, INPUT_PULLUP);
  delay(100);
  SerialPZEM.begin(9600, SERIAL_8N1, PIN_RX_ESP, PIN_TX_ESP);
  Serial.println("-> Jalur SerialPZEM (Pin 18 & 17) Berhasil Diaktifkan!");

  // Setup RFID
  //--------------------------------------------------------------------------------
  nfc.begin();
  versiondata = nfc.getFirmwareVersion();
  if (! versiondata) {
    Serial.print("Didn't find PN53x board");
    while (1); // halt
  }
  Serial.print("Found chip PN5"); Serial.println((versiondata >> 24) & 0xFF,
HEX);
  Serial.print("Firmware ver. "); Serial.print((versiondata >> 16) & 0xFF, DEC);
  Serial.print('.'); Serial.println((versiondata >> 8) & 0xFF, DEC);
  nfc.SAMConfig();
  // --------------------------------------------------------------------------------

  // SETUP RTC MODULE
  //--------------------------------------------------------------------------------
  if (!rtc.begin()) {
    Serial.println("RTC module is NOT found");
```

© 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

```
  Serial.flush();
  while (true)
    ;
}
// automatically sets the RTC to the date & time on PC this sketch was compiled
// rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
//----------------------------------------------------------------------------

lcd.init(); // initialize the lcd
lcd.backlight();
lcd.clear();
lcd.setCursor(0, 0);            // move cursor the first row
lcd.print("     POLITEKNIK     ");          // print message at the first row
lcd.setCursor(0, 1);            // move cursor to the second row
lcd.print("       NEGERI       "); // print message at the second row
lcd.setCursor(0, 2);            // move cursor to the third row
lcd.print("      JAKARTA       ");          // print message at the third row
lcd.setCursor(0, 3);            // move cursor to the third row
lcd.print("      ( PNJ )       ");          // print message at the third row
delay(3000);
lcd.clear();

lcd.setCursor(0, 0);            // move cursor to the fourth row
lcd.print("   Irfan Mahrubi    ");   // print message the fourth row
lcd.setCursor(0, 1);            // move cursor to the fourth row
lcd.print("   ( 2409511003 )   ");   // print message the fourth row
lcd.setCursor(0, 2);            // move cursor to the fourth row
lcd.print("  MAGISTER TERAPAN  ");   // print message the fourth row
lcd.setCursor(0, 3);            // move cursor to the fourth row
lcd.print("   TEKNIK ELEKTRO   ");   // print message the fourth row
delay(3000);
lcd.clear();

// Koneksi Wifi dan MQTT
//----------------------------------------------------------------------------
WiFi.mode(WIFI_STA);
wifiMulti.addAP("Galaxy Bima Sakti", "cobasaja");
wifiMulti.addAP("RAFA-2G", "luthfi123");
wifiMulti.addAP("Wifi A4", "B3d4h4n*");

Serial.println("ESP32 - Connecting to Wi-Fi");
lcd.setCursor(0, 0);            // move cursor the first row
lcd.print("Menghubungkan");          // print message at the first row
lcd.setCursor(0, 1);            // move cursor the first row
lcd.print("Ke Jaringan ...");          // print message at the first row
delay(2000);
lcd.clear();
```

© 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

```
  while (wifiMulti.run() != WL_CONNECTED) {
    Serial.print(".");
    lcd.setCursor(0, 0); lcd.print("Menghubungkan");
    lcd.setCursor(0, 1); lcd.print("Ke Jaringan ...");
    delay(500);
  }

  lcd.clear();
  lcd.setCursor(0, 0);            // move cursor the first row
  lcd.print("Sukses Terhubung");          // print message at the first row
  lcd.setCursor(0, 1);            // move cursor the first row
  lcd.print("Ke Jaringan");          // print message at the first row
  delay(2000);
  lcd.clear();
  //---------------------------------------------------------------------------

  while (kunciRFID) {
    lcd.setCursor(0, 0); lcd.print("   Tempelkan Kartu  ");
    lcd.setCursor(0, 1); lcd.print("        Untuk       ");
    lcd.setCursor(0, 2); lcd.print(" Mengaktifkan Mesin ");

    success = nfc.readPassiveTargetID(PN532_MIFARE_ISO14443A, uid,
&uidLength);
    if (success) {
      idcard = "";
      for (byte i = 0; i <= uidLength - 1; i++) {
        idcard += (uid[i] < 0x10 ? "0" : "") + String(uid[i], HEX);
      }
      Serial.print("ID Kartu: ");
      Serial.println(idcard);

      bool kartuValid = false; // Penanda validasi

      for (byte i = 0; i < 5; i++) {
        if (idcard == id_allowed[i]) {
          kartuValid = true;
          nama_kartu = "Operator " + String(i + 1);
          Serial.println("ID kartu Diterima");

          lcd.clear();
          lcd.setCursor(0, 1); lcd.print("Nama: " + nama_kartu);
          lcd.setCursor(0, 2); lcd.print("Mesin Diaktifkan....");
          delay(2000);
          lcd.clear();

          kunciRFID = 0; // <--- Kunci HANYA dibuka jika ID cocok!
          break;        // Keluar dari loop pencarian array
        }
```

© 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

```
    }

    // Jika setelah dicek ke semua array ternyata tidak cocok
    if (!kartuValid) {
      Serial.println("Kartu DITOLAK! ID Tidak Dikenal.");
      lcd.clear();
      lcd.setCursor(0, 1); lcd.print("   KARTU DITOLAK!   ");
      lcd.setCursor(0, 2); lcd.print("  ID Tidak Dikenal  ");
      delay(2000);
      lcd.clear();
      // kunciRFID tetap bernilai 1, sistem TETAP TERKUNCI di dalam while
loop
    }
   }
   delay(100); // Mencegah pembacaan sensor berulang terlalu cepat
  }

  DateTime now = rtc.now();
  year = now.year();
  month = now.month();
  day = now.day();
  hour = now.hour();
  minute = now.minute();
  second = now.second();

  connectToMQTT();
  // END SETUP MONITORING
  //---------------------------------------------------------

  // SETUP CNN
  //---------------------------------------------------------
  Serial.println("\n=== CNN Anomaly Detection (Normal/Tidak Normal) - ESP32
===");

  model = tflite::GetModel(g_cnc_model_int8);
  if (model->version() != TFLITE_SCHEMA_VERSION) {
   Serial.println("Schema mismatch");
   while (true) {}
  }

  //---------------------------------------------------------
  static tflite::MicroMutableOpResolver<9> resolver;
  resolver.AddExpandDims();
  resolver.AddConv2D();
  resolver.AddMul();
  resolver.AddAdd();
  resolver.AddReshape();
  resolver.AddMaxPool2D();
```

© 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

```
  resolver.AddMean();
  resolver.AddFullyConnected();
  resolver.AddSoftmax();

  static tflite::MicroInterpreter static_interpreter(
    model, resolver, tensor_arena, kTensorArenaSize
  );
  interpreter = &static_interpreter;

  if (interpreter->AllocateTensors() != kTfLiteOk) {
    Serial.println("Tensor alloc FAILED - coba naikkan kTensorArenaSize");
    while (true) {}
  }

  input_tensor  = interpreter->input(0);
  output_tensor = interpreter->output(0);

  Serial.println("CNN Model siap. Menunggu buffer terisi (butuh 30 sampel ~30
detik)...");
  // END SETUP CNN
  //------------------------------------------------------------
}
// END SETUP
//============================================================
============================


// MAIN LOOP
//============================================================
============================
void loop() {
  // LOOP MONITORING
  //------------------------------------------------------------
  DateTime now = rtc.now();
  year = now.year();
  month = now.month();
  day = now.day();
  hour = now.hour();
  minute = now.minute();
  second = now.second();

  button1 = digitalRead(pinButton1);
  if(button1 == 0 && k1 == 0){
    k1 = 1;
  }
  if(button1 == 1 && k1 == 1){
    menuLCD++;
    if(menuLCD > 3){
```

© Hak Cipta milik Politeknik Negeri Jakarta

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

```
    menuLCD = 0;
  }
  lcd.clear();
  k1 = 0;
}

button2 = digitalRead(pinButton2);
if(button2 == 0 && k2 == 0){
  k2 = 1;
}
if(button2 == 1 && k2 == 1){
  if(kunciRelay == 0){
    kunciRelay = 1;
    digitalWrite(pinRelay1, 1);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Switch On");
    delay(1000);
    lcd.clear();
  }
  else{
    kunciRelay = 0;
    digitalWrite(pinRelay1, 0);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Switch Off");
    delay(1000);
    lcd.clear();
  }
  k2 = 0;
}

if(kunciRelay == 1){
  if(secondUseStart != second){
    secondUse++;

    if(minuteUse > 59){
      hourUse++;
    }
    if(secondUse > 59){
      minuteUse++;
    }
    if(secondUse == 60){
      secondUse = 0;
    }
    if(minuteUse == 60){
      minuteUse = 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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
    if(hourUse == 24){
      hourUse = 0;
    }
    secondUseStart = second;
  }
}

if(millis() - millisLamaPZEM >= PUBLISH_INTERVAL){
  millisLamaPZEM = millis();
  // Serial.print("Custom Address:");
  // Serial.println(pzem.readAddress(), HEX);

  voltage1 = pzem.voltage();
  voltage1 = zeroIfNan(voltage1);
  current1 = pzem.current();
  current1 = zeroIfNan(current1);
  power1 = pzem.power();
  power1 = zeroIfNan(power1);
  energy1 = pzem.energy() / 1000; //kwh
  energy1 = zeroIfNan(energy1);
  frequency1 = pzem.frequency();
  frequency1 = zeroIfNan(frequency1);
  pf1 = pzem.pf();
  pf1 = zeroIfNan(pf1);

  if(voltage1 == 0){
    current1 = 0;
    power1 = 0;
    frequency1 = 0;
    pf1 = 0;
  }

  if (pf1 == 0) {
    va1 = 0;
  } else {
    va1 = power1 / pf1;
  }
  if (pf1 == 0) {
    VAR1 = 0;
  } else {
    VAR1 = power1 / pf1 * sqrt(1-sq(pf1));
  }

  if (voltage1 > 300 || current1 > 20 || power1 > 5000) { // nilai tidak realistis?
    current1 = 0;
    power1 = 0;
    frequency1 = 0;
    pf1 = 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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
  }

  if (isnan(frequency1) || frequency1 < 40.0) frequency1 = 50.0;

  runCnnInference();

  Serial.println("");
  Serial.printf("Voltage        : %.1f\ V\n", voltage1);
  Serial.printf("Current        : %.2f\ A\n", current1);
  Serial.printf("Power Active   : %.2f\ W\n", power1);
  Serial.printf("Frequency      : %.2f\ Hz\n", frequency1);
  Serial.printf("Cosine Phi     : %.2f\ PF\n", pf1);
  Serial.printf("Energy         : %.2f\ kWh\n", energy1);
  Serial.printf("Apparent Power : %.2f\ VA\n", va1);
  Serial.printf("Reactive Power : %.2f\ VAR\n", VAR1);
  Serial.printf("---------- END ----------");
  Serial.println("");
}

  // Loop MQTT
//-------------------------------------------------------------------------------
mqtt.loop();

if (millis() - lastPublishTime > PUBLISH_INTERVAL) {
  lastPublishTime = millis();

  // Perhitungan kWh & Biaya hanya berjalan jika mesin sedang aktif
  if (kunciRelay == 1) {
    total_energy_Wh += power1 * (5.0 / 3600.0);   // Tambah energi dalam Wh
    biaya = (total_energy_Wh / 1000.0) * 1444.70; // Hitung biaya
  }

  sendToMQTT();
}
//-------------------------------------------------------------------------------

switch (menuLCD){
  case 1:
    if(millis() - millisLama >= 500){
      millisLama = millis();
      lcd.setCursor(0,0);
      lcd.print("P Apparent:");
      lcd.setCursor(11, 0);
      lcd.print("         ");
      lcd.setCursor(11, 0);
      lcd.print(va1);
      lcd.print("VA");
      lcd.print(" ");
```

© Hak Cipta milik Politeknik Negeri Jakarta

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

```
    lcd.setCursor(0,1);
    lcd.print("Energi    :");
    lcd.setCursor(11, 1);
    lcd.print("          ");
    lcd.setCursor(11, 1);
    lcd.print(total_energy_Wh);
    lcd.print("Wh");
    lcd.print(" ");
    lcd.setCursor(0,2);
    lcd.print("P Faktor  :");
    lcd.setCursor(11, 2);
    lcd.print("          ");
    lcd.setCursor(11, 2);
    lcd.print(pf1);
    lcd.print(" ");
    lcd.setCursor(0,3);
    lcd.print("Frekuensi :");
    lcd.setCursor(11, 3);
    lcd.print("          ");
    lcd.setCursor(11, 3);
    lcd.print(frequency1);
    lcd.print("Hz");
    lcd.print(" ");
    }
break;
case 2:
  if(millis() - millisLama >= 500){
    millisLama = millis();
    lcd.setCursor(0, 0);  // start to print at the first row
    lcd.print("Date  : ");
    lcd.print(day);
    lcd.print("/");
    lcd.print(month);
    lcd.print("/");
    lcd.print(year);
    lcd.setCursor(0, 1);  // start to print at the second row
    lcd.print("Time  : ");
    lcd.setCursor(8, 1);
    lcd.print("            ");
    lcd.setCursor(8, 1);
    lcd.print(hour);
    lcd.print(":");
    lcd.print(minute);
    lcd.print(":");
    lcd.print(second);
    lcd.setCursor(0, 2);  // start to print at the second row
    lcd.print("T. Use: ");
    lcd.setCursor(8, 2);
```

© Hak Cipta milik Politeknik Negeri Jakarta

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

```
    lcd.print("           ");
    lcd.setCursor(8, 2);
    lcd.print(hourUse);
    lcd.print(":");
    lcd.print(minuteUse);
    lcd.print(":");
    lcd.print(secondUse);
    // lcd.print(total_energy_Wh);
    lcd.setCursor(0, 3);  // start to print at the second row
    lcd.print("Cost  : Rp.");
    lcd.print(biaya);
  }
break;
case 3:
  if (millis() - millisLama >= 500) {
    millisLama = millis();
    char line0[21], line1[21], line2[21], line3[21];

    // Baris 0: Status AI
    snprintf(line0, sizeof(line0), "AI STATUS: %-9s", hasil_cnn.c_str());

    // Baris 1: Probabilitas Normal
    snprintf(line1, sizeof(line1), "P.Normal : %-6.2f", prob_normal);

    // Baris 2: Probabilitas Anomali
    snprintf(line2, sizeof(line2), "P.Anomali: %-6.2f", prob_abnormal);

    // Baris 3: Info Arus (Biar tau alasan override) & Buffer
    // snprintf(line3, sizeof(line3), "I:%.2fA | B:%02d/30", current1, bufCount);

    lcd.setCursor(0, 0); lcd.print(line0);
    lcd.setCursor(0, 1); lcd.print(line1);
    lcd.setCursor(0, 2); lcd.print(line2);
    lcd.setCursor(0, 3); lcd.print(line3);
  }
break;
default:
  if(millis() - millisLama >= 500){
    millisLama = millis();
    lcd.setCursor(0,0);
    lcd.print("Tegangan :");
    lcd.setCursor(10, 0);
    lcd.print("          ");
    lcd.setCursor(10, 0);
    lcd.print(voltage1);
    lcd.print("V");
    lcd.setCursor(0,1);
    lcd.print("Arus     :");
```

© 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

```
      lcd.setCursor(10, 1);
      lcd.print("          ");
      lcd.setCursor(10, 1);
      lcd.print(current1);
      lcd.print("A");
      lcd.setCursor(0,2);
      lcd.print("P Aktif  :");
      lcd.setCursor(10, 2);
      lcd.print("          ");
      lcd.setCursor(10, 2);
      lcd.print(power1);
      lcd.print("W");
      lcd.setCursor(0,3);
      lcd.print("P Reaktif:");
      lcd.setCursor(10, 3);
      lcd.print("          ");
      lcd.setCursor(10, 3);
      lcd.print(VAR1);
      lcd.print("VAR");
    }
   break;
  }
  // END LOOP MONITORING
  //-----------------------------------------------------------------
}
// END LOOP
//=============================================================
============================

// Function
//=============================================================
============================
void printValue(String label, float value) {
  if (value != NAN) {
    Serial.print(label); Serial.println(value);
  } else {
    Serial.println("Error reading");
  }
}
//=============================================================
============================

// Function
//=============================================================
============================
float zeroIfNan(float v) {
  if (isnan(v)) {
    v = 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 mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
  }
  return v;
}
//=============================================================
============================

// Function
//=============================================================
============================
void connectToMQTT() {
  // Connect to the MQTT broker
  mqtt.begin(MQTT_BROKER_ADRRESS, MQTT_PORT, network);
  // Create a handler for incoming messages
  // mqtt.onMessage(messageHandler);
  Serial.print("ESP32 - Connecting to MQTT broker");

  while (!mqtt.connect(MQTT_CLIENT_ID, MQTT_USERNAME,
MQTT_PASSWORD)) {
    Serial.print(".");
    delay(100);
  }
  Serial.println();

  if (!mqtt.connected()) {
    Serial.println("ESP32 - MQTT broker Timeout!");
    return;
  }

  // Subscribe to a topic, the incoming messages are processed by
messageHandler() function
  if (mqtt.subscribe(SUBSCRIBE_TOPIC))
    Serial.print("ESP32 - Subscribed to the topic: ");
  else
    Serial.print("ESP32 - Failed to subscribe to the topic: ");

  Serial.println(SUBSCRIBE_TOPIC);
  Serial.println("ESP32 - MQTT broker Connected!");
}
//=============================================================
============================

// Function
//=============================================================
============================
void sendToMQTT() {
  // Isi data MQTT
  //---------------------------------------------------------------------------
  msgStr = String(voltage1) + "," + String(current1) + "," + String(power1) + "," +
```

© 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

```
String(VAR1) + "," +
  String(va1) + "," + String(total_energy_Wh) + "," + String(pf1) + "," +
String(frequency1) + "," +
  String(biaya) + "," + String(nama_kartu) + "," + hasil_cnn + ",";
  //-------------------------------------------------------------------------------

  byte arrSize = msgStr.length() + 1;
  char msg[arrSize];
  Serial.print("PUBLISH DATA: ");
  Serial.println(msgStr);
  msgStr.toCharArray(msg, arrSize);
  mqtt.publish(PUBLISH_TOPIC, msg);
  msgStr = "";
}
//============================================================
============================

// Function
//============================================================
============================
void messageHandler(String &topic, String &payload) {
  Serial.println("ESP32 - received from MQTT:");
  Serial.println("- topic: " + topic);
  Serial.println("- payload:");
  Serial.println(payload);
}
//============================================================
============================

// ============================== FUNGSI CNN
=========================================
// rolling std 5 sampel terakhir untuk kolom raw tertentu (colIdx), posisi window
ke-t
float rollingStd(int t, int colIdx) {
  int n = min(t + 1, 5);
  float sum = 0, sumsq = 0;
  for (int k = 0; k < n; k++) {
    int idx = (bufHead + t - k + WINDOW_SIZE) % WINDOW_SIZE;
    float v = rawBuffer[idx][colIdx];
    sum += v; sumsq += v * v;
  }
  float m = sum / n;
  float var = (n > 1) ? (sumsq / n - m * m) : 0.0f;
  if (var < 0) var = 0;
  return sqrt(var);
}

void fillModelInput() {
```

© Hak Cipta milik Politeknik Negeri Jakarta

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

```
  for (int t = 0; t < WINDOW_SIZE; t++) {
    int idx = (bufHead + t) % WINDOW_SIZE;
    int idx_prev = (bufHead + t - 1 + WINDOW_SIZE) % WINDOW_SIZE;
    bool has_prev = (t > 0);

    float arus_v          = rawBuffer[idx][1];
    float daya_aktif_v    = rawBuffer[idx][2];
    float faktor_daya_v   = rawBuffer[idx][5];

    float arus_delta      = has_prev ? (arus_v - rawBuffer[idx_prev][1]) : 0.0f;
    float daya_aktif_delta  = has_prev ? (daya_aktif_v - rawBuffer[idx_prev][2]) :
0.0f;
    float faktor_daya_delta = has_prev ? (faktor_daya_v - rawBuffer[idx_prev][5])
: 0.0f;

    float arus_rstd        = rollingStd(t, 1);
    float daya_aktif_rstd  = rollingStd(t, 2);
    float faktor_daya_rstd  = rollingStd(t, 5);

    float feat[N_FEATURES] = {
    rawBuffer[idx][0], arus_v, daya_aktif_v, rawBuffer[idx][3], rawBuffer[idx][4],
    faktor_daya_v, 50.0,
    arus_delta, daya_aktif_delta, faktor_daya_delta,
    arus_rstd, daya_aktif_rstd, faktor_daya_rstd
    };
    for (int f = 0; f < N_FEATURES; f++) {
      float normed = (feat[f] - FEATURE_MEAN[f]) / FEATURE_STD[f];
      int32_t q = (int32_t)round(normed / INPUT_SCALE) +
INPUT_ZERO_POINT;
      if (q > 127) q = 127;
      if (q < -128) q = -128;
      input_tensor->data.int8[t * N_FEATURES + f] = (int8_t)q;

      // --- TAMBAHAN DEBUG ---
      // Kita intip data pada baris pertama (t=0) saja biar Serial Monitor tidak terlalu
penuh
      if (t == 0) {
        Serial.printf("[DEBUG INPUT] Feat[%d] raw: %.2f | norm: %.2f | q_int8:
%d\n", f, feat[f], normed, q);
      }
      // ----------------------
    }

    // for (int f = 0; f < N_FEATURES; f++) {
    //   float normed = (feat[f] - FEATURE_MEAN[f]) / FEATURE_STD[f];
    //   int32_t q = (int32_t)round(normed / INPUT_SCALE) +
INPUT_ZERO_POINT;
    //   if (q > 127) q = 127;
```

© 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

```
    //   if (q < -128) q = -128;
    //   input_tensor->data.int8[t * N_FEATURES + f] = (int8_t)q;
    // }
  }
}

const float IDLE_CURRENT_THRESHOLD = 0.03f;  // Ampere
bool wasIdle = true;  // asumsi awal saat boot: idle/kalibrasi (mesin belum
motong)

void runCnnInference() {
  // --- 1. GERBANG STATUS MESIN & IDLE ---
  // Buffering HANYA berjalan jika mesin sedang ON (kunciRelay == 1) dan ada
arus mengalir
  if (kunciRelay == 0 || current1 < IDLE_CURRENT_THRESHOLD) {
    bufCount = 0;
    bufHead = 0;
    consecutiveAbnormal = 0; // Reset hitungan beruntun jika mesin mati/idle
    if (hasil_cnn != "ANOMALI") {
      hasil_cnn = "IDLE";
    }
    prob_normal = 0.00;
    prob_abnormal = 0.00;
    return;
  }

  // --- 2. ISI BUFFER (Proses buffering berjalan otomatis karena lolos filter di
atas) ---
  int writeIdx = (bufCount < WINDOW_SIZE) ? bufCount : bufHead;
  if (bufCount < WINDOW_SIZE) bufCount++;
  else bufHead = (bufHead + 1) % WINDOW_SIZE;

  rawBuffer[writeIdx][0] = voltage1;
  rawBuffer[writeIdx][1] = current1;
  rawBuffer[writeIdx][2] = power1;
  rawBuffer[writeIdx][3] = VAR1;
  rawBuffer[writeIdx][4] = va1;
  rawBuffer[writeIdx][5] = pf1;
  rawBuffer[writeIdx][6] = frequency1;

  if (bufCount < WINDOW_SIZE) {
    hasil_cnn = "BUFFERING";
    return;
  }

  // --- 3. JALANKAN CNN ---
  fillModelInput();
  if (interpreter->Invoke() != kTfLiteOk) {
```

© 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

```
    Serial.println("Inference TFLite Gagal!");
    return;
  }

  float p_anom_raw = (output_tensor->data.int8[0] - OUTPUT_ZERO_POINT) *
OUTPUT_SCALE;
  float p_norm_raw = (output_tensor->data.int8[1] - OUTPUT_ZERO_POINT) *
OUTPUT_SCALE;

  if (current1 < 0.19f) {
    hasil_cnn = "NORMAL";
    prob_normal = 1.00;
    prob_abnormal = 0.00;
    consecutiveAbnormal = 0;
  } else {
    prob_normal = p_norm_raw;
    prob_abnormal = p_anom_raw;

    if (prob_abnormal > prob_normal) {
      consecutiveAbnormal++;
      if (consecutiveAbnormal >= CONSECUTIVE_ABNORMAL_TO_ALERT) {
        hasil_cnn = "ANOMALI";

        // === AKTIVASI PROTEKSI: SSR LANGSUNG CUT TEGANGAN ===
        kunciRelay = 0;             // Ubah status logika sistem menjadi OFF
        digitalWrite(pinRelay1, 0); // Putus tegangan fisik pada pin Relay/SSR

        Serial.println("\n[DANGER] ANOMALI KONSISTEN! SSR DI-CUT
SEGERA!");
      } else {
        hasil_cnn = "WARNING"; // Status transisi menuju alert
      }
    } else {
      hasil_cnn = "NORMAL";
      consecutiveAbnormal = 0; // Reset jika sempat anomali tapi kembali normal
    }
  }
}

//   // Debug Serial tetap bisa liat aslinya kalau mau
//   Serial.printf("[CNN] Result: %s | AI_Norm: %.2f | Cur: %.2fA\n",
hasil_cnn.c_str(), p_norm_raw, current1);
// }
// ============================== END FUNGSI CNN
=========================================
```