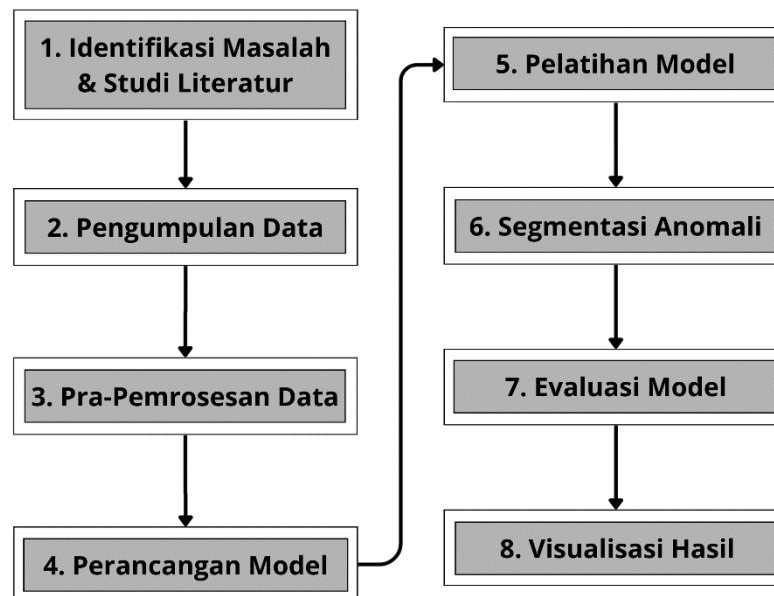


## BAB III

### METODOLOGI

#### 3.1 Kerangka Penelitian

Penelitian ini mengadopsi kerja *Design Science Research Methodology* (DSRM) (Peffer dkk., 2007) dengan menerapkan desain eksperimental–kuantitatif (Creswell, 2009) untuk menguji kinerja model *Adaptive Bottleneck Autoencoder* secara sistematis pada dataset *MVTec AD*. Pendekatan eksperimental digunakan dengan tujuan mengetahui pengaruh dan efektivitas mekanisme *Gating* terhadap akurasi segmentasi anomali citra industri dan ukuran komputasi yang digunakan. Variabel utama yang dikaji berupa variasi kategori dalam dataset yaitu *Grid* dan *Zipper*, arsitektur *Adaptive Bottleneck*, serta mekanisme *Gating* yang digunakan. Target kebaruan (*novelty*) dalam penelitian ini adalah penerapan kemampuan adaptif model untuk menentukan ukuran *bottleneck* yang optimal secara otomatis pada citra industri tanpa harus melalui proses *trial-and-error* dengan penggunaan beban komputasi yang ringan. Mengimplementasikan metodologi penelitian DSRM dan desain penelitian eksperimental-kuantitatif tersebut, ditentukan 8 Tahapan Penelitian yang berisikan Identifikasi Masalah dan Studi Literatur, Pengumpulan Data, Pra-Pemrosesan Data, Perancangan Model, Pelatihan Model, Segmentasi Anomali, Evaluasi Model, dan Visualisasi Hasil. Tahapan penelitian secara menyeluruh digambarkan pada gambar 3.1.



Gambar 3.1 Kerangka Penelitian

### 3.1.1 Identifikasi Masalah & Studi Literatur

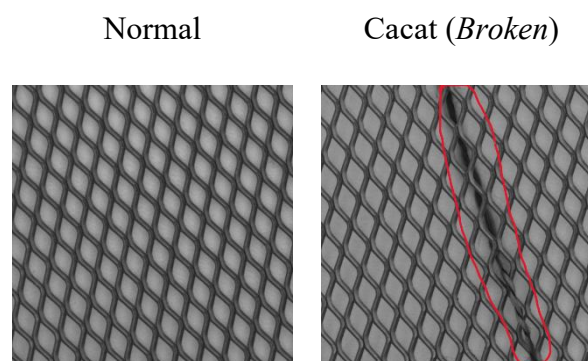
Melalui tahapan identifikasi dan studi literatur ini, diperoleh pemahaman bahwa dalam penentuan ukuran *bottleneck* yang bersifat adaptif diperlukan mekanisme seleksi fitur secara dinamis pada ruang laten (*bottleneck*) untuk membatasi aliran informasi anomali tanpa mengurangi kualitas rekonstruksi pola normal. Pemahaman ini kemudian menjadi dasar untuk merumuskan metode baru dalam penelitian, yaitu pengembangan *Adaptive Bottleneck Autoencoder* yang terintegrasi dengan *Gating Network* dan regularisasi sparsitas untuk menghasilkan segmentasi anomali yang presisi dengan biaya komputasi yang ringan.

### 3.1.2 Pengumpulan Data

Data yang dikumpulkan pada tahapan awal dalam penelitian ini bersumber dari *website MVTec AD*, yang menyediakan 5354 gambar beresolusi tinggi. Terdiri dari 15 kategori yang mencakup berbagai objek dan tekstur, berisi 3629 citra pelatihan

(*train*) dan 1725 citra pengujian (*test*) (Software, 2019). Penelitian ini hanya menggunakan dataset *Grid* dan *Zipper*, adapun pertimbangan dalam pemilihan dataset adalah sebagai berikut:

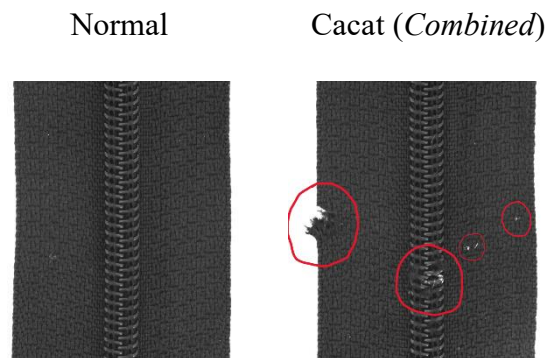
1. *Grid* mewakili data dengan jenis utama *Texture* (Tekstur), yang memiliki karakteristik berupa motif pola berulang tanpa orientasi tetap. Kategori *Grid* dipilih untuk mengevaluasi efektivitas mekanisme *Adaptive Bottleneck* dalam mengompresi representasi fitur dari struktur repetitif secara efisien, serta menguji kemampuan *Gating* dalam membedakan antara variasi tekstur normal dan gangguan pola (anomali) terpengaruh oleh kompleksitas visual yang berulang. Contoh dataset *Grid* ditampilkan pada Gambar 3.2.



Gambar 3.2 Contoh Dataset *MVTec AD* Kategori *Grid* Normal dan Anomali (*Broken*)

2. *Zipper* mewakili data dengan jenis utama *object* (Objek), yaitu benda tunggal dengan posisi tetap. Dataset *Zipper* digunakan untuk memvalidasi kinerja mekanisme *Gating* dalam memfokuskan rekonstruksi pada area objek utama (*foreground*) dan mengabaikan latar belakang (*background*), sekaligus menguji kemampuan model dengan arsitektur adaptif dalam mempertahankan detail struktural halus yang krusial untuk segmentasi

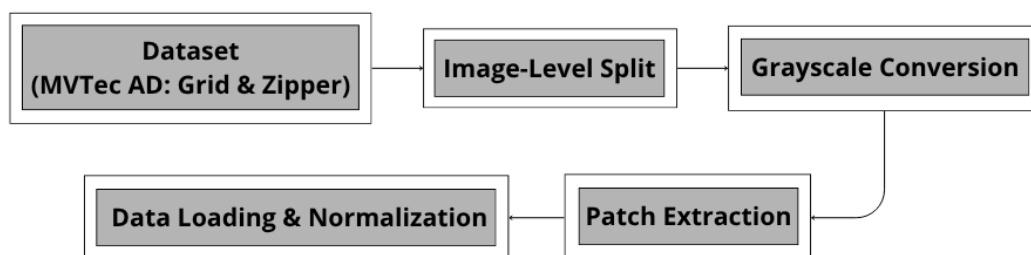
anomali pada *rigid object* (benda tegar). Contoh dataset *Zipper* ditampilkan pada gambar 3.3.



Gambar 3.3 Contoh Dataset *MVTec AD* Kategori *Zipper* Normal dan Anomali (*Combined*)

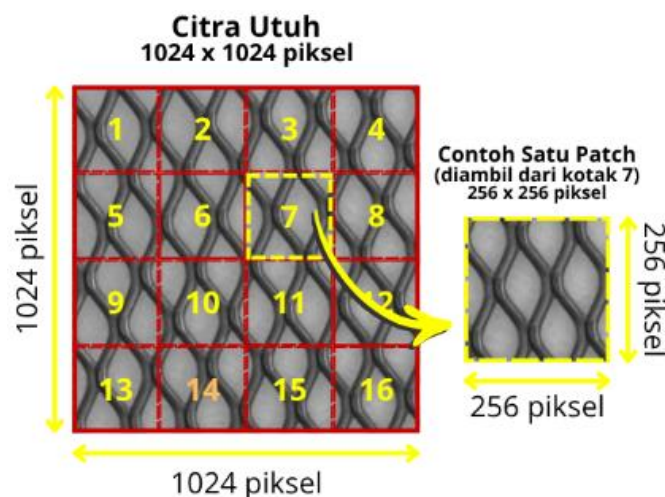
### 3.1.3 Pra-Pemrosesan Data (*Data Preprocessing*)

Tahapan pra-pemrosesan data (*data preprocessing data*) merupakan tahapan pengolahan data mentah menjadi data yang siap digunakan untuk proses pelatihan dan pengujian model. Tujuan utama dari tahap *preprocessing data* adalah untuk memastikan bahwa data memiliki kualitas yang konsisten, bebas dari variasi yang tidak relevan, serta siap digunakan agar model dapat belajar secara efektif. Pada tahap *preprocessing*, citra dari dataset *MVTec AD* kategori *Grid* dan *Zipper* yang masih berupa citra mentah dimuat untuk kemudian dipersiapkan sebagai data *input* bagi model. Tahapan lengkap *preprocessing* akan dijabarkan pada Gambar 3.4.



Gambar 3.4 Tahap *Preprocessing*

Berdasarkan Gambar 3.4, dataset *MVTec AD* dengan resolusi 1024x1024 piksel membagi daftar jalur file (*file paths*) dari citra normal menjadi himpunan data latih (*training set*) dan data validasi (*validation set*) dengan proporsi 80:20 yang dilakukan pada tingkat gambar utuh (*image-level*) untuk menjamin tidak adanya kebocoran informasi (*data leakage*) antar potongan citra. Citra asli dikonversi menjadi format skala keabuan (*grayscale*) dengan satu saluran (*channel*). Citra hasil konversi kemudian dipecah menjadi potongan-potongan kecil (*patches*) berukuran 256x256 piksel yang selanjutnya dilakukan perubahan dimensi (*resizing*) menjadi 128x128 piksel seperti yang diilustrasikan pada Gambar 3.5.

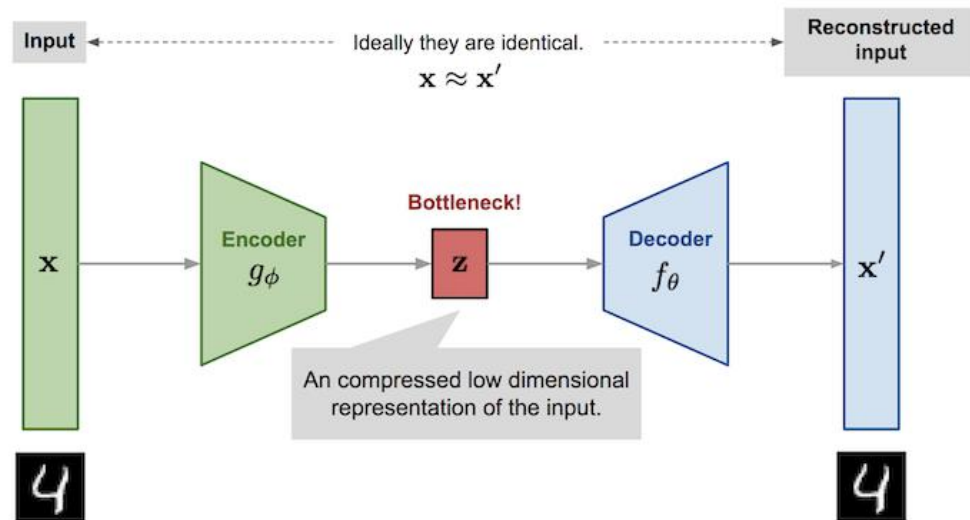


Gambar 3.5 Ilustrasi Proses Pemotongan Citra

Tahap terakhir yaitu mentransformasi nilai intensitas piksel (0-255) ke dalam rentang skala 0.0 hingga 1.0 (normalisasi) secara otomatis dengan pemuatan data (*data loading*).

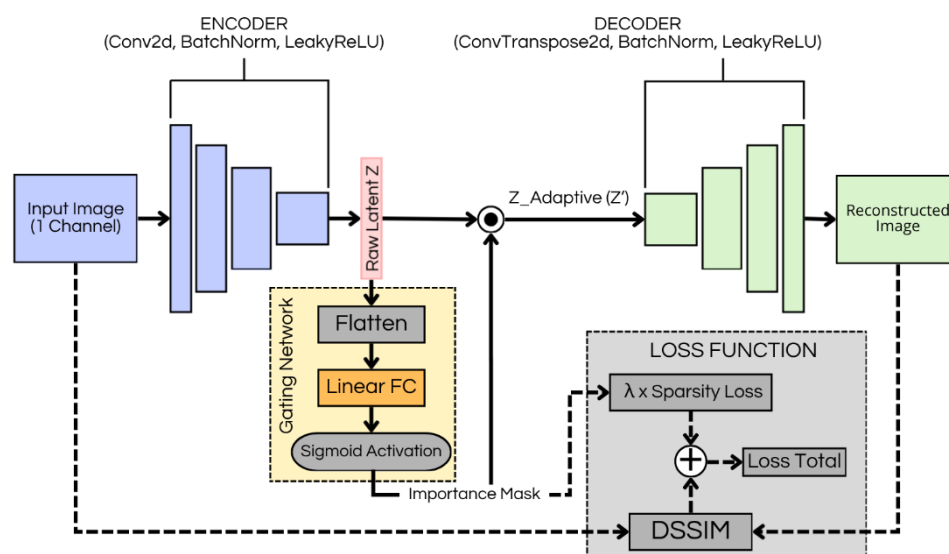
#### 3.1.4 Perancangan Model *Adaptive Bottleneck Autoencoder*

Penelitian ini merancang arsitektur *Adaptive Bottleneck Autoencoder* dengan mengintegrasikan mekanisme *Gating*. Perbandingan konseptual antara arsitektur CAE standar dan arsitektur usulan direpresentasikan pada Gambar 3.5 dan 3.6.



Gambar 3.6 Arsitektur Convolutional Autoencoder (Allahyari, 2023)

Gambar 3.5 merupakan gambar Arsitektur dasar *Autoencoder* dimana representasi laten di desain untuk menangkap fitur-fitur intrinsik dari data, menghilangkan redundansi dan *noise* yang bekerja secara statis, berbeda dengan *Adaptive Bottleneck Autoencoder* dengan mekanisme *Gating* yang digambarkan pada Gambar 3.6.



Gambar 3.7 Arsitektur *Adaptive Bottleneck Autoencoder* (dimodifikasi dari model

Allahyari, 2023; Jayeprokash & Gonski, 2025)

Gambar 3.6 merupakan gambaran arsitektur *Adaptive Bottleneck Autoencoder* hasil modifikasi dari model *Convolutional Autoencoder* standar dengan menambahkan mekanisme *Sparsity Gating Network* yang menjadi model utama dalam penelitian ini.

Arsitektur yang diusulkan dirancang secara khusus untuk menangani kompleksitas tekstur (*Zipper*) dan objek (*object*) pada citra produk industri. Seperti yang diilustrasikan pada gambar 3.6, model ini merupakan pengembangan dari *Convolutional Autoencoder* (CAE) yang terdiri dari tiga komponen utama: *Encoder*, *Adaptive Bottleneck Module*, dan *Decoder*.

Bagian *Encoder* digambarkan sebagai susunan blok lapisan yang secara progresif mengompresi citra *input* (*Input Image*) melalui beberapa lapisan konvolusi. Struktur blok lapisan yang semakin mengecil merepresentasikan proses ekstraksi fitur dan reduksi dimensi, di mana informasi visual dari citra dipadatkan menjadi representasi fitur laten, tanpa menampilkan kompleksitas spesifik dari konfigurasi *Kernel* atau lapisan normalisasi di dalamnya, yang kemudian menghasilkan laten mentah (*Raw Latent Z*).

Vektor laten mentah (*Raw Latent Z*) diproses melalui *Gating Network* yang terdiri dari operasi *Flatten*, lapisan *Linear FC*, dan aktivasi *Sigmoid*. Jalur paralel ini menghasilkan *Importance Mask* yang kemudian dikalikan (*element-wise multiplication*) dengan vektor laten mentah. Tahapan pemrosesan vektor laten mentah melalui *gating network* yang merupakan konsep inti adaptivitas model dalam memfilter fitur penting sebelum rekonstruksi. Bagian *Decoder* ditampilkan

sebagai cerminan simetris dari *Encoder* yang bertugas melakukan rekonstruksi citra. Rangkaian blok yang membesar secara bertahap merepresentasikan proses pemulihan informasi (*upsampling*) dari vektor laten adaptif ( $Z'$ ) hingga kembali menjadi citra utuh (*Reconstructed Image*), menggambarkan aliran balik dari representasi padat ke dimensi spasial asli.

### 3.1.5 Pelatihan Model (*Training*)

Pada tahap pelatihan (*training*), model *Adaptive Bottleneck Autoencoder* mempelajari citra *patch* secara tak terawasi (*unsupervised*) dari sampel data normal tanpa melibatkan label anomali apapun agar mampu mengekstraksi dan merekonstruksi representasi fitur spasial secara optimal.

#### 3.1.5.1 Konfigurasi Hiperparameter dan Optimasi

Proses pelatihan model *Adaptive Bottleneck Autoencoder* dilakukan pada lingkungan komputasi GPU T4 melalui *Google Colab*. Model dikonfigurasi dengan batas maksimum dimensi laten sebesar 500 dan dilatih selama 100 *epoch* dengan ukuran *batch* sebesar 16 guna memastikan estimasi gradien yang akurat serta efisiensi penggunaan memori. Pengoptimalan parameter dilakukan menggunakan algoritma Adam dengan laju pembelajaran  $1 \times 10^{-4}$  dan *weight decay*  $1 \times 10^{-5}$  sebagai bentuk regularisasi untuk mencegah terjadinya *overfitting*. Kualitas rekonstruksi visual dipertahankan melalui pengaturan *window size* SSIM sebesar 11. Seleksi fitur esensial pada *bottleneck* diregulasi melalui penerapan koefisien sparsitas ( $\lambda$ ) sebesar 0.001 (Ghosh & Kirby, 2023). Untuk menjamin performa akhir yang optimal, strategi penyimpanan bobot model dilakukan secara otomatis dengan menyimpan *checkpoint* terakhir secara berkala, serta menyimpan model terbaik

berdasarkan nilai *validation loss* terendah yang dicapai selama proses pelatihan (*training*) guna menjaga integritas kemampuan generalisasi model.

### 3.1.5.2 Formulasi Fungsi Tujuan (*Objective Function*)

Formulasi Fungsi Tujuan (*Objective Function*) merupakan penggabungan *loss function* (DSSIM) dengan *sparsity loss* ( $\lambda$ ) yang diperlukan untuk memandu proses pembaruan bobot selama pelatihan agar model *Adaptive Bottleneck Autoencoder* dapat mempelajari representasi fitur yang optimal. Fungsi tujuan dalam penelitian ini dirancang secara khusus untuk memastikan citra hasil rekonstruksi memiliki kemiripan visual yang tinggi dengan citra input, dan memaksa mekanisme *Gating* untuk bekerja secara selektif (adaptif) dalam memilih fitur. Secara matematis, fungsi kerugian total dari penggabungan tersebut diformulasikan sebagai gabungan dari *Reconstruction Loss* (DSSIM) dan *Sparsity Loss* ( $L_1$  Norm):

$$L_{total} = L_{DSSIM} + \lambda \times L_{Sparsity} \quad (3.1)$$

Dimana  $\lambda$  (*lambda*) adalah hiperparameter skalar yang mengatur keseimbangan antara akurasi rekonstruksi dan tingkat *Sparsity*.

### 3.1.6 Segmentasi Anomali

Tahapan segmentasi bertujuan untuk melokalisasi anomali secara presisi dengan mengonversi error map kontinu menjadi keputusan biner. Proses ini diawali dengan reproduksi citra yang bertujuan untuk menyatukan kembali potongan-potongan (*patches*) hasil rekonstruksi *Autoencoder* menjadi satu citra utuh beresolusi spasial asli tanpa mengorbankan akurasi lokalisasi.

Potongan-potongan citra (*patch*) ini kemudian digabungkan kembali (*stitching*) menjadi peta anomali beresolusi penuh. Dalam proses penggabungan area yang saling tumpang tindih (*overlapping*), model menerapkan pendekatan *Hann window* (pembobotan spasial) yang diterapkan khusus untuk memperhalus penggabungan visual citra rekonstruksi. Dilanjut pembentukan *error map* berbasis jarak absolut (*L1-Distance Map/Absolute Error Map*) yang mengekstraksi deviasi visual melalui komputasi selisih nilai intensitas secara langsung (*Pixel-to-Pixel*) pada seluruh area citra. Setelah peta anomali beresolusi penuh diperoleh melalui proses penggabungan (*stitching*), dilakukan kalibrasi ambang batas (*thresholding*) secara otomatis untuk menentukan batas keputusan. *Threshold* ditetapkan secara adaptif dengan mengambil nilai persentil ke-90 dari distribusi galat rekonstruksi tingkat piksel pada himpunan data validasi yang murni berisi citra normal. Pendekatan berbasis distribusi statistik ini memungkinkan model untuk mentoleransi variasi alami yang muncul pada produk normal, sekaligus memberikan pelokalan piksel anomali yang presisi tanpa bergantung pada prosedur penyetelan manual yang bersifat heuristik dan subjektif. Tahapan segmentasi anomali diilustrasikan pada Gambar 3.7.



Gambar 3.8 Tahapan Segmentasi Anomali

### 3.1.7 Evaluasi Model

Pengukuran kinerja model dalam melokalisasi area cacat dilakukan menggunakan metrik *Area Under the Receiver Operating Characteristic Curve* (AUROC) pada tingkat piksel. Penentuan nilai AUROC secara kuantitatif dihitung berdasarkan rata-rata metrik AUROC individu dari seluruh citra uji yang memiliki anomali. Pendekatan agregasi *Macro-Average* juga diterapkan, di mana kurva ROC dan nilai AUC dihitung secara mandiri untuk setiap citra uji yang memiliki anomali berdasarkan prediksi skor anomali piksel terhadap *ground truth* masing-masing kategori cacat. Hasil akhir kemudian diperoleh dengan merata-ratakan nilai AUC dari seluruh citra uji yang memiliki anomali (cacat) pada kategori tersebut. Untuk menghasilkan kurva ROC representatif, dilakukan interpolasi linear pada nilai *True Positive Rate* (TPR) di sepanjang sumbu *False Positive Rate* (FPR) yang seragam, sebelum akhirnya nilai AUC dirata-ratakan dari seluruh sampel citra beranomali pada kategori tersebut.

Strategi pemisahan evaluasi per citra ini dipilih untuk mencegah bias evaluasi akibat dominasi jumlah piksel normal serta meminimalisasi efek disparitas ukuran cacat antar sampel (*global pixel distribution bias*). Hal ini bertujuan untuk mencegah bias evaluasi akibat perbedaan ukuran cacat antar citra, sehingga setiap sampel memiliki bobot pengaruh yang sama terhadap skor akhir. Hasil akhirnya metrik ini secara objektif mencerminkan reliabilitas model dalam melakukan segmentasi presisi pada berbagai variasi geometri dan skala anomali tanpa terdistorsi oleh dominasi jumlah piksel normal pada distribusi data secara keseluruhan.

### 3.1.8 Visualisasi Hasil

Untuk melengkapi evaluasi kuantitatif, penelitian ini melakukan analisis kualitatif melalui visualisasi hasil segmentasi sebagai pembuktian kemampuan model dalam mendeteksi area anomali dan dibandingkan dengan *ground truth*. Visualisasi yang dihasilkan akan menampilkan citra original, citra hasil rekonstruksi, citra *ground truth*, *error map*, dan citra biner (*prediction*) untuk masing-masing kategori cacat pada dataset *Grid* dan *Zipper*.