

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Dalam dunia pengolahan citra digital, noise atau gangguan visual pada gambar merupakan salah satu permasalahan utama yang dapat menurunkan kualitas informasi visual. Salah satu jenis noise yang umum dijumpai adalah Gaussian Noise, yang muncul akibat gangguan acak dari sensor kamera atau transmisi data digital. Noise jenis ini menyebabkan distorsi berupa bintik-bintik acak pada citra yang mengganggu proses identifikasi, segmentasi, hingga pengenalan objek dalam gambar.

Permasalahan yang sering terjadi adalah kurangnya kemampuan aplikasi mobile dalam menangani noise secara efisien dan waktu yang nyata. Banyak aplikasi pemrosesan gambar berbasis mobile yang belum menyediakan fitur filtering noise secara optimal, khususnya untuk Gaussian Noise. Hal ini menjadi tantangan tersendiri mengingat perangkat mobile memiliki keterbatasan dalam hal daya komputasi dan penyimpanan dibandingkan dengan sistem desktop.

Untuk menyelesaikan permasalahan ini, penelitian ini mengusulkan penerapan metode Midpoint sebagai solusi penyaringan citra untuk mengurangi efek Gaussian Noise secara efisien di platform mobile. Metode Midpoint merupakan teknik filter spasial non-linear yang bekerja dengan menghitung rata-rata dari nilai piksel minimum dan maksimum dalam sebuah jendela (kernel) pada citra. Teknik ini cukup ringan dan efektif dalam mereduksi noise acak tanpa merusak detail penting pada gambar.

Penelitian ini akan merancang dan mengimplementasikan aplikasi berbasis mobile yang mampu memuat citra, memprosesnya dengan metode Midpoint, dan menampilkan hasil perbandingan antara citra asli dan citra hasil filtering. Penilaian efektivitas metode dilakukan dengan pendekatan kuantitatif melalui pengukuran kualitas citra, seperti PSNR (Rasio Sinyal terhadap Derau Puncak) dan MSE (Rata-rata kuadrat galat), serta pendekatan kualitatif berdasarkan persepsi visual pengguna.

Metode *Mid Point* Merupakan sebuah *filter* yang berfungsi menggantikan nilai sebuah piksel dengan nilai rata-rata dari nilai terkecil dan nilai terbesar dari *gray-level* dalam *subimage* dibawah jendela ketetanggaan ukuran  $M \times N$ . Secara sederhana filter midpoint bertujuan untuk menghitung titik tengah antara nilai maksimum dan nilai minimum dalam daerah citra yang diliputi oleh filter. Yang kemudian menghasilkan aplikasi perangkat lunak yang mampu mengurangi efek *noise* yang mengganggu pada citra digital dengan menggunakan metode *MidPoint Filter*. Metode *Mid Point Filter* ini mengganti nilai sebuah *pixel* dengan nilai rata-rata dari nilai *pixel* terkecil dan nilai *pixel* terbesar dari *gray-level* dalam *subimage* di bawah jendela ketetanggaan ukuran  $M \times N$ . *Filter* ini merupakan kombinasi *order statistic* dan *average*. (Fajar, Astuti,2013).

Penelitian terkait tentang pengolahan citra menggunakan metode gaussian filtering untuk menghilangkan noise, didapatkan hasil bahwa jika gambar yang digunakan berwarna hitam putih maka noise dapat dikurangi lebih dari 80%, namun jika gambar yang digunakan memiliki warna hasil gaussian filter maka akan mempengaruhi warna pada gambar. Penelitian lain dalam peningkatan kualitas citra CT-scan dengan cara memburamkan citra dan menghilangkan noise menggunakan

metode gaussian filtering, kemudian menggunakan median filtering untuk mengurangi distribusi noise yang tidak normal. (Satya et al., 2022).

Penelitian ini menggunakan metode linier berurutan yaitu analisa, desain, penulisan kode program, dan pengujian. Dengan mengkombinasikan penggunaan android dan algoritma ini maka diharapkan pengguna lebih mudah dalam mengenali dan mendeteksi warna secara waktu yang nyata. Pengolahan citra berhubungan erat dengan warna, namun ada beberapa metode pada pengolahan citra yang mempunyai banyak kekurangan seperti metode rona, kejenuhan dan nilai kecerahan (Metode HSV) yang hanya dapat mengenali 6 warna saja dan beberapa model pengenalan citra lainnya yang penggunaannya tidak secara waktu yang nyata (Tamsir, 2020).

Penelitian ini mengaplikasikan algoritma midpoint untuk mendeteksi titik tengah layar kamera Android secara waktu yang nyata, kemudian mengekstrak nilai warna RGB dari koordinat tersebut. Aplikasi ini dikembangkan menggunakan Java dan JSON untuk menyimpan hasil deteksi warna secara otomatis (Amrullah, 2020).

Penelitian ini menerapkan Midpoint Filter untuk mengurangi efek *speckle noise* pada citra ultraviolet. Hasilnya menunjukkan peningkatan kualitas citra dengan metode sederhana yang mengganti nilai piksel tengah dengan nilai rata-rata maksimum dan minimum dalam jendela filter (Nainggolan, 2021).

Salah satu format citra adalah *bmp, jpeg*. Kita dapat menganggap suatu citra digital sebagai *matriks*, dimana indeks baris dan kolomnya menyatakan sebuah koordinat sebuah titik pada citra tersebut, dan nilai masing-masing elemennya menyatakan intensitas cahaya pada titik tersebut.

Berdasarkan latar belakang diatas maka penulis memiliki beberapa gagasan untuk menggugat judul tugas akhir ini :”**Penerapan Metode Mid Point Filter Mengurangi Efek Noise Gaussian Pada Citra Berbasis Android**”.

## **1.2 Rumusan Masalah**

Adapun rumusan masalah adalah sebagai berikut :

1. Bagaimana cara kerja metode midpoint dalam mengurangi efek gaussian noise pada citra digital.
2. Bagaimana implementasi metode midpoint dalam aplikasi berbasis android untuk pemrosesan citra.
3. Bagaimana menganalisa tingkat efektivitas metode mid point.

## **1.3 Batasan Masalah**

Adapun batasan masalah adalah sebagai berikut :

1. Penelitian hanya difokuskan pada pengurangan noise gaussian pada gambar digital (grayscale maupun RGB).
2. Perangkat lunak aplikasi pengolahan citra hanya menggunakan format citra jpeg dan bmp.
3. Perangkat lunak aplikasi pengolahan citra dibangun dengan menggunakan bahasa pemogramman java dengan software eclipse berbasis android.

#### **1.4 Tujuan Penelitian**

Adapun tujuan penelitian ini adalah :

1. Implementasi metode Midpoint dalam pengolahan citra digital guna mengurangi efek Gaussian Noise.
2. Mengembangkan aplikasi berbasis mobile yang dapat melakukan filtering citra menggunakan metode Midpoint secara efisien.
3. Menganalisis tingkat efektivitas metode Midpoint dalam mengurangi noise Gaussian.

#### **1.5 Manfaat Penelitian**

Hasil manfaat penelitian ini adalah :

1. Memberikan solusi nyata bagi pengguna perangkat mobile untuk membersihkan citra dari noise Gaussian secara cepat dan efisien.
2. Membantu mengembangkan aplikasi dalam merancang fitur pengolahan citra ringan untuk keperluan dokumentasi, medis, keamanan, dan lainnya.
3. Menunjukkan potensi metode Midpoint untuk digunakan dalam sistem real-time dengan keterbatasan perangkat.

## 1.6 Sistematika Penulisan

Adapun untuk sistematika penulisan tugas akhir ini terdiri dari 5 bab, yaitu:

### **BAB I : PENDAHULUAN**

Bab ini menguraikan tentang latar belakang masalah, rumusan masalah, batasan masalah, tujuan dan manfaat penelitian, metode penelitian dan sistematika penulisan.

### **BAB II : TINJAUAN PUSTAKA**

Bab ini berisi tentang kajian teori-teori yang digunakan sebagai landasan dalam penelitian dan pengertian aplikasi yang digunakan.

### **BAB III : METODE PENELITIAN**

Bab ini membahas langkah dari proses pembuatan analisa dan perancangan sistem dan perancangan antarmuka sistem android dan metode *Mid Point*.

### **BAB IV : HASIL DAN PEMBAHASAN**

Bab ini menunjukkan hasil disertai dengan analisa sehingga didapatkan bukti kuat dari penelitian skripsi yang dilakukan.

### **BAB V : KESIMPULAN DAN SARAN**

Bab ini menyimpulkan apa yang ada pada bab-bab terdahulu serta memberikan saran atas penulisan skripsi ini.

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Citra**

Citra merupakan salah satu bentuk informasi yang diperlukan manusia selain teks, suara dan video. Informasi yang terkandung dalam sebuah citra dapat diinterpretasikan berbeda-beda oleh manusia satu dengan yang lain. Citra analog dihasilkan dari alat akuisisi citra analog, contohnya adalah mata manusia dan kamera analog. Gambaran yang tertangkap oleh mata manusia dan foto atau film yang tertangkap oleh kamera analog merupakan contoh dari citra analog. Citra digital merupakan representasi dari fungsi intensitas cahaya dalam bentuk diskrit pada bidang dua dimensi.

Citra tersusun oleh sekumpulan piksel (picture element) yang memiliki koordinat  $(x,y)$  dan amplitudo  $f(x,y)$ . Koordinat  $(x,y)$  menunjukkan letak/posisi piksel dalam suatu citra, sedangkan amplitudo  $f(x,y)$  menunjukkan nilai intensitas warna citra. Pengolahan citra (image Processing) adalah proses mengolah piksel-piksel di dalam citra digital yang digunakan untuk tujuan tertentu. Awalnya pengolahan citra dilakukan untuk memperbaiki kualitas citra, dengan berkembangnya dunia komputasi yang ditandai dengan semakin meningkatnya kapasitas dan kemampuan komputer memungkinkan manusia dapat Mengambil informasi dari suatu citra.

Citra digital adalah gambar dua dimensi yang dihasilkan dari gambar analog dua dimensi yang kontinu menjadi gambar diskrit melalui proses sampling. Agar dapat direpresentasikan secara numerik dengan nilai-nilai diskrit. Representasi citra

dari fungsi malar (kontinu) menjadi nilai-nilai diskrit disebut digitalisasi (Ratna, 2020).

## 2.2 Pengelolaan Citra

Pengolahan Citra Digital (Digital Image Processing) merupakan disiplin ilmu yang mempelajari teknik dalam mengolah citra, citra yang dimaksud adalah merupakan gambar diam (foto) atau gambar yang bergerak (seperti video yang direkam) . Sedangkan arti digital adalah pengolahan citra/gambar dilakukan menggunakan komputer secara digital. RGB adalah singkatan dari Red-Green-Blue, merupakan tiga warna dasar yang secara umum dijadikan acuan warna lainnya. Dari basis RGB, kita dapat mengkonversi warna menjadi kode-kode angka yang membuat warna tersebut akan tampil universal. Komputer sudah mengemas informasi warna menjadi model warna yang sama sehingga membuat pengolahan warna RGB dapat dilakukan dengan mudah (Ratna, 2020).

Pengolahan citra digital menggunakan berbagai algoritma dalam teknik pengolahan gambar atau citra. Salah satu proses yang dapat dilakukan digital image processing merupakan citra yang berawal dari foto yang gelap dan diproses menjadi lebih terang. Digital image processing sudah diterapkan di berbagai bidang dan aplikasi. Digital image processing juga dapat menganalisis hal-hal yang tidak kasat mata tetapi bisa dilihat dengan kamera. Contohnya seperti gelombang elektromagnetik yang menampilkan spektrum warna (Tri Laksono Aditiya et al., 2022).

RGB merupakan jenis citra yang terdiri dari komponen warna merah (R), hijau (G) dan biru (B). Apabila warna-warna tersebut dikombinasikan, maka akan

terbentuk warna citra asli. Oleh karena itu RGB sering disebut juga sebagai additive color. Penambahan warna merupakan percampuran dua warna primer akan menghasilkan warna sekunder RGB. Pengaturan warna RGB mempunyai skala 0-255. Warna dari tiap piksel ditentukan dengan kombinasi intensitas warna merah, biru dan hijau Format file grafik menyimpan citra RGB sebagai 24-bit dengan komponen merah, hijau dan biru dengan masing- masing komponen 8 bit (Tri Laksono Aditiya et al., 2022).

### 2.3 Piksel

Piksel adalah nilai tiap-tiap entri matriks pada *bitmap*. Rentang nilai-nilai piksel ini dipengaruhi oleh banyaknya warna yang dapat ditampilkan. Jika suatu bitmap dapat menampilkan 256 warna maka nilai-nilai *pixel*nya dibatasi dari 0 hingga 255. Suatu bitmap dianggap mempunyai ketepatan yang tinggi jika dapat menampilkan lebih banyak warna. Prinsip ini dapat dilihat dari contoh pada gambar yang memberikan contoh dua buah bitmap dapat memiliki perbedaan dalam menangani transisi warna putih ke warna hitam.



Gambar 2. 1 Pixel Citra

(Fadli & Dewi, 2020).

Perbedaan ketepatan warna bitmap pada Gambar 2.2 menjelaskan bahwa bitmap sebelah atas memberikan nilai untuk warna lebih sedikit daripada bitmap di bawahnya. Untuk bitmap dengan pola yang lebih kompleks dan dimensi yang lebih besar, perbedaan keakuratan dalam memberikan nilai warna akan terlihat lebih

jasas. Sebuah citra adalah kumpulan piksel-piksel yang disusun dalam larik dua dimensi. Indeks baris dan kolom  $(x,y)$  dari sebuah *pixel* dinyatakan dalam bilangan bulat. Piksel  $(0,0)$  terletak pada sudut kiri atas pada citra, indeks  $x$  bergerak ke kanan dan indeks  $y$  bergerak ke bawah.

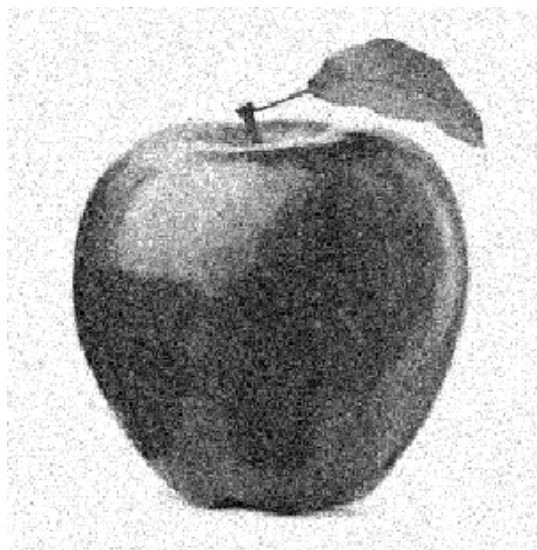
Konvensi ini dipakai merujuk pada cara penulisan larik yang digunakan dalam pemrograman komputer. Letak titik origin pada koordinat grafik citra dan koordinat pada grafik matematika terdapat perbedaan. Hal yang berlawanan untuk arah vertikal berlaku pada kenyataan dan juga pada sistem grafik dalam matematika yang sudah lebih dulu dikenal (Fadli & Dewi, 2020).

## 2.4 Noise Gaussian

Noise merupakan gangguan yang disebabkan oleh menyimpangnya data digital yang diterima oleh alat penerima data gambar yang mana dapat mengganggu kualitas citra. Derau dapat disebabkan oleh gangguan fisis (optik) pada alat penangkap citra misalnya kotoran debu yang menempel pada lensa foto maupun akibat proses pengolahan yang tidak sesuai. Ada tiga jenis noise yaitu gaussian noise, speckle noise, dan salt and pepper noise. Gaussian noise merupakan model noise yang mengikuti distribusi normal standard dengan rata-rata nol dan standard deviasi 1. Efek dari gaussian noise ini pada gambar adalah munculnya titik-titik berwarna yang jumlahnya sama dengan persentase noise. Noise speckle merupakan model noise yang memberikan warna hitam pada titik yang terkena noise.

Gaussian Noise adalah jenis gangguan (noise) yang sering muncul dalam citra digital dan sinyal digital, serta merupakan model gangguan yang paling umum digunakan dalam bidang pengolahan citra dan komunikasi digital. Noise ini

dinamakan berdasarkan distribusi Gaussian (normal) karena nilai-nilai gangguan yang ditambahkan ke sinyal atau citra mengikuti distribusi probabilitas normal. Dalam konteks citra digital, Gaussian Noise muncul sebagai fluktuasi nilai intensitas piksel secara acak, yang dapat menyebabkan citra tampak berbintik atau kabur. Noise ini biasanya disebabkan oleh faktor-faktor seperti sensor kamera, proses transmisi data, atau gangguan lingkungan.



Gambar 2. 2 Noise Gaussain

Noise adalah suatu sinyal gangguan yang bersifat akustik (suara), listrik, maupun elektronik yang hadir dalam suatu sistem dalam bentuk gangguan yang bukan merupakan sinyal yang diinginkan. Rekaman suara yang terdistorsi noise menyebabkan terganggunya proses pengenalan suara terutama pada rekaman pembicaraan. Penulisan ini membahas mengenai implementasi metode penyaring noise untuk meredam bagian yang hanya dianggap noise, filter lolos-rendah untuk frekuensi batas untuk noise frekuensi tinggi dan penghilang bagian yang sunyi untuk menghilangkan bagian diam pada file rekaman suara pembicaraan. Setiap file

suara memiliki nilai rentang dinamis dan faktor puncak yang berbeda yang dijadikan sebagai acuan dalam pengisian parameter sehingga bersifat dinamis. Hasil yang didapat dari pembuatan aplikasi ini terdapat dua komparasi antara sinyal asli dengan sinyal asli + noise dan sinyal asli dengan output aplikasi, kedua komparasi tersebut menghasilkan peningkatan keberhasilan dalam pengenalan suara dari penggunaan metode penyaring noise, filter lolos-rendah dan penghilang bagian yang sunyi untuk menghilangkan noise adalah 3,5 kali lipat dibandingkan dengan pengenalan suara tanpa dilakukan penghilangan noise pada file suara sebelumnya (Fauzi, 2022).

## 2.5 Metode Mid Point Filter

Metode Mid Point Filter ini mengganti nilai sebuah piksel dengan nilai rata-rata dari nilai piksel terkecil dan nilai piksel terbesar dari gray-level dalam subcitra di bawah jendela ketetanggaan ukuran  $M \times N$ . Filter ini merupakan kombinasi urutan statistik dan rata-rata. Mid-Point Filtering berfungsi mengganti nilai sel bitmap dengan nilai tengah diantara nilai terkecil dan terbesar dari area lokal.

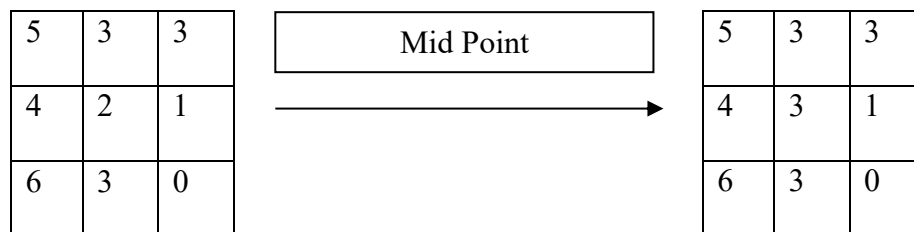
Midpoint Filter menghitung titik tengah antara nilai maksimum dan minimum dalam daerah yang diliputi filter tersebut. (Fajar, Astuti, 2013).

Secara sederhana filter midpoint bertujuan untuk menghitung titik tengah antara nilai maksimum dan nilai minimum dalam daerah citra yang diliputi oleh filter.

Untuk filter ini dapat menggunakan rumus :

$$f'(x, y) = \frac{1}{2} [\max_{(s, t) \in S_{xy}} \{g(s, t)\} + \min_{(s, t) \in S_{xy}} \{g(s, t)\}] \dots \dots \dots (1)$$

Dari proses pemilihan dan pengurutan kernel pertama pada Gambar 6, dapat diperoleh hasil midpoint-nya adalah 3. Proses pengurutan dan pemilihan nilai tengah antara nilai maksimum dan nilai minimum dapat diperlihatkan pada gambar berikut.



$$h = \frac{1}{2} [\max + \min]$$

$$= \frac{1}{2} [6-0] = 3$$

## 2.6 Android

Perkembangan teknologi perangkat seluler memungkinkan kebutuhan akan informasi dapat terpenuhi. Semakin kayanya fitur dan semakin terjangkau harga jual produk perangkat seluler, memungkinkan perangkat seluler menjadi sebuah solusi yang tepat bagi kebutuhan komunikasi masyarakat. Salah satu contoh alat telekomunikasi yang mengalami perkembangan yang signifikan adalah alat telekomunikasi bergerak yang menggunakan Android.

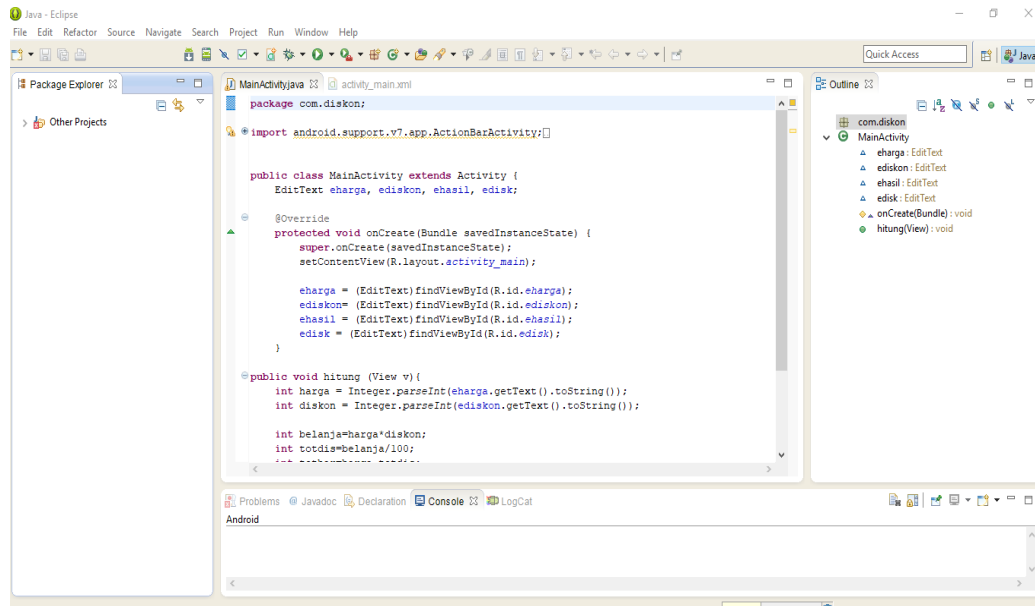
Android merupakan sebuah sistem operasi pada handphone yang bersifat terbuka dan berbasis pada sistem operasi Linux “. Android bisa digunakan oleh setiap orang yang ingin menggunakannya pada perangkat mereka. Android bersifat terbuka (opensource) yang berdampak pada meningkatnya jumlah pengguna maupun pengembang aplikasi secara *continue* dan signifikan (Lapi & Prayitno, 2023).

Android merupakan sistem operasi yang sekarang sedang terkenal di pasaran smartphone saat ini. Android adalah sebuah sistem operasi perangkat mobile berbasis linux yang mencakup sistem operasi, perangkat lunak perantara dan aplikasi. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka.

Awalnya, Google Inc. membeli Android Inc. yang merupakan pendatang baru yang membuat peranti lunak untuk ponsel atau smartphone. Kemudian untuk mengembangkan Android, dibentuklah Aliansi Perangkat Genggam Terbuka, konsorsium dari 34 perusahaan peranti keras, peranti lunak dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia. Android bersama Aliansi Perangkat Genggam Terbuka menyatakan mendukung pengembangan terbuka pada perangkat mobile. Di lain pihak, Google merilis kode-kode Android di bawah lisensi Apache, sebuah lisensi perangkat lunak dan open platform perangkat seluler (Lapi & Prayitno, 2023).

### **2.6.1 Android Studio**

Android studio adalah IDE (Integrated Development Environment) resmi untuk pengembangan aplikasi Android dan bersifat terbuka atau gratis. Peluncuran Android Studio ini diumumkan oleh Google pada 16 mei 2013 pada event Google I/O Conference untuk tahun 2013. Sejak saat itu, Android Studio menggantikan Eclipse sebagai IDE resmi untuk mengembangkan aplikasi Android.



Gambar 2. 3 Ruang Kerja Android

Android studio sendiri dikembangkan berdasarkan IntelliJ IDEA yang mirip dengan Eclipse disertai dengan ADT plugin (Android Development Tools). Android studio memiliki fitur :

- a. Projek berbasis pada Gradle Build.
- b. Refactory dan pembenahan bug yang cepat.
- c. Tools baru yang bernama “Lint” dikalim dapat memonitor kecepatan, kegunaan, serta kompetibilitas aplikasi dengan cepat.
- d. Mendukung Proguard And App-signing untuk keamanan.
- e. Memiliki GUI aplikasi android lebih mudah
- f. Didukung oleh Google Cloud Platfrom untuk setiap aplikasi yang dikembangkan.

Kit pengembang java (JDK) adalah sekumpulan perangkat lunak yang dapat kamu gunakan untuk mengembangkan perangkat lunak yang berbasis java, sedangkan lingkungan runtime java (JRE) adalah sebuah implementasi dari mesin


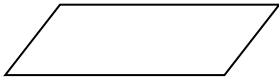

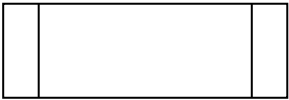
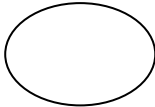
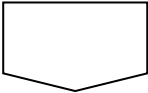
virtual java yang benar-benar digunakan untuk menjalankan program java. Biasanya, setiap jdk berisi satu atau lebih jre dan berbagai alat pengembangan lain seperti sumber kompilasi java, penggabungan, alat debugging, perpustakaan pengembangan dan lain sebagainya.

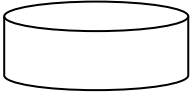
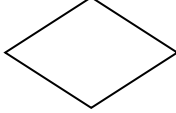
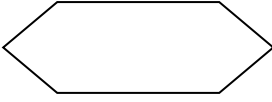

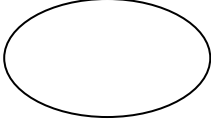
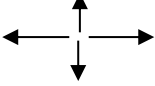
## **2.7 Pengertian Flowchart**

Flowchart adalah representasi secara simbolik dari suatu algoritma atau prosedur untuk menyelesaikan suatu masalah, dengan menggunakan flowchart akan memudahkan pengguna melakukan pengecekan bagian-bagian yang terlupakan dalam analisis masalah, disamping itu flowchart juga berguna sebagai fasilitas untuk berkomunikasi antara pemrogram yang bekerja dalam tim suatu proyek (Kus Indrani Listyoningrum et al., 2023).

Flowchart membantu memahami urutan-urutan logika yang rumit dan panjang. Flowchart membantu mengkomunikasikan jalannya program ke orang lain (bukan pemrogram) akan lebih mudah.

Tabel 2. 1 Simbol Flowchart

Symbol	Kegunaan
 <p><i>Terminal Point Symbol</i></p>	Symbol titik terminal digunakan untuk awal dan akhir dari suatu proses
 <p><i>Input/Output Symbol</i></p>	Symbol input/output yang digunakan untuk mewakili data input/output
 <p><i>Proses Symbol</i></p>	Symbol proses yang digunakan untuk mewakili data input/output
 <p><i>Predefine Process Symbol</i></p>	Symbol proses terdefinisi yang digunakan untuk menunjukkan suatu operasi yang rinciannya ditunjukkan ditempat lain
 <p><i>Connector Symbol</i></p>	Symbol penghubung digunakan untuk menunjukkan sambungan dari bagian alir yang terputus dihalaman yang masih sama
 <p><i>Offline Connector Symbol</i></p>	Offline connector merupakan symbol untuk masuk dan keluarnya suatu prosedur pada lembar kertas yang lain

 <i>Magnetic Disk Symbol</i>	<p>Symbol untuk database yang digunakan dalam program</p>
 <i>Decision Symbol</i>	<p>Symbol keputusan yang digunakan untuk mempersiapkan penyimpanan yang digunakan sebagai tempat pengolahan di dalam storage</p>
 <i>Preparation Symbol</i>	<p>Symbol persiapan digunakan untuk mempersiapkan penyimpanan yang digunakan sebagai tempat pengolahan di dalam storage</p>
 <i>Document Symbol</i>	<p>Document merupakan symbol untuk data yang berbentuk kertas maupun informasi</p>
 <i>Display Symbol</i>	<p>Symbol untuk output yang menunjukkan ke suatu device, seperti printer, plotters</p>
 <i>Line Connector</i>	<p>Arus/flow dari prosedur yang dapat dilakukan dari atas ke bawah, dari bawah ke atas.</p>

(Sumber : Kus Indrani Listyoningrum et al., 2023)

## 2.8 Bahasa Pemodelan Terpadu (UML)

Bahasa pemodelan terpadu merupakan alat perancangan sistem yang berorientasi pada objek. Secara filosofi kemunculan UML diilhami oleh konsep yang telah ada yaitu konsep permodelan berorientasi objek, karena konsep ini

menganalogikan sistem seperti kehidupan nyata yang didominasi oleh obyek dan digambarkan atau dinotasikan dalam simbol-simbol yang cukup spesifik maka OO memiliki proses standard dan bersifat independen (Sukma & Abhyanda, 2020).

UML diagram memiliki tujuan utama untuk membantu tim pengembangan proyek berkomunikasi, mengeksplorasi potensi desain, dan memvalidasi desain arsitektur perangkat lunak atau pembuat program. Komponen atau notasi UML diturunkan dari 3 (tiga) notasi yang telah ada sebelumnya yaitu grady booch desain berorientasi objek (ood), jim rumbaugh teknik pemodelan objek (omt) dan ivar jacobson rekayasa perangkat lunak berorientasi objek (oose). UML mempunyai tiga kategori utama yaitu struktur diagram, perilaku diagram dan interaksi diagram. Dimana masing-masing kategori tersebut memiliki diagram yang menjelaskan arsitektur sistem dan saling terintegrasi.

Bahasa pemodelan terpadu (UML) adalah suatu alat untuk memvisualisasikan dan mendokumentasikan hasil analisa dan desain yang berisi sintak dalam memodelkan sistem secara visual. Juga merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem software yang terkait dengan objek. Sejarah UML sendiri terbagi dalam dua fase, sebelum dan sesudah munculnya UML. Dalam fase sebelum, UML sebenarnya sudah mulai diperkenalkan sejak tahun 1990-an namun notasi yang dikembangkan oleh para ahli analisis dan desain berbeda-beda, sehingga dapat dikatakan belum memiliki standarisasi (Noviana, 2022).

Fase kedua, dilandasi dengan pemikiran untuk mempersatukan metode tersebut dan dimotori oleh kelompok manajemen objek (omg) maka pengembangan UML dimulai pada akhir tahun 1994 ketika grady booch dengan metode desain

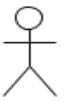
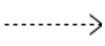
berorientasi objek, jim rumbaugh dengan metode teknik pemodelan objek (omt) mereka ini bekerja pada rasional software corporation dan ivar jacobson dengan metode rekayasa perangkat lunak berorientasi objek (oose) yang bekerja pada perusahaan objectory rasional. (Anisah R, 2017).


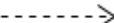




Sebagai pencetus metode-metode tersebut mereka bertiga berinisiatif untuk menciptakan bahasa pemodelan terpadu sehingga pada tahun 1996 mereka berhasil merilis UML versi 0.9 dan 0.91 melalui permintaan proposal yang dikeluarkan oleh OMG.


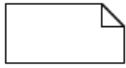
#### 1. Kasus penggunaan

Diagram yang menggambarkan aktor, use case dan relasinya sebagai suatu urutan tindakan yang memberikan nilai terukur untuk aktor. Sebuah use case digambarkan sebagai elips horizontal dalam suatu diagram UML use case. memiliki dua istilah yaitu system use case; interaksi dengan sistem.

Tabel 2. 2 Simbol Use Case Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempengaruhi elemen yang bergantung padanya

			elemen yang tidak mandiri ( <i>independent</i> ).
3		<i>Generalization</i>	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ).
4		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor

9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen.
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi






(Sumber : Noviana 2022)

## 2. Diagram Aktivitas

Menggambarkan aktifitas-aktifitas, objek, *state*, transisi *state* dan *event*.

Dengan kata lain kegiatan diagram alur kerja menggambarkan perilaku sistem untuk aktivitas.

Tabel 2. 3 Simbol Activity Diagram

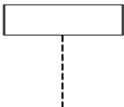

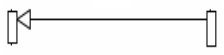
NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actifity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4		<i>Actifity</i> <i>Final Node</i>	Bagaimana objek dibentuk dan dihancurkan
5		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

(Sumber : Noviana 2022)

### 3. Diagram Urutan

Diagram urutan menjelaskan interaksi objek yang disusun berdasarkan urutan waktu. Secara mudahnya *sequence* diagram adalah gambaran tahap demi tahap, termasuk kronologi (urutan) perubahan secara logis yang seharusnya dilakukan untuk menghasilkan sesuatu sesuai dengan *use case* diagram.

Tabel 2. 4 Simbol Sequence Diagram



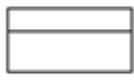

NO	GAMBAR	NAMA	KETERANGAN
1		<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
2		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
3		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi


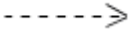

(Sumber : Noviana 2022)

#### 4. Class Diagram

*Class* diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. kelas memiliki 3 bagian utama yaitu *attribute*, *operation*, dan *name*. kelas-kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem.

Tabel 2. 5 Simbol Class Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Generalization</i>	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ).
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor.

5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

(Sumber : Noviana 2022)

## 2.9 Penelitian Terkait

### 1. Implementasi Metode Mid Point Filter Untuk Mengurangi Efek Noise Pada Citra Bersih.

Reduksi Noise adalah sebuah teknik yang digunakan untuk memperbaiki gambar yang memiliki noise, teknik yang digunakan adalah dengan mengganti pixel noise dengan pixel gambar yang mendekati pixel noise tersebut. Metode MidPoint filter ini mengganti nilai sebuah piksel dengan nilai rata-rata dari piksel terkecil dan nilai piksel terbesar dari gray-level dalam subimage dibawah jendela ketetanggan ukuran MxN.

Berdasarkan hasil dari reduksi noise pada citra menggunakan mid point filter, maka dapat disimpulkan bahwa mid point filter tidak cukup akurat dalam mereduksi noise karena hanya mengganti nilai titik tengahnya saja. Metode mean filter dan

mid point filter dapat diterapkan dalam reduksi noise dengan operasi menghitung nilai piksel awal sampai pada piksel akhir gambar.

## 2. Pengurangan Derau (Noise) pada Citra Paper Dokumen menggunakan Metode Gaussian Filter dan Median Filter.

Noise dapat dikurangi dengan metode filtering citra, Filtering citra merupakan suatu proses untuk menggantikan nilai dari suatu pixel pada citra. Tujuan dari filtering pada pemrosesan citra adalah untuk menyeleksi suatu nilai piksel sehingga memiliki variasi nilai yang mampu menggambarkan kenampakan dengan lebih jelas dari citra asli. Metode yang dapat digunakan dalam mengurangi noise pada citra adalah dengan menggunakan metode gaussian filter dan metode median filter.

Penerapan metode gaussian filter dan median filter pada proses filtering citra dapat mengubah nilai masing-masing pixel RGB pada citra. Dimana proses metode gaussian filter menggunakan bobot untuk menaikkan nilai dari pixel yang di proses dan metode median filter memanfaatkan pixel tengah dari pixel yang diolah berdasarkan kernel yang digunakan, sehingga dengan kedua metode filtering tersebut dapat meningkatkan kualitas dari citra yang di proses dengan pengurangan derau (noise) pada citra.

## 3. Metode Pemfilteran Gaussian Untuk Menghilangkan Noise Pada Gambar.

Noise merupakan gangguan yang disebabkan oleh penyimpanan data digital yang diterima oleh penerima data gambar yang dapat mengganggu kualitas gambar. Noise dapat disebabkan oleh gangguan fisik (optik) pada perangkat penangkap gambar, seperti debu pada lensa kamera atau akibat pemrosesan yang tidak tepat.

Untuk menghilangkan noise ini, Anda dapat menggunakan berbagai metode, salah satunya adalah Gaussian Filtering.

Hasil citra yang awalnya terdapat noise dan setelah dilakukan proses perbaikan kualitas citra menggunakan Matlab menunjukkan hasil yang signifikan. Dimana, noise pada citra yang telah mengalami proses perbaikan kualitas citra menggunakan Matlab tampak menghilang.

## 2.10 Perbandingan Penelitian Terdahulu

Beberapa penelitian sebelumnya telah dilakukan dalam bidang pengolahan citra digital, khususnya dalam mengatasi gangguan *noise* pada citra menggunakan berbagai metode penyaringan (*filtering*). Namun, terdapat perbedaan dalam hal pendekatan, algoritma yang digunakan, serta platform penerapannya. Berikut perbandingan beberapa penelitian terdahulu dengan penelitian ini:

No	Peneliti & Tahun	Judul Penelitian	Hasil / Temuan Utama	Perbandingan dengan Penelitian Ini
1	Andika, (2019)	Penerapan Median Filter untuk Mengurangi Noise pada Citra Digital	Median filter mampu mengurangi <i>salt and pepper noise</i> secara efektif, namun hasilnya cenderung menghaluskan detail citra.	Penelitian ini menggunakan Mid Point Filter untuk mempertahankan detail tepi citra dan menekan <i>Gaussian noise</i> , bukan <i>salt and pepper noise</i> .
2	Sari dan Putra, (2020)	Analisis Pengaruh Gaussian Filter terhadap Reduksi Noise Citra Medis	Gaussian filter efektif mengurangi <i>random noise</i> , tetapi citra menjadi agak buram.	Penelitian ini memperbaiki kelemahan tersebut dengan menggunakan Mid Point Filter yang menjaga kontras citra lebih baik.