

## KLASIFIKASI INTENSITAS HUJAN PER JAM MENGGUNAKAN 1D-CNN BERBASIS TINYML DENGAN KALIBRASI AMBANG PRECISION-RECALL

Revangga Kusuma Dhani<sup>1\*</sup>; Agustina Rachmawardani<sup>2</sup>; Marzuki Sinambela<sup>3</sup>; Adi Widiatmoko Wastumirad<sup>4</sup>

Instrumentasi-MKG<sup>1,2,3,4</sup>

Sekolah Tinggi Meteorologi Klimatologi dan Geofisika, Kota Tangerang, Indonesia<sup>1,2,3,4</sup>

[www.stmkg.ac.id](http://www.stmkg.ac.id)<sup>1,2,3,4</sup>

[refanggadali@gmail.com](mailto:refanggadali@gmail.com)<sup>1\*</sup>, [agustina.rahmawardani@stmkg.ac.id](mailto:agustina.rahmawardani@stmkg.ac.id)<sup>2</sup>, [sinambela.m@gmail.com](mailto:sinambela.m@gmail.com)<sup>3</sup>, [widiatmokoadi@gmail.com](mailto:widiatmokoadi@gmail.com)<sup>4</sup>

(\*) Corresponding Author



Ciptaan disebarluaskan di bawah Lisensi Creative Commons Atribusi-NonKomersial 4.0 Internasional.

**Abstract**—Hydrometeorological disasters dominate annual disaster occurrences in Indonesia, yet local rainfall prediction remains challenging due to atmospheric complexity and limited resolution of numerical weather models. Server-based forecasting systems further depend on high-performance computing infrastructure and stable network connectivity that are not always available in the field. This study develops an hourly rainfall intensity classification model based on One-Dimensional Convolutional Neural Network (1D-CNN) deployable on an ESP32-S3 microcontroller as a proof-of-concept inference component for rainfall early warning systems. The model uses nine meteorological features arranged in an 18×9 sliding window derived from observational data from the BMKG Soekarno-Hatta Meteorological Station AWS from 2018 to 2025. Logarithmic class weighting and Precision-Recall curve threshold calibration were applied to address extreme class imbalance in hourly resolution data. The model was compared against five baseline models under identical configurations. Threshold calibration increased K2 recall from 0.037 to 0.236 and improved Macro-F1 from 0.461 to 0.530, outperforming all baseline models in terms of Macro-F1. Post-training quantization INT8 reduced model size from 185.8 KB to 64.8 KB with 99.02% decision agreement against the Float32 model. On-device inference on ESP32-S3 achieved a total latency of 13.873 ms with 36.5 KB tensor arena and 234.7 KB free heap, confirming real-time operation without dependence on servers or internet connectivity.

**Keywords:** 1D-CNN, Precision-Recall Threshold Calibration, Rainfall Intensity Classification, TinyML.

**Abstrak**—Bencana hidrometeorologi mendominasi kejadian bencana tahunan di Indonesia, tetapi prakiraan hujan lokal masih menghadapi tantangan akibat kompleksitas atmosfer dan keterbatasan resolusi model cuaca numerik. Sistem prakiraan berbasis server juga bergantung pada infrastruktur komputasi berkinerja tinggi dan konektivitas jaringan yang tidak selalu tersedia di lapangan. Penelitian ini mengembangkan model klasifikasi intensitas hujan per jam berbasis One-Dimensional Convolutional Neural Network (1D-CNN) yang dapat dijalankan secara *on-device* pada mikrokontroler ESP32-S3 sebagai prototipe komponen inferensi untuk sistem peringatan dini hujan yang beroperasi secara mandiri. Model menggunakan sembilan fitur meteorologi dalam jendela geser 18×9 dari data observasi AWS BMKG Stasiun Meteorologi Soekarno-Hatta periode 2018 hingga 2025. Pembobotan kelas berskala logaritmik dan kalibrasi ambang berbasis kurva *Precision-Recall* diterapkan untuk menangani ketimpangan kelas yang ekstrem pada data resolusi per jam. Model dibandingkan dengan lima model pembanding pada konfigurasi identik. Kalibrasi ambang meningkatkan *recall* kelas K2 dari 0,037 menjadi 0,236 dan Macro-F1 dari 0,461 menjadi 0,530, melampaui seluruh model pembanding berdasarkan Macro-F1. *Post-training quantization* INT8 mereduksi ukuran model dari 185,8 KB menjadi 64,8 KB dengan kesesuaian keputusan 99,02% terhadap model Float32. Inferensi *on-device* pada ESP32-S3 mencapai latensi total 13,873 ms dengan *tensor arena* 36,5 KB dan *heap* bebas 234,7 KB, membuktikan model dapat beroperasi secara *real-time* tanpa ketergantungan pada server atau konektivitas internet.

**Kata kunci:** 1D-CNN, Kalibrasi Ambang Batas Precision-Recall, Klasifikasi Intensitas Curah Hujan, TinyML.

## PENDAHULUAN

Bencana hidrometeorologi masih menjadi ancaman utama di Indonesia dan mendominasi kejadian bencana setiap tahunnya. Pada tahun 2023, Badan Nasional Penanggulangan Bencana (BNPB) mencatat lebih dari 4.900 kejadian bencana, dengan sebagian besar berupa banjir, tanah longsor, dan cuaca ekstrem yang mengakibatkan 267 korban meninggal serta lebih dari sembilan juta jiwa terdampak (Badan Nasional Penanggulangan Bencana, 2024). Kompleksitas pola curah hujan di Indonesia dipengaruhi oleh interaksi laut-darat, topografi, proses konveksi, dan variabilitas iklim, sehingga karakteristik hujan menjadi sangat kompleks (Hidayat dkk., 2025). Karakteristik tersebut menyebabkan hujan konvektif lokal dan ekstrem masih sulit diprakirakan secara akurat oleh sistem prakiraan cuaca, sehingga diperlukan pendekatan yang mampu meningkatkan kemampuan prediksi jangka pendek (Zhang dkk., 2023).

Sistem peringatan dini cuaca di Indonesia diselenggarakan BMKG melalui produk *nowcasting* berbasis stasiun cuaca, radar, dan satelit. Namun, model prakiraan cuaca numerik masih memiliki keterbatasan dalam merepresentasikan hujan lokal dan ekstrem akibat kompleksitas atmosfer serta keterbatasan resolusi spasial-temporal (Glawion dkk., 2025). Distribusi stasiun hujan dan cakupan radar cuaca yang belum merata masih menyisakan celah pemantauan di berbagai wilayah Indonesia (Putra dkk., 2024). Di sisi lain, sistem prakiraan berbasis *server* bergantung pada infrastruktur komputasi berkinerja tinggi dan konektivitas jaringan yang tidak selalu tersedia atau stabil di lapangan (Bauer, 2024; Zaidi dkk., 2022). Beberapa keterbatasan ini bersama-sama mengurangi kemampuan peringatan dini yang responsif, terutama di wilayah dengan cakupan infrastruktur yang terbatas.

Beberapa penelitian terdahulu telah mengkaji klasifikasi intensitas hujan menggunakan pendekatan *machine learning*, namun dengan kondisi eksperimen yang berbeda. Widipturo (2022) mengklasifikasikan curah hujan harian menggunakan 1D-CNN pada data AWS BMKG Stasiun Soekarno-Hatta dan melaporkan F1 kelas hujan hanya 0,091, mencerminkan tantangan deteksi kelas minoritas yang belum terpecahkan pada resolusi harian. Mdegela dkk. (2023) dan Yu dkk. (2026) mengkaji klasifikasi hujan ekstrem dengan pendekatan optimasi *recall* dan augmentasi fisik, namun seluruh inferensi dilakukan di *server* tanpa pertimbangan *deployment* pada perangkat terbatas, dan ukuran model tidak dilaporkan. Dari sisi *deployment* TinyML pada perangkat tertanam,

Taheri Moghadar dkk. (2025) membuktikan kelayakan 1D-CNN berbasis TinyML untuk klasifikasi deret waktu secara *real-time* pada Arduino Nano 33 IoT dengan latensi 275 ms dan penggunaan SRAM 26 KB. Kebaruan penelitian ini terletak pada integrasi 1D-CNN, pembobotan kelas logaritmik, kalibrasi ambang berbasis kurva *Precision-Recall*, dan kuantisasi INT8 dalam satu sistem klasifikasi intensitas hujan per jam yang dapat beroperasi secara mandiri pada mikrokontroler ESP32-S3, kombinasi yang belum ditemukan pada penelitian sebelumnya.

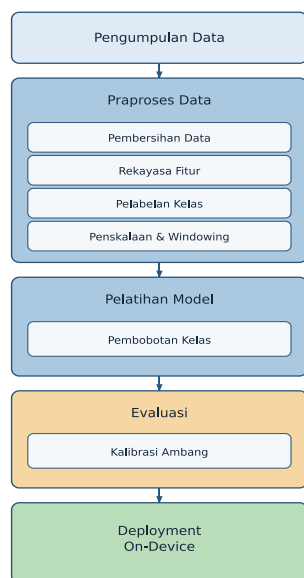
Untuk mengatasi kendala infrastruktur tersebut, sistem perlu memproses data secara mandiri di titik pengamatan. Pendekatan *machine learning* terbukti efektif untuk klasifikasi dan estimasi curah hujan sebagai dasar sistem peringatan dini banjir (Mdegela dkk., 2023; Rachmawardani dkk., 2022) dan estimasi curah hujan resolusi tinggi di Indonesia (Putra dkk., 2024). Namun, studi-studi tersebut umumnya menjalankan model di *server* sehingga bergantung pada konektivitas internet dan belum menjawab kebutuhan inferensi mandiri di lapangan.

Pendekatan *Tiny Machine Learning* (TinyML) memungkinkan inferensi model *machine learning* dilakukan langsung di perangkat mikrokontroler tanpa ketergantungan pada *server* atau internet (Esposito dkk., 2022). Dari berbagai arsitektur yang kompatibel dengan perangkat tertanam, One-Dimensional Convolutional Neural Network (1D-CNN) memiliki keunggulan dalam menangkap pola temporal lokal pada data deret waktu multivariat dengan beban komputasi rendah (Rozzy dkk., 2024; Ullah dkk., 2024), dan telah terbukti dapat dijalankan secara *real-time* pada mikrokontroler seperti ESP32 (Gupta & Shivhare, 2025; Taheri Moghadar dkk., 2025). Dalam sistem peringatan dini hujan, data resolusi per jam diperlukan agar perubahan intensitas hujan dapat dideteksi lebih cepat. Namun, penggunaan resolusi tersebut menyebabkan ketidakseimbangan kelas yang ekstrem karena kejadian hujan berintensitas tinggi relatif jarang dibandingkan kondisi tidak hujan. Kondisi ini membuat model cenderung bias terhadap kelas mayoritas sehingga kemampuan mendeteksi kelas hujan minoritas menurun (Farhadpour dkk., 2024). Pada kasus seperti ini, kurva *Precision-Recall* lebih informatif dibandingkan kurva ROC dalam mengevaluasi kinerja deteksi kelas minoritas dan dapat digunakan untuk menentukan ambang yang menyeimbangkan *precision* dan *recall* (Rajaraman dkk., 2022). Oleh karena itu, penelitian ini menerapkan kalibrasi ambang berbasis *Precision-Recall* sebagai strategi untuk meningkatkan kemampuan deteksi kelas hujan minoritas.

Berdasarkan permasalahan tersebut, penelitian ini bertujuan mengembangkan model klasifikasi intensitas hujan per jam berbasis One-Dimensional Convolutional Neural Network (1D-CNN) yang dapat dijalankan secara *on-device* pada mikrokontroler ESP32-S3 sebagai prototipe komponen inferensi dalam sistem peringatan dini hujan. Kebaruan penelitian ini bukan terletak pada komponen metode secara individual, melainkan pada integrasi 1D-CNN, pembobotan kelas logaritmik, kalibrasi ambang berbasis kurva *Precision-Recall*, dan kuantisasi INT8 dalam satu sistem klasifikasi intensitas hujan per jam berbasis TinyML. Kontribusi spesifiknya meliputi penerapan kalibrasi ambang berbasis kurva *Precision-Recall* untuk meningkatkan deteksi kelas hujan minoritas pada data resolusi per jam yang sangat tidak seimbang, serta pembuktian kelayakan inferensi model 1D-CNN pada ESP32-S3 secara mandiri tanpa ketergantungan pada *server*.

## BAHAN DAN METODE

Gambar 1 menunjukkan alur penelitian secara keseluruhan. Penelitian ini terdiri dari delapan tahap berurutan, yaitu pengumpulan data, praproses data, rekayasa fitur, pelabelan kelas, penskalaan dan *windowing*, pelatihan model, evaluasi, dan *deployment on-device*.



Sumber: (Hasil Penelitian, 2026)

Gambar 1. Diagram Alur Penelitian

### Data dan Praproses

Penelitian ini menggunakan data observasi meteorologi permukaan dari Automatic Weather Station (AWS) BMKG di Bandara Soekarno-Hatta

(kode stasiun STA5082) periode 2018 hingga 2025. Variabel yang digunakan meliputi curah hujan (rr), suhu udara (T), kelembapan relatif (RH), dan tekanan udara (P). Sebagian data tercatat pada resolusi satu menit dan sebagian pada resolusi sepuluh menit, sehingga keduanya diseragamkan ke interval sepuluh menit melalui agregasi sebelum digunakan. Data berasal dari satu stasiun dengan karakteristik iklim perkotaan pesisir, sehingga generalisasi model ke stasiun dengan topografi atau pola hujan yang berbeda perlu diverifikasi lebih lanjut.

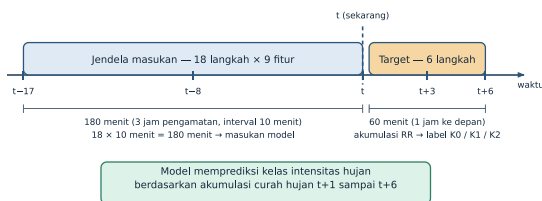
Tahap pembersihan data dilakukan dengan memeriksa kesesuaian nilai terhadap rentang yang ditetapkan berdasarkan karakteristik klimatologis data di lokasi penelitian, sejalan dengan World Meteorological Organization (2024) yang menyatakan bahwa batas rentang variabel meteorologi bergantung pada kondisi klimatologis setempat. Rentang pemeriksaan yang digunakan meliputi suhu 20–36 °C, kelembapan relatif 30–100%, tekanan 996–1020 hPa, dan curah hujan 0–50 mm per sepuluh menit. Dari total 396.404 baris data gabungan, sebanyak 34.317 baris (8,7%) dibuang karena nilai di luar rentang fisik, sehingga tersisa 362.087 baris yang valid. *Missing value* pada suhu, kelembapan, dan tekanan diisi dengan interpolasi linear (Rizal dkk., 2022), sedangkan *missing value* curah hujan tidak diimputasi karena sifatnya yang diskret.

Setelah pembersihan, dilakukan rekayasa fitur untuk memperkaya representasi kondisi atmosfer. Titik embun (Td) dihitung menggunakan pendekatan Magnus-Tetens pada persamaan (1) dengan konstanta  $a = 17,27$  dan  $b = 237,7$  °C, kemudian dew point depression (DPD) diperoleh sebagai selisih  $T - T_d$  yang mencerminkan tingkat kejenuhan atmosfer dan potensi pembentukan hujan. Tendensi tekanan jangka pendek (dP\_short), 30 menit, dan jangka menengah (dP\_mid), 60 menit, dihitung sebagai laju perubahan tekanan udara yang berkaitan dengan kedatangan sistem cuaca. Pasangan sinus-kosinus jam (hour\_sin dan hour\_cos) ditambahkan untuk merepresentasikan pola diurnal hujan tanpa diskontinuitas pada pergantian hari.

$$T_d = \frac{b\gamma}{a-\gamma}, \gamma = \ln\left(\frac{RH}{100}\right) + \frac{aT}{b+T} \quad (1)$$

Total fitur masukan berjumlah sembilan dan disusun ke dalam jendela waktu 18 langkah sepuluh menit, sehingga setiap sampel berbentuk matriks  $18 \times 9$  yang merepresentasikan tiga jam pengamatan untuk memprediksi kondisi satu jam ke depan. Model menerima masukan berupa jendela

deret waktu sepanjang 18 langkah dari  $t-17$  hingga  $t$ , setara dengan pengamatan tiga jam pada interval 10 menit. Label kelas ditentukan dari akumulasi curah hujan pada enam langkah berikutnya ( $t+1$  hingga  $t+6$ ), yang merepresentasikan satu jam ke depan. Dengan demikian, setiap prediksi model mencerminkan intensitas hujan yang diperkirakan akan terjadi satu jam setelah waktu pengamatan terakhir. Struktur jendela waktu ini ditunjukkan pada Gambar 2.



Sumber: (Hasil Penelitian, 2026)

Gambar 2. Struktur Jendela Waktu Masukan dan Target Prediksi

Setiap sampel diberi label berdasarkan akumulasi curah hujan satu jam ke depan menggunakan tiga kelas yang diadaptasi dari kategori BMKG (Akbar & Sanjaya, 2023). Kategori Hujan Sedang hingga Sangat Lebat digabung menjadi K2 karena frekuensinya yang sangat rendah pada resolusi per jam. Skema pelabelan ditunjukkan pada Tabel 1.

Tabel 1. Klasifikasi Intensitas Hujan dan Kelas yang Dirancang

| Intensitas (mm/jam) | Kategori                    | Kelas |
|---------------------|-----------------------------|-------|
| <1                  | Tidak Hujan                 | K0    |
| 1 - < 5             | Hujan Ringan                | K1    |
| ≥ 5                 | Hujan Sedang - Sangat Lebat | K2    |

Sumber: (Akbar & Sanjaya, 2023)

Seluruh fitur dinormalisasi menggunakan Min-Max Normalization dengan parameter yang dihitung dari data latih saja untuk menghindari kebocoran informasi (Shantal dkk., 2023), mengikuti persamaan (2).

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \quad (2)$$

Dataset dibagi secara berurutan menjadi tiga himpunan tanpa pengacakan untuk mempertahankan struktur temporal data. Himpunan latih mencakup periode 2018 hingga Juni 2023, himpunan validasi mencakup tahun 2024, dan himpunan uji mencakup Januari hingga November 2025. Periode Juli hingga Desember 2023 dikecualikan karena file Juli 2023 kosong dan

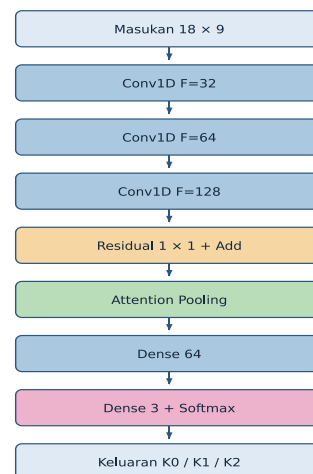
proporsi data hilang pada semester kedua 2023 mencapai 30,7% untuk curah hujan dan 44% untuk suhu dan tekanan, jauh melampaui proporsi tahun lain yang hanya 0,1% hingga 7%. *Windowing* dilakukan secara terpisah per partisi tahun setelah pembagian temporal, sehingga tidak ada jendela yang melewati batas antar-split. Jendela di awal setiap partisi yang tidak memiliki 18 langkah *lookback* lengkap dalam partisi yang sama dibuang, demikian pula jendela yang mengandung nilai hilang pada curah hujan di periode target prediksi karena label kelas dihitung dari akumulasi curah hujan satu jam ke depan. Setelah *windowing*, sebanyak 5.787 *timestep* (1,6%) tidak dapat membentuk jendela lengkap sehingga dibuang, menghasilkan 356.300 jendela final. Distribusi sampel per kelas ditunjukkan pada Tabel 2. Kelas K0 mendominasi seluruh himpunan dengan proporsi di atas 96%, sementara K2 hanya mencakup sekitar 1,47% pada data uji, yang mencerminkan tingkat ketimpangan kelas yang ekstrem.

Tabel 2. Distribusi Sampel Per Kelas pada Himpunan Data

| Split    | Periode      | Jumlah  | K0      | K1    | K2    |
|----------|--------------|---------|---------|-------|-------|
| Latih    | 2018-2023    | 258.195 | 249.631 | 5.886 | 2.678 |
| Validasi | 2024         | 50.720  | 49.212  | 923   | 585   |
| Uji      | Jan-Nov 2025 | 47.385  | 45.627  | 1.063 | 695   |

Sumber: (Hasil Penelitian, 2026)

### Arsitektur Model 1D-CNN



Sumber: (Hasil Penelitian, 2026)

Gambar 3. Arsitektur Model 1D-CNN

Arsitektur model ditunjukkan pada Gambar 3 dan konfigurasi lengkapnya pada Tabel 3. Masukan berupa matriks  $18 \times 9$  diproses melalui tiga blok Conv1D dengan filter yang meningkat bertahap (32,

64, 128), masing-masing dilengkapi Batch Normalization, ReLU, dan Dropout 0,2. Untuk mempertahankan kelancaran aliran gradien pada jaringan, setiap blok konvolusi menerapkan koneksi residual dengan proyeksi  $1 \times 1$  (Chi dkk., 2025). Keluaran blok konvolusi terakhir diagregasi menggunakan *attention pooling* yang memberi bobot lebih pada bagian deret waktu yang paling relevan terhadap prediksi hujan, sebelum diteruskan ke lapisan Dense dan keluaran Softmax tiga kelas.

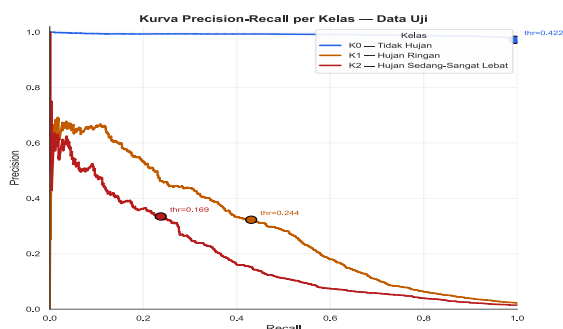
Model dilatih menggunakan categorical cross-entropy dengan label smoothing 0,1 dan dioptimasi dengan Adam pada laju pembelajaran 0,001. Macro-F1 digunakan sebagai metrik pemantauan karena lebih adil terhadap kelas minoritas dibandingkan akurasi (Farhadpour dkk., 2024). Total parameter model berjumlah 45.860.

Tabel 3. Konfigurasi Model 1D-CNN

| Komponen               | Spesifikasi                                             |
|------------------------|---------------------------------------------------------|
| Bentuk Masukan         | $18 \times 9$                                           |
| Blok Konvolusi         | 3 blok Conv1D, filter 32/64/128, kernel 3, padding same |
| Normalisasi & Aktivasi | Batch Normalization, ReLU, Dropout 0,2 per blok         |
| Koneksi Residual       | Proyeksi $1 \times 1$                                   |
| Agregasi               | <i>Attention pooling</i>                                |
| Lapisan Dense          | 64 unit, Batch Normalization, ReLU, Dropout 0,2         |
| Keluaran               | Dense 3 unit, Softmax                                   |
| Fungsi Galat           | Categorical cross-entropy, label smoothing 0,1          |
| Optimizer              | Adam, lr 0,001, clipnorm 1,0                            |
| Jumlah Parameter       | 45.860                                                  |

Sumber: (Hasil Penelitian, 2026)

### Penanganan Ketidakseimbangan Kelas



Sumber: (Hasil Penelitian, 2026)

Gambar 4. Kurva *Precision-Recall* per Kelas Beserta Titik Operasi pada Ambang Hasil Kalibrasi

Ketimpangan distribusi kelas yang ekstrem ditangani melalui dua mekanisme yang diterapkan pada tahap berbeda. Pada tahap pelatihan, fungsi galat diberi bobot kelas berskala logaritmik agar kelas minoritas mendapat perhatian lebih tanpa

menyebabkan ketidakstabilan pelatihan. Pada tahap inferensi, keputusan kelas tidak ditentukan dengan memilih probabilitas tertinggi, melainkan menggunakan ambang per kelas yang dikalibrasi dari kurva *Precision-Recall* pada data validasi

### Algoritma 1. Pengambilan Keputusan Berbasis Ambang Prioritas

|        |                                                                                   |
|--------|-----------------------------------------------------------------------------------|
| Input  | : $p[0], p[1], p[2] \rightarrow$ probabilitas keluaran model untuk K0, K1, K2     |
|        | THR_K0, THR_K1, THR_K2 $\rightarrow$ ambang hasil kalibrasi PR pada data validasi |
| Output | : kelas prediksi {0, 1, 2}                                                        |
| 1:     | IF $p[2] \geq \text{THR\_K2}$ THEN                                                |
| 2:     | RETURN 2 {K2—Hujan Sedang-Sangat Lebat}                                           |
| 3:     | ELSE IF $p[1] \geq \text{THR\_K1}$ THEN                                           |
| 4:     | RETURN 1 {K1—Hujan Ringan}                                                        |
| 5:     | ELSE                                                                              |
| 6:     | RETURN 0 {K0—Tidak Hujan, default}                                                |
| 7:     | END IF                                                                            |

Sumber: (Hasil Penelitian, 2026)

Kalibrasi ambang dilakukan dengan memilih nilai ambang yang mengoptimalkan F1 per kelas pada data validasi menggunakan kurva *Precision-Recall*. Ambang yang ditetapkan adalah 0,4222 untuk K0, 0,2439 untuk K1, dan 0,1691 untuk K2. Ambang K2 yang paling rendah membuat model lebih peka terhadap kejadian hujan yang paling berbahaya. Pendekatan serupa telah diterapkan pada klasifikasi data medis yang juga timpang (Kong dkk., 2025). Kurva *Precision-Recall* beserta titik operasi pada ambang hasil kalibrasi ditunjukkan pada Gambar 4. Mekanisme pengambilan keputusan pada tahap inferensi menggunakan aturan prioritas yang memeriksa probabilitas absolut setiap kelas terhadap ambang individualnya secara berurutan dari kewaspadaan tertinggi ke terendah, seperti ditunjukkan pada Algoritma 1. Mekanisme ini bersifat konservatif, K2 selalu diperiksa pertama terlepas dari besar probabilitasnya. Jika lebih dari satu kelas melewati ambangnya secara bersamaan, kelas dengan prioritas tertinggi dipilih sehingga K2 selalu diutamakan atas K1 maupun K0. Sebagai contoh, jika  $p(K2) = 0,17$  (melewati  $\text{THR\_K2} = 0,1691$ ) dan  $p(K1) = 0,80$ , keputusan tetap K2. Desain ini didasarkan pada asimetri risiko sistem peringatan dini, di mana kegagalan mendeteksi hujan berbahaya berakibat lebih fatal dibandingkan alarm yang sesekali keliru.

### Evaluasi Model dan Deployment

Model 1D-CNN sebagai model utama dikuantisasi ke format INT8 melalui *post-training quantization* (Liu dkk., 2023). Model tersebut kemudian ditanam pada mikrokontroler ESP32-S3 menggunakan TensorFlow Lite for

Microcontrollers, sebuah kerangka kerja yang dirancang untuk menjalankan inferensi *machine learning* pada mikrokontroler dengan sumber daya terbatas (Zaidi dkk., 2022). Untuk memverifikasi bahwa pemilihan arsitektur 1D-CNN memberikan keseimbangan terbaik antara kinerja klasifikasi dan kelayakan *deployment*, model ini dibandingkan dengan lima model pembandingan yaitu Logistic Regression, Random Forest, MLP, LSTM, dan GRU pada konfigurasi identik. Konfigurasi terbaik setiap model diperoleh melalui pencarian grid dengan Macro-F1 validasi sebagai kriteria pemilihan. Konfigurasi final seluruh model pembandingan disajikan pada Tabel 4.

Tabel 4. Konfigurasi Model Pembandingan

| Model | Konfigurasi Utama                                           | Penanganan Kelas       |
|-------|-------------------------------------------------------------|------------------------|
| LR    | C=10, solver=lbfgs, max_iter=1000                           | class_weight=balanced  |
| RF    | n_estimators=100, max_depth=20                              | class_weight=balanced  |
| MLP   | hidden_layer_sizes=(256, 128), max_iter=300, early_stopping | sample_weight=balanced |
| LSTM  | 2×LSTM(64), Dropout 0,2, Dense 64, unroll=True              | log-scale class_weight |
| GRU   | 2×GRU(64), Dropout 0,2, Dense 64, unroll=True               | log-scale class_weight |

Sumber: (Hasil Penelitian, 2026)

Masukan untuk LR, RF, dan MLP berupa vektor datar 162 dimensi hasil flatten matriks 18×9, sedangkan masukan LSTM dan GRU berupa sekuens 3D (18×9). LSTM dan GRU dilatih dengan Adam lr=0,001, categorical cross-entropy dengan label smoothing 0,1, maksimal 80 epoch dengan early stopping patience=10, dan unroll=True diterapkan untuk kompatibilitas konversi TFLite. Seluruh model dievaluasi menggunakan data, fitur, pembagian waktu kronologis, dan prosedur kalibrasi ambang yang sama. Mekanisme penanganan ketimpangan kelas disesuaikan dengan karakteristik masing-masing model, Logistic Regression dan Random Forest menggunakan *class\_weight=balanced*, sedangkan model berbasis *neural network* menggunakan pembobotan kelas berskala logaritmik.

Evaluasi kinerja klasifikasi menggunakan *precision*, *recall*, F1 per kelas, dan Macro-F1 yang didefinisikan pada persamaan (3) hingga (6).

$$Precision = \frac{TP}{TP+FP} \quad (3)$$

$$Recall = \frac{TP}{TP+FN} \quad (4)$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (5)$$

$$Macro-F1 = \frac{1}{K} \sum_{k=1}^K F1_k \quad (6)$$

Kesesuaian kuantisasi diverifikasi menggunakan Percentage Agreement (PA), mengukur proporsi sampel dengan keputusan kelas yang sama antara model Float32 dan INT8, dan Mean Absolute Quantization Error (MAQE), mengukur rata-rata selisih probabilitas keluaran keduanya yang masing-masing dihitung menggunakan persamaan (7) dan (8).

$$PA = \frac{1}{N} \sum_{i=1}^N 1[y_i^{f32} = y_i^{int8}] \quad (7)$$

$$MAQE = \frac{1}{N \times K} \sum_{i=1}^N \sum_{k=1}^K |p_{ik}^{f32} - p_{ik}^{int8}| \quad (8)$$

Pengukuran kinerja komputasi dilakukan secara terisolasi menggunakan satu jendela masukan tetap untuk mengukur latensi dan penggunaan memori tanpa gangguan peripheral.

## HASIL DAN PEMBAHASAN

### Kinerja Kalibrasi Ambang

Penggunaan keputusan argmax pada model 1D-CNN menghasilkan akurasi keseluruhan 0,965, tetapi *recall* K2 hanya 0,037, yang berarti hampir seluruh kejadian hujan berintensitas tinggi gagal terdeteksi. Hasil ini merupakan konsekuensi langsung dari ketimpangan kelas yang ekstrem karena model yang dioptimalkan untuk akurasi akan cenderung mengabaikan kelas yang hanya mencakup 1,47% data. Perbandingan kinerja sebelum dan sesudah penyesuaian ambang ditunjukkan pada Tabel 5.

Tabel 5. Kinerja Model 1D-CNN Sebelum dan Sesudah Penyesuaian Ambang

| Argmax    | Kelas | Precision | Recall | F1    |
|-----------|-------|-----------|--------|-------|
|           | K0    | 0,972     | 0,995  | 0,984 |
|           | K1    | 0,452     | 0,258  | 0,328 |
|           | K2    | 0,542     | 0,037  | 0,070 |
| Macro-F1  |       |           |        | 0,461 |
| Akurasi   |       |           |        | 0,965 |
| Ambang PR | Kelas | Precision | Recall | F1    |
|           | K0    | 0,978     | 0,985  | 0,982 |
|           | K1    | 0,354     | 0,312  | 0,332 |
|           | K2    | 0,333     | 0,236  | 0,276 |
| Macro-F1  |       |           |        | 0,530 |
| Akurasi   |       |           |        | 0,959 |

Sumber: (Hasil Penelitian, 2026)

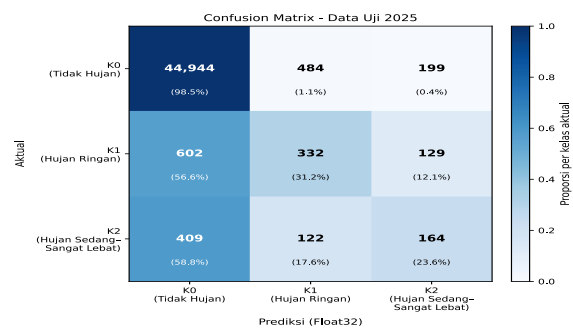
Setelah penyesuaian ambang, *recall* K2 meningkat dari 0,037 menjadi 0,236 atau sekitar 6,4 kali lipat, dan Macro-F1 naik dari 0,461 menjadi 0,530. Peningkatan ini ditukar dengan penurunan akurasi sebesar 0,6 poin persen yang dapat diabaikan mengingat tujuan utama sistem adalah mendeteksi kejadian hujan berbahaya. Untuk kelas K1, *recall* meningkat dari 0,258 menjadi 0,312 dengan penurunan presisi dari 0,452 menjadi 0,354, yang mencerminkan *trade-off* yang wajar antara sensitivitas dan ketepatan ketika ambang diturunkan. Sebaran kesalahan klasifikasi setelah penyesuaian ambang ditunjukkan pada Gambar 5. Dari 695 sampel K2 pada data uji, sebanyak 164 sampel (23,6%) berhasil dikenali dengan benar, 122 sampel (17,6%) terklasifikasi sebagai K1, dan 409 sampel (58,8%) masih terlewat sebagai K0. Dari sisi alarm palsu, presisi K2 sebesar 0,333 berarti sekitar dua dari tiga prediksi K2 tidak terkonfirmasi.

Mengingat jendela waktu yang saling bertumpang tindih dapat menghasilkan banyak sampel dari kejadian yang sama, evaluasi berbasis peristiwa dilakukan sebagai pelengkap. Dari 695 sampel K2, terdapat 81 kejadian K2 independen dengan durasi 10–340 menit dan rata-rata 88 menit, menggunakan definisi pemisahan minimal 60 menit kondisi non-K2. Model berhasil mendeteksi 52 dari 81 kejadian (64,2%), jauh lebih optimis dari *recall* tingkat jendela (23,6%) karena hanya satu jendela yang benar dalam satu kejadian sudah dihitung terdeteksi. Alarm palsu hanya 322 jendela (0,7% dari total jendela uji), menunjukkan sistem jarang memberikan peringatan keliru di luar periode hujan aktual. Meskipun demikian, 409 sampel K2 yang terlewat sebagai K0 menunjukkan bahwa ambang K2 saat ini ( $THR_{K2} = 0,1691$ ) dapat disesuaikan lebih rendah di lapangan jika pengguna menginginkan margin of safety yang lebih konservatif, dengan konsekuensi peningkatan alarm palsu yang perlu dipertimbangkan sesuai konteks operasional.

Dari sudut pandang operasional, peningkatan *recall* K2 lebih bermakna daripada akurasi keseluruhan karena kegagalan mendeteksi hujan intensitas tinggi berisiko lebih besar daripada alarm yang sesekali keliru. Dalam konteks sistem peringatan dini, alarm palsu yang sesekali muncul masih dapat diterima selama sistem diposisikan sebagai lapis peringatan awal yang memerlukan konfirmasi sebelum tindakan lanjutan diambil. *Recall* K2 yang masih moderat mencerminkan tantangan yang wajar pada data resolusi per jam, di mana proporsi kejadian hujan tinggi hanya sekitar 1,47% sehingga model memiliki sangat sedikit sampel untuk mempelajari pola kelas tersebut.

Kondisi ini berbeda dengan data resolusi harian yang distribusi kelasnya lebih seimbang, sehingga hasil penelitian ini tidak dapat dibandingkan langsung dengan studi resolusi harian (Widiputro, 2022).

Tantangan klasifikasi hujan ekstrem dengan ketimpangan kelas juga dilaporkan pada studi yang mengevaluasi CNN-LSTM dan Transformer dengan berbagai strategi augmentasi data, di mana F1 tertinggi pada kelas hujan lebat hanya mencapai 0,1429 meskipun telah menerapkan augmentasi berbasis fisik (Yu dkk., 2026). Macro-F1 sebesar 0,530 yang dicapai penelitian ini lebih tinggi, meskipun perbandingan langsung perlu dilakukan dengan hati-hati mengingat perbedaan resolusi temporal, jumlah kelas, dan skala data yang digunakan.



Sumber: (Hasil Penelitian, 2026)

Gambar 5. Confusion Matrix Model 1D-CNN pada Data Uji setelah Penyesuaian Ambang

### Perbandingan Antarmodel

Perbandingan dilakukan menggunakan data, fitur, pembagian waktu kronologis, dan prosedur kalibrasi ambang yang sama untuk seluruh model. Secara keseluruhan, seluruh model menunjukkan Macro-F1 yang lebih tinggi dibandingkan konfigurasi argmax, yang mengonfirmasi bahwa penyesuaian ambang PR memberikan manfaat lintas arsitektur, bukan hanya pada 1D-CNN. Perbandingan lengkap kinerja seluruh model disajikan pada Tabel 6.

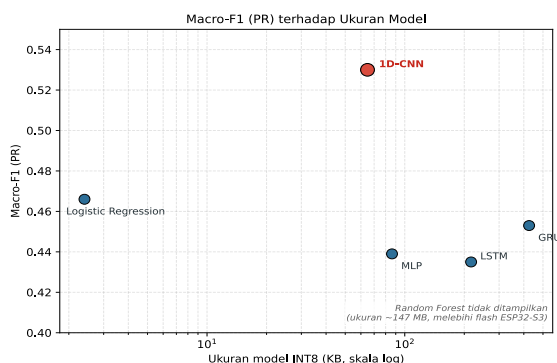
Tabel 6. Perbandingan Kinerja Antarmodel pada Konfigurasi Identik

| Model  | Macro-F1 (PR) | Recall K2 | K1-F1 | K2-F1 | Ukuran (KB) |
|--------|---------------|-----------|-------|-------|-------------|
| LR     | 0,466         | 0,102     | 0,254 | 0,165 | 2,4         |
| RF     | 0,415         | 0,288     | 0,193 | 0,107 | N/A         |
| MLP    | 0,439         | 0,056     | 0,242 | 0,100 | 86,2        |
| LSTM   | 0,435         | 0,111     | 0,202 | 0,139 | 216,5       |
| GRU    | 0,453         | 0,094     | 0,230 | 0,153 | 425,4       |
| 1D-CNN | 0,530         | 0,236     | 0,332 | 0,276 | 64,8        |

Sumber: (Hasil Penelitian, 2026)

Di antara model klasik, Logistic Regression mencapai Macro-F1 tertinggi sebesar 0,466, melampaui MLP yang mencapai 0,439. Hasil ini menunjukkan bahwa hubungan linear antara fitur meteorologi dan kelas intensitas hujan sudah cukup informatif, namun kapasitas model yang terbatas membuat keduanya tidak mampu mendeteksi kelas K2 secara memadai dengan K2-F1 masing-masing hanya 0,165 dan 0,100. Seluruh ukuran model pada Tabel 6, kecuali Random Forest, diperoleh dari berkas .tflite INT8 menggunakan TensorFlow Lite Converter. Model LR dan MLP dikonversi melalui wrapper Keras sebelum diekspor ke .tflite INT8, sedangkan 1D-CNN, LSTM, dan GRU diekspor langsung dari model Keras. Random Forest tidak dapat dikonversi ke format TFLite karena keterbatasan teknis yang inheren pada arsitektur pohon keputusan, sehingga ukurannya tidak tersedia dalam format yang setara. Random Forest menunjukkan *recall* K2 tertinggi di antara seluruh model pembandingan sebesar 0,288 namun presisinya sangat rendah sehingga K2-F1-nya hanya 0,107. Lebih jauh, Random Forest tidak dapat diterapkan pada perangkat target karena representasi kode C melalui *emlearn* mencapai ~147 MB, melebihi kapasitas *flash* ESP32-S3 sebesar 16 MB.

Model berbasis deret waktu LSTM dan GRU menghasilkan Macro-F1 sebesar 0,435 dan 0,453, lebih rendah dari Logistic Regression. Hasil ini mengindikasikan bahwa mekanisme gating rekuren tidak memberikan keunggulan yang diharapkan pada data hujan per jam dengan jendela waktu 18 langkah, kemungkinan karena pola temporal yang relevan lebih bersifat lokal dan tidak memerlukan memori jangka panjang. Ukuran keduanya juga jauh lebih besar dari 1D-CNN, yaitu 216,5 KB untuk LSTM dan 425,4 KB untuk GRU. Augmentasi data untuk keduanya sudah dicoba namun menghasilkan Macro-F1 validasi yang lebih rendah sehingga konfigurasi standar dipertahankan.



Sumber: (Hasil Penelitian, 2026)

Gambar 6. Sebaran Macro-F1 (PR) Terhadap Ukuran Model (skala log)

Model 1D-CNN mencapai Macro-F1 tertinggi sebesar 0,530 dengan K1-F1 0,332 dan K2-F1 0,276, sekaligus memiliki ukuran model sebesar 64,8 KB yang paling ringkas di antara seluruh model berbasis neural network. Hal ini menunjukkan bahwa mekanisme konvolusi lokal dengan koneksi residual dan *attention pooling* lebih sesuai untuk menangkap pola temporal pada data hujan per jam dibandingkan pendekatan rekuren maupun klasik. Hubungan antara kinerja dan ukuran model ditunjukkan pada Gambar 6, yang memperlihatkan bahwa 1D-CNN menempati posisi paling unggul untuk deployment pada perangkat tertanam berbiaya rendah.

### Kesesuaian Kuantisasi dan Kinerja *On-Device*

*Post-training quantization* INT8 mereduksi ukuran model dari 185,8 KB menjadi 64,8 KB, setara dengan rasio kompresi 2,87 kali. Pada ESP32-S3 N16R8 dengan kapasitas *flash* 16 MB, model hanya mengonsumsi sekitar 0,4% dari total kapasitas *flash* sehingga menyisakan ruang yang lebih dari cukup untuk *firmware*, sistem file, dan komponen sistem lainnya. Hasil pengukuran lengkap disajikan pada Tabel 7.

Tabel 7. Kinerja Inferensi Model pada ESP32-S3

| Parameter                                   | Nilai     |
|---------------------------------------------|-----------|
| Ukuran model Float32 (.tflite)              | 185,8 KB  |
| Ukuran model INT8 (.tflite)                 | 64,8 KB   |
| Memori kerja ( <i>tensor arena</i> )        | 36,5 KB   |
| Latensi pembentukan fitur                   | 0,002 ms  |
| Latensi inferensi model                     | 13,853 ms |
| Latensi total (preprocessing dan inferensi) | 13,873 ms |
| Heap bebas saat berjalan                    | 234,7 KB  |

Sumber: (Hasil Penelitian, 2026)

Sebelum model INT8 digunakan untuk inferensi pada perangkat, kesesuaian hasilnya terhadap model Float32 perlu diverifikasi terlebih dahulu. Nilai PA sebesar 99,02% menunjukkan bahwa hanya sekitar 1% sampel uji yang menghasilkan keputusan kelas berbeda antara kedua model. Nilai MAQE sebesar 0,0148 melengkapi temuan ini dengan menunjukkan bahwa selisih probabilitas keluaran antara keduanya rata-rata kurang dari 1,5 poin persentase. Kedua nilai ini secara bersama mengonfirmasi bahwa proses kuantisasi tidak mengubah perilaku keputusan model secara berarti, sehingga model INT8 dapat digunakan sebagai pengganti model Float32 tanpa kehilangan kualitas inferensi yang signifikan.

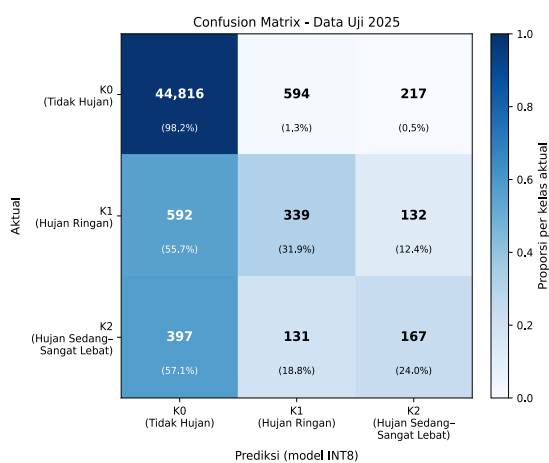
Untuk memverifikasi dampak kuantisasi terhadap kinerja klasifikasi secara lebih rinci, evaluasi metrik lengkap dilakukan pada model INT8 menggunakan data uji. Hasil disajikan pada Tabel 8 dan Gambar 7. Macro-F1 model INT8 sebesar 0,525

hanya turun 0,005 dibandingkan model Float32 (0,530). Yang lebih penting, *recall* K2 model INT8 sebesar 0,240 bahkan sedikit meningkat dibandingkan model Float32 (0,236), menunjukkan bahwa kuantisasi tidak mengorbankan kemampuan deteksi kelas hujan berbahaya. Penurunan terbesar terjadi pada presisi K1 dari 0,354 menjadi 0,319, namun tidak mengubah kesimpulan tentang kelayakan model secara keseluruhan.

Tabel 8. Perbandingan Metrik Klasifikasi Model Float32 dan INT8

| Float32  | Kelas | Precision | Recall | F1    |
|----------|-------|-----------|--------|-------|
|          | K0    | 0,978     | 0,985  | 0,982 |
|          | K1    | 0,354     | 0,312  | 0,332 |
|          | K2    | 0,333     | 0,236  | 0,276 |
| Macro-F1 |       |           |        | 0,530 |
| Akurasi  |       |           |        | 0,959 |
| INT8     | Kelas | Precision | Recall | F1    |
|          | K0    | 0,978     | 0,982  | 0,980 |
|          | K1    | 0,319     | 0,319  | 0,319 |
|          | K2    | 0,324     | 0,240  | 0,276 |
| Macro-F1 |       |           |        | 0,525 |
| Akurasi  |       |           |        | 0,957 |

Sumber: (Hasil Penelitian, 2026)



Sumber: (Hasil Penelitian, 2026)

Gambar 7. Confusion Matrix Model 1D-CNN INT8 pada Data Uji

Dengan kesesuaian kuantisasi yang terkonfirmasi, pengujian kinerja *on-device* dilakukan pada ESP32-S3. Dari sisi memori, *tensor arena* sebesar 36,5 KB merupakan satu-satunya alokasi memori dinamis yang dibutuhkan selama inferensi berlangsung. *Heap* bebas sebesar 234,7 KB yang tersisa saat model berjalan menunjukkan bahwa perangkat masih memiliki kapasitas memori yang memadai untuk operasi *firmware* lainnya seperti pembacaan sensor, komunikasi, dan operasi penyimpanan data.

Pengukuran latensi dilakukan sebanyak 100 kali pengulangan secara terisolasi menggunakan satu jendela masukan tetap tanpa gangguan *peripheral*. Latensi rata-rata pembentukan fitur sebesar 0,002 ms (min 0,001 ms, maks 0,003 ms), inferensi model sebesar 13,853 ms (min 13,847 ms, maks 13,864 ms), dan latensi total sebesar 13,873 ms (min 13,868 ms, maks 13,882 ms). Rentang min-maks yang sangat sempit mengkonfirmasi latensi yang stabil dan deterministik. Standar deviasi tidak dilaporkan karena nilai individual tidak disimpan selama pengujian.

Latensi total sebesar 13,873 ms mencakup latensi pembentukan fitur sebesar 0,002 ms dan latensi inferensi model sebesar 13,853 ms. Nilai ini mencerminkan waktu yang dibutuhkan dari pembentukan fitur hingga keluarnya keputusan kelas, tidak termasuk pembacaan sensor, komunikasi, dan penyimpanan data. Latensi total ini sangat jauh di bawah interval pengamatan sistem meteorologi, sehingga model dapat dijalankan secara *real-time* tanpa membebani siklus kerja perangkat. Dengan demikian, model 1D-CNN berbasis TinyML terbukti layak dijalankan secara mandiri pada ESP32-S3 tanpa ketergantungan pada *server* atau konektivitas internet.

Keunggulan 1D-CNN pada perangkat tertanam dikonfirmasi oleh studi yang menerapkan model 1D-CNN berbasis TinyML pada ESP32 untuk klasifikasi data sensor secara *real-time*, yang menunjukkan bahwa arsitektur ini mampu mencapai akurasi deteksi di atas 92% dengan jejak komputasi yang ringkas tanpa ketergantungan pada *server* (Gupta & Shivhare, 2025). Temuan ini konsisten dengan hasil penelitian ini di mana 1D-CNN menghasilkan latensi 13,873 ms dan ukuran model 64,8 KB, lebih ringkas dibandingkan LSTM (216,5 KB) dengan kinerja Macro-F1 yang lebih tinggi.

## KESIMPULAN

Penelitian ini berhasil mengembangkan model klasifikasi intensitas hujan per jam berbasis 1D-CNN yang layak dijalankan secara *on-device* pada ESP32-S3 sebagai prototipe komponen inferensi untuk sistem peringatan dini hujan. Kalibrasi ambang berbasis kurva *Precision-Recall* terbukti efektif meningkatkan kemampuan deteksi kelas hujan minoritas. *Recall* kelas K2 meningkat dari 0,037 menjadi 0,236 dan Macro-F1 naik dari 0,461 menjadi 0,530 setelah penyesuaian ambang diterapkan. Perbandingan antarmodel menunjukkan bahwa 1D-CNN mencapai Macro-F1 tertinggi sebesar 0,530 dengan ukuran model 64,8 KB, lebih ringkas dibandingkan LSTM (216,5 KB)

dan GRU (425,4 KB) yang menghasilkan Macro-F1 lebih rendah. Hasil ini memvalidasi pemilihan 1D-CNN sebagai arsitektur yang paling sesuai untuk konteks klasifikasi data hujan per jam pada perangkat tertanam berbiaya rendah.

Dari sisi implementasi, *post-training quantization* INT8 mereduksi ukuran model sebesar 2,87 kali dari 185,8 KB menjadi 64,8 KB dengan kesesuaian keputusan sebesar 99,02% dan MAQE 0,0148 terhadap model Float32. Model berjalan pada ESP32-S3 dengan latensi 13,873 ms, *tensor arena* 36,5 KB, dan *heap* bebas 234,7 KB, membuktikan bahwa inferensi dapat dilakukan secara *real-time* dengan konsumsi sumber daya yang rendah. Keseluruhan hasil ini menunjukkan bahwa model 1D-CNN berbasis TinyML layak menjadi komponen inferensi yang beroperasi secara mandiri dalam sistem peringatan dini hujan, khususnya di wilayah dengan keterbatasan infrastruktur jaringan.

Penelitian selanjutnya dapat mempertimbangkan penggunaan data dari lebih dari satu stasiun untuk meningkatkan generalisasi model, termasuk validasi pada stasiun dengan karakteristik topografi dan pola hujan yang berbeda seperti stasiun di dataran tinggi atau wilayah kepulauan. Eksplorasi strategi penanganan ketimpangan kelas yang lebih lanjut seperti augmentasi data sintetis atau fungsi galat lainnya perlu dilakukan untuk meningkatkan kemampuan deteksi kelas hujan langka. Selain itu, pengujian latensi dengan pengulangan dan pelaporan standar deviasi perlu dilakukan untuk memperoleh estimasi kinerja komputasi yang lebih *robust*, serta validasi model melalui pengujian langsung pada sistem peringatan dini yang terintegrasi.

## REFERENSI

- Akbar, H., & Sanjaya, W. K. (2023). Kajian Performa Metode Class Weight Random Forest pada Klasifikasi Imbalance Data Kelas Curah Hujan. *Jurnal Sains, Nalar, dan Aplikasi Teknologi Informasi*, 3(1), 42–49. <https://doi.org/10.20885/snati.v3i1.30>
- Badan Nasional Penanggulangan Bencana. (2024). *Buku data bencana Indonesia tahun 2023*. <https://bnpb.go.id/buku/buku-data-bencana-indonesia-tahun-2023>
- Bauer, P. (2024). What if? Numerical weather prediction at the crossroads. *Journal of the European Meteorological Society*, 1, 100002. <http://arxiv.org/abs/2407.03787>
- Chi, D. T. K., Trang, N. T. M., Son, T. B. M., Thao, N. N., & Nguyen, T. Q. (2025). Hybrid Deep Learning Framework for Robust Time-Series Classification: Integrating Inception Modules with Residual Networks. *Journal of Algorithms & Computational Technology*, 19, 1–20. <https://doi.org/10.1177/17483026251348851>
- Esposito, M., Palma, L., Belli, A., Sabbatini, L., & Pierleoni, P. (2022). Recent Advances in Internet of Things Solutions for Early Warning Systems: A Review. *Sensors*, 22(6), 2124. <https://doi.org/10.3390/s22062124>
- Farhadpour, S., Warner, T. A., & Maxwell, A. E. (2024). Selecting and Interpreting Multiclass Loss and Accuracy Assessment Metrics for Classifications with Class Imbalance: Guidance and Best Practices. *Remote Sensing*, 16(3), 533. <https://doi.org/10.3390/rs16030533>
- Glawion, L., Polz, J., Kunstmann, H., Fersch, B., & Chwala, C. (2025). Global Spatio-Temporal ERA5 Precipitation Downscaling to km and Sub-hourly Scale Using Generative AI. *npj Climate and Atmospheric Science*, 8(1), 219. <https://doi.org/10.1038/s41612-025-01103-y>
- Gupta, S., & Shivhare, S. N. (2025). Embedded TinyML for Predictive Maintenance: Vibration Analysis on ESP32 with Real-Time Fault Detection in Industrial Equipment. *International Journal on Computational Modelling Applications*, 2(2), 1–17. <https://doi.org/10.63503/j.ijcma.2025.114>
- Hidayat, R., Saputra, E., & Alsepan, G. (2025). The Impact of Convective Available Potential Energy (CAPE) on Spatio-Temporal Variations of Indonesian Extreme Rainfall. *Journal of Mathematical and Fundamental Sciences*, 57(1), 70–85. <https://doi.org/10.5614/j.math.fund.sci.2025.57.1.5>
- Kong, L., Snášel, V., Bai, Z., Vilimek, D., Mirjalili, S., Pan, J. S., Horakova, J., Martinek, R., & Vilimkova Kahankova, R. (2025). Enhancing Cardiotocography Classification via Ensemble Learning and Threshold Optimization. *Scientific Reports*, 15(1), 38528. <https://doi.org/10.1038/s41598-025-18990-z>
- Liu, J., Niu, L., Yuan, Z., Yang, D., Wang, X., & Liu, W. (2023). PD-Quant: Post-Training Quantization Based on Prediction Difference Metric. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 24427–24437. <https://doi.org/10.1109/CVPR52729.2023.02340>
- Mdegela, L., Municio, E., De Bock, Y., Luhanga, E., Leo, J., & Mannens, E. (2023). Extreme Rainfall Event Classification Using Machine Learning

- for Kikuletwa River Floods. *Water*, 15(6), 1021. <https://doi.org/10.3390/w15061021>
- Putra, M., Rosid, M. S., & Handoko, D. (2024). High-Resolution Rainfall Estimation Using Ensemble Learning Techniques and Multisensor Data Integration. *Sensors*, 24(15), 5030. <https://doi.org/10.3390/s24155030>
- Rachmawardani, A., Wijaya, S. K., & Shopaheluwakan, A. (2022). Sistem Peringatan Dini Banjir Berbasis Machine Learning: Studi Literatur. *METHOMIKA: Jurnal Manajemen Informatika dan Komputerisasi Akuntansi*, 6(2), 188–198. <https://doi.org/10.46880/jmika.Vol6No2.pp188-198>
- Rajaraman, S., Ganesan, P., & Antani, S. (2022). Deep Learning Model Calibration for Improving Performance in Class-Imbalanced Medical Image Classification Tasks. *PLOS ONE*, 17(1), e0262838. <https://doi.org/10.1371/journal.pone.0262838>
- Rizal, M. E., Wigena, A. H., & Afendi, F. M. (2022). Time Series Imputation Using VAR-IM (Case Study: Weather Data in Meteorological Station of Citeko). *Barekeng*, 16(4), 1373–1384. <https://doi.org/10.30598/barekengvol16iss4pp1373-1384>
- Rozzy, F., Candra Rini Novitasari, D., Yuliati, D., & Permata Sani, P. (2024). Forecasting Sea Surface Salinity in the Eastern Madura Strait Using 1D Convolutional Neural Network. *Jurnal Informatika dan Teknologi Informasi*, 21(1), 14–30. <https://doi.org/10.31515/telematika.v21i1.8959>
- Shantal, M., Othman, Z., & Bakar, A. A. (2023). A Novel Approach for Data Feature Weighting Using Correlation Coefficients and Min–Max Normalization. *Symmetry*, 15(12), 2185. <https://doi.org/10.3390/sym15122185>
- Taheri Moghadar, S., Gandolfi, R., & Torti, E. (2025). TinyML-Based Real-Time Doorway Activity Recognition with a Time-of-Flight Sensor. *Electronics*, 14(17), 3533. <https://doi.org/10.3390/electronics14173533>
- Ullah, S., Ullah, N., Siddique, M. F., Ahmad, Z., & Kim, J. M. (2024). Spatio-Temporal Feature Extraction for Pipeline Leak Detection in Smart Cities Using Acoustic Emission Signals: A One-Dimensional Hybrid Convolutional Neural Network–Long Short-Term Memory Approach. *Applied Sciences (Switzerland)*, 14(22). <https://doi.org/10.3390/app142210339>
- Widiputro, R. (2022). *Klasifikasi time-series data hujan harian dengan metode deep learning 1D-CNN dan LSTM pada Stasiun Soekarno Hatta, Tangerang, Indonesia* [Universitas Mercu Buana]. <https://repository.mercubuana.ac.id/69137/>
- World Meteorological Organization. (2024). *Guide to Instruments and Methods of Observation Volume I-Measurement of Meteorological Variables*. <https://library.wmo.int/viewer/68695/>
- Yu, W., Sun, Y., Yue, Z., Li, Z., & Liu, Y. (2026). An Architecture-Feature-Enhanced Decision Framework for Deep Learning-Based Prediction of Extreme and Imbalanced Precipitation. *Water (Switzerland)*, 18(2). <https://doi.org/10.3390/w18020176>
- Zaidi, S. A. R., Hayajneh, A. M., Hafeez, M., & Ahmed, Q. Z. (2022). Unlocking Edge Intelligence Through Tiny Machine Learning (TinyML). *IEEE Access*, 10, 100867–100877. <https://doi.org/10.1109/ACCESS.2022.3207200>
- Zhang, Y., Long, M., Chen, K., Xing, L., Jin, R., Jordan, M. I., & Wang, J. (2023). Skilful nowcasting of extreme precipitation with NowcastNet. *Nature*, 619(7970), 526–532. <https://doi.org/10.1038/s41586-023-06184-4>