BAB II
TINJAUAN PUSTAKA

2.1 Kajian Literatur

Penelitian terkait tentang perbaikan citra digital sudah cukup banyak. Berikut adalah beberapa penelitian terkait perbaikan citra digital

Putut Hendra Wijaya, (2021) dalam penelitiannya yang berjudul Perbaikan Citra Dengan Menggunakan Metode *Gaussian* Dan *Mean* Filter, penelitian ini bertujuan untuk mengurangi *noise* pada citra digital dengan menggunakan filter *Gaussian* dan *Mean*. Penggunaan kedua metode ini efektif untuk menekan *noise* yang ada pada citra yang menjadi sampel karena berhasil mengurangi *noise* yang ada. Proses dilakukan dalam dua tahap, pertama dengan filter *Gaussian* kemudian Filter *mean*. Hal ini efektif jika hanya untuk mengurangi *noise* yang ada.

Barany Fachry. 2018. Dalam penelitiannya yang berjudul aplikasi perbaikan citra efek *noise salt* & *papper* menggunakan metode *contraharmonic mean* filter. Penelitian ini bertujuan untuk membangun sebuah aplikasi yang mampu memperbaiki kualitas citra digital menggunakan metode *Contrahamonic Mean filter*. Dengan menggunakan metode ini, kualitas citra berhasil di perbaiki dengan menghilangkan noise yang ada. *Noise* yang berhasil dihilangkan hanya terbatas salt&*papper*.

Yustika Septiani Muzahardin. 2022. Dalam penelitiannya yang berjudul perbaikan citra digital pada foto dengan menggunakan metode *retinex*. Penelitian ini bertujuan untuk menerapkan algoritma *retinex* dalam memperbaiki kualitas citra digital. Filter yang digunakan adalah *Gaussian* dan dalam prosesnya berhasil mendapatkan kualitas yang lebih baik dari citra awal setelah dilakukan proses perbaikan dengan menggunakan metode *retinex*. Sehingga sistem dapat digunakan dalam memperbaiki dan meningkatkan kualitas citra pada foto buram.

Tito Sugiharto. 2018 dalam penelitiannya yang berjudul rancang bangun aplikasi perbaikan citra digital menggunakan proses konvolusi. Penelitian ini bertujuan untuk menghasilkan suatu aplikasi perangkat lunak untuk meningkatkan kualitas citra digital dengan menggunakan proses konvolusi. Target penelitian yang ingin dicapai adalah adanya aplikasi perbaikan kualitas citra digital untuk file citra digital bertipe *jpg* dan *bmp*. Proses konvolusi dapat digunakan dalam proses perbaikan kualitas citra digital sehingga dengan proses konvolusi ini dapat dihasilkan citra digital dengan kualitas yang lebih baik. Penelitian ini dibangun menggunakan bordlan delphi 7. Berikut adalah penelitian terdahulu yang dapat dilihat pada Tabel 2.1.

Tabel 2.1 Penelitian Terkait

No	Judul	Peneliti & Tahun	Filter	Algoritma	Hasil
1	Perbaikan Citra Dengan Menggunakan Metode Gaussian Dan Mean Filter	Putut Hendra Wijaya, Resty Wulanningrum ,Risa Halilintar. 2021	Gaussian Dan Mean	-	Metode yang digunakan lumayan baik untuk membersihkan noise atau derau yaitu dengan menggunakan metode Mean filter
2	Aplikasi Perbaikan Citra Efek Noise Salt & Papper Menggunakan Metode Contraharmonic Mean Filter	Barany Fachri. 2018	contraharmonic mean filter		Dengan menggunakan metode ini, kulitas citra berhasil di perbaiki dengan menghilangkan noise yang ada. Noise yang berhasil dihilangkan hanya terbatas salt&papper.

Tabel 2.1 Penelitian Terkait (Lanjutan)

No	Judul	Peneliti & Tahun	Filter	Algoritma	Hasil
3	Perbaikan Citra Digital Pada Foto Dengan Menggunakan Metode Retinex	Yustika Septiani Muzahardin,Achmad Fauzi, Nurhayati. 2022	Gaussian	Retinex	mendapatkan hasil citra baru dengan kualitas yang lebih baik dari citra awal setelah dilakukan proses perbaikan dengan menggunakan metode retinex. Sehingga sistem dapat digunakan dalam memperbaiki dan meningkatkan kualitas citra pada foto buram
4	Rancang Bangun Aplikasi Perbaikan Citra Digital Menggunakan Proses Konvolusi	Tito Sugiharto,S.Kom., M.Eng	Domain Spasial	Konvolusi	Proses knvolusi dapat digunakan dalam proses perbaikan kualitas citra digital sehingga dengan proses konvolusi ini dapat dihasilkan citra digital dengan kualitas yang lebih baik.
5	Menginvestigasi Kemampuan GenerativeAdversarial Network	Ferianda Satya (2019)			Implementasi penggunaan GAN dalam pengolahan citra digital yang dapat membangkitkan citra baru. Terbatas pada Algoritma GAN.

Tabel 2.1 Penelitian Terkait (Lanjutan)

No	Judul	Peneliti & Tahun	Filter	Algoritma	Hasil
6	Analisis Restorasi Citra Kabur Algoritma Wiener Menggunakan Indeks Kualitas Citra	Nur Afiyat (2017)		Wiener	restorasi citra menggunakan algoritma Wiener dengan citra masukan yang terdegradasi motion blur dan Gaussian blur.Kualitas citra hasil restorasi dianalisis menggunakan indeks kualitas citra, dengan cara membandingkan citra hasil restorasi terhadap citra asli sebagai referensi
7	Implementasi Noise Removal Dan Image Restoration Pada Citra Kabur (Blur) Dengan Menggunakan Metode Lucy – Richardson Algorihm Techniques	Dicky Rachmat Fauzi, Anggunmeka Luhur Prasasti, Ashri Dinimaharawati (2021)		Lucy – Richardson Algorihm Techniques	Deteksi Masker Wajah dengan Single Shot Detector (SSD) dan RCNN. Lucy Richardson bisa memulihkan citra dari citra yang blur, tetapi terdapat artefak yang ada pada citra yang blur.

Berdasarkan penelitian terdahulu dalam tabel di atas yang terkait dan merupakan rujukan pada penelitian yang penulis akan lakukan. Dan peneliti bertujuan untuk menerapkan penggunaan Super *resolution* dalam perbaikan citra digital menggunakan *Gerative Adversarial Network*.

2.2 Jupyter Notebook

Menurut (Wikipedia) Jupyter Notebook adalah Layanan kerja antarmuka notebook berbasis web yang interaktif untuk membuat sebuah Notebook. Jupyter Notebook bekerja dengan berbasis web yang berisi sel *input*/*output* yang secara berurutan dapat diisi kode, teks, perhitungan matematika, plot, media. Dan menjalankan *Kernel* berbahasa Python. Antarmuka *Notebook* atau komputasi *notebook* adalah antarmuka khusus untuk pemrograman yang memiliki ciri berbentuk baris berurutan untuk proses dan keluarannya. Dan juga merupakan sebuah metode penulisan kode dalam pemrograman. Berikut merupakan antarmuka dari Juypiter notebook. Dapat dilihat pada gambar 2.1

Gambar 2.1 Interface Juypiter notebook.

2.3 Super Resolution

Menurut (Xintao Wang, 2021) *Super Resolution* adalah sebuah metode dalam pengolahan citra digital yang mampu merekonstruksi citra dengan resolusi rendah dan detail minim menjadi citra resolusi tinggi dengan detail yang lebih kentara.

Teknik citra super resolusi adalah salah satu teknik untuk mendapatkan citra yang beresolusi tinggi dari sekumpulan citra yang beresolusi rendah. Resolusi tinggi yang dihasilkan dapat berupa citra tunggal atau lebih. Citra resolusi tinggi didapat

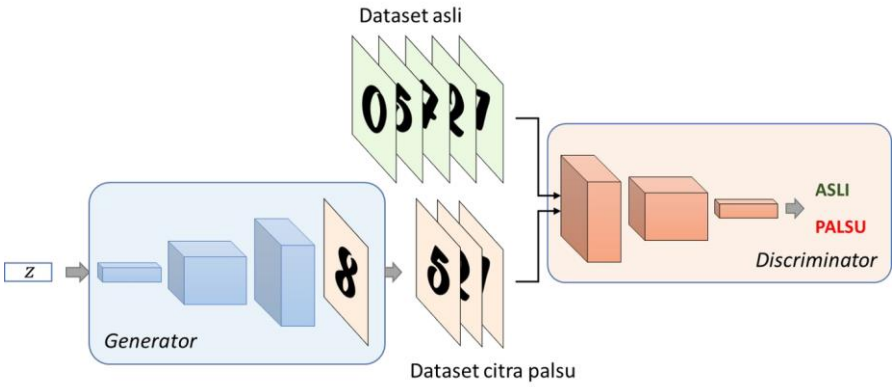
dari sekumpulan resolusi rendah yang diambil dari scene (adegan) yang sama. Karena dari scene yang sama akan menyediakan informasi yang mungkin dapat digunakan untuk merekonstruksi citra resolusi tinggi (S. Chauduri, N.P. Galatsanos and B.C. Tom, 2001). Berdasarkan output yang dihasilkan (High Resolution), super resolusi dibedakan menjadi 2, yaitu super resolusi statis dan super resolusi dinamis. Super resolusi statis adalah metode super resolusi yang menghasilkan citra keluaran resolusi tinggi tunggal dan super resolusi dinamis adalah metode super resolusi yang menghasilkan citra keluaran resolusi tinggi yang lebih dari satu (Robert Ari Sand. 2009). Super resolusi secara umum terdiri dari tiga tahap algoritma yaitu registrasi, interpolasi dan rekonstruksi (S. Chauduri, N.P. Galatsanos and B.C. Tom, 2001).

Secara umum arsitektur GAN terdiri dari 2 jaringan yang disebut sebagai jaringan Generator dan Discriminator. Bentuk jaringan Generator dapat dilihat berkebalikan dengan struktur jaringan saraf pada umumnya. Jaringan Generator menerima input sebuah vektor angka z, kemudian mengubahnya menjadi output gambar tiga dimensi. Vektor input z umumnya dibangkitkan secara acak, lalu dari angka sembarang tersebut Generator membangkitkan gambar yang juga sembarang

2.4 Generative Adversarial Network

Generative Adversarial Network adalah sebuah arsitektur jaringan saraf tiruan yang bertujuan untuk membentuk atau membangkitkan suatu data yang benar-benar baru, dari tidak ada menjadi ada. GAN terdiri dari dua model, yaitu Generator dan Discriminator. Model Generator dilatih untuk membuat data baru berdasarkan data training. Pelatihan model Generator dilakukan untuk memaksimalkan kemungkinan model Discriminator melakukan kesalahan. Lalu, model Discriminator dilatih untuk membandingkan data training dengan data dari model Generator. Model Discriminato rdapat membedakan data yang asli dengan data yang baru dari model Generator. Algoritma GAN berhasil ketika model Generator tidak dapat membedakan data asli dengan data baru. Umumnya target utama dari GAN adalah data citra. Secara singkat, jaringan GAN dilatih untuk mampu membangkitkan suatu gambar baru berdasarkan kumpulan gambar yang

telah ia lihat sebelumnya selama proses pelatihan. (Ferdianda Satya 2019). Berikut merupakan gambar Arsitektur GAN yang dapat dilihat pada gambar 2.2.



Gambar 2.2 Arsitektur GAN

Jaringan *Discriminator* merupakan jaringan biner yang menerima *input* gambar tiga dimensi dan mengeluarkan klasifikasi menyatakan *input* gambar adalah gambar asli dari *dataset* atau merupakan gambar buatan Generator. *Discriminator* dilatih dengan sekumpulan data yang dibangkitkan oleh Generator, dan sekumpulan data dari *dataset*, dan dilatih untuk bisa membedakan keduanya Gambar-gambar Generator yang berhasil dikenali *Discriminator* sebagai gambar "palsu", akan dikembalikan sebagai *feedback* pada jaringan Generator. Tugas Generator kini adalah dilatih untuk bisa membuat sekumpulan gambar palsu, yang saat dilihat oleh *Discriminator*, *Discriminator* tidak bisa membedakan antara asli dan palsunya. [Ferianda Satya, 2019]

Inferensi dilakukan dengan tujuan mengembalikan citra resolusi tinggi dari citra yang beresolusi rendah karena degradasi yang kompleks. Menggunakan metode degradasi klasik. Secara umum cara kerjanya dengan mengolah gambar asli y yang diberikan pengolahan blurkernel k. Kemudian operasi downsampling dengan faktor skala r dilakukan, kemudian citra resolusi rendah x didapatkan dengan menambahkan noise n. dan akhirnya file JPEG yang terkompresi digunakan.
$X = D(y) = [y*k)\downarrow r + n]JPEG$. Dimana D menunjukkan degradasi.

Noise yang digunakan adalah *Gaussian noise* dan *poission*. *Gaussian noise* merupakan model *noise* yang mengikuti distribusi normal standar dengan rata-rata 0 dan standar deviasi 1. Efek dari *Gaussian Noise* ini pada gambar adalah munculnya titik-titik berwarna yang jumlahnya sama dengan persentase *noise*.
Resize (*downsampling*) operasi *bilinear* and *bicubic* .

Kompresi JPEG adalah teknik kompresi *lossy* yang umum digunakan untuk gambar digital. Ini pertama-tama mengubah gambar menjadi ruang warna YCbCr dan menurunkan sampel saluran chroma. Gambar kemudian dibagi menjadi 8 × 8 blok dan setiap blok ditransformasikan dengan dua dimensi discrete cosinus transform (DCT), diikuti dengan kuantisasi koefisien DCT. Artefak blok yang tidak menyenangkan biasanya diperkenalkan oleh kompresi JPEG.
Kualitas gambar terkompresi ditentukan oleh faktor kualitas q [0, 100], di mana q yang lebih rendah menunjukkan rasio kompresi yang lebih tinggi dan kualitas yang lebih buruk.

Dalam proses inferensi dan implementasi ini akan didapati beberapa hal yang berbeda dari setiap gambar. Seperti waktu proses, dan kapasitas Vram dari GPU. Semakin besar ukuran gambar yang akan diproses, semakin banyak resource yang dibutuhkan. Dalam kasus ini dengan Vram 6Gb dan CUDA Cores sebanyak 1536 pun masih mendapkan kendala berupa VRAM penuh unutk gambar yang berukuran lebih dari batas maksimal. Ukuran VRAM ini jua dipengaruhi dengan aplikasi background yang berjalan dibelakang dan menggunakan VRAM. Maka penting untuk mengubah mode dari sistem operasi Pop!OS menjadi Compute Mode. Compute mode memungkinkan aplikasi background diminamilir san dofokuskan ke aplikasi yang membutuhkan resource besar. Gambr berikut menunjukan proses inverensi sekaligus kondisi VRAM dan GPU Utilization.

2.5 Python

Menurut (Pazriyah 2016) Python adalah sebuah bahasa pemrograman yang bisa digunakan pada beberapa platform (*multiplatform*), dan bersifat sumber perangkat bebas terbuka (*opensource*), pertama kali dikembangkan oleh Guido van Rosuum

pada tahun 1990 di CWI, Belanda. Bahasa ini dikategorikan sebagai bahasa tingkat tinggi (*very-high-level language*) dan merupakan bahasa berorientasi objek yang dinamis (*object-oriented-dynamic language*). Hal utama yang membedakan Python dengan bahasa lain adalah dalam hal aturan penulisan kode program. Python memiliki aturan yang berbeda dengan bahasa lain, seperti indentasi, tipe data, *tuple*, dan *dictionary*. Python adalah bahasa pemrograman dinamis yang mendukung pemrograman berorientasi obyek. Python dapat digunakan untuk berbagai keperluan pengembangan perangkat lunak dan dapat berjalan di berbagai sistem operasi seperti Linux, Windows, Unix, Symbian dan masih banyak lagi.

2.6 Nvidia CUDA

Menurut (Wikipedia) CUDA (*Compute Unified Device Architecture*) adalah platform komputasi paralel dan antarmuka pemrograman aplikasi (API) yang memungkinkan perangkat lunak menggunakan jenis unit pemrosesan grafis (GPU) tertentu untuk pemrosesan tujuan umum, suatu pendekatan yang disebut komputasi tujuan umum pada GPU (General Purpose GPU). CUDA adalah lapisan perangkat lunak yang memberikan akses langsung ke set instruksi virtual GPU dan elemen komputasi paralel, untuk eksekusi kernel komputasi.

CUDA adalah platform komputasi paralel dan model pemrograman yang memungkinkan peningkatan dramatis dalam kinerja komputasi dengan memanfaatkan kekuatan unit pemrosesan grafis (GPU) dan secara eksklusif terdaftar sebagai merek dan produk dari Nvidia.

2.7 Prototype

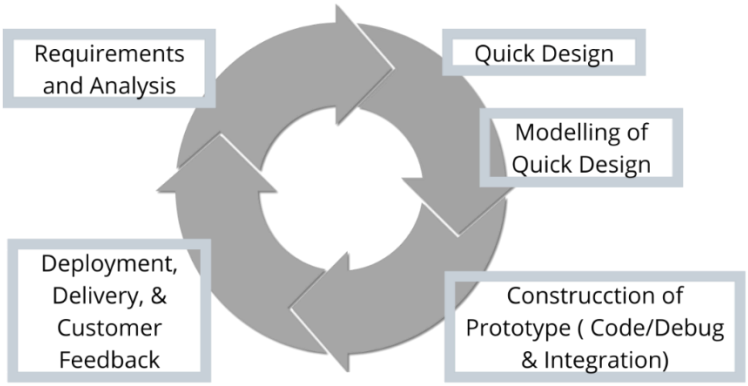
Menurut (Pressman, 2012) Prototype merupakan metode proses pembuatan model sederhana *software* yang mengizinkan pengguna memiliki gambaran dasar tentang program serta melakukan pengujian awal. Prototipe memberikan fasilitas bagi pengembang dan pemakai untuk saling berinteraksi selama proses pembuatan, sehingga pengembang dapat dengan mudah memodelkan perangkat lunak yang akan di buat. Metode ini cocok digunakan untuk mengembangkan sebuah perangkat lunak yang dikembangkan kembali. Metode ini dimulai dengan pengumpulan kebutuhan pengguna. Kemudian membuat sebuah rancangan kilat yang selanjutnya

akan dievaluasi kembali sebelum di produksi secara benar. Prototipe bukanlah merupakan sesuatu yang lengkap, tetapi sesuatu yang harus dievaluasi dan dimodifikasi kembali. Segala perubahan dapat terjadi pada saat prototipe dibuat untuk memenuhi kebutuhan pengguna dan saat yang sama memungkinkan pengembangan untuk lebih memahami kebutuhan pengguna secara baik.

Terdapat beberapa tahapan dalam pengembangan *prototype* antara lain:

1. Komunikasi (*Communication*) dan pengumpulan data awal, yaitu komunikasi dengan klien dan user untuk menentukan kebutuhan.
2. Perencanaan cepat (*Quick Plan*), yaitu pembuatan perencanaan analisis terhadap kebutuhan pengguna.
3. Pemodelan perancangan cepat (*Modeling Quick Design),* yaitu membuat rancangan desain program.
4. Pembentukan *prototype* (*Construction of prototype),* yaitu pembuatan aplikasi berdasarkan dari pemodelan desain yang telah dibuat.
5. Penyerahan sistem dan umpan balik *(Development Delevery and Feedback*), yaitu memproduksi perangkat secara benar sehingga dapat digunakan oleh pengguna.

Berikut merupakan paradigma pembuatan model Prototype yang dapat dilihat pada gambar 2.3



Gambar 2.3 Paradigma pembuatan Model Prototype