

**UJIAN AKHIR SEMESTER
MACHINE LEARNING**

Dosen Pengampu;
Izhan Fakhruzi S.T., M.Sc.



Disusun Oleh:

Rizky Aditia P.S	221230009
Dimas Nur Mi'Raj	221230053
Wega Ramadhan	221230055

**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS TEKNIK DAN ILMU KOMPUTER
UNIVERSITAS MUHAMMADIYAH
PONTIANAK**

2026

1. Deskripsi Dataset

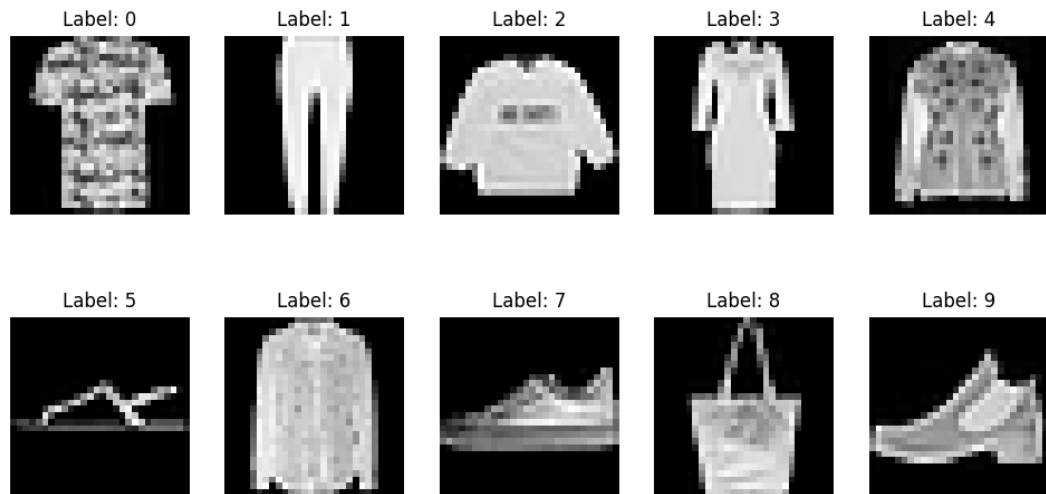
Fashion-MNIST adalah dataset gambar artikel Zalando—terdiri dari set pelatihan sebanyak 60.000 contoh dan set pengujian sebanyak 10.000 contoh. Setiap contoh adalah gambar skala abu-abu berukuran 28x28, yang dikaitkan dengan label dari 10 kelas. Zalando bermaksud agar Fashion-MNIST berfungsi sebagai pengganti langsung untuk dataset MNIST asli untuk pengujian kinerja algoritma pembelajaran mesin. Dataset ini memiliki ukuran gambar dan struktur pembagian pelatihan dan pengujian yang sama.

Dataset MNIST asli berisi banyak angka tulisan tangan. Anggota komunitas AI/ML/Ilmu Data menyukai dataset ini dan menggunakannya sebagai tolok ukur untuk memvalidasi algoritma mereka. Bahkan, MNIST seringkali menjadi dataset pertama yang dicoba oleh para peneliti.

Setiap gambar memiliki tinggi 28 piksel dan lebar 28 piksel, dengan total 784 piksel. Setiap piksel memiliki nilai piksel tunggal yang terkait dengannya, yang menunjukkan tingkat terang atau gelap piksel tersebut, dengan angka yang lebih tinggi berarti lebih gelap. Nilai piksel ini adalah bilangan bulat antara 0 dan 255. Kumpulan data pelatihan dan pengujian memiliki 785 kolom. Kolom pertama terdiri dari label kelas (lihat di atas), dan mewakili jenis pakaian. Setiap contoh pelatihan dan pengujian diberi salah satu label berikut:

- 0 kaos/atasan
- 1 Celana Panjang
- 2 Pullover
- 3 Gaun
- 4 Lapisan
- 5 Sandal
- 6 Kemeja
- 7 Sepatu Sneaker
- 8 Tas
- 9 Sepatu bot pergelangan kaki

Contoh Gambar untuk Setiap Kelas Fashion-MNIST

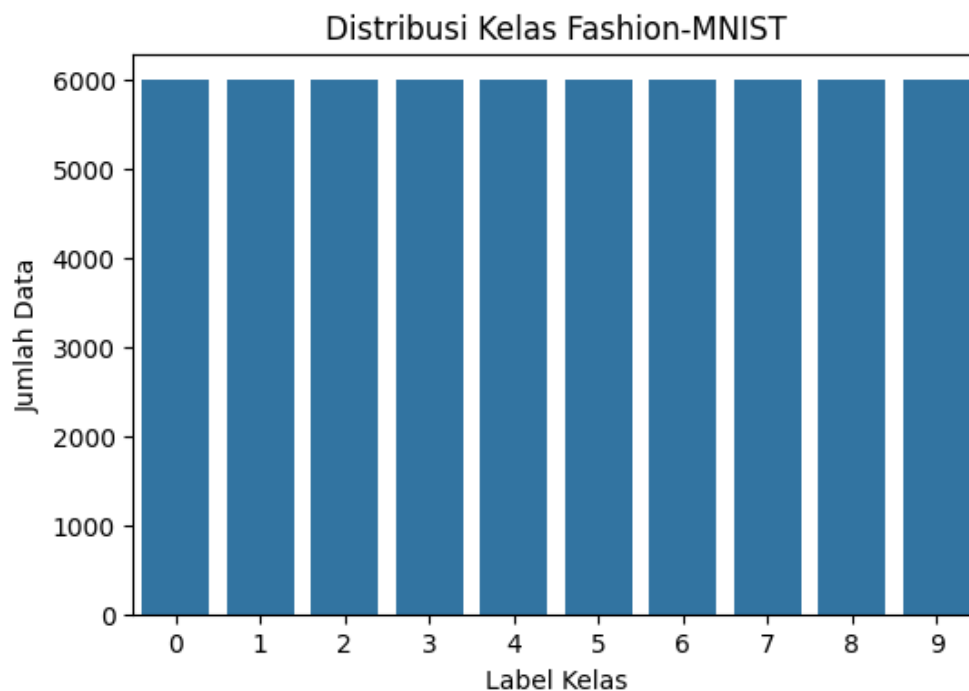


Gambar 1. Label untuk kelas 0 - 9

2. EDA

Grafik distribusi kelas menunjukkan jumlah data pada setiap label, yaitu kelas 0 sampai kelas 9. Berdasarkan grafik, terlihat bahwa setiap kelas memiliki jumlah data yang hampir sama, yaitu sekitar 6.000 data pada data latih. Hal ini menunjukkan bahwa dataset Fashion-MNIST memiliki distribusi kelas yang seimbang (balanced dataset).

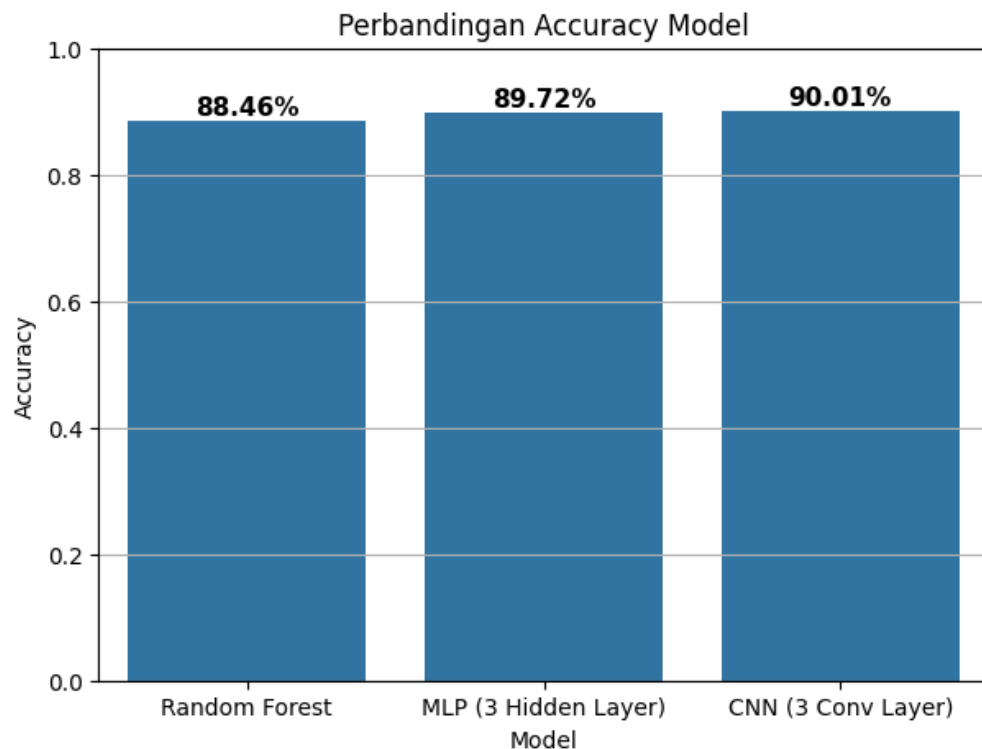
Kondisi ini sangat penting dalam proses machine learning karena ketidakseimbangan kelas (class imbalance) dapat menyebabkan model cenderung memihak pada kelas tertentu dan menghasilkan performa yang tidak objektif. Dengan distribusi kelas yang seimbang, setiap kategori pakaian memiliki kesempatan yang sama untuk dipelajari oleh model, sehingga hasil evaluasi menjadi lebih adil dan representatif.



Gambar 2. EDA

3. Accuracy Model

Accuracy digunakan sebagai metrik utama untuk mengukur performa model. Accuracy menunjukkan persentase jumlah data yang berhasil diklasifikasikan dengan benar dibandingkan dengan seluruh data uji. Hasil pengujian menunjukkan bahwa Random Forest sebagai metode traditional machine learning menghasilkan nilai akurasi yang cukup baik sebesar 88,46%, namun masih lebih rendah dibandingkan metode deep learning. Model MLP dengan tiga hidden layer menghasilkan akurasi yang lebih tinggi sebesar 89,72%, yang menunjukkan bahwa jaringan saraf mampu mempelajari hubungan non-linear dari data citra lebih baik dibandingkan Random Forest. Model CNN dengan tiga convolution layer memperoleh nilai akurasi tertinggi sebesar 90,01%. Hal ini disebabkan CNN memiliki kemampuan khusus dalam mengekstraksi fitur spasial dari citra, seperti tepi, pola, dan bentuk objek. Dengan demikian, CNN lebih efektif dalam mengenali karakteristik visual pada data Fashion-MNIST. Hasil ini membuktikan bahwa arsitektur deep learning, khususnya CNN, lebih sesuai digunakan untuk permasalahan klasifikasi citra dibandingkan metode machine learning tradisional.



Gambar 3. Perbandingan Akurasi Model

4. Learning Curve DL Model

- **MLP**

Learning curve MLP menunjukkan hubungan antara jumlah epoch dengan nilai akurasi pada data training dan data validation. Grafik memperlihatkan bahwa akurasi training meningkat secara konsisten seiring bertambahnya epoch. Hal ini menandakan bahwa model mampu mempelajari pola dari data latih dengan baik. Akurasi validation meningkat pada awal proses pelatihan, kemudian cenderung stabil dan memiliki selisih dibandingkan dengan akurasi training. Kondisi ini menunjukkan adanya kecenderungan overfitting ringan, yaitu ketika model mulai terlalu menyesuaikan diri terhadap data training sehingga peningkatan kemampuan generalisasi pada data baru tidak terlalu signifikan. Meskipun demikian, selisih antara training accuracy dan validation accuracy tidak terlalu besar, sehingga model MLP masih dapat dikategorikan memiliki performa yang cukup baik dan mampu melakukan generalisasi dengan baik

```

/usr/local/lib/python3.12/dist-packages/keras/src/layers/core/dense.py:93: UserWarning: Do not pass an `input_shape`/`input_dim`
super().__init__(activity_regularizer=activity_regularizer, **kwargs)
... Epoch 1/20
469/469 ----- 8s 14ms/step - accuracy: 0.7702 - loss: 0.6520 - val_accuracy: 0.8640 - val_loss: 0.3752
Epoch 2/20
469/469 ----- 8s 17ms/step - accuracy: 0.8661 - loss: 0.3709 - val_accuracy: 0.8731 - val_loss: 0.3430
Epoch 3/20
469/469 ----- 8s 17ms/step - accuracy: 0.8807 - loss: 0.3211 - val_accuracy: 0.8421 - val_loss: 0.4153
Epoch 4/20
469/469 ----- 6s 14ms/step - accuracy: 0.8850 - loss: 0.3104 - val_accuracy: 0.8883 - val_loss: 0.2973
Epoch 5/20
469/469 ----- 8s 17ms/step - accuracy: 0.8970 - loss: 0.2763 - val_accuracy: 0.8859 - val_loss: 0.3145
Epoch 6/20
469/469 ----- 6s 14ms/step - accuracy: 0.9021 - loss: 0.2609 - val_accuracy: 0.8832 - val_loss: 0.3136
Epoch 7/20
469/469 ----- 8s 17ms/step - accuracy: 0.9052 - loss: 0.2542 - val_accuracy: 0.8912 - val_loss: 0.3018
Epoch 8/20
469/469 ----- 8s 17ms/step - accuracy: 0.9100 - loss: 0.2404 - val_accuracy: 0.8918 - val_loss: 0.2960
Epoch 9/20
469/469 ----- 7s 15ms/step - accuracy: 0.9143 - loss: 0.2277 - val_accuracy: 0.8940 - val_loss: 0.2957
Epoch 10/20
469/469 ----- 8s 17ms/step - accuracy: 0.9194 - loss: 0.2143 - val_accuracy: 0.8864 - val_loss: 0.3151
Epoch 11/20
469/469 ----- 7s 14ms/step - accuracy: 0.9178 - loss: 0.2175 - val_accuracy: 0.8958 - val_loss: 0.2897
Epoch 12/20
469/469 ----- 8s 17ms/step - accuracy: 0.9241 - loss: 0.1980 - val_accuracy: 0.8970 - val_loss: 0.2981
Epoch 13/20
469/469 ----- 6s 14ms/step - accuracy: 0.9235 - loss: 0.1993 - val_accuracy: 0.8991 - val_loss: 0.2933
Epoch 14/20
469/469 ----- 8s 17ms/step - accuracy: 0.9283 - loss: 0.1849 - val_accuracy: 0.8914 - val_loss: 0.3210
Epoch 15/20
469/469 ----- 7s 14ms/step - accuracy: 0.9306 - loss: 0.1806 - val_accuracy: 0.8920 - val_loss: 0.3332
Epoch 16/20
469/469 ----- 10s 14ms/step - accuracy: 0.9332 - loss: 0.1753 - val_accuracy: 0.8997 - val_loss: 0.3021
Epoch 17/20
469/469 ----- 10s 14ms/step - accuracy: 0.9368 - loss: 0.1615 - val_accuracy: 0.8980 - val_loss: 0.3176
Epoch 18/20
469/469 ----- 8s 16ms/step - accuracy: 0.9388 - loss: 0.1577 - val_accuracy: 0.8927 - val_loss: 0.3503
Epoch 19/20
469/469 ----- 10s 15ms/step - accuracy: 0.9379 - loss: 0.1622 - val_accuracy: 0.9052 - val_loss: 0.3160
Epoch 20/20
469/469 ----- 7s 14ms/step - accuracy: 0.9430 - loss: 0.1468 - val_accuracy: 0.8972 - val_loss: 0.3441
313/313 ----- 1s 3ms/step - accuracy: 0.8942 - loss: 0.3445
Accuracy MLP (3 hidden layer): 0.8971999883651733

```

Gambar 4. Learning curve MLP

- CNN

Learning curve CNN menampilkan hubungan antara jumlah epoch dengan nilai akurasi training dan validation. Grafik menunjukkan bahwa akurasi training meningkat secara bertahap seiring bertambahnya epoch. Hal ini menandakan bahwa model CNN mampu mempelajari fitur-fitur penting dari citra Fashion-MNIST dengan baik. Akurasi validation mengalami peningkatan yang cukup signifikan pada epoch-epoch awal, kemudian cenderung stabil pada kisaran nilai tertentu meskipun akurasi training terus meningkat. Kondisi ini menunjukkan adanya kecenderungan overfitting ringan, yaitu ketika model semakin menyesuaikan diri terhadap data training, namun peningkatan kemampuan generalisasi terhadap data baru tidak terlalu besar. Meskipun demikian, akurasi validation tetap berada pada nilai yang tinggi dan relatif stabil, serta selisih antara training accuracy dan validation accuracy tidak terlalu besar. Hal ini menunjukkan bahwa model CNN memiliki performa yang baik dan mampu melakukan generalisasi dengan cukup baik pada data uji.

```

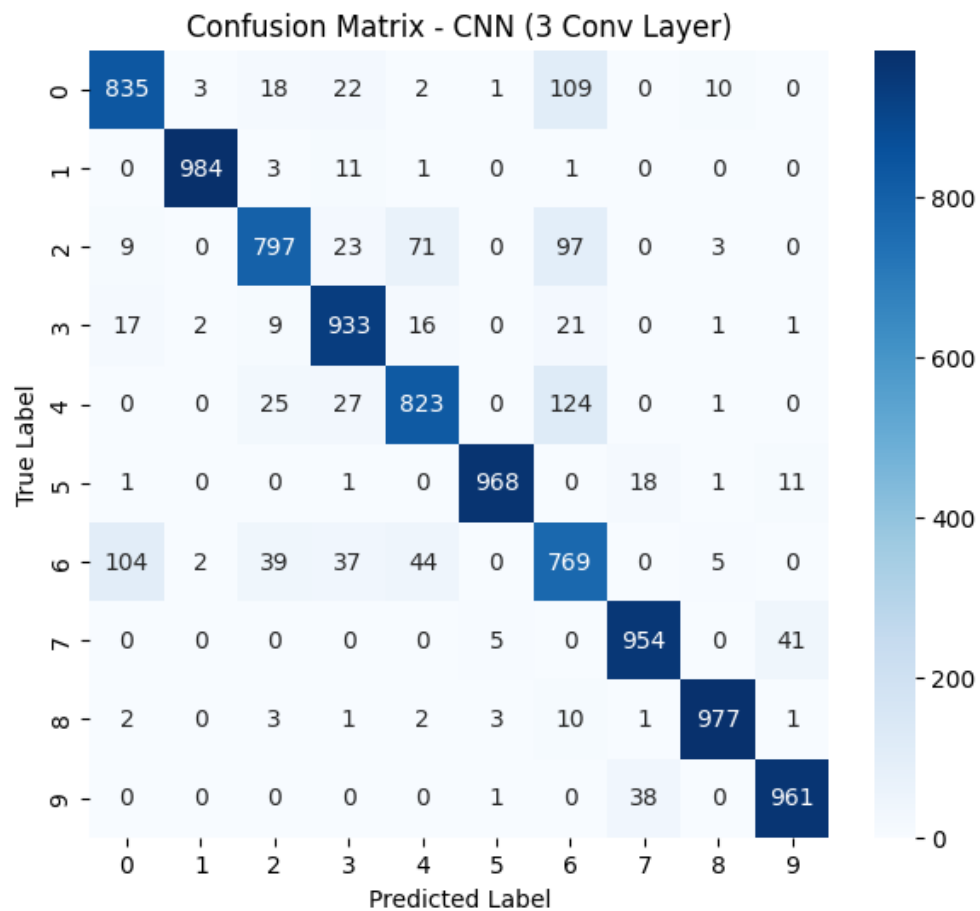
/usr/local/lib/python3.12/dist-packages/keras/src/layers/convolutional/base_conv.py:113: UserWarning: Do not pass an `input_shape` argument to `Conv2D` layers.
super().__init__(activity_regularizer=activity_regularizer, **kwargs)
Epoch 1/25
469/469 — 83s 167ms/step - accuracy: 0.6296 - loss: 1.1164 - val_accuracy: 0.6991 - val_loss: 0.8476
Epoch 2/25
469/469 — 78s 166ms/step - accuracy: 0.8120 - loss: 0.5188 - val_accuracy: 0.8632 - val_loss: 0.3623
Epoch 3/25
469/469 — 83s 168ms/step - accuracy: 0.8364 - loss: 0.4505 - val_accuracy: 0.8664 - val_loss: 0.3500
Epoch 4/25
469/469 — 78s 167ms/step - accuracy: 0.8493 - loss: 0.4134 - val_accuracy: 0.8848 - val_loss: 0.3067
Epoch 5/25
469/469 — 79s 167ms/step - accuracy: 0.8600 - loss: 0.3801 - val_accuracy: 0.8834 - val_loss: 0.3062
Epoch 6/25
469/469 — 83s 170ms/step - accuracy: 0.8654 - loss: 0.3657 - val_accuracy: 0.8912 - val_loss: 0.2906
Epoch 7/25
469/469 — 78s 167ms/step - accuracy: 0.8714 - loss: 0.3550 - val_accuracy: 0.8928 - val_loss: 0.2852
Epoch 8/25
469/469 — 84s 173ms/step - accuracy: 0.8741 - loss: 0.3424 - val_accuracy: 0.8676 - val_loss: 0.3310
Epoch 9/25
469/469 — 79s 167ms/step - accuracy: 0.8799 - loss: 0.3297 - val_accuracy: 0.8985 - val_loss: 0.2707
Epoch 10/25
469/469 — 78s 166ms/step - accuracy: 0.8833 - loss: 0.3221 - val_accuracy: 0.9001 - val_loss: 0.2687
Epoch 11/25
469/469 — 78s 167ms/step - accuracy: 0.8844 - loss: 0.3159 - val_accuracy: 0.8891 - val_loss: 0.2902
Epoch 12/25
469/469 — 82s 166ms/step - accuracy: 0.8871 - loss: 0.3091 - val_accuracy: 0.8924 - val_loss: 0.2908
Epoch 13/25
469/469 — 81s 173ms/step - accuracy: 0.8915 - loss: 0.2991 - val_accuracy: 0.8959 - val_loss: 0.2773
Epoch 14/25
469/469 — 79s 167ms/step - accuracy: 0.8912 - loss: 0.2979 - val_accuracy: 0.9004 - val_loss: 0.2733
Epoch 15/25
469/469 — 82s 167ms/step - accuracy: 0.8938 - loss: 0.2948 - val_accuracy: 0.8991 - val_loss: 0.2705

```

Gambar 5. Learning Curve CNN

5. Confusion Matrix Accuracy Terbaik

Confusion matrix digunakan untuk menganalisis performa model secara lebih detail. Berbeda dengan accuracy yang hanya memberikan satu nilai keseluruhan, confusion matrix menunjukkan jumlah prediksi benar dan salah pada setiap kelas. Pada confusion matrix CNN, terlihat bahwa sebagian besar nilai berada pada diagonal utama. Hal ini menunjukkan bahwa sebagian besar data berhasil diklasifikasikan dengan benar oleh model. Semakin besar nilai pada diagonal, semakin baik kemampuan model dalam mengenali kelas tersebut. Namun, masih terdapat beberapa kesalahan klasifikasi yang muncul di luar diagonal. Kesalahan ini terutama terjadi pada kelas-kelas yang memiliki kemiripan visual, seperti T-shirt/top, pullover, coat, dan shirt. Hal ini menunjukkan bahwa model masih mengalami kesulitan dalam membedakan kategori pakaian dengan bentuk dan tekstur yang mirip. Meskipun demikian, secara keseluruhan CNN menunjukkan performa klasifikasi yang sangat baik.



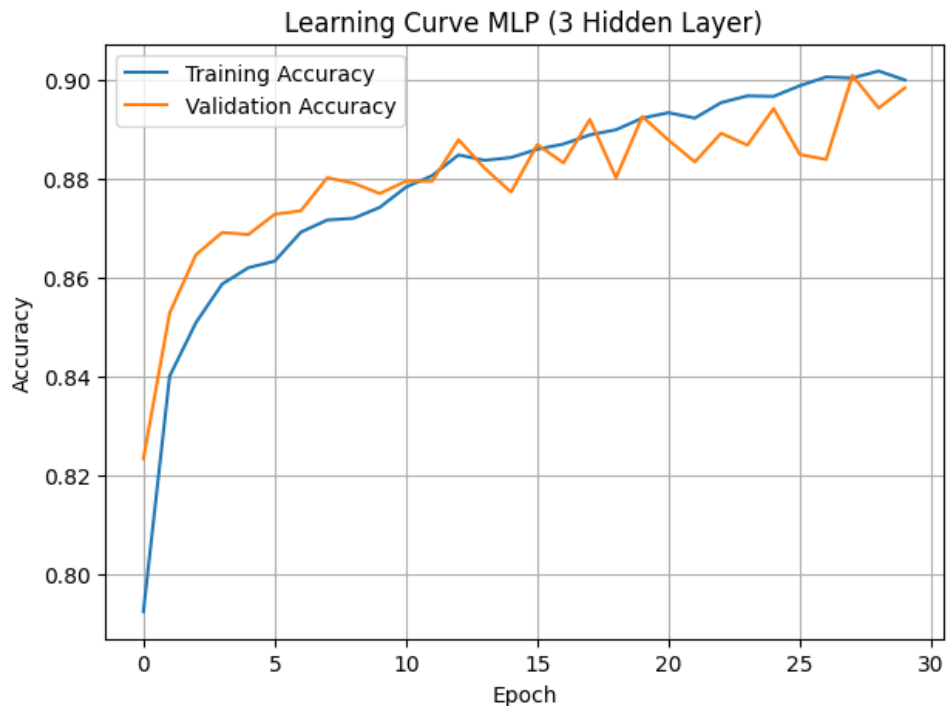
Gambar 6. Confusion Matrix Model CNN

6. Tabel/Kurva Perbandingan

• Learning Curve MLP

Grafik learning curve MLP menunjukkan hubungan antara epoch dan nilai akurasi training serta validation. Pada epoch awal, akurasi training meningkat cukup tajam dari sekitar 79% hingga di atas 85%. Hal ini menunjukkan bahwa model MLP dengan tiga hidden layer mampu dengan cepat mempelajari pola dasar dari citra Fashion-MNIST. Seiring bertambahnya epoch, akurasi training terus meningkat secara bertahap hingga mendekati 89%. Akurasi validation juga mengalami peningkatan, meskipun cenderung berfluktuasi di kisaran 86%–89%. Fluktuasi ini menunjukkan bahwa model mampu melakukan generalisasi terhadap data yang tidak dilatih, namun peningkatannya mulai melambat pada epoch-epoch akhir. Terlihat adanya selisih kecil antara akurasi training dan

validation. Hal ini mengindikasikan adanya kecenderungan overfitting ringan, di mana model sedikit lebih baik dalam mengenali data training dibandingkan data validation. Namun, karena jarak kedua kurva tidak terlalu besar dan tidak terjadi penurunan drastis pada validation accuracy, model MLP masih dapat dikategorikan memiliki performa yang cukup baik dan stabil.

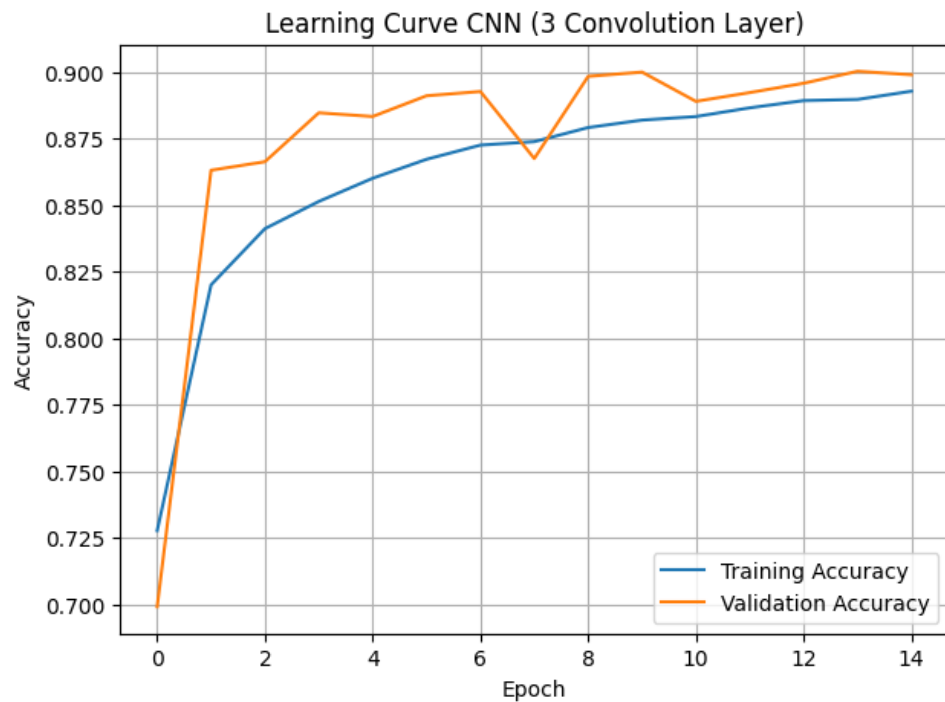


Gambar 7. Curve MLP

- **Learning Curve CNN**

Grafik learning curve CNN menunjukkan peningkatan akurasi yang lebih cepat dan lebih stabil dibandingkan MLP. Pada epoch awal, baik training maupun validation accuracy meningkat tajam, menandakan bahwa CNN mampu dengan efektif mengekstraksi fitur visual dasar seperti tepi dan bentuk objek. Pada epoch menengah hingga akhir, akurasi training dan validation meningkat secara konsisten dan berada di kisaran 89%–91%. Jarak antara kedua kurva relatif kecil dan bahkan pada beberapa epoch validation accuracy lebih tinggi daripada training accuracy. Hal ini menunjukkan bahwa model CNN memiliki kemampuan generalisasi yang baik dan tidak mengalami overfitting yang signifikan. Selain itu, fluktuasi

validation accuracy relatif kecil, menandakan proses pembelajaran yang stabil. Dengan bertambahnya epoch, model masih mampu meningkatkan performanya tanpa menunjukkan tanda-tanda penurunan akurasi validation.



Gambar 7. Curve CNN

7. Penggunaan AI

Dalam pengerjaan tugas ini, kami menggunakan Artificial Intelligence berupa ChatGPT sebagai alat bantu pembelajaran. ChatGPT digunakan untuk membantu menghasilkan template kode awal, memberikan bantuan debugging, Secara khusus, ChatGPT digunakan pada beberapa bagian berikut:

- Membantu menyusun template kode untuk implementasi Random Forest, MLP, dan CNN.
- Membantu debugging kode, seperti perbaikan error path dataset, penyesuaian arsitektur model, dan visualisasi hasil.
- Membantu menyusun template EDA (distribusi kelas, visualisasi gambar, correlation matrix).
- Membantu menyusun template evaluasi, seperti learning curve, confusion matrix, dan grafik perbandingan model.