

## **BAB III**

### **METODOLOGI PENELITIAN**

#### **3.1 Metodologi Penelitian**

Penelitian ini menerapkan metodologi eksperimental dengan pendekatan kuantitatif sebagai metode utama dan pendekatan kualitatif deskriptif sebagai pendukung. Eksperimen dilakukan dengan menguji Stable Diffusion v1.4 dan v1.5 dalam menghasilkan gambar wajah berdasarkan kompleksitas *prompt*. Kompleksitas *prompt* dirancang dalam sepuluh tingkat kompleksitas deskriptif, dari yang paling sederhana hingga kompleks.

Eksperimen dilakukan pada lingkungan sumber daya terbatas (*hardware host*) untuk mengamati respons model dan efisiensi penggunaan sumber daya komputasi. Setiap *prompt* digunakan untuk menghasilkan gambar wajah menggunakan *dataset CelebA* sebagai *ground truth*. Hasil generasi dievaluasi menggunakan metrik kualitas gambar dan semantik (*Face-FID*, *IS*, *SSIM*, dan *CLIP Score*) serta metrik efisiensi komputasi berupa waktu inferensi dan penggunaan CPU, GPU, dan RAM.

Pendekatan kuantitatif digunakan untuk menganalisis pola tren setiap metrik evaluasi, hubungan *trade-off* antar metrik evaluasi, sensitivitas *prompt*, dan perubahan model. Pendekatan kualitatif deskriptif digunakan untuk menginterpretasikan respons dan perilaku antar model berdasarkan hasil pengukuran metrik evaluasi.

## 3.2 Tahapan Penelitian

Penelitian ini meliputi lima tahapan yang terdiri dari beberapa uraian kegiatan serta dilengkapi indikator capaian yang disusun pada tabel 3.1.

Tabel 3. 1 Tahapan Penelitian

TAHAP PENELITIAN	URAIAN	INDIKATOR CAPAIAN
Studi Literatur	<ul style="list-style-type: none"> <li>• Kajian teori dan penelitian terkait.</li> <li>• Mengidentifikasi masalah penelitian.</li> </ul>	<ul style="list-style-type: none"> <li>• Memahami konteks penelitian dan bidang terkait.</li> <li>• Mengidentifikasi rumusan masalah, tujuan penelitian, dan <i>gap analysis</i>.</li> </ul>
Perancangan Eksperimen	<ul style="list-style-type: none"> <li>• Merancang lingkungan, kebutuhan, dan skenario eksperimen.</li> <li>• Menentukan variasi <i>prompt</i> dan <i>dataset</i>.</li> </ul>	<ul style="list-style-type: none"> <li>• Parameter evaluasi telah ditetapkan.</li> <li>• Alur penelitian (<i>flowchart</i>).</li> <li>• Variasi <i>prompt</i> dan <i>dataset</i> siap digunakan.</li> </ul>
Eksperimen	<ul style="list-style-type: none"> <li>• Mengimplementasikan model serta pengujian untuk memperoleh data untuk dievaluasi.</li> </ul>	<ul style="list-style-type: none"> <li>• Seluruh gambar berhasil di-<i>generate</i> dan disimpan.</li> <li>• Seluruh nilai pengukuran dicatat.</li> </ul>
Analisis Hasil Eksperimen	<ul style="list-style-type: none"> <li>• Mengukur dan mencatat metrik penelitian.</li> <li>• Menginterpretasikan hasil evaluasi.</li> <li>• Menilai pola <i>trade-off</i> dan respons setiap model berdasarkan hasil evaluasi.</li> </ul>	<ul style="list-style-type: none"> <li>• Nilai evaluasi sudah lengkap dan terhitung.</li> <li>• Perbedaan signifikan antarmodel berdasarkan variasi <i>prompt</i> telah teridentifikasi.</li> <li>• Menyampaikan interpretasi hasil analisis.</li> </ul>
Menarik Kesimpulan	<ul style="list-style-type: none"> <li>• Mengecek ulang akurasi hasil, keutuhan data, dan interpretasi hasil.</li> <li>• Menarik kesimpulan berdasarkan temuan penelitian.</li> </ul>	<ul style="list-style-type: none"> <li>• Hasil penelitian sudah terdokumentasi dengan baik dalam bentuk laporan skripsi dan artikel ilmiah.</li> </ul>

### 3.2.1 Studi Literatur

Studi literatur dilakukan untuk menunjang dasar pengetahuan dalam menginterpretasikan konteks penelitian. Tahap ini mencakup proses kajian terhadap berbagai sumber informasi, termasuk jurnal penelitian, situs web resmi, dan laporan penelitian yang relevan dengan konteks penelitian. Tujuan dari tahap ini, diantaranya pemahaman terkait ranah penelitian, metodologi, teknis, serta mengidentifikasi peluang penelitian.

### 3.2.2 Perancangan Eksperimen

#### 1. Persiapan *Research Environment*

Seluruh eksperimen setiap model dilakukan pada lingkungan yang sama untuk menjaga konsistensi hasil. Eksperimen dijalankan sepenuhnya di *Google Colab* dengan dukungan GPU untuk mempercepat proses *generate* gambar. Spesifikasi umum *research environment* yang digunakan diuraikan pada tabel 3.2.

Tabel 3. 2 Spesifikasi Umum *Research Environment*

Komponen	Spesifikasi
<i>Platform</i>	<i>Google Colab</i>
<i>GPU/Akselerator</i>	<i>NVIDIA Tesla T4 (16 GB VRAM)</i> (akses intermiten)
<i>CPU &amp; RAM</i>	2vCPU, RAM 12 GB
<i>Sistem Operasi</i>	<i>Default Google Colab</i>
<i>Bahasa Pemrograman</i>	<i>Python 3</i>
<i>Penyimpanan</i>	Lokal <i>Colab</i> (sementara) & <i>Google Drive</i> ( <i>data backup</i> )
<i>Pustaka Utama</i>	torch, diffusers, transformers, accelerate, numpy, pandas, Pillow, scikit-image, matplotlib, tqdm, clip
<i>Pustaka Pemantauan</i>	Psutil dan GPUUtil

Pemilihan *environment* dilakukan berdasarkan pertimbangan kompatibilitas dengan model generatif berbasis difusi, kemampuan pemrosesan GPU yang memadai, serta kemudahan integrasi dengan *Google Drive* untuk menyimpan hasil eksperimen.

#### 2. Persiapan *Dataset & Variasi Prompt*

##### 1) *Dataset*

Penelitian menggunakan *dataset CelebA (CelebFaces Attributes Dataset)* resmi sebagai *ground truth* untuk evaluasi kualitas dan kesesuaian gambar. Jumlah data *ground truth* yang diambil dari *dataset CelebA* sebanyak 500 dari 202.599 gambar asli. Pemilihan gambar dilakukan secara berurutan dan konsisten

berdasarkan urutan indeks terbawah *file* resmi *dataset CelebA*. *Dataset* kemudian disimpan di dalam direktori *Google Drive*. Seluruh pengujian kualitas gambar pada setiap model menggunakan *ground truth* yang sama untuk menjaga konsistensi hasil.

## 2) Variasi *Prompt* Uji

Penyusunan variasi *prompt* uji pada penelitian ini dilakukan melalui pendekatan pengelompokan sistematis yang mengacu pada penelitian (Liu & Chilton, 2022). Justifikasi pemilihan atribut semantik sebagai parameter kompleksitas didasarkan pada mekanisme *Text Encoder* (CLIP) dalam arsitektur LDM. Semakin banyak atribut yang diberikan, semakin padat ruang *embedding* yang harus diproses oleh model, sehingga meningkatkan potensi terjadinya ketidaksesuaian semantik.

Berdasarkan pendekatan ini, kompleksitas *prompt* dikategorikan menjadi tiga tingkat untuk menguji batas kemampuan model, yaitu sederhana (0 – 3 atribut) untuk menguji kemampuan dasar model dalam pembentukan fitur wajah tanpa banyak variabel, menengah (4 – 5 atribut) untuk menguji struktur wajah saat diberikan atribut pendukung, dan tinggi (6 atribut atau lebih) untuk menguji ambang batas pemahaman semantik model terhadap instruksi yang sangat detail.

Kategorisasi ini ditujukan untuk memberi landasan komparatif yang terukur dalam menganalisis hubungan antara kompleksitas *prompt* terhadap kualitas gambar dan efisiensi komputasi. Kompleksitas *prompt* yang diimplementasikan pada penelitian ini diuraikan pada tabel 3.3.

Tabel 3. 3 Susunan Kompleksitas Prompt

Prompt ID	Tingkat Kompleksitas	Prompt	Atribut
0	Sederhana	<i>A face</i>	1 atribut ( <i>subject</i> )
1	Sederhana	<i>A portrait photo of a woman</i>	2 atribut ( <i>framing, gender</i> )
2	Sederhana	<i>A portrait photo of a young woman</i>	3 atribut ( <i>framing, gender, age</i> )
3	Menengah	<i>A close-up portrait photo of a young woman</i>	4 atribut ( <i>framing, visibility, gender, age</i> )
4	Menengah	<i>A close-up portrait photo of a smiling young woman</i>	5 atribut ( <i>framing, visibility, gender, age, expression</i> )
5	Tinggi	<i>A close-up portrait photo of a smiling young woman, plain background</i>	6 atribut ( <i>framing, visibility, gender, age, expression, background</i> )
6	Tinggi	<i>A close-up portrait photo of a smiling young woman with black hair, studio lighting</i>	7 atribut ( <i>framing, visibility, gender, age, expression, hair color, lighting</i> )
7	Tinggi	<i>A close-up portrait of a smiling young woman with long black hair, neutral background, front-facing</i>	9 atribut ( <i>framing, visibility, gender, age, expression, hair length, hair color, background, pose</i> )
8	Tinggi	<i>A close-up portrait photo of a smiling young woman with long black hair, wearing a white shirt, soft lighting, neutral background</i>	10 atribut ( <i>framing, gender, age, expression, hair length, hair color, clothing, lighting, background</i> )
9	Tinggi	<i>A professional studio portrait of a smiling young woman with long black hair, wearing a white shirt, front-facing, soft lighting, neutral background</i>	11 atribut ( <i>framing, style studio professional, gender, age, expression, hair length, hair color, clothing, pose, lighting, background</i> )

### 3. Parameter Evaluasi & Metrik yang Digunakan

Evaluasi penelitian ini difokuskan pada dua aspek utama yaitu, kualitas gambar dan efisiensi komputasi. Kualitas gambar diukur menggunakan metrik *Face-FID* untuk mengukur kemiripan distribusi fitur wajah, *IS* untuk menilai keragaman dan kejelasan gambar, *SSIM* untuk mengukur kemiripan struktur, dan *CLIP Score* untuk mengukur kesesuaian semantik antara gambar dan *prompt*.

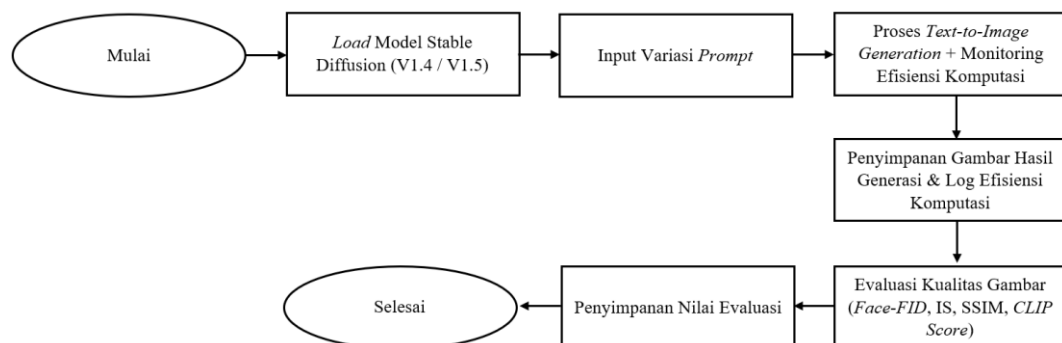
Efisiensi komputasi diamati melalui waktu inferensi untuk mengukur lama proses inferensi setiap model, *GPU usage* untuk menilai beban komputasi grafis yang diperlukan, *CPU usage* untuk menilai keterlibatan prosesor utama, dan penggunaan memori RAM untuk menilai kebutuhan sumber daya memori selama proses generasi gambar.

#### 3.2.3 Eksperimen

Tahap ini merupakan fase penerapan model untuk menghasilkan gambar berdasarkan variasi kompleksitas *prompt* serta melakukan pengukuran performa model melalui metrik kualitas gambar dan efisiensi komputasi.

##### 1. Flowchart Eksperimen

Alur proses eksperimen dijelaskan melalui *flowchart* pada Gambar 3.1.



Gambar 3. 1 *Flowchart* Eksperimen

Gambar 3.1 merepresentasikan urutan proses generasi gambar hingga diperoleh nilai evaluasi. Proses dimulai dengan *load* model Stable Diffusion v1.4 atau v1.5. Setelah model berhasil dimuat, variasi *prompt* uji dimasukkan sebagai input teks untuk proses generasi. Tahap berikutnya adalah proses *text-to-image generation*, yaitu proses model menghasilkan gambar berdasarkan masukkan *prompt* uji. Pada tahap ini dilakukan pemantauan efisiensi komputasi yang mencakup waktu inferensi, penggunaan CPU, GPU, dan RAM selama proses generasi berlangsung.

Setelah proses generasi selesai, gambar hasil generasi beserta log efisiensi komputasi disimpan. Selanjutnya dilakukan evaluasi kualitas gambar menggunakan metrik evaluasi yang sudah ditentukan. Nilai hasil evaluasi kemudian disimpan sebagai data evaluasi penelitian.

## **2. Implementasi Model dan *Generate Gambar***

Penelitian ini menggunakan model Stable Diffusion yang tersedia secara publik melalui repositori Hugging Face Hub (Wolf et al., 2020), yaitu Stable Diffusion v1.4 dengan *checkpoint* resmi (*CompVis/stable-diffusion-v1-4*) dan v1.5 (*runwayml/stable-diffusion-v1-5*). Pemilihan Stable Diffusion didasarkan pada beberapa pertimbangan. Model ini bersifat *open source*, sehingga memungkinkan transparansi metodologis serta fleksibilitas dalam pengaturan parameter dan lingkungan komputasi. Selain itu, model ini memiliki karakteristik batasan teknis yang jelas, seperti ketergantungan pada *prompt* tekstual dan sensitivitas terhadap kompleksitas *prompt*, yang relevan untuk studi evaluatif (Rombach dkk., 2022). Model ini juga didukung oleh ekosistem implementasi yang matang melalui

pustaka *diffusers*, yang menyediakan pipeline terstandarisasi untuk evaluasi kualitas gambar dan efisiensi komputasi (Hugging Face, 2023).

Implementasi model dilakukan menggunakan *pipeline diffusers*, diinstal pada perangkat *runtime* Google Colab dengan konfigurasi *precision* dinamis. Sistem otomatis mendeteksi ketersediaan GPU dan menetapkan *precision* float16 untuk efisiensi memori dan waktu inferensi. Jika GPU tidak tersedia, *pipeline* akan menggunakan *precision* float32 di CPU sebagai *fallback*.

Proses *generate* gambar dilakukan secara terkontrol pada lingkungan GPU, dengan pencatatan waktu inferensi, utilisasi GPU/CPU, dan penggunaan memori. Hal ini memungkinkan pemantauan efisiensi komputasi secara *real-time* selama *generate* gambar. Hasil *generate* disimpan secara terstruktur untuk memudahkan proses evaluasi. Algoritma inisialisasi model Stable Diffusion ditunjukkan tabel 3.4.

Tabel 3. 4 Algoritma Inisialisasi Stable Diffusion

```
ALGORITHM Inisialisasi_Stable_Diffusion
BEGIN
  DETEKSI Ketersediaan GPU
  IF GPU tersedia THEN
    SET device ← "cuda"
    SET precision ← float16
  ELSE
    SET device ← "cpu"
    SET precision ← float32
  END IF

  PILIH model Stable Diffusion (v1.4 atau v1.5)
  LOAD model dari repositori pretrained
  KONFIGURASI pipeline dengan precision dan device
  NONAKTIFKAN safety checker (opsional)
END
```

### 3. Pengumpulan Data

Proses pengumpulan data dilakukan melalui mekanisme eksperimen terkontrol menggunakan 10 variasi kompleksitas *prompt*. Merujuk pada (Henderson et al., 2018), penggunaan iterasi dalam jumlah yang konsisten pada setiap variabel uji sangat krusial dalam penelitian AI untuk memastikan bahwa hasil yang diperoleh lebih stabil dan meminimalisasi bias stokastik model.

Setiap *prompt* digunakan untuk 50 kali iterasi generasi, sehingga menghasilkan total sampel sebanyak 500 gambar untuk setiap versi model. Jumlah iterasi ini dipilih untuk menyeimbangkan antara kedalaman data eksperimen dan batasan operasional *computing* (Dodge et al., 2019). Penetapan ini dilakukan untuk menjamin seluruh variasi *prompt* mendapatkan porsi sumber daya yang setara, sehingga analisis kualitas gambar dan efisiensi komputasi menjadi lebih valid dan konsisten.

### 4. Evaluasi Hasil

Evaluasi eksperimen dilakukan pada dua aspek utama: kualitas gambar yang mencakup kualitas visual, performa generatif dan interpretasi semantik serta efisiensi komputasi mencakup waktu inferensi dan beban sumber daya *hardware* (CPU, GPU, RAM *usage*).

#### 1) Kualitas Gambar

Kualitas gambar dievaluasi menggunakan beberapa metrik yang komprehensif dan relevan, khususnya dengan objek wajah manusia. Perhitungan metrik dimulai dari *Face Fréchet Inception Distance (Face-FID)*, *Inception Score (IS)*, *Structural Similarity Index Measure (SSIM)*, lalu *CLIP Score*. Data setiap

metrik kualitas gambar dihasilkan dengan menghitung nilai rata-rata dari 50 iterasi pada setiap variasi *prompt*.

a) *Face Fréchet Inception Distance (Face-FID)*

Metrik ini mengukur kesamaan distribusi fitur wajah antara gambar asli (*CelebA*) dan gambar generatif. Pengukuran *Face-FID* dilakukan menggunakan model *InceptionResnetV1* dari *facenet-pytorch* yang telah dilatih pada *dataset VGGFace2*. Model ini digunakan untuk mengekstraksi fitur wajah dalam bentuk *embedding vektor numerik* berdimensi tinggi, yang mewakili struktur dan karakteristik wajah. Semakin rendah nilai *Face-FID*, semakin mirip distribusi fitur wajah gambar generatif dengan *ground truth*, menandakan kualitas sintesis yang lebih baik.

b) *Inception Score (IS)*

Metrik ini menilai kejelasan visual dan keragaman semantik berdasarkan prediksi kelas yang diprediksi oleh model *InceptionV3*. Metrik ini mengasumsikan bahwa gambar berkualitas tinggi akan menghasilkan distribusi prediksi kelas yang tajam (*low entropy* pada  $p(\mathbf{y}|x)$ ), sementara kumpulan gambar yang beragam secara semantik akan menghasilkan distribusi kelas marginal yang lebih merata ( $p(\mathbf{y})$ ). Maka dari itu, keragaman yang diukur oleh IS merepresentasikan keragaman semantik global, bukan variasi visual atau detail struktural gambar secara langsung. Nilai IS yang lebih tinggi mengindikasikan gambar yang lebih realistis dan keragaman semantik yang baik.

c) *Structural Similarity Index Measure (SSIM)*

Metrik ini mengukur kesamaan struktural antara gambar asli (*CelebA*) dan generatif pada tingkat struktural, luminansi, dan kontras. Nilai SSIM yang tinggi menandakan kesamaan dan stabilitas visual yang lebih baik.

d) *CLIP Score*

Metrik ini menilai kesesuaian semantik antara *prompt* dan gambar generatif menggunakan *Contrastive Language-Image Pre-Training (CLIP)*, dengan arsitektur *ViT-B/32*. Evaluasi *CLIP Score* dapat memetakan representasi teks dan gambar ke dalam ruang vektor yang sama, sehingga kemiripan antara keduanya dapat dihitung melalui *cosine similarity*. Nilai *CLIP Score* yang tinggi menunjukkan model mampu memahami dan mengeksekusi *prompt* dengan tepat.

**2) Efisiensi Komputasi**

Evaluasi efisiensi komputasi dilakukan selama proses generasi gambar pada lingkungan Google Colab dengan dukungan GPU. Pengukuran dilakukan secara *real-time* dengan memonitor waktu inferensi, serta penggunaan CPU, GPU, dan RAM. Seluruh metrik direkam pada setiap iterasi generasi gambar dan disimpan dalam format *log* terstruktur. Data setiap metrik efisiensi komputasi dihasilkan dengan menghitung nilai rata-rata dari 50 iterasi pada setiap variasi *prompt*.

Hasil pengukuran metrik evaluasi kualitas gambar dan efisiensi komputasi selanjutnya dianalisis untuk meninjau kualitas gambar, efisiensi komputasi, hubungan *trade-off* antara kualitas gambar dan efisiensi komputasi, serta respons setiap model terhadap kompleksitas *prompt*.

### **3.2.4 Analisis Hasil Penelitian**

Analisis hasil penelitian dilakukan untuk mengkaji kualitas gambar, efisiensi komputasi, hubungan *trade-off* antara kualitas gambar dan efisiensi komputasi, serta respons model terhadap kompleksitas *prompt*. Analisis ini fokus pada bagaimana kompleksitas *prompt* mempengaruhi kualitas gambar dan efisiensi komputasi yang dihasilkan oleh setiap model pada setiap metrik evaluasi.

#### **1. Analisis Kualitas Gambar**

Analisis kualitas gambar dilakukan dengan mengevaluasi hasil generasi gambar pada setiap variasi *prompt* menggunakan metrik kuantitatif yang relevan. Nilai metrik dianalisis secara komparatif antar tingkat kompleksitas *prompt* untuk mengamati perubahan kualitas visual, kesamaan struktural, serta kesesuaian semantik antara gambar dan deskripsi teks. Pendekatan ini digunakan untuk mengidentifikasi pola respons model terhadap peningkatan kompleksitas *prompt* dari aspek kualitas hasil generasi.

#### **2. Analisis Efisiensi Komputasi**

Analisis efisiensi komputasi dilakukan berdasarkan data waktu inferensi dan penggunaan sumber daya *hardware* (CPU, GPU, dan RAM) yang direkam selama proses generasi gambar. Analisis difokuskan pada perbandingan nilai rata-rata metrik efisiensi komputasi pada setiap variasi *prompt* untuk mengamati pengaruh kompleksitas *prompt* terhadap beban komputasi model. Hasil analisis ini digunakan untuk menilai konsistensi dan stabilitas performa komputasi selama proses generasi.

### **3. Analisis Trade-Off Model**

Analisis pola *trade-off* dilakukan untuk mengidentifikasi dan memahami hubungan *invers* antara kualitas gambar yang dihasilkan dan efisiensi komputasi pada setiap model yang diuji. Dalam konteks penelitian ini, analisis *trade-off* dilakukan dengan membandingkan hasil evaluasi kualitas gambar yang diukur menggunakan metrik evaluasi kualitas gambar dengan hasil pengukuran efisiensi komputasi berdasarkan metrik evaluasi efisiensi komputasi yang digunakan.

Pendekatan ini digunakan untuk meninjau dampak peningkatan kompleksitas *prompt* terhadap performa model, sehingga dapat diamati sejauh mana peningkatan kualitas gambar diikuti dengan biaya komputasi yang dikeluarkan. Melalui analisis ini, pola hubungan antara kualitas gambar dan efisiensi komputasi dapat diidentifikasi, termasuk kemungkinan adanya titik keseimbangan antara kedua aspek tersebut. Hasil analisis ini memberikan dasar untuk memahami karakteristik performa dan keterbatasan relatif dari setiap model yang diuji.

### **4. Analisis Respons Model terhadap Kompleksitas Prompt**

Analisis respons model bertujuan untuk mengamati perilaku dan sensitivitas Stable Diffusion v1.4 dan v1.5 terhadap peningkatan kompleksitas *prompt* berdasarkan aspek kualitas gambar dan efisiensi komputasi. Analisis respons model dilakukan melalui dua pendekatan, yaitu analisis sensitivitas *prompt* dan analisis perubahan performa antar model.

Analisis sensitivitas *prompt* diukur melalui nilai penyimpangan (%) setiap *prompt* terhadap rata-rata setiap metrik evaluasi sebagai *baseline* internal model.

Analisis ini dilakukan untuk mengidentifikasi *prompt* yang memiliki sensitivitas ekstrem (jauh dari rata-rata) di internal model. Analisis perubahan performa antar model diukur melalui persentase perubahan (%) dari rata-rata Stable Diffusion v1.4 dan v1.5 terhadap rata-rata Stable Diffusion v1.4 sebagai *baseline*. Analisis ini dilakukan untuk mengidentifikasi model yang memiliki kinerja yang lebih baik.

Batasan deviasi relatif dalam penelitian ini dikategorikan secara operasional untuk mempermudah interpretasi perubahan performa model dalam merespons variasi kompleksitas *prompt* maupun perbedaan karakteristik antar model. Penyimpangan <5% dikategorikan sebagai fluktuasi kecil (stabil), penyimpangan 5-10% sebagai fluktuasi moderat, dan penyimpangan >10% sebagai fluktuasi besar (sensitif) (Dodge et al., 2019). Melalui kategorisasi ini, variasi performa yang muncul akibat peningkatan kompleksitas *prompt* maupun perbedaan karakteristik model dapat diidentifikasi secara lebih sistematis.

Melalui analisis ini, respons model terhadap perubahan beban semantik *prompt* dapat diidentifikasi. Temuan ini dapat memberikan gambaran tingkat konsistensi, stabilitas, dan adaptabilitas model terhadap skenario penggunaan yang bervariasi, sehingga mendukung pemahaman yang lebih komprehensif mengenai performa relatif setiap model yang diuji.

### **3.2.5 Menarik Kesimpulan**

Kesimpulan ditarik berdasarkan hasil analisis yang telah dilakukan. Kesimpulan ini memberikan gambaran umum mengenai kinerja relatif dari berbagai model generatif *text-to-image* dalam menghasilkan sintesis wajah manusia

serta bagaimana variasi *prompt* mempengaruhi kualitas gambar, interpretasi semantik, serta efisiensi komputasi.