

PROPOSAL SKRIPSI

**MEDIA PEMBELAJARAN PENGOLAHAN CITRA
MENGUNAKAN VISUALISASI INTERAKTIF BERBASIS
WEB**



RAHMAT ARDIANSYAH
193510147

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS ISLAM RIAU
2024

KATA PENGANTAR

Puji syukur kehadiran Allah SWT yang telah memberikan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan proposal tugas akhir yang berjudul “**Media Pembelajaran Pengolahan Citra Menggunakan Visualisasi Interaktif Berbasis Web**” Proposal tugas akhir ini disusun sebagai salah satu langkah awal dalam menyelesaikan program studi Sarjana Teknik Informatika.

Penulis menyadari bahwa pembuatan media pembelajaran ini tidak mudah, karena dibutuhkan pemahaman yang cukup tentang pengolahan citra digital dan teknologi web. Namun, dengan bimbingan dan dukungan dari dosen pembimbing, penulis yakin dapat menyelesaikan tugas akhir ini dengan baik.

Akhir kata, penulis mengucapkan terima kasih kepada semua pihak yang telah membantu dan mendukung penulis dalam menyelesaikan proposal tugas akhir ini.

Pekanbaru, 1 Juli 2024

Rahmat Ardiansyah

DAFTAR ISI

KATA PENGANTAR.....	i
DAFTAR ISI.....	ii
DAFTAR GAMBAR.....	iv
DAFTAR TABEL.....	v
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Identifikasi Masalah.....	3
1.3 Rumusan Masalah.....	3
1.4 Batasan Masalah.....	4
1.5 Tujuan Penelitian.....	4
1.6 Manfaat Penelitian.....	5
BAB II LANDASAN TEORI.....	6
2.1 Tinjauan Pustaka.....	6
2.2 Dasar Teori.....	8
2.2.1 Pengolahan Citra Digital.....	8
2.2.1.1 Operasi <i>Grayscale</i>	9
2.2.1.2 Operasi <i>Invert</i>	9
2.2.1.3 Operasi <i>Brightness</i>	10
2.2.1.4 Operasi <i>Threshold</i>	10
2.2.1.5 Operasi <i>Image Blending</i>	11
2.2.1.6 Operasi <i>Image Subtraction</i>	11
2.2.2 Media Pembelajaran.....	12
2.2.3 Metode <i>Waterfall</i>	12
2.2.4 Visualisasi.....	13
2.2.5 Web.....	14
2.3 Alat Bantu Pengembangan di Perancangan Sistem.....	15
2.3.1 Activity Diagram.....	15
2.3.2 Flowchart.....	17

2.3.3 <i>Software</i> Pendukung.....	18
2.3.3.1 Browser.....	18
2.3.3.2 HTML.....	18
2.3.3.3 CSS.....	19
2.3.3.4 JavaScript.....	19
2.3.3.5 React.js.....	20
2.3.3.6 Node.js.....	21
2.3.3.7 Tailwind CSS.....	22
2.3.3.8 Visual Studio Code.....	24
BAB III METODOLOGI PENELITIAN.....	26
3.1 Metode Penelitian.....	26
3.1.1 Metode Pengumpulan Data.....	26
3.1.1.1 Studi Pustaka.....	26
3.1.1.2 Wawancara.....	26
3.1.1.3 Observasi.....	27
3.1.2 Metode Pengembangan Aplikasi.....	27
3.1.2.1 Analisis Kebutuhan.....	27
3.1.2.2 Perancangan.....	27
3.1.2.3 Implementasi.....	45
3.1.2.4 Pengujian.....	45
3.1.2.5 Pemeliharaan.....	45
3.1.3 Alat dan Bahan Penelitian.....	46
3.1.3.1 Spesifikasi Perangkat Keras (<i>Hardware</i>).....	46
3.1.3.2 Spesifikasi Perangkat Lunak (<i>Software</i>).....	46
DAFTAR PUSTAKA.....	47

DAFTAR GAMBAR

Gambar 2.1: Simbol - simbol activity diagram.....	16
Gambar 2.2: Simbol - Simbol pada <i>flowchart</i>	17
Gambar 3.1 <i>Context Diagram</i> Media Pembelajaran Pengolahan Citra Menggunakan Visualisasi Interaktif Berbasis Web.....	28
Gambar 3.2 Hierarchy Chart.....	29
Gambar 3.3 Activity Diagram Media Pembelajaran Pengolahan Citra Menggunakan Visualisasi Interaktif Berbasis Web.....	30
Gambar 3.4. <i>Use Case Diagram</i> Media Pembelajaran Visualisasi Pengolahan Citra	31
Gambar 3.5: <i>Sequence Diagram</i> Media Pembelajaran Visualisasi Pengolahan Citra	33
Gambar 3.6 Flowchart Grayscale.....	34
Gambar 3.7 Flowchart Invert.....	35
Gambar 3.8 Flowchart Brightness.....	36
Gambar 3.9 Flowchart Threshold.....	37
Gambar 3.10 Flowchart <i>Image Blending</i>	38
Gambar 3.11 Flowchart <i>Image Substraction</i>	39
Gambar 3.12 Desain Input <i>Grayscale</i>	40
Gambar 3.13 Desain Input <i>Invert</i>	41
Gambar 3.14 Desain Input Operasi <i>Brightness</i>	42
Gambar 3.15 Desain Input Operasi <i>Image Blending</i>	43
Gambar 3.16 Desain Input Operasi Image Substraction.....	44

DAFTAR TABEL

Tabel 2.1 Hasil Penelitian Sebelumnya.....	6
--------------------------------------------	---

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pengolahan citra digital telah menjadi bidang yang semakin penting dalam era digital ini. Dalam berbagai bidang seperti ilmu komputer, grafika komputer, pengenalan pola, penglihatan komputer, dan banyak aplikasi lainnya, pengolahan citra digital berperan penting dalam analisis, manipulasi, dan interpretasi gambar digital. Pengolahan citra digital memungkinkan identifikasi dan ekstraksi fitur pada gambar digital, yang seringkali menggunakan warna sebagai komponen utama dalam representasi objek (Jumadi et al., 2021).

Pada saat yang sama, perkembangan teknologi informasi dan komunikasi (TIK) telah membawa dampak signifikan dalam dunia pendidikan. Pemanfaatan media pembelajaran dalam proses pendidikan telah menjadi semakin umum, dan penggunaan media berbasis web menjadi salah satu tren yang semakin populer. Media berbasis web memiliki keunggulan dalam aksesibilitas dan fleksibilitasnya yang tinggi, yang memungkinkan mahasiswa untuk mengakses dan berinteraksi dengan materi pembelajaran secara efektif.

Namun, media pembelajaran interaktif berbasis web dalam bidang pengolahan citra digital masih terbatas. Beberapa masalah yang dihadapi antara lain keterbatasan media pembelajaran interaktif yang dapat membantu memahami konsep kompleks pengolahan citra, rendahnya aksesibilitas karena sebagian media hanya tersedia dalam bentuk cetak atau pada platform terbatas, serta sulitnya memahami konsep abstrak seperti transformasi gambar dan konversi warna tanpa visualisasi yang dinamis. Media cetak seperti buku sering kali tidak menyediakan contoh visual atau animasi yang diperlukan untuk menjelaskan transformasi citra secara detail. Hal ini mengakibatkan mahasiswa atau pembelajar sering kali

kesulitan untuk membayangkan proses dan hasil dari teknik pengolahan citra tertentu.

Dalam konteks ini, penelitian ini bertujuan untuk mengembangkan sebuah media pembelajaran pengolahan citra digital yang menggunakan visualisasi interaktif berbasis web. Dengan memanfaatkan potensi media berbasis web, penelitian ini bertujuan untuk menyediakan alat yang interaktif dan dinamis bagi mahasiswa untuk belajar dan mengajarkan konsep pengolahan citra digital secara lebih efektif.

Media pembelajaran yang diusulkan akan memberikan visualisasi yang menarik dan interaktif untuk membantu pemahaman konsep-konsep dasar dalam pengolahan citra digital. Mahasiswa akan dapat berinteraksi dengan gambar dan algoritma pengolahan citra, memanipulasi parameter, dan melihat hasil transformasi citra secara langsung melalui antarmuka berbasis web yang ramah pengguna. Dengan adanya media pembelajaran ini, diharapkan pembelajar akan lebih terlibat dan tertarik dalam proses pembelajaran pengolahan citra digital. Selain itu, dosen juga akan mendapatkan alat yang berguna untuk mengajarkan konsep-konsep ini dengan cara yang lebih menarik dan mudah dipahami. Teknik pengolahan citra digital bertujuan untuk meningkatkan kualitas gambar agar lebih mudah diinterpretasikan, baik oleh manusia maupun oleh mesin komputer (Hutahaeen et al., 2019).

Penelitian ini diharapkan dapat memberikan kontribusi dalam pengembangan media pembelajaran yang inovatif dan interaktif dalam bidang pengolahan citra digital. Dengan meningkatkan kualitas pembelajaran di bidang ini, diharapkan dapat mempersiapkan mahasiswa untuk menghadapi tantangan dan peluang di dunia yang semakin digital.

1.2 Identifikasi Masalah

Berikut adalah beberapa identifikasi masalah yang dapat diambil dari latar belakang di atas:

1. Saat ini, masih terbatasnya media pembelajaran berbasis web yang interaktif dalam pengolahan citra digital.
2. Kurangnya aksesibilitas dan fleksibilitas. Beberapa media pembelajaran pengolahan citra digital mungkin hanya tersedia dalam bentuk cetak atau terbatas pada platform tertentu.
3. Pengolahan citra digital melibatkan konsep-konsep yang kompleks dan abstrak.

1.3 Rumusan Masalah

Berdasarkan latar belakang di atas, maka dapat dirumuskan masalah utama dari proposal tugas akhir ini sebagai berikut:

Bagaimana mengembangkan media pembelajaran pengolahan citra digital yang lebih interaktif dan mudah dipahami oleh mahasiswa dengan memanfaatkan teknologi web dan visualisasi interaktif?

Untuk menjawab masalah tersebut, diperlukan beberapa pertanyaan penelitian yang lebih spesifik, yaitu:

1. Apa saja konsep-konsep dasar yang harus dipahami oleh mahasiswa dalam pembelajaran pengolahan citra digital?
2. Bagaimana mengembangkan media pembelajaran pengolahan citra digital yang interaktif dan mudah dipahami dengan memanfaatkan teknologi web dan visualisasi interaktif?
3. Bagaimana mengukur efektivitas media pembelajaran yang telah dikembangkan?

1.4 Batasan Masalah

Dalam penelitian ini, terdapat beberapa batasan masalah yang perlu diperhatikan. Batasan-batasan tersebut adalah:

1. Penelitian ini akan difokuskan pada pengolahan citra digital sebagai subjek utama dalam media pembelajaran yang diusulkan. Lingkup pengolahan citra digital meliputi konsep dasar seperti operasi *grayscale*, operasi *invert*, operasi *threshold*, operasi *image blending* dan operasi *image subtraction*.
2. Media pembelajaran akan dikembangkan dalam bentuk aplikasi berbasis web sehingga dapat diakses dengan mudah melalui berbagai perangkat.
3. Aplikasi media pembelajaran yang dikembangkan akan memiliki beberapa fungsi dan fitur, seperti tampilan interaktif, kontrol dan manipulasi citra, dan pemrosesan citra secara *real-time*.
4. Pengembangan media pembelajaran pengolahan citra digital dengan visualisasi interaktif berbasis web ini akan dilakukan dalam waktu tertentu, sehingga batasan waktu pengembangan akan menjadi salah satu batasan yang harus diperhatikan dalam tugas akhir ini.

1.5 Tujuan Penelitian

1. Mengembangkan media pembelajaran yang lebih interaktif dan mudah dipahami dengan memanfaatkan teknologi web dan visualisasi interaktif.
2. Meningkatkan kualitas pembelajaran pengolahan citra digital, terutama dalam mempermudah mahasiswa memahami konsep-konsep yang kompleks dan abstrak.
3. Memberikan kontribusi positif terhadap dunia pendidikan serta pengembangan teknologi informasi dan komunikasi.

1.6 Manfaat Penelitian

Berikut adalah beberapa manfaat yang dapat diperoleh melalui penelitian tugas akhir ini:

1. Dengan adanya media pembelajaran yang lebih interaktif dan mudah dipahami, diharapkan kualitas pembelajaran pengolahan citra digital dapat meningkat.
2. Membantu mahasiswa memahami konsep-konsep yang kompleks dan abstrak. Dengan adanya media pembelajaran ini diharapkan dapat membantu mahasiswa dalam memahami konsep-konsep tersebut.
3. Media pembelajaran yang saat ini digunakan masih terbatas dan belum banyak memanfaatkan teknologi terkini. Dengan adanya media pembelajaran yang memanfaatkan teknologi web dan visualisasi interaktif, diharapkan dapat memberikan alternatif media pembelajaran yang lebih modern.
4. Penelitian ini diharapkan dapat memberikan sumbangsih positif bagi dunia pendidikan dan pengembangan teknologi informasi dan komunikasi. Dengan adanya media pembelajaran yang lebih inovatif dan efektif, diharapkan dapat membantu meningkatkan kualitas pendidikan dan pengembangan teknologi informasi dan komunikasi.

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Dalam penelitian ini, beberapa referensi kepustakaan yang bersumber pada penelitian sebelumnya diambil sebagai bahan referensi. Referensi ini digunakan untuk membantu dalam menyelesaikan penelitian yang sedang dilakukan. Peneliti menggunakan referensi-referensi tersebut sebagai acuan dan sumber informasi untuk memperdalam pemahaman tentang topik yang sedang diteliti.

Tabel 2.1 Hasil Penelitian Sebelumnya

No	Penelitian	Metode	Hasil
1	(Suryadibrata & Young, 2020)	Waterfall	Visualisasi meningkatkan pemahaman pelajar terhadap proses pengelompokan data.
2	(Kapti et al., 2023)	MDLC	Ada kenaikan rata-rata nilai post-test terhadap rata-rata nilai pre-test sebesar 11 poin. Hal ini membuktikan terjadinya peningkatan pemahaman siswa tentang daur hidup hewan
3	(Armansyah et al., 2019)	Model Lee dan Owens	Uji coba terhadap subjek penelitian menunjukkan dari nilai rata-rata 46,8 pada saat pre-test menjadi mendapatkan nilai rata-rata 72,7 pada saat post-test dari

		27 siswa.	
4	(Kollo & Boimau, 2020)	ADDIE	Pembelajaran hasil pengembangan dengan memvisualisasigua Jepang di Kupang, kemudian diterapkan di dalam pembelajaran dipandang efektif untuk meningkatkan kesadaran sejarah siswa kelas XI SMA Sudirman Kupang.

Berdasarkan tinjauan pustaka yang telah disajikan, dapat disimpulkan bahwa penggunaan aplikasi visualisasi dan media pembelajaran interaktif dapat meningkatkan kualitas pembelajaran serta motivasi belajar mahasiswa. Penelitian terdahulu yang telah dilakukan oleh para peneliti sebelumnya dapat menjadi acuan dan bahan pertimbangan dalam menganalisis hasil penelitian ini. Diharapkan penelitian ini dapat memberikan sumbangsih bagi pengembangan media pembelajaran yang lebih inovatif dan efektif untuk meningkatkan kualitas pendidikan.

Penelitian ini memiliki kesamaan dengan penelitian sebelumnya yaitu keduanya menggunakan aplikasi visualisasi sebagai media pembelajaran. Namun, terdapat perbedaan dimana penelitian sebelumnya memvisualisasikan *K-Means Clustering* dengan menggunakan bahasa pemrograman *C#* berbasis aplikasi *desktop*, sedangkan penelitian ini memvisualisasikan operasi pengolahan citra digital dengan menggunakan bahasa pemrograman *JavaScript* berbasis web.

2.2 Dasar Teori

2.2.1 Pengolahan Citra Digital

(Munantri et al., 2020) menjelaskan bahwa citra digital adalah gambar dua dimensi yang dihasilkan dari proses sampling pada gambar analog dua dimensi yang kontinu, sehingga menjadi gambar diskrit yang dapat diolah oleh komputer. Citra digital disimpan dalam bentuk data numerik yang menunjukkan besar intensitas pada masing-masing pikselnya. Oleh karena itu, citra digital dapat diolah dengan menggunakan komputer untuk berbagai keperluan seperti pengolahan gambar, analisis citra, dan pengenalan pola.

Masih menurut (Munantri et al., 2020) menjelaskan bahwa pengolahan citra digital merupakan ilmu yang mempelajari berbagai hal terkait dengan perbaikan kualitas gambar, transformasi gambar, pemilihan citra ciri yang optimal, penyimpanan data, transmisi data, dan waktu proses data. Proses pengolahan citra dapat dijelaskan dengan menggunakan diagram sederhana yang meliputi beberapa tahap, seperti perbaikan kualitas citra, transformasi citra, pemilihan citra ciri, reduksi dan kompresi data, transmisi data, dan waktu proses data. Dalam pengolahan citra digital, teknik-teknik pengolahan yang digunakan dapat berbeda-beda tergantung pada jenis citra dan tujuan analisis yang ingin dicapai.

Dari uraian diatas penulis menyimpulkan bahwa citra digital adalah hasil dari proses sampling pada gambar analog dua dimensi yang kemudian diubah menjadi gambar diskrit dengan data numerik yang merepresentasikan besar intensitas pada masing-masing piksel. Citra digital dapat diolah dengan menggunakan komputer dan teknik-teknik pengolahan citra yang berbeda-beda tergantung pada jenis citra dan tujuan analisis yang ingin dicapai. Proses pengolahan citra digital meliputi berbagai tahap seperti perbaikan kualitas citra, transformasi citra, pemilihan citra ciri, reduksi dan kompresi data, transmisi data, dan waktu proses data. Dalam ilmu pengolahan citra digital, terdapat berbagai

macam teknik yang dapat digunakan untuk mengambil informasi dari citra digital atau memperbaiki kualitas citra.

2.2.1.1 Operasi *Grayscale*

Operasi *grayscale* adalah proses normalisasi tiga lapisan warna RGB dari sebuah citra berwarna menjadi satu lapisan *grayscale* (Suryowinoto & Hamid, 2017). Dalam citra digital, *grayscale* memiliki gradasi warna dari putih ke hitam, seperti yang ditunjukkan oleh rumus dibawah. Rentang ini menunjukkan bahwa setiap piksel direpresentasikan oleh 8 bit. Karena citra digital grayscale sebenarnya adalah hasil rata-rata (dinormalisasi) dari citra berwarna, persamaannya dapat dituliskan sebagai berikut:

$$f_o(x, y) = \frac{f_i^R(x, y) + f_i^G(x, y) + f_i^B(x, y)}{3}$$

Di mana $f_i^R(x, y)$ adalah nilai piksel warna merah pada titik (x, y) , $f_i^G(x, y)$ adalah nilai piksel warna hijau pada titik (x, y) , dan $f_i^B(x, y)$ adalah nilai piksel warna biru pada titik (x, y) .

2.2.1.2 Operasi *Invert*

Citra invert, atau citra negatif, adalah citra yang merupakan kebalikan dari citra asli (Mahardika et al., 2017). Mirip dengan film negatif yang dihasilkan dari kamera konvensional. Jika sebuah citra memiliki jumlah *gray level* L dengan rentang dari 0 hingga $L-1$, citra negatif dapat diperoleh melalui transformasi negatif yang dijelaskan oleh persamaan berikut:

$$s = L - 1 - r$$

Di mana:

s = citra hasil transformasi negatif

L = jumlah *gray level* sebuah citra

r = citra asli

2.2.1.3 Operasi *Brightness*

Proses operasi *brightness* dilakukan dengan menambahkan atau mengurangi nilai setiap piksel dengan suatu konstanta (Yasa et al., 2018). Penyesuaian tingkat kecerahan suatu citra dapat dinyatakan sebagai:

$$U = U + c$$

Dengan U dan U berturut-turut menyatakan citra setelah dan sebelum *brightness adjustment* sedangkan c adalah suatu konstanta yang merupakan variabel penyesuaian.

2.2.1.4 Operasi *Threshold*

Thresholding adalah metode segmentasi citra yang memisahkan objek dari latar belakang berdasarkan perbedaan kecerahan (Setiawan et al., 2019). Daerah yang lebih gelap akan semakin gelap (hitam sempurna dengan nilai intensitas 0), sedangkan daerah yang lebih terang akan semakin terang. Operasi ambang batas tunggal membagi nilai piksel menjadi dua kelompok seperti ditunjukkan oleh rumus berikut:

$$g(x,y) = \left\{ \begin{array}{ll} 0, & \text{jika } f(x,y) < T \\ 255, & \text{jika } f(x,y) \geq T \end{array} \right\}$$

Piksel dengan nilai intensitas di bawah T akan diubah menjadi hitam (nilai intensitas 0), sedangkan piksel dengan nilai intensitas di atas T akan diubah menjadi putih (nilai intensitas 255).

2.2.1.5 Operasi *Image Blending*

Image blending atau penjumlahan citra adalah teknik menggabungkan dua atau lebih citra untuk menghasilkan citra baru. Teknik ini sering digunakan dalam aplikasi seperti penggabungan panorama, pengurangan noise, dan efek khusus. Salah satu metode umum adalah blending linear, di mana dua citra digabungkan dengan bobot tertentu. Jika ada dua citra I_1 dan I_2 , rumusnya adalah:

$$I_{\text{blended}}(x, y) = \alpha \cdot I_1(x, y) + \beta \cdot I_2(x, y)$$

Di mana $I_1(x, y)$ dan $I_2(x, y)$ adalah intensitas piksel pada posisi (x, y) dari citra pertama dan kedua, sedangkan α dan β adalah bobot untuk masing-masing citra, dengan $\alpha + \beta = 1$.

2.2.1.6 Operasi *Image Subtraction*

Image *subtraction* adalah teknik pengolahan citra yang melibatkan pengurangan nilai intensitas piksel dari satu citra dengan nilai intensitas piksel yang bersesuaian dari citra lain. Teknik ini sering digunakan untuk mendeteksi perubahan atau mengurangi latar belakang. Jika ada dua citra I_1 dan I_2 , pengurangan citra dapat dinyatakan sebagai:

$$I_{\text{subtracted}}(x, y) = I_1(x, y) - I_2(x, y)$$

Di mana $I_{\text{subtracted}}(x, y)$ adalah intensitas piksel pada posisi (x, y) dari citra hasil pengurangan, $I_1(x, y)$ adalah intensitas piksel pada posisi (x, y) dari citra pertama, dan $I_2(x, y)$ adalah intensitas piksel pada posisi (x, y) dari citra

kedua. Untuk mengatasi nilai negatif, bisa menambahkan nilai konstan yang cukup besar untuk menggeser semua nilai ke rentang positif, misalnya dengan menambahkan 255 ke setiap nilai hasil pengurangan, lalu melakukan clipping untuk memastikan nilai tidak melebihi batas atas (255) sebagai berikut:

$$I_{\text{normalized}}(x, y) = \max(0, \min(255, I_{\text{subtracted}}(x, y) + 255))$$

2.2.2 Media Pembelajaran

Menurut Arsyad dan Rahman (2015) kata media berasal dari bahasa latin *medius* yang secara harfiah berarti “tengah”, “perantara” atau “pengantar”, yang dalam bahasa arab diartikan sebagai perantara atau pengantar pesan dari pengirim kepada penerima pesan. Oleh karena itu, media diartikan sebagai alat yang menyampaikan atau mengantarkan pesan-pesan pengajaran.

Nurrita (2018) menjelaskan bahwa media pembelajaran berperan sebagai alat yang dapat membantu proses belajar mengajar sehingga makna pesan yang disampaikan dapat menjadi lebih jelas dan tujuan pendidikan atau pembelajaran dapat tercapai dengan efektif dan efisien.

Berdasarkan uraian diatas dapat disimpulkan bahwa media pembelajaran merupakan alat atau sarana yang berperan sebagai perantara atau pengantar pesan dalam proses belajar mengajar. Dalam proses pembelajaran, media pembelajaran dapat membantu membuat pesan yang disampaikan menjadi lebih jelas dan tujuan pembelajaran dapat tercapai dengan efektif dan efisien. Oleh karena itu, penggunaan media pembelajaran sangat penting dalam proses belajar mengajar untuk mencapai hasil yang optimal.

2.2.3 Metode *Waterfall*

Metodologi *Waterfall* melibatkan pengembangan perangkat lunak secara berurutan dan linier. Proses pengembangan dimulai dengan tahap perencanaan,

dilanjutkan dengan analisis kebutuhan, perancangan sistem, implementasi, validasi, dan pemeliharaan (Wahid, 2020).

Setiap tahap harus diselesaikan secara menyeluruh tanpa meninjau atau mengulangi tahap sebelumnya sebelum melanjutkan ke tahap berikutnya. Model ini sangat cocok untuk proyek yang memiliki persyaratan yang jelas sejak awal dan jarang berubah selama proses pengembangan. Metode *waterfall* pertama kali diperkenalkan oleh Winston Royce pada tahun 1970. Meskipun metode ini sering dianggap sebagai model klasik, namun metode ini masih banyak digunakan hingga saat ini karena strukturnya yang jelas dan sistematis.

Keunggulan utama metode ini adalah membangun sistem mutu dengan melalui proses detail pada setiap tahapannya. Namun kelemahannya adalah periode pengembangan yang lama dan kurangnya fleksibilitas terhadap perubahan yang terjadi selama pengembangan (seperti analisis air terjun). Model *waterfall* sangat cocok untuk proyek besar, terutama yang melibatkan pembuatan sistem baru. Namun, untuk proyek yang memerlukan perubahan cepat atau dipengaruhi oleh persyaratan yang sering berubah, model ini kurang efisien karena sifatnya yang kaku (seperti analisis air terjun).

2.2.4 Visualisasi

Visualisasi memiliki beberapa landasan teori yang mendasar, di antaranya adalah teori pengolahan informasi, teori persepsi visual, dan teori kognitif. Teori pengolahan informasi menjelaskan bahwa manusia memproses informasi dengan cara menerima, menyimpan, mengorganisir, dan mengambil kembali informasi dalam memori. Teori ini menjadi dasar bagi visualisasi karena visualisasi digunakan untuk mempermudah pengolahan informasi dengan menggambarkan data dalam bentuk visual yang mudah dipahami.

Teori persepsi visual menjelaskan tentang bagaimana manusia mengamati dan menginterpretasikan dunia visual. Teori ini menjelaskan bagaimana manusia memahami bentuk, warna, ukuran, dan posisi objek dalam lingkungan visual.

Teori ini menjadi penting dalam visualisasi karena visualisasi berupaya untuk membuat gambaran yang tepat dan mudah dipahami oleh pengamat.

Sedangkan teori kognitif membahas tentang bagaimana manusia memproses, menyimpan, mengambil, dan menggunakan informasi yang telah dipelajari. Teori ini menjadi dasar bagi pengembangan visualisasi karena visualisasi bertujuan untuk meningkatkan pemahaman dan retensi informasi pada manusia dengan menyajikan informasi dalam bentuk yang mudah diingat dan dipahami.

Dari ketiga landasan teori di atas, dapat disimpulkan bahwa visualisasi sebagai alat bantu memahami informasi berusaha mempermudah pengolahan informasi manusia dengan menggunakan teori pengolahan informasi, teori persepsi visual, dan teori kognitif sebagai dasar pengembangan.

2.2.5 Web

Web atau *World Wide Web (WWW)* adalah sistem informasi global yang terhubung melalui jaringan internet dan digunakan untuk mengakses dan berbagi informasi di seluruh dunia. Web pertama kali diperkenalkan pada tahun 1989 oleh Tim Berners-Lee, seorang ilmuwan komputer dari Inggris. Dalam pengertian umum, web adalah file teks murni yang berisi HTML, yang dapat diterjemahkan dengan Internet Browser (Maiyana, 2017).

Pada awalnya, web hanya digunakan sebagai alat untuk membagikan informasi dan sebagai tempat untuk mengakses situs web. Namun, seiring dengan perkembangan teknologi, web kini telah menjadi lebih interaktif dan dinamis. Contohnya adalah adanya aplikasi web yang memungkinkan pengguna untuk melakukan transaksi online, berinteraksi dengan orang lain, bermain game, bahkan menjadi media pembelajaran.

Web sendiri terdiri dari tiga komponen utama, yaitu bahasa markup (HTML), *style sheet language* (CSS), dan bahasa scripting (*JavaScript*). HTML digunakan untuk membuat struktur dasar halaman web, sedangkan CSS digunakan


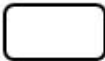




untuk mendesain tampilan halaman web, dan JavaScript digunakan untuk membuat halaman web menjadi interaktif.

2.3 Alat Bantu Pengembangan di Perancangan Sistem

Adapun alat bantu untuk pengembangan perancangan sistem aplikasi media pembelajaran ini adalah

2.3.1 Activity Diagram

Activity diagram adalah salah satu jenis diagram dalam *Unified Modeling Language (UML)* yang digunakan untuk memodelkan alur kerja atau proses bisnis dalam sebuah sistem. *Activity diagram* menggambarkan berbagai alur kegiatan dalam sistem yang sedang dirancang, mulai dari kondisi awal setiap alur, keputusan yang mungkin diambil, hingga akhir dari kegiatan tersebut (M. Pratama & Delianti, 2021). Diagram ini memberikan representasi visual dari aktivitas dan urutan tindakan yang terjadi dalam sebuah proses, menunjukkan bagaimana satu aktivitas mengarah ke aktivitas lainnya. Dengan kata lain, *activity diagram* menggambarkan aliran kontrol dari satu aktivitas ke aktivitas lain dan bagaimana berbagai kondisi dan keputusan mempengaruhi aliran ini.

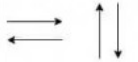











Simbol	Nama	Keterangan
	Status awal	Sebuah diagram aktivitas memiliki sebuah status awal.
	Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
	Percabangan / Decision	Percabangan dimana ada pilihan aktivitas yang lebih dari satu.
	Penggabungan / Join	Penggabungan dimana yang mana lebih dari satu aktivitas lalu digabungkan jadi satu.
	Status Akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
	Swimlane	Swimlane memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

Gambar 2.1: Simbol - simbol activity diagram

Activity diagram berguna dalam menggambarkan proses yang kompleks, memperjelas logika alur kerja, dan mengidentifikasi potensi perbaikan atau optimasi. Diagram ini terdiri dari elemen-elemen seperti *activity nodes* (aktivitas), *control flows* (aliran kontrol), *decision nodes* (simpul keputusan), dan *merge nodes* (simpul penggabungan). Elemen-elemen ini memungkinkan pengembang dan pemangku kepentingan untuk memahami proses secara menyeluruh, memfasilitasi komunikasi yang efektif, dan memastikan bahwa semua aspek dari alur kerja telah dipertimbangkan. Berikut adalah simbol – simbol yang biasanya digunakan di dalam activity diagram yang ditunjukkan pada gambar 2.1.

2.3.2 Flowchart

Flowchart, atau diagram alir, adalah sebuah diagram yang menggunakan simbol-simbol dan garis untuk merepresentasikan alur dari suatu algoritma atau proses (Karim, 2018). Setiap langkah dalam proses tersebut diwakili oleh simbol-simbol tertentu, seperti persegi panjang untuk langkah proses, belah ketupat untuk keputusan, dan oval untuk awal atau akhir proses seperti yang ditunjukkan pada gambar 2.2. Dengan menggunakan *flowchart*, seseorang dapat dengan mudah memahami alur kerja atau prosedur yang kompleks, mengidentifikasi potensi hambatan, serta memperbaiki efisiensi suatu sistem. Flowchart juga membantu dalam perencanaan, pengelolaan proyek, dan dokumentasi proses, karena memberikan representasi yang jelas dan terstruktur dari setiap langkah yang harus diambil.

	Flow Simbol yang digunakan untuk menggabungkan antara simbol yang satu dengan simbol yang lain. Simbol ini disebut juga dengan Connecting Line.		Input/output Simbol yang menyatakan proses input atau output tanpa tergantung peralatan.
	On-Page Reference Simbol untuk keluar - masuk atau penyambungan proses dalam lembar kerja yang sama.		Manual Operation Simbol yang menyatakan suatu proses yang tidak dilakukan oleh komputer.
	Off-Page Reference Simbol untuk keluar - masuk atau penyambungan proses dalam lembar kerja yang berbeda.		Document Simbol yang menyatakan bahwa input berasal dari dokumen dalam bentuk fisik, atau output yang perlu dicetak.
	Terminator Simbol yang menyatakan awal atau akhir suatu program.		Predefine Proses Simbol untuk pelaksanaan suatu bagian (sub-program) atau prosedur.
	Process Simbol yang menyatakan suatu proses yang dilakukan komputer.		Display Simbol yang menyatakan peralatan output yang digunakan.
	Decision Simbol yang menunjukan kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban, yaitu ya dan tidak.		Preparation Simbol yang menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberikan nilai awal.

Gambar 2.2: Simbol - Simbol pada *flowchart*

Flowchart biasanya dibaca dari atas ke bawah atau dari kiri ke kanan, tergantung pada bagaimana alur proses tersebut diatur. Garis atau panah digunakan untuk menghubungkan setiap simbol, menunjukkan urutan langkah-

langkah yang harus diikuti. Penggunaan *flowchart* sangat umum dalam berbagai bidang, termasuk pemrograman, manajemen proyek, dan teknik industri, karena kemampuannya untuk memvisualisasikan informasi secara sederhana dan efektif.

2.3.3 Software Pendukung

2.3.3.1 Browser

Browser adalah perangkat lunak atau aplikasi yang berfungsi untuk mengakses dan menampilkan halaman web. Secara lebih spesifik, browser web merupakan aplikasi perangkat lunak yang digunakan untuk menelusuri, mengambil, serta menampilkan informasi di *World Wide Web*, termasuk halaman web, gambar, video, dan berbagai file lainnya (Noviantoro et al., 2022). Dengan menggunakan browser, pengguna dapat menjelajahi berbagai situs web, menonton video, membaca artikel, dan berinteraksi dengan konten online lainnya. Browser bekerja dengan cara mengirim permintaan ke server web untuk menampilkan halaman yang diinginkan, kemudian menampilkan halaman tersebut dalam format yang dapat dibaca dan dinavigasi oleh pengguna. Browser modern juga mendukung berbagai fungsi tambahan, seperti pengelolaan tab, penyimpanan data sementara (cache), dan ekstensi yang menambah fitur baru atau meningkatkan keamanan selama berselancar di internet.

2.3.3.2 HTML

HTML, atau *HyperText Markup Language*, adalah bahasa standar yang digunakan untuk membuat dan menyusun konten pada halaman web. HTML terdiri dari serangkaian elemen atau tag yang digunakan untuk mendefinisikan berbagai jenis konten, seperti teks, gambar, tautan, dan tabel. Setiap elemen HTML memiliki fungsi tertentu yang menentukan bagaimana konten tersebut akan ditampilkan di browser. Dengan menggunakan HTML, pengembang web dapat membuat struktur dasar dari sebuah halaman web, yang kemudian dapat ditambahkan gaya menggunakan CSS dan interaktivitas menggunakan JavaScript. HTML adalah fondasi dari semua halaman web yang ada di internet. Dapat

disimpulkan bahwa HTML adalah protokol yang digunakan untuk mentransfer data atau dokumen dari web server ke browser (Oktavianto et al., 2022).

2.3.3.3 CSS

Cascading Style Sheet (CSS) adalah aturan yang digunakan untuk mengatur berbagai komponen dalam sebuah web agar tampil lebih terstruktur dan konsisten (Kumalasari et al., 2023). CSS memungkinkan pengembang untuk memisahkan konten dari desain visual, sehingga mereka dapat mengontrol berbagai aspek seperti warna, font, margin, dan posisi elemen di halaman web. Dengan CSS, tampilan halaman dapat disesuaikan agar konsisten di seluruh situs, serta responsif terhadap berbagai ukuran layar dan perangkat. CSS bekerja bersama dengan HTML, di mana HTML menyusun struktur halaman, sedangkan CSS mengatur bagaimana halaman tersebut akan terlihat di browser.

2.3.3.4 JavaScript

JavaScript adalah bahasa pemrograman yang sangat penting dalam pembuatan website dan aplikasi web. Bahasa pemrograman ini sering digunakan untuk menambahkan interaktifitas pada website, membuat efek animasi, memvalidasi input pengguna, dan banyak lagi. Dalam konteks pengolahan citra digital menggunakan visualisasi interaktif berbasis web, *JavaScript* dapat digunakan untuk membuat tampilan interaktif yang memudahkan pengguna dalam memahami proses pengolahan citra yang sedang dilakukan.

Salah satu contoh penggunaan *JavaScript* dalam pembuatan media pembelajaran pengolahan citra digital adalah dengan membuat animasi yang menggambarkan proses sampling pada citra analog dan proses konversi menjadi citra digital. Dalam animasi ini, *JavaScript* dapat digunakan untuk mengatur waktu tampilan dan memanipulasi objek-objek visual seperti gambar, teks, dan grafik. Selain itu, *JavaScript* juga dapat digunakan untuk membuat interaksi pengguna yang memungkinkan mereka untuk memilih dan memanipulasi citra digital dengan mudah.

Penggunaan *JavaScript* dalam media pembelajaran pengolahan citra digital juga dapat diperluas untuk mencakup fitur-fitur lain seperti pengolahan filter dan perubahan kontras pada citra digital. Dengan menggunakan *JavaScript*, pengguna dapat dengan mudah memilih filter yang ingin digunakan dan melihat perbedaan sebelum dan sesudah filter diterapkan pada citra digital. Selain itu, *JavaScript* juga dapat digunakan untuk menampilkan grafik dan diagram yang membantu pengguna dalam memahami data citra digital yang kompleks.

Dalam keseluruhan, penggunaan *JavaScript* dalam media pembelajaran pengolahan citra digital dengan visualisasi interaktif berbasis web sangat penting dalam meningkatkan interaktivitas dan pemahaman pengguna terhadap proses pengolahan citra. Dengan penggunaan teknologi yang tepat dan penanganan dengan hati-hati, media pembelajaran ini dapat menjadi alat yang efektif untuk memperkenalkan dan mengajarkan konsep-konsep pengolahan citra digital kepada mahasiswa dan pembelajar di mana saja dan kapan saja.

2.3.3.5 React.js

React.js adalah *library JavaScript* yang digunakan untuk membangun antarmuka pengguna (*user interfaces*) secara efisien. React.js memiliki beberapa keunggulan diantaranya memberikan kecepatan, *simplicity*, dan *scalability* (Panjaitan & Pakpahan, 2021). Dikembangkan oleh Facebook, React memungkinkan pengembang untuk membuat aplikasi web yang interaktif dan dinamis dengan lebih mudah.

Salah satu konsep utama dalam React adalah komponen. Komponen adalah blok penyusun dalam sebuah aplikasi React. Setiap komponen bisa dianggap sebagai bagian kecil dari UI yang dapat digunakan kembali. Misalnya, tombol, formulir, atau daftar item bisa dijadikan sebagai komponen terpisah. Setiap komponen ini didefinisikan menggunakan fungsi atau kelas dalam *JavaScript*.

React juga dikenal dengan pendekatan *declarative* yang digunakan dalam membangun UI. Alih-alih menginstruksikan komputer tentang cara menciptakan

elemen-elemen UI, kita hanya mendeklarasikan apa yang seharusnya terlihat berdasarkan keadaan aplikasi. React kemudian menangani detail pembaruan DOM di bawah permukaan.

Virtual DOM adalah fitur penting lainnya dalam React. Saat terjadi perubahan pada UI, React tidak langsung memperbarui DOM di browser. Sebaliknya, ia menciptakan salinan virtual dari DOM yang kemudian dibandingkan dengan DOM yang asli. Hanya perbedaan (difference) antara dua DOM inilah yang diterapkan pada DOM nyata, yang membuat pembaruan UI lebih cepat dan efisien.

Selain itu, React mendukung **state management** yang memungkinkan komponen untuk memiliki dan mengelola statusnya sendiri. Ketika status (state) berubah, React secara otomatis memperbarui tampilan sesuai dengan perubahan tersebut.

Dengan berbagai fiturnya, React.js telah menjadi salah satu pustaka paling populer di kalangan pengembang web untuk membangun aplikasi web yang cepat, modular, dan mudah dikelola.

2.3.3.6 Node.js

Node.js adalah lingkungan runtime *JavaScript* yang bersifat open-source dan dapat digunakan di berbagai platform. Dengan Node.js, kita bisa menjalankan kode JavaScript di sisi server. (N. F. Pratama et al., 2023). Ini berarti *JavaScript*, yang biasanya digunakan untuk membangun antarmuka pengguna di sisi klien, kini dapat digunakan untuk menangani backend dari aplikasi, termasuk mengelola permintaan HTTP, mengakses basis data, dan banyak lagi.

Salah satu fitur utama Node.js adalah arsitektur non-blocking I/O-nya, yang memungkinkan server untuk menangani banyak permintaan secara bersamaan tanpa menunggu operasi yang memakan waktu, seperti membaca file atau mengakses basis data, selesai. Ini dicapai melalui penggunaan *event-driven* yang

membuat aplikasi Node.js sangat efisien dan mampu menangani sejumlah besar koneksi dengan konsumsi sumber daya yang minimal.

Node.js dibangun di atas *engine* V8 milik Google, yang merupakan mesin yang juga digunakan oleh browser Chrome untuk mengeksekusi *JavaScript*. Ini berarti kode *JavaScript* yang berjalan di Node.js sangat cepat dan efisien, karena V8 terus dioptimalkan oleh Google.

Node.js juga memiliki npm (Node Package Manager), sebuah ekosistem besar yang menyediakan berbagai pustaka dan modul yang dapat digunakan untuk mempercepat pengembangan. Dengan npm, pengembang dapat dengan mudah menambahkan fitur tambahan ke dalam aplikasi mereka, seperti autentikasi, akses basis data, atau pemrosesan file, dengan hanya beberapa baris perintah.

Dalam pengembangan web modern, Node.js sering digunakan untuk membangun aplikasi *full-stack*, di mana *JavaScript* digunakan di sisi klien dan server. Ini memungkinkan pengembang untuk menggunakan satu bahasa pemrograman di seluruh aplikasi, yang dapat mempercepat pengembangan dan mempermudah pemeliharaan kode.

Dengan kemampuannya untuk menangani operasi asinkron secara efisien, dukungan ekosistem yang luas, dan kecepatan eksekusi, Node.js telah menjadi pilihan populer untuk membangun aplikasi web, API, layanan mikro, dan bahkan aplikasi real-time seperti chat atau game online.

2.3.3.7 Tailwind CSS

Tailwind CSS adalah sebuah kerangka kerja CSS yang berbeda dari yang lain karena pendekatannya yang unik dalam menangani gaya. Alih-alih menyediakan komponen yang sudah jadi seperti tombol atau kartu, Tailwind menyediakan *utility classes*, yaitu kelas-kelas kecil yang dirancang untuk melakukan satu tugas tertentu, seperti memberikan margin, padding, atau mengatur warna. Dengan pendekatan ini, pengguna dapat dengan mudah

mengelola tata letak dan desain tanpa harus menulis CSS secara manual (Musyaffa et al., 2024).

Pendekatan berbasis *utility* ini memungkinkan pengembang untuk membangun antarmuka pengguna yang sepenuhnya kustom tanpa harus menulis CSS dari awal. Misalnya, jika ingin membuat sebuah tombol dengan latar belakang biru, teks putih, dan padding tertentu, cukup menambahkan kelas-kelas Tailwind seperti `bg-blue-500`, `text-white`, dan `p-4` langsung ke elemen HTML. Hal ini membuat proses pengembangan menjadi cepat dan efisien karena tidak perlu bolak-balik antara file HTML dan CSS.

Selain itu, Tailwind juga sangat fleksibel dan dapat diubahsuaikan sesuai kebutuhan proyek. Anda dapat menyesuaikan skala warna, ukuran font, atau bahkan membuat utility classes baru yang sesuai dengan kebutuhan spesifik proyek Anda melalui file konfigurasi Tailwind.

Salah satu keunggulan lain dari Tailwind adalah kemampuannya dalam membantu menciptakan desain responsif dengan mudah. Dengan utility classes yang telah disediakan, pengembang dapat dengan cepat mengatur tampilan untuk berbagai ukuran layar, dari mobile hingga desktop, menggunakan kelas-kelas seperti `md:text-lg` atau `lg:px-8`.

Selain itu, Tailwind mendukung *purging*, yaitu proses yang secara otomatis menghapus kelas-kelas CSS yang tidak digunakan saat aplikasi di-*deploy*, sehingga ukuran file CSS yang digunakan di produksi menjadi lebih kecil dan aplikasi menjadi lebih cepat.

Karena fleksibilitas dan pendekatannya yang modern, Tailwind CSS telah menjadi pilihan populer bagi banyak pengembang web yang mencari cara cepat dan efisien untuk membangun antarmuka pengguna yang indah dan responsif tanpa harus terjebak dalam kerumitan CSS tradisional.

2.3.3.8 Visual Studio Code

Visual Studio Code (VS Code) adalah editor *open source* yang dikembangkan oleh Microsoft, dirancang untuk menjadi ringan namun kuat. Meskipun VS Code terlihat sederhana, ia menawarkan berbagai fitur canggih yang membuatnya sangat populer di kalangan pengembang.

Salah satu keunggulan utama VS Code adalah dukungan *multibahasa pemrograman*. Dengan ekstensi yang tepat dapat menggunakan VS Code untuk menulis, mengedit, dan men-debug kode dalam berbagai bahasa seperti JavaScript, Python, Java, C++, dan banyak lagi. Ekstensi ini dapat dengan mudah diunduh dan diinstal dari marketplace bawaan, memungkinkan untuk menyesuaikan editor sesuai dengan kebutuhan spesifik proyek.

VS Code juga memiliki fitur *IntelliSense*, yang tidak hanya menawarkan pelengkapan otomatis untuk kode Anda, tetapi juga menyediakan informasi kontekstual, seperti tipe data atau definisi fungsi, yang membantu meningkatkan produktivitas dan mengurangi kesalahan. Fitur ini mendukung banyak bahasa dan dapat disesuaikan untuk memenuhi kebutuhan pengembangan.

Selain itu, VS Code terintegrasi dengan Git, memungkinkan untuk mengelola kontrol versi langsung dari editor. Anda dapat melakukan commit, push, pull, melihat perbedaan antara versi file, dan menyelesaikan konflik merge tanpa harus meninggalkan editor. Ini membuat pengelolaan proyek kolaboratif menjadi lebih efisien.

Kemampuan *debugging* juga menjadi fitur penting VS Code. Dengan alat debugging bawaan, pengguna dapat menjalankan kode pengguna, menempatkan breakpoint, dan menelusuri proses eksekusi untuk menemukan dan memperbaiki bug dengan cepat. Ini bekerja secara langsung dengan berbagai lingkungan runtime dan bahasa, menjadikan VS Code alat yang serba bisa.

VS Code juga menawarkan *terminal terintegrasi*, yang memungkinkan pengguna menjalankan perintah shell tanpa meninggalkan editor. Terminal ini

mendukung berbagai shell, termasuk Bash dan PowerShell, dan dapat digunakan untuk mengelola proyek, menjalankan skrip, atau bahkan mengelola server.

Karena kemudahan penggunaan, fleksibilitas, dan komunitas yang kuat yang terus mengembangkan ekstensi dan fitur baru, VS Code telah menjadi salah satu editor kode paling populer dan banyak digunakan oleh pengembang di seluruh dunia. VS Code dirancang dengan sangat baik secara keseluruhan, dan keunggulan utamanya adalah dukungan untuk arsitektur berbasis ekstensi (Setiawan, 2022).

BAB III

METODOLOGI PENELITIAN

3.1 Metode Penelitian

3.1.1 Metode Pengumpulan Data

Dalam penelitian ini, metode pengumpulan data yang digunakan meliputi studi pustaka, wawancara, dan observasi. Ketiga metode ini akan memberikan kontribusi yang beragam dalam mendapatkan informasi yang diperlukan untuk penelitian.

3.1.1.1 Studi Pustaka

Studi pustaka merupakan metode yang penting dalam proses penelitian ini. Melalui studi pustaka, peneliti akan mengumpulkan informasi dari sumber-sumber yang relevan, seperti buku dan jurnal ilmiah. Dalam konteks pengembangan media pembelajaran pengolahan citra digital menggunakan visualisasi interaktif berbasis web, studi pustaka akan memberikan pemahaman yang mendalam tentang konsep pengolahan citra digital, prinsip-prinsip desain pembelajaran, dan teknologi terkait pengembangan aplikasi web.

3.1.1.2 Wawancara

Metode wawancara akan digunakan untuk mendapatkan perspektif dan informasi dari para ahli dalam bidang pengolahan citra digital. Wawancara akan dilakukan dengan para akademisi yang memiliki pengetahuan dan pengalaman dalam pengolahan citra digital. Pertanyaan-pertanyaan terkait dengan konsep pengolahan citra digital, penggunaan visualisasi interaktif dalam pembelajaran, tantangan yang dihadapi dalam pengembangan media pembelajaran, dan saran atau rekomendasi akan diajukan kepada responden.

3.1.1.3 Observasi

Metode observasi dilakukan dengan mengamati penggunaan media pembelajaran yang ada atau situasi pembelajaran yang terkait dengan pengolahan citra digital. Observasi dilakukan secara langsung untuk melihat bagaimana mahasiswa berinteraksi dengan materi pembelajaran, sejauh mana mereka memahami konsep-konsep pengolahan citra digital, serta kendala atau masalah yang mungkin timbul selama proses pembelajaran. Observasi ini dilakukan dalam lingkungan kelas atau laboratorium komputer.

3.1.2 Metode Pengembangan Aplikasi

Metode pengembangan aplikasi yang akan digunakan dalam penelitian ini adalah metode waterfall. Metode waterfall adalah salah satu pendekatan pengembangan perangkat lunak yang terstruktur dan linear, di mana setiap fase pengembangan dilakukan secara berurutan dan tidak ada overlap antar fase. Berikut adalah tahapan-tahapan pengembangan aplikasi menggunakan metode waterfall:

3.1.2.1 Analisis Kebutuhan

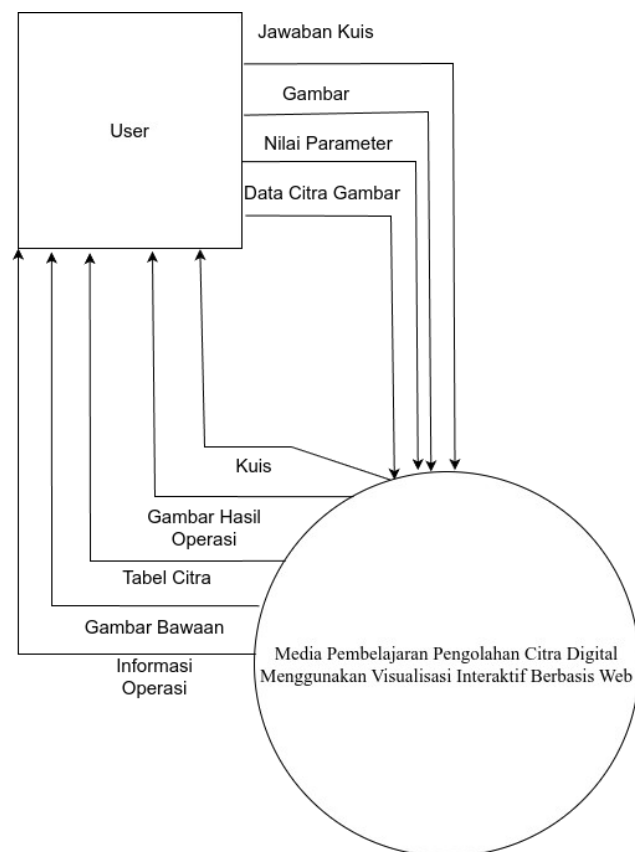
Tahap analisis kebutuhan dilakukan untuk memahami secara mendalam kebutuhan pengguna dan tujuan dari pengembangan aplikasi. Pada tahap ini, dilakukan identifikasi kebutuhan pengolahan citra digital yang perlu disajikan dalam media pembelajaran berbasis web. Analisis kebutuhan ini melibatkan wawancara dengan para pengguna potensial dan pengumpulan informasi terkait konsep-konsep pengolahan citra digital yang penting.

3.1.2.2 Perancangan

Setelah kebutuhan dianalisis, tahap perancangan dilakukan untuk merancang struktur dan fungsionalitas media pembelajaran. Pada tahap ini, dilakukan perancangan antarmuka pengguna, aliran informasi, dan navigasi aplikasi. Perancangan ini juga melibatkan pemilihan algoritma pengolahan citra yang akan digunakan dalam media pembelajaran.

1. Context Diagram

Context diagram adalah sebuah diagram yang digunakan untuk menggambarkan sistem secara keseluruhan serta hubungannya dengan entitas eksternal yang berinteraksi dengan sistem tersebut. Diagram ini memberikan pandangan menyeluruh mengenai sistem tanpa menyelidiki detail internalnya. Pada context diagram, sistem digambarkan dengan sebuah lingkaran atau persegi panjang di tengah diagram, sementara entitas eksternal digambarkan dengan bentuk yang berbeda di sekitarnya. Hubungan antara sistem dan entitas eksternal ditunjukkan dengan garis yang menghubungkan mereka, menggambarkan aliran data atau informasi.

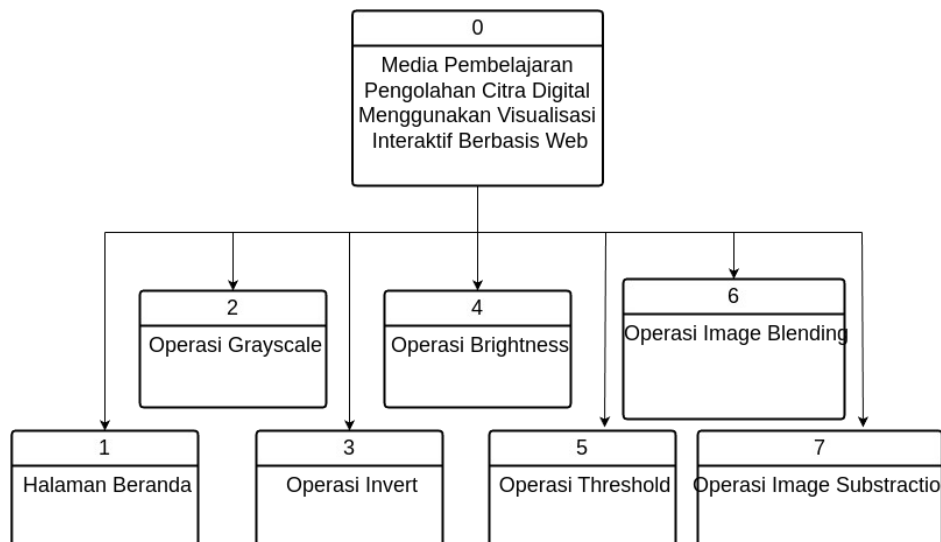


Gambar 3.1 Context Diagram Media Pembelajaran Pengolahan Citra Menggunakan Visualisasi Interaktif Berbasis Web

Context diagram sangat berguna pada tahap awal perancangan sistem karena membantu para pemangku kepentingan memahami batasan dan lingkup sistem. Diagram ini juga memudahkan identifikasi entitas eksternal yang berinteraksi dengan sistem, seperti pengguna, perangkat lunak lain, atau sistem fisik. Dengan demikian, context diagram membantu menentukan kebutuhan fungsional dan non-fungsional sistem, serta menjadikan dasar untuk perancangan lebih detail. Adapun *context diagram* pada aplikasi media pembelajaran pengolahan citra digital berbasis web bisa dilihat pada gambar 3.1.

2. Hierarchy Chart

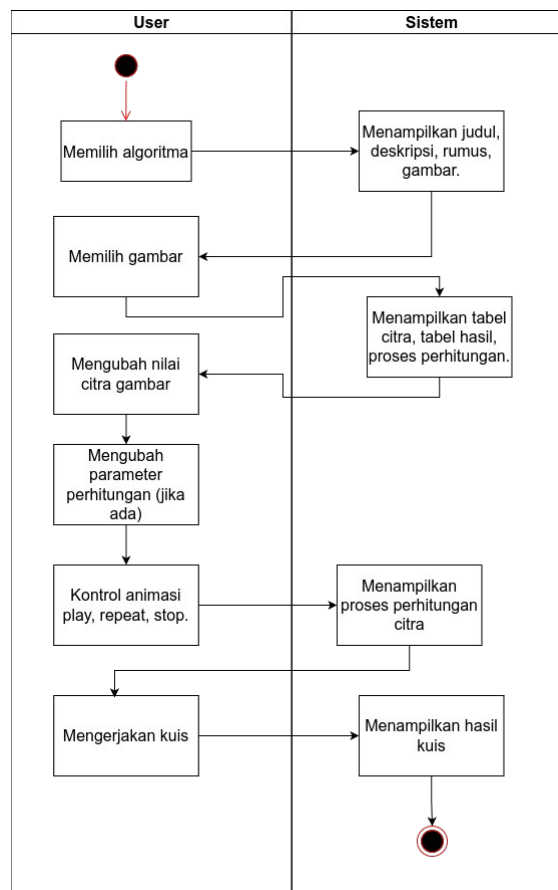
Hierarchy chart adalah sebuah alat visual yang digunakan untuk menggambarkan struktur hirarkis suatu sistem atau proses, menampilkan hubungan antara komponen atau modul dalam bentuk hierarki. Diagram ini sering digunakan dalam pengembangan perangkat lunak, perencanaan organisasi, dan manajemen proyek untuk memberikan pemahaman yang jelas mengenai bagaimana berbagai elemen berinteraksi dan berhubungan satu sama lain. *Hierarchy chart* yang akan dibangun bisa dilihat pada gambar 3.2.



Gambar 3.2 Hierarchy Chart

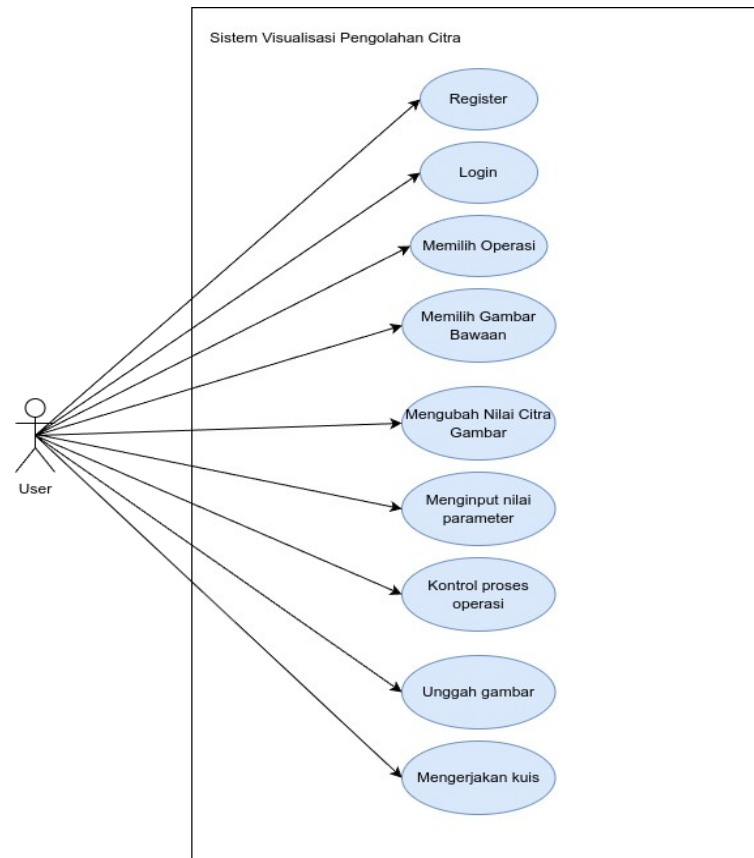
3. Activity Diagram

Activity diagram adalah salah satu jenis diagram dalam *Unified Modeling Language (UML)* yang sangat berguna dalam memodelkan alur kerja atau proses bisnis yang terjadi di dalam sebuah sistem. Diagram ini memberikan representasi visual yang jelas dari berbagai aktivitas serta urutan tindakan yang berlangsung dalam sebuah proses, memperlihatkan bagaimana satu aktivitas mengarah dan bertransisi ke aktivitas lainnya. Hal ini, seperti yang dapat dilihat pada gambar 3.3, membantu dalam memahami serta menganalisis alur kerja secara lebih mendalam dan terstruktur.



Gambar 3.3 Activity Diagram Media Pembelajaran
Pengolahan Citra Menggunakan Visualisasi Interaktif
Berbasis Web

4. Use Case Diagram

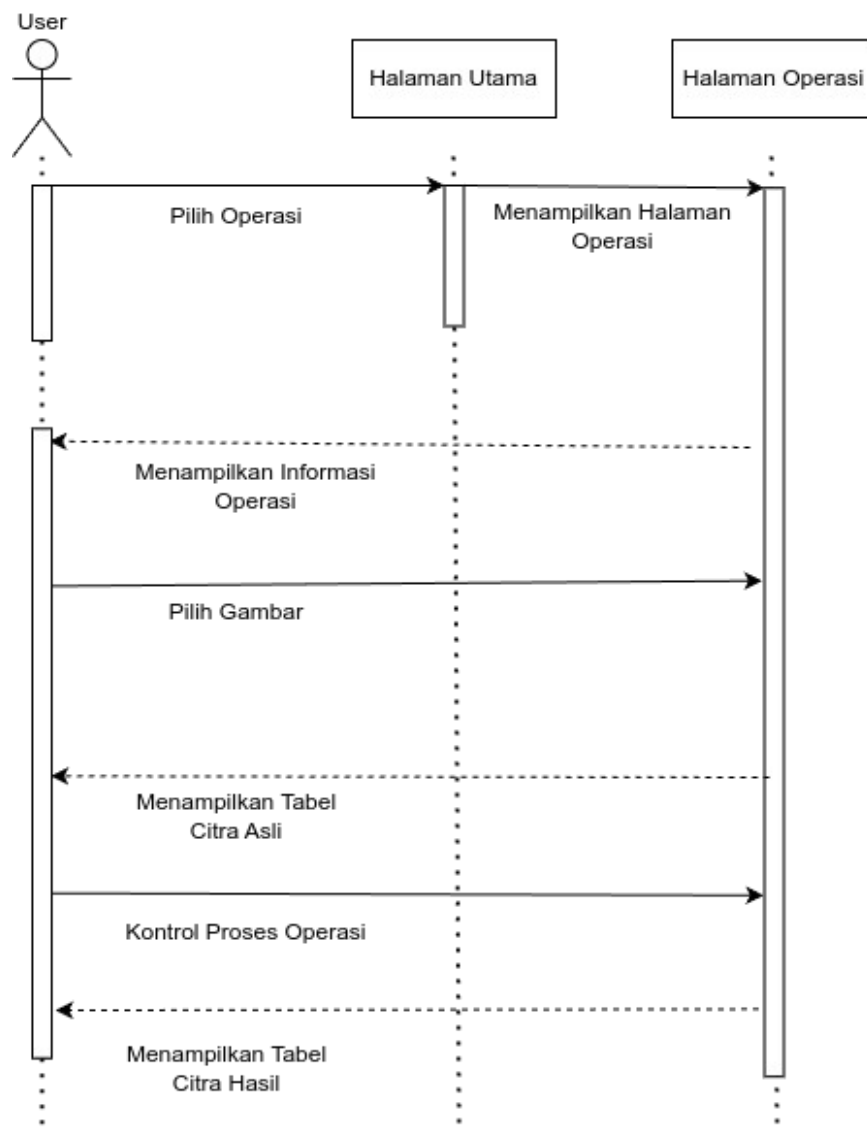


Gambar 3.4. *Use Case Diagram* Media Pembelajaran Visualisasi Pengolahan Citra

Use case diagram adalah representasi grafis yang menggambarkan interaksi antara pengguna (aktor) dan sistem yang sedang dikembangkan. Diagram ini menunjukkan bagaimana aktor berinteraksi dengan fungsi-fungsi utama sistem. Tujuan dari *use case diagram* adalah untuk menjelaskan kebutuhan fungsional sistem dari sudut pandang pengguna, sehingga membantu memahami bagaimana sistem bekerja dan memastikan semua kebutuhan pengguna teridentifikasi.

Berikut adalah *use case diagram* pada sistem media pembelajaran yang dapat dilihat pada gambar 3.4.

5. Sequence Diagram



Gambar 3.5: *Sequence Diagram* Media Pembelajaran Visualisasi Pengolahan Citra

Sequence diagram adalah jenis diagram interaksi dalam *Unified Modeling Language* (UML) yang menggambarkan bagaimana objek berkomunikasi satu

sama lain dalam urutan waktu tertentu. Diagram ini menunjukkan aliran pesan antara berbagai objek untuk menyelesaikan suatu fungsi atau skenario.

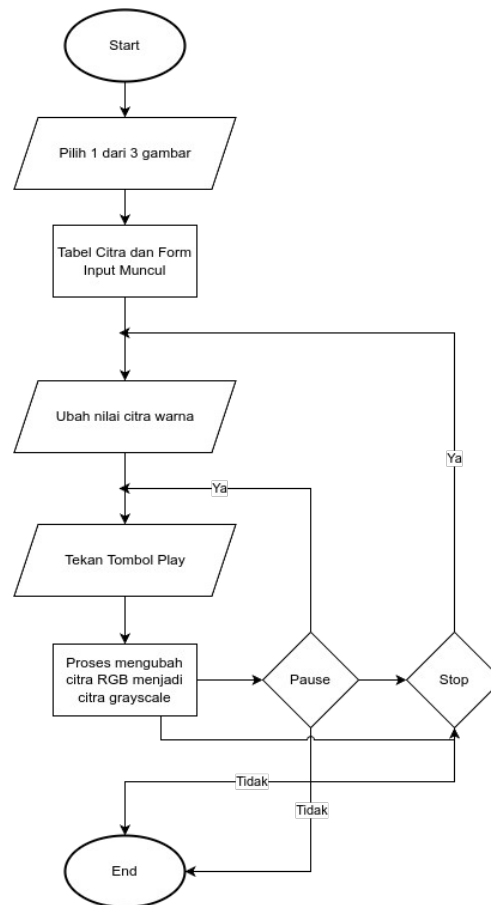
Sequence diagram digunakan untuk memvisualisasikan bagaimana komponen dalam sistem berinteraksi secara dinamis, terutama dalam skenario yang melibatkan banyak langkah dan pesan antar objek. Berikut adalah *sequence diagram* media pembelajaran pengolahan citra yang dapat dilihat pada gambar 3.5.

6. Program Flowchart

Berikut adalah *flowchart* yang menunjukkan sistem media pembelajaran pengolahan citra digital berbasis web yang memperlihatkan alur atau langkah-langkah yang terjadi dalam sistem tersebut. Flowchart tersebut merupakan gambaran visual tentang bagaimana sistem media pembelajaran tersebut beroperasi.

a. Program *Flowchart* Operasi *Grayscale*

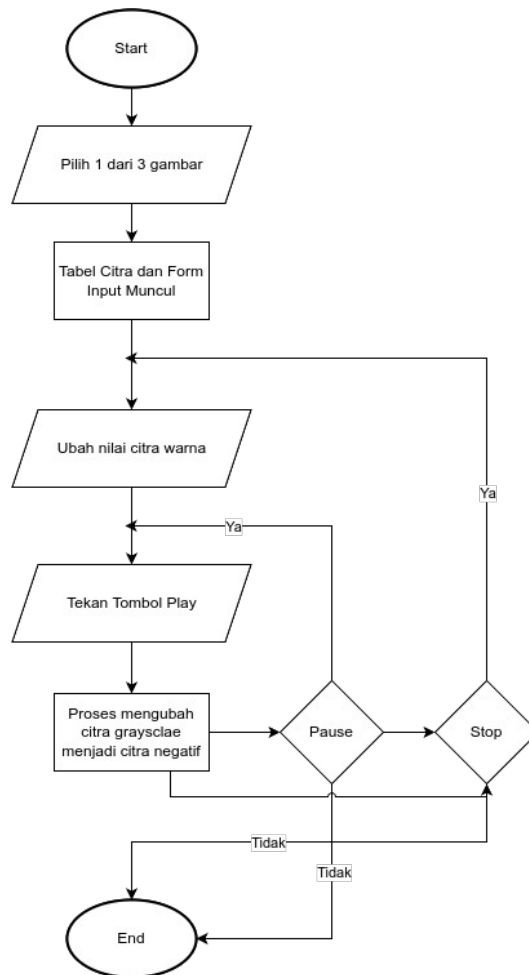
Flowchart Operasi *Grayscale* adalah sebuah gambaran langkah demi langkah saat pengguna melakukan memulai proses mengubah tabel citra warna menjadi citra grayscale. Pada menu grayscale pengguna diminta memilih salah satu dari gambar kemudian pengguna bisa mengubah nilai dari citra warna tersebut. Jika pengguna tidak mengubah nilai dari citra warna pengguna bisa menekan tombol *start*, *pause*, *stop* dan *repeat*.



Gambar 3.6 Flowchart Grayscale

b. Program Flowchart Operasi Invert

Flowchart operasi *invert* atau negatif adalah sebuah gambaran langkah demi langkah saat pengguna melakukan memulai proses mengubah tabel citra grayscale menjadi citra negatif. Pada menu *invert* pengguna diminta memilih salah satu dari gambar kemudian pengguna bisa mengubah nilai dari citra warna tersebut. Jika pengguna tidak mengubah nilai dari citra warna pengguna bisa menekan tombol *start*, *pause*, *stop* dan *repeat*.

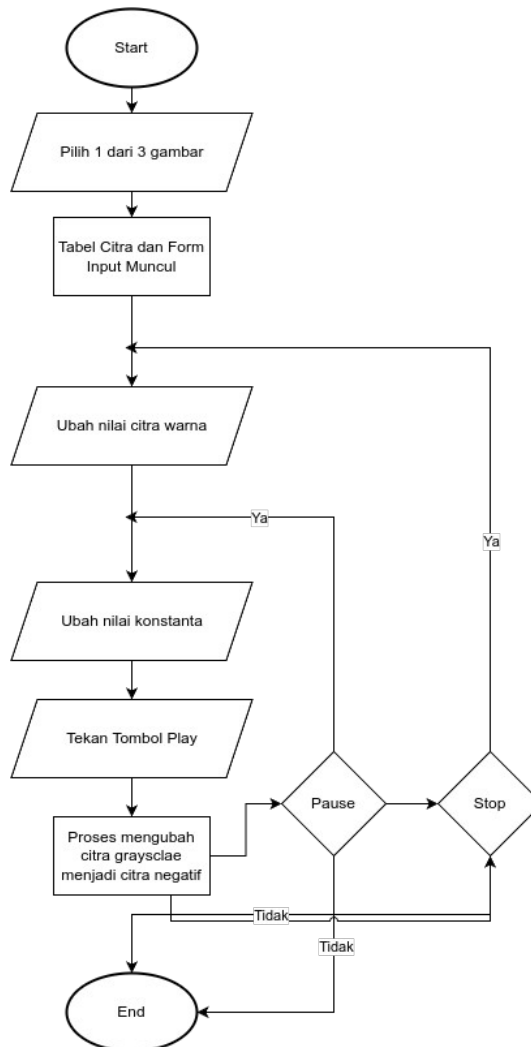


Gambar 3.7 Flowchart Invert

c. Program Flowchart Operasi Brightness

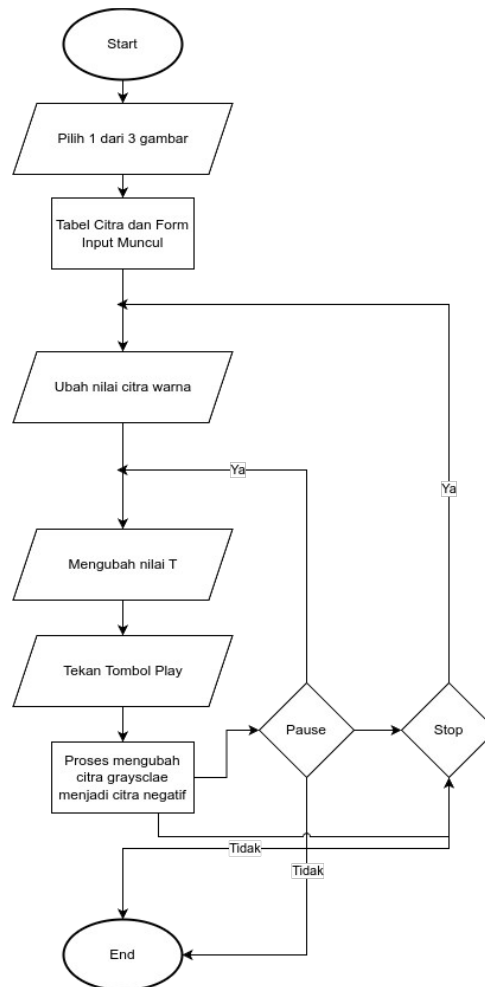
Flowchart Operasi *Brightness* adalah sebuah gambaran langkah demi langkah saat pengguna melakukan memulai proses mengubah tabel citra grayscale menjadi citra dengan nilai baru yaitu peningkatan kecerahan. Pada menu *brightness* pengguna diminta memilih salah satu dari gambar kemudian pengguna bisa mengubah nilai dari citra grayscale tersebut. Jika pengguna tidak mengubah nilai dari citra warna pengguna bisa menekan tombol *start*, *pause*, *stop* dan *repeat*.

Pengguna juga bisa mengatur berapa besar konstanta atau parameter yang akan ditingkatkan menggunakan *input number*.



Gambar 3.8 Flowchart Brightness

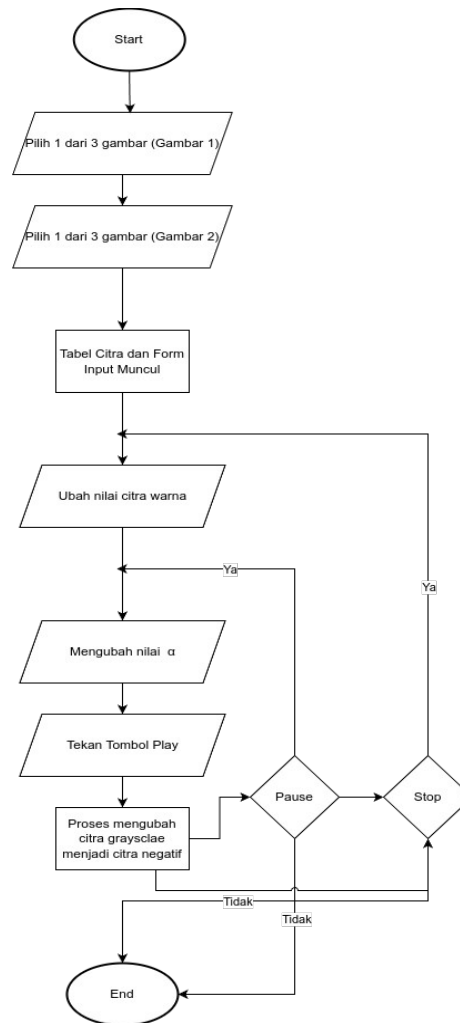
d. Program Flowchart Operasi *Threshold*



Gambar 3.9 Flowchart Threshold

Flowchart operasi *threshold* adalah sebuah gambaran langkah demi langkah saat pengguna melakukan memulai proses mengubah tabel citra grayscale menjadi citra dengan antara 0 atau 255. Pada menu *threshold* pengguna diminta memilih salah satu dari gambar kemudian pengguna bisa mengubah nilai dari citra grayscale tersebut. Jika pengguna tidak mengubah nilai dari citra warna pengguna bisa menekan tombol *start*, *pause*, *stop* dan *repeat*. Pengguna juga bisa mengatur berapa besar nilai T yang akan ditingkatkan menggunakan *input number*.

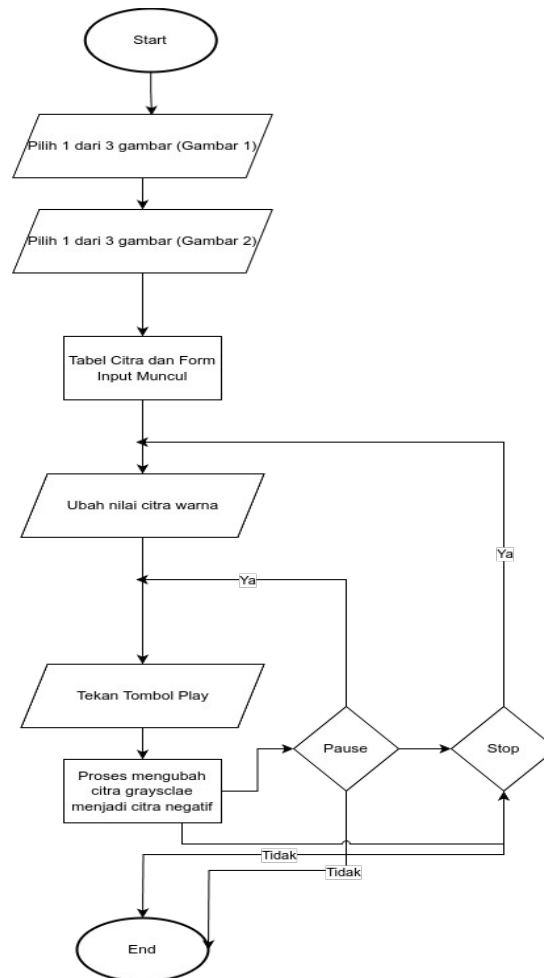
e. Program Flowchart Operasi *Image Blending*



Gambar 3.10 Flowchart *Image Blending*

Flowchart Operasi *Image Blending* adalah sebuah gambaran langkah demi langkah saat pengguna melakukan memulai proses penjumlahan antara 2 citra grayscale. Pada menu *image blending* pengguna diminta memilih 2 gambar kemudian pengguna bisa mengubah nilai dari citra grayscale tersebut. Jika pengguna tidak mengubah nilai dari citra warna pengguna bisa menekan tombol *start*, *pause*, *stop* dan *repeat*. Pengguna juga bisa mengatur berapa besar nilai wA antara 0.1 sampai dengan 0.9 menggunakan input number.

f. Program Flowchart Operasi Image Substraction



Gambar 3.11 Flowchart *Image Substraction*

Flowchart Operasi *Image Substraction* atau pengurangan citra adalah sebuah gambaran langkah demi langkah saat pengguna melakukan memulai proses pengurangan antara 2 citra grayscale. Pada menu *image blending* pengguna diminta memilih 2 gambar kemudian pengguna bisa mengubah nilai dari citra grayscale tersebut. Jika pengguna tidak mengubah nilai dari citra warna pengguna bisa menekan tombol *start*, *pause*, *stop* dan *repeat*.

7. Desain Input

Desain input mengacu pada pengembangan antarmuka sistem yang melibatkan form input untuk pemrosesan operasi dan informasi yang inputkan akan mempengaruhi pada proses operasi itu sendiri.

a. Desain input operasi *grayscale*

Halaman ini adalah visualisasi pengolahan citra digital yang bertujuan mengubah gambar berwarna RGB menjadi grayscale. Terdapat beberapa elemen utama di halaman ini, mulai dari judul “Operasi RGB ke Grayscale” hingga deskripsi singkat tentang konversi warna.

Atau ubah nilai pada field dibawah

```
[[[168,160,113],[255,255,202],[255,255,180],[255,255,198],[154,147,128]],
[[174,166,119],[252,247,191],[247,245,171],[248,248,186],[142,135,116]],
[[247,239,193],[237,232,176],[246,244,170],[228,228,166],[252,245,227]],
[[233,224,181],[189,184,129],[220,217,146],[180,180,120],[243,236,220]],
[[255,248,207],[229,223,171],[244,241,172],[229,228,171],[238,230,217]]]
```

Table Citra RGB

R: 168 G: 160 B: 113	R: 255 G: 255 B: 202	R: 255 G: 255 B: 180	R: 255 G: 255 B: 198	R: 154 G: 147 B: 128
R: 174 G: 166 B: 119	R: 252 G: 247 B: 191	R: 247 G: 245 B: 171	R: 248 G: 248 B: 186	R: 142 G: 135 B: 116
R: 247 G: 239 B: 193	R: 237 G: 232 B: 176	R: 246 G: 244 B: 170	R: 228 G: 228 B: 166	R: 252 G: 245 B: 227
R: 233 G: 224 B: 181	R: 189 G: 184 B: 129	R: 220 G: 217 B: 146	R: 180 G: 180 B: 120	R: 243 G: 236 B: 220
R: 255 G: 248 B: 207	R: 229 G: 223 B: 171	R: 244 G: 241 B: 172	R: 229 G: 228 B: 171	R: 238 G: 230 B: 217

Table Citra Hasil

Gambar 3.12 Desain Input *Grayscale*

Pengguna dapat memilih salah satu dari tiga gambar untuk diproses. Setelah memilih gambar, tabel RGB muncul dengan nilai awal, dan pengguna dapat mengubah nilai RGB melalui input *text area*.

b. Desain input operasi *invert*

[[147,237,230,236,143],[153,230,221,227,131],[226,215,220,207,241], [213,167,194,160,233],[237,208,219,209,228]]

Table Citra Grayscale

147	237	230	236	143
153	230	221	227	131
226	215	220	207	241
213	167	194	160	233
237	208	219	209	228

Table Citra Hasil Negatif

Gambar 3.13 Desain Input *Invert*

Halaman ini adalah visualisasi pengolahan citra digital yang bertujuan mengubah gambar grayscale menjadi negatif. Terdapat beberapa elemen utama di halaman ini, mulai dari judul “Operasi Negatif”, deskripsi singkat tentang konversi negatif, contoh kasus, tabel citra *grayscale* serta tabel citra hasil.

Pengguna dapat memilih salah satu dari tiga gambar *grayscale* untuk diproses. Setelah memilih gambar, tabel *grayscale* muncul dengan nilai awal, dan pengguna dapat mengubah nilai *grayscale* melalui input *text area*.

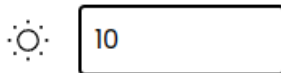
c. Desain input operasi *Brightness*

Untuk halaman ini yang bertujuan untuk mengubah kecerahan dari sebuah gambar. Terdapat beberapa element mulai dari judul, deskripsi singkat, rumus, tabel citra awal table citra hasil.

Pengguna dapat memilih salah satu dari tiga gambar *grayscale* untuk diproses. Setelah memilih gambar, tabel *grayscale* muncul dengan nilai awal, dan pengguna dapat mengubah nilai *grayscale* melalui input text area. Pengguna juga dapat mengubah nilai konstanta menggunakan *input number*.

```
[[147,237,230,236,143],[153,230,221,227,131],
[226,215,220,207,241],[213,167,194,160,233],
[237,208,219,209,228]]
```

Anda juga bisa mengubah nilai parameter yang akan digunakan dalam proses perhitungan operasi brightness menggunakan input di bawah.



Gambar 3.14 Desain Input Operasi *Brightness*

d. Desain input operasi Threshold

Pada halaman ini yang bertujuan untuk melakukan proses operasi *threshold* atau ambang batas dari sebuah gambar. Terdapat beberapa element mulai dari judul, deskripsi singkat, rumus, tabel citra awal table citra hasil.

Pengguna dapat memilih salah satu dari tiga gambar *grayscale* untuk diproses. Setelah memilih gambar, tabel *grayscale* muncul dengan nilai awal, dan pengguna dapat mengubah nilai *grayscale* melalui input text area. Pengguna juga dapat mengubah nilai T menggunakan *input number*.

e. Desain input operasi *Image Blending*

<pre>[[166,147,158,153,136],[189,168,154,138,124],[164,178,178,157,147], [171,173,177,146,171],[183,178,175,137,159]]</pre>
<pre>[[184,185,172,181,159],[170,182,168,160,150],[181,169,159,162,160], [150,165,160,157,163],[172,168,162,159,161]]</pre>

wA

Table Citra Grayscale 1

166	147	158	153	136
189	168	154	138	124
164	178	178	157	147
171	173	177	146	171
183	178	175	137	159

Table Citra Grayscale 2

184	185	172	181	159
170	182	168	160	150
181	169	159	162	160
150	165	160	157	163
172	168	162	159	161

Gambar 3.15 Desain Input Operasi *Image Blending*

Halaman ini adalah visualisasi pengolahan citra digital yang bertujuan menggabungkan 2 buah citra. Terdapat beberapa elemen utama di halaman ini, mulai dari judul “Operasi Penjumlahan Citra”, deskripsi singkat tentang *image blending*, contoh kasus, 2 tabel citra *grayscale* serta tabel citra hasil.

Pengguna dapat memilih dua gambar *grayscale* untuk diproses. Setelah memilih gambar, tabel *grayscale* muncul dengan nilai awal, dan pengguna dapat mengubah nilai *grayscale* melalui dua *input text area*. Pengguna juga dapat menginput nilai *threshold* dari angka 1 – 255.

f. Desain input operasi *Image Substraction*

[[166,147,158,153,136],[189,168,154,138,124],[164,178,178,157,147], [171,173,177,146,171],[183,178,175,137,159]]
[[107,140,132,169,157],[106,108,120,157,122],[127,152,137,123,120], [129,134,145,133,118],[136,138,160,151,124]]

Table Citra Grayscale 1

166	147	158	153	136
189	168	154	138	124
164	178	178	157	147
171	173	177	146	171
183	178	175	137	159

Table Citra Grayscale 2

107	140	132	169	157
106	108	120	157	122
127	152	137	123	120
129	134	145	133	118
136	138	160	151	124

Table Citra Hasil

Table Citra Normalisasi

Gambar 3.16 Desain Input Operasi Image Substraction

Halaman ini adalah visualisasi pengolahan citra digital yang bertujuan mengurangi dua buah citra. Terdapat beberapa elemen utama di halaman ini, mulai dari judul “*Operasi Image Substraction*”, deskripsi singkat tentang *image subtraction*, contoh kasus, dua tabel citra *grayscale* serta tabel citra hasil. Pengguna dapat memilih dua gambar *grayscale* untuk diproses. Setelah memilih gambar, tabel *grayscale* muncul dengan nilai awal, dan pengguna dapat mengubah nilai *grayscale* melalui dua input *text area*.

3.1.2.3 Implementasi

Tahap implementasi melibatkan pembuatan kode program berdasarkan perancangan yang telah disusun sebelumnya. Pada tahap ini, dilakukan

pengembangan aplikasi berbasis web yang mendukung pengolahan citra digital dan visualisasi interaktif. Web yang dibuat menggunakan HTML, CSS, dan JavaScript, dengan *React* dan *Tailwind CSS* sebagai library tambahannya. Algoritma pengolahan citra yang dipilih akan diimplementasikan dalam kode program yang sesuai.

3.1.2.4 Pengujian

Setelah tahap implementasi, pengujian aplikasi dilakukan untuk memastikan bahwa aplikasi berfungsi dengan baik dan memenuhi kebutuhan pengguna. Pengujian ini melibatkan 30 responden yang terdiri dari mahasiswa dan masyarakat umum, yang diminta menjawab 5 pertanyaan terkait pengalaman pengguna (*user experience*) dalam menggunakan aplikasi tersebut. Hasil kuisioner akan dianalisis menggunakan metode skala *Likert*.

3.1.2.5 Pemeliharaan

Setelah aplikasi diimplementasikan dan diuji, langkah selanjutnya adalah pemeliharaan. Tahap pemeliharaan melibatkan pemantauan kinerja aplikasi secara berkala dan memperbaiki masalah yang muncul setelah aplikasi berada dalam lingkungan produksi. Pemeliharaan juga mencakup penambahan fitur baru atau perubahan fungsionalitas sesuai dengan kebutuhan yang berkembang dari pengguna.

Metode waterfall memberikan pendekatan yang terstruktur dan terurut dalam pengembangan aplikasi. Setiap tahap harus diselesaikan dengan baik sebelum melanjutkan ke tahap berikutnya. Pendekatan ini memastikan bahwa setiap fase pengembangan dapat diselesaikan dengan baik sebelum melanjutkan ke fase berikutnya, sehingga mengurangi risiko perubahan yang tidak terduga.

3.1.3 Alat dan Bahan Penelitian

3.1.3.1 Spesifikasi Perangkat Keras (*Hardware*)

Perangkat keras (*hardware*) yang digunakan dalam pembangunan aplikasi media pembelajaran yang dibuat adalah sebagai berikut:

1. Prosesor AMD Ryzen 7 4700U with Radeon Graphics 2.00 GHz
2. SSD 512 GB
3. RAM 8 GB

3.1.3.2 Spesifikasi Perangkat Lunak (*Software*)

Perangkat Lunak (*software*) yang digunakan dalam pembangunan aplikasi lowongan kerja yang dibangun adalah sebagai berikut:

1. Sistem Operasi : Linux Mint
2. Bahasa Pemrograman : Javascript
3. Library : ReactJS, TailwindCSS
4. Aplikasi desain logika program : draw.io
5. Code Editor : Visual Studio Code

DAFTAR PUSTAKA

- Armansyah, F., Sulton, S., & Sulthoni, S. (2019). Multimedia interaktif sebagai media visualisasi dasar-dasar animasi. *Jurnal Kajian Teknologi Pendidikan*, 2(3), 224–229.
- Arsyad, A., & Rahman, A. (2015). *Media Pembelajaran* (A. Rahman, Ed.). Raja Grafindo Persada.
- Hutahaean, H. D., Waluyo, B. D., & Rais, M. A. (2019). Teknologi Identifikasi Objek Berbasis Drone Menggunakan Algoritma Sift Citra Digital. *Jurnal Teknik Informatika UNIKA Santo Thomas*, 4(2), 202–207.
- Jumadi, J., Yupianti, Y., & Sartika, D. (2021). PENGOLAHAN CITRA DIGITAL UNTUK IDENTIFIKASI OBJEK MENGGUNAKAN METODE HIERARCHICAL AGGLOMERATIVE CLUSTERING. *JST (Jurnal Sains Dan Teknologi)*, 10(2), 148–156.
<https://doi.org/10.23887/jstundiksha.v10i2.33636>
- Kapti, K., Waluyo, S., & Widiati, I. S. (2023). MEDIA PEMBELAJARAN VISUALISASI DAUR HIDUP HEWAN BAGI SISWA KELAS IV SD BERBASIS ANDROID. *TRANSFORMASI*, 19(1).
- Karim, A. (2018). Sistem Informasi Pendataan Penduduk Kelurahan Kampung Masjid Berbasis Web. *Seminar Nasional Sains Dan Teknologi Informasi (SENSASI)*, 1(1).
- Kollo, M., & Boimau, S. (2020). Pengembangan Media Pembelajaran Berbasis Visualisasi Gua Peninggalan Jepang di Kupang Untuk Meningkatkan Kesadaran Sejarah Siswa. *Ciencias: Jurnal Penelitian Dan Pengembangan Pendidikan*, 3(1), 31–42.

- Kumalasari, I., Saputra, A. A., Pakpahan, A. G. S., Kurtubi, A., Amiruddin, A., Fridaniarta, B., Wicaksono, E. Y., Saputra, H., Putra, M. Y. A., & Azahra, R. Y. (2023). PELATIHAN DAN PEMBUATAN WEBSITE MENGGUNAKAN HTML DAN CSS. *Beujroh: Jurnal Pemberdayaan Dan Pengabdian Pada Masyarakat*, 1(1), 119–125.
- Mahardika, F., Purwanto, K. A., & Saputra, D. I. S. (2017). Implementasi Metode Waterfall pada Proses Digitalisasi Citra Analog. *VOLT: Jurnal Ilmiah Pendidikan Teknik Elektro*, 2(1), 63–72.
- Maiyana, E. (2017). Perancangan aplikasi media informasi lowongan kerja perusahaan bagi pencari kerja berbasis web. *Jurnal Sains Dan Informatika: Research of Science and Informatic*, 3(2), 118–125.
- Munantri, N. Z., Sofyan, H., & Florestiyanto, M. Y. (2020). Aplikasi Pengolahan Citra Digital Untuk Identifikasi Umur Pohon. *Telematika: Jurnal Informatika Dan Teknologi Informasi*, 16(2), 97–104.
- Musyaffa, A. I., Zulfa, M. I., & Alim, M. S. (2024). Rancang Bangun Purecompute Platform E-Commerce Untuk Belanja Laptop Berbasis Website. *Jurnal SINTA: Sistem Informasi Dan Teknologi Komputasi*, 1(1), 21–29.
- Noviantoro, A., Silviana, A. B., Fitriani, R. R., & Permatasari, H. P. (2022). Rancangan Dan Implementasi Aplikasi Sewa Lapangan Badminton Wilayah Depok Berbasis Web. *Jurnal Teknik Dan Science*, 1(2), 88–103.
- Nurrita, T. (2018). Pengembangan media pembelajaran untuk meningkatkan hasil belajar siswa. *Jurnal Misykat*, 3(1), 171–187.
- Oktavianto, R. N., Jalil², A., Amsyah³, A., Julianto, B. D., Ramdhani, D. S., Saputra, E. P., Kurnianto, E., Subhan, F., Ardiansyah, M. V., &

- Oktavianto10, R. N. (2022). PENGENALAN SERTA PELATIHAN BAHASA DASAR PEMOGRAMAN WEB (HTML DAN CSS) KEPADA KARANG TARUNA KEL.KEDAUNG KALIANGKE, CENGKARENG, JAKBAR: PENGENALAN SERTA PELATIHAN BAHASA DASAR PEMOGRAMAN WEB (HTML DAN CSS) KEPADA KARANG TARUNA KEL.KEDAUNG KALIANGKE, CENGKARENG, JAKBAR. *AMMA : Jurnal Pengabdian Masyarakat*, 1(01), 8–12.
- Panjaitan, J., & Pakpahan, A. F. (2021). Perancangan Sistem E-Reporting Menggunakan ReactJS dan Firebase. *Jurnal Teknik Informatika Dan Sistem Informasi*, 7(1).
- Pratama, M., & Delianti, V. I. (2021). RANCANG BANGUN APLIKASI PRESENSI DENGAN GLOBAL POTITIONING SYSTEM (GPS) BERBASIS ANDROID (Studi Kasus: PT. Perkebunan Nusantara VI Unit Usaha Kayu Aro). *INCARE, International Journal of Educational Resources*, 2(2), 141–154.
- Pratama, N. F., LuqmanulHakim, H., Nugroho, D. S., Putra, A. A., Saputra, B. A., & Alit, R. (2023). Inovasi Pembaruan Desain Website Penyederhana Link Unesa Menggunakan Typescript Dan Node.js. *TEKTONIK : Jurnal Ilmu Teknik*, 1(2), 35–40. <https://doi.org/10.62017/tektonik.v1i2.202>
- Setiawan, I. (2022). Komparasi Kinerja Integrated Development Environment (IDE) Dalam Mengeksekusi Perintah Python. *SATESI: Jurnal Sains Teknologi Dan Sistem Informasi*, 2(1), 52–59.
- Setiawan, I., Dewanta, W., Nugroho, H., & Supriyono, H. (2019). Pengolah Citra Dengan Metode Thresholding Dengan Matlab R2014A. *JURNAL MEDIA INFOTAMA*, 15(2). <https://doi.org/10.37676/jmi.v15i2.868>

- Suryadibrata, A., & Young, J. C. (2020). Visualisasi Algoritma sebagai Sarana Pembelajaran K-Means Clustering. *Ultimatics: Jurnal Teknik Informatika*, 12(1), 25–29.
- Suryowinoto, A., & Hamid, A. (2017). Penggunaan Pengolahan Citra Digital dengan Algoritma Edge Detection dalam Mengidentifikasi Kerusakan Kontur Jalan. *Semin. Nas. Sains Dan Teknol. Terap.* V, 149–154.
- Wahid, A. A. (2020). Analisis metode waterfall untuk pengembangan sistem informasi. *J. Ilmu-Ilmu Inform. Dan Manaj. STMIK*, No. November, 1(1), 1–5.
- Yasa, I. M. R. P., Widyantera, I. M. O., & Wirastuti, N. D. (2018). PERBAIKAN KONTRAS CITRA MENGGUNAKAN TEKNIK BRIGHTNESS ADJUSTMENT UNTUK SISTEM EKSTRAKSI GARIS PANTAI. *Jurnal SPEKTRUM*, 4(2), 42–47.
<https://doi.org/10.24843/SPEKTRUM.2017.v04.i02.p06>