

BAB I

PENDAHULUAN

A. Latar Belakang Masalah

Pengolahan citra digital menjadi semakin penting seiring dengan perkembangan teknologi kamera dan kebutuhan untuk menghasilkan gambar berkualitas tinggi di berbagai bidang. Seperti medis, keamanan, dan hiburan. Kualitas gambar sangat dipengaruhi oleh kondisi pencahayaan, resolusi, dan gangguan (noise) yang mungkin terjadi selama pengambilan gambar. Salah satu metode yang digunakan untuk meningkatkan kualitas gambar adalah operasi titik, yaitu teknik yang menerapkan fungsi matematika pada setiap piksel gambar tanpa memperhitungkan piksel-piksel di sekitarnya. Meskipun sederhana, operasi titik sangat efektif dalam memperbaiki kontras, kecerahan, dan menghilangkan noise pada gambar. Penelitian ini bertujuan untuk merancang dan mengimplementasikan algoritma operasi titik, guna meningkatkan kualitas gambar melalui aplikasi yang dapat berfungsi secara real-time.

Teknologi informasi pada zaman sekarang ini tidak dapat dipisahkan dari bidang multimedia. Pernahkah kamu berpikir tentang kapan bermulanya proses digitalisasi atau perkembangan perangkat media di sekitar kita? Atau, kamu penasaran tentang apa itu multimedia dan jenis-jenisnya? Nah, bisa jadi kata “multimedia” ini seringkali kita gunakan dalam bahasan mengenai teknologi, namun beberapa dari kita belum memahami tentang apa yang sebenarnya dimaksud dengan multimedia ini

Teknologi dibidang multimedia sangat erat kaitannya dengan pengetahuan mengenai citra gambar. Citra gambar merupakan salah satu komponen multimedia yang memegang peran penting untuk membentuk informasi visual. Kualitas citra seringkali mengalami penurunan kualitas seperti adanya noise, blur, dan masalah lainnya yang dapat mempengaruhi kualitas citra gambar menjadi kurang baik. Hal ini mengakibatkan citra yang kurang baik lebih sulit dikenali karena informasi yang ingin disampaikan menjadi berkurang. Citra yang kurang baik dapat diperbaiki menjadi kualitas citra yang lebih baik dengan menggunakan salah satu operasi pengolahan citra.

Penulis akan menggunakan operasi titik. Operasi titik dilakukan hanya pada pixel tunggal didalam operasi citra. Dimana pada citra gambar akan dihilangkannya citra yang tidak diperlukan seperti noise atau derau, dan juga akan meningkatkan kualitas citra gambar agar mudah diamati oleh manusia.

B. Fokus Penelitian

Penelitian ini difokuskan pada pengembangan dan penerapan algoritma operasi titik untuk meningkatkan kualitas gambar digital, yang akan diimplementasikan dalam sebuah aplikasi yang dapat memproses gambar secara real-time. Penelitian ini juga mengevaluasi efektivitas berbagai fungsi operasi titik, seperti transformasi linear, logaritmik, dan eksponensial.

C. Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan diatas maka dirumuskan masalah sebagai berikut :

1. Bagaimana cara meningkatkan kualitas gambar digital menggunakan operasi titik?
2. Fungsi operasi titik apa saja yang paling efektif dalam meningkatkan kualitas gambar?
3. Bagaimana mengimplementasikan algoritma operasi titik ke dalam aplikasi yang bekerja secara real-time?

D. Batasan Masalah

Untuk dapat menghasilkan penelitian yang lengkap, akurat dan tepat waktu dalam Sistem yang dibangun, Maka penulis membatasi beberapa permasalahan berikut:

1. Aplikasi yang dibuat untuk meningkatkan kualitas gambar yang mengalami noise atau derau.
2. Aplikasi ini hanya dapat berjalan maksimal pada gambar dengan citra tunggal.
3. Aplikasi ini dapat memperbaiki citra gambar dengan sempurna.
4. Aplikasi ini dibuat berbaasis *desktop*.
5. Aplikasi ini hanya berjalan pada gambar dengan format BMP (bitmap)

E. Tujuan Penelitian

Berikut ini adalah rincian tujuan berdasarkan rumusan masalah yang telah di definisikan dalam penelitian ini adalah :

1. Membuat sistem yang dapat meningkatkan kualitas citra gambar.
2. Menghilangkan citra yang tidak dibutuhkan pada citra gambar.

F. Manfaat Penelitian

Adapun manfaat yang diharapkan dalam pembuatan sistem ini yaitu:

1. Manfaat untuk penulis

Adapun manfaat bagi penulis adalah dapat mengembangkan ilmu dalam bidang multimedia dan menciptakan aplikasi yang bermanfaat.

2. Manfaat untuk masyarakat umum

Adapun manfaat bagi masyarakat adalah dapat meningkatkan kualitas citra pada foto yang mengalami noise atau kurang jelas.

3. Manfaat untuk universitas

Adapun manfaat bagi universitas adalah dapat menjadi referensi dan bahan untuk meningkatkan kemampuan di bidang multimedia dan lebih tertarik dalam mengembangkan aplikasi di bidang multimedia.

G. Sistematika Penulisan

Dalam menyusun sistematika penulisan, penulis menguraikan ke dalam lima Bab yaitu:

BAB I : PENDAHULUAN

Menguraikan tentang latar belakang penulisan, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, dan sistematika penulisan.

BAB II: TINJAUAN PUSTAKA

Berisi uraian tentang tinjauan penelitian terdahulu dan teori yang digunakan sebagai referensi perancangan aplikasi.

BAB III: METODE PENELITIAN

Membahas tentang waktu dan tempat penelitian, metode penelitian, jenis penelitian, alat dan bahan yang digunakan.

BAB IV: HASIL PENELITIAN DAN PEMBAHASAN

Menjelaskan bagaimana tentang proses cara kerja user melakukan proses mendapatkan foto yang diinginkan.

BAB V: PENUTUP

Berisi uraian tentang kesimpulan dan saran.

BAB II

TINJAUAN PUSTAKA

A. Penelitian Sebelumnya

Penelitian ini memberikan kontribusi baru dalam implementasi real-time dari operasi titik dalam pengolahan citra digital, yang memungkinkan pengguna untuk memperbaiki kualitas gambar secara langsung dan efisien.

Beberapa sumber pustaka yang relevan untuk mendukung penelitian ini meliputi:

1. **Gonzalez, R. C., & Woods, R. E. (2018).** “Digital Image Processing (4th ed.). Pearson”. Buku ini memberikan landasan teori dan teknik pengolahan citra, termasuk operasi titik yang dibahas dalam penelitian ini.
2. **Pratt, W. K. (2007).** “Digital Image Processing: PIKS Inside (4th ed.). Wiley- Interscience”. Sumber ini menyajikan berbagai algoritma pengolahan citra, termasuk transformasi linear, logaritmik, dan eksponensial.
3. **Jain, A. K. (1989).** “Fundamentals of Digital Image Processing. Prentice Hall”. Buku ini merupakan referensi klasik yang mencakup konsep dasar pengolahan citra dan penerapan operasi titik.

Artikel Penelitian yang Sama di Indonesia

Berikut adalah beberapa artikel penelitian di Indonesia yang relevan dengan topik ***“Operasi Titik pada Pengolahan Citra Digital”***:

1. **“Peningkatan Kualitas Citra dengan Metode Transformasi Logaritmik pada Pengolahan Citra Digital”** oleh Hendra Wijaya dan Ahmad Suryadi, Jurnal Teknologi Informasi, 2021. Artikel ini membahas penggunaan transformasi logaritmik untuk meningkatkan kualitas citra dalam pengolahan citra digital.
2. **“Implementasi Operasi Titik untuk Peningkatan Kualitas Citra Medis Menggunakan Transformasi Eksponensial”** oleh Fitriani dan Dewi Rahayu, jurnal Informatika dan Komputer, 2020. Penelitian ini fokus pada penggunaan transformasi eksponensial dalam citra medis untuk meningkatkan kontras.
3. **“Pengaruh Transformasi Linear terhadap Peningkatan Kualitas Gambar pada Citra Digital”** oleh Arif Susanto dan Yulia Sari, jurnal Ilmiah Teknologi Informasi, 2019. Artikel ini mengevaluasi efektivitas transformasi linear dalam meningkatkan kualitas gambar digital.

B. Tinjauan Teori

1. Operasi Titik

Citra digital direpresentasikan dengan matriks. Operasi pada citra digital pada dasarnya adalah manipulasi elemen-elemen matriks. Elemen matriks yang dimanipulasi dapat berupa elemen tunggal sebuah pixel atau sekumpulan elemen yang berdekatan, atau keseluruhan elemen matriks.

Operasi yang dilakukan untuk mentransformasikan suatu citra menjadi citra lain dapat dikategorikan berdasarkan tujuan transformasi maupun cakupan operasi yang dilakukan terhadap citra.

Operasi pada aras titik hanya dilakukan pada pixel tunggal di dalam citra. Operasi titik dikenal juga dengan nama operasi pointwise. Operasi ini terdiri dari pengaksesan pixel pada lokasi yang diberikan, memodifikasinya dengan operasi operasi linjar (linear) atau nirlinjar (nonlinear), dan menempatkan nilai pixel baru pada lokasi yang bersesuaian di dalam citra yang baru. Operasi ini diulangi untuk keseluruhan pixel di dalam citra.

Operasi-operasi yang dilakukan pada pengolahan citra dapat dikelompokkan berdasarkan keadaan diatas, maka :

$$\text{minInt} = \text{minimum} \quad (5, 7, 7, 5, 4, 6, 7, 8) = 4$$

$$\text{maksInt} = \text{maksimum} \quad (5, 7, 7, 5, 4, 6, 7, 8) = 8$$

mengingat $f(y, x)$ bernilai 9 dan lebih besar daripada 8 (maksInt)

maka $g(y, x)$ bernilai 8.

Seandainya $f(y, x)$ pada keadaan di atas bernilai 2 (bukan 9), $g(y, x)$ akan bernilai 4 dan memodifikasinya dengan operasi-operasi linjar (linear) atau nirlinjar (nonlinear), dan menempatkan nilai pixel baru pada lokasi yang bersesuaian di dalam citra yang baru. Operasi ini diulangi untuk keseluruhan pixel di dalam citra. Secara matematis, operasi pada aras titik dinyatakan sebagai :

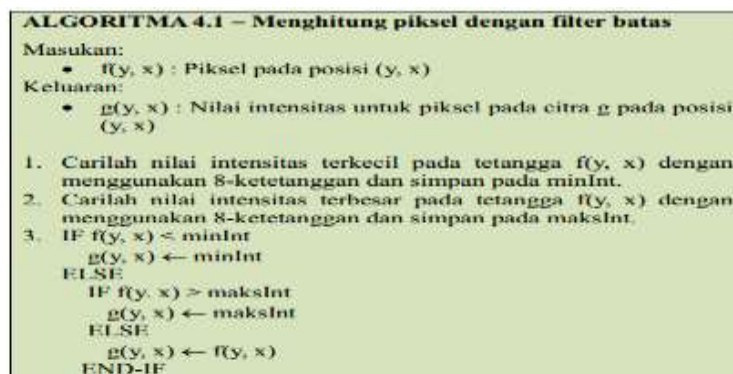
$$fB(x, y) = O_{titik}\{fA(x, y)\}$$

Dalam hal ini fA dan fB masing-masing adalah citra masukan dan citra keluaran, O_{titik} dapat berupa operasi linjar (linear) atau nirlinjar (nonlinear). Yang dimaksud dengan operasi linjar adalah operasi yang dapat dinyatakan secara matematis sebagai persamaan linjar, kebalikannya adalah persamaan nirlinjar. Operasi pada aras titik dapat dibagi menjadi tiga macam: berdasarkan

intensitas, berdasarkan geometri, atau gabungan keduanya.

2. Filter Batas

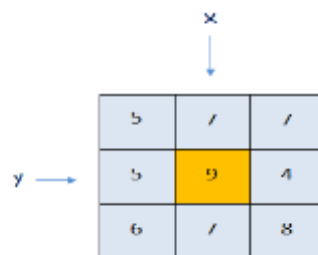
Filter batas adalah filter yang dikemukakan dalam Davies (1990). Idenya



Gambar 2. 1 Algoritma menghitung piksel dengan filter batas

adalah mencegah piksel yang intensitasnya di luar intensitas piksel-piksel tetangga. Algoritma yang digunakan untuk keperluan ini dapat dilihat berikut ini :

Contohnya, terdapat piksel seperti pada gambar berikut :



Gambar 2. 2 Contoh piksel dan tetangga

Berdasarkan keadaan diatas, maka :

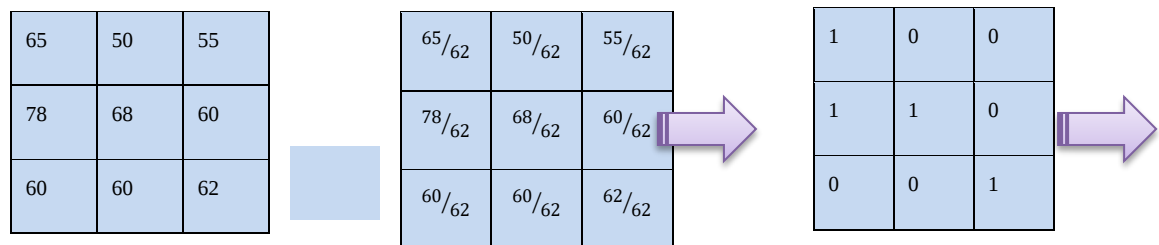
- a. $\text{minInt} = \text{minimum } (5, 7, 7, 5, 4, 6, 7, 8) = 4;$
- b. $\text{maksInt} = \text{maksimum } (5, 7, 7, 5, 4, 6, 7, 8) = 8;$
- c. mengingat $f(y, x)$ bernilai 9 dan lebih besar daripada 8
 (maksInt) maka $g(y, x)$ bernilai 8 ;

- d. seandainya $f(y, x)$ pada keadaan di atas bernilai 2 (bukan 9), $g(y, x)$ akan bernilai 4.

3. Filter Pemerataan

Filter pererataan menurut Costa dan Cesar (2001) dilakukan dengan menggunakan rumus : $g(y, x) = \frac{1}{9} \sum_p^1 = -1 \sum_q^1 = -1 \int (y + p, x + q)$

Sebagai contoh, piksel pada $f(y, x)$ dan kedelapan tetangganya memiliki nilai-nilai kecerahan seperti berikut:



Pada contoh di atas, yang diarsir (yaitu yang bernilai 68) merupakan nilai pada $f(y, x)$. Nilai rerata pengganti untuk $g(y, x)$ dihitung dengan cara seperti berikut:

$$g(y, x) = \frac{1}{9} \times (65 + 50 + 55 + 78 + 68 + 60 + 60 + 60 + 62)$$

$$= 61,7778 \text{ dibulatkan menjadi } 62.$$

Jadi, nilai 68 pada $f(y, x)$ diubah menjadi nilai 62 pada $g(y, x)$. Dan nilai 1 0 0 1 1 0 0 0 1 adalah hasil dari nilai 62 pada $g(y, x)$ disebut bilangan biner yang merupakan sistem bilangan berbasis 2 yang mana penulisannya menggunakan 2 simbol angka, yaitu 0 dan 1.

4. Filter Median

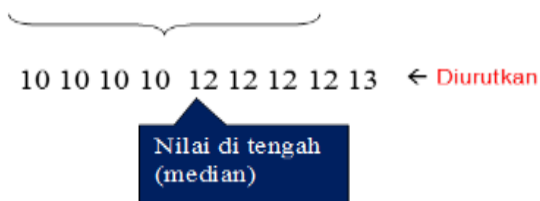
Filter median sangat populer dalam pengolahan citra. Teorinya, filter ini sangat efektif dipakai untuk menghilangkan derau bintang-bintang. Nilai yang lebih baik digunakan untuk suatu piksel ditentukan oleh nilai median dari setiap piksel dan kedelapan piksel tetangga pada 8-ketetanggaan. Secara matematis, filter dapat dinotasikan seperti berikut: $g(y, x) = \text{median}$

$$f(y-1, x-1), f(y-1, x), f(y-1, x+1)$$

$$f(y, x-1), f(y, x), f(y, x+1)$$

$$f(y+1, x-1), f(y+1, x), f(y+1, x+1))$$

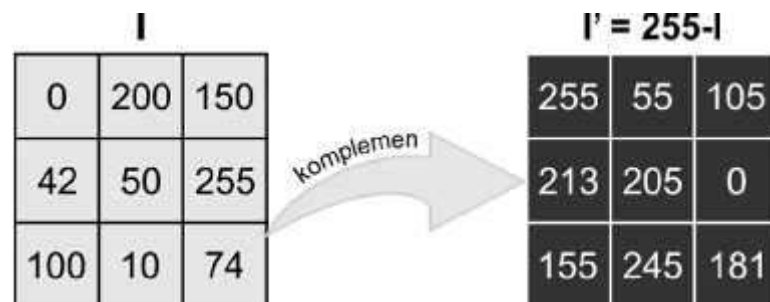
10	13	10
10	10	12
12	12	12



Gambaran operasi filter median dapat dilihat pada gambar sebagai berikut :
Terlihat pada contoh diatas, untuk mendapatkan median, diperlukan pengurutan (sorting) terlebih dahulu.

5. CitraNegatif

Citra negatif merupakan citra yang nilai pikselnya berkebalikan dengan citra aslinya. Untuk citra grayscale 8-bit apabila citra asli disimbolkan dengan I , maka negatif dari citra tersebut adalah $I' = 255 - I$. Contoh perhitungan nilai piksel dari citra negatif ditunjukkan pada gambar di bawah ini:



6. Derau (Noise)

Derau adalah gambar atau piksel yang mengganggu kualitas citra. Derau perlu dihilangkan karena menghalangi pengambilan informasi pada citra tersebut. Seringkali citra yang dihasilkan atau didapatkan memiliki kualitas yang kurang baik. Hal ini dapat disebabkan oleh gangguan teknis, peralatan yang digunakan atau juga dikarenakan kurang baiknya suatu pencahayaan. Solusi masalah tersebut adalah melakukan pemrosesan citra agar dapat mendapatkan citra yang diinginkan. Dalam beberapa aplikasi medis, citra hasil pemindai MRI, CT Scan maupun USG, juga dapat terkena noise. Terlebih lagi USG, citra hitam putih analog yang dihasilkan sangat banyak memiliki noise, sehingga bila dibutuhkan analisa berbantuan komputer, perlu dilakukan preproses untuk memperbaiki kualitas citra (enhancement), agar deteksi selanjutnya bisa lebih tepat atau presisi.

Beberapa gangguan mungkin saja terjadi saat pengambilan gambar, seperti kamera tidak fokus atau munculnya bintik-bintik yang bisa terjadi karena proses pengambilan gambar yang tidak sempurna. Setiap gangguan pada citra dinamakan derau, yang tidak hanya terjadi karena ketidaksempurnaan dalam proses pengambilan gambar, tetapi dapat disebabkan juga oleh noda kotoran yang terjadi pada citra setelah pengambilan citra. Berdasarkan bentuk dan karakteristiknya, derau pada citra dibedakan menjadi beberapa macam yaitu:

a. Derau salt and papper

Derau ini disebut juga dengan derau impulse (impulse noise), shout noise atau derau biner (binary noise). Degradasi derau ini disebabkan oleh gangguan yang tajam atau tiba-tiba (sharp and sudden) pada sinyal citra. Kenampakan pada citra berupa titik-titik (piksel) hitam atau putih yang tersebar pada citra.

b. Derau Gaussian

Derau Gaussian merupakan bentuk ideal dari derau putih, Derau Gaussian dikatakan derau putih karena mempunyai distribusi normal.

c. Derau Speckle

Derau Speckle dapat disebut juga dengan derau multiplikatif (multiplicative noise). Derau speckle sering dijumpai pada aplikasi radar.

d. Derau Periodik

Derau Periodik sifatnya periodik (bukan acak) menghasilkan derau periodik. Citra yang terkorupsi oleh derau periodik secara visual tampak terdapat garis-garis pada citra.

7. Cara Kerja Operasi Titik

Untuk mengimplementasikan proses pengolahan citra dengan operasi titik, penulis membuat program dengan menggunakan MATLAB. Secara garis besar program ini akan melakukan langkah-langkah berikut:

- a. Memilih file gambar yang terkena noise.
- b. Menampilkan citra asli.
- c. Memilih salah satu filter.
- d. Melakukan pemrosesan pada citra yang ingin diperbaiki dengan menggunakan Operasi Titik
- e. Menampilkan citra hasil pemrosesan.

Langkah pertama yang dilakukan adalah membaca file citra yang akan diproses dan menyimpan informasi dari semua pixelnya ke dalam sebuah matriks. Setelah itu baru digunakan perintah `imshow` untuk menampilkan citra yang asli yang sudah terkena noise dan muncul pada `axes1`. Untuk menghilangkan noise, lebih dulu dibuat sebuah matriks `h` yang dimensinya sama dengan dimensi dari matriks untuk menyimpan hasil, yaitu 256×256 . Matriks `h` ini diinisialisasi dengan diisi nilai satu pada setiap elemennya. Selanjutnya bagian tertentu dari matriks yang menunjukkan area dimana noise berada pada citra diubah menjadi nol. Setelah bagian tertentu dari matriks tersebut diberi nilai nol, sedangkan yang lainnya tetap bernilai satu, maka masing-masing elemen matriks tersebut dikalikan dengan elemen-elemen yang bersesuaian dari matriks yang lainnya.

Perkalian ini akan menghasilkan pixel baru dimana area noise telah hilang karena dikalikan dengan nol. Pada visualisasi hilangnya area noise ini ditunjukkan dengan warna yang bagus menyerupai gambar yang asli sebelum gambar bernoise. Setelah itu hasil citra yang telah di perbaiki tadi muncul di axes2 dengan gambar yang berbeda pada axes1.

8. Contrast

Jika sebuah citra yang mempunyai nilai keabuan yang tidak terlalu berbeda untuk semua titik, dimana titik tergelap dalam citra tidak mencapai hitam pekat dan titik paling terang dalam citra tidak berwarna putih cemerlang. Dengan peningkatan kontras maka titik yang cenderung gelap menjadi lebih gelap dan yang cenderung terang menjadi lebih cemerlang. Peningkatan kontras dapat dilakukan dengan bermacam rumus, salah satunya adalah :

$$Ko = G(Ki - P) + P$$

G = Koefisien penguatan kontras

P = Nilai skala keabuan yang dipakai sebagai pusat pengontrasan

9. Pixel (Picture Element)

Piksel atau *Pixel* adalah singkatan dari Picture Element, yaitu representasi terkecil sebuah gambar atau grafis yang membuatnya dapat membentuk sebuah objek dengan jelas. Elemen ini berbentuk kotak kecil yang dapat mengambil lebih dari satu juta warna berbeda. Piksel memiliki tiga warna utama yang biasanya disebut dengan istilah RGB (Red, Green, and Blue). Setiap perubahan warna pada piksel dapat menghasilkan 16 juta warna yang berbeda, sehingga akan menghasilkan visual objek dengan warna sesuai tujuannya. RGB juga dapat

digunakan pada teks HTML untuk memberikan warna pada latar belakang website. RGB ini biasa digunakan untuk menghasilkan warna pada grafik komputer. Selain warna, Anda juga harus memperhatikan *density* dari piksel, yaitu kepadatan yang menunjukkan keseluruhan jumlahnya pada objek tertentu. Hal itu dapat mempengaruhi ketajaman objek dan kecerahannya. Singkatnya, semakin tinggi pixel density, maka objeknya akan semakin jelas dan tajam.

10. Brightness

Brightness adalah proses untuk kecerahan citra, jika intensitas pixel dikurangi dengan nilai tertentu maka citra akan menjadi lebih gelap, dan sebaliknya jika intensitas pixelnya ditambah dengan nilai tertentu maka akan lebih terang. Brightness adalah nama lain dari tingkat kecerahan/intensitas cahaya. Elemen ini menyatakan banyaknya cahaya yang diterima oleh mata. Elemen ini dapat dirasakan sebagai lampu penerang berwarna putih ketika kita melihat suatu benda. Semakin terang cahaya lampu tersebut (Tingkat kecerahan/brightness tinggi), benda yang kita lihat akan semakin putih. Semakin redup (Tingkat kecerahan/brightness rendah), benda yang kita lihat semakin gelap. Dan ketika tidak ada cahaya lampu (Tingkat kecerahan/brightness = 0), benda yang kita lihat berwarna hitam. Elemen brightness ini dapat kita peroleh dengan menggunakan rumus: $B1 = B0(x,y) + konstanta$

11. Pengolahan Citra Digital

Saat ini, istilah citra digital tentunya sudah sangat populer. Ada banyak peralatan elektronik yang dapat menghasilkan citra digital, misalnya kamera digital dan mikroskop digital. Selain itu, kini software untuk mengolah citra

digital juga sudah sangat populer digunakan oleh para pengguna. Misalnya Adobe Photoshop yang digunakan untuk mengolah foto.

Pengolahan citra digital yaitu bidang ilmu yang mempelajari mengenai bagaimana suatu citra itu dibentuk, diolah dan dianalisis sehingga dapat menghasilkan sebuah informasi yang bisa dipahami oleh manusia. Dapat pula dijelaskan bahwa pengolahan citra digital adalah manipulasi dan interpretasi digital dari citra dengan bantuan komputer. Citra sendiri merupakan fungsi dari intensitas cahaya yang direpresentasikan ke dalam bidang 2 dimensi.

Pengertian Pengolahan Citra Digital Menurut Para Ahli:

- a. **Pengolahan Citra Digital Menurut Jian (1989)** adalah pemrosesan gambar dua dimensi melalui computer digital.
- b. **Pengolahan Citra Digital Menurut Efford (2000)** adalah untuk memanipulasi dan memodifikasi citra dengan menggunakan berbagai cara.
- c. **Pengolahan Citra Digital Menurut Suhandy (2003)** adalah sebuah teknologi visual yang dipakai untuk mengamati dan menganalisis sebuah objek tanpa berhubungan secara langsung dengan objek yang diamati itu.

12. Multimedia

Secara bahasa, multimedia terdiri dari dua kata, yaitu multi dan media. Multi banyak, sedangkan media artinya adalah sesuatu yang digunakan untuk menyampaikan pesan (perantara) jadi pengertian multimedia adalah perantara pesan yang terdiri lebih dari 1 elemen atau banyak.

Multimedia adalah sarana komunikasi yang mengintegrasikan teks, grafik, gambar diam dan bergerak, animasi audio dan media lain. Kombinasi elemen tersebut dapat ditampilkan, disimpan, dikirim dan diproses dengan sebuah perangkat tertentu.

Pengertian Multimedia Menurut Para Ahli:

- a. **Menurut McCormick (1996)** Multimedia adalah kombinasi tiga elemen, suara, gambar dan video.
- b. **Menurut Rosch (1996)** Multimedia sebagai kombinasi dari komputer dan video.
- c. **Menurut Gumelar (2014)** Multimedia adalah penggunaan komputer untuk menyajikan dan menggabungkan teks, animasi, suara, gambar dan video dengan tools (alat bantu) dan link sehingga pengguna dapat melakukan navigasi, interaksi, berkarya dan berkomunikasi.

Sesuai dengan namanya, suatu media dapat dikatakan multimedia jika terdiri dari 2 atau lebih elemen. Sebagai contoh, sebuah brosur dapat dikatakan multimedia karena ia terdiri dari elemen teks dan gambar.

Sebuah video klip dapat dikatakan media karena terdapat unsur suara, gambar atau juga teks. Ini berbeda dengan misalnya rekaman suara kaset. Ia tidak bisa dikatakan multimedia karena hanya mengandung 1 elemen, yakni suara.

13. Sumbu x, y

Sumbu X dan sumbu Y merupakan konsep dasar dalam matematika yang digunakan untuk memberikan koordinat pada sebuah titik dalam bidang dua dimensi, yang sering disebut juga dengan koordinat kartesian. Sumbu X dan sumbu Y membentuk suatu sistem koordinat di mana sumbu X melintang secara horizontal dan sumbu Y melintang secara vertikal.

Sumbu X biasanya disebut juga sebagai sumbu horizontal. Pada sumbu ini, bilangan yang bernilai positif berada di sebelah kanan nol, sedangkan yang bernilai negatif berada di sebelah kiri nol. Titik yang terletak di sumbu X memiliki koordinat $(x, 0)$, di mana x adalah jarak titik tersebut terhadap origo (titik $(0, 0)$). Sumbu Y, atau sumbu vertikal, berada tegak lurus terhadap sumbu X. Bilangan yang bernilai positif berada di atas nol, sedangkan yang bernilai negatif berada di bawah nol. Titik yang terletak di sumbu Y memiliki koordinat $(0, y)$, di mana y adalah jarak titik tersebut terhadap origo.

14. Adobe Photoshop

Photoshop banyak digunakan oleh *fotografer* digital dan perusahaan iklan sehingga dianggap sebagai pemimpin pasar (*market leader*) sebagai perangkat lunak pengolah gambar atau foto dan bersama *Adobe Acrobat*, *photoshop* juga merupakan produk produksi terbaik dari *Adobe Systems*.

Photoshop CS (Creative Suite) merupakan versi kedelapan dari *Adobe Photoshop*, *Adobe Photoshop CS2* merupakan versi kesembilan, *Adobe Photoshop CS3* merupakan versi sepuluh, *Adobe Photoshop CS4* merupakan versi kesebelas, *Adobe Photoshop CS5* merupakan versi keduabelas, *Adobe Photoshop CS6*

merupakan versi ketigabelas dan yang terakhir adalah *Adobe Photoshop CS7* merupakan versi keempatbelas dan lanjut ke versi seterusnya.

Adapun fungsi *adobe photoshop* secara teknik:

- a. Mengkombinasi 2 gambar atau lebih menjadi satu gambar baru.
- b. Menghilangkan atau menghapus objek pada suatu gambar.
- c. Memberi atau merubah warna
- d. Merubah ukuran gambar menjadi lebih kecil ataupun menjadi lebih besar
- e. Mempertajam warna dan kualitas gambar
- f. Membersihkan atau memperindah gambar
- g. Mengaburkan gambar agar terlihat samar-samar
- h. Meningkatkan intensitas cahaya gambar agar terlihat lebih terang.
- i. Memotong gambar

15. Histogram

Merupakan representasi grafis untuk distribusi warna dari citra digital. Sumbu ordinat vertikal merupakan representasi piksel dengan nilai tonal dari tiap-tiap deret bin pada sumbu axis horizontalnya. Sumbu axis terdiri dari deret logaritmik bin *densitometry* yang membentuk rentang luminasi atau *exposure range* yang mendekati respon *spectral sensitivity* visual mata manusia. Deret bin pada kerapatan yang terpadat mempunyai interval yang relatif sangat linear dengan variabel mid-tone terletak tepat di tengahnya.

Pada histogram fotografis, grafis batang tidak mempunyai luasan yang menunjukkan jumlah piksel pada tiap bin. Grafis batang menjadi grafis garis vertikal yang mewakili seluruh jumlah piksel pada deret bin luminasi tersebut. Sebagai contoh, sebuah foto ukuran 4288x2848 piksel yang mempunyai 1 tone akan mempunyai histogram dengan 1 garis lurus vertikal pada nilai bin luminasinya, bukan berupa 12.212.224 garis vertikal yang mempunyai panjang sama.

16. MATLAB (Matrix Laboratory)

Pertama dirilis di tahun 1970 oleh *MathWorks*, *MATLAB* adalah salah satu platform yang paling banyak digunakan untuk mengolah angka dan bahasa pemrograman.

Menurut *MathWorks*, *MATLAB* adalah platform pemrograman yang menggunakan bahasa berbasis matriks sehingga umumnya digunakan untuk menganalisis data serta menciptakan pemodelan dan aplikasi. Aplikasi ini juga sering dimanfaatkan untuk mengembangkan *deep learning*, *machine learning*, dan hal-hal terkait lainnya.

17. JPG/JPEG (Joint Photographic Experts)

JPG adalah jenis data yang dikembangkan oleh *Joint Photographic Experts Assemble (JPEG)* yang dijadikan standar untuk para fotografer profesional. Seperti metode yang digunakan oleh format ZIP yang digunakan untuk menemukan pengulangan (redundancy) dalam data untuk kemudian dikompresi. JPG mengompresi data gambar dengan cara mengurangi bagian-bagian dari gambar untuk memblok pixel dalam gambar tersebut. Kompresi JPG mempunyai

kekurangan yang bersifat permanen, namun teknologi ini hanya digunakan untuk menyimpan data yang besar di media penyimpanan yang terbatas, bukan untuk manipulasi foto. JPG sudah digunakan dan menjadi standar gambar di internet karena ia bisa dikompresi hingga ukuran kecil.

Data JPG tertentu bisa dikompres dengan rasio perbandingan 2:1 sampai paling tinggi 100:1, tergantung pengaturan yang anda berikan. Sewaktu koneksi internet yang tersedia di bumi ini masih berupa koneksi dial-up, JPG adalah satu-satunya jenis data yang bisa dikirimkan dan dilihat secara bebas. File JPG menggunakan teknik kompresi yang menyebabkan kualitas gambar turun (lossy compression), maka format gambar ini tidak terlalu baik untuk digunakan menyimpan gambar pajangan atau artistik. Setiap kali menyimpan ke tipe JPG dari tipe lain, ukuran gambar biasanya mengecil, dan kualitasnya turun dan tidak dapat dikembalikan lagi. Ukuran file BMP dapat turun menjadi seper sepuluh setelah dikonversi menjadi JPG. Meskipun dengan penurunan kualitas gambar, pada gambar-gambar tertentu (misalnya pemandangan), penurunan kualitas gambar hampir tidak terlihat mata.

18. PNG (Portable Network Graphics)

PNG adalah kepanjangan dari *Portable Network Graphics* atau bisa diplesetkan menjadi “PNG-Not-GIF“. Dikembangkan sebagai alternatif lain untuk GIF, yang menggunakan paten dari LZW–algoritma kompresi. PNG adalah format gambar yang sangat baik untuk grafis internet, karena mendukung transparansi didalam perambah (browser) dan memiliki keindahan tersendiri yang tidak bisa diberikan GIF atau bahkan JPG. Bisa disebut sebagai salah satu format

yang merupakan gabungan dari format JPG dan GIF. Untuk tipe ini mampu untuk gradiasi warna. Karena sifat transparannya yang tidak pecah-pecah.

PNG yang masuk kelas 24-bit ini cocok untuk membuat screenshoot. Ia bisa mereproduksi gambar desktop dari tiap piksel ke piksel secara detil. PNG juga mendukung kelas 8-bit seperti GIF, sekaligus 24-bit seperti JPG. Ia juga tidak pecah-pecah, bisa mengkompresi gambar dari proses fotografi tanpa banyak mengurangi kualitas gambarnya. Namun PNG cukup besar ukurannya diantara JPG dan GIF, serta tidak didukung oleh perambah / browser yang lama. Tipe file PNG merupakan solusi kompresi yang powerful dengan warna yang lebih banyak (24 bit RGB + alpha). Berbeda dengan JPG yang menggunakan teknik kompresi yang menghilangkan data, file PNG menggunakan kompresi yang tidak menghilangkan data (lossles compression).

Kelebihan file PNG adalah adanya warna transparan dan alpha. Warna alpha memungkinkan sebuah gambar transparan, tetapi gambar tersebut masih dapat dilihat mata seperti samar-samar atau bening. File PNG dapat diatur jumlah warnanya hingga 64 bit (true color + alpha) sampai indexed color 1 bit. Dengan jumlah warna yang sama, kompresi file PNG lebih baik daripada GIF, tetapi memiliki ukuran file yang lebih besar daripada JPG. Kekurangan tipe PNG adalah belum populer sehingga sebagian browser tidak mendukungnya. Format PNG ini diperkenalkan untuk menggantikan format GIF.

PNG mempunyai faktor kompresi yang lebih baik dibandingkan dengan GIF (kurang lebih 5%-25% lebih baik dibanding format GIF). Tetapi ada satu fasilitas dari GIF yang tidak terdapat pada PNG format yaitu dukungan terhadap

penyimpanan multi format untuk keperluan animasi. Untuk keperluan pengolahan gambar, meskipun format PNG bisa dijadikan alternatif selama proses pengolahan grafis namun format JPEG masih menjadi pilihan yang lebih baik.

19. BMP (Bitmap)

Bitmap adalah representasi dari citra grafis yang terdiri dari susunan titik (pixel) yang tersimpan di memori komputer. Nilai setiap titik diawali oleh satu bit data (untuk gambar hitam putih) atau lebih (untuk gambar berwarna). Kerapatan titik-titik tersebut dinamakan resolusi, yang menunjukkan seberapa tajam gambar ini ditampilkan, ditunjukkan dengan jumlah baris dan kolom (contoh 1024×768).

Untuk menampilkan citra bitmap pada monitor atau mencetaknya pada printer, komputer menterjemahkan bitmap ini menjadi pixel (pada layar) atau titik tinta (pada printer). Beberapa format file bitmap yang populer adalah BMP, PCX dan TIFF. Gambar bitmap sangat bergantung pada resolusi. Jika gambar diperbesar maka gambar akan tampak kurang halus atau pecah, sehingga mengurangi detailnya. Selain itu gambar bitmap akan mempunyai ukuran file yang lebih besar. Semakin besar resolusi gambar akan semakin besar pula ukuran filenya.

Gambar dengan tampilan 100% Gambar dengan tampilan 500% akan mempunyai perbedaan yakni biasanya kalau pada gambar yang berukuran 100%, maka gambar masih terlihat jernih. Namun pada gambar yang dizoom sampai dengan 500% maka gambar akan tampak tidak tajam lagi dan terkesan membentuk kotak-kotak yang itu merupakan pixel. Contoh software yang berbasis bitmap adalah Adobe Photoshop, Paint, CorelPhotoPaint, dan lain-lain.

20. Dasar Pemrograman

Dasar-dasar pemrograman adalah konsep dan teknik fundamental yang perlu dipahami oleh siapa pun yang ingin mulai belajar pemrograman. Pemahaman yang kuat tentang dasar-dasar ini akan membantu dalam memahami logika, struktur, dan cara kerja program

a. Pengertian Pemrograman

Pemrograman adalah proses merancang, menulis, menguji, dan memelihara kode atau program yang menginstruksikan komputer untuk melakukan tugas tertentu. Dalam konteks komputasi, pemrograman memungkinkan manusia untuk membuat perangkat lunak yang dapat menjalankan berbagai macam tugas, mulai dari kalkulasi matematis sederhana hingga operasi kompleks dalam kecerdasan buatan.

Bahasa Pemrograman adalah bahasa yang digunakan untuk menulis program. Setiap bahasa memiliki aturan (syntax) dan struktur yang berbeda. Contoh populer termasuk:

- 1) Python: Digunakan dalam pengembangan web, data science, machine learning, dan otomatisasi.
- 2) JavaScript: Bahasa yang umum digunakan untuk pengembangan web, terutama pada sisi front-end.
- 3) Java: Sering digunakan dalam aplikasi berbasis Android dan sistem enterprise.
- 4) C++: Dikenal karena kecepatan dan sering digunakan dalam pengembangan sistem perangkat keras dan game.

b. Pengertian Algoritma

Serangkaian instruksi atau langkah-langkah yang dirancang secara logis dan sistematis untuk menyelesaikan suatu masalah atau mencapai tujuan tertentu. Dalam komputasi, algoritma biasanya digunakan untuk mengolah data, memecahkan masalah, atau menjalankan tugas dalam urutan yang telah ditentukan.

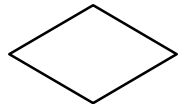
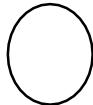
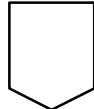
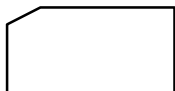
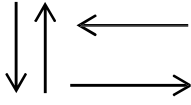
1. Algoritma Teks

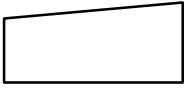
Metode atau serangkaian instruksi yang digunakan untuk memproses, menganalisis, dan memanipulasi data teks. Algoritma ini sering digunakan dalam berbagai aplikasi seperti pengenalan pola, pemrosesan bahasa alami (Natural Language Processing atau NLP), dan pencarian informasi. Algoritma teks biasanya bekerja dengan menguraikan teks menjadi komponen-komponen yang lebih kecil seperti kata atau frasa untuk dianalisis atau digunakan dalam proses lebih lanjut.

2. Algoritma Flowchart

Flowchart atau bagan alur adalah diagram yang menampilkan langkah-langkah dan keputusan untuk melakukan sebuah proses dari suatu program. Setiap langkah digambarkan dalam bentuk diagram dan dihubungkan dengan garis atau arah panah.

Tabel 2. 1 Simbol Flowchart.

GAMBAR	NAMA	KETERANGAN
	<i>Terminal</i>	Berfungsi untuk mengawali suatu program dan mengakhirinya.
	<i>Input / Output</i>	Berfungsi untuk melakukan suatu proses input atau output.
	<i>Process</i>	Berfungsi untuk tempat melakukan suatu proses yang terjadi pada sistem.
	<i>Decision</i>	Berfungsi untuk menyelesaikan suatu kondisi yang akan menghasilkan dua jawaban berbeda iya/tidak
	<i>Connector</i>	Berfungsi sebagai sambungan proses yang berbeda tetapi tetap halaman yang sama.
	<i>Offline Connector</i>	Berfungsi sebagai sambungan proses yang berbeda tetapi pada halaman yang berbeda.
	<i>Predefined Process</i>	Berfungsi sebagai tempat pengolahan yang akan diberikan suatu harga.
	<i>Punched Card</i>	Berfungsi sebagai penyampaian <i>input</i> bermula dari kartu atau <i>output</i> ditulis pada kartu.
	<i>Document</i>	Berfungsi sebagai percetakan dalam bentuk suatu dokumen.
	<i>Flow</i>	Menghubungkan antara simbol yang satu dengan simbol yang lain.

	<i>Manual Operation</i>	Menunjukkan pengolahan yang tidak dilakukan oleh <i>computer</i> .
	<i>Manual Input</i>	Dilakukan untuk memasukkan data secara manual <i>on-line keyboard</i> .
	<i>Predefine Process</i>	Digunakan untuk pelaksanaan suatu bagian (<i>sub-program</i>) / prosedur.

3. UML (Unified Modelling Language)

UML (Unified Modelling Language) adalah suatu metode dalam pemodelan secara visual yang digunakan sebagai sarana perancangan sistem berorientasi objek. Awal mulanya, *UML* diciptakan oleh *Object Management Group* dengan versi awal 1.0 pada bulan Januari 1997.

UML juga dapat didefinisikan sebagai suatu bahasa standar visualisasi, perancangan, dan pendokumentasian sistem, atau dikenal juga sebagai bahasa standar penulisan blueprint sebuah *software*.

Perlu kamu ketahui bahwa sistem yang baik itu berawal dari perancangan dan pemodelan yang matang. Salah satu yang bisa kamu praktekan, yaitu dengan menggunakan *UML*. Adapun tujuan dan fungsi perlu adanya *UML* yaitu sebagai berikut:

- a. Dapat memberikan bahasa pemodelan visual atau gambar kepada para pengguna dari berbagai macam pemrograman maupun proses umum rekayasa.
- b. Dapat memberikan bahasa pemodelan visual atau gambar kepada para

pengguna dari berbagai macam pemrograman maupun proses umum rekayasa.

- c. Memberikan suatu gambaran model atau sebagai bahasa pemodelan visual yang ekspresif dalam pengembangan sistem.
- d. Tidak hanya menggambarkan model sistem software saja, namun dapat memodelkan sistem berorientasi objek.
- e. Mempermudah pengguna untuk membaca suatu sistem.

Adapun diagram *UML* yang digunakan dalam penelitian ini antara lain, sebagai berikut:

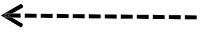
4. Use Case Diagram

Use Case diagram sendiri adalah proses penggambaran yang dilakukan untuk menunjukkan hubungan antara pengguna dengan sistem yang dirancang. Hasil representasi dari skema tersebut dibuat secara sederhana dan bertujuan untuk memudahkan *user* dalam membaca informasi yang diberikan.

Tabel 2. 2 Simbol Use Case Diagram

GAMBAR	NAMA	KETERANGAN
	<i>Use case</i>	Menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktif, yang dinyatakan dengan menggunakan kata kerja.
	<i>Actor</i>	<i>Actor</i> atau Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i> , tetapi tidak memiliki control terhadap <i>use case</i>
	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek yang lainnya.
	<i>Generalization</i>	Hubungan dimana objek anak berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk.
	<i>Include</i>	Pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.

Lanjutan Tabel 2.1 Simbol Use Case Diagram

GAMBAR	NAMA	KETERANGAN
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri akan mempengaruhi elemenyang bergantung padanya elemen yang tidak mandiri.
	<i>Extend</i>	Menspesifikasikan bahwa use case target memperluas perilaku dari use case sumber pada suatu titik yang diberikan
	<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
	<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya.
	<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

5. Activity Diagram

Activity diagram, dalam bahasa Indonesia diagram aktivitas, yaitu diagram yang dapat memodelkan proses-proses yang terjadi pada sebuah sistem. Runtutan proses dari suatu sistem digambarkan secara *vertikal*. *Activity diagram* merupakan pengembangan dari *Use Case* yang memiliki alur aktivitas.

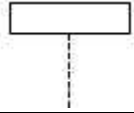
Tabel 2. 3 Simbol Actifity Diagram

GAMBAR	NAMA	KETERANGAN
	<i>Actifitye</i>	Menampilkan suatu objek bagaimna cara saling berinteraksi antara satu dan lainnya.
	<i>Actione</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi
	<i>Initial Nodee</i>	Berfungsi untuk mengawali suatu objek
	<i>Actifity Finale Nodee</i>	Berfungsi untuk mengakhiri suatu objek
	<i>Fork Nodee</i>	Suatu alur yang akan berubah menjadi beberapa aliran pada kondisi tertentu.

6. Sequence Diagram

Sequence diagram atau diagram urutan adalah sebuah diagram yang digunakan untuk menjelaskan dan menampilkan interaksi antar objek- objek dalam sebuah sistem secara terperinci. Selain itu sequence diagraph juga akan menampilkan pesan atau perintah yang dikirim, beserta waktu pelaksanaannya. Objek-objek yang berhubungan dengan berjalannya proses operasi biasanya diurutkan dari kiri ke kanan.

Tabel 2. 4 Simbol Sequence Diagram

GAMBAR	NAMA	KETERANGAN
	<i>LifeLinee</i>	Suatu objek berfungsi untuk melakukan interaksi antarmuka.
	<i>Messagee</i>	Berfungsi untuk berkomunikasi antara objek yang memiliki informasi tentang perilaku yang terjadi
	<i>Messagee</i>	Berfungsi untuk berkomunikasi antara objek yang memiliki informasi tentang perilaku yang terjadi

7. Class Diagram

Class Diagram adalah jenis diagram yang digunakan dalam UML (Unified Modeling Language) untuk memodelkan struktur sistem perangkat lunak berbasis objek. Class diagram menggambarkan kelas-kelas, atribut-atribut, metode-metode, dan hubungan antar kelas dalam suatu sistem. Ini adalah alat visual yang penting dalam pemrograman berorientasi objek (OOP).

Komponen Utama dalam Class Diagram:

- Kelas (Class): Kelas diwakili oleh persegi panjang yang dibagi menjadi tiga bagian:
- Nama Kelas: Bagian paling atas yang menampilkan nama kelas.
- Atribut: Bagian tengah yang berisi atribut atau properti kelas.
- Metode: Bagian bawah yang berisi operasi atau metode yang dilakukan oleh kelas. Contoh:

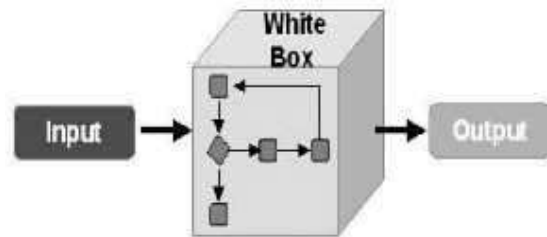


Gambar 2.3 *Class diagram*

8. Pengujian Aplikasi

a. White Box Testing

Whitebox testing adalah metode pengujian perangkat lunak yang memungkinkan penguji memiliki akses penuh ke struktur internal, algoritma, dan kode sumber dari sistem yang diuji. Dalam metode ini, penguji memahami bagaimana sistem bekerja secara internal dan mengujinya berdasarkan pengetahuan tersebut. Whitebox juga dikenal sebagai structural testing atau glass-box testing. Secara keseluruhan, kedua metode ini saling melengkapi dalam proses pengembangan perangkat lunak. Whitebox testing membantu dalam menemukan kesalahan internal pada kode, sedangkan blackbox testing memastikan bahwa sistem bekerja sesuai harapan dari perspektif pengguna.



Gambar 2. 4 *White box testing*

Ciri-ciri Pengujian *White Box* Testing:

- 1) Pengujian berbasis pengetahuan: Penguji mengetahui cara kerja internal sistem dan menggunakan informasi ini untuk membuat tes yang tepat.
- 2) Fokus pada struktur: Metode ini menguji arsitektur, aliran data, jalur eksekusi, dan logika kode yang digunakan dalam program.
- 3) Deteksi bug mendalam: Dapat menemukan bug yang tersembunyi dalam kode, seperti kesalahan logika atau cacat dalam aliran kontrol program.
- 4) Alat dan teknik: Teknik yang umum digunakan dalam whitebox testing termasuk path testing, control flow testing, data flow testing, dan branch testing.

Berikut adalah langkah-langkah umum dalam pengujian *white box*:

- 1) Analisis Kode Sumber: Pelajari kode sumber aplikasi untuk memahami struktur internal dan alur logika. Identifikasi fungsi, alur kontrol, dan alur data.
- 2) Identifikasi Kasus Uji: Berdasarkan analisis kode, identifikasi berbagai kondisi, cabang, dan jalur dalam kode. Ini bisa termasuk semua jalur eksekusi, kondisi logika, dan pernyataan

dalam kode.

- 3) Desain Kasus Uji: Buat kasus uji yang mencakup semua jalur dan kondisi yang diidentifikasi. Pastikan kasus uji dirancang untuk menguji semua cabang dan jalur eksekusi yang mungkin terjadi.
- 4) Buat Skrip Uji: Kembangkan skrip uji atau prosedur uji yang diperlukan untuk menjalankan kasus uji. Skrip ini akan menjalankan pengujian sesuai dengan desain yang telah dibuat.
- 5) Jalankan Kasus Uji: Eksekusi kasus uji dengan menggunakan skrip uji yang telah dibuat. Pastikan untuk memeriksa hasil dan bandingkan dengan hasil yang diharapkan.
- 6) Tulis Laporan Hasil Uji: Dokumentasikan hasil uji, termasuk hasil yang berhasil dan masalah yang ditemukan. Laporkan masalah yang memerlukan perhatian lebih lanjut.
- 7) Perbaiki dan Uji Ulang: Jika ditemukan masalah, koordinasikan dengan pengembang untuk memperbaikinya. Setelah perbaikan dilakukan, jalankan uji ulang untuk memastikan bahwa masalah telah teratasi dan tidak ada masalah baru yang muncul.
- 8) Lakukan Pengujian Tambahan: Jika perlu, lakukan pengujian tambahan untuk memastikan bahwa perubahan atau perbaikan tidak memengaruhi bagian lain dari aplikasi.
- 9) Evaluasi dan Penutupan: Setelah semua pengujian selesai, evaluasi keseluruhan hasil pengujian dan pastikan semua kasus

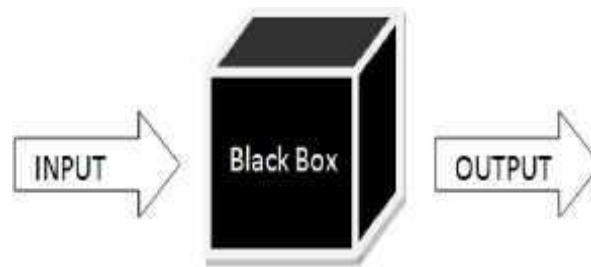
uji telah berhasil. Dokumentasikan proses pengujian secara menyeluruh dan tutup fase pengujian.

Contoh penggunaan *white box testing* sebagai berikut:

- 1) Menguji bagaimana variabel diproses dalam suatu program.
- 2) Memeriksa jalur logika dalam algoritma, seperti memastikan semua cabang if-else dieksekusi.
- 3) Digunakan dalam pengujian unit (*unit testing*), di mana pengembang memeriksa bagian kecil dari kode, seperti fungsi atau modul.

b. Black Box Testing

Blackbox testing adalah metode pengujian di mana penguji tidak memiliki pengetahuan atau akses ke kode sumber atau struktur internal dari sistem yang diuji. Penguji hanya mengetahui apa yang seharusnya dilakukan oleh sistem berdasarkan spesifikasi dan input-output, tanpa mengetahui bagaimana cara kerja internalnya. Secara keseluruhan, kedua metode ini saling melengkapi dalam proses pengembangan perangkat lunak. Whitebox testing membantu dalam menemukan kesalahan internal pada kode, sedangkan blackbox testing memastikan bahwa sistem bekerja sesuai harapan dari perspektif pengguna.



Gambar 2. 5 *Black box testing*

Ciri-ciri pengujian *Blackbox testing*:

- 1) Pendekatan berbasis fungsional: Pengujian dilakukan berdasarkan fungsionalitas eksternal sistem tanpa memperhatikan bagaimana sistem tersebut diprogram.
- 2) Fokus pada hasil: Penguji memvalidasi bahwa sistem bekerja sesuai dengan yang diharapkan berdasarkan input yang diberikan dan keluaran yang dihasilkan.
- 3) Pengujian pengguna akhir: *Blackbox testing* sering kali mirip dengan cara pengguna akhir akan menggunakan perangkat lunak.
- 4) Alat dan teknik : Teknik umum termasuk *equivalence partitioning*, *boundary value analisis* dan *decision table testing*.

Berikut adalah langkah-langkah umum dalam pengujian *black box*:

- 1) Memahami Spesifikasi dan Persyaratan:
 - a) Pelajari dokumentasi spesifikasi fungsional, persyaratan sistem, dan dokumen desain untuk memahami apa yang diharapkan dari sistem.

- b) Identifikasi fungsionalitas utama, alur kerja, dan kebutuhan pengguna.

2) Perencanaan Pengujian:

- a) Buat rencana pengujian yang mencakup tujuan pengujian, ruang lingkup, sumber daya yang diperlukan, dan jadwal.
- b) Tentukan jenis pengujian *black box* yang akan dilakukan, seperti pengujian fungsional, pengujian batas, pengujian komponen, dll.

3) Desain Kasus Uji:

- a) Buat kasus uji berdasarkan spesifikasi fungsional dan persyaratan sistem. Pastikan untuk mencakup berbagai skenario pengguna, input yang valid, dan input yang tidak valid.
- b) Rancang kasus uji untuk menguji fungsi-fungsi sistem, alur kerja, dan interaksi antar komponen.

4) Persiapkan Lingkungan Pengujian:

- a) Siapkan lingkungan pengujian yang mencakup perangkat keras, perangkat lunak, dan konfigurasi yang diperlukan untuk menjalankan pengujian.
- b) Pastikan semua data yang diperlukan dan konfigurasi sistem sesuai dengan spesifikasi pengujian.

5) Jalankan Kasus Uji:

- a) Eksekusi kasus uji yang telah dirancang. Masukkan data uji

dan amati bagaimana sistem merespons.

- b) Bandingkan hasil yang diperoleh dengan hasil yang diharapkan untuk menentukan apakah fungsionalitas sistem berfungsi dengan baik.

6) Dokumentasikan Hasil Uji:

- a) Catat hasil pengujian, termasuk kasus uji yang berhasil dan yang gagal.
- b) Dokumentasikan setiap masalah yang ditemukan, lengkap dengan deskripsi masalah, langkah-langkah untuk mereproduksi, dan hasil yang diharapkan versus yang diperoleh.

7) Tindak Lanjut dan Perbaikan:

- a) Laporkan masalah kepada tim pengembangan untuk diperbaiki.
- b) Setelah perbaikan dilakukan, jalankan uji ulang untuk memastikan masalah telah teratasi dan tidak ada masalah baru.

8) Pengujian Regresi:

- a) Lakukan pengujian regresi untuk memastikan bahwa perbaikan atau perubahan tidak memengaruhi bagian lain dari sistem yang sudah ada.

9) Evaluasi dan Penutupan:

- a) Tinjau hasil pengujian secara keseluruhan dan pastikan

semua kasus uji telah dieksekusi.

- b) Dokumentasikan proses pengujian dan hasilnya secara menyeluruh. Persiapkan laporan akhir yang mencakup ringkasan temuan dan rekomendasi.

Contoh penggunaan *black box testing* sebagai berikut:

- 1) Menguji sebuah aplikasi berdasarkan input yang diberikan dan memeriksa apakah output sesuai dengan spesifikasi.
- 2) Digunakan dalam pengujian sistem (*system testing*) atau pengujian penerimaan (*acceptance testing*), di mana penguji memvalidasi bahwa aplikasi berfungsi sesuai dengan kebutuhan pengguna.

BAB III

METODE PENELITIAN

A. Tempat dan Waktu Penelitian

Waktu yang dibutuhkan dalam pelaksanaan penelitian ini adalah $\pm$ 6 (enam) bulan difebruari – agustus 2024, adapun lokasi penelitian akan dilakukan di perpustakaan mengambil referensi dari buku pengolahan citra, website atau situs - situs yang berhubungan dengan penelitian ini..

B. Jenis Penelitian

Jenis penelitian yang digunakan adalah penelitian *eksperimen* dimana memberikan gambaran mengenai fenomena yang sesungguhnya terjadi dan menggunakan pendekatan kualitatif. Dalam pendekatan kualitatif perlu menekankan pada pentingnya kedekatan dengan orang – orang dan situasi penelitian, agar peneliti memperoleh pemahaman jelas tentang realitas dan kondisi kehidupan nyata.

1. Ciri-Ciri Penelitian Eksperimen Kualitatif:

- a. Kontekstual: Penelitian dilakukan dalam konteks alami yang tidak sepenuhnya terkontrol. Misalnya, intervensi dapat dilakukan dalam lingkungan sosial atau budaya tertentu.
- b. Intervensi Terbatas: Meskipun ada intervensi, eksperimen lebih fleksibel dan dinamis, di mana peneliti berusaha memahami bagaimana intervensi mempengaruhi persepsi, pengalaman, atau perilaku partisipan dalam suatu konteks sosial.

- c. Pengumpulan Data Kualitatif: Data yang dikumpulkan umumnya berupa observasi, wawancara mendalam, dokumen, atau catatan lapangan. Fokusnya adalah pada cerita, pengalaman, dan interpretasi partisipan.
- d. Fokus pada Makna: Peneliti lebih tertarik pada makna di balik tindakan dan interaksi, serta bagaimana intervensi mempengaruhi makna yang diberikan oleh partisipan pada pengalaman mereka.
- e. Analisis Kualitatif: Analisis data dilakukan menggunakan pendekatan tematik, naratif, atau fenomenologis, untuk menggali pola dan pemahaman yang lebih mendalam tentang dampak intervensi.

2. Tahapan dalam Penelitian Eksperimen Kualitatif:

- a. Desain Eksperimen: Peneliti merancang eksperimen yang melibatkan manipulasi atau intervensi terhadap suatu kelompok, sambil tetap berfokus pada konteks sosial di mana penelitian dilakukan. Misalnya, peneliti mungkin mengubah cara pengajaran di dalam kelas untuk melihat bagaimana siswa merespons secara kualitatif.
- b. Pengumpulan Data: Data kualitatif diperoleh melalui berbagai metode seperti wawancara mendalam, observasi partisipatif, catatan lapangan, atau analisis dokumen. Proses ini

berlangsung selama atau setelah intervensi dilakukan untuk memahami dampaknya pada partisipan.

- c. Analisis Data: Peneliti mengidentifikasi tema-tema atau pola yang muncul dari data. Mereka mencari makna di balik perubahan perilaku, pengalaman partisipan, atau bagaimana mereka merespon intervensi.
- d. Interpretasi: Hasil penelitian diinterpretasikan dengan cara yang mendalam, menekankan pada konteks sosial dan makna yang diberikan oleh partisipan terhadap pengalaman mereka selama eksperimen.

Kesimpulan dari Penelitian eksperimen kualitatif adalah metode yang digunakan untuk mengeksplorasi efek suatu intervensi dalam konteks sosial dengan pendekatan yang fokus pada pemahaman mendalam. Meskipun lebih fleksibel dan kontekstual dibandingkan eksperimen kuantitatif, pendekatan ini memberikan pemahaman yang lebih kaya tentang bagaimana suatu perubahan atau intervensi mempengaruhi persepsi, pengalaman, dan interaksi manusia.

C. Metode Pengumpulan Data

Teknik pengumpulan data dapat didefinisikan sebagai suatu cara yang digunakan untuk memperoleh data yang dibutuhkan sebagai bahan masukan bagi penulis dalam penyusunan skripsi ini.

Proses dimulai dengan mengumpulkan data-data yang diperlukan dalam penelitian, adapun metode pengumpulan data dalam penelitian dilakukan melalui Penelitian Kepustakaan (library research). Teknik ini digunakan dengan

mempelajari literature berupa buku, jurnal, dokuemen, atau arsip di buku-buku pedoman, buku-buku perpustakaan yang dianggap dapat mendukung pengumpulan data.

D. Analisis Kebutuhan Sistem

Dalam melakukan penelitian, maka diperlukan alat dan bahan penelitian yang mendukung dalam kegiatan penelitian tersebut. Alat dan bahan yang diperlukan antara lain:

1. Alat Yang Digunakan

Alat penelitian yang digunakan selama proses penelitian antara lain sebagai berikut:

a. Hardware (Perangkat Keras)

Jenis	Spesifikasi
Laptop	<i>Lenovo Ideapad slim 3</i>
Processor	<i>i3 Gen 11</i>
RAM	<i>8 GB</i>
Display	<i>14" FDH</i>
Drive	<i>512 GB</i>

b. Software (Perangkat Lunak)

Jenis	Spesifikasi
Sistem Operasi	<i>Windows 10 Pro</i>
Desain	<i>Photoshop C S3</i>
Aplikasi	<i>Matlab R2024a</i>
Bahasa Pemrograman	<i>Matrix Laboratory</i>

E. Tahapan Penelitian

Tahap – tahap metode SDLC model waterfall yang digunakan peneliti untuk pengembangan aplikasi yaitu:

1. Requirement Analysis

Metode pengumpulan informasi ini dapat diperoleh dengan berbagai macam cara diantaranya, diskusi, observasi, survei, wawancara, dan sebagainya. Informasi yang diperoleh kemudian diolah dan dianalisa sehingga didapatkan data atau informasi yang lengkap mengenai spesifikasi kebutuhan pengguna akan perangkat lunak yang akan dikembangkan.

2. System and Software Design

Pada tahap ini akan membantu pengembang untuk menyiapkan kebutuhan hardware dalam pembuatan arsitektur sistem perangkat lunak yang akan dibuat secara keseluruhan.

3. Implementation and Unit Testing

Pada tahap ini dilakukan pengujian dan pemeriksaan terhadap fungsionalitas modul yang sudah dibuat, apakah sudah memenuhi kriteria yang diinginkan atau belum.

1. Integration and System Testing

Selanjutnya diintegrasikan dalam sistem secara keseluruhan. Setelah proses integrasi selesai, selanjutnya dilakukan pemeriksaan dan pengujian sistem secara keseluruhan untuk mengidentifikasi kemungkinan adanya kegagalan dan kesalahan sistem.

2. Operation and Maintenance

Perangkat lunak yang sudah jadi dioperasikan pengguna dan dilakukan pemeliharaan. Pemeliharaan memungkinkan pengembang untuk melakukan perbaikan atas kesalahan yang tidak terdeteksi pada tahap-tahap sebelumnya. Pemeliharaan meliputi perbaikan kesalahan, perbaikan implementasi unit sistem, dan peningkatan dan penyesuaian sistem sesuai dengan kebutuhan.

F. Jenis Data

Jenis data yang dikumpulkan dalam penelitian ini adalah data primer dan data sekunder yang relevan dengan masalah yang akan dibahas.

1. Data Primer

Data Primer adalah data yang berasal atau data yang diperoleh langsung dari sumber-sumber yang telah di wawancarai sebelumnya.

2. Data Sekunder

Data sekunder adalah data yang diperoleh tidak secara langsung dari objek penelitian seperti buku dan jurnal terkait dengan metode yang digunakan.

G. Kerangka Pikir

Pengolahan citra digital menjadi semakin penting seiring dengan perkembangan teknologi kamera dan kebutuhan untuk menghasilkan gambar berkualitas tinggi di berbagai bidang.



Seperti medis, keamanan, dan hiburan. Kualitas gambar sangat dipengaruhi oleh kondisi pencahayaan, resolusi, dan gangguan (noise) yang mungkin terjadi selama pengambilan gambar.



Salah satu metode yang digunakan untuk meningkatkan kualitas gambar adalah operasi titik, yaitu teknik yang menerapkan fungsi matematika pada setiap piksel gambar tanpa memperhitungkan piksel-piksel di sekitarnya.

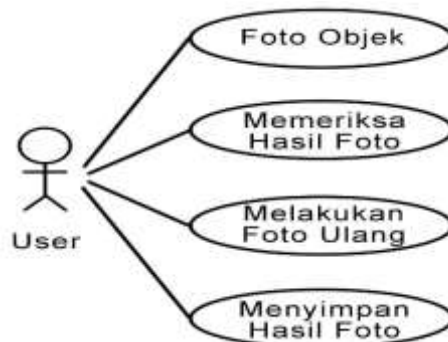


operasi titik sangat efektif dalam memperbaiki kontras, kecerahan, dan menghilangkan noise pada gambar.

BAB IV

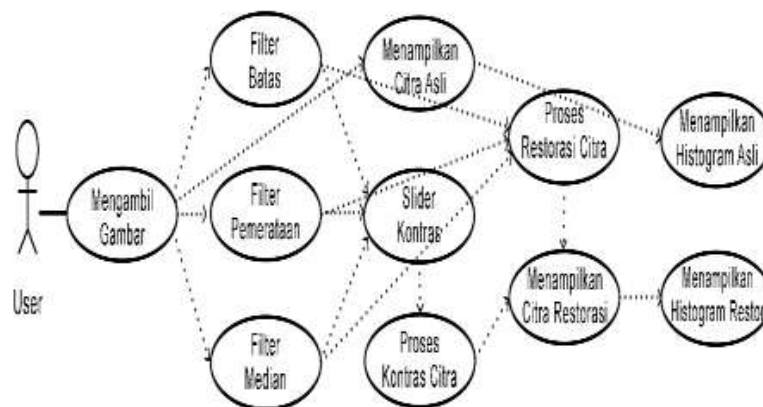
HASIL PENELITIAN DAN PEMBAHASAN

A. Sistem Yang Berjalan



Gambar 4. 1 Sistem yang berjalan

Gambar diatas menjelaskan bagaimana tentang proses *user* melakukan proses mendapatkan foto yang diinginkan. Dimana *user* akan memfoto sebuah objek. Selanjutnya *user* akan memeriksa hasil foto yang telah dipotret apakah sudah sesuai yang diinginkan atau belum. Jika tidak selanjutnya *user* akan melakukan foto ulang objek yang sama. Setelah dirasa sudah sesuai selanjutnya *user* akan menyimpan foto tersebut.



Gambar 4. 2 Sistem yang diusulkan

B. Analisis Sistem Yang Diusulkan

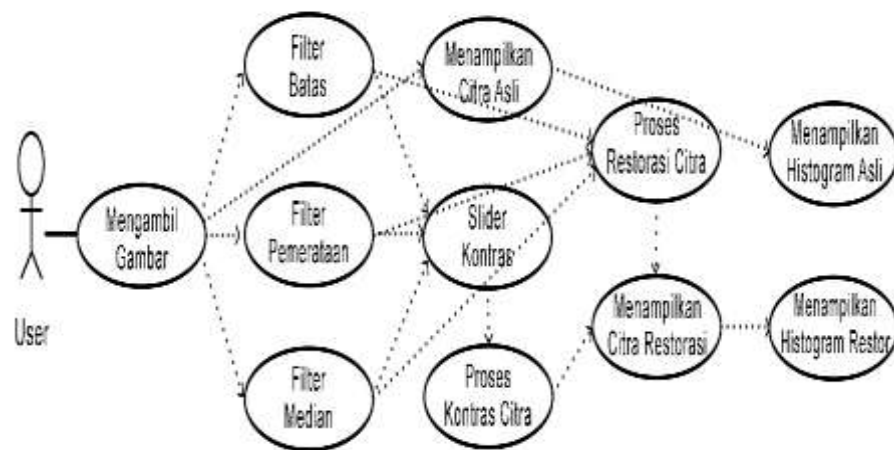
Gambar diatas menjelaskan bagaimana penulis membuat sistem yang dapat memperbaiki kualitas citra gambar yang memiliki derau bitnik. Dimana *user* dapat dengan mudah mengoperasikan aplikasi ini. Pertama mengambil gambar yang ingin di perbaiki. Selanjutnya *user* dapat memilih opsi Filter Batas, Filter Pemerataan, dan Filter median untuk mode perbaikan citra gambar. Kemudian akan ditampilkan Citra asli dan memunculkan *Histogram* Citra Asli tersebut. Selanjutnya Citra Asli akan melewati Proses Restorasi Citra sesuai mode perbaikan yang dipilih. Kemudian akan ditampilkan Citra yang telah direstorasi dan memunculkan *Histogram* Citra yang telah direstorasi. *User* juga dapat menaikkan dan menurunkan kontras pada Citra yang telah direstorasi dengan cara menggeser menu Slider, arah ke kiri untuk menurunkan kontras dan ke kanan untuk menaikkan kontras.

C. Analisis Aliran Data Dengan UML

Analisis aliran data yang digunakan penulis dalam menyusun sistem ini dengan menggunakan orientasi objek *Use case Diagram*, *Activity Diagram*, dan *Sequence Diagram* yang bertujuan untuk mengetahui aliran proses informasi.

1. Use Case Diagram

Use Case Diagram berfungsi untuk menjalankan manfaat sistem jika dilihat menurut pandangan orang yang berada diluar sistem (*actor*).



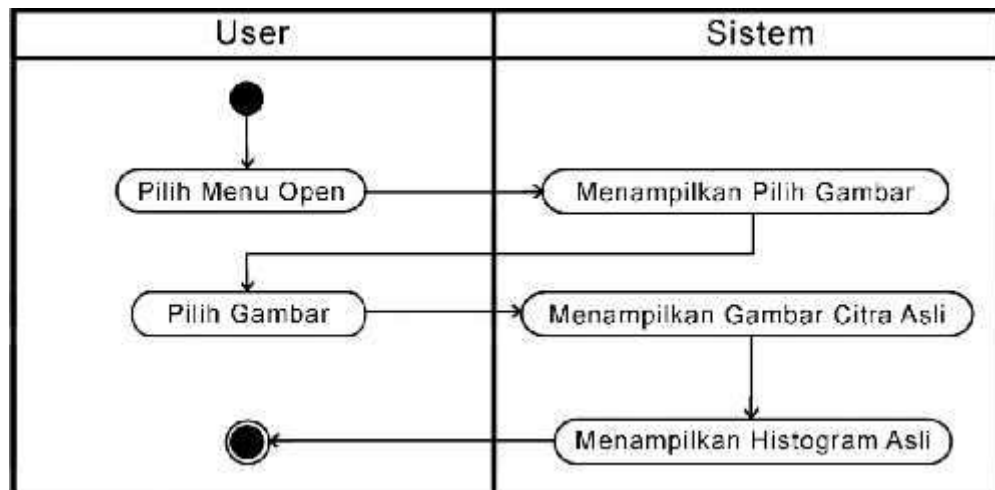
Gambar 4. 3 Use case diagram

Tabel 4. 1 *Penjelasan use case diagram user*

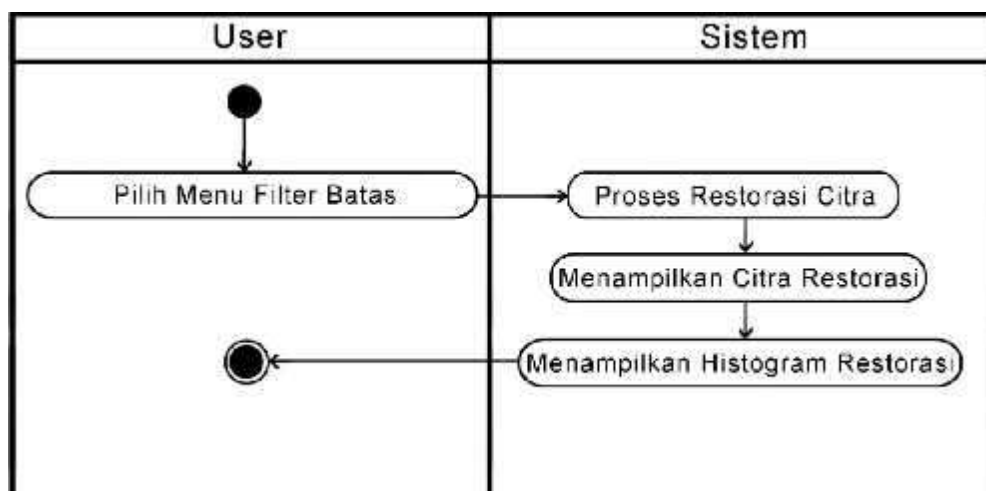
Nama Use Case	Deskripsi Use Case
Mengambil Gambar	<i>Use Case</i> ini menjelaskan tentang <i>user</i> memilih gambar yang ingin di restorasi.
Menampilkan Citra Asli	<i>Use Case</i> ini menjelaskan tentang menampilkan citra asli yang ingin di restorasi.
Menampilkan Histogram Asli	<i>Use Case</i> ini menjelaskan tentang menampilkan Histogram/grafik pada citra asli yang ingin di restorasi.
Filter Batas	<i>Use Case</i> ini menjelaskan tentang <i>user</i> memilih menu Filter Batas untuk mencegah piksel intensitasnya di luar intensitas piksel tetangga pada citra asli.
Filter Pemerataan	<i>Use Case</i> ini menjelaskan tentang <i>user</i> memilih menu Filter Pemerataan untuk menghilangkan derau bintik pada citra asli.
Filter Median	<i>Use Case</i> ini menjelaskan tentang <i>user</i> memilih menu Filter Median untuk menghilangkan derau bintik pada delapan <i>pixel</i> tetangga pada citra asli.
Slider Kontras	<i>Use Case</i> ini menjelaskan tentang <i>user</i> memilih menu Slider Kontras untuk menurunkan atau menaikkan nilai kontras pada citra asli.
Proses Kontras Citra	<i>Use Case</i> ini menjelaskan tentang sistem melakukan proses menurunkan atau menaikkan kontras sesuai arah yang di pilih oleh <i>user</i> .
Proses Restorasi Citra	<i>Use Case</i> ini menjelaskan tentang sistem melakukan proses restorasi citra asli sesuai menu yang telah di pilih.
Menampilkan Citra Restorasi	<i>Use Case</i> ini menjelaskan tentang menampilkan citra yang telah direstorasi.
Menampilkan Histogram Restorasi	<i>Use Case</i> ini menjelaskan tentang menampilkan Histogram/grafik pada citra yang telah direstorasi.

2. Activity Diagram

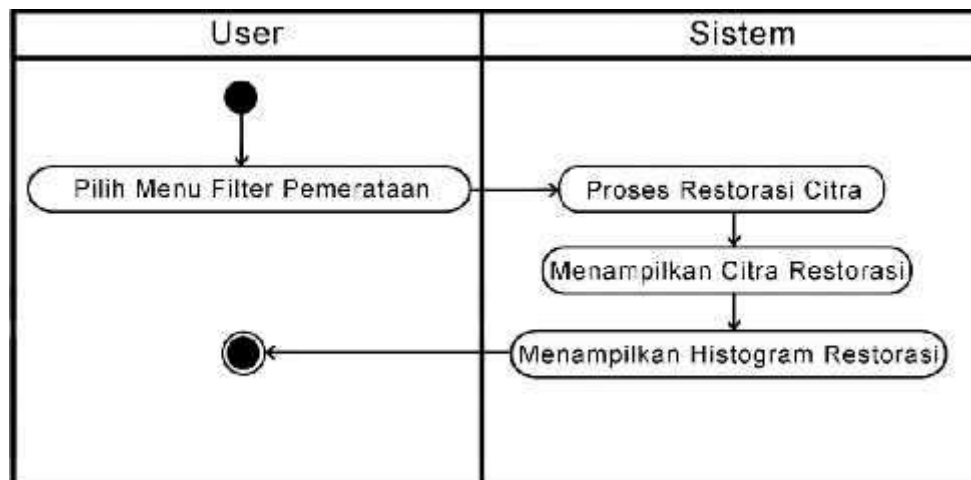
Activity diagram ini menjelaskan tentang aktifitas-aktifitas yang terjadi dalam sebuah aliran proses pada sebuah sistem.



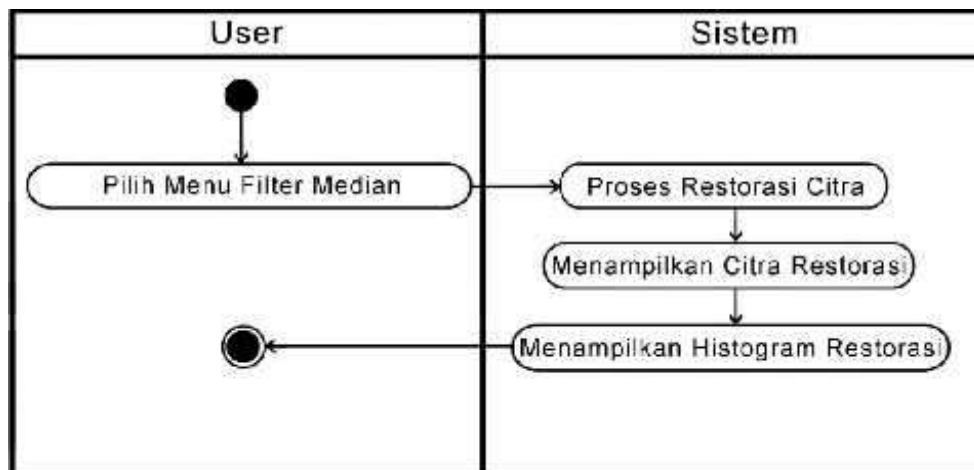
Gambar 4. 4 Activity diagram open.



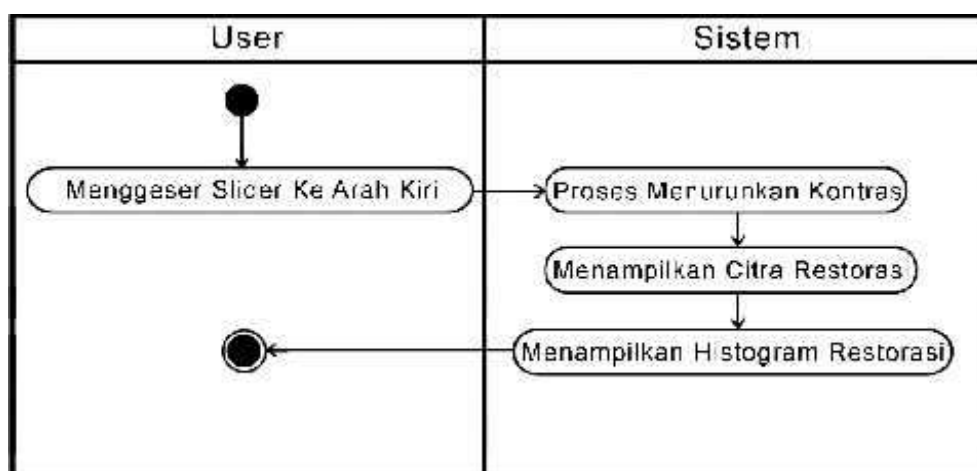
Gambar 4. 5 Activity diagram filter batas.



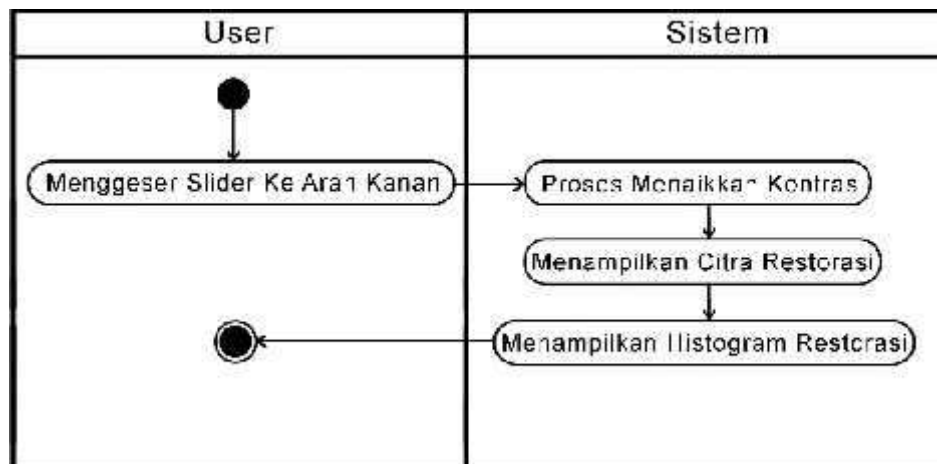
Gambar 4. 6 Activity diagram filter pemerataan.



Gambar 4. 7 Activity diagram filter median.



Gambar 4. 8 Activity diagram slider menurunkan kontras.

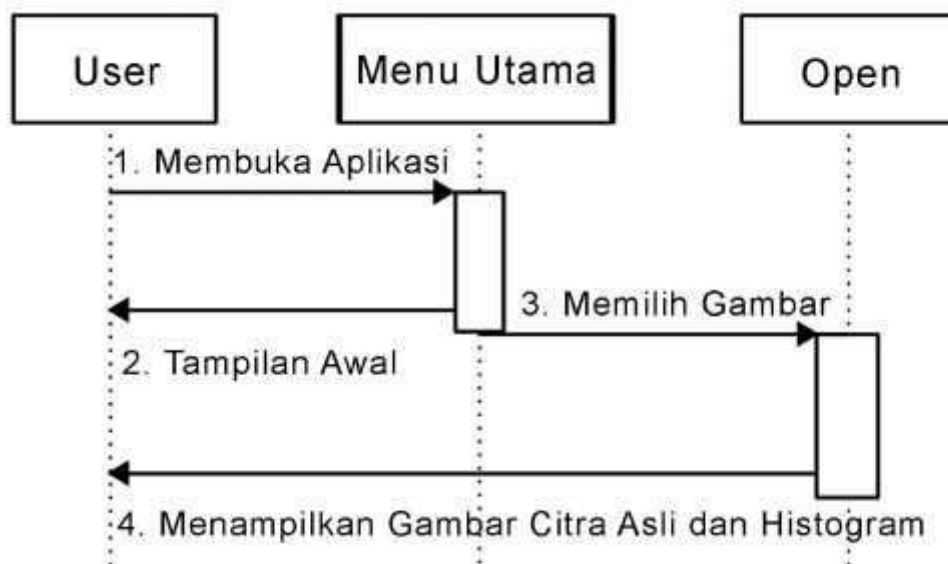


Gambar 4. 9 Activity diagram slider menaikkan kontras.

3. Sequence diagram

Diagram Sequence merupakan salah satu diagram *Interaction* yang menjelaskan bagaimana suatu operasi itu dilakukan ; *message* (pesan) apa yang dikirim dan kapan pelaksanaannya.

a. Sequence Diagram User Open



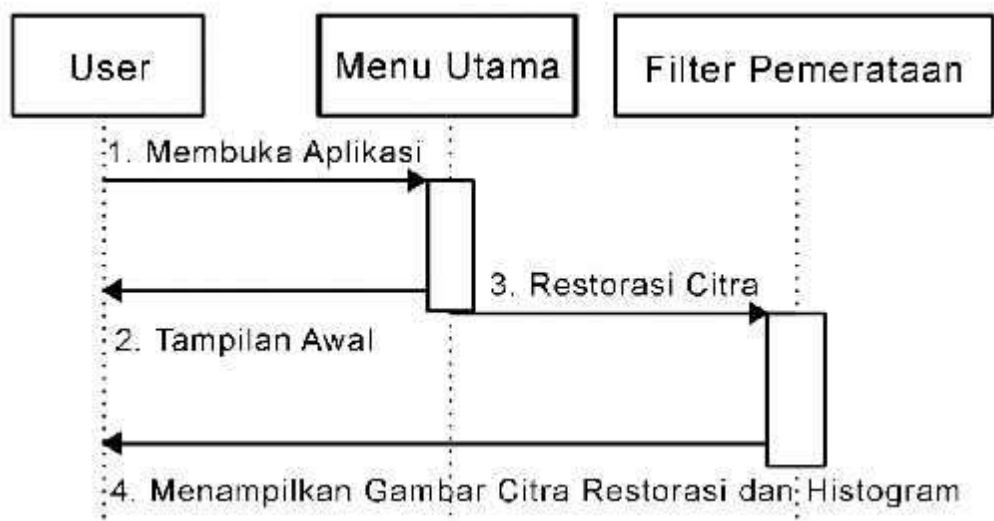
Gambar 4. 10 Sequence diagram user open.

b. Sequence Diagram filter batas



Gambar 4. 11 Sequence diagram filter batas.

c. Sequence Diagram filter pemerataan



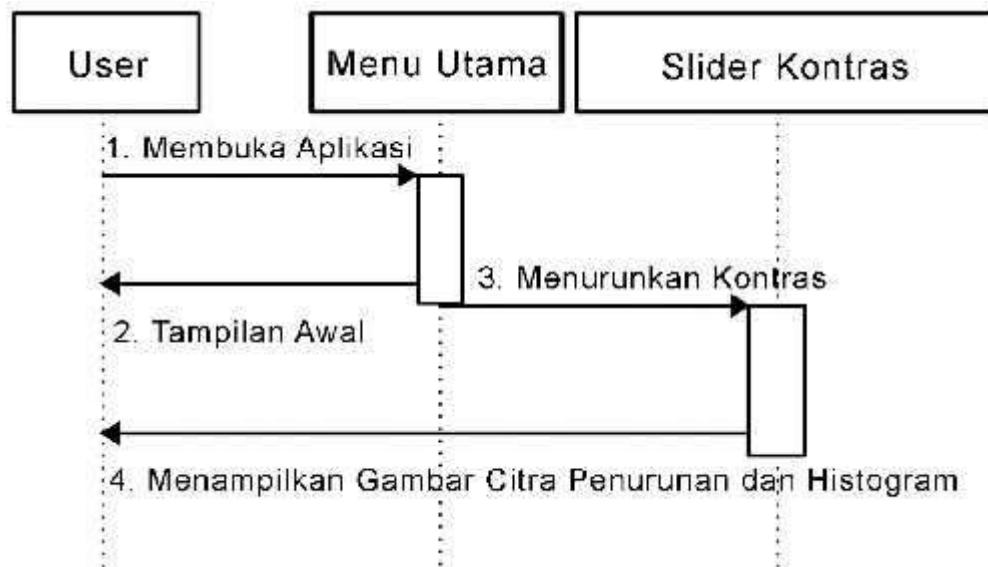
Gambar 4. 12 Sequence diagram filter pemerataan.

d. Sequence Diagram filter median



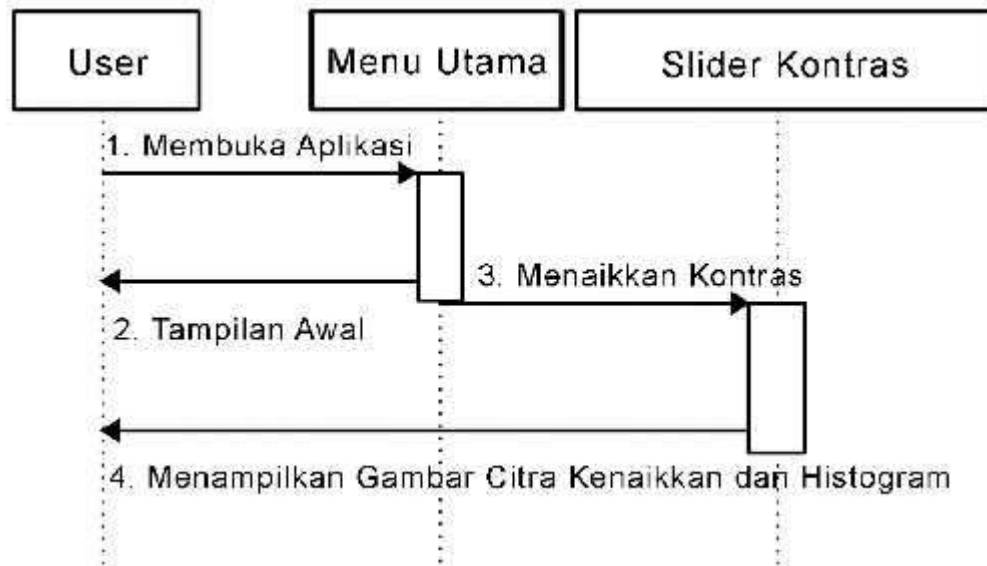
Gambar 4. 13 Sequence diagram filter median.

e. Sequence Diagram Slider Menurunkan



Gambar 4. 14 Sequence diagram slider menurunkan kontras.

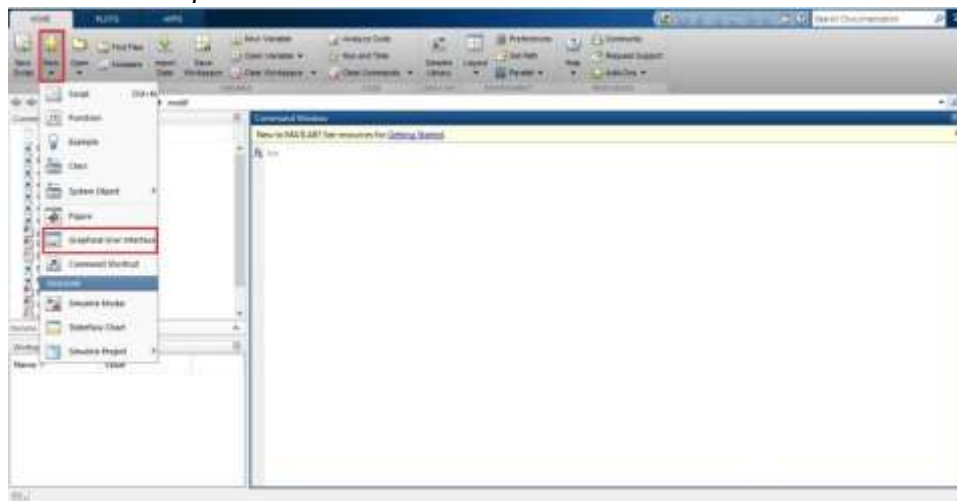
f. Sequence Diagram Slider Menaikkan



Gambar 4. 15 Sequence diagram slider menaikkan.

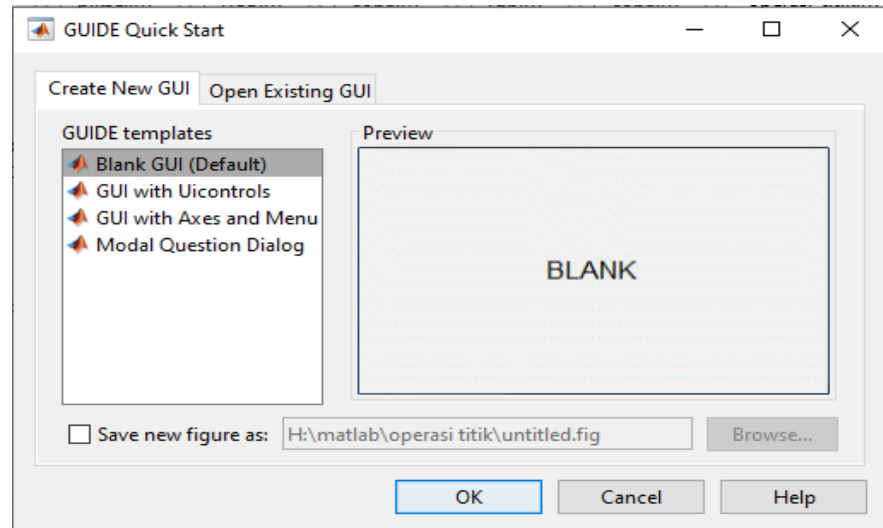
D. Pembuatan Aplikasi

1. Buka aplikasi *MATLAB*, kemudian pilih *New > Graphical User Interface*.



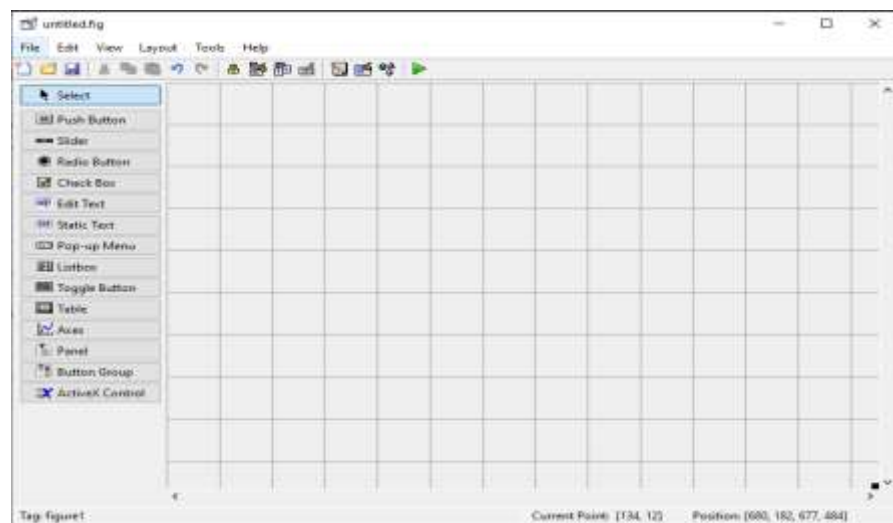
Gambar 4. 16 Tampilan awal MATLAB

2. Selanjutnya pilih *Blank GUI (Default)* kemudian *OK* untuk melanjutkan pembuatan *GUI* atau *Graphical User Interface*.



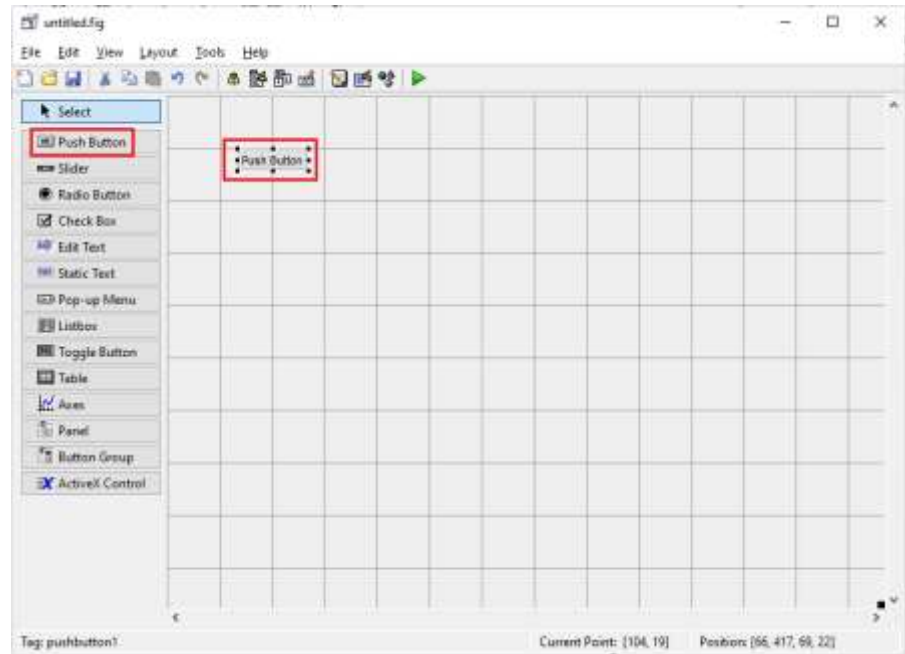
Gambar 4. 17 Tampilan *GUIDE Quick Start*

3. Selanjutnya akan tampil awal *layer GUI*, dimana *layer* ini berfungsi sebagai tempat mendesain tampilan aplikasi.



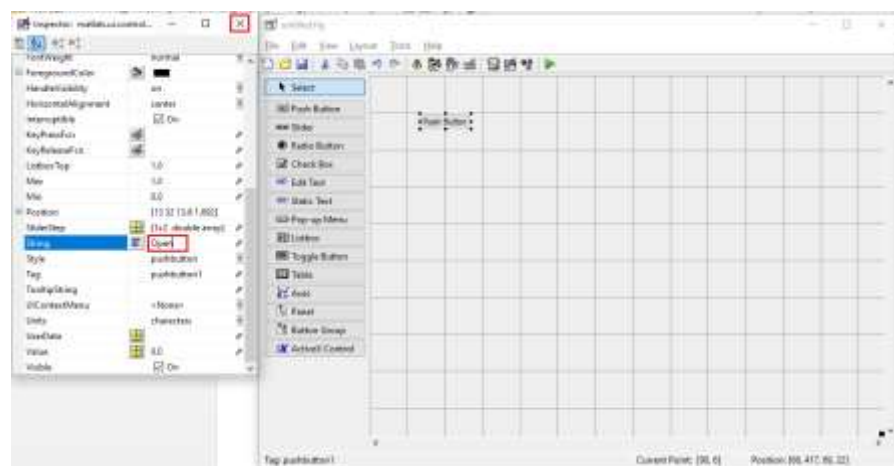
Gambar 4. 18 Tampilan *layer GUI*

4. Selanjutnya pilih *Push Button* kemudian letakkan di layer sesuka hati kalian.



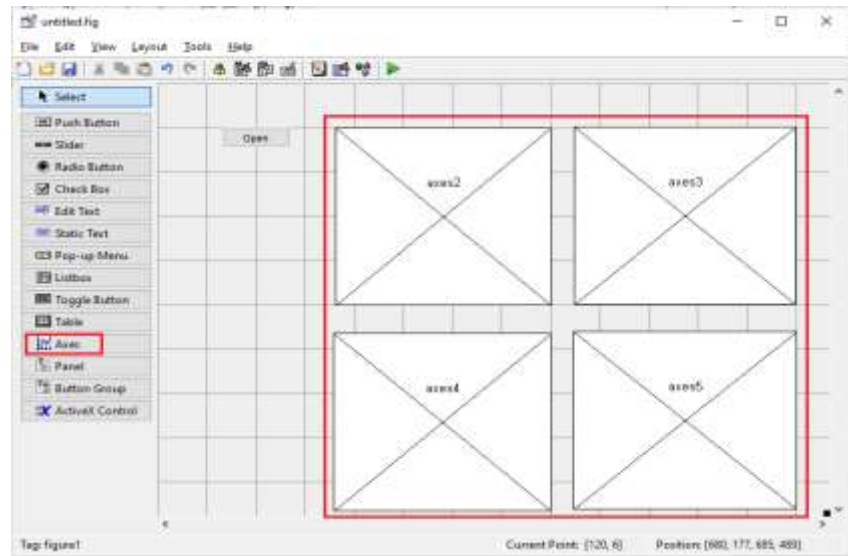
Gambar 4. 19 Membuat push button

5. Selanjutnya klik dua kali pada *Push Button* yang telah kita buat tadi untuk memunculkan halaman, dimana halaman ini kita akan mengubah nama *Push Button* menjadi *Open* pada bagian *String*, kemudian *Close* jika selesai.



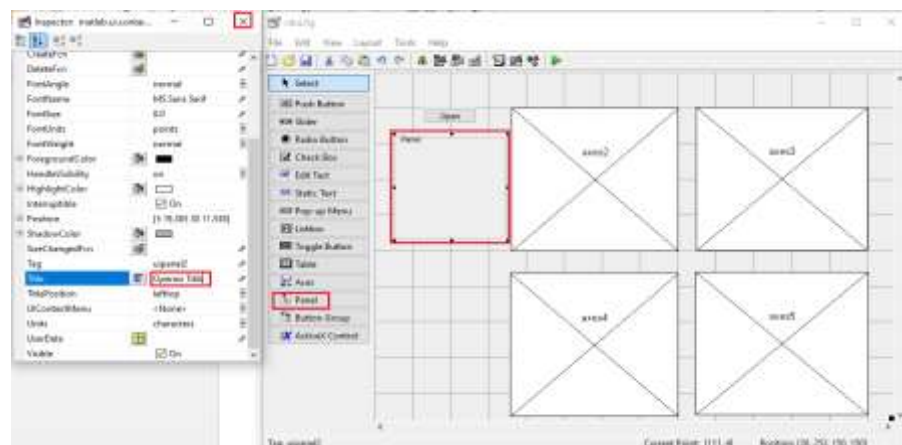
Gambar 4. 20 Mengubah nama push button

6. Selanjutnya kita pilih *Axes* kemudian posisi kan pada *layer*, atur tinggi dan lebar *Axes* sesuka kalian dan buat empat untuk menampilkan citra asli, *histogram* asli, citra restorasi dan *histogram* citra yang telah direstorasi.



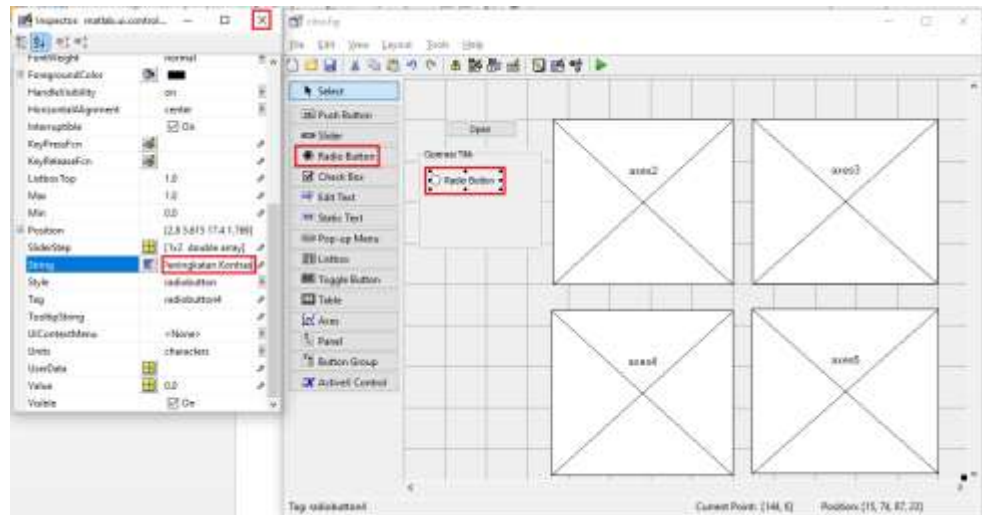
Gambar 4. 21 Membuat axes.

7. Selanjutnya kita pilih *Panel* kemudian posisikan di *layer*, kemudain atur tinggi dan lebarnya. Klik dua kali pada *Panel* yang telah dibuat, kemudain ubah namanya, selanjutnya *Close* jika selesai.



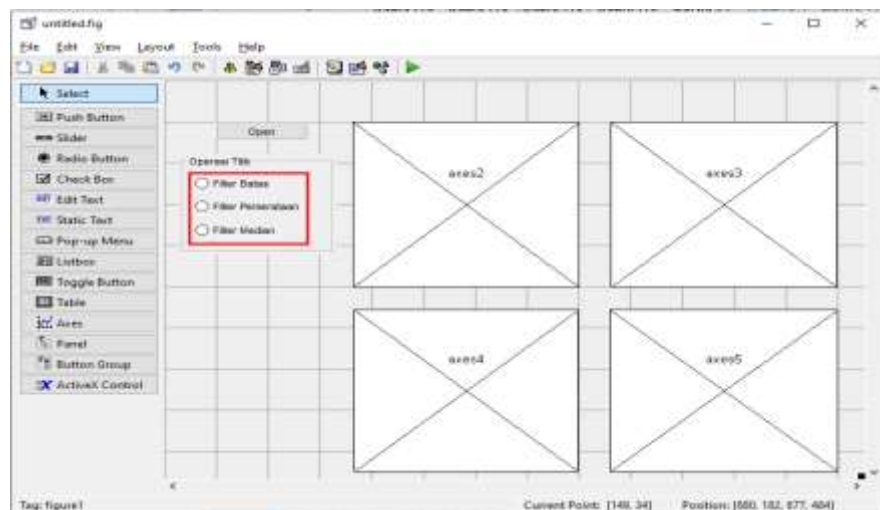
Gambar 4. 22 Membuat panel dan mengubah namanya

8. Selanjutnya kita pilih *Radio Button* kemudain posisikan di *Panel* yang sudah dibuat sebelumnya. Klik dua kali pada *Radio Button* yang telah dibuat, kemudian ubah namanya, selanjutnya *Close* jika selesai.



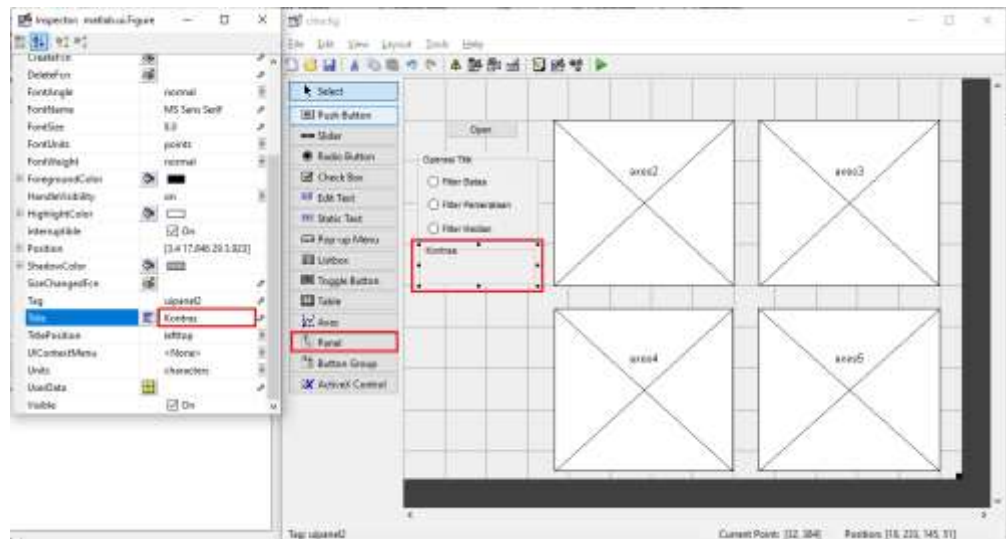
Gambar 4. 23 Membuat radio button dan mengubah namanya.

9. Selanjutnya kita buat tiga *Radio Button* yang diberi nama Filter Batas, Filter Pemerataan, dan Filter Median.



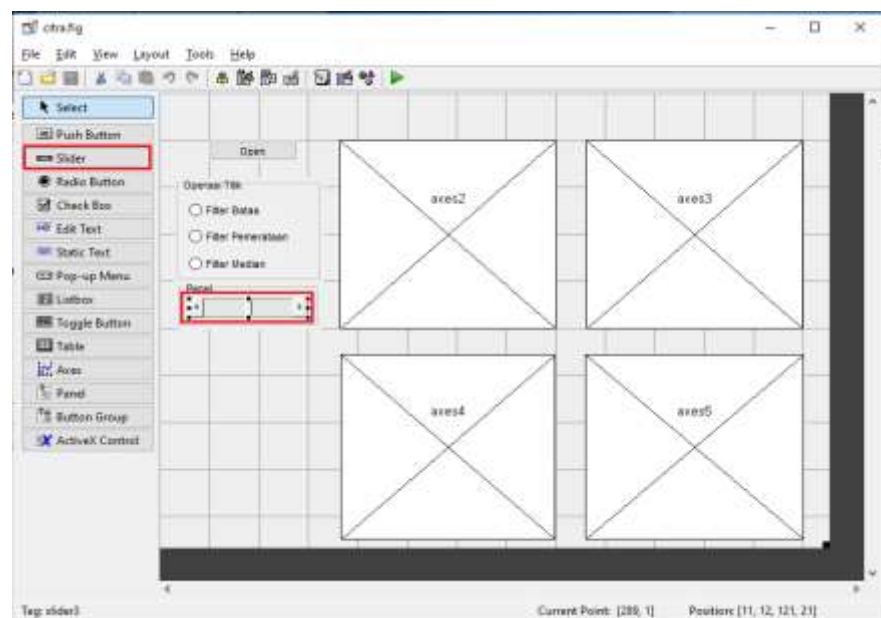
Gambar 4. 24 Membuat radio button masing – masing operasi.

10. Selanjutnya kita buat tiga *Radio Button* yang diberi nama Filter Batas, Filter Pemerataan, dan Filter Median.



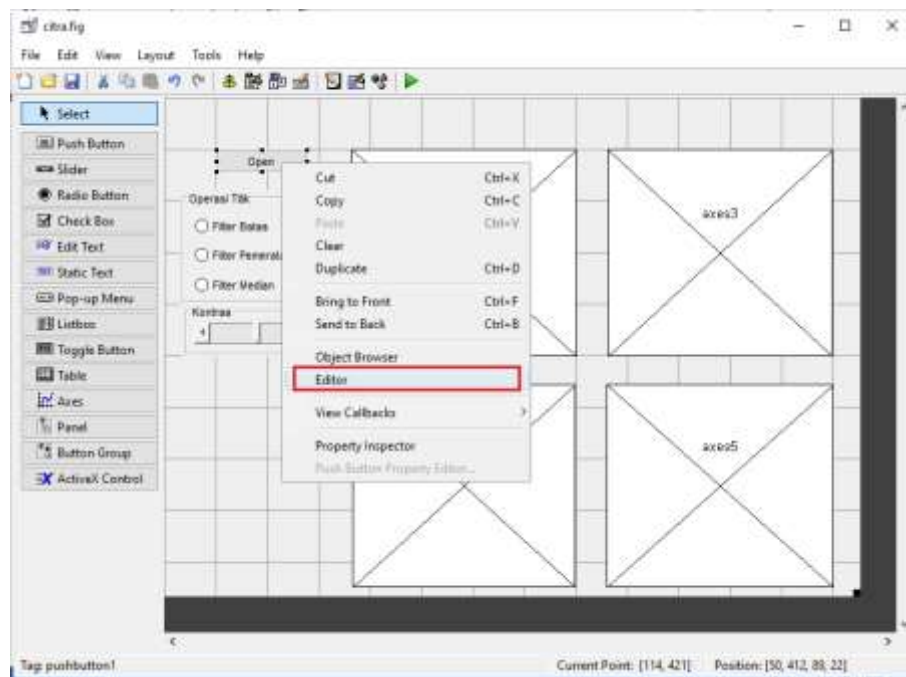
Gambar 4. 25 Membuat panel dan mengubah namanya.

11. Selanjutnya kita buat tiga *Radio Button* yang diberi nama Filter Batas, Filter Pemerataan, dan Filter Median.



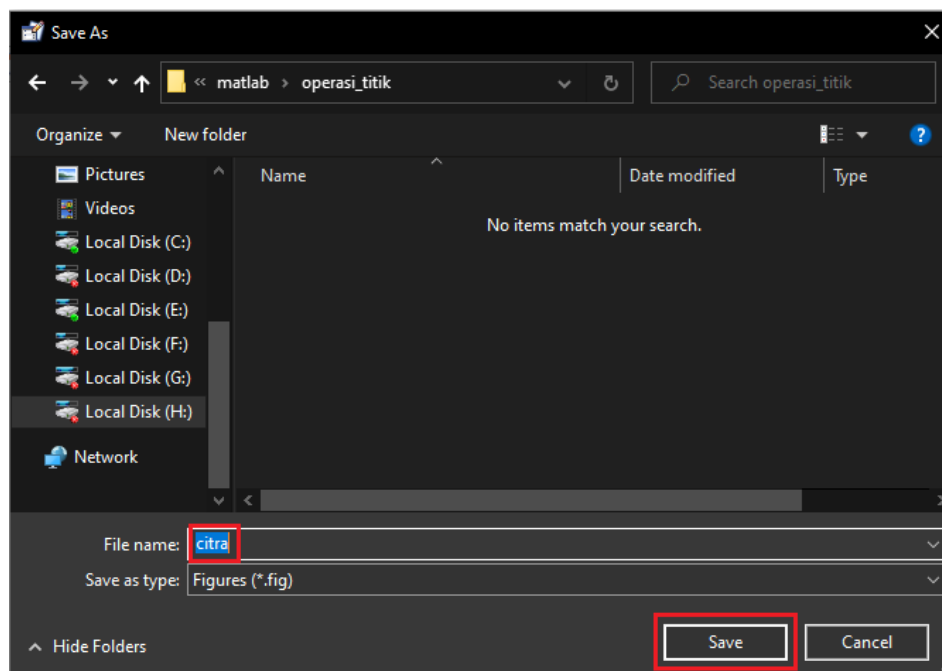
Gambar 4. 26 Membuat slider.

12. Selanjutnya klik kanan pada *Open*, kemudian pilih *Editor*.



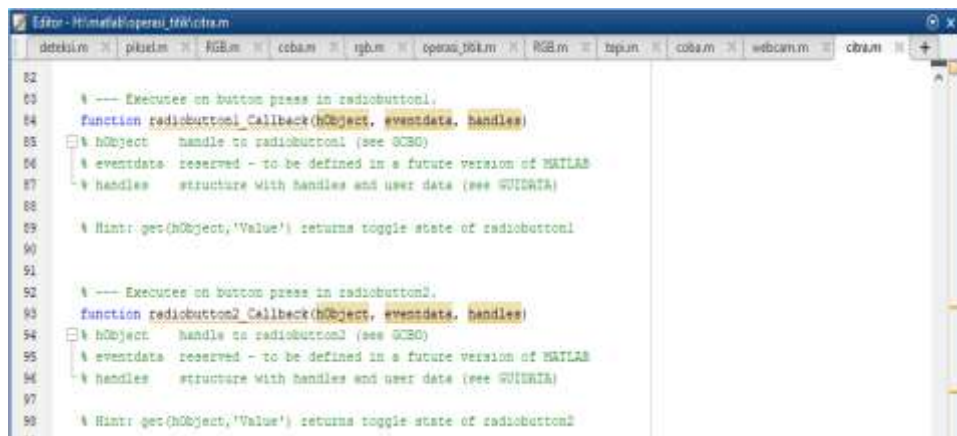
Gambar 4. 27 Memilih editor.

13. Selanjutnya berikan nama kemudian pilih *Save*.



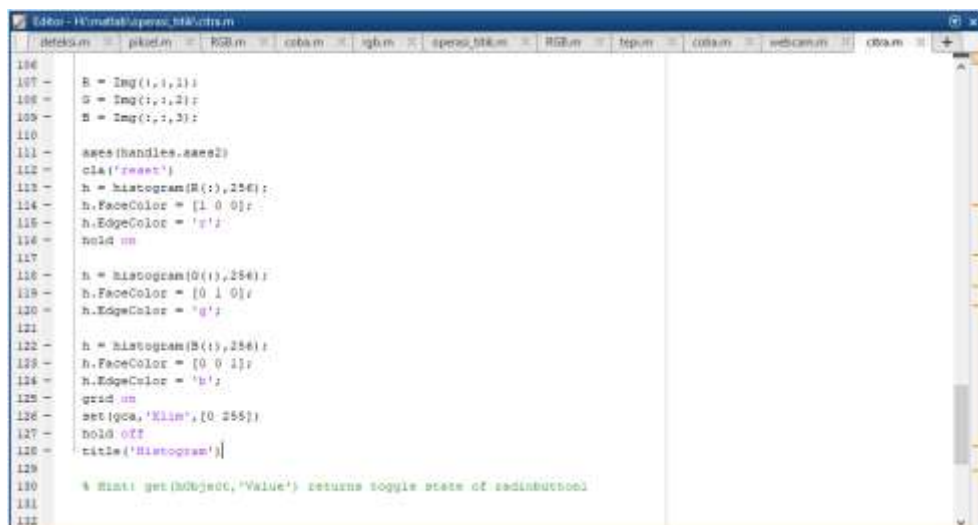
Gambar 4. 28 Menyimpan GUI.

14. Selanjutnya akan menampilkan *Script* seperti dibawah ini.



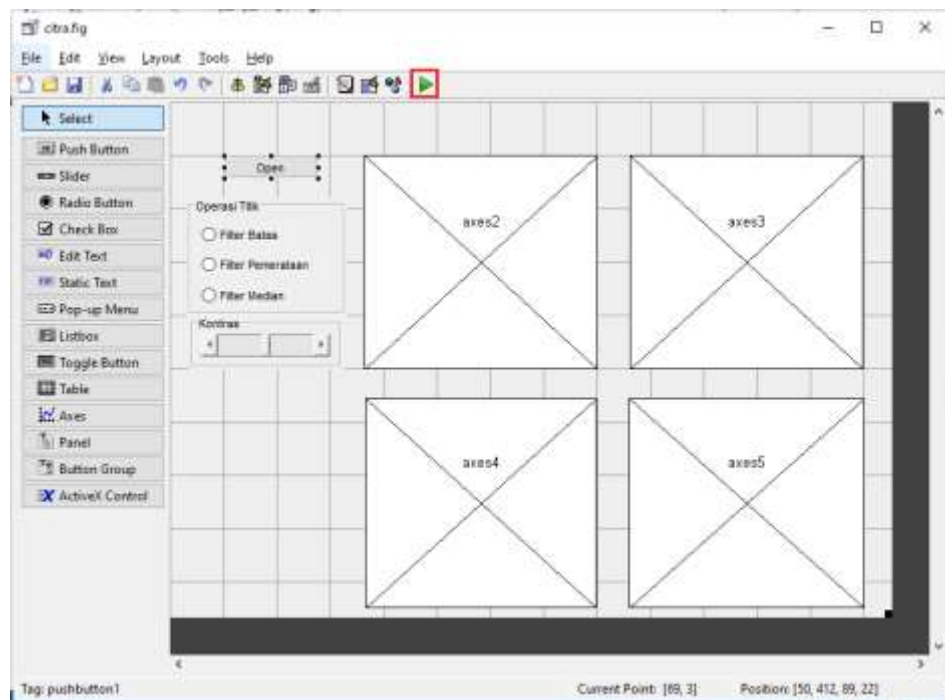
Gambar 4. 29 Tampilan awal script.

15. Selanjutnya kita tuliskan *Script* pada masing – masing *Button* dan *Radio Button* yang telah kita buat sebelumnya.



Gambar 4. 30 Tampilan script yang sudah dibuat.

16. Setelah selesai menuliskan semua *Script*, selanjutnya kita kembali pada tampilan *GUI*, kemudian kita pilih *Run* atau simbol *Play* untuk menjalankan aplikasi.



Gambar 4. 31 Menjalankan aplikasi.

E. Implementasi

Adapun kebutuhan yang digunakan untuk membangun sebuah sistem dengan judul **“Operasi Titik Pada Pengolahan Citra Digital Untuk Meningkatkan Kualitas Gambar.** Yaitu sebagai berikut:

1. Kebutuhan *Hardware*

Spesifikasi *hardware* pengguna aplikasi ini adalah sebagai berikut:

Tabel 4. 2 Spesifikasi *hardware* komputer

Jenis	Spesifikasi
Laptop	<i>Lenovo Ideapad Slim 3</i>
<i>Processor</i>	<i>i3 Gen 11</i>
<i>RAM</i>	8 GB
<i>Drive</i>	HDD 512 GB
<i>Display</i>	14" <i>FDH</i>

2. Kebutuhan *Software*

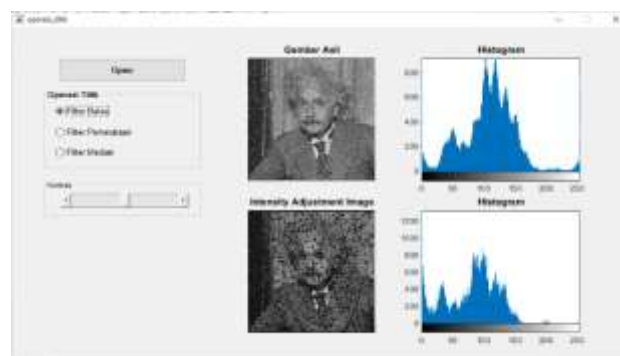
Spesifikasi *software* pengguna aplikasi ini sebagai berikut:

Tabel 4. 3 Spesifikasi *software* komputer

Jenis	Spesifikasi
Sistem Operasi	<i>Windows 10 Pro</i>
Desain	<i>Photoshop CS 3</i>
Aplikasi	<i>Matlab R2024a</i>
Bahasa Pemrograman	<i>Matrix Laboratory</i>

F. Pengujian *Sample*

1. Pada gambar dibawah ini menampilkan citra asli, histogram asli, citra filter batas, dan histogram peningkatan kontras.



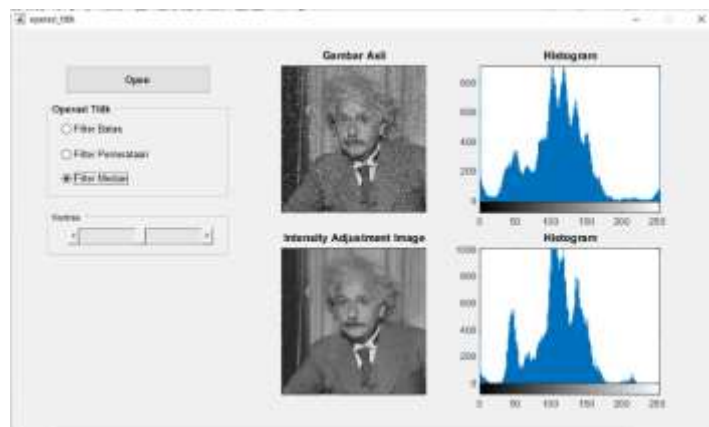
Gambar 4. 32 Tampilan filter batas

2. Pada gambar dibawah ini menampilkan citra asli, histogram asli, citra filter pemerataan, dan histogram penurunan kontras.



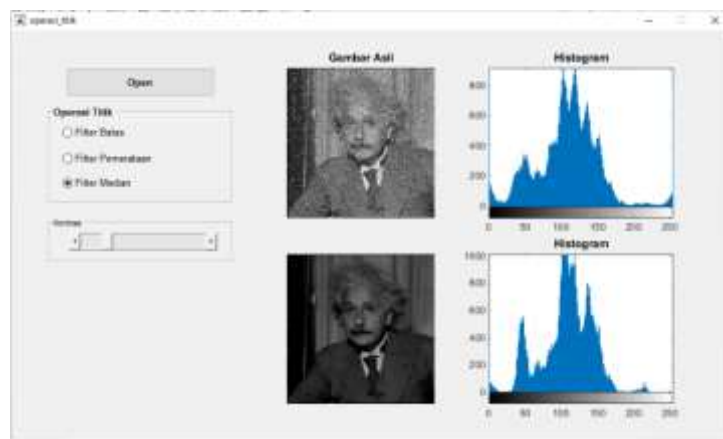
Gambar 4. 33 *Tampilan filter pemerataan*

3. Pada gambar dibawah ini menampilkan citra asli, histogram asli, citra filter median, dan histogram ekualisasi histogram.



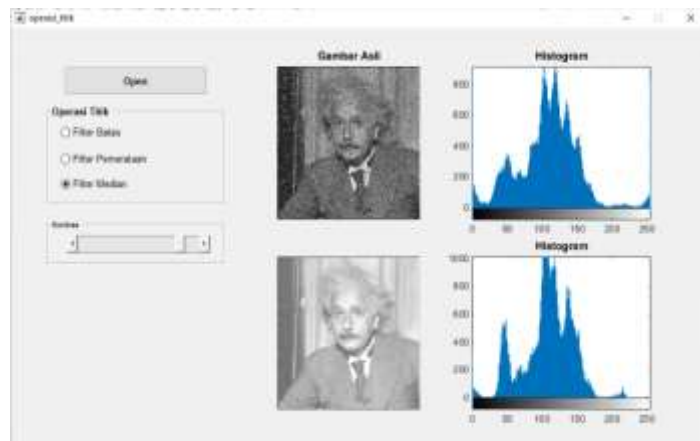
Gambar 4. 34 *Tampilan filter median*

4. Pada gambar dibawah ini menampilkan citra asli, histogram asli, citra filter median, penurunan kontras, dan histogram ekualisasi histogram.



Gambar 4. 35 *Tampilan menurunkan kontras*

5. Pada gambar dibawah ini menampilkan citra asli, histogram asli, citra filter median, kenaikan kontras dan histogram ekualisasi histogram.



Gambar 4. 36 *Tampilan menaikkan kontras*

6. Pada gambar dibawah ini menampilkan citra negatif, histogram, perbaikan citra dan histogram ekualisasi histogram.



Gambar 4. 37 Tampilan citra negatif

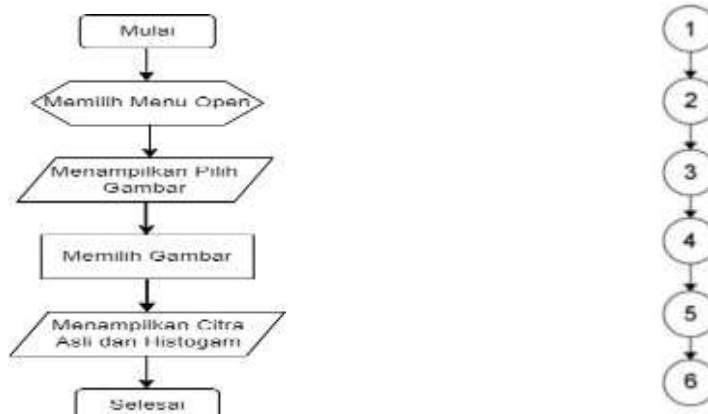
F. Pengujian Sistem

Pengujian aplikasi ini dilakukan dengan menggunakan dua metode pengujian yaitu Pengujian *Black Box* dan Pengujian *White Box*

1. Pengujian White Box

a. Pengujian Button Open

1) Flowchart button open dan Flowgraph button open



Gambar 4. 38 Flowchart button open dan Flowgraph button open

Proses perhitungan *button open*

Proses perhitungan dari gambar diatas dapat dilakukan sebagai berikut:

- 2) Menghitung *cyclomatic complexcity* $V(G)$ dari *Egde* dan *Node*:

Dengan Rumus :

$$V(G) = E - N + 2$$

$$N (node) = 6$$

$$E (edge) = 5$$

$$P (\text{predikat } node) = 0$$

$$\text{Penyelesaian : } V(G) = E - N + 2$$

$$= 5 - 6 + 2$$

$$= 1$$

$$\text{Predikat } Node (N) = P + 1$$

$$= 0 + 1$$

$$= 1$$

- 3) Berdasarkan perhitungan *cyclomatic complexcity* dari *Flowgraph* diatas memiliki *Region* = 1

- 4) *Independent path* pada *flowgraph*

didas yaitu: Path 1

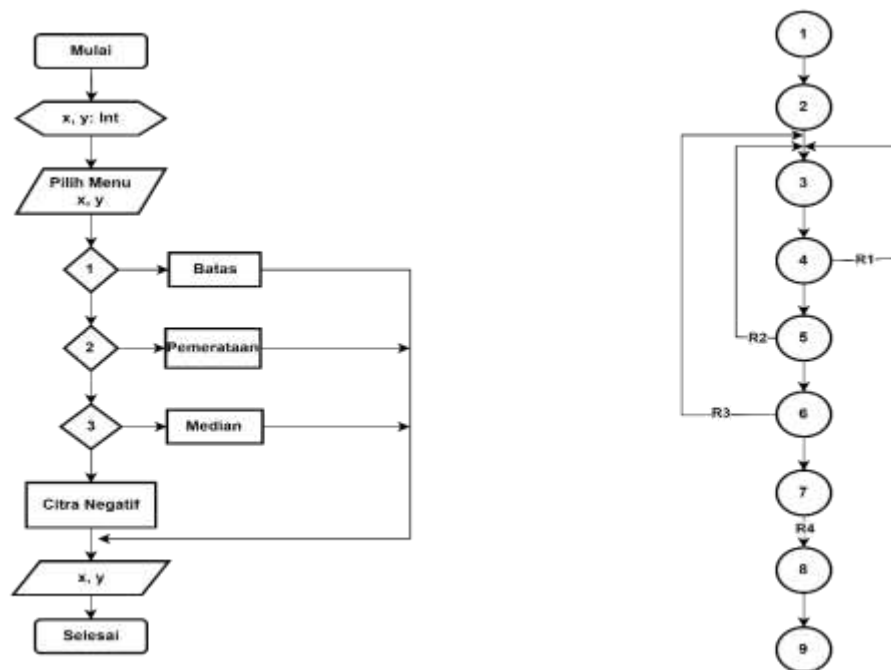
$$= 1 - 2 - 3 - 4 - 5 - 6$$

- 5) *Grafik matriks button open*

Tabel 4. 4 Grafik matriks button open

NO	1	2	3	4	5	6	E-1
1		1					$1-1=0$
2			1				$1-1=0$
3				1			$1-1=0$
4					1		$1-1=0$
5						1	$1-1=0$
SUM(E+1)							$0+1=1$

2) Flowchart dan Flowgraph radiobutton filter batas, pemerataan, median dan citra negatif

**Gambar 4. 39** Flowchart dan Flowgraph radiobutton

Proses perhitungan *radiobutton* filter batas, pemerataan, median dan citra negatif, proses perhitungan dari gambar diatas dapat dilakukan sebagai berikut

Menghitung *cyclomatic complexity* $V(G)$ dari *Edge* dan *Node*:

Dengan Rumus :

$$V(G) = E - N + 2$$

$$N(\text{node}) = 5$$

$$E(\text{edge}) = 4$$

$$P(\text{predikat node}) = 0$$

$$\text{Penyelesaian : } V(G) = E - N + 2$$

$$= 4 - 5 + 2$$

$$= 1$$

$$\text{Predikat Node (N)} = P + 1$$

$$= 0 + 1$$

$$= 1$$

6) Berdasarkan perhitungan *cyclomatic complexity* dari *Flowgraph*

diatas memiliki *Region* = 1

7) *Independent path* pada

flowgraph diatas yaitu:

Path 1

$$= 1 - 2 - 3 - 4 - 5$$

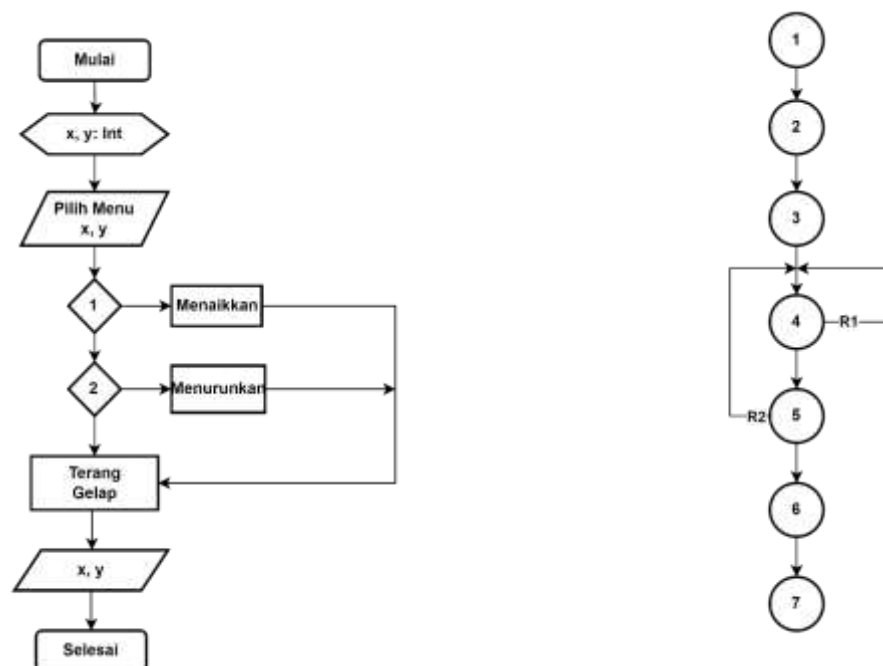
8) *Grafik matriks radiobutton* filter batas, pemerataan, median dan citra negatif

Tabel 4. 5 Grafik radiobutton filter batas, pemerataan, median dan citra negatif

NO	1	2	3	4	5	E-1
1		1				$1-1=0$
2			1			$1-1=0$
3				1		$1-1=0$
4					1	$1-1=0$
5						$0-1=0$
SUM(E+1)						$0+1=1$

b. Pengujian Slider Menaikkan dan Menurunkan Kontras

1) Flowchart dan Flowgraph slider menaikkan dan menurunkan kontras



Gambar 4. 40 Flowchart dan Flowgraph slider menaikkan dan menurunkan kontras

Proses perhitungan *slider* menaikkan
menurunkan kontras

Proses perhitungan dari gambar diatas dapat dilakukan sebagai
berikut :

2) Menghitung *cyclomatic complexity* $V(G)$ dari *Edge* dan

Node:

Dengan Rumus :

$$V(G) = E - N + 2$$

$$N(\text{node}) = 5$$

$$E(\text{edge}) = 4$$

$$P(\text{predikat node}) = 0$$

$$\text{Penyelesaian : } V(G) = E - N + 2$$

$$= 4 - 5 + 2$$

$$= 1$$

$$\text{Predikat Node (N)} = P + 1$$

$$= 0 + 1$$

$$= 1$$

3) Berdasarkan perhitungan *cyclomatic complexity* dari
Flowgraph

$$\text{diatas memiliki Region} = 1$$

4) *Independent path* pada

flowgraph diatas yaitu:

Path 1

$$= 1 - 2 - 3 - 4 - 5$$

5) Grafik matriks slider menaikkan

Tabel 4. 6 Grafik slider menaikkan

NO	1	2	3	4	5	E-1
1		1				1-1=0
2			1			1-1=0
3				1		1-1=0
4					1	1-1=0
5						0-1=0
SUM(E+1)						0+1=1

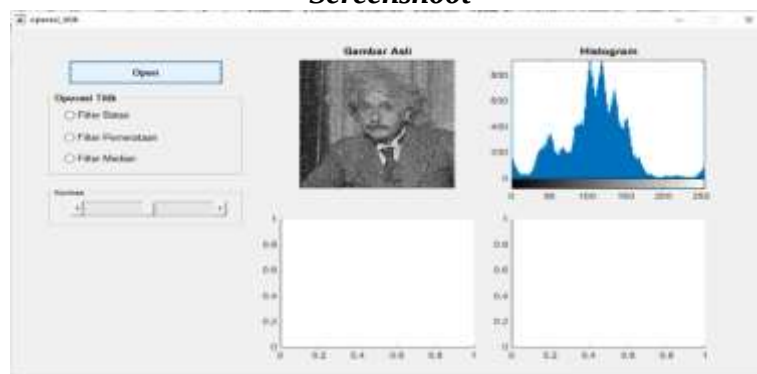
2. Pengujian Black Box

a. Black Box Button Open

Tabel 4. 7 Black box button open

Test Factor	Hasil	Kesimpulan
User memilih button open.	✓	Berhasil karena akan menampilkan citra asli pada axes2 dan menampilkan histogram citra asli.

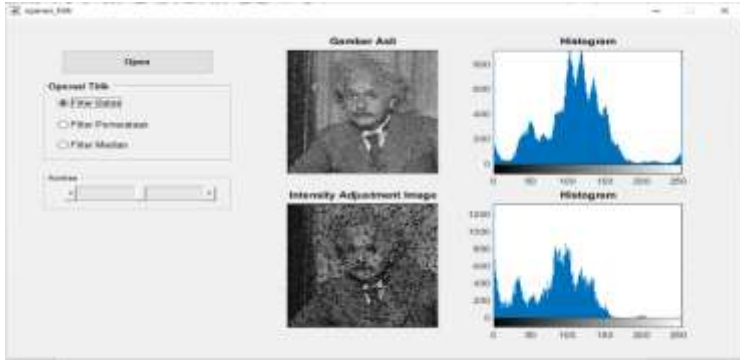
Screenshoot



Tampilan citra asli, dan histogram asli.

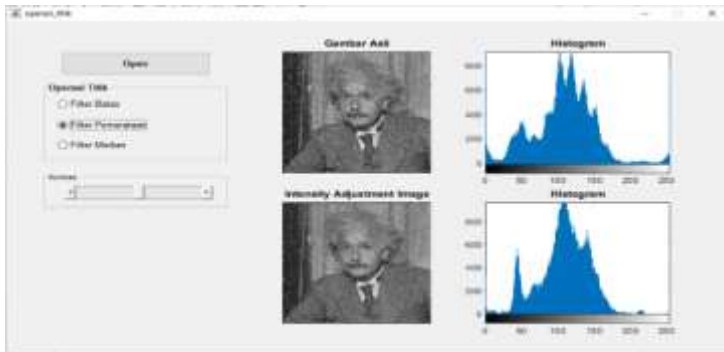
b. Black Box Radio button Filter Batas

Tabel 4. 8 *Black box radio button filter batas*

Test Factor	Hasil	Kesimpulan
User memilih radiobutton Filter Batas.	✓	Berhasil karena akan menampilkan restorasi citra pada axes4.
Screen Shot  Tampilan citra filter batas		

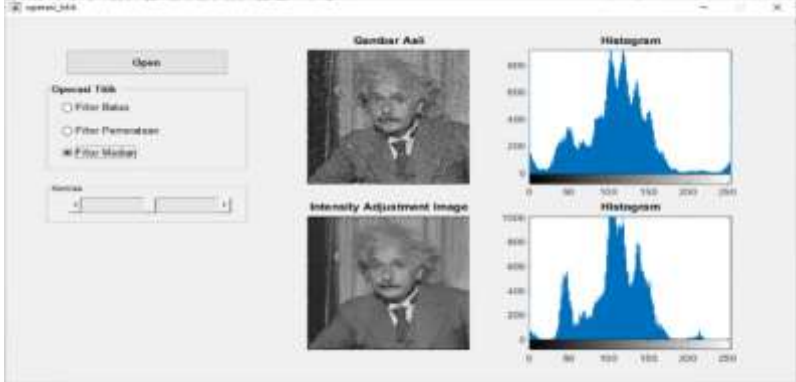
x Radiobutton Filter Pemerataan

Tabel 4. 9 *Black box radiobutton filter pemerataan*

Test Factor	Hasil	Kesimpulan
User memilih radiobutton Filter Pemerataan.	✓	Berhasil karena akan menampilkan restorasi citra pada axes4.
Screen Shot  Tampilan citra filter pemerataan		

d. Black Box Radiobutton Filter Median

Tabel 4. 10 *Black box radiobutton filter median*

Test Factor	Hasil	Kesimpulan
User memilih radiobutton Filter Median.	✓	hasil karena akan menampilkan restorasi citra pada axes4.
Screen Shot  Tampilan citra filter median		

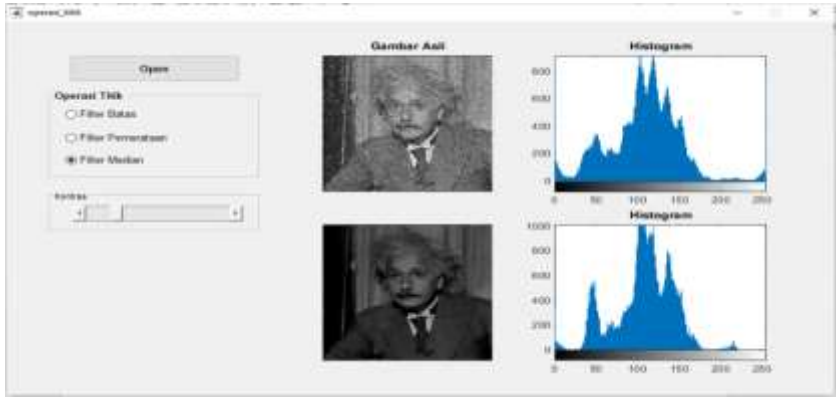
e. Black Box Radiobutton Citra negatif

Tabel 4. 11 *BlackBox Radiobutton Citra Negatif*

Test Factor	Hasil	Kesimpulan
User memilih radiobutton Citra Negatif.	✓	berhasil karena akan menampilkan restorasi citra pada axes4.
Screen Shot  Tampilan citra negatif		

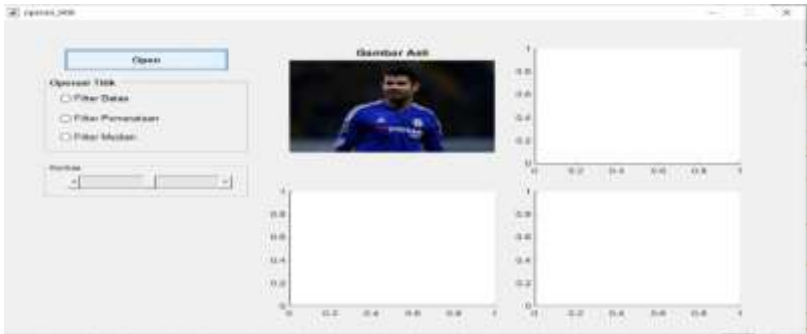
f. Black Box Slider Kontras

Tabel 4. 12 *Black box slider kontras menurunkan*

Test Factor	Hasil	Kesimpulan
User menggeser <i>slider</i> ke arah kiri.	✓	Berhasil karena akan menurunkan kontras citra pada <i>axes4</i> .
Screen Shot  Tampilan menurunkan kontras		

e. Black Box Citra Tidak Dapat Direstorasi

Tabel 4. 13 *Black box citra tidak dapat direstorasi*

Test Factor	Hasil	Kesimpulan
User memilih <i>radiobutton</i> Filter Batas.	✗	Tidak berhasil karena tidak menampilkan restorasi citra dan <i>histogram</i>
Screen Shot  Tampilan tidak dapat direstorasi		

BAB V

PENUTUP

A. Kesimpulan

Berdasarkan hasil penelitian dan pengujian yang telah dilakukan penulis dalam pembuatan Aplikasi Operasi Titik Pada Pengolahan Citra Digital Untuk Meningkatkan Kualitas Gambar disimpulkan bahwa :sss

1. Aplikasi ini hanya dapat berjalan maksimal pada gambar dengan citra tunggal.
2. Aplikasi ini hanya berjalan pada gambar dengan format *.BMP (Bitmap)*

B. Saran

Perencanaan Aplikasi Operasi Titik Pada Pengolahan Citra Digital Untuk Meningkatkan Kualitas Gambar Menggunakan Matlab penulis merasa masih banyak hal yang dapat dikembangkan. Untuk penulis selanjutnya diharapkan dapat mengembangkan dengan menambahkan menu yang dapat menampilkan perbandingan nilai pixel dari citra asli dengan citra restorasi dan juga dapat mengatur nilai RGB pada citra yang telah di restorasi.

Demikian saran yang dapat penulis berikan, semoga saran tersebut dapat dijadikan sebagai bahan masukan yang dapat bermanfaat bagi pembaca.

DAFTAR PUSTAKA

- Abd Fauzan (2018) Operasi Ketetanggaan Piksel pada Filter Batas, Filter Pemerataan dan Filter Median
- Ade Hastuty Hasyim Dasar Pemograman (2021)
- Dicoding Intern (2021) Apa itu Activity Diagram? Beserta Pengertian, Tujuan, Komponen
- Dicoding Intern (2021) Apa itu UML? Beserta Pengertian dan Contohnya
- Ketutrare (2020) Kategori dan Tujuan Operasi Pengolahan Citra Digital
- Nadiyah Rahmalia (2021) MATLAB, Platform Pemrograman dengan Segudang Kegunaan.
- Nida Khoiruroh (2017) ANALISA OPERASI TITIK MENGGUNAKAN METODE BRIGHTNESS, CONTRAST DAN TRESHOLDING
- Polma Saut Martua Sihotang (2018) Peningkatan Kualitas Citra Digital Menggunakan Metode Shock Filter
- Putra (2021) Pengertian Multimedia: Sejarah, Manfaat, Komponen & Contohnya Rizqi Hidayat Tanjung (2020) Perbaikan Kualitas Citra Digital Menggunakan Metode Multiscale Retinex Dan Median Filter
- Rizki Sari Dewi (2019) Pengertian Adobe Photoshop Beserta Sejarah, Fungsi, Kelebihan & Kekurangannya
- Robith Adani (2021) Mengenal Apa itu Use Case dan Teknik Pembuatannya Rony Setiawan (2021) Apa Itu Sequence Diagram dan Contohnya
- Rony Setiawan (2021) Flowchart Adalah: Fungsi, Jenis, Simbol, dan Contohnya
- Temukan Pengertian (2019) Pengertian Pengolahan Citra Digital Tujuan Konsep Operasi & Contohnya
- Tiara Elia (2018) Aplikasi Peningkatan Kualitas Citra Menggunakan Metode Histogram Equalization