

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Kerangka Teori**

##### **2.3.1 Generative Adversarial Network**

Generative Adversarial Network (GANs) memiliki peran penting dalam upaya meningkatkan kualitas citra digital dalam analisis forensik digital. GANs memungkinkan perbaikan citra digital yang mungkin rusak atau cacat, sehingga menghasilkan bukti digital yang lebih berkualitas[10]. GANs digunakan untuk menciptakan citra digital yang sangat mendekati citra digital asli, yang dapat membantu memperkuat bukti dalam analisis forensik digital[11]. Untuk memaksimalkan manfaat GANs dalam hal perbaikan citra forensik digital, diperlukan pemahaman mendalam tentang teknik dan validasi model yang cermat. Penelitian terkait dan implementasi praktis adalah langkah penting dalam mengembangkan aplikasi GANs dalam konteks ini[12].

##### **2.3.2 Footage Forensik Digital**

Salah satu bukti jejak dalam forensik digital merujuk pada bukti elektronik atau digital yang ditinggalkan oleh aktivitas komputer atau perangkat elektronik salah satunya adalah citra digital[13][14]. Ini termasuk bukti yang dapat menjadi dalam penyelidikan kejahatan, karena mereka dapat membantu mengidentifikasi pelaku, mengungkap tindakan ilegal, dan menyediakan bukti yang kuat dalam pengadilan[15]. Analisis jejak digital melibatkan pengumpulan, pemeliharaan, dan pemahaman terhadap data digital yang dapat mengungkap pelanggaran atau kegiatan yang mencurigakan[16].

##### **2.3.3 Image Processing**

*Image processing* (pengolahan citra) adalah proses dilakukan dengan masukan (input) berupa citra (*image*) dan hasilnya (output) juga berupa citra (*image*). Tujuan utama dari *image processing* adalah

untuk meningkatkan, menganalisis, atau memahami informasi yang terkandung dalam citra digital[17]. Ini melibatkan serangkaian operasi komputasi seperti pemrosesan citra, pengolahan sinyal, dan analisis data untuk mengubah atau mengekstrak informasi dari citra tersebut. *Image processing* memiliki banyak aplikasi, termasuk di bidang computer vision, kedokteran, pemrosesan citra satelit, pengenalan pola, dan banyak lagi, serta dalam berbagai industri yang memanfaatkan analisis citra digital untuk keperluan pengembangan, diagnosis, atau pengawasan[18].

#### **2.3.4 OpenCV**

*Open Computer Vision* (OpenCV) merupakan *library open source* yang tujuannya dikhususkan untuk melakukan pengolahan citra. Maksudnya adalah agar komputer mempunyai kemampuan yang mirip dengan cara pengolahan visual pada manusia. OpenCV telah menyediakan banyak algoritma visi komputer dasar. OpenCV juga menyediakan modul pendeteksian objek yang menggunakan metode *computer vision*[19].

#### **2.3.5 Python**

Python adalah bahasa pemrograman yang populer dan sering digunakan dalam pengembangan software, aplikasi web, data science, machine learning, game, dan lainnya. Python memiliki sintaks yang mudah dipahami dan fleksibel, sehingga cocok digunakan oleh orang yang baru belajar bahasa pemrograman. Python dapat digunakan untuk mengotomatisasi tugas-tugas tertentu pada komputer serta membuat skrip yang dapat menjalankan tugas-tugas tertentu pada sistem operasi seperti Linux dan Windows. Python juga dapat digunakan dalam pengolahan citra, termasuk dalam meningkatkan kualitas citra digital[19][20].

#### **2.3.6 Google Colab**

Google Colab adalah layanan notebook Jupyter yang dihosting dan dapat digunakan tanpa penyiapan, serta menyediakan akses tanpa

biaya ke resource komputasi termasuk GPU. Colab memungkinkan siapa saja menulis dan mengeksekusi kode Python arbitrer melalui browser, dan sangat cocok untuk machine learning, analisis data, serta pendidikan.

Colab menawarkan beberapa keunggulan dibandingkan dengan solusi lokal, seperti aksesibilitas, kemudahan penggunaan, dan kekuatan komputasi. Colab dapat diakses dari mana saja dengan koneksi internet, tidak memerlukan instalasi atau konfigurasi perangkat lunak, dan menyediakan akses ke GPU dan TPU.

Fitur utama Google Colab meliputi dukungan kode Python, R, dan Julia; akses ke data dari Google Drive, GitHub, dan sumber lainnya; tools untuk visualisasi data secara interaktif; serta kemampuan untuk berkolaborasi dalam notebook yang sama[21].

### **2.3.7 Gaussian Blur**

Gaussian blur adalah teknik filter spasial yang umum digunakan untuk menghilangkan noise dan blur pada gambar dengan meredam transisi intensitas piksel yang tajam. Teknik ini didasarkan pada fungsi Gaussian yang berbentuk kurva lonceng simetris. Implementasinya dilakukan dengan konvolusi kernel filter Gaussian ke gambar asli, menghasilkan gambar yang lebih halus dan kabur. Manfaatnya termasuk menghilangkan noise, memperhalus gambar, dan mempermudah analisis gambar. Kekurangannya adalah hilangnya detail penting dan tidak efektif pada noise non-acak. Gaussian blur memiliki banyak aplikasi dalam pengolahan citra digital seperti pengurangan noise, image smoothing, edge detection, dan image enhancement[22].

## 2.2 Penelitian Terkait

Tabel 2.1 berikut ini adalah sampel yang mendekati tema penelitian deteksi ketersediaan parkir:

**Table 2. 1 Penelitian Terkait**

No	Judul	Penulis	Metode	Data set	Deskripsi data (kasus yang digunakan)	Kesimpulan dan Hasil
1.	Real-ESRGAN: Training Real-World Blind Super-Resolution with Pure Synthetic Data	Xintao Wang, Liangbin Xie, Chao Dong, Ying Shan. 2021	Generative Adversarial Network	dataset untuk pelatihan Real-ESRNet for 1000K iterations with learning rate $2 \times 10^{-4}$ while training Real-ESRGAN for 400K iterations with learning rate $1 \times 10^{-4}$ . dataset pengujian yang beragam	Real Esrgan menggunakan sample <i>image</i> anime, pemandangan dan hewan untuk menghasilkan citra yang lebih realistis dan detail.	Real-ESRGAN merupakan model peningkatan resolusi citra buram yang inovatif dan efektif. Model ini dilatih dengan data sintetis buatan, sehingga tidak membutuhkan data asli yang buram. Real-ESRGAN mampu menangani penurunan kualitas citra yang umum ditemui di dunia nyata, seperti ringing dan overshoot. Hasilnya, Real-ESRGAN dapat meningkatkan detail

				termasuk RealSR, DRealSR, OST300, DPED, validasi ImageNet, dan validasi ADE20K.		citra sambil menghilangkan noise pada sebagian besar citra buram. Namun, model ini masih memiliki keterbatasan, seperti garis-garis yang terdistorsi dan penguatan artefak di luar distribusi[23]
2.	Recovering Realistic Texture in Image Super-resolution by Deep Spatial Feature Transform	Xintao Wang, Ke Yu, Chao Dong, Chen Change Loy. 2018	Generative Adversarial Network,	Dalam pelatihan, sekitar 450 ribu citra digunakan setelah menghapus citra resolusi rendah dengan ukuran di bawah 30kB. Sample yang diuji coba sebanyak 300 citra.	Dalam pengujian, citra-citra yang diuji coba adalah citra dari kategori luar ruangan, seperti langit, gunung, tanaman, rumput, air, hewan, dan bangunan.	Metode SFT-GAN menggunakan peta probabilitas segmentasi untuk memodulasi peta fitur dalam jaringan super-resolusi, menghasilkan tekstur yang lebih alami dan realistis dalam citra yang dihasilkan. Metode ini berhasil mengungguli metode berbasis PSNR dan metode

						berbasis GAN dalam hal kualitas visual, seperti yang ditunjukkan dalam studi pengguna. Pendekatan ini dievaluasi pada adegan luar ruangan dan menunjukkan hasil yang menjanjikan. Meskipun demikian, penelitian ini masih berfokus pada adegan luar ruangan, dan penelitian masa depan bertujuan untuk mengatasi kekurangan dan mengeksplorasi perbaikan bersama dengan segmentasi.[24]
3.	Blind Image Quality Assessment	Hongtao yang, ping shi1, dixiu	model BIQA-GAN yang	Total 1,162 citra alami yang distorsi, di mana	<i>Image</i> yang ter-distorsi dari penelitian ini termasuk kebisingan	penelitian ini berhasil mengusulkan metode penilaian kualitas citra

	of Natural Distorted Image Based on Generative Adversarial Networks	zhong, dapan, and zefeng ying, 2019	berbeda, yaitu NQM-GAN, NWQM-GAN, dan NQS-GAN	80% dari kumpulan data (sekitar 930 citra) digunakan sebagai set pelatihan, dan sisanya 20% (sekitar 232 citra) sebagai set uji. Untuk meningkatkan	dengan pencahayaan rendah, gangguan pencahayaan berlebih, kesalahan kompresi, dan jenis distorsi baru yang dihasilkan dari campuran jenis yang berbeda.	menggunakan Generative Adversarial Networks (GAN) yang mampu memprediksi skor kualitas citra dengan baik, terutama untuk citra-citra yang mengalami distorsi alami. Model NQS-GAN terbukti memiliki kinerja yang superior dan kompleksitas komputasi yang lebih rendah dibandingkan dengan model-model yang sudah ada. Metode ini juga mampu mengatasi tantangan dalam mengevaluasi citra-citra yang mengalami distorsi alami dan memiliki kinerja generalisasi yang baik[25].
--	---	-------------------------------------	---	---	---	--

4.	Image Quality Improvement of Hand-held Ultrasound Devices with a Two-stage Generative Adversarial Network	Zixia Zhou, Yuanyuan Wang, Yi Guo, Yanxing Qi and Jinhua Yu, 2018	Generative Adversarial Network	Ada 112 pasang data simulasi, 80 data phantom dan 188 pasang data klinis. Data transfer learning PWI terdiri dari 13.505 pasang.	Penelitian ini menguji kualitas citra ultrasonografi portabel menggunakan model jaringan generatif musuh (GAN) dua tahap. Mereka menguji citra ultrasonografi berkualitas rendah dan berkualitas tinggi dari berbagai sumber, termasuk data fantom, data klinis, dan data transfer learning PWI.	Hasil eksperimen memvalidasi keberhasilan metode ini, mendukung penggunaan perangkat USG genggam untuk diagnosis yang akurat dalam pengobatan komunitas dan telemedis[26].
5.	Towards Real-World Blind Face Restoration with	Xintao Wang, Yu Li, Honglun Zhang, Ying Shan.	Generative Facial Prior (GFP)-GAN	70.000 gambar digunakan dalam training	Kasus yang digunakan dalam penelitian ini adalah restorasi wajah buta secara nyata menggunakan Generative	GFP-GAN framework menggunakan generative facial prior untuk restorasi wajah buta, dan mampu mengungguli metode

	Generative Facial Prior	2021			<p>Facial Prior (GFP) yang terdapat dalam pretrained face GAN. GFP-GAN memungkinkan restorasi wajah yang realistis dan setia dengan menggunakan lapisan transformasi fitur spasial.</p>	<p>sebelumnya dalam restorasi wajah bersamaan dan peningkatan warna untuk gambar dunia nyata. Metode dievaluasi menggunakan berbagai metrik perseptual dan piksel, dan perbandingan kualitatif menunjukkan kinerja yang lebih unggul. Studi ablation menyoroti pentingnya lapisan modulasi spasial, generative facial prior, dan kerugian restorasi piramida.</p>
--	-------------------------	------	--	--	---	---

### 2.3 Penerapan Algoritma

Fungsi *Loss* untuk melatih jaringan *generator* dan *discriminator* secara bersamaan, digunakan *adversarial loss* yang merupakan fungsi loss utama dari metode GANs[27]. *Discriminator* D dilatih untuk memaksimalkan probabilitas untuk menentukan label yang benar untuk data asli maupun data hasil *generator* G, sedangkan *generator* G dilatih untuk meminimalisasi  $\log(1 - D(G(x)))$  dengan  $D(x)$  merupakan probabilitas data  $x$  merupakan citra asli.  $Z$  mempresentasikan citra beresolusi rendah dan  $G(z)$  hasil rekonstruksi. Secara matematis fungsi adversarial loss adalah sebagai berikut:

$$\frac{\min}{C} \frac{\max}{D} E_{x \sim P_{data(x)}} [\log D(x)] + E_{z \sim P_z(z)} [\log(1 - D(G(x)))] \quad (1)$$

Keterangan:

- $\frac{\min}{C}$  : mengindikasikan bahwa tujuan adalah untuk mencari nilai minimum dalam hal objektif yang dibagikan oleh "C."
- $\frac{\max}{D}$  : menunjukkan bahwa dalam proses pelatihan GANs, kita mencari nilai maksimum dalam hal objektif yang dibagikan oleh "D."
- $E_{x \sim P_{data(x)}} [\log D(x)]$  : mengukur seberapa baik *discriminator* D membedakan data asli  $x$  dari data palsu yang dihasilkan oleh *generator*. Tujuan *generator* adalah menghasilkan data yang D tidak dapat membedakan dengan baik, sehingga nilai ini lebih tinggi.
- $E_{z \sim P_z(z)} [\log(1 - D(G(x)))]$  : mengukur seberapa baik *generator* G menghasilkan data palsu yang dapat "mengelabui" *discriminator* D. Tujuan

D adalah meminimalkan nilai ini sehingga D lebih baik dalam membedakan data asli dan palsu.

Formulasi tersebut memungkinkan *generator* untuk menghasilkan citra yang mirip dengan citra asli dan sulit untuk dibedakan oleh *discriminator*, dan untuk meningkatkan kualitas citra hasil rekonstruksi, digunakan fungsi *loss* tambahan yaitu *pixel-wise loss*, *feature-wise loss* dan *PSNR loss*.

*Pixel-wise Loss*. Untuk membuat nilai piksel citra hasil rekonstruksi menjadi mirip dengan *ground truth*, dihitung kemiripan antara nilai piksel citra hasil rekonstruksi dengan citra aslinya. Jika  $\{z_t, i = 1..N\}$  merupakan citra beresolusi rendah dan  $\{z_t, i = 1..N\}$  merupakan *ground-truth*, maka kemiripan dihitung dengan:

$$\frac{1}{N} \sum_{t=1}^N \sqrt{(G(z_i) - x_i)^2} \quad (2)$$

- $\frac{1}{N}$  : faktor normalisasi yang menghitung rata-rata dari kuadrat selisih antara rekonstruksi *generator* dan data asli dari N sampel.
- $\sum_{t=1}^N$  : simbol yang menunjukkan penjumlahan dari t=1 hingga N, di mana N adalah jumlah sampel yang digunakan dalam evaluasi.
- $\sqrt{\quad}$  : simbol akar kuadrat
- $G(z_i)$  : hasil dari *generator* G ketika diberikan data acak  $z_i$ . *Generator* G mencoba menghasilkan data yang menyerupai data asli  $x_i$ .
- $x_i$  : data asli yang digunakan sebagai referensi atau target yang ingin dicocokkan oleh *generator*.
- $(G(z_i) - x_i)$  : selisih antara data yang dihasilkan oleh *generator* dan data asli. Ini mengukur sejauh mana rekonstruksi oleh *generator* mendekati data asli.

*Feature-wise Loss.* Untuk memperoleh hasil yang lebih realistis, dihitung jarak antara fitur dari citra hasil. Fitur citra didapatkan dari hasil fungsi aktivasi layer konvolusi ke-5 jaringan *discriminator*, fitur tersebut mempresentasikan abstraksi informasi struktural dari suatu citra. Berbeda dengan *feature matching* [28]. Jarak fitur diukur menggunakan *L2 distance* sebagaimana yang digunakan pada formulasi (2), didefinisikan sebagai:

$$\frac{1}{N} \sum_{t=1}^N \sqrt{\Phi_{\theta}(G(z_i)) - \Phi_{\theta}(x_i)}^2 \quad (3)$$

Dengan  $\Phi_{\theta}(x)$  merupakan fitur yang di ekstraksi dari  $x$ . formulasi tersebut akan membuat citra hasil rekonstruksi memiliki fitur yang mirip dengan fitur citra asli.

- MSE* : merupakan metrik yang digunakan untuk mengukur kesalahan atau perbedaan antara dua matriks atau citra.
- N* : jumlah pengamatan atau sampel yang digunakan dalam perhitungan MSE.
- $\Sigma$  : penjumlahan atau sumasi dari perbedaan kuadrat yang akan dihitung.
- $1/(mn)$  : faktor normalisasi yang digunakan dalam perhitungan. Ini adalah kebalikan dari perkalian dari jumlah elemen dalam matriks "m" dan "n," yang menghasilkan rata-rata perbedaan kuadrat.
- m dan n : ukuran dari matriks atau citra  $G(z_t)$  dan  $(j_x) \cdot x_1(j_x)$ . Mereka mewakili jumlah baris dan kolom dalam matriks tersebut.
- i dan k : indeks yang digunakan untuk mengakses elemen-elemen

*PSNR Loss*. untuk mencegah *noise* pada citra hasil rekonstruksi, digunakan *PSNR loss* yang didefinisikan sebagai :

$$MSE = \frac{1}{N} \sum_{t=1}^N \frac{1}{m \cdot n} \sum_{i=0}^{m-1} \sum_{k=0}^{n-1} [G(z_t)_{(jx)} - x_{1(jx)}]^2 \quad (4)$$

matriks.

- $G(z_t)$  : matriks atau citra yang mewakili data pengamatan atau sampel.
- $(jx)$ - : matriks atau citra lain yang digunakan sebagai pembanding.
- $x_{1(jx)}$  : Ini mungkin merupakan citra referensi atau citra target yang ingin Anda bandingkan dengan data pengamatan.

$$PSNR = 10 \log_{10} \left( \frac{MAX^2}{MSE} \right) \quad (5)$$

- PSNR** : singkatan dari Peak Signal-to-Noise Ratio, yang merupakan metrik yang digunakan untuk mengukur kualitas citra. PSNR menghasilkan nilai dalam desibel (dB) dan mengindikasikan sejauh mana citra yang telah diproses atau dikompresi mendekati citra asli.
- 10** : faktor konversi yang digunakan untuk mengubah hasil perhitungan menjadi skala desibel.
- $\log_{10}$  : fungsi logaritma dengan basis 10. Ini digunakan untuk menghitung perbandingan antara kuadrat nilai maksimum ( $MAX^2$ ) dan Mean Squared Error (MSE).

$MAX_i^2$  : nilai maksimum yang dapat diwakili oleh piksel dalam citra.

$MSE$  : kesalahan kuadrat rata-rata antara citra asli dan citra yang telah mengalami pemrosesan atau kompresi.

Dengan  $m$  dan  $n$  adalah lebar dan tinggi citra, dan  $MAX$ , adalah nilai pixel maksimum. *Generator* dilatih untuk memaksimalkan nilai PSNR