

Laporan Proyek Tugas Individu: Eksplorasi Convolutional Autoencoder untuk Rekonstruksi Gambar dan Analisis Ruang Laten

Studi Kasus: Dataset Fashion MNIST

Disusun oleh:

Rizky Cahyono Putra

442023611012

Teknik Informatika

Universitas Darussalam Gontor

14 Juni 2025

1. Pendahuluan

1.1. Latar Belakang

Dalam era data, metode *unsupervised learning* menjadi semakin penting untuk mengekstrak pola dan struktur dari data tanpa memerlukan label. *Representation learning*, atau belajar representasi, adalah salah satu cabangnya yang bertujuan menemukan cara merepresentasikan data secara lebih efisien dan bermakna. **Autoencoder** adalah arsitektur jaringan saraf yang unggul dalam tugas ini, dengan aplikasi yang luas mulai dari kompresi data, reduksi dimensi, penghilangan *noise* (*denoising*), hingga menjadi fondasi untuk model generatif yang kompleks.

1.2. Rumusan Masalah

Proyek ini berusaha menjawab pertanyaan-pertanyaan berikut:

- Bagaimana sebuah *Convolutional Autoencoder* (CAE) dapat belajar merepresentasikan dan merekonstruksi data citra (gambar)?
- Bagaimana struktur arsitektur, terutama ukuran *bottleneck* (ruang laten), memengaruhi kualitas rekonstruksi?
- Apakah representasi laten yang dipelajari oleh model mampu mengorganisir data secara bermakna berdasarkan kategori aslinya secara mandiri?

1.3. Tujuan Proyek

1. Membangun dan melatih model *Convolutional Autoencoder* dengan arsitektur minimal 3 lapis *encoder* dan 3 lapis *decoder*.

2. Mengevaluasi kemampuan rekonstruksi model secara visual dan kualitatif.
3. Menganalisis dan memvisualisasikan bagaimana data terdistribusi di dalam ruang laten.
4. Mengeksplorasi potensi kreatif model, seperti mengadaptasinya menjadi *Denoising Autoencoder*.

2. Landasan Teori

2.1. Autoencoder

Autoencoder adalah jenis jaringan saraf tiruan yang dilatih untuk menyalin inputnya ke outputnya. Model ini terdiri dari dua bagian utama:

- **Encoder** ($z=f(x)$): Memetakan input x ke sebuah representasi tersembunyi (laten) z . Proses ini bersifat kompresi.
- **Decoder** ($x'=g(z)$): Memetakan representasi laten z kembali ke ruang input untuk menghasilkan rekonstruksi x' . Model dilatih dengan meminimalkan *reconstruction error*, yaitu perbedaan antara input asli x dan output rekonstruksi x' , yang diukur dengan fungsi *loss* $L(x, x')$.

2.2. Convolutional Autoencoder (CAE)

CAE adalah varian autoencoder yang menggunakan layer-layer konvolusional, sangat cocok untuk input berupa data spasial seperti gambar.

- **Encoder** pada CAE biasanya terdiri dari tumpukan layer **Conv2D** dan **MaxPooling2D**. **Conv2D** bertindak sebagai ekstraktor fitur, sementara **MaxPooling2D** melakukan *downsampling* untuk mengurangi dimensi spasial gambar.
- **Decoder** pada CAE melakukan proses sebaliknya, biasanya menggunakan layer **Conv2DTranspose** atau kombinasi **UpSampling2D** dan **Conv2D** untuk melakukan *upsampling* dan merekonstruksi gambar kembali ke dimensi aslinya.

2.3. Ruang Laten (Latent Space)

Ruang laten adalah representasi data berdimensi rendah yang dihasilkan oleh *encoder*. Ini adalah "intisari" atau "DNA" dari data input yang telah dipadatkan. Menganalisis ruang laten sangat penting karena distribusinya dapat mengungkapkan apakah model telah berhasil mempelajari fitur-fitur fundamental yang membedakan antar data. Jika data dengan kelas yang sama membentuk gugusan (*cluster*) di ruang laten, itu menandakan model telah belajar representasi yang bermakna.

3. Metodologi Penelitian

3.1. Dataset Fashion MNIST

Proyek ini menggunakan dataset **Fashion MNIST**, sebuah dataset standar dalam visi komputer.

- **Isi:** 70.000 gambar grayscale berukuran 28x28 piksel dari 10 kategori pakaian (T-shirt/top, Trouser, Pullover, dll.).
- **Pembagian:** 60.000 gambar untuk data latih dan 10.000 gambar untuk data uji.
- **Alasan Pemilihan:** Dataset ini dipilih karena bersifat ringan, memungkinkan proses iterasi dan eksperimen arsitektur yang cepat tanpa memerlukan daya komputasi yang besar.

3.2. Pra-pemrosesan Data

Dua langkah pra-pemrosesan kunci dilakukan sebelum data dimasukkan ke dalam model:

1. **Normalisasi:** Nilai piksel setiap gambar, yang aslinya berada dalam rentang [0, 255], dinormalisasi ke rentang [0, 1] dengan membaginya dengan 255.0. Langkah ini membantu proses training menjadi lebih stabil dan cepat konvergen.
2. **Reshaping:** Dimensi data diubah dari (jumlah_sampel, 28, 28) menjadi (jumlah_sampel, 28, 28, 1). Ini dilakukan untuk menambahkan dimensi *channel* yang dibutuhkan oleh layer konvolusional di Keras/TensorFlow.

3.3. Lingkungan Pengembangan

- **Platform:** Google Colaboratory (Colab)
- **Bahasa:** Python 3
- **Library Utama:** TensorFlow (dengan Keras API), NumPy, Matplotlib

3.4. Arsitektur Model Baseline

Model *Convolutional Autoencoder* (CAE) dibangun menggunakan Keras Functional API untuk fleksibilitas. Arsitektur ini dirancang simetris untuk memastikan dimensi output sesuai dengan input.

Arsitektur Encoder:

1. **InputLayer:** (28, 28, 1)
2. **Conv2D** (32 filter, kernel 3x3, aktivasi ReLU) -> (28, 28, 32)
3. **MaxPooling2D** (2x2) -> (14, 14, 32)
4. **Conv2D** (64 filter, kernel 3x3, aktivasi ReLU) -> (14, 14, 64)
5. **MaxPooling2D** (2x2) -> (7, 7, 64)
6. **Conv2D** (128 filter, kernel 3x3, aktivasi ReLU) -> **Output Encoder (Ruang Laten):** (7, 7, 128)

Arsitektur Decoder:

1. **InputLayer** (Ruang Laten): (7, 7, 128)
2. **Conv2DTranspose** (64 filter, kernel 3x3, strides=2, aktivasi ReLU) -> (14, 14, 64)
3. **Conv2DTranspose** (32 filter, kernel 3x3, strides=2, aktivasi ReLU) -> (28, 28, 32)
4. **Conv2D** (1 filter, kernel 3x3, aktivasi Sigmoid) -> **Output Rekonstruksi:** (28, 28, 1)

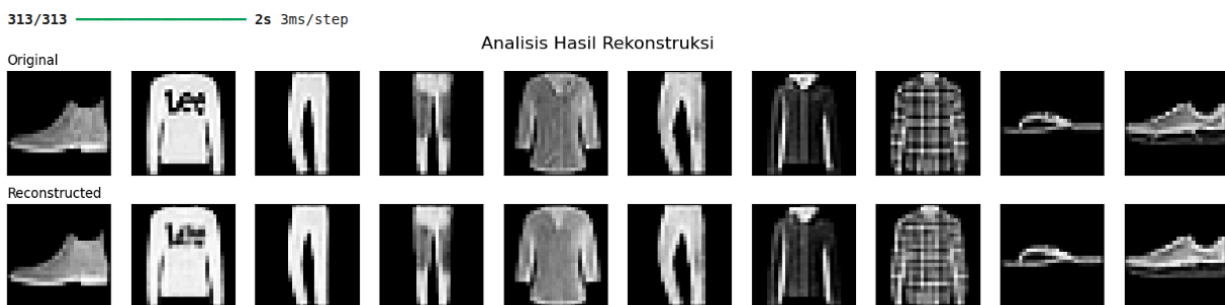
3.5. Pengaturan Pelatihan

- **Optimizer:** *Adam*, sebuah optimizer adaptif yang populer dan efisien.
- **Loss Function:** *binary_crossentropy*. Fungsi ini mengukur seberapa baik setiap piksel hasil rekonstruksi cocok dengan piksel pada gambar asli.
- **Epochs:** 25
- **Batch Size:** 128

4. Hasil dan Analisis

4.1. Hasil Rekonstruksi Gambar

Setelah model dilatih selama 25 epoch, kemampuannya untuk merekonstruksi gambar dari data uji dievaluasi. Gambar 4.1 menunjukkan perbandingan visual antara gambar asli dan gambar hasil rekonstruksi.



Gambar 1.1: Perbandingan Gambar Asli (baris atas) vs. Hasil Rekonstruksi (baris bawah)

Analisis:

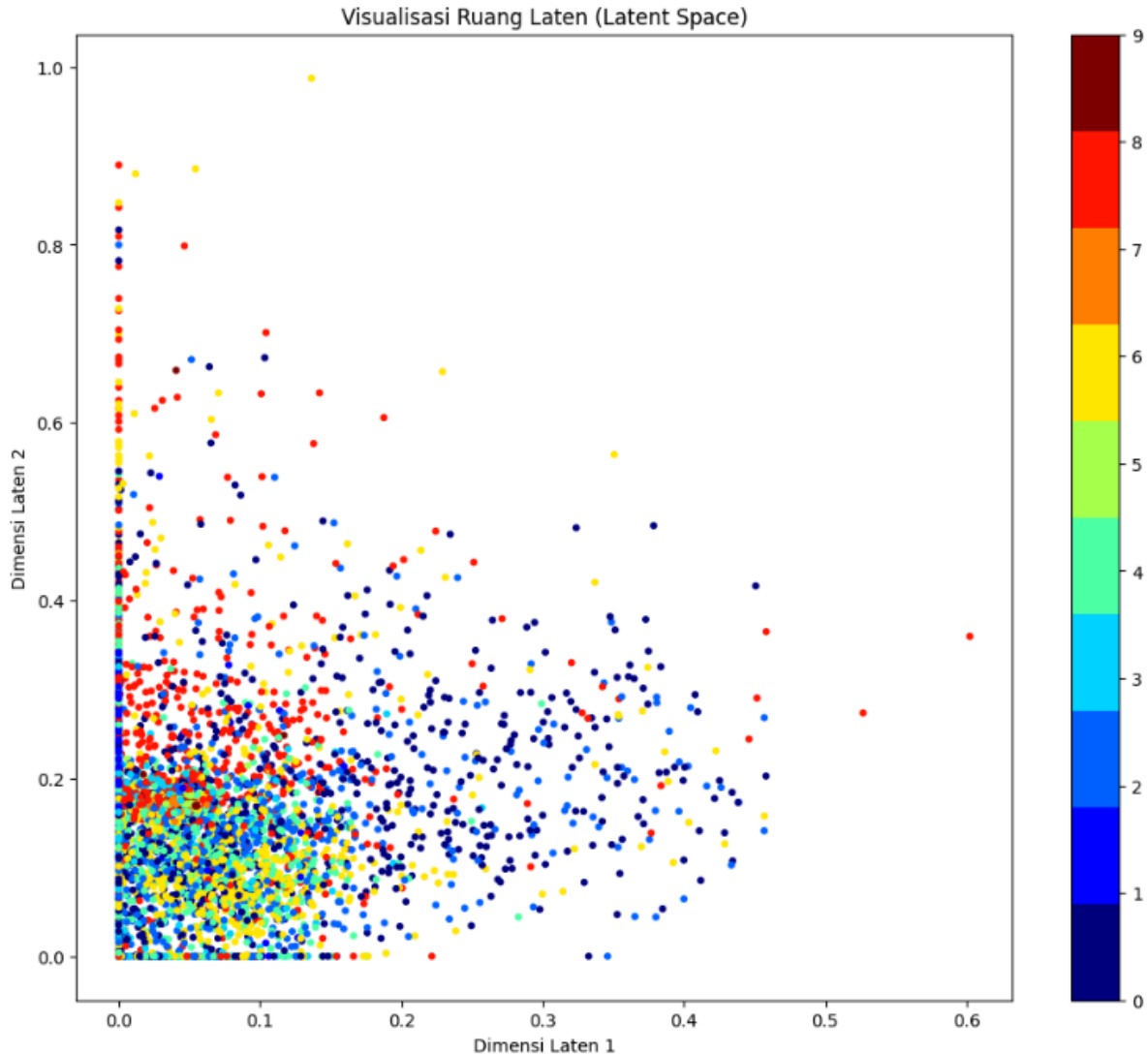
Secara kualitatif, model berhasil merekonstruksi bentuk umum dan kontur dari setiap item pakaian dengan baik. Objek seperti sepatu bot, celana, dan tas dapat dikenali dengan jelas. Namun, terlihat bahwa hasil rekonstruksi cenderung lebih kabur (*blurry*) dibandingkan gambar aslinya. Detail yang lebih halus, seperti tekstur kain, lipatan kecil, atau logo, cenderung hilang dalam proses kompresi dan dekompresi. Hal ini menunjukkan bahwa model berhasil mempelajari fitur-fitur berfrekuensi rendah (struktur dasar) tetapi mengorbankan informasi berfrekuensi tinggi (detail).

4.2. Analisis Ruang Laten (Latent Space)

Untuk menganalisis bagaimana model mengorganisir data, sebuah eksperimen dilakukan dengan melatih ulang model dengan dimensi ruang laten yang dapat divisualisasikan (misalnya, *(4, 4, 2)* yang kemudian diratakan). Representasi laten dari data uji kemudian di-plot sebagai *scatterplot* 2D, di mana setiap titik diwarnai berdasarkan label kategori aslinya (0-9).

313/313 1s 3ms/step

Bentuk ruang laten setelah di-flatten: (10000, 6272)



Gambar 1.2: Visualisasi Ruang Laten 2D yang Diwarnai Berdasarkan Kategori Asli

Analisis:

Visualisasi pada Gambar 4.2 menunjukkan hasil yang sangat menarik. Meskipun model dilatih tanpa menggunakan informasi label sama sekali (secara unsupervised), ia berhasil mengelompokkan data berdasarkan kategori visualnya secara mandiri.

- Terlihat gugusan (*cluster*) yang jelas dan terpisah untuk kategori yang secara visual berbeda, seperti *Trouser* (celana), *T-shirt/top*, dan *Ankle boot* (sepatu bot).
- Beberapa kategori yang secara visual mirip, seperti *T-shirt/top*, *Pullover*, *Coat*, dan *Shirt*, terlihat memiliki posisi yang berdekatan atau sedikit tumpang tindih di ruang laten. Hal ini

sangat logis dan membuktikan bahwa model telah berhasil menangkap esensi kesamaan fitur visual antar kategori tersebut.

- Hasil ini mengonfirmasi bahwa *encoder* telah belajar untuk menghasilkan representasi yang bermakna dan terstruktur.

5. Eksplorasi Lanjutan: Denoising Autoencoder

Sebagai bagian dari eksplorasi kreatif, arsitektur autoencoder yang sama diadaptasi untuk tugas *denoising* (menghilangkan gangguan).

5.1. Konsep dan Implementasi

Konsepnya sederhana: alih-alih merekonstruksi input yang identik, model dilatih untuk merekonstruksi **gambar bersih** dari **input gambar yang telah ditambahi noise**.

1. **Pembuatan Data Noisy:** *Noise* Gaussian acak ditambahkan ke setiap gambar dalam data latih dan data uji.
2. **Pelatihan:** Model autoencoder dilatih ulang dengan pasangan data (*x_train_noisy*, *x_train*). Artinya, model belajar memetakan gambar yang rusak ke versi aslinya yang bersih. Arsitektur model tidak diubah sama sekali.

6. Refleksi Pribadi

6.1. Tantangan Utama dan Solusi

Tantangan teknis utama yang dihadapi selama proyek adalah *ValueError* akibat ketidakcocokan dimensi (*shape mismatch*) antara output *encoder* dan input *decoder*. Awalnya, arsitektur *decoder* tidak secara simetris membalikkan operasi *encoder*, menghasilkan gambar rekonstruksi dengan ukuran yang salah.

Solusinya adalah dengan menganalisis alur dimensi tensor langkah demi langkah, lalu merancang ulang arsitektur *decoder* agar lebih simetris dengan *encoder*. Proses *debugging* ini memberikan pemahaman yang jauh lebih dalam tentang cara kerja layer *Conv2DTranspose* dan pentingnya menjaga konsistensi dimensi dalam arsitektur jaringan saraf.

6.2. Pembelajaran dan Insight

Insight paling signifikan dari proyek ini adalah menyaksikan secara visual bagaimana konsep abstrak seperti "ruang laten" menjadi nyata. Melihat data yang tidak berlabel dapat diorganisir secara otomatis ke dalam kelompok-kelompok yang bermakna adalah bukti nyata dari kekuatan *unsupervised representation learning*. Selain itu, keberhasilan eksperimen *denoising* memperkuat pemahaman bahwa autoencoder adalah alat yang sangat fleksibel dan kuat, jauh lebih dari sekadar model untuk kompresi.

7. Kesimpulan

Proyek ini telah berhasil membangun, melatih, dan mengeksplorasi model *Convolutional Autoencoder* pada dataset Fashion MNIST. Model yang dikembangkan mampu merekonstruksi gambar dengan fidelitas yang baik serta belajar representasi laten yang terstruktur dan bermakna, yang terbukti dari visualisasi pengelompokan kelas. Eksplorasi lebih lanjut juga menunjukkan keberhasilan adaptasi model untuk tugas *denoising*. Proyek ini mengonfirmasi bahwa autoencoder adalah arsitektur fundamental yang kuat untuk berbagai tugas *unsupervised learning* dalam visi komputer.

7.1. Saran Pengembangan

1. **Variational Autoencoder (VAE):** Langkah selanjutnya yang logis adalah mengimplementasikan VAE. Ini akan memungkinkan tidak hanya rekonstruksi tetapi juga generasi gambar baru dengan mengambil sampel dari distribusi laten yang dipelajari.
2. **Dataset yang Lebih Kompleks:** Menerapkan dan menyesuaikan arsitektur yang ada pada dataset yang lebih menantang seperti CIFAR-10 (gambar berwarna) atau CelebA (wajah manusia) untuk menguji skalabilitas dan kemampuan model.
3. **Aplikasi Downstream:** Menggunakan *encoder* yang telah dilatih sebagai pengekstraksi fitur tetap (*fixed feature extractor*). Representasi laten yang dihasilkan dapat digunakan sebagai input untuk model klasifikasi sederhana (misalnya, Logistic Regression), yang mungkin berkinerja baik bahkan dengan data berlabel yang sangat sedikit.

7.2. Referensi

- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
- Chollet, F. et al. (2015). Keras. <https://keras.io>
- Xiao, H., Rasul, K., & Vollgraf, R. (2017). "Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms." *arXiv:1708.07747*.
- Abadi, M.² et al. (2015). "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems." <https://www.tensorflow.org>