BAB II

LANDASAN TEORI

2.1 Deteksi & Segmentasi Anomali (*unsupervised*)

Deteksi anomali merupakan teknik kecerdasan buatan untuk menemukan pola asing atau pola tidak normal yang berbeda dari pola yang sudah dipelajari pada fase *training* (Samariya & Thakkar, 2021). *Unsupervised* yang memiliki arti pembelajaran tak terawasi, merupakan pendekatan yang mempelajari sesuatu hanya dengan sampel data normal tanpa membutuhkan sampel data yang telah diberi label sebelumnya atau disebut dengan *unlabeled data* (Aqeel dkk., 2025). Model deteksi anomali dapat bekerja secara unsupervised atau satu kelas (*one-class*), yaitu hanya memodelkan citra normal sebagai sampel belajar dan menang*gap* contoh yang tidak sesuai sebagai anomali, sehingga setiap citra abnormal menghasilkan skor kesalahan tinggi saat diuji (Bandyopadhyay, 2021) yang kemudian menjadi segmentasi anomali.

Segmentasi anomali berkembang dari sekadar deteksi pada level gambar (*Image-level detection*) menjadi lokalisasi pada level piksel (*Pixel-Level localization*) (Bauer dkk., 2024). Tugas dari segmentasi anomali *unsupervised* yaitu menghasilkan peta anomali (*anomaly map*) yang presisi di mana intensitas setiap piksel berkorelasi dengan derajat penyimpangannya dari normalitas (Bauer dkk., 2024). Segmentasi anomali biasanya dicapai melalui analisis residu rekonstruksi, dengan output berupa *masking* piksel demi piksel yang menyoroti area rusak pada skala lokal, berupa koordinat atau wilayah spesifik (Bauer dkk., 2024; Hassanaly dkk., 2024).

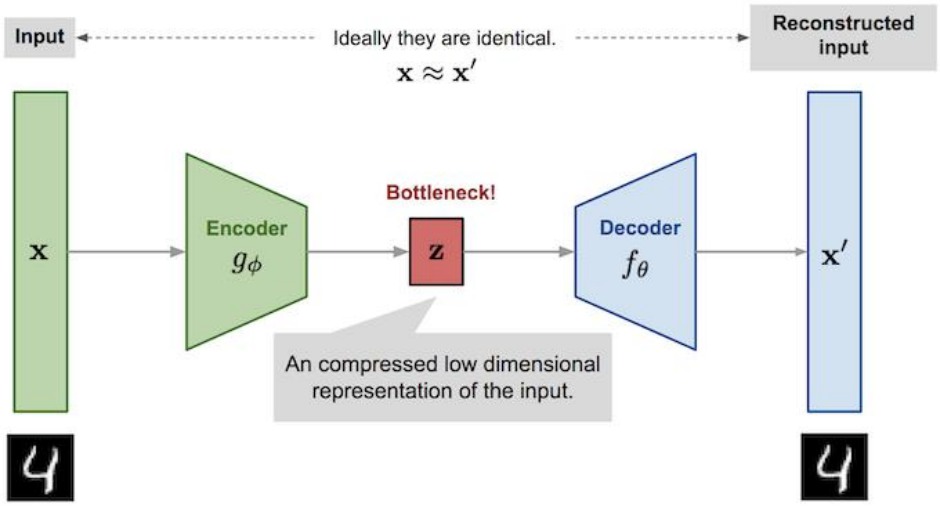
2.2 *Autoencoder*

2.2.1 Konsep Dasar *Autoencoder* (AE)

Autoencoder secara teknis dikategorikan sebagai model *Deep Learning* karena memiliki tiga karakteristik utama dari sebuah model Jaringan Saraf Tiruan (*Artificial Neural Network*) di mana arsitektur jaringan saraf tiruan yang terstruktur secara fisik, meliputi lapisan *input*, *encoder*, *bottleneck*, *decoder*, hingga *output*. Statusnya sebagai model diperkuat oleh sifatnya yang dapat dilatih (*trainable*), di mana jutaan parameter bobot dan bias diperbarui melalui algoritma *backpropagation*. Dan proses pembelajaran ini dipandu secara matematis oleh fungsi objektif untuk meminimalkan *reconstruction loss*, yakni selisih antara data *input* asli dengan hasil rekonstruksinya.

Konsep *Autoencoder* pertama kali diperkenalkan pada tahun 1986 oleh Rumelhart, Hinton, dan Williams dari kelompok *Parallel Distributed Processing* (PDP) dalam makalah yang mendemonstrasikan kemampuan jaringan saraf untuk mengompresi dan merekonstruksi data melalui *Bottleneck* (Rumelhart dkk., 1986). Istilah *Autoencoder* diresmikan secara luas oleh Hinton dan Zemel pada tahun 1993, menggantikan sebutan *Auto-associative Neural Network* (Hinton & Zemel, 1993).

Autoencoder (AE) merupakan arsitektur jaringan saraf tiruan yang dirancang untuk mempelajari representasi identitas dari data *input* melalui proses kompresi dan dekompresi data (Jayeprokash & Gonski, 2025). Struktur fundamental dari sebuah *Autoencoder* terdiri dari dua komponen utama yang simetris yaitu *Encoder* dan *Decoder* (Enttsel dkk., 2025).



Gambar 2.1 Arsitektur *Autoencoder* (Allahyari, 2023)

Gambar 2.1 menggambarkan arsitektur dasar model *Autoencoder* dimana *Encoder* berfungsi untuk memetakan data *input* berdimensi tinggi x ke dalam representasi laten berdimensi rendah z yang sering disebut sebagai *bottleneck* atau kode laten, dirumuskan sebagai $z = g\varphi(x)$. Representasi laten di desain untuk menangkap fitur-fitur paling esensial dan intrinsik dari data, menghilangkan redundansi dan *noise* yang tidak relevan. *Decoder* bertugas merekonstruksi kembali data *input* dari representasi laten tersebut menjadi x' dengan formulasi $x' = f\theta(g\varphi(x))$, dimana tujuannya adalah meminimalkan perbedaan antara *input* asli x dan rekonstruksi x' (Allahyari, 2023).

Landasan teori penggunaan AE untuk deteksi anomali yaitu AE memiliki *bottleneck* sebagai kompresi informasi. Saat model dihadapkan pada data anomali di fase pengujian, *Encoder* akan gagal memetakan fitur abnormal tersebut ke dalam *manifold laten* yang valid, dan *Decoder* akan gagal merekonstruksi anomali secara akurat, sehingga tercipta kesalahan rekonstruksi yang krusial (Bauer dkk., 2024).

2.2.2 *Reconstruction Error* sebagai Indikator Anomali

Reconstruction Error atau Galat Rekonstruksi adalah angka yang mengukur seberapa banyak informasi yang hilang ketika data dikompresi dan dikembalikan kedalam bentuk aslinya, dengan menghitung selisih atau perbedaan antara data asli (*input*) dengan data yang telah di rekonstruksi (*reconstructed output*) oleh sebuah model *machine learning* (Kamat dkk., 2024). Secara matematis, skor anomali $A(x)$ untuk *input* x didefinisikan sebagai jarak antara *input* asli dan hasil rekonstruksi $\hat{x}$ (Bouman & Heskes, 2025):

$$A(x) = L(x, \hat{x}) \tag{2.1}$$

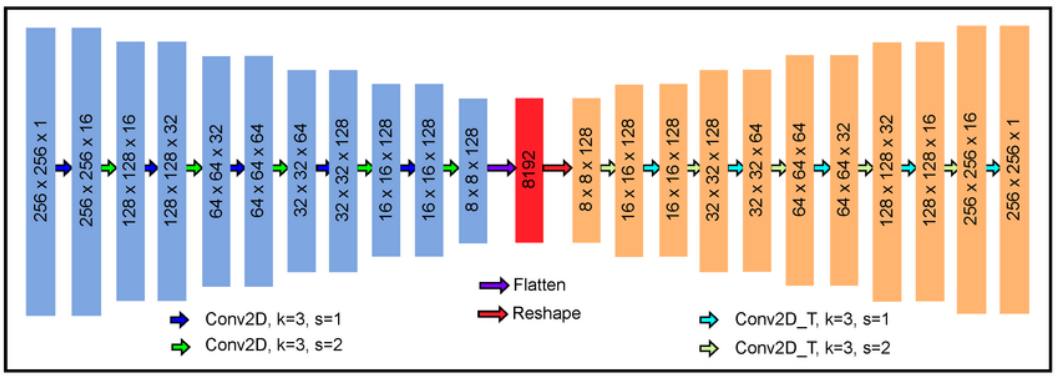
di mana formula ini sering diimplementasikan sebagai selisih absolut atau kuadrat antara *input* dan *Output*. Metrik ini digunakan dengan asumsi bahwa anomali lebih sulit direkonstruksi dibandingkan data normal karena tidak memiliki representasi yang kuat dalam ruang laten model, sehingga menghasilkan *loss* yang lebih tinggi (Bouman & Heskes, 2025).

Penelitian menunjukkan bahwa distribusi *reconstruction error* pada data normal cenderung terpusat di nilai rendah sekitar rata-rata μ, sementara anomali berada di nilai tinggi atau melebihi ambang batas tertentu misalnya $\mu + 2\sigma$ (Tuli dkk., 2022). Efektivitas *reconstruction error* sangat bergantung pada kemampuan model untuk gagal melakukan rekonstruksi anomali yaitu jika model terlalu hafal dengan data pelatihan (*overfitting*) atau mempelajari pola dasar dari data normal (*identity mapping*), *reconstruction error* untuk anomali akan menjadi rendah dan menyebabkan *False Negatives* (Zavrtanik dkk., 2021).

2.2.3 *Convolutional Autoencoder* (CAE)

Convolutional Autoencoder (CAE) adalah jenis jaringan saraf tiruan (*Neural Network*) yang menggabungkan struktur *Convolutional Neural Networks* (CNN) dengan arsitektur *Autoencoder* (Chen & Guo, 2023). Berbeda dengan *Autoencoder* tradisional yang menggunakan lapisan *fully-connected*, CAE memanfaatkan lapisan konvolusi (*convolutional layers*) untuk menangkap pola spasial pada data, menjadikannya sangat efektif untuk pemrosesan gambar dan sinyal (Pintelas dkk., 2021; Solovyeva & Abdullah, 2022). Tujuan utamanya adalah mempelajari representasi data yang lebih ringkas (kompresi) di *latent space* dan kemudian merekonstruksi kembali data tersebut seakurat mungkin (Jayeprokash & Gonski, 2025; Yeh dkk., 2025).

Dalam arsitektur CAE, operasi perkalian matriks standar digantikan oleh operasi konvolusi, yang memungkinkan model untuk mempelajari fitur-fitur lokal yang konstan terhadap translasi, seperti tepi, tekstur, dan bentuk. Contoh arsitektur *Convolutional Autoencoder* digambarkan pada gambar 2.2.



Gambar 2.2 Contoh Arsitektur *Convolutional Autoencoder*

(Jayeprokash & Gonski, 2025)

1. *Encoder*: Terdiri dari serangkaian lapisan konvolusi yang diikuti oleh fungsi aktivasi non-linear (seperti *ReLU* atau *Leaky ReLU*) dan operasi *pooling* atau *Strided convolution* untuk mereduksi dimensi spasial secara bertahap. Proses ini mengekstraksi hierarki fitur visual dari tingkat rendah ke tinggi, menghasilkan peta fitur (*feature Maps*) yang kaya informasi namun padat secara spasial (Jayeprokash & Gonski, 2025).
2. *Bottleneck*: Berbeda dengan AE standar yang meratakan data menjadi vektor tunggal, CAE sering kali mempertahankan struktur tensor 3D di *Bottleneck* atau menggunakan lapisan *flatten* yang kemudian diproyeksikan ke ruang laten dimensi rendah, memaksa kompresi informasi semantik yang tinggi (Jayeprokash & Gonski, 2025).
3. *Decoder*: Menggunakan lapisan *transposed convolution* (atau *up-sampling* diikuti konvolusi) untuk memulihkan dimensi spasial asli dari representasi laten. Tujuannya adalah merekonstruksi detail visual *input* berdasarkan fitur abstrak yang tersimpan di *Bottleneck* (Jayeprokash & Gonski, 2025).

2.3 Mekanisme Adaptif & *Sparsity*

Konsep Adaptif dalam konteks penelitian ini merujuk pada kemampuan model jaringan saraf tiruan untuk mengatur dan menyesuaikan kapasitas representasi ruang laten (*latent space*) secara dinamis berdasarkan kompleksitas citra masukan, tanpa memerlukan penentuan manual dalam menentukan ukuran dimensi (Stępień dkk., 2026). Sifat adaptif diimplementasikan melalui modifikasi struktural pada area *bottleneck Autoencoder* (Stępień dkk., 2026). Sistem tidak lagi memaksa data untuk muat ke dalam dimensi yang kaku, melainkan menggunakan dimensi laten

dasar yang besar, lalu secara cerdas dan selektif menyeleksi fitur mana yang relevan dengan pola normalitas (Stępień dkk., 2026). Sifat adaptif ini dicapai melalui perpaduan dua mekanisme utama yaitu *Gating Mechanism* yang bertindak sebagai filter dinamis, dan *Sparsity* (regularisasi) yang meregulasi filter tersebut agar mematikan aliran informasi yang tidak relevan.

2.3.1 *Gating Mechanism*

Gating adalah bentuk khusus dari atensi yang bertindak sebagai gerbang logis yang dapat dilatih untuk mengontrol aliran informasi antar lapisan, menyesuaikan tingkat kepentingan fitur *input* secara dinamis (Sellam dkk., 2025). *Gate* biasanya menggunakan fungsi aktivasi *Sigmoid* untuk menghasilkan vektor koefisien bernilai antara 0 dan 1, yang kemudian dikalikan secara elemen per elemen (*element-wise*). Secara matematis, mekanisme *Gating* dalam arsitektur *Autoencoder* sebagai fungsi pembelajaran bobot relevansi diformulasikan:

$$Z_{adaptive} = Z \odot Mask \tag{2.2}$$

Di mana Z adalah representasi fitur laten (*bottleneck*), W dan b adalah parameter bobot dan bias yang dapat dilatih, σ adalah fungsi aktivasi *Sigmoid* yang membatasi nilai keluaran (*Mask*) antara 0 hingga 1, dan operasi $\odot$ menunjukkan perkalian elemen-bijak (*element-wise multiplication*) (Preetham, 2024). Pendekatan ini memungkinkan model untuk secara selektif membiarkan fitur yang esensial untuk lanjut ketahap *Decoder* dan menekan fitur yang tidak relevan seperti *noise* atau anomali sebelum proses rekonstruksi.

2.3.2 Regularisasi L1 dan Induksi *Sparsity*

L1 Regularization dalam konteks standar sering digunakan untuk memberikan penalti pada nilai absolut bobot model (Deepa dkk., 2025). Pada arsitektur berbasis *Gating*, konsep L1 diadaptasi menjadi *Activity Regularization* atau *Sparse Activation* (Kaliyapillai dkk., 2023). Karakteristik utama dari pendekatan ini adalah kemampuannya untuk melakukan induksi *Sparsity* secara langsung pada hasil keluaran *Gate* nerupa aktivasi, di mana model dipaksa untuk menekan nilai *Gate* yang kurang relevan hingga mendekati angka nol (Sheng dkk., 2025). Hal ini membuat model hanya menyalakan fitur laten yang paling relevan untuk merekonstruksi data normal.

Secara matematis, formulasi fungsi kerugian (*loss function*) gabungan direpresentasikan sebagai berikut:

$$L_{total} = L_{reconstruction} + \lambda \frac{1}{N} \sum_{i=1}^{N} (Mask_i) \tag{2.3}$$

Nilai *Mask* yang dihasilkan oleh fungsi aktivasi *Sigmoid* selalu bernilai positif pada rentang 0 hingga 1, sehingga komputasi rata-rata aritmatika pada keluaran *Gate* tersebut secara matematis ekuivalen dengan penerapan norma L_1 pada vektor aktivasi (Bricken dkk., 2023). Penalti λ pada persamaan tersebut berfungsi sebagai regulator yang secara dinamis menyeleksi fitur laten esensial di *bottleneck*, memastikan bahwa aliran informasi yang merepresentasikan anomali diberhentikan dan tidak dilanjutkan ke tahap *Decoder* di mana proses rekonstruksi terjadi (Ghosh & Kirby, 2023a).

2.4 SSIM sebagai *Loss Function* (DSSIM)

Structural Similarity Index Measure (SSIM) adalah metode pengukuran kualitas citra yang dirancang untuk dirancang untuk mengukur kemiripan antara dua citra berdasarkan cara kerja mata manusia (*Perceptual Image Quality*) (Ma dkk., 2025). Berbeda dengan metrik tradisional seperti *Mean Squared Error* (MSE) atau *Peak Signal-to-Noise Ratio* (PSNR) yang hanya menghitung selisih nilai piksel secara absolut, SSIM beroperasi dengan premis bahwa sistem visual manusia sangat sensitif terhadap perubahan struktur pola spasial pada objek, bukan sekadar perubahan intensitas cahaya (Najjar, 2024). Nilai SSIM berkisar antara -1 hingga 1, di mana nilai 1 menunjukkan kemiripan sempurna dengan citra referensi.

Untuk mengatasi keterbatasan MSE yang hanya mengukur perbedaan intensitas piksel absolut, *Structural Similarity Index Measure* (SSIM) digunakan sebagai metrik yang memodelkan persepsi perubahan informasi struktural (Wu dkk., 2024). Melalui pendekatan ini, SSIM mampu memodelkan persepsi perubahan informasi struktural sehingga lebih relevan dalam mendeteksi anomali visual yang bermakna.

Pendekatan ini juga diterapkan pada DSSIM yang merupakan turunan dari SSIM dalam rekonstruksi anomali (Sabri, 2022). Mengacu pada konsep dasar yang dikembangkan oleh (Wang dkk., 2004) serta analisis estimasi statistik terbaru oleh (Osorio dkk., 2022), SSIM mendekomposisi kesamaan antara dua citra menjadi tiga komponen independen yang merefleksikan cara mata manusia memproses informasi visual yaitu *Luminance* (Kecerahan), *Contrast* (kontras), dan *Structure* (Struktur). Ketiga komponen tersebut didefinisikan secara matematis sebagai berikut:

1. ***Luminance*** **(** $l(x,y)$ **)**

   Komponen ini mengukur kesamaan rata-rata intensitas piksel (μ_x, μ_y) untuk menormalisasi dampak perubahan pencahayaan global:

$$l(x,y) = \frac{2\mu_x 2\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \tag{2.4}$$

2. ***Contrast*** **(** $c(x,y)$ **)**

   Komponen ini mengukur kesamaan standar deviasi (σ_x, σ_y), yang merepresentasikan variasi sinyal atau tekstur dalam citra:

$$c(x,y) = \frac{2\sigma_x 2\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \tag{2.5}$$

3. ***Structure*** **(** $s(x,y)$ **)**

   Komponen ini mengukur korelasi struktural (σ_{xy}) antara dua citra setelah dilakukan normalisasi luminans dan kontras, guna menangkap kesamaan pola tepi dan bentuk objek:

$$s(x,y) = \frac{\sigma_{xy} + C_3}{\sigma_x \sigma_y + C_3} \tag{2.6}$$

Nilai SSIM keseluruhan dihitung sebagai kombinasi perkalian dari ketiga komponen tersebut, biasanya dengan bobot eksponen $\alpha = \beta = \gamma = 1$ untuk memberikan kontribusi yang seimbang (Baker dkk., 2024). Secara matematis, definisi ini dinyatakan sebagai:

$$SSIM(x,y) = [l(x,y)]^{\alpha}.[c(x,y)]^{\beta}.[s(x,y)]^{\gamma} \quad (2.7)$$

Secara umum, nilai SSIM berkisar dari -1 hingga 1, di mana 1 menunjukkan kesamaan sempurna, yang menjadi indikator krusial dalam menentukan seberapa baik model mampu merekonstruksi fitur normal pada dataset permukaan.

Penggunaan SSIM secara langsung sebagai fungsi tujuan memiliki kendala mendasar karena adanya perbedaan orientasi optimasi, tujuan pelatihan model adalah meminimalkan nilai *Loss* (*Error*), sedangkan metrik SSIM bertujuan untuk dimaksimalkan hingga mendekati nilai 1. Menjembatani hal tersebut, diterapkan konsep *Structural Dissimilarity* (DSSIM) yang membalik logika SSIM melalui persamaan:

$$DSSIM = \frac{1-SSIM(p,q)}{2} \quad (2.8)$$

(Sabri, 2022)

Penggunaan SSIM sebagai fungsi dalam pelatihan *Autoencoder* telah terbukti meningkatkan kinerja deteksi anomali video dan medis secara signifikan (Fan dkk., 2025; Shiferaw & Yao, 2024). Model yang dilatih dengan SSIM lebih sensitif terhadap perubahan bentuk dan batas objek (anomali struktural) dan lebih toleran terhadap variasi pencahayaan global yang sering memicu alarm palsu pada model berbasis MSE (Fan dkk., 2025; Zavrtanik dkk., 2021).

2.5 *Mean Absolute Error* (MAE) dalam Ekstraksi Peta Anomali

Mean Absolut Error (MAE) adalah metrik yang digunakan untuk mengukur rata-rata besarnya kesalahan (*error*) antara nilai prediksi dengan nilai yang sebenarnya., yaitu mengukur seberapa jauh tebakan model dengan target aslinya

tanpa menghiraukan area kesalahan yang terlalu tinggi atau terlalu rendah. Pada konteks ekstraksi peta anomali, MAE berfungsi untuk memproduksi intensitas anomali tingkat piksel (*Pixel-Level Anomaly Score Map*) dan mengukur besaran selisih absolut secara langsung antara intensitas piksel pada citra masukan asli dengan intensitas piksel pada citra hasil rekonstruksi (Kohler dkk., 2025). Secara matematis, ekstraksi peta galat menggunakan jarak absolut pada koordinat spasial $|(i,j)|$ diformulasikan sebagai berikut (Xu dkk., 2021):

$$E(i,j) = \left|X(i,j) - \hat{X}(i,j)\right| \tag{2.9}$$

Keterangan:

- $E(i,j)$ = Nilai galat (*error*) pada koordinat piksel (i,j) yang membentuk peta anomali spasial.
- $X(i,j)$ = Intensitas piksel citra asli (masukan).
- $\hat{X}(i,j)$ = Intensitas piksel citra hasil rekonstruksi (*Output* dari *Autoencoder*).

Penggunaan MAE pada tahap evaluasi didasarkan pada asumsi fundamental *Autoencoder* dalam deteksi anomali yaitu karena model hanya dilatih menggunakan citra normal, model akan cenderung merekonstruksi area yang mengandung cacat seperti robekan pada *Zipper* atau kawat putus pada *Grid* menjadi pola normal (Fang dkk., 2024). Ketidakmampuan model untuk memulihkan pola abnormal tersebut akan menghasilkan selisih nilai intensitas absolut yang sangat tinggi tepat pada lokasi spasial piksel yang rusak (Fang dkk., 2024). Hasil komputasi dari matriks selisih absolut ini akan menghasilkan sebuah peta skor anomali kontinu (*continuous error map*) (Fang dkk., 2024).

Pada peta ini, piksel dengan nilai MAE yang mendekati nol merepresentasikan area normal yang berarti model berhasil merekonstruksi dengan baik, sedangkan piksel dengan nilai MAE yang tinggi mengindikasikan probabilitas anomali yang kuat (Bouman & Heskes, 2025). Peta galat spasial tingkat *patch* ini yang kemudian akan digabungkan kembali (*stitching*) menjadi peta anomali resolusi penuh sebelum akhirnya diklasifikasikan melalui proses *thresholding* (Cui dkk., 2023; Fang dkk., 2024).

2.6 Reproduksi Citra

Reproduksi citra merupakan teknik penggabungan kembali citra input yang tidak dieksekusi secara secara utuh melainkan dipecah menjadi potongan-potongan kecil (*patches*) melalui metode *sliding window* dengan ukuran dan langkah (*stride*) tertentu (Hermann dkk., 2022). Citra input yang melewati model *Autoencoder* akan menghasilkan representasi *patch* rekonstruksi (Behrendt dkk., 2023). Evaluasi peta galat (*error map*) pada tahap inferensi dihitung berdasarkan galat absolut antar piksel (*Absolute Error*) (Pan dkk., 2025), sehingga metrik galat tidak dihitung secara global untuk satu citra penuh secara langsung, melainkan diperlukan tahapan reproduksi untuk menggabungkan kembali potongan-potongan *patch* tersebut menjadi peta anomali beresolusi penuh yang tahapannya disebut dengan *stitching* (Behrendt dkk., 2023).

Pada praktiknya, prosedur reproduksi diawali dengan mengaplikasikan teknik *reflect padding* pada batas luar citra asli untuk menjaga kontinuitas informasi di area tepi saat proses ekstraksi *patch* (Huang dkk., 2022). Selama proses penyatuan

area yang saling tumpang tindih (*overlapping*), model menerapkan dua pendekatan berbeda. Pendekatan pertama, untuk menghasilkan visualisasi citra rekonstruksi yang mulus tanpa garis batas (*seam artifacts*), intensitas piksel pada area yang tumpang tindih diakumulasikan menggunakan pembobotan spasial fungsi jendela Hann (*Hann Window*) dua dimensi (2D). Fungsi jendela Hann satu dimensi (1D) diedifinisikan secara matematis pada Persamaan 2.10 (Bäckström dkk., 2025).

$$w(n) = 0.5 \times \left(1 - cos\left(\frac{2\pi n}{N-1}\right)\right) \tag{2.10}$$

Dimana n merupakan indeks elemen dalam jendela ($0 \leq n \leq N-1$) dan N menyatakan ukuran total jendela (*crop size*) yang bernilai 256 piksel. Karena penggabungan citra dilakukan pada ruang spasial dua dimensi, maka matriks bobot spasial jendela Hann 2D ($w(x,y)$) diperoleh melalui perkalian tensor (*outer product*) dari dua fungsi jendela Hann 1D, seperti yang dirumuskan pada Persamaan 2.11 (Lim, 1990).

$$w(x,y) = w(x) \cdot w(y) \tag{2.11}$$

Substitusi Persamaan 2.10 ke dalam Persamaan 2.11 menghasilkan formulasi lengkap pembobotan spasial 2D pada Persamaan 2.12.

$$w(x,y) = 0.25\left[1 - cos\left(\frac{2\pi x}{N-1}\right)\right]\left[1 - cos\left(\frac{2\pi y}{N-1}\right)\right] \tag{2.12}$$

Di mana x dan y merepresentasikan koordinat piksel lokal di dalam potongan citra (*patch*) dengan rentang $0 \leq x,y \leq N-1$.

Kemudian dinormalisasi melalui pembagian dengan total bobot (*weighted averaging*). Nilai intensitas piksel akhir pada kanvas citra rekonstruksi ($R_{final}(x,y)$) diperoleh melalui rata-rata tertimbang (*weighted average*) dari akumulasi seluruh bagian *patch* yang saling tumpang tindih, menggunakan rumus pada Persamaan 4.4 (Ross, 2014).

$$R_{final} = \frac{\sum_{i=1}^{M} R_i(x,y) \cdot W_i(x,y)}{\sum_{i=1}^{M} W_i(x,y)} \tag{2.13}$$

Dimana $R_i(x,y)$ adalah citra hasil rekonstruksi dari *patch* ke-i, $W_i(x,y)$ adalah masker bobot Hann yang bersesuaian, dan M adalah jumlah *patch* yang saling tumpang tindih pada koordinat spasial tersebut.

Pendekatan yang kedua, untuk menyusun peta galat beresolusi penuh, algoritma menggunakan pendekatan galat minimum (*Min-Error Stitching*). Pada titik koordinat yang bersinggungan, nilai galat terkecil yang akan dipertahankan untuk meminimalkan *false positive* pada batas *patch*, sekaligus menjaga konsistensi distribusi spasial yang menjadi indikator penting dalam segmentasi anomali tingkat piksel (Han dkk., 2024; Li dkk., 2023). Persamaan 2.13 merupakan formula *Min-Error Stitching* yang digunakan.

$$E_{final}(x,y) = min_{i=1}^{M}\big(E_i(x,y)\big) \tag{2.14}$$

Di mana $E_{final}(x,y)$ adalah nilai galat akhir di koordinat tertentu, min adalah fungsi untuk mencari nilai terkecil, $i = 1$ sampai M berarti evaluasi semua *patch* (dari *patch* pertama sampai patch terakhir yang bersinggungan di titik tersebut), dan $E_i(x,y)$ adalah nilai galat dari patch ke- i di koordinat tersebut.

2.7 Matriks untuk Segmentasi

2.5.1 *Thresholding* (Penentuan Ambang Batas)

Proses *Thresholding* berfungsi untuk mengkonversi peta skor probabilitas menjadi keputusan biner yang menentukan Normal atau Anomali ($S(x, y)$) (Anjali & Don, 2025; Giannoulidis dkk., 2022). Peta kesalahan kontinu $E(x, y)$ dikonversi menjadi keputusan biner (0 atau 1) menggunakan nilai ambang batas dinamis (τ) yang diperoleh dari persentil ke-90 distribusi galat data validasi normal, dengan formulasi matematika sesuai Persamaan 2.15.

$$S(x,y)\begin{cases}1, jika\ E(x,y) \geq \tau \\ 0, jika\ E(x,y) < \tau\end{cases} \tag{2.15}$$

Di mana $S(x, y) = 1$ merepresentasikan piksel yang tersegmentasi sebagai anomali (cacat permukaan) dan $S(x, y) = 0$ merepresentasikan piksel normal.

Pemilihan nilai ambang batas (*threshold*) sangat menentukan keseimbangan antara sensitivitas dan spesifisitas sistem, di mana nilai yang optimal diperlukan untuk meminimalkan *False Positives* dan menghindari terjadinya *False Negatives* (Anjali & Don, 2025; Giannoulidis dkk., 2022).

2.5.2 *Area Under the Receiver Operating Characteristic* (AUROC)

AUROC atau AUC-ROC adalah metrik evaluasi tolok ukur utama untuk membandingkan kinerja algoritma deteksi anomali secara global, terlepas dari pemilihan ambang batas (*thresholding*) tertentu (Mcdermott dkk., 2025).

Dalam mengevaluasi kinerja model deteksi dan segmentasi anomali, pengukuran tidak dapat dilakukan secara langsung, melainkan harus membandingkan hasil prediksi model dalam hal ini peta anomali (anomaly map) dengan data kebenaran dasar (*ground truth*).

True Positive (TP) merupakan sampel anomali yang dideteksi dengan benar sebaga anomali, sedangkan *False Positive* (FP) yaitu sampel normal yang salah diklasifikasikan sebagai anomali (*False Alarm*). Tingkat FP yang tinggi dapat menyebabkan kelelahan operator (*alert fatigue*) dalam sistem keamanan.

Kurva ROC menempatkan *True Positive Rate* (TPR/*Recall*) pada sumbu Y melawan *False Positive Rate* (FPR) pada sumbu X untuk semua kemungkinan nilai ambang batas berupa klasifikasi (Bouman & Heskes, 2025).

$$FPR = \frac{FP}{(TN+FP)} \tag{2.16}$$

AUROC adalah integral area di bawah kurva tersebut. Nilai AUROC 1.0 menunjukkan pemisahan sempurna antara distribusi skor normal dan anomali, sementara 0.5 menunjukkan kinerja acak (Humblot-Renaux dkk., 2023).

Dalam konteks segmentasi tingkat piksel, evaluasi sering kali menggunakan pendekatan *Macro-Average* ROC daripada menghitung metrik secara global dengan menggabungkan seluruh piksel uji menjadi satu distribusi besar.

Pendekatan rata-rata makro ini bekerja dengan cara menghitung kurva ROC dan nilai AUC secara individual untuk setiap citra yang diasumsikan memiliki anomali, kemudian menginterpolasi dan merata-ratakan keseluruhan kurva tersebut untuk mendapatkan satu metrik evaluasi akhir.

Pendekatan berbasis ROC ini umum digunakan dalam evaluasi deteksi anomali karena kurva ROC merepresentasikan hubungan antara *true positive rate* dan *false positive rate* yang kemudian sering diinterpolasi secara linier untuk membentuk kurva evaluasi yang lebih representatif (Chang, 2022).

Penghitungan metrik yang dirata-ratakan makro per citra dilakukan untuk mencegah dominasi statistik dari citra produk yang memiliki dimensi cacat atau robekan sangat besar hasil sebaran piksel *True Positive* yang masif. Dominasi statistik disebabkan oleh perhitungan nilai dengan cara menggabungkan total seluruh piksel dari semua gambar. Hal ini penting karena dalam tugas segmentasi anomali, distribusi ukuran cacat sangat variatif dan seringkali tidak seimbang, sehingga metrik berbasis piksel global dapat bias terhadap anomali berukuran besar dan mengabaikan performa pada anomali kecil (Kerssies & Vanschoren, 2023).

Dengan memberikan bobot evaluasi yang setara pada setiap citra terlepas dari luas area anomalinya, performa model dalam mendeteksi citra dengan cacat berukuran kecil seperti goresan halus agar tetap tervalidasi secara adil dan tidak bias terhadap anomali berskala besar. Variasi ukuran area anomali yang signifikan pada dataset segmentasi telah dilaporkan sebagai faktor yang dapat menyebabkan ketidakseimbangan evaluasi jika tidak ditangani dengan strategi agregasi yang tepat.

2.8 Penelitian Terkait

Penelitian terkait berfungsi sebagai acuan untuk memposisikan kontribusi penelitian ini terhadap studi terdahulu. Tabel 2.1 merangkum 13 penelitian relevan yang disusun berdasarkan referensi utama, mencakup pengembangan *Autoencoder*, penggunaan dataset *MVTec AD*, serta metode adaptif terbaru yang menjadi landasan penelitian ini.

Tabel 2.1 *State Of The Art*

No.	Penulis, Tahun Terbit	Judul	Metode	Hasil
1.	(Stepien, 2026)	SoftSAE : *Dynamic Top-K Selection for Adaptive Sparse Autoencoder*s	SoftSAE (*Sparse Autoencoder*)	SoftSAE berhasil memberikan kontrol tingkat kepadatan fitur (*sparsity*) yang sangat ketat dengan tetap mempertahankan ketepatan rekonstruksi yang kompetitif jika dibandingkan dengan model pembanding yang kuat seperti TopK SAE, BatchTopK, dan Matryoshka.
2.	(Yang & Guo, 2024)	*An Unsupervised Method for Industrial Image Anomaly Detection with Vision Transformer-Based Autoencoder*	Vision Transformer (ViT)	Peningkatan rata-rata skor AUROC sekitar 20% pada tingkat citra (*image-level*) dengan pemanfaatan *Memory Module*, *Vision Transformer-Based Autoencoder*, dan Coordinate Attention.
3.	(Sabri, 2022)	Segmentasi Anomali Produk Dengan *SSIM Autoencoder*	*SSIM Autoencoder* (*Fixed Bottleneck*)	Menemukan korelasi ukuran *bottleneck* dengan jenis objek. *Fixed Bottleneck* memerlukan penyetelan manual yang rentan kesalahan.
4.	(Fan dkk., 2025)	SSIM *over* MSE: *A new perspective for video anomaly detection*	SSIM-*based Loss* & STE-3D	Meningkatkan relevansi deteksi dengan persepsi visual manusia dan mencapai performa AUC unggul pada dataset *benchmark* (misalnya 98.9% pada *UCSD Ped2* dan 88.4% pada *UCSD Ped1* menggunakan basis *STEAL Net*).

No.	Penulis, Tahun Terbit	Judul	Metode	Hasil
5.	(Ghosh & Kirby, 2023)	*Feature Selection using Sparse Adaptive Bottleneck* Centroid-*Encoder*	*Sparse Adaptive Bottleneck*	Mengusulkan seleksi fitur adaptif pada *Bottleneck* untuk mengurangi dimensi secara otomatis, relevan dengan konsep adaptivitas.
6.	(Shiferaw & Yao, 2024)	*Autoencoder-Based Unsupervised Surface Defect Detection Using Two-Stage Training*	*U-Net AE + Adaptive Weighted SSIM*	Mengembangkan pembobotan adaptif pada *loss function* SSIM untuk meningkatkan sensitivitas terhadap cacat halus.
7.	(Anjali & Don, 2025)	*Deep Learning-Based Video Anomaly Detection Using Optimised Attention-Enhanced Autoencoder*s	SESAA (*Self-Attention SE Autoencoder*)	Mencapai skor AUC 0.98 pada dataset *UCSD Ped1*, 0.83 pada *UCSD Ped2*, dan 0.91 pada *Avenue*, mengungguli metode *Thresholding* dinamis konvensional dalam menangani variasi pencahayaan.
8.	(Ye dkk., 2024)	*Deep Equilibrium Convolutional Sparse Coding* for *Hyperspectral Image Denoising*	*Deep Equilibrium Sparse Coding*	Menggunakan *Sparse coding* untuk *Denoising* Citra kompleks, menunjukkan efektivitas representasi jarang (*Sparse*) untuk data visual.
9.	(Cheng dkk., 2024)	*Hyperspectral Anomaly Detection via Low-Rank Representation with Dual Graph Regularization*s *and Adaptive Dictionary*	*Dual Graph Regularization and Adaptive Dictionary Low-Rank Representation*	Mencapai performa *state-of-the-art* pada 6 dataset hiperspektral (*Gulfport*, *Texas Coast*, *Los Angeles*, *Salinas*, *San Diego-1 & 2*) dengan skor AUC (D,F) rata-rata di atas 0.99, mengungguli metode LRR konvensional dan *Deep Learning* (*Auto-AD*) dalam akurasi deteksi dan stabilitas.
10.	(Ayyadurai dkk., 2025)	*Deep Convolutional Autoencoder*s *for Fraud Detection in Digital Banking*: *Anomaly Detection with Reconstruction Error*	*Deep Convolutional Autoencoder* (DCAE)	Mencapai akurasi 98.7% pada dataset *PaySim* dengan tingkat kesalahan yang sangat rendah (FPR 0.47%, FNR 0.81%), mengungguli metode konvensional dalam efisiensi deteksi *real-time*.
11.	(Benabderrahmane dkk., 2025)	*Rank*ing-*Enhanced Anomaly Detection Using Active Learning-Assisted Attention Adversarial Dual Autoencoder*s	*Dual Adversarial Autoencoders* + mekanisme *Attention*	Mencapai performa *state-of-the-art* pada dataset *DARPA Transparent Computing* (*Windows*, *Linux*, BSD, *Android*) dengan skor nDCG tertinggi (hingga 0.98 pada skenario sulit), mengungguli metode *Deep learning* terbaru seperti *MAE-AD* dan *RT-APT*, serta menunjukkan peningkatan performa signifikan (>100%) melalui iterasi *active learning*.

No.	Penulis, Tahun Terbit	Judul	Metode	Hasil
12.	(Karimi & Renani, 2025)	*A Mathematical Framework for Anomaly Detection in High-Dimensional Data Using Sparse Autoencoder*s *and Mutual Information Filtering*	*Variational Autoencoder* + *L1-norm*	Mencapai Akurasi 93.7% dan AUC 0.96 pada dataset *UNSW-NB15*; berhasil mereduksi fitur dari 49 menjadi 22 serta mengungguli metode *Deep Autoencoder* standar dan *CNN-LSTM*.
13.	(Najafi & Asemani, 2024)	*Attention And Autoencoder* Hybrid Model For *Unsupervised* Online *Anomaly Detection*	*Hybrid Attention & Autoencoder*	Mengungguli model *state-of-the-art* (seperti VAE-LSTM) pada dataset *Numenta Anomaly Benchmark* (NAB). Mencapai *F1-Score* 100% pada dataset *CPU Utilization* dan *Traffic Occupancy*, serta 99% pada *Machine Temperature*.
14.	(Sellam dkk., 2025)	*Mamba Adaptive Anomaly Transformer with association discrepancy for time series*	*Sparse Attention* dan Mamba-SSM (*State Space Model*) serta mekanisme *Gated Attention*	Mencapai performa *state-of-the-art* (SOTA) pada 7 dataset *benchmark* (termasuk SMD, MSL, S*MAP*, SWaT). Mengungguli *Anomaly Transformer* dan *DCdetector* dalam metrik *F1-Score*, *Recall*, dan Akurasi, serta menunjukkan ketahanan lebih baik terhadap *noise* dan dependensi jangka panjang.

Penelitian sebelumnya telah mengeksplorasi berbagai pendekatan dalam deteksi anomali tak terawasi (*unsupervised*) pada citra industri, dengan referensi terdekat yaitu penelitian (Sabri, 2022) yang mengembangkan model *SSIM Autoencoder* untuk segmentasi anomali produk dengan keterbatasan pada mekanisme *fixed bottleneck* (statis) yang mengharuskan penentuan ukuran dimensi laten secara manual sehingga rentan memicu generalisasi berlebih (*over-generalization*). Keterbatasan tersebut diatasi oleh (Yang & Guo, 2024) melalui penggunaan *Memory Module*, *Vision Transformer-Based Autoencoder*, dan *Coordinate Attention* yang menghasilkan peningkatan rata-rata skor AUROC sekitar 20% pada tingkat citra (*image-level*) dengan mekanisme *bottleneck* adaptif, namun proses pencocokkan (*matching*) di setiap iterasi membuat beban komputasi besar. Penelitian (Stępień dkk., 2026) melalui metode *Dynamic Top-K Selection* yang juga berhasil memberikan kontrol tingkat kepadatan fitur (*sparsity*) dengan tetap mempertahankan ketepatan rekonstruksi yang kompetitif, namun proses pengurutan (*sorting*) yang secara kontinu memakan biaya komputasi yang tinggi dan membatasi skalabilitas model. Mengisi celah (*gap*) penelitian dari ketiga studi tersebut, penelitian ini mengusulkan *Adaptive Bottleneck Autoencoder* dengan mekanisme *Gating* menggunakan *Sparsity Gating Network* berbasis satu lapisan linear dan aktivasi *Sigmoid* yang mampu menekan fitur anomali secara dinamis dengan kompleksitas waktu konstan yang jauh lebih ringan.

2.9 Matriks Penelitian

Tabel 2.2 Matriks Penelitian

No.	Peneliti,Tahun	Metode Dasar	*Mekanisme Bottleneck*	*Attention*/ *Gating*	*Output* (*Pixel-Level*)
1.	(Stepien, 2026)	SoftSAE : *Dynamic Top-K Selection* for *Adaptive* Sparse *Autoencoder*s	Adaptif	-	×
2.	(Yang & Guo, 2024)	Vision Transformer (ViT)	Adaptif	*Attention*	✓
3.	(Sabri, 2022)	*SSIM Autoencoder*	Statis	-	✓
4.	(Fan dkk., 2025)	SSIM-*based Loss* & STE-3D	Statis	*Attention*	✓
5.	(Ghosh & Kirby, 2023)	*Sparse Adaptive Bottleneck*	Adaptif	-	×
6.	(Shiferaw & Yao, 2024)	*U-Net AE + Adaptive Weighted SSIM*	Statis	-	✓
7.	(Anjali & Don, 2025)	SESAA (*Self-Attention SE Autoencoder*)	Statis	*Gating* + *Attention*	×
8.	(Ye dkk., 2024)	*Deep Equilibrium Sparse Coding*	Adaptif	*Attention*	×
9.	(Cheng dkk., 2024)	*Dual Graph Regularization and Adaptive Dictionary Low-Rank Representation*	Adaptif	-	✓
10.	(Ayyadurai dkk., 2025)	*Deep Convolutional Autoencoder*	Statis	-	×
11.	(Benabderrahmane dkk., 2025)	*Dual Adversarial Autoencoders* + mekanisme *Attention*	Statis	*Attention*	×
12.	(Karimi & Renani, 2025)	*Variational Autoencoder + L1-norm*	Statis	-	×
13.	(Najafi & Asemani, 2024)	*Hybrid Attention & Autoencoder*	Statis	*Attention*	×
14.	(Sellam dkk., 2025)	*Sparse Attention* dan Mamba-SSM serta mekanisme *Gated Attention*	Statis	*Gating* + *Attention*	×
15.	Usulan Penelitian	*Adaptive* AE	Adaptif	*Gating*	✓

Tabel 2.2 Matriks Penelitian disusun untuk menampilkan perbandingan komprehensif dengan empat aspek utama yaitu metode dasar yang digunakan, mekanisme *bottleneck*, penerapan *attention* atau *gating*, serta orientasi output pada tingkat piksel (*pixel-level*). Berdasarkan literatur yang dikaji, banyak penelitian terdahulu mengeksplorasi berbagai variasi arsitektur seperti *Autoencoder konvensional*, *Vision Transformer*, hingga model berbasis *Sparse Coding*, dan mayoritas dari penelitian tersebut masih mempertahankan mekanisme *bottleneck* yang bersifat statis di dalam arsitektur jaringannya. Studi terdahulu cenderung memanfaatkan mekanisme *attention* secara tunggal, menggabungkannya dengan *gating*, atau bahkan tidak menggunakan mekanisme tambahan sama sekali. Penelitian ini difokuskan secara khusus pada penerapan mekanisme *gating* guna mengontrol dan menyaring aliran informasi di dalam jaringan. Pendekatan dalam penelitian ini mengintegrasikan *Adaptive Bottleneck Autoencoder* berbasis mekanisme *Gating Network* untuk regulasi fitur laten yang memangkas beban komputasi secara signifikan menjadi berkecepatan konstan, yang menitikberatkan pada peningkatan presisi piksel dengan mempertimbangkan aspek generalisasi berlebih (*over-generalization*) serta hubungan spasial antarpiksel yang lebih kompleks, sehingga penelitian ini berkontribusi dalam meningkatkan efektivitas model segmentasi anomali dengan menggabungkan pendekatan komputasi yang ringan untuk mencapai hasil yang lebih optimal. *Output* hasil akhir usulan penelitian ini dirancang untuk mampu memproduksi luaran pada tingkat piksel (*pixel-level*) berupa segmentasi anomali pada citra industri.