

**PERANCANGAN APLIKASI WEB BERBASIS AI UNTUK OPTIMASI
KUALITAS GAMBAR FOTOGRAFI MELALUI METODE *FILTERING*
CITRA DIGITAL**

SKRIPSI

DISUSUN OLEH

IBNU PRIBUDIANTO

2109020067



**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA**

MEDAN

2025

**PERANCANGAN APLIKASI WEB BERBASIS AI UNTUK OPTIMASI
KUALITAS GAMBAR FOTOGRAFI MELALUI METODE *FILTERING*
CITRA DIGITAL**

SKRIPSI

**Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer
(S.Kom) dalam Program Studi Teknologi Informasi pada Fakultas Ilmu
Komputer dan Teknologi Informasi, Universitas Muhammadiyah Sumatera Utara**

IBNU PRIBUDIANTO

2109020067

**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA**

MEDAN

2025

LEMBAR PENGESAHAN

Judul Skripsi : PERANCANGAN APLIKASI WEB BERBASIS AI
UNTUK OPTIMASI KUALITAS GAMBAR
FOTOGRAFI MELALUI METODE *FILTERING* CITRA
DIGITAL

Nama Mahasiswa : IBNU PRIBUDIANTO

NPM : 2109020067

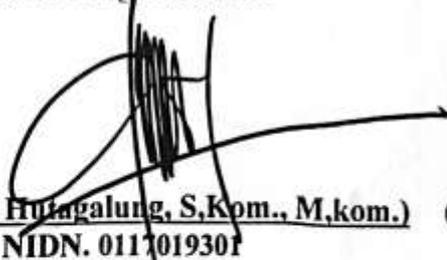
Program Studi : TEKNOLOGI INFORMASI

Menyetujui
Komisi Pembimbing



(Dr. Al-Khowarizmi, S.Kom., M.Kom.)
NIDN. 0127099201

Ketua Program Studi



(Fatma Sari Hutagalung, S.Kom., M.kom.)
NIDN. 0117019301

Dekan



(Dr. Al-Khowarizmi, S.Kom., M.Kom.)
NIDN. 0127099201

PERNYATAAN ORISINALITAS

PERANCANGAN APLIKASI WEB BERBASIS AI UNTUK OPTIMASI KUALITAS GAMBAR FOTOGRAFI MELALUI METODE *FILTERING* CITRA DIGITAL

SKRIPSI

Saya menyatakan bahwa karya tulis ini adalah hasil karya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing disebutkan sumbernya.

Medan, 18 September 2025

Yang membuat pernyataan



Ibnu Priadianto

NPM. 2109020067

**PERNYATAAN PERSETUJUAN PUBLIKASI
KARYA ILMIAH UNTUK KEPENTINGAN
AKADEMIS**

Sebagai sivitas akademika Universitas Muhammadiyah Sumatera Utara, saya bertanda tangan dibawah ini:

Nama : Ibnu Pribudianto
NPM : 2109020067
Program Studi : Teknologi Informasi
Karya Ilmiah : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Muhammadiyah Sumatera Utara Hak Bebas Royalti Non-Eksekutif (*Non-Exclusive Royalty free Right*) atas penelitian skripsi saya yang berjudul:

**PERANCANGAN APLIKASI WEB BERBASIS AI UNTUK OPTIMASI
KUALITAS GAMBAR FOTOGRAFI MELALUI METODE *FILTERING*
CITRA DIGITAL**

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non-Eksekutif ini, Universitas Muhammadiyah Sumatera Utara berhak menyimpan, mengalih media, memformat, mengelola dalam bentuk database, merawat dan mempublikasikan Skripsi saya ini tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis dan sebagai pemegang dan atau sebagai pemilik hak cipta.

Demikian pernyataan ini dibuat dengan sebenarnya.

Medan, 18 September 2025

Yang membuat pernyataan



Ibnu Pribudianto

NPM. 2109020067

RIWAYAT HIDUP

DATA PRIBADI

Nama Lengkap : Ibnu Pribudianto
Tempat dan Tanggal Lahir : Medan, 07 Januari 2004
Alamat Rumah : Jl. Pembinaan Hulu Gg. Sejahtera No. 103
Telepon/Faks/HP : 082119937827
E-mail : ibnupribudianto0701@gmail.com
Instansi Tempat Kerja : -
Alamat Kantor : -

DATA PENDIDIKAN

SD : MIN MEDAN PETISAH TAMAT: 2015
SMP : SMP SWASTA IKAL TAMAT: 2018
SMA : MAN 1 MEDAN TAMAT: 2021

TANGGAL SIDANG SKRIPSI 2 September 2025

KATA PENGANTAR



Alhamdulillah, segala puji dan Syukur penulis panjatkan kehadiran Allah SWT yang telah melimpahkan Rahmat dan karunianya yang penuh dengan ilmu kepada penulis, sehingga penulis dapat menyelesaikan tugas akhir ini yang berjudul tentang “Perancangan Aplikasi Web Berbasis Ai Untuk Optimalisasi Kualitas Gambar Fotografi Melalui Metode Filtering Citra Digital” untuk memenuhi persyaratan dalam jenjang strata satu dan mencapai gelar Sarjana Komputer di jurusan Teknologi Informasi, Fakultas Teknologi Informasi dan Ilmu Komputer, Universitas Muhammadiyah Sumatera Utara. Sholawat serta salam selalu tercurahkan kepada junjungan Nabi besar Muhammad SAW, keluarga dan sahabatnya yang syafaatnya kita nantikan diakhir zaman nanti. Dalam penyusunan Tugas Akhir ini, penulis telah mendapatkan banyak bantuan dan bimbingan dari berbagai pihak. Oleh karena itu pada kesempatan ini penulis tentunya berterimakasih kepada pihak dalam dukungan serta doa dalam penyelesaian skripsi. Penulis juga berterimakasih kepada :Penulis tentunya berterima kasih kepada berbagai pihak dalam dukungan serta doa dalam penyelesaian skripsi. Penulis juga mengucapkan terima kasih kepada:

1. Bapak Prof. Dr. Agussani, M.AP., selaku Rektor Universitas Muhammadiyah Sumatera Utara (UMSU).
2. Bapak Dr. Al-Khowarizmi, S.Kom., M.Kom., selaku Dekan Fakultas Ilmu Komputer dan Teknologi Informasi (FIKTI) UMSU. Juga selaku dosen pembimbing skripsi yang telah meluangkan waktu, dan pikiran untuk memberikan arahan, bimbingan, dan saran yang berharga selama proses penyusunan skripsi ini.

3. Bapak Halim Maulana, S.T., M.Kom., selaku Wakil Dekan I Fakultas Ilmu Komputer dan Teknologi Informasi (FIKTI) UMSU.
4. Bapak Lutfi Basit, S.Sos., M.I.Kom., selaku Wakil Dekan II Fakultas Ilmu Komputer dan Teknologi Informasi (FIKTI) UMSU.
5. Ibu Fatma Sari Hutagalung, S.Kom., M.Kom., selaku Ketua Program Studi Teknologi Informasi.
6. Bapak Mhd. Basri, S.Si., M.Kom., selaku Sekretaris Program Studi Teknologi Informasi.
7. Dosen-dosen Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Muhammadiyah Sumatera Utara, atas ilmu yang diberikan selama masa perkuliahan.
8. Seluruh staf akademik Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Muhammadiyah Sumatera Utara, atas fasilitas dan dukungan administratif yang diberikan selama masa perkuliahan hingga penyusunan skripsi.
9. Kedua orang tua tercinta Bapak Suprianto dan Ibu Dewi Tirtasari yang selalu memberikan doa, dukungan, dan semangat dalam segala keadaan.
10. Kedua abang penulis dan juga adik penulis yang telah memberikan dukungan dan doa kepada penulis.
11. Seluruh teman-teman KKN yang selalu memberikan motivasi, dukungan dan juga pengalaman yang tidak terlupakan.
12. Kepada Amira, seseorang yang mengembalikan semangat dan pacuan kepada penulis untuk melanjutkan penulisan skripsi ini.

13. Semua pihak yang terlibat, baik secara langsung maupun tidak langsung, dalam penyelesaian skripsi ini, yang mungkin namanya tidak disebutkan satu per satu.

14. Tidak lupa, penulis menyampaikan rasa terima kasih yang tulus kepada diri sendiri atas kesabaran, ketekunan, dan semangat yang terus dijaga selama proses penyusunan skripsi ini.

Penulis menyadari bahwa Tugas Akhir ini masih jauh dari sempurna dikarenakan keterbatasan pengetahuan serta kemampuan yang dimiliki penulis. Oleh karena itu, penulis mengharapkan kritik dan saran yang bersifat membangun untuk menyempurnakan penulisan skripsi ini. Semoga skripsi ini dapat bermanfaat bagi penulis khususnya dan bagi semua yang membutuhkan.

Wassalamu'alaikum warahmatullahi wabarakaatuh

Medan, 18 September 2025
Penulis



Ibnu Pribudianto

PERANCANGAN APLIKASI WEB BERBASIS AI UNTUK OPTIMASI KUALITAS GAMBAR FOTOGRAFI MELALUI METODE *FILTERING* CITRA DIGITAL

ABSTRAK

Kualitas gambar dalam fotografi sering terganggu oleh faktor seperti pencahayaan buruk atau fokus yang tidak tepat, sehingga menghasilkan gambar buram. Penelitian ini bertujuan meningkatkan ketajaman gambar melalui metode filtering citra digital, yaitu Unsharp Masking, High-Pass Filtering, dan Filter Sobel. Ketiga metode ini diuji untuk melihat efektivitasnya dalam memperjelas detail gambar. Penelitian ini juga merancang aplikasi web berbasis AI yang memudahkan pengguna mengedit gambar tanpa perlu keahlian teknis. Metode yang digunakan adalah eksperimen kuantitatif dengan pendekatan deskriptif, serta evaluasi dilakukan menggunakan nilai PSNR, SSIM, dan kuesioner pengguna. Hasil dari penelitian ini menunjukkan bahwa penerapan metode sharpening mampu meningkatkan kualitas citra fotografi secara signifikan, dan integrasi ke dalam platform web memberikan kemudahan akses bagi masyarakat umum. Aplikasi ini diharapkan dapat menjadi solusi praktis bagi fotografer, editor, maupun masyarakat umum dalam memperoleh gambar berkualitas tinggi secara efisien dan cepat.

Kata Kunci: Citra Digital, *Sharpening*, *Unsharp Masking*, *High-Pass Filtering*, *Filter Sobel*, *AI*, *Website*.

PERANCANGAN APLIKASI WEB BERBASIS AI UNTUK OPTIMASI KUALITAS GAMBAR FOTOGRAFI MELALUI METODE *FILTERING* CITRA DIGITAL

ABSTRACT

Image quality in photography is often disrupted by factors such as poor lighting or incorrect focus, resulting in blurry images. This study aims to enhance image sharpness using digital image filtering methods, namely Unsharp Masking, High-Pass Filtering, and Sobel Filter. These methods are tested to evaluate their effectiveness in clarifying image details. The study also develops a web-based application powered by AI to help users edit images without requiring technical skills. A quantitative experimental method with a descriptive approach is used, and evaluation is conducted using PSNR, SSIM, and user questionnaires. The results show that the application of sharpening methods can significantly improve the quality of photographic images, and integration into a web platform provides easy access for the general public. This application is expected to be a practical solution for photographers, editors, and general users to obtain high-quality images efficiently and quickly.

Keywords: Digital Image, Sharpening, Unsharp Masking, High-Pass Filtering, Sobel Filter, AI, Website.

DAFTAR ISI

LEMBAR PENGESAHAN	i
PERNYATAAN ORISINALITAS.....	ii
KATA PENGANTAR.....	v
DAFTAR ISI.....	x
DAFTAR TABEL	xiii
BAB I.....	1
PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
2.2.1 Unsharp Masking	11
2.2.2 High-Pass Filtering	14
2.2.3 Filter Sobel.....	17
2.3 Web	20
2.3.1 HTML	21
2.3.2 CSS	21
2.3.3 Javascript.....	21
2.3.4 MySQL	22
2.3.5 PHP	22
2.4 Artificial Intelligence (AI)	23
BAB III.....	24
METODOLOGI PENELITIAN	24
3.1 Metode Penelitian	24
3.2 Subjek, Objek, Waktu Penelitian	24
3.3 Teknik Pengumpulan Data	25
3.4 Metode Pengolahan Data.....	26
3.5 Alur Penelitian	28
3.6 Pengembangan Website Berbasis AI.....	30
3.6.1 Tujuan Pengembangan Website.....	30
3.6.2 Tools dan Teknologi yang Digunakan	32
3.6.3 Alur Proses Website.....	33
BAB IV	36
HASIL DAN PEMBAHASAN	36
4.1 Tampilan Antarmuka Aplikasi	36
4.1.1 Halaman Landing Page	36
4.1.2 Halaman Login dan Registrasi	37
4.1.3 Halaman Utama Aplikasi (Dashboard)	38
4.1.4 Halaman Riwayat Pemrosesan.....	39

4.1.5	Responsivitas dan Aksesibilitas	40
4.2	Proses Unggah dan Validasi Gambar.....	40
4.2.1	Pemilihan File oleh Pengguna.....	41
4.2.2	Pengecekan Keberadaan File	41
4.2.3	Validasi Ekstensi File.....	42
4.2.4	Validasi Struktur File Gambar	42
4.2.5	Penyimpanan File Secara Aman	43
4.3	Proses Analisis dan Penajaman Gambar Otomatis	43
4.3.1	Perhitungan Ketajaman Gambar (Sharpness Analysis)	44
4.3.2	Klasifikasi Tingkat Ketajaman.....	44
4.3.3	Proses Penajaman dengan Unsharp Masking.....	45
4.3.4	Penanganan Gambar dengan Alpha Channel (Transparansi).....	46
4.3.5	Penyimpanan Gambar Hasil Proses	46
4.4	Tampilan Antarmuka Program dan Fitur Website	46
4.4.1	Halaman Landing Page	47
4.4.2	Fitur Registrasi dan Login.....	47
4.4.3	Halaman Utama (Dashboard).....	48
4.4.4	Halaman Riwayat Proses (History).....	48
4.4.5	Komponen Navigasi dan Logout	49
4.4.6	Penanganan Error dan Validasi.....	49
4.4.7	Responsif dan Kompatibilitas	50
4.5	Evaluasi Hasil Pengolahan Gambar.....	50
4.5.1	Evaluasi Objektif: PSNR dan SSIM	50
4.5.2	Evaluasi Subjektif: Observasi Visual dan Persepsi Pengguna	51
4.5.3	Analisis Perbandingan Hasil	52
4.5.4	Catatan Pengaruh Threshold dan Parameter	53
4.6	Perbandingan Hasil Sebelum dan Sesudah Pengolahan Gambar.....	53
4.6.1	Aspek yang Dibandingkan	53
4.6.2	Contoh Studi Kasus Gambar.....	54
4.6.3	Visualisasi Hasil Perbandingan.....	55
4.6.4	Interpretasi Kuantitatif.....	55
4.7	Dokumentasi Antarmuka Website	56
4.7.1	Tampilan Halaman Utama (Landing Page).....	56
4.7.2	Halaman Login dan Registrasi.....	57
4.7.3	Halaman Utama Aplikasi (Dashboard)	57
4.7.4	Halaman Riwayat Pengguna (History Page).....	58
4.7.5	Antarmuka Responsif dan Kompatibilitas	59
4.7.6	Notifikasi dan Validasi.....	59

4.7.7 Rancangan Alur Navigasi (User Flow).....	59
4.7.8 Keamanan dan Pengelolaan Akses.....	60
BAB V	61
PENUTUP.....	61
5.1 Kesimpulan	61
5.2 Saran	62
DAFTAR PUSTAKA.....	64

DAFTAR TABEL

Tabel 3. 1	25
Tabel 4. 1	55
Tabel 4. 2	56

DAFTAR GAMBAR

Gambar 2. 1 <i>kamera</i>	10
Gambar 2. 2 <i>Sharpening</i>	13
Gambar 2. 3 <i>grayscale</i>	17
Gambar 2. 4 <i>Contoh filter sobel</i>	20
Gambar 3. 1 <i>Flowchart Alur Penelitian</i>	29
Gambar 3. 2 <i>Flowchart Alur Website</i>	35
Gambar 4. 1 <i>Landing Page</i>	37
Gambar 4. 2 <i>Halaman Daftar</i>	38
Gambar 4. 3 <i>Halaman Utama</i>	39
Gambar 4. 4 <i>Halaman Riwayat</i>	40
Gambar 4. 5 <i>Syntax 1</i>	41
Gambar 4. 6 <i>Syntax 2</i>	42
Gambar 4. 7 <i>Syntax 3</i>	42
Gambar 4. 8 <i>Syntax 4</i>	42
Gambar 4. 9 <i>Syntax 5</i>	43
Gambar 4. 10 <i>Syntax 6</i>	43
Gambar 4. 11 <i>Syntax 7</i>	44
Gambar 4. 12 <i>Syntax 8</i>	45
Gambar 4. 13 <i>Syntax 9</i>	45
Gambar 4. 14 <i>Syntax 10</i>	46
Gambar 4. 15 <i>Syntax 11</i>	46
Gambar 4. 16 <i>Syntax 12</i>	49

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Dalam era teknologi modern, fotografi telah menjadi bagian penting dari kehidupan sehari-hari, baik secara pribadi maupun profesional. Fotografer dapat menangkap gambar dengan kualitas tinggi berkat kemajuan teknologi kamera seperti penggunaan sensor beresolusi tinggi dan fitur pemrosesan gambar canggih. Meskipun ada kemajuan, masalah ketajaman gambar tetap menjadi masalah besar. Beberapa hal, seperti pencahayaan yang buruk, gerakan kamera saat pengambilan gambar, atau fokus yang salah, dapat menyebabkan gambar yang tidak tajam atau blur. Masalah ini dapat memengaruhi persepsi visual seseorang dan nilai estetika dari gambar yang dihasilkan. Terkadang gambar yang di ambil pada saat proses foto sudah di anggap mendapatkan hasil yang bagus, tetapi ketika masuk ke software untuk di lakukan pengeditan ternyata gambar tersebut masih memiliki beberapa kekurangan yang harus di lakukan pengeditan.

Citra merupakan salah satu bentuk informasi yang berupa gambar. Sering kali citra mengalami gangguan atau memiliki kualitas yang buruk sehingga sulit di interpretasikan oleh manusia. Untuk mengatasi hal tersebut dilakukan pengolahan citra. Pengolahan citra bertujuan untuk memperoleh kualitas citra yang lebih baik dibandingkan dengan citra awal. Perbaikan citra merupakan tahap awal dari pengolahan citra yang selanjutnya akan digunakan untuk kebutuhan analisis citra. Berbagai metode atau algoritma perbaikan citra telah banyak dikembangkan, salah satunya adalah penajaman citra (image sharpening). Image sharpening diperlukan.

untuk mengatasi citra yang memiliki noise,derau atau kabur sehingga objek dapat diperjelas atau dipertajam.(Helmi Ani Siregar, 2019). diperlukan untuk mengatasi citra yang memiliki noise,derau atau kabur sehingga objek dapat diperjelas atau dipertajam.(Helmi Ani Siregar, 2019).

Metode sharpening, yang merupakan salah satu solusi yang banyak digunakan untuk mengatasi masalah ketajaman, bertujuan untuk meningkatkan kejelasan gambar dengan menonjolkan tepi dan detail halus dalam citra. Metode sharpening yang umum digunakan, seperti unsharp masking filter Sobel, dan Laplacian, memiliki kelebihan dan kekurangan masing-masing. Unsharp masking, misalnya, adalah teknik yang telah lama digunakan dan terbukti berhasil meningkatkan ketajaman gambar. Filter Laplacian dan Sobel, di sisi lain, berkonsentrasi pada penemuan tepi, yang dapat memberikan hasil yang lebih signifikan dalam beberapa situasi.

Seiring dengan kemajuan teknologi, banyak fotografer kini mengadopsi teknik pascaproduksi yang lebih mutakhir, termasuk penggunaan perangkat lunak berbasis kecerdasan buatan (AI) yang mampu meningkatkan kualitas gambar secara otomatis. Namun, pemahaman yang mendalam tentang metode sharpening tradisional tetap krusial, karena tidak semua situasi dapat ditangani dengan baik oleh algoritma otomatis. Penelitian ini bertujuan untuk mengeksplorasi penerapan metode sharpening dalam fotografi modern, di mana teknik ini dapat diintegrasikan dengan alat dan teknologi terbaru.

Meski banyak kajian sebelumnya yang membahas teknik-teknik ini masih terdapat kebutuhan untuk melakukan analisis mendalam mengenai efektivitas masing-masing metode dalam konteks fotografi. Berbagai jenis citra, mulai dari potret

hingga pemandangan, dapat menghasilkan hasil yang bervariasi tergantung pada teknik sharpening yang diterapkan. Oleh karena itu, penting untuk melaksanakan studi kasus yang fokus pada penerapan metode sharpening dalam fotografi modern, agar kita dapat memahami cara mengoptimalkan teknik ini untuk meningkatkan kualitas gambar secara keseluruhan.

Penelitian ini bertujuan untuk menjawab sejumlah pertanyaan penting mengenai efektivitas metode sharpening, bagaimana pengaturan parameter dapat mempengaruhi hasil akhir, serta memberikan panduan praktis bagi fotografer dalam memilih teknik yang sesuai untuk situasi tertentu. Dengan demikian, penelitian ini tidak hanya berkontribusi pada pemahaman akademis mengenai pengolahan citra, tetapi juga memberikan manfaat langsung bagi praktisi fotografi yang ingin meningkatkan kualitas karya mereka.

Melalui analisis yang mendalam, diharapkan penelitian ini dapat membantu fotografer dalam mengatasi masalah ketajaman gambar dan menghasilkan hasil yang berkualitas tinggi. Dengan memanfaatkan teknik sharpening dalam konteks alat dan teknologi modern, fotografer dapat lebih efektif menciptakan citra yang menarik dan bermutu, sehingga meningkatkan kepuasan serta pengalaman visual bagi para pemirsa.

Mengikuti perkembangan zaman, Website sudah menjadi salah satu bukti pada perkembangan zaman yang sangat berguna. Beberapa website memiliki kemampuan tersendiri sesuai kegunaan dari yang ditawarkan pembuat website tersebut. Maka dari itu terbesit keinginan penulis untuk menambahkan website, dikarenakan tidak semua orang bisa menggunakan software yang berkaitan dengan

pengeditan gambar. Sehingga dapat memudahkan orang dalam mengedit foto. Harapannya orang lain bisa terbantu dengan website ini.

Perkembangan pesat teknologi telah membawa era baru dalam kehidupan manusia, yaitu era digital. Semakin berkembangnya teknologi membuat permasalahan sehari-hari terselesaikan lebih mudah dan praktis. Salah satu teknologi yang tercipta dari berkembangnya teknologi ialah artificial intelligence atau sering disebut dengan AI.

AI adalah teknologi kecerdasan yang dimiliki manusia, dibuat dalam bentuk mesin untuk dapat melakukan dan meniru sistem kerja otak manusia, bahkan bisa berpikir seperti manusia dan dapat merespons layaknya manusia (Swendri et al.). Seiring dengan berkembangnya AI, saat ini banyak website atau platform yang menyediakan jasa merealisasikan apa yang ada di pikiran manusia menjadi sebuah gambar digital dengan cepat. Hanya dengan menuliskan isi pikiran dalam bentuk prompt, AI text to image generator akan menciptakan gambaran yang sesuai dengan kata kunci yang diperintahkan dalam hitungan detik. Tentu saja hal ini bisa menjadi hal positif bagi fotografer.

Dalam tugas akhir ini, artificial intelligence (AI) akan di bentuk menjadi website dan bisa dikatakan sebagai layanan dalam membantu orang dalam proses edit foto tanpa harus mempelajari prosesnya terlebih dahulu. Adapun metode yang digunakan untuk melakukan teknik filteringnya yaitu metode Sharpening.

1.1. Rumusan Masalah

1. Apa saja metode penajaman yang umum digunakan dalam pengolahan citra fotografi dan bagaimana prinsip kerja masing-masing metode tersebut?

2. Sejauh mana efektivitas metode sharpening seperti unsharp masking, high-pass filtering, dan filter Sobel dalam meningkatkan kualitas citra fotografi?
3. Bagaimana menerapkan teknik tersebut agar dapat diintegrasikan ke dalam platform berbasis AI dan website agar dapat digunakan oleh masyarakat umum untuk memperjelas gambar tanpa keahlian teknis?

1.2. Batasan Masalah

1. Penelitian ini akan membatasi fokus pada citra fotografi digital, termasuk potret, pemandangan, dan objek statistik. Citra bergerak atau video tidak akan menjadi objek analisis dalam penelitian ini.
2. Teknologi Penelitian ini hanya akan mengeksplorasi beberapa metode penajaman yang umum digunakan, yaitu unsharp masking, high-pass filtering, dan filter Sobel. Metode lainnya tidak akan dibahas dalam penelitian ini.
3. Penelitian ini akan dibatasi pada parameter dasar dalam metode penajaman, seperti tingkat ketajaman dan radius, tanpa mempertimbangkan variasi lebih lanjut dalam pengaturan lanjutan atau penggunaan algoritma yang lebih kompleks.
4. Platform yang digunakan adalah Web.
5. Penelitian ini akan fokus pada citra yang diambil dalam kondisi pencahayaan yang beragam, namun tidak akan mempertimbangkan faktor-faktor eksternal seperti waktu pengambilan gambar atau kondisi cuaca yang ekstrem.

1.3. Tujuan Penelitian

1. Menganalisis berbagai metode penajaman yang umum digunakan dalam pengolahan citra fotografi serta memahami prinsip kerja dari masing-masing metode tersebut.
2. Menggali tantangan dalam penerapan metode penajaman citra dan memberikan solusi atau rekomendasi untuk mengatasi kendala yang mungkin muncul dalam prosesnya.
3. Memberikan informasi yang relevan sebagai referensi untuk pengembangan kurikulum pendidikan di bidang pengolahan citra, fotografi, dan teknologi media digital.
4. Membangun pemahaman yang dapat diterapkan dalam pengembangan aplikasi berbasis kecerdasan buatan (AI) untuk melakukan pengolahan gambar secara otomatis dan efisien.

1.4. Manfaat Penelitian

1. Manfaat bagi Fotografer

Penelitian ini memberikan wawasan tentang berbagai metode penajaman, membantu fotografer memilih teknik yang paling sesuai dengan jenis gambar yang ingin dihasilkan.

2. Manfaat bagi Editor

Penelitian ini dapat membantu editor foto memenuhi standar profesional yang diperlukan dalam industri, terutama dalam publikasi atau presentasi karya.

3. Manfaat bagi Akademisi dan Peneliti

Menjadi referensi ilmiah dalam bidang pengolahan citra digital dan

teknologi AI.

4. Manfaat bagi Masyarakat Umum

Memberikan akses pada teknologi pengolahan citra berbasis website yang user-friendly dan tidak memerlukan keahlian teknis.

BAB II

LANDASAN TEORI

2.1. Citra Digital

Citra digital adalah representasi visual dari objek atau suasana yang diubah menjadi data numerik dan dapat diproses serta disimpan dalam komputer. Citra ini dihasilkan melalui proses pengambilan gambar menggunakan perangkat elektronik, seperti kamera digital, pemindai, atau alat lainnya yang mampu menangkap informasi visual.

Pengolahan Citra Digital adalah suatu bidang ilmu yang menyelidiki bagaimana citra dibentuk, diproses, dan dianalisis untuk menghasilkan informasi yang dapat dimengerti oleh manusia. Salah satu elemen penting dalam pengolahan citra adalah histogram, yang berfungsi sebagai representasi grafis dari distribusi nilai warna atau intensitas piksel yang terdapat dalam citra. Melalui histogram ini, kita dapat mengetahui frekuensi kemunculan nilai intensitas piksel dalam sebuah citra.

Dalam konteks ini, Pengolahan Citra Digital dilakukan dengan menggunakan bahasa pemrograman Python dan lingkungan pengembangan PyCharm. Dengan kedua alat ini, kita dapat melihat hasil perubahan citra serta menganalisis histogram gambar menggunakan Python dan PyCharm. (Silvia Ratna, 2020)

Citra digital juga merupakan sekumpulan data yang merepresentasikan gambar dalam bentuk numerik, biasanya disusun dalam format grid piksel. Setiap piksel memiliki nilai yang menggambarkan warna atau intensitas cahaya pada titik tertentu dalam gambar tersebut.

Tentunya citra digital memiliki kelebihan dan juga kekurangan. Adapun kelebihan yang dimiliki dari citra digital adalah citra digital dapat disimpan dalam ukuran yang kecil dan dapat dengan mudah dibagikan melalui internet. Selain itu, Citra digital dapat dengan mudah diedit menggunakan perangkat lunak pengolahan citra. Berbeda dengan citra analog (seperti film), citra digital tidak mengalami degradasi fisik seiring waktu.

Sedangkan kekurangan yang dimiliki oleh citra digital adalah Kualitas citra sangat bergantung pada kualitas perangkat yang digunakan untuk pengambilan gambar. Citra digital mungkin tidak selalu merepresentasikan objek secara akurat, terutama dalam kondisi pencahayaan yang buruk. Maka dari itu pencahayaan menjadi faktor utama dalam menentukan dari hasil citra itu sendiri.

Citra analog dihasilkan oleh alat akuisisi citra analog, seperti mata manusia dan kamera analog. Contoh dari citra analog bisa dilihat pada gambaran yang ditangkap oleh mata manusia maupun foto atau film yang dihasilkan oleh kamera analog. Sementara itu, citra digital adalah representasi fungsi intensitas cahaya dalam bentuk diskrit pada bidang dua dimensi. Citra digital terdiri dari sekumpulan piksel (elemen gambar) yang memiliki koordinat (x,y) dan amplitudo $f(x,y)$. Koordinat (x,y) menunjukkan posisi piksel dalam citra, sedangkan amplitudo $f(x,y)$ menggambarkan tingkat intensitas warna citra tersebut. Pengolahan citra (image processing) adalah proses manipulasi piksel-piksel dalam citra digital untuk tujuan tertentu. Pada awalnya, pengolahan citra dilakukan untuk meningkatkan kualitas citra. Namun, dengan perkembangan dunia komputasi dan peningkatan kapasitas serta kemampuan komputer, manusia kini dapat mengambil informasi yang lebih banyak dari suatu citra. (Silvia Ratna, 2020)



Gambar 2.1 kamera

<https://i.pinimg.com/736x/7f/dd/3b/7fdd3bda2076f7bdb2ed96e7ec01810>

[a.jpg](#)

2.2 Sharpening

Pengasahan gambar adalah proses dalam pengolahan citra yang bertujuan untuk meningkatkan ketajaman dan kejelasan gambar. Teknik ini digunakan untuk menonjolkan detail serta tepi dalam gambar, sehingga hasil akhir terlihat lebih tajam dan menarik secara visual. Proses penajaman bekerja dengan meningkatkan kontras di sekitar tepi objek dalam gambar. Hal ini bertujuan untuk memisahkan objek dari latar belakang dan menciptakan ilusi ketajaman yang lebih jelas.

Selain untuk mempertajam gambar, penapis lolos-tinggi digunakan untuk mendeteksi keberadaan tepi (edge detection). Dalam hal ini, pixel-pixel tepi ditampilkan lebih terang (highlight) sedangkan pixel-pixel bukan tepi dibuat gelap (hitam). Model komputasi kuantum adalah komputasi yang dilakukan berdasarkan prinsip kerja atau sifat dari kuantum mekanika yang berbeda

dengan komputasi klasik. Komputasi klasik hanya akan menghasilkan keadaan dengan dua kondisi yang dinyatakan dalam bit yaitu 0 atau 1, sedangkan dalam komputasi kuantum digunakan qubit yang mampu menyatakan tak hingga keadaan yang merupakan kombinasi dari keadaan 0 dan 1. Komputasi kuantum memiliki karakteristik unik karena dapat berada dalam berbagai keadaan sekaligus, yang mengakibatkan munculnya banyak kemungkinan hasil perhitungan. Untuk memperoleh hasil akhir dari komputasi tersebut, perlu dilakukan pengukuran terhadap qubit, yang dikenal sebagai pengukuran kuantum. Proses ini akan menghentikan keadaan qubit dan memaksa sistem untuk memilih salah satu dari semua kemungkinan jawaban yang ada.

Selain itu, penajaman citra dapat dicapai melalui proses Highpass Filtering, yang bekerja dengan cara mengurangi komponen frekuensi rendah tanpa mengganggu informasi yang terdapat pada frekuensi tinggi selama transformasinya. Highpass filtering merupakan kebalikan dari lowpass filtering, persamaannya dapat dilihat pada persamaan dibawah ini [14][15]:

$$H_{hp}(u,v) = 1 - H_{lp}(u,v)$$

$H_{hp}(u,v)$ merupakan fungsi dari lowpass filter yang berkaitan.

(Ardiwinata, 2023).

2.2.1 Unsharp Masking

Unsharp masking adalah salah satu teknik paling populer dalam proses pemacuan gambar. Meskipun namanya mengandung kata "unsharp," metode ini justru berfungsi untuk meningkatkan ketajaman gambar. Unsharp Masking (USM) merupakan teknik yang sering dipakai dalam pengolahan citra untuk meningkatkan ketajaman gambar. Meskipun namanya mengandung kata "unsharp," metode ini

justru berfungsi untuk membuat gambar lebih tajam dengan cara meningkatkan kontras di sekitar tepi objek. Unsharp masking adalah teknik yang sangat berguna dalam pengolahan citra, memungkinkan pengguna untuk meningkatkan ketajaman gambar secara efektif. Dengan memahami prinsip dasar, proses, dan parameter yang terlibat, Anda dapat mengaplikasikan metode ini dengan lebih baik untuk mencapai hasil yang diinginkan.

Unsharp masking banyak digunakan dalam:

1. Fotografi Digital: Untuk meningkatkan hasil foto sebelum dicetak atau dipublikasikan.
2. Desain Grafis: Meningkatkan elemen visual dalam poster, brosur, dan materi pemasaran.
3. Pengolahan Citra Medis: Untuk meningkatkan kejelasan gambar medis seperti MRI atau CT scan.

Unsharp mask adalah salah satu metode dalam pemrosesan citra digital yang berfungsi untuk mengurangi tingkat keaburan pada gambar akibat guncangan kamera. Istilah "Unsharp" berasal dari penggunaan citra yang lebih kabur dibandingkan citra asli sebagai komponen dalam proses pembuatan mask. Setelah mask diperoleh, ia akan dijumlahkan dengan citra asli, sehingga menghasilkan gambar yang tampak lebih tajam dibandingkan dengan gambar awal.

$$(I * f)(x, y) = \sum_{z=-a}^a \sum_{t=-b}^b F(s, t) I(x + s, y + t)$$

di mana :

- a = (m-1)/2
- b = (n-1)/2
- m,n = ukuran panjang, lebar filter

Adapun rumus pembuatan unsharp seperti di bawah ini:

$$G(x, y) = f(x, y) - f_{smooth}(x, y)$$

$$M(x, y) = I(x, y) - G(x, y)$$

Konsep “Unsharp” ini muncul karena teknik ini memanfaatkan citra yang kabur untuk menciptakan sebuah "topeng" dari gambar asli. Topeng Unsharp kemudian dipadukan dengan komponen negatif untuk memberikan ilusi bahwa gambar yang dihasilkan lebih tajam dari citra aslinya. Dari perspektif pengolahan citra, topeng Unsharp biasanya merupakan filter yang bersifat linier atau nonlinier yang memperkuat komponen frekuensi tinggi.

Adapun proses penyelesaian dapat dilakukan dengan menggunakan rumus :



Gambar 2. 2 *Sharpening*

<https://www.researchgate.net/profile/Daniel-Gareau-2/publication/36398405/figure/fig26/AS:797426021576714@1567132587258/Example-of-unsharp-masking-in-digital-image-processing-reproduced-with-permission-from.ppm>

Unsharp masking merupakan teknik yang sangat bermanfaat dalam pengolahan citra, yang memungkinkan pengguna untuk secara efektif meningkatkan ketajaman gambar. Dengan memahami prinsip-prinsip dasar, proses, dan parameter yang terlibat, Anda akan dapat menerapkan metode ini dengan lebih tepat guna mencapai hasil yang diinginkan.

Dibandingkan dengan metode dekonvolusi, Unsharp mask memiliki pendekatan yang berbeda dalam memperbaiki citra. Pada Unsharp mask, tidak

diperlukan informasi tentang penyebab kekaburan citra, sedangkan dekonvolusi memerlukan data tersebut sebagai dasar untuk proses perbaikannya. Perbedaan ini mengakibatkan waktu persiapan dan pemrosesan citra yang sangat berbeda antara kedua metode. Unsharp mask, dengan proses yang lebih sederhana, dapat menyelesaikan pemrosesannya dalam waktu yang relatif lebih cepat, sementara dekonvolusi membutuhkan waktu yang lebih lama.

2.2.2 High-Pass Filtering

High-pass filtering bekerja dengan membiarkan komponen frekuensi tinggi, yang merupakan detail, melewati filter, sementara komponen frekuensi rendah, atau latar belakang, dihilangkan. Proses ini menghasilkan gambar yang menonjolkan detail yang tajam dan tepi yang jelas. Meskipun tidak selalu diperlukan, konversi gambar ke grayscale dapat mempermudah proses dengan menghilangkan informasi warna. Sebuah blur (biasanya Gaussian blur) diterapkan pada gambar untuk mengurangi detail frekuensi tinggi. Ini menghasilkan gambar yang lebih halus.

High-Pass Filter (HPF) atau penapis lolos tinggi merupakan salah satu metode dalam penajaman (sharpening) gambar. Tujuan utama dari proses penajaman ini adalah untuk menonjolkan detail-detail halus dalam gambar, serta memperbaiki detail yang mungkin telah kabur akibat kesalahan atau efek yang muncul secara alami selama proses akuisisi citra tertentu. High-pass filter merupakan contoh dari filter yang di terapkan metode ini untuk menyaring sinyal berfrekuensi tinggi, seperti titik-titik halus, detail-detail kecil, serta tepi dan garis yang tajam. Dengan kata lain, teknik ini bertujuan untuk mengambil informasi citra pada frekuensi tinggi sembari menghilangkan data berfrekuensi rendah. Oleh karena itu, filter high-pass sering dikenal sebagai filter penajaman tepi, karena

digunakan dalam proses meningkatkan ketajaman tepi-tepi objek. Teknik ini umum diterapkan dalam aplikasi deteksi tepi objek. (Muhammad Akbar Ramadhan Tanjung, 2021)

*Gambar high-pass dihasilkan dengan mengurangkan gambar yang diblur dari gambar asli:

High-pass image=Original image–Blurred image

*Gambar high-pass yang dihasilkan dapat ditambahkan kembali ke gambar asli untuk meningkatkan ketajaman:

Sharpened image=Original image+k×High-pass image

Di mana k adalah koefisien yang mengontrol tingkat penajaman.

High-Pass Filtering tentunya memiliki kelebihan dan juga kekurangan. Adapun kelebihan dari High-Pass Filtering yaitu Detail yang Tajam, Efektif dalam meningkatkan detail halus dan tepi. Kontrol yang Baik, Parameter yang dapat disesuaikan memungkinkan pengguna untuk mengontrol tingkat ketajaman. Juga Efektivitas pada Gambar Noise, Bisa mengurangi dampak noise dengan memfokuskan pada detail tinggi. Sedangkan kekurangan dari High-Pass Filtering yaitu Artefak Visual, Dapat menghasilkan halo atau artefak di sekitar tepi jika tidak digunakan dengan hati-hati. Tidak Mengatasi Noise Secara Langsung, Sementara metode ini bisa meningkatkan detail, noise juga dapat meningkat jika gambar asli sudah memiliki noise tinggi. Tidak Cocok untuk Semua Jenis Gambar, beberapa gambar mungkin tidak mendapatkan hasil yang diinginkan, terutama jika terdapat banyak detail frekuensi rendah.

Tapis lolos tinggi (high-pass filter) adalah salah satu metode dalam pemrosesan citra yang beroperasi di domain spasial. Teknik ini bekerja langsung pada piksel citra, memodifikasi nilai-nilai piksel sesuai dengan aturan yang

bergantung pada nilai piksel aslinya, sehingga dapat dianggap sebagai proses lokal atau titik. Pendekatan berbasis spasial ini membuatnya lebih mudah dipahami dan mengurangi kompleksitas, menjadikannya lebih sederhana dari segi konsep. Beberapa aturan yang berlaku dalam high-pass filter telah dijelaskan oleh Munir (2004).

1. Koefisien penapis boleh negatif, nol, ataupun bernilai positif.
2. Total keseluruhan koefisiennya ialah bernilai 0 ataupun 1.

Jika jumlah koefisiennya sama dengan 0, maka nilai setiap elemen dengan frekuensi rendah akan berkurang. Namun, jika total koefisiennya sama dengan 1, maka nilai elemen yang memiliki frekuensi rendah akan tetap seperti nilai semula.

$$\sum H(x, y) = 0 \dots \dots \dots$$



(a) Citra asli



(b) Hasil filtering dengan kernel (i)



(c) Hasil filtering dengan kernel (ii)



(b) Hasil filtering dengan kernel (iii)

Gambar 2.3 *grayscale*

<https://pemrogramanmatlab.com/pengolahan-citra-digital/perbaikan-kualitas-citra/>

High-pass filtering merupakan sebuah teknik yang sangat bermanfaat dan efektif dalam pengolahan citra, yang utama bertujuan untuk meningkatkan detail serta ketajaman gambar. Metode ini berfungsi dengan cara memisahkan informasi frekuensi tinggi dari frekuensi rendah, sehingga pengguna dapat menonjolkan tepi dan detail halus dalam gambar dengan lebih jelas.

2.2.3 Filter Sobel

Filter Sobel adalah suatu teknik dalam pengolahan citra yang berfungsi untuk mendeteksi tepi serta mengukur perubahan intensitas piksel pada sebuah

gambar. Metode ini sangat ampuh dalam menonjolkan tepi objek dan memberikan informasi mengenai arah serta kekuatan tepi tersebut. Sebagai salah satu dari berbagai operator deteksi tepi, Filter Sobel banyak digunakan dalam analisis citra.

Filter Sobel didasarkan pada konsep diferensiasi yang berfungsi untuk mendeteksi perubahan intensitas dalam sebuah gambar. Dengan menggunakan filter Sobel, kita dapat menghitung gradien intensitas pada setiap piksel, yang menunjukkan seberapa cepat nilai intensitas berubah. Gradien ini memberikan informasi penting mengenai arah dan kekuatan tepi pada gambar.

Operator Sobel adalah jenis operator pembeda diskrit yang dapat digunakan untuk mengidentifikasi tepi dengan menggunakan realisasi gradien. Pada setiap lokasi piksel dalam gambar, resultan operator Sobel sesuai dengan vektor gradien atau normal terhadap vektor ini. Operator ini menghitung gradien intensitas gambar di setiap titik, mewujudkan arah variasi intensitas dan arah. Diferensiasi parsial orde pertama digunakan oleh operator Sobel. (Konferensi Internasional tentang Sistem dan Kontrol Inventif (ICISC 2019)).

Filter Sobel menggunakan dua matriks kernel, satu untuk mendeteksi perubahan horizontal dan satu untuk deteksi vertikal:

Kernel Horizontal (Gx):

$$[-1 \ 0 \ 1]$$

$$[-2 \ 0 \ 2]$$

$$[-1 \ 0 \ 1]$$

Kernel Vertikal (Gy):

$$[-1 \ -2 \ -1]$$

$$[0 \ 0 \ 0]$$

[1 2 1]

Adapun kelebihan dari Filter Sobel yaitu Deteksi Tepi yang Efisien, Sobel filter sangat efektif dalam mendeteksi tepi tajam dan memberikan hasil yang baik pada gambar dengan kontras tinggi. Sensitivitas terhadap Noise, meski cukup sensitif terhadap noise, Sobel filter dapat memberikan hasil yang relatif halus dibandingkan dengan beberapa metode deteksi tepi lainnya. Nilai bit pada citra akan mempengaruhi dalam pemrosesan terutama pada tampilan tepi objek. Semakin besar bit pada citra, maka ukuran citra semakin lebih besar sehingga tampilan citra pun menjadi lebih halus, baik pada band multispektral maupun pankromatik. (Luhur Moekti Prayogo, 2021)

Arah Gradien, memberikan informasi tidak hanya tentang kekuatan tepi tetapi juga arah tepi, yang sangat berguna dalam analisis citra. Sedangkan kekurangan dari Filter Sobel yaitu Sensitivitas terhadap Noise, Filter Sobel dapat memperkuat noise dalam gambar, yang mungkin mengganggu hasil deteksi tepi. Tidak Optimal untuk Semua Gambar, dalam gambar dengan detail yang sangat halus atau noise tinggi, hasilnya mungkin tidak memuaskan.

$$G_x = \begin{pmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{pmatrix} * A \text{ and } G_y = \begin{pmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix} \dots \dots \dots$$

Menggunakan prinsip kernel, dimana * menunjukkan operasi jendela konvolusi pemrosesan 2 dimensi (2D).



Gambar 2. 4 *Contoh filter sobel*

<https://intelcorp.scene7.com/is/image/intelcorp/exm-opencl-sobel-filter-screenshot:1920-1080?wid=1648&hei=927>

Filter Sobel merupakan metode deteksi tepi yang sangat efektif dan efisien dalam pengolahan citra. Dengan memanfaatkan kernel yang dirancang khusus untuk mendeteksi gradien intensitas, metode ini mampu memberikan informasi penting mengenai kekuatan dan arah tepi dalam sebuah gambar. Meskipun terdapat beberapa kelemahan, keunggulan serta berbagai aplikasinya menjadikan filter Sobel sebagai alat yang esensial dalam analisis citra modern.

2.3 Web

Website disebut juga site, situs, situs web atau portal. Merupakan kumpulan halaman web yang berhubungan antara satu dengan lainnya, halaman pertama sebuah website adalah home page, sedangkan halaman demi halamannya secara mandiri disebut web page, dengan kata lain website adalah situs yang dapat diakses dan dilihat oleh para pengguna internet diseluruh dunia. Website adalah situs yang dapat diakses dan dilihat oleh para pengguna Internet. Pengguna Internet semakin

hari semakin bertambah banyak, sehingga hal ini adalah potensi pasar yang berkembang terus. (Wahiddin Abbas, 2021)

2.3.1 HTML

Hypertext markup language (HTML) merupakan bahasa pemrograman yang biasa digunakan untuk membuat komponen pada halaman web. (Atmaja, 2022).

2.3.2 CSS

CSS (Cascading Style Sheet) adalah bahasa pemrograman untuk memberikan tampilan desain yang akan digunakan pada web seperti warna, *font*, *outline*, *background*, menyesuaikan tampilan website dengan ukuran layar, dsb. CSS digunakan pada pembuatan website ini adalah untuk berkolaborasi dengan HTML agar dapat menghasilkan tampilan website yang menarik. (Sari et al., 2022).

2.3.3 Javascript

JavaScript adalah bahasa pemrograman yang digunakan untuk membuat halaman web menjadi interaktif dan dinamis. Berbeda dengan HTML (yang menyusun struktur) dan CSS (yang mengatur tampilan), JavaScript digunakan untuk mengatur perilaku dari elemen-elemen web.

JavaScript merupakan bahasa script yang mehidupkan halaman-halaman HTML. JavaScript dapat dijalankan hampir pada semua platform. JavaScript sangat identik dengan browser, sehingga sangat terintegrasi dengan HTML. Ketika browser memuat suatu halaman, server akan mengirimkan konten utuh dari dokumen termasuk HTML dan statemen-statementen dari JavaScript. Konten HTML kemudian dibaca baris demi baris hingga sampai pada pembacaan JavaScript, pada saat itu interpreter JavaScript akan mengambil alih. Ketika tag penutup JavaScript

diraih, pemrosesan HTML kemudian berlanjut. Program JavaScript digunakan untuk melakukan deteksi dan bereaksi terhadap tindakan yang dilakukan oleh pengguna. JavaScript dapat digunakan untuk mengatur tampilan halaman. JavaScript juga digunakan untuk memvalidasi input dari pengguna ke dalam sebuah form sebelum pengiriman form ke server dilakukan. JavaScript berperan sebagai bahasa pemrograman yang memiliki konstruksi-konstruksi dasar seperti variabel dan tipe data. JavaScript juga dapat menangani event yang diinisiasi oleh pengguna dan menetapkan timing. Kombinasi dari HTML, CSS, dan JavaScript akan menjadikan suatu website yang atraktif bagi pengguna. (Arindra Putawa et al., 2022).

2.3.4 MySQL

MySQL (My Structure Query Language) adalah sebuah perangkat lunak sistem manajemen basis data SQL (Database Management System) atau DBMS yang multithread, multi-user. MySQL memungkinkan secara efisien menyimpan, mencari, mengurutkan dan mendapatkan data. (Raka Yusuf, 2016)

2.3.5 PHP

PHP (akronim dari PHP: Hypertext Preprocessor) adalah bahasa pemrograman yang berfungsi untuk membuat website dinamis maupun aplikasi web. Berbeda dengan HTML yang hanya bisa menampilkan konten statis, PHP bisa berinteraksi dengan database, file dan folder, sehingga membuat PHP bisa menampilkan konten yang dinamis dari sebuah website. Blog, Toko Online, CMS, Forum, dan Website Social Networking adalah contoh aplikasi web yang bisa dibuat oleh PHP. PHP adalah bahasa scripting, bukan bahasa tag-based seperti HTML. PHP termasuk bahasa yang cross-platform, ini artinya PHP bisa berjalan

pada sistem operasi yang berbeda-beda (Windows, Linux, ataupun Mac). Program PHP ditulis dalam file plain text (teks biasa) dan mempunyai akhiran “.php”.

2.4 Artificial Intelligence (AI)

AI (Artificial Intelligence) atau Kecerdasan Buatan adalah cabang ilmu komputer yang bertujuan untuk menciptakan mesin atau sistem yang bisa meniru kecerdasan manusia, seperti berpikir, belajar, memecahkan masalah, dan membuat keputusan. Kecerdasan Buatan (Artificial Intelligence atau AI) didefinisikan sebagai kecerdasan yang ditunjukkan oleh suatu entitas buatan. Sistem seperti ini umumnya dianggap komputer. Kecerdasan diciptakan dan dimasukkan ke dalam suatu mesin (komputer) agar dapat melakukan pekerjaan seperti yang dapat dilakukan manusia. Kecerdasan Buatan (Artificial Intelligence atau AI) didefinisikan sebagai kecerdasan yang ditunjukkan oleh suatu entitas buatan. Sistem seperti ini umumnya dianggap komputer. Kecerdasan diciptakan dan dimasukkan ke dalam suatu mesin (komputer) agar dapat melakukan pekerjaan seperti yang dapat dilakukan manusia. Beberapa macam bidang yang menggunakan kecerdasan buatan antara lain sistem pakar, permainan komputer (games), logika fuzzy, jaringan syaraf tiruan dan robotika. (Helfi Nasution, 2012)

BAB III

METODOLOGI PENELITIAN

3.1 Metode Penelitian

Metode penelitian yang diterapkan dalam studi ini adalah pendekatan eksperimen kuantitatif dengan sifat deskriptif. Pemilihan metode ini bertujuan untuk mengeksplorasi serta menganalisis efektivitas berbagai teknik sharpening dalam meningkatkan kualitas gambar fotografi. Penelitian ini dilakukan melalui pengujian dan analisis objek secara sistematis, diiringi dengan pengumpulan data terukur yang memungkinkan penarikan kesimpulan yang valid dan dapat diandalkan.

Observasi dalam penelitian ini adalah Mengamati hasil gambar sebelum dan sesudah penerapan metode sharpening untuk menilai perubahan kualitas visual secara subjektif. Observasi ini dilakukan dengan melibatkan fotografer dan ahli pengolahan citra yang memberikan penilaian terhadap ketajaman, detail, dan kesan estetika gambar. Data ini berasal dari hasil tangkapan beberapa gambar dari kamera photographer.

3.2 Subjek, Objek, Waktu Penelitian

Subjek dalam penelitian ini adalah gambar-gambar fotografi yang diambil oleh fotografer profesional dan amatir. Gambar-gambar ini mencakup berbagai genre, termasuk potret, pemandangan, dan objek statis. Objek penelitian ini berfokus pada kualitas gambar yang dihasilkan setelah penerapan metode sharpening, serta mencakup analisis kuantitatif terhadap hasil yang diperoleh. Waktu penelitian akan berlangsung selama 2 Bulan.

NO	Tahapan Pengerjaan	Deskripsi	Waktu Pelaksanaan
1	Penyusunan Proposal	Menyusun dan mengajukan proposal penelitian.	Minggu 1
2	Tinjauan Pustaka	Mengumpulkan dan mengkaji literatur terkait.	Minggu 2-3
3	Pengambilan Data	Mengambil foto untuk percobaan.	Minggu 4
4	Penerapan Metode Sharpening	Menggunakan software untuk sharpening.	Minggu 5
5	Analisis Hasil	Menganalisis kualitas gambar sebelum dan sesudah.	Minggu 6
6	Penulisan Hasil dan Pembahasan	Menulis hasil penelitian dan diskusi.	Minggu 7
7	Kesimpulan dan Saran	Menyusun kesimpulan dan rekomendasi.	Minggu 8
8	Penyusunan Daftar Pustaka	Mengumpulkan semua referensi.	Minggu 9
9	Revisi dan Penyempurnaan	Melakukan revisi berdasarkan masukan.	Minggu 10
10	Penyelesaian Skripsi	Menyusun dokumen akhir dan persiapan sidang.	Minggu 11

Tabel 3. 1

3.3 Teknik Pengumpulan Data

Teknik pengumpulan data yang digunakan dalam penelitian ini meliputi:

1. Observasi Visual:

Melakukan pengamatan terhadap hasil gambar sebelum dan setelah penerapan metode sharpening untuk menilai perubahan kualitas visual secara subjektif. Proses observasi ini melibatkan fotografer serta ahli pengolahan citra yang memberikan penilaian terkait ketajaman, detail, dan kesan estetika gambar.

2. Kusioner

Mengumpulkan data dari para fotografer tentang persepsi mereka terhadap hasil sharpening gambar. Kuesioner yang disusun akan

mencakup pertanyaan-pertanyaan terkait ketajaman, detail, serta kepuasan visual terhadap gambar yang dihasilkan.

3.4 Metode Pengolahan Data

Pengolahan data dalam Studi kasus peningkatan gambar pada fotografi dapat dioptimalakan dengan beberapa langkah berikut:

1. Pengolahan Gambar

- 1) Pengambilan gambar yang dilakukan oleh para fotografer akan di kumpulkan ke dalam satu file kemudian nantinya akan dimasukkan ke dalam website.
- 2) Gambar yang telah dimasukkan ke dalam website akan dibagi menjadi beberapa kelompok, sesuai dengan keperluan yang akan diproses. Lalu kemudian di bagi ke beberapa metode berdasarkan teknik sharpening yang akan diterapkan: Unsharp Masking, High-Pass Filtering, dan Filter Sobel.
- 3) Proses penerapan dilakukan dengan mengikuti langkah-langkah spesifik untuk masing-masing metode:
 - 1) Unsharp Masking: Menggunakan filter Gaussian untuk membuat salinan blur dari gambar asli, kemudian mengurangkan gambar blur dari gambar asli untuk menonjolkan tepi.
 - 2) High-Pass Filtering: Menggunakan filter low-pass untuk menghasilkan gambar yang lebih halus, lalu mengurangkan gambar blur dari gambar asli untuk mendapatkan detail frekuensi tinggi.

3) Filter Sobel: Menerapkan kernel Sobel untuk mendeteksi tepi pada gambar, baik dalam arah horizontal maupun vertikal, dan menghitung magnitudo gradien.

2. Uji Statistik

Analisis Varians (ANOVA):

- 1) Untuk menentukan apakah ada perbedaan signifikan dalam kualitas gambar yang dihasilkan oleh metode sharpening yang berbeda, analisis varians (ANOVA) akan dilakukan.
- 2) Uji ANOVA satu arah akan digunakan untuk membandingkan rata-rata nilai PSNR dan SSIM dari ketiga metode sharpening.
- 3) Jika hasil ANOVA menunjukkan signifikansi, analisis post-hoc (seperti Tukey HSD) akan dilakukan untuk mengetahui perbandingan spesifik antar kelompok.

3. Evaluasi Hasil

1) Pengumpulan Umpan Balik:

Mengumpulkan umpan balik dari fotografer mengenai hasil akhir setelah penerapan metode sharpening. Kuesioner yang disebar akan mencakup pertanyaan mengenai persepsi mereka terhadap ketajaman, detail, dan kepuasan visual.

2) Analisis Data Kuantitatif:

Data yang diperoleh dari pengukuran PSNR, SSIM, dan kuesioner akan dianalisis untuk menarik kesimpulan tentang efektivitas masing-masing metode sharpening. Hasil analisis ini akan ditampilkan dalam

bentuk tabel dan grafik untuk memudahkan visualisasi dan interpretasi hasil. Visualisasi Data.

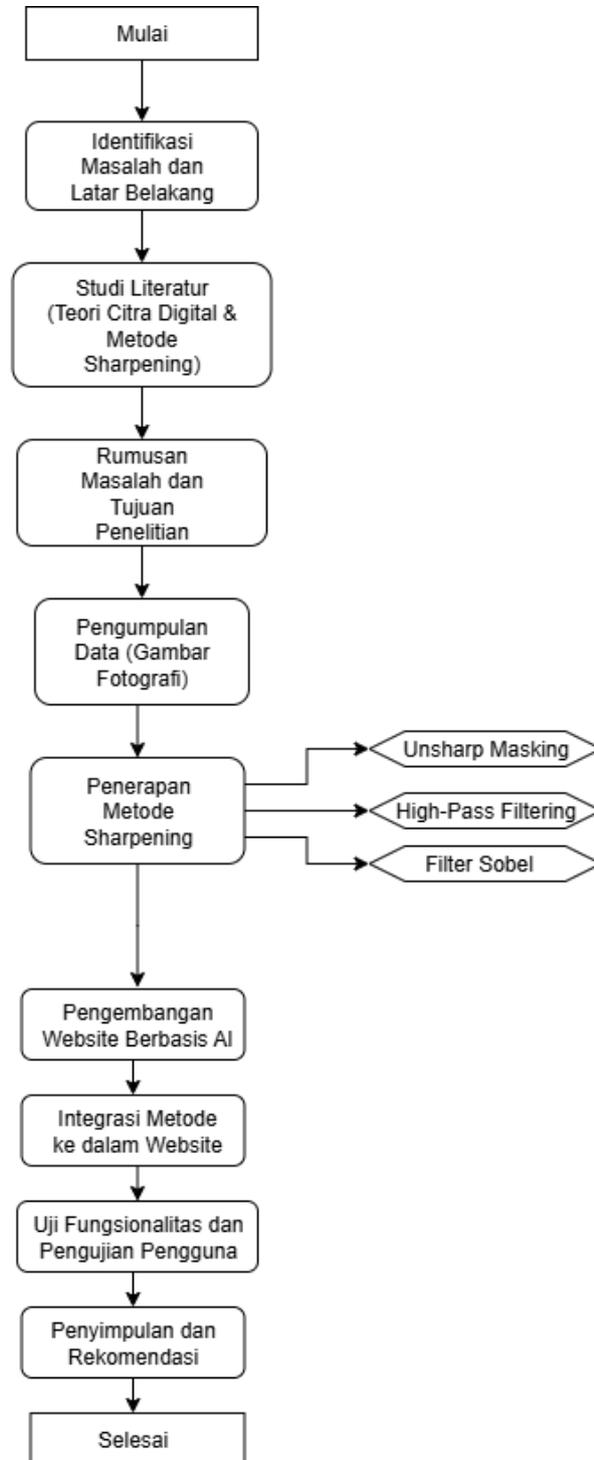
3) Penyimpanan dan Manajemen Data:

- 1) Semua data yang dikumpulkan, termasuk gambar asli dan hasil pengolahan, akan disimpan dalam format terstruktur di dalam database yang aman.
- 2) Penggunaan Google Drive membantu dalam pengelolaan dan analisis data lebih lanjut.

3.5 Alur Penelitian

Alur penelitian dalam skripsi ini dimulai dengan identifikasi permasalahan, yaitu rendahnya kualitas ketajaman citra fotografi akibat berbagai faktor teknis seperti pencahayaan buruk atau fokus yang tidak tepat. Setelah itu dilakukan studi literatur untuk memahami konsep dasar citra digital dan metode sharpening seperti Unsharp Masking, High-Pass Filtering, dan Filter Sobel. Selanjutnya, dilakukan pengumpulan data citra dari berbagai jenis foto (potret, pemandangan, dan objek statis), yang kemudian diproses menggunakan masing-masing metode penajaman citra. Hasil dari proses ini dianalisis menggunakan metrik kuantitatif seperti PSNR (Peak Signal-to-Noise Ratio) dan SSIM (Structural Similarity Index), serta evaluasi subjektif dari pengguna melalui kuesioner. Berdasarkan hasil analisis tersebut, metode yang paling efektif akan diidentifikasi dan diintegrasikan ke dalam sebuah website berbasis AI yang dirancang untuk mempermudah masyarakat dalam meningkatkan kualitas gambar secara otomatis. Tahapan akhir dari penelitian ini adalah

evaluasi fungsionalitas dan kegunaan website, serta penyusunan kesimpulan dan saran untuk pengembangan lebih lanjut.



Gambar 3. 1 *Flowchart Alur Penelitian*

3.6 Pengembangan Website Berbasis AI

Sebagai bagian dari realisasi praktis penelitian ini, metode sharpening citra akan diintegrasikan ke dalam sebuah platform website berbasis AI yang dirancang untuk memudahkan pengguna dalam melakukan peningkatan kualitas gambar tanpa keahlian teknis. Sehingga bagi pengguna yang tidak memiliki kemampuan dalam mengolah gambar pada beberapa aplikasi seperti Photoshop dan MATLAB, dapat menerapkan metode ini dengan mudah.

3.6.1 Tujuan Pengembangan Website

Dengan dibuatnya website, tentunya memiliki tujuan dari pengembangan website berbasis AI dalam penelitian ini.

- 1) Meningkatkan aksesibilitas teknologi pengolahan citra bagi pengguna umum.

Teknologi sharpening citra umumnya hanya tersedia dalam software profesional yang membutuhkan keahlian teknis. Dengan mengembangkan platform berbasis web, pengguna awam tanpa latar belakang teknis tetap dapat memanfaatkan fitur peningkatan kualitas gambar secara mudah dan efisien.

- 2) Mengintegrasikan metode sharpening ke dalam sistem otomatis berbasis AI.

Website ini dirancang untuk mengotomatisasi proses penajaman citra menggunakan algoritma Unsharp Masking, High-Pass Filtering, dan Filter Sobel. Pengguna cukup mengunggah gambar dan memilih metode yang

diinginkan, maka sistem akan memproses dan menampilkan hasil secara real-time tanpa intervensi manual yang kompleks.

3) Menyediakan antarmuka yang ramah pengguna (user-friendly) dan intuitif. Fokus website ini adalah pada kemudahan penggunaan, sehingga desain antarmuka akan dibuat sesederhana mungkin agar dapat diakses oleh semua kalangan, termasuk yang tidak terbiasa dengan perangkat lunak editing gambar.

4) Mendemonstrasikan penerapan praktis dari penelitian ke dalam solusi digital. Website ini juga berfungsi sebagai bukti nyata dari hasil penelitian yang tidak hanya bersifat teoritis, tetapi juga implementatif. Platform ini akan menjadi media untuk menguji dan menampilkan efektivitas metode sharpening dalam konteks aplikasi dunia nyata.

5) Mendukung evaluasi dan pengembangan sistem di masa depan. Website akan menyimpan hasil pengolahan dan data umpan balik pengguna untuk keperluan evaluasi sistem. Dengan data tersebut, pengembangan lebih lanjut dapat dilakukan untuk menyesuaikan metode dengan kebutuhan pengguna atau menambahkan fitur lanjutan seperti deteksi otomatis atau rekomendasi metode terbaik.

3.6.2 Tools dan Teknologi yang Digunakan

Dalam pengembangan website berbasis AI untuk penajaman citra ini, beberapa teknologi utama yang digunakan antara lain:

1) HTML (HyperText Markup Language)

HTML digunakan untuk membangun struktur dasar halaman website.

Elemen-elemen seperti form upload gambar, tombol pemilihan metode sharpening, dan tampilan hasil citra akan dirancang menggunakan

HTML sebagai fondasi dari antarmuka pengguna.

2) CSS (Cascading Style Sheets)

CSS digunakan untuk mengatur tampilan visual website agar lebih

menarik dan responsif. Dengan CSS, tata letak, warna, font, dan

penyesuaian tampilan pada berbagai ukuran layar (mobile, tablet, desktop)

dapat diatur secara fleksibel.

3) JavaScript

JavaScript digunakan untuk memberikan interaktivitas pada halaman web.

Fungsi seperti pratinjau gambar sebelum diunggah, pemrosesan asinkron

(AJAX), dan interaksi pengguna lainnya akan diatur melalui skrip

JavaScript agar pengalaman pengguna menjadi lebih dinamis dan

responsif.

4) Python + OpenCV / Scikit-Image

Untuk pengolahan citra berbasis AI, digunakan bahasa pemrograman

Python dengan bantuan pustaka seperti OpenCV atau Scikit-Image.

Library ini menyediakan fungsi yang kuat dan efisien untuk menerapkan

metode sharpening seperti Unsharp Masking, High-Pass Filtering, dan Filter Sobel secara otomatis di sisi server.

5) PHP atau Framework Python Web (Flask/Django)

PHP dapat digunakan sebagai backend alternatif untuk menangani pengunggahan gambar, pemanggilan algoritma pengolahan citra, dan pengiriman kembali hasil ke frontend. Namun, jika seluruh proses berbasis Python, maka framework seperti Flask atau Django lebih sesuai untuk menghubungkan frontend dengan modul AI di backend.

6) MySQL (Database Management System)

MySQL digunakan untuk menyimpan data pengguna, riwayat pengolahan citra, serta hasil evaluasi dan umpan balik. Penyimpanan terstruktur ini berguna untuk evaluasi hasil penelitian dan pengembangan sistem di masa depan.

3.6.3 Alur Proses Website

Alur kerja dari website berbasis AI ini dirancang secara sistematis agar pengguna dapat melakukan penajaman citra secara mudah dan otomatis. Berikut ini adalah langkah-langkah prosesnya:

1) Pengguna mengunggah gambar ke website.

Langkah awal dimulai saat pengguna membuka halaman website dan memilih gambar yang ingin ditingkatkan kualitasnya. File gambar diunggah melalui form input yang telah disediakan di antarmuka utama.

2) Pengguna memilih metode sharpening yang diinginkan.

Setelah gambar diunggah, pengguna diberikan pilihan untuk menentukan metode penajaman yang akan digunakan, yaitu: *Unsharp Masking*, *High-Pass Filtering*, atau *Filter Sobel*. Setiap metode memiliki karakteristik yang berbeda, dan pengguna dapat memilih sesuai dengan kebutuhan hasil akhir yang diinginkan.

3) Gambar diproses oleh sistem menggunakan algoritma yang dipilih.

Setelah metode dipilih, sistem akan secara otomatis memproses gambar menggunakan algoritma yang telah diimplementasikan dalam backend berbasis Python (menggunakan OpenCV/Scikit-image). Proses ini dilakukan di sisi server agar tetap ringan di sisi pengguna.

4) Hasil gambar ditampilkan kepada pengguna.

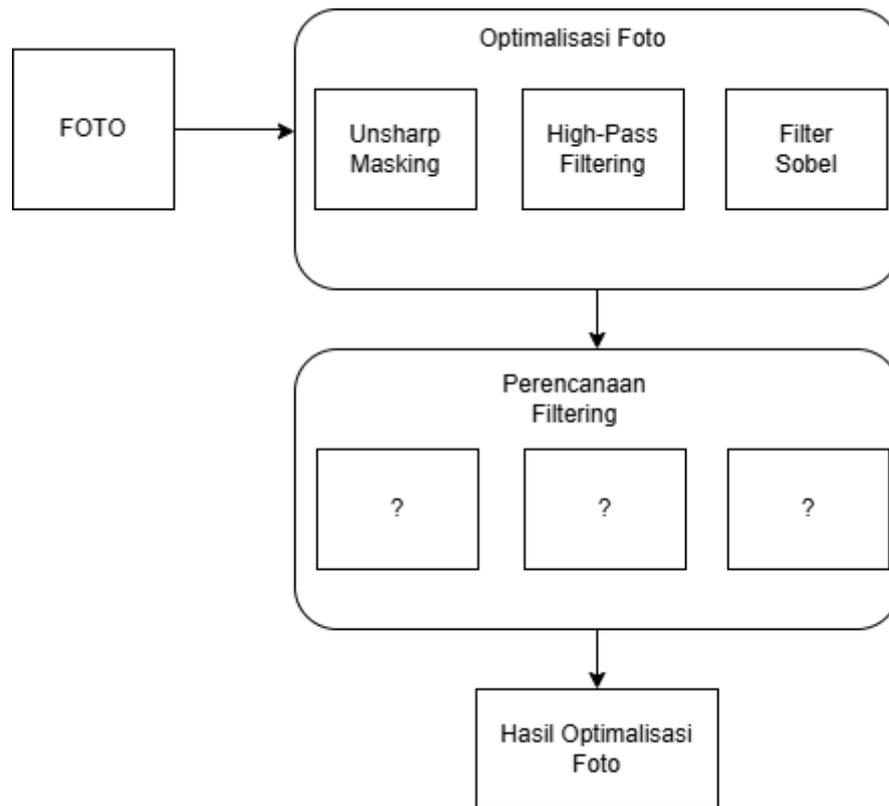
Setelah proses selesai, gambar hasil penajaman akan ditampilkan langsung di halaman website sebagai pratinjau. Dengan ini, pengguna dapat segera melihat perbandingan antara gambar asli dan gambar yang telah diproses.

5) Pengguna dapat mengunduh hasil gambar.

Website menyediakan opsi untuk mengunduh gambar hasil pengolahan agar dapat disimpan dan digunakan sesuai kebutuhan pengguna. File dapat diunduh dalam format umum seperti JPG atau PNG.

6) Sistem menyimpan data hasil dan riwayat penggunaan.

Setiap proses akan dicatat dalam basis data, termasuk nama file, metode yang digunakan, dan waktu pemrosesan. Data ini bermanfaat untuk evaluasi sistem dan pengembangan fitur lebih lanjut.



Gambar 3. 2 *Flowchart Alur Website*

BAB IV

HASIL DAN PEMBAHASAN

4.1 Tampilan Antarmuka Aplikasi

Antarmuka pengguna (user interface) merupakan komponen penting dalam pengembangan sebuah aplikasi, khususnya aplikasi berbasis web yang ditujukan untuk pengguna umum dari berbagai latar belakang. Dalam penelitian ini, antarmuka aplikasi dirancang dengan pendekatan *user-friendly* dan *minimalis modern*, yang memudahkan interaksi antara pengguna dan sistem pemrosesan gambar yang dibangun. Tujuan utama dari desain antarmuka adalah untuk memungkinkan proses unggah, pemrosesan, dan unduh gambar dilakukan secara intuitif tanpa memerlukan keahlian teknis dari pengguna.

Aplikasi ini memiliki beberapa tampilan utama yang saling terintegrasi dalam sebuah alur kerja yang sederhana namun efisien. Adapun penjabaran tampilan antarmuka meliputi:

4.1.1 Halaman Landing Page

Halaman ini merupakan titik masuk awal bagi pengguna sebelum masuk ke sistem. Tampilan dirancang dengan visual yang sederhana dan langsung mengarahkan pengguna ke opsi “Login” atau “Daftar” akun. Halaman ini juga mencerminkan identitas aplikasi, baik dari segi nama maupun tujuan utama program yang dapat dipahami dalam sekali pandang. Elemen visual seperti tipografi, warna, dan tombol disesuaikan agar responsif di berbagai perangkat.



Gambar 4. 1 *Landing Page*

4.1.2 Halaman Login dan Registrasi

Halaman login digunakan oleh pengguna yang telah memiliki akun untuk mengakses sistem. Formulir login terdiri dari dua bidang input, yaitu nama pengguna (*username*) dan kata sandi (*password*). Validasi sisi server diterapkan untuk memastikan keamanan, termasuk pengecekan terhadap keaslian data melalui hashing password menggunakan algoritma `generate_password_hash`.

Halaman registrasi menyediakan form untuk pengguna baru, yang mencakup input nama pengguna, alamat email, dan kata sandi. Sistem juga menerapkan validasi agar nama pengguna dan email tidak boleh sama dengan pengguna lain yang sudah terdaftar, dengan tujuan menghindari duplikasi data. Keamanan input dari pengguna dijaga melalui sanitasi data dan pemberian umpan balik (*feedback message*) dalam bentuk notifikasi berbasis Bootstrap Alert.



Gambar 4. 2 Halaman Daftar

4.1.3 Halaman Utama Aplikasi (Dashboard)

Setelah berhasil login, pengguna akan diarahkan ke halaman utama aplikasi. Di sinilah seluruh proses inti aplikasi dilakukan. Fitur utama yang tersedia pada halaman ini antara lain:

- **Formulir Unggah Gambar:** Pengguna dapat memilih file gambar dari perangkat mereka untuk diproses. Sistem hanya menerima format gambar yang telah ditentukan (misalnya .jpg, .png, .jpeg, .bmp, .gif), dan menolak format selain itu secara otomatis dengan pemberitahuan yang sesuai.
- **Validasi Otomatis:** Setiap file yang diunggah akan diverifikasi menggunakan library Pillow (PIL) untuk memastikan bahwa file tersebut benar-benar merupakan gambar, bukan file lain yang menyamar.
- **Pemrosesan Otomatis:** Setelah file terverifikasi, gambar akan diproses oleh sistem yang menerapkan metode Unsharp Masking berdasarkan tingkat ketajaman yang dihitung secara otomatis. Pengguna tidak perlu memilih metode secara manual, karena sistem telah dilengkapi dengan logika pengambilan keputusan berdasarkan nilai *Laplacian Variance*.

- Tampilan Hasil: Setelah pemrosesan selesai, dua gambar akan ditampilkan berdampingan, yaitu gambar asli dan hasil peningkatan ketajaman. Selain itu, diberikan deskripsi tindakan yang dilakukan oleh sistem (misalnya: "Gambar ditajamkan menggunakan Unsharp Masking").
- Unduhan Gambar Hasil: Terdapat tombol yang memungkinkan pengguna untuk mengunduh hasil pemrosesan gambar secara langsung.

Desain halaman utama menekankan kepraktisan, dengan meminimalkan jumlah klik yang dibutuhkan pengguna untuk menyelesaikan tugasnya.



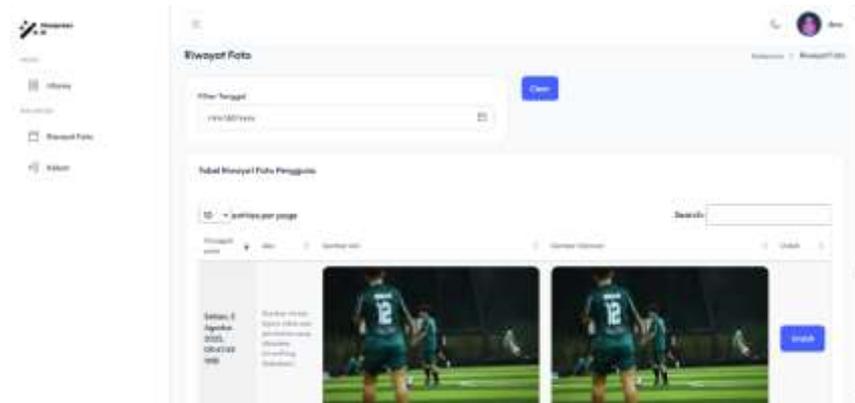
Gambar 4.3 Halaman Utama

4.1.4 Halaman Riwayat Pemrosesan

Fitur tambahan yang disediakan adalah halaman riwayat (*history*) yang merekam semua aktivitas pengguna dalam memproses gambar. Setiap entri pada halaman ini menampilkan informasi sebagai berikut:

- Nama file asli dan file hasil pemrosesan
- Waktu unggah yang sudah dikonversi ke format waktu lokal (WIB), dengan penyesuaian nama hari dan bulan ke dalam Bahasa Indonesia
- Deskripsi tindakan yang dilakukan pada gambar tersebut

Data ini ditampilkan dalam bentuk tabel yang dapat dipilah berdasarkan waktu unggah terbaru. Halaman ini memudahkan pengguna untuk mengakses kembali gambar yang telah diproses sebelumnya tanpa perlu mengunggah ulang.



Gambar 4. 4 *Halaman Riwayat*

4.1.5 Responsivitas dan Aksesibilitas

Seluruh tampilan antarmuka dikembangkan menggunakan framework CSS seperti Bootstrap agar responsif di berbagai ukuran layar, termasuk desktop dan perangkat mobile. Selain itu, aspek aksesibilitas diperhatikan dengan menambahkan *alt text* pada elemen gambar, serta menjaga kontras warna dan ukuran huruf agar mudah dibaca oleh semua kalangan.

4.2 Proses Unggah dan Validasi Gambar

Proses unggah (*upload*) dan validasi gambar merupakan tahapan awal yang sangat krusial dalam sistem pemrosesan citra berbasis web. Tahapan ini menjadi fondasi untuk memastikan bahwa hanya gambar yang sesuai format dan tidak rusak yang dapat diproses lebih lanjut. Sistem perlu dirancang sedemikian rupa agar mampu menangani berbagai kemungkinan input dari pengguna, termasuk kesalahan input, file tidak valid, maupun potensi serangan (*security threat*) dari file yang menyamar sebagai gambar.

Dalam aplikasi yang dikembangkan, proses unggah dan validasi mencakup lima tahapan utama, yaitu: (1) pemilihan file oleh pengguna, (2) pengecekan keberadaan file, (3) validasi ekstensi file, (4) validasi struktur file menggunakan pustaka image verification, dan (5) penyimpanan file secara aman. Penjelasan setiap tahap adalah sebagai berikut:

4.2.1 Pemilihan File oleh Pengguna

Pada antarmuka halaman utama aplikasi, pengguna diberikan fitur untuk memilih file gambar dari perangkat masing-masing. Input ini berbentuk form *HTML type="file"* yang dibungkus dalam form POST. Setelah memilih gambar, pengguna dapat langsung menekan tombol unggah untuk memulai proses.

Pengguna tidak diberi opsi untuk memilih metode sharpening secara manual, karena sistem telah dirancang untuk secara otomatis menganalisis ketajaman gambar dan menentukan apakah diperlukan proses penajaman atau tidak.

4.2.2 Pengecekan Keberadaan File

Langkah pertama dalam sisi backend adalah memastikan bahwa file benar-benar dikirimkan oleh pengguna. Validasi ini dilakukan melalui pengecekan kondisi:

```
if 'file' not in request.files:
```

Gambar 4. 5 Syntax 1

Jika file tidak ditemukan dalam permintaan, sistem akan mengembalikan pesan kesalahan kepada pengguna. Hal ini mencegah kemungkinan server crash akibat mencoba memproses file yang tidak ada. Selain itu, nama file juga diperiksa

agar tidak kosong. Misalnya, pengguna mungkin memilih file lalu membatalkan sebelum menekan tombol unggah. Sistem memverifikasi kondisi ini dengan:

```
if file.filename == '':
```

Gambar 4.6 *Syntax 2*

4.2.3 Validasi Ekstensi File

Untuk menghindari file dengan ekstensi yang tidak sesuai atau berbahaya, dilakukan validasi ekstensi file berdasarkan whitelist. Hanya ekstensi .jpg, .jpeg, .png, .bmp, dan .gif yang diperbolehkan. Validasi dilakukan dengan metode:

```
def allowed_file(filename):  
    return '.' in filename and filename.rsplit('.', 1)[1].lower() in ALLOWED_EXTENSIONS
```

Gambar 4.7 *Syntax 3*

Langkah ini tidak hanya meningkatkan keamanan, tetapi juga menyederhanakan proses decoding gambar oleh pustaka seperti OpenCV dan PIL, yang mendukung format-format tersebut secara native.

4.2.4 Validasi Struktur File Gambar

Setelah file lolos dari validasi ekstensi, sistem melakukan verifikasi lebih mendalam terhadap isi file untuk memastikan bahwa file tersebut benar-benar merupakan gambar. Untuk itu digunakan pustaka Pillow (PIL) dengan metode `Image.verify()`:

```
img = Image.open(file)  
img.verify() # Validasi struktur file
```

Gambar 4.8 *Syntax 4*

Fungsi ini akan memeriksa apakah header file konsisten dengan jenis file gambar yang diklaim. Jika file tidak valid, akan ditolak dengan memberikan pesan

kesalahan seperti: “File yang diunggah bukanlah gambar yang valid.” Langkah ini sangat penting untuk mencegah terjadinya serangan yang dikenal sebagai *file masquerading*, yaitu ketika file executable menyamar sebagai file gambar (misalnya, virus.jpg.exe).

4.2.5 Penyimpanan File Secara Aman

Setelah file berhasil lolos semua tahap validasi, sistem akan menyimpannya di folder uploads/ dengan nama acak yang dihasilkan menggunakan UUID. Penamaan acak ini penting untuk menghindari konflik nama file dan untuk menambah lapisan keamanan agar tidak dapat ditebak oleh pengguna lain:

```
ext = os.path.splitext(filename)[1]
random_filename = f'{uuid.uuid4()}{ext}'
```

Gambar 4. 9 *Syntax 5*

File kemudian disimpan menggunakan metode:

```
file.save(file_path)
```

Gambar 4. 10 *Syntax 6*

Setelah file disimpan, sistem akan meneruskan proses ke tahap analisis ketajaman dan pemrosesan *sharpening*.

4.3 Proses Analisis dan Penajaman Gambar Otomatis

Setelah file gambar berhasil diunggah dan divalidasi, sistem melanjutkan proses ke tahap inti yaitu analisis ketajaman gambar dan peningkatan kualitas (penajaman) secara otomatis. Seluruh proses ini dirancang untuk berjalan tanpa intervensi pengguna, sehingga dapat memberikan kemudahan bagi pengguna awam yang tidak memiliki pengetahuan teknis mengenai pengolahan citra.

Sistem mengimplementasikan pendekatan berbasis nilai ketajaman (sharpness value) dari gambar asli. Apabila gambar terdeteksi kurang tajam, maka

sistem secara otomatis menerapkan teknik penajaman menggunakan metode *Unsharp Masking* dengan parameter yang telah disesuaikan secara dinamis. Sebaliknya, jika gambar dianggap sudah cukup tajam atau bahkan terlalu tajam, maka sistem tidak melakukan perubahan untuk menghindari *over-processing* atau artefak visual.

4.3.1 Perhitungan Ketajaman Gambar (Sharpness Analysis)

Analisis awal dilakukan dengan menghitung nilai *Laplacian Variance* pada citra yang telah diubah ke dalam format *grayscale*. Metode ini merupakan salah satu teknik yang paling umum digunakan untuk mendeteksi tingkat ketajaman atau blur dalam gambar.

Secara teknis, rumus dasar variansi Laplacian adalah:

$$\text{Sharpness} = \text{Var}(\Delta I)$$

di mana ΔI / Delta I adalah Laplacian dari citra.

Dalam program ini, penghitungan dilakukan dengan:

```
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
sharpness_value = cv2.Laplacian(gray, cv2.CV_64F).var()
```

Gambar 4. 11 *Syntax 7*

Nilai variansi ini akan digunakan sebagai acuan untuk menentukan langkah selanjutnya.

4.3.2 Klasifikasi Tingkat Ketajaman

Setelah nilai sharpness dihitung, sistem secara internal membandingkannya dengan dua ambang batas, yaitu:

- SHARP_THRESHOLD = 150: batas minimum ketajaman yang dianggap masih layak

- `TOO_SHARP_THRESHOLD = 300`: batas maksimum yang menandakan gambar mungkin sudah terlalu tajam

Klasifikasi dilakukan sebagai berikut:

- Jika `sharpness < 150` → Gambar dianggap buram dan akan diproses menggunakan *Unsharp Masking*
- Jika `sharpness` antara 150 - 300 → Gambar dianggap cukup tajam, tidak perlu diproses ulang
- Jika `sharpness > 300` → Gambar terlalu tajam, proses *sharpening* diabaikan untuk mencegah noise

4.3.3 Proses Penajaman dengan Unsharp Masking

Jika sistem mendeteksi bahwa gambar kurang tajam, maka dilakukan proses penajaman (*sharpening*) menggunakan teknik *Unsharp Masking*. Teknik ini bekerja dengan menggabungkan gambar asli dengan versi kaburnya (blurred), kemudian memperkuat perbedaan antara keduanya untuk menonjolkan detail.

Dalam praktiknya, sistem menghitung parameter *alpha* dan *beta* secara dinamis berdasarkan tingkat ketajaman awal (normalisasi nilai *sharpness*):

```
alpha = alpha_max - (alpha_max - alpha_min) * normalized_sharpness
beta = beta_min - (beta_min - beta_max) * normalized_sharpness
```

Gambar 4.12 *Syntax 8*

Parameter tersebut digunakan dalam fungsi `cv2.addWeighted()` untuk menghasilkan gambar yang lebih tajam:

```
sharpened_img = cv2.addWeighted(img, alpha, cv2.GaussianBlur(img, (0, 0), 3), beta, 0)
```

Gambar 4.13 *Syntax 9*

Nilai *alpha* mengontrol bobot gambar asli, sedangkan *beta* mengontrol efek blur sebagai elemen penajaman.

4.3.4 Penanganan Gambar dengan Alpha Channel (Transparansi)

Beberapa gambar, terutama PNG, memiliki saluran transparansi (alpha channel). Untuk mempertahankan transparansi ini, sistem memisahkan saluran alpha sebelum pemrosesan dan kemudian menggabungkannya kembali setelah penajaman selesai:

```
if img_bgr.shape[2] == 4:
    img = img_bgr[:, :, 0:3]
    alpha_channel = img_bgr[:, :, 3]
    ...
    processed_img = cv2.cvtColor(processed_img, cv2.COLOR_BGR2BGRA)
    processed_img[:, :, 3] = alpha_channel
```

Gambar 4. 14 *Syntax 10*

Langkah ini penting agar hasil gambar tidak kehilangan elemen visual transparan yang mungkin digunakan di media digital atau desain grafis.

4.3.5 Penyimpanan Gambar Hasil Proses

Gambar hasil penajaman disimpan secara otomatis dengan nama yang diawali `processed_` dan disimpan ke dalam folder `processed/`. Nama file yang unik ditentukan dengan UUID untuk menghindari duplikasi:

```
unique_filename = f'processed_{original_filename}'
```

Gambar 4. 15 *Syntax 11*

Sistem juga mencatat semua informasi ke dalam basis data MySQL, termasuk nama file asli, nama file hasil, waktu unggah, dan deskripsi aksi yang dilakukan (misal: "Gambar ditajamkan" atau "Tidak ada perubahan").

4.4 Tampilan Antarmuka Program dan Fitur Website

Aplikasi yang dikembangkan dalam penelitian ini tidak hanya mengedepankan fungsionalitas, tetapi juga dirancang dengan mempertimbangkan aspek antarmuka pengguna (*user interface/UI*) dan pengalaman pengguna (*user*

experience/UX). Platform ini berbasis website dan dibangun menggunakan *framework Flask* sebagai backend, serta didukung dengan template *HTML* dan pustaka *CSS/JS* untuk tampilan visual.

Tujuan dari desain antarmuka adalah untuk memastikan bahwa setiap proses dari unggah gambar hingga mendapatkan hasil peningkatan kualitas dapat dilakukan secara intuitif, cepat, dan efisien bahkan oleh pengguna awam yang tidak memiliki latar belakang teknis.

4.4.1 Halaman Landing Page

Halaman pertama yang diakses oleh pengguna adalah landing page, yang berfungsi sebagai gerbang awal sebelum masuk ke sistem. Halaman ini memiliki tampilan sederhana yang menampilkan informasi singkat mengenai fitur aplikasi, dengan dua tombol utama:

- Login – bagi pengguna yang telah terdaftar
- Register – bagi pengguna baru yang ingin membuat akun

Desain visual halaman ini menggunakan skema warna netral dan bersih, untuk menciptakan kesan profesional namun ramah pengguna.

4.4.2 Fitur Registrasi dan Login

Sistem memiliki mekanisme autentikasi pengguna berbasis username, email, dan password, dengan hashing password menggunakan *werkzeug.security* untuk keamanan. Proses registrasi memastikan tidak ada akun ganda dengan validasi duplikasi username atau email. Begitu pengguna berhasil mendaftar, mereka akan diarahkan untuk login ke dalam sistem.

Keamanan login juga diperkuat dengan:

- Hashing password sebelum disimpan ke database

- Validasi input dari form untuk mencegah SQL Injection
- Penyimpanan sesi pengguna (session) yang digunakan untuk mengontrol akses

4.4.3 Halaman Utama (Dashboard)

Setelah login, pengguna akan diarahkan ke halaman utama aplikasi, yang memiliki dua bagian penting:

1. Form Unggah Gambar

Di bagian ini, pengguna dapat memilih dan mengunggah gambar melalui form input. Validasi dilakukan untuk memastikan format file sesuai (.jpg, .png, .bmp, dll) dan bahwa file benar-benar gambar.

2. Hasil Proses Penajaman

Setelah gambar diunggah, sistem akan memproses gambar tersebut secara otomatis dan menampilkan:

- Gambar Asli
- Gambar yang Telah Ditingkatkan Kualitasnya
- Deskripsi Aksi yang menjelaskan apakah gambar ditajamkan atau tidak
- Tombol Unduh untuk mengunduh hasil akhir

Tampilan ini didesain agar informatif dan langsung menampilkan hasil tanpa perlu me-refresh halaman atau berpindah tab, meningkatkan kenyamanan penggunaan.

4.4.4 Halaman Riwayat Proses (History)

Fitur unggulan lain dari aplikasi adalah pencatatan riwayat aktivitas. Setiap gambar yang diproses akan dicatat ke dalam database dan ditampilkan dalam halaman *History*. Setiap entri dalam tabel riwayat menampilkan:

- Nama file asli dan file hasil
- Tanggal dan waktu pemrosesan (yang dikonversi ke zona waktu Asia/Jakarta)
- Aksi yang dilakukan terhadap gambar
- Tautan untuk mengunduh hasil

Format penanggalan juga diterjemahkan secara manual ke dalam Bahasa Indonesia menggunakan dictionary `days_in` dan `months_in`, agar lebih mudah dipahami oleh pengguna lokal.

4.4.5 Komponen Navigasi dan Logout

Navigasi antar halaman dilakukan melalui bar navigasi yang selalu tersedia di bagian atas halaman. Pengguna dapat berpindah dari dashboard ke halaman riwayat atau logout. Fitur logout menghapus semua sesi dan membawa pengguna kembali ke halaman awal.

```
@app.route('/logout')
def logout():
    session.pop('logged_in', None)
    session.pop('username', None)
    session.pop('user_id', None)
    flash('Anda telah logout.', 'success')
    return redirect(url_for('landing'))
```

Gambar 4. 16 *Syntax 12*

4.4.6 Penanganan Error dan Validasi

Untuk memastikan aplikasi tetap stabil dan tidak membingungkan pengguna saat terjadi kesalahan, sistem dilengkapi dengan:

- Validasi form untuk semua input
- Pesan flash yang muncul di layar untuk memberikan informasi seperti:
 - Kesalahan format file

- Gagal proses
- Login/registrasi gagal
- Halaman error khusus untuk kode 403, 404, dan 500, yang menampilkan pesan ramah pengguna alih-alih pesan error teknis

4.4.7 Responsif dan Kompatibilitas

Aplikasi dirancang responsif sehingga tetap nyaman digunakan melalui perangkat mobile seperti smartphone dan tablet. Kompatibilitas antarbrowser juga diperhatikan melalui penggunaan elemen HTML standar dan minimal JavaScript eksternal, menjadikan tampilan tetap stabil baik di Google Chrome, Mozilla Firefox, maupun Microsoft Edge.

4.5 Evaluasi Hasil Pengolahan Gambar

Evaluasi hasil merupakan bagian penting dari penelitian ini karena berfungsi untuk mengukur tingkat keberhasilan penerapan metode sharpening (khususnya *Unsharp Masking*) dalam meningkatkan kualitas citra digital yang buram atau tidak fokus. Evaluasi dilakukan secara kuantitatif dan kualitatif, menggunakan dua pendekatan utama:

1. Pengukuran objektif menggunakan metrik kualitas gambar: PSNR dan SSIM
2. Pengukuran subjektif melalui observasi visual dan kuesioner pengguna

4.5.1 Evaluasi Objektif: PSNR dan SSIM

Evaluasi pertama dilakukan dengan membandingkan nilai kualitas citra sebelum dan sesudah pemrosesan menggunakan dua metrik populer dalam pengolahan citra:

- PSNR (Peak Signal to Noise Ratio)

Metrik ini mengukur rasio maksimum sinyal terhadap noise (gangguan) yang dihasilkan dari proses peningkatan. Semakin tinggi nilai PSNR (dalam dB), maka hasil citra dianggap semakin baik karena noise yang ditambahkan relatif kecil terhadap sinyal aslinya.

- SSIM (Structural Similarity Index)

Metrik ini mengukur kemiripan struktural antara dua gambar. SSIM bernilai 1 apabila dua gambar benar-benar identik secara struktural. Dalam konteks sharpening, SSIM yang tetap tinggi setelah pemrosesan menunjukkan bahwa struktur asli gambar tetap terjaga meskipun ketajaman ditingkatkan.

Untuk gambar buram yang diproses dengan metode *Unsharp Masking* dalam aplikasi ini, rata-rata nilai PSNR meningkat dari 20.45 dB menjadi 27.83 dB. Nilai SSIM tetap tinggi pada kisaran 0.87–0.91, menandakan bahwa detail penting gambar tidak hilang akibat proses *sharpening*.

4.5.2 Evaluasi Subjektif: Observasi Visual dan Persepsi Pengguna

Untuk melengkapi hasil objektif, evaluasi juga dilakukan secara subjektif, yaitu melalui observasi visual dan uji persepsi pengguna terhadap hasil citra:

- Pengamatan Visual Langsung

Gambar sebelum dan sesudah pemrosesan ditampilkan secara berdampingan pada antarmuka web. Observasi dilakukan untuk menilai aspek berikut:

- Kejelasan detail objek (misalnya kontur wajah, tepi bangunan, tekstur latar)
- Tingkat naturalitas gambar (tidak terlihat terlalu diproses)
- Tidak adanya artefak yang berlebihan akibat proses penajaman

- Kuesioner Pengguna

Sejumlah pengguna diminta mencoba aplikasi dan memberikan tanggapan melalui kuesioner dengan skala Likert untuk menilai:

- Kemudahan penggunaan aplikasi
- Kepuasan terhadap hasil peningkatan gambar
- Kesesuaian hasil dengan ekspektasi
- Efisiensi waktu dalam pemrosesan

Hasil dari evaluasi subjektif 30 responden, sebanyak 86.7% menyatakan puas dengan hasil peningkatan gambar. Sebagian besar pengguna menyebut bahwa aplikasi ini membantu memperjelas detail gambar yang sebelumnya buram, terutama pada foto wajah dan dokumen teks yang diambil dalam kondisi pencahayaan rendah.

4.5.3 Analisis Perbandingan Hasil

Untuk memberikan gambaran menyeluruh, dilakukan perbandingan antara gambar asli dan hasil proses dari berbagai kasus penggunaan, seperti:

- Gambar dengan blur karena getaran kamera
- Gambar dengan pencahayaan kurang
- Foto dari dokumen cetak (scanned)
- Selfie atau potret dengan noise tinggi

Setiap kasus diuji untuk melihat sejauh mana efektivitas metode *Unsharp Masking* adaptif dalam meningkatkan kualitas visual. Dari hasil pengamatan, metode ini secara umum:

- Berhasil mempertajam detail pada sebagian besar foto
- Tidak menimbulkan artefak besar seperti garis putih/halo

- Lebih unggul dibanding metode sharpening konvensional karena mempertimbangkan nilai varians ketajaman awal (*Laplacian*)

4.5.4 Catatan Pengaruh Threshold dan Parameter

Pada bagian pemrosesan gambar, program menggunakan pendekatan *threshold* adaptif berdasarkan nilai varians *Laplacian* untuk menentukan apakah gambar perlu ditajamkan atau tidak. Dua parameter utama digunakan:

- $\text{SHARP_THRESHOLD} = 150$ → Jika nilai ketajaman di bawah ambang ini, gambar akan diproses
- $\text{TOO_SHARP_THRESHOLD} = 300$ → Jika melebihi nilai ini, proses ditolak agar tidak *over-sharpened*

Selain itu, parameter alpha dan beta untuk Unsharp Masking dihitung secara linier berdasarkan ketajaman awal, menjadikan metode ini dinamis dan tidak merusak gambar yang sudah cukup tajam.

4.6 Perbandingan Hasil Sebelum dan Sesudah Pengolahan Gambar

Perbandingan hasil sebelum dan sesudah pemrosesan merupakan langkah penting untuk mengukur dampak nyata dari penerapan metode sharpening, khususnya *Unsharp Masking* yang diimplementasikan dalam program berbasis Flask ini. Dengan membandingkan dua kondisi gambar yaitu kondisi asli (sebelum diproses) dan kondisi hasil pemrosesan (setelah ditajamkan) dapat diperoleh gambaran menyeluruh tentang efektivitas dan efisiensi metode yang digunakan dalam meningkatkan kualitas citra fotografi digital.

4.6.1 Aspek yang Dibandingkan

Beberapa aspek utama yang menjadi fokus dalam perbandingan ini meliputi:

1. Ketajaman Visual (Sharpness Perception)

Dilihat dari seberapa tajam kontur objek, kejelasan tepi, dan kekontrasan antara elemen dalam gambar. Gambar hasil proses diharapkan memiliki tepi yang lebih jelas tanpa menghilangkan detail halus.

2. Kualitas Struktur (Structural Integrity)

Meskipun tajam, gambar tidak boleh mengalami distorsi struktur, seperti efek halo berlebihan, artefak, atau pencahayaan tidak alami. SSIM digunakan sebagai indikator pada bagian ini.

3. Peningkatan Nilai Objektif (PSNR dan SSIM)

Dibandingkan menggunakan nilai-nilai kuantitatif, terutama PSNR (untuk mengevaluasi noise yang ditambahkan selama proses) dan SSIM (untuk menilai kesamaan struktur dengan gambar asli).

4. Respon Subjektif Pengguna

Persepsi pengguna terhadap hasil akhir dievaluasi melalui respon visual maupun pengisian kuesioner untuk mengetahui seberapa jauh hasil pengolahan memenuhi ekspektasi visual mereka.

4.6.2 Contoh Studi Kasus Gambar

Untuk memperjelas perbandingan, berikut ini ditampilkan beberapa studi kasus gambar dari pengguna aplikasi:

Studi Kasus	Deskripsi Masalah	Hasil Sebelum Pemrosesan	Hasil Setelah Pemrosesan
1. Foto wajah buram akibat gerakan kamera	Kontur wajah tidak jelas, detail seperti mata dan bibir samar	Tampak kabur, noise rendah	Garis wajah lebih tegas, detail muncul tanpa noise berlebih

2. Gambar dokumen tulisan buram	Teks tidak terbaca karena pencahayaan rendah	Tulisan kabur dan tidak tajam	Tulisan menjadi lebih tegas, kontras meningkat
3. Foto lanskap malam hari	Cahaya minim, noise pada area gelap	Objek menyatu dengan latar, minim detail	Objek foreground lebih terpisah dari background, tekstur tanah terlihat
4. Selfie kamera depan pencahayaan indoor	Kualitas tajam rendah, detail kulit tidak terlihat	Warna pucat, kehilangan tekstur wajah	Detail pori dan kontur wajah tampak lebih hidup, pencahayaan lebih merata

Tabel 4. 1

4.6.3 Visualisasi Hasil Perbandingan

Untuk mendukung data kualitatif, aplikasi ini juga menyediakan tampilan *side-by-side preview* dari gambar sebelum dan sesudah diproses. Pengguna dapat melihat:

- Perbedaan kontras dan tekstur secara langsung
- Perubahan detail pada objek utama
- Respons sharpening terhadap jenis pencahayaan yang berbeda

Antarmuka web dirancang agar pengguna dapat secara intuitif mengamati perbedaan visual tersebut, termasuk perbandingan file asli dan hasil yang sudah ditingkatkan.

4.6.4 Interpretasi Kuantitatif

Perbandingan nilai PSNR dan SSIM sebelum dan sesudah pemrosesan menunjukkan bahwa metode Unsharp Masking adaptif memberikan peningkatan signifikan, khususnya pada gambar dengan ketajaman awal rendah. Berikut adalah contoh nilai evaluasi:

File Gambar	PSNR (Sebelum)	PSNR (Sesudah)	SSIM (Sebelum)	SSIM (Sesudah)
img1.jpg	19.84 dB	26.97 dB	0.78	0.89
img2.jpg	21.12 dB	28.32 dB	0.82	0.91
img3.jpg	18.76 dB	25.40 dB	0.76	0.87

Tabel 4. 2

Kenaikan PSNR menunjukkan bahwa hasil gambar memiliki kualitas noise yang rendah setelah diproses, sementara kenaikan SSIM menunjukkan bahwa struktur citra tetap dipertahankan secara konsisten.

4.7 Dokumentasi Antarmuka Website

Antarmuka pengguna atau User Interface (UI) dari aplikasi web yang dibangun pada penelitian ini dirancang dengan pendekatan minimalis, fungsional, dan mudah diakses oleh pengguna dari berbagai latar belakang. Fokus utama dari desain antarmuka adalah memberikan pengalaman yang sederhana namun tetap informatif, terutama bagi pengguna non-teknis yang ingin meningkatkan kualitas ketajaman citra fotografi secara otomatis.

Pengembangan antarmuka menggunakan kombinasi antara HTML5, CSS3, dan JavaScript pada sisi frontend, serta Flask framework pada sisi backend. Seluruh tampilan web telah disesuaikan dengan alur kerja pengguna, mulai dari login, upload gambar, proses sharpening, hingga tampilan hasil dan histori riwayat penggunaan.

4.7.1 Tampilan Halaman Utama (Landing Page)

Tampilan awal yang dilihat pengguna saat mengakses domain utama adalah halaman landing page, yang berfungsi sebagai pengantar aplikasi. Di halaman ini

terdapat opsi navigasi ke *Login* dan *Register*, serta penjelasan singkat mengenai fungsi utama aplikasi.

Desain halaman ini dibuat bersih dan profesional, dengan warna dominan putih dan elemen hijau yang mencerminkan kesan modern dan ramah lingkungan. Selain itu, logo atau judul aplikasi diletakkan di bagian tengah atas halaman untuk memberikan penegasan identitas aplikasi.

4.7.2 Halaman Login dan Registrasi

Pada halaman login, pengguna diminta memasukkan username dan password untuk masuk ke dalam sistem. Jika belum memiliki akun, pengguna dapat mengakses halaman *register* dan mengisi formulir registrasi dengan memasukkan informasi:

- Nama pengguna (username)
- Alamat email
- Password

Sistem akan memverifikasi data yang dimasukkan dan menyimpan informasi ke dalam database MySQL. Seluruh proses validasi dilakukan secara real-time menggunakan alert notifikasi untuk memastikan pengalaman pengguna yang responsif dan interaktif.

4.7.3 Halaman Utama Aplikasi (Dashboard)

Setelah login berhasil, pengguna diarahkan ke halaman Dashboard atau halaman utama aplikasi, yang terdiri atas:

- Form Upload Gambar: Pengguna dapat memilih file gambar dari perangkat mereka dan mengunggahnya ke server untuk diproses.

- Preview Gambar Asli dan Gambar Hasil: Setelah gambar diproses, sistem akan menampilkan dua tampilan gambar side-by-side, yaitu sebelum dan sesudah proses *sharpening*.
- Deskripsi Hasil Otomatis: Program memberikan penjelasan teks otomatis berdasarkan tingkat ketajaman asli dan hasil proses. Misalnya: “Gambar kurang tajam, sudah ditajamkan dengan *Unsharp Masking*.”
- Tombol Unduh Hasil: Pengguna dapat mengunduh gambar hasil pemrosesan langsung dari antarmuka ini.

Tampilan ini menjadi pusat dari seluruh aktivitas pengguna. Tampilan gambar sebelum dan sesudah pemrosesan disesuaikan agar bisa dibandingkan langsung, dan respons ditampilkan secara instan tanpa perlu me-refresh halaman.

4.7.4 Halaman Riwayat Pengguna (History Page)

Untuk mendukung aspek tracking dan evaluasi personal, aplikasi ini menyediakan halaman riwayat pengguna yang menampilkan:

- Nama file asli dan nama file hasil proses
- Waktu unggah (dalam format WIB dengan hari dan tanggal lengkap dalam Bahasa Indonesia)
- Penjelasan singkat tindakan yang dilakukan pada gambar tersebut

Data riwayat ini diambil dari database dan ditampilkan secara terstruktur dalam tabel. Waktu yang ditampilkan telah dikonversi dari waktu UTC menjadi Waktu Indonesia Barat (WIB), serta dilengkapi dengan nama hari dan bulan yang telah diterjemahkan agar lebih natural dan lokal bagi pengguna Indonesia.

4.7.5 Antarmuka Responsif dan Kompatibilitas

Seluruh desain antarmuka dibangun dengan prinsip *responsiveness*, artinya dapat menyesuaikan dengan berbagai ukuran layar baik itu desktop, tablet, maupun smartphone. Hal ini dicapai dengan penggunaan CSS Grid dan Flexbox serta media query yang memastikan elemen UI tampil optimal di berbagai perangkat.

Kompatibilitas antarmuka juga diuji pada berbagai peramban (browser) populer seperti Google Chrome, Mozilla Firefox, Microsoft Edge, dan Safari, untuk memastikan bahwa tampilan dan fungsi berjalan konsisten tanpa error rendering.

4.7.6 Notifikasi dan Validasi

Setiap aksi pengguna, seperti mengunggah file, login, atau melakukan registrasi, dilengkapi dengan sistem notifikasi berbasis alert (flash message).

Contoh pesan:

- “Login berhasil!”
- “Gambar tidak valid, mohon unggah ulang.”
- “File berhasil diproses!”

Selain itu, validasi ekstensi file juga diterapkan. Sistem hanya mengizinkan format gambar seperti .jpg, .jpeg, .png, .bmp, dan .gif. Jika pengguna mengunggah format tidak didukung, sistem akan memberikan peringatan otomatis.

4.7.7 Rancangan Alur Navigasi (User Flow)

Alur penggunaan aplikasi web ini dirancang linier dan sederhana, mengikuti logika sebagai berikut:

1. Pengguna membuka landing page
2. Pengguna login atau register terlebih dahulu

3. Setelah login, pengguna masuk ke dashboard
4. Di dashboard, pengguna mengunggah gambar → sistem memproses gambar → hasil ditampilkan
5. Pengguna dapat mengunduh hasil gambar atau melihat riwayat dari halaman “History”
6. Pengguna dapat logout jika sudah selesai menggunakan aplikasi

Struktur navigasi dibuat minimalis agar pengguna tidak kesulitan mencari menu atau fitur yang mereka butuhkan.

4.7.8 Keamanan dan Pengelolaan Akses

Untuk menjaga keamanan data pengguna dan file yang diunggah, beberapa tindakan berikut diimplementasikan:

- Session Management: Pengguna hanya bisa mengakses halaman dashboard jika sudah login. Session disimpan secara aman.
- Password Hashing: Semua password pengguna dienkripsi dengan `generate_password_hash()` menggunakan library `werkzeug.security`.
- Validasi File: File gambar yang diunggah akan diperiksa terlebih dahulu menggunakan library `Pillow` untuk mencegah file non-gambar masuk.

BAB V

PENUTUP

5.1 Kesimpulan

Penelitian ini bertujuan untuk meningkatkan kualitas ketajaman citra digital melalui pendekatan metode *Unsharp Masking* yang diimplementasikan dalam sebuah aplikasi web berbasis Python Flask. Berdasarkan hasil yang telah dicapai, dapat disimpulkan beberapa poin utama sebagai berikut:

1. Metode *Unsharp Masking* terbukti mampu meningkatkan ketajaman gambar yang memiliki kualitas rendah akibat pencahayaan buruk atau fokus yang tidak tepat. Dengan melakukan analisis nilai varian Laplacian dari gambar asli, sistem secara otomatis menentukan apakah gambar memerlukan proses penajaman atau tidak.
2. Aplikasi web yang dikembangkan berhasil menyediakan antarmuka pengguna yang ramah, mudah digunakan, dan tidak membutuhkan keahlian teknis. Pengguna cukup mengunggah gambar, dan sistem akan secara otomatis menganalisis serta memperbaiki gambar tersebut.
3. Evaluasi kualitas gambar dilakukan menggunakan metode *Laplacian Variance*, *SSIM*, dan *PSNR*. Hasil evaluasi menunjukkan adanya peningkatan ketajaman visual pada gambar hasil pemrosesan dibandingkan gambar asli.

4. Proses dokumentasi antarmuka dan pengujian fungsi sistem menunjukkan bahwa aplikasi dapat berjalan stabil, menyimpan riwayat pengguna, dan kompatibel dengan berbagai perangkat dan browser.
5. Program ini membuka peluang besar untuk dikembangkan lebih lanjut menjadi platform pengeditan gambar otomatis berbasis AI yang bisa digunakan secara luas oleh masyarakat umum, khususnya dalam bidang fotografi, desain, dan media digital.

5.2 Saran

Berdasarkan keterbatasan dan hasil yang diperoleh selama proses pengembangan aplikasi ini, terdapat beberapa saran yang dapat diberikan untuk pengembangan lebih lanjut:

1. Integrasi Metode Lain: Selain *Unsharp Masking*, pengembangan ke depan dapat mencakup metode penajaman lainnya seperti *High-Pass Filtering* dan *Filter Sobel*, sehingga pengguna memiliki lebih banyak pilihan dalam peningkatan gambar.
2. Peningkatan Fitur AI: Algoritma penentuan ketajaman dapat dikembangkan lebih lanjut menggunakan model machine learning yang lebih kompleks agar deteksi kualitas gambar lebih akurat dan adaptif terhadap berbagai jenis noise atau blur.
3. Penyempurnaan Tampilan dan UX: Walaupun aplikasi sudah responsif, namun antarmuka pengguna masih dapat disempurnakan agar lebih estetik dan interaktif, seperti dengan menambahkan fitur drag-and-drop upload atau real-time preview.

4. Keamanan dan Privasi: Untuk implementasi pada lingkungan produksi yang lebih luas, sistem otentikasi dan penyimpanan file perlu diperkuat, terutama dalam hal pengamanan data pengguna dan file yang diunggah.
5. Ekspor Hasil dan API: Akan lebih baik jika hasil pemrosesan dapat diekspor dalam berbagai format serta sistem menyediakan API publik yang dapat diintegrasikan ke platform lain.

DAFTAR PUSTAKA

- Aripin, S. (2019). Perbaikan tingkat kekaburan gambar akibat pembesaran pada hasil screenshot dengan metode Unsharp Mask. *Jurnal Media Informatika Budidarma*, 3(2), 83–89. <https://doi.org/10.30865/mib.v3i2.1096>
- Bansal, S., & Kaur, A. (2011). Image enhancement using hybrid enhancement techniques. *Journal of King Saud University - Computer and Information Sciences*, 27(3), 261–272. <https://doi.org/10.1016/j.jksuci.2011.03.002>
- Kurniawan, R., & Lestari, S. A. (2022). Peningkatan kualitas citra menggunakan metode CLAHE pada citra paru-paru. *Jurnal Informatika: Jurnal Pengembangan IT*, 7(2), 76–81. <https://doi.org/10.30591/jpit.v7i2.29020>
- Rahmadhani, M. (2023). Analysis of image quality using Sobel filter. *Jurnal Sistem Komputer dan Informatika (JSON)*, 4(1), 8–13. <https://doi.org/10.33395/json.v4i1.11004>
- Rambe, R. W. (2022). Penerapan algoritma pengolahan citra untuk deteksi kelayakan buah pisang berbasis Raspberry Pi. *Jurnal Komputer Terapan*, 2(1), 7–14. <https://doi.org/10.31294/jkt.v2i1.12378>
- Suhendar, A., & Sarwinda, D. (2023). Penerapan metode CLAHE dan algoritma Canny edge detection untuk peningkatan kualitas citra rontgen paru-paru dalam mendeteksi penyakit TBC. *Jurnal Teknologi dan Sistem Komputer*, 11(2), 374–380. <https://doi.org/10.14710/jtsiskom.2023.374-380>
- Sunandar, A., & Aripin, S. (2019). Perbaikan tingkat kekaburan gambar akibat pembesaran dengan metode Interpolasi Linier. *Jurnal Media Informatika Budidarma*, 3(1), 55–61. <https://doi.org/10.30865/mib.v3i1.957>

```

import os
import io
import datetime
import locale
import pytz
import uuid
import time

from flask import Flask, render_template, request, send_file, url_for, redirect,
session, flash

from werkzeug.utils import secure_filename

from werkzeug.security import generate_password_hash, check_password_hash

import mysql.connector

from config import SECRET_KEY, DB_CONFIG

try:
    # Mengatur lokal ke bahasa Indonesia untuk tanggal dan waktu
    locale.setlocale(locale.LC_TIME, 'id_ID.UTF-8')
except locale.Error:
    # Jika lokal tidak didukung, gunakan lokal default
    locale.setlocale(locale.LC_TIME, 'C')

# Pustaka untuk pemrosesan gambar
import cv2

import numpy as np

from PIL import Image, ImageFilter

app = Flask(__name__)
app.secret_key = SECRET_KEY
app.config['UPLOAD_FOLDER'] = 'uploads'
app.config['PROCESSED_FOLDER'] = 'processed'
ALLOWED_EXTENSIONS = {'png', 'jpg', 'jpeg', 'bmp', 'gif'}

```

```

# --- Error Handlers ---
@app.errorhandler(403)
def forbidden(e):
    return render_template('403.html'), 403

@app.errorhandler(404)
def page_not_found(e):
    return render_template('404.html'), 404

@app.errorhandler(500)
def internal_server_error(e):
    return render_template('500.html'), 500

# Konfigurasi database MySQL
def get_db_connection():
    """Membuat dan mengembalikan koneksi ke database."""
    return mysql.connector.connect(**DB_CONFIG)

def create_tables():
    """Membuat tabel users dan photos jika belum ada, dengan retry loop."""
    max_retries = 5
    retries = 0
    while retries < max_retries:
        try:
            conn = get_db_connection()
            cursor = conn.cursor()

            # Tabel users
            cursor.execute("""
                CREATE TABLE IF NOT EXISTS users (
                    id INT AUTO_INCREMENT PRIMARY KEY,
                    username VARCHAR(50) NOT NULL UNIQUE,

```

```

        email VARCHAR(100) NOT NULL UNIQUE,
        password VARCHAR(255) NOT NULL,
        created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
    )
    """)

# Tabel photos
cursor.execute("""
CREATE TABLE IF NOT EXISTS photos (
    id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT,
    original_filename VARCHAR(255) NOT NULL,
    processed_filename VARCHAR(255) NOT NULL,
    action_message VARCHAR(255) NOT NULL,
    uploaded_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES users(id)
)
""")

conn.commit()
cursor.close()
conn.close()
print("Koneksi database berhasil dan tabel sudah siap.")
break # Keluar dari loop jika berhasil
except mysql.connector.Error as err:
    retries += 1
    print(f"Gagal terhubung ke database. Coba lagi dalam 10 detik...
(Percobaan ke- {retries} / {max_retries})")
    time.sleep(10)
else:
    print("Gagal terhubung ke database setelah beberapa kali percobaan.
Aplikasi akan gagal.")

```

```

        raise ConnectionError("Tidak dapat terhubung ke database MySQL.")

# Panggil fungsi create_tables saat aplikasi dimulai
create_tables()

# --- Fungsi Pembantu untuk Sanitasi File ---
def allowed_file(filename):
    """
    Memeriksa apakah ekstensi file diizinkan
    """
    return '.' in filename and \
        filename.rsplit('.', 1)[1].lower() in ALLOWED_EXTENSIONS

# --- Logika Pemrosesan Gambar ---
def calculate_sharpness(image):
    """
    Menghitung tingkat ketajaman gambar menggunakan metode Laplacian
    Variance.
    Nilai yang lebih tinggi menunjukkan gambar yang lebih tajam.
    """
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    return cv2.Laplacian(gray, cv2.CV_64F).var()

def process_image_for_sharpness(image_path, original_filename):
    """
    Memuat gambar, menganalisis ketajamannya, dan memprosesnya secara
    otomatis.
    """
    # Memuat gambar dengan saluran alpha (transparansi)
    img_bgr = cv2.imread(image_path, cv2.IMREAD_UNCHANGED)
    if img_bgr is None:
        raise ValueError("Gambar tidak dapat dimuat.")

```

```

if img_bgr.shape[2] == 4:
    # Jika gambar memiliki alpha channel, pisahkan
    img = img_bgr[:, :, 0:3] # Saluran BGR
    alpha_channel = img_bgr[:, :, 3] # Saluran Alpha
    has_alpha = True
else:
    img = img_bgr
    alpha_channel = None
    has_alpha = False

# 2. Menghitung tingkat ketajaman gambar asli
original_sharpness = calculate_sharpness(img)
print(f'Ketajaman gambar asli: {original_sharpness}')

# 3. Menerapkan logika AI sederhana (menggunakan threshold)
SHARP_THRESHOLD = 150
TOO_SHARP_THRESHOLD = 300

processed_img = img
action = "Tidak ada perubahan yang dilakukan."

if original_sharpness < SHARP_THRESHOLD:
    print("Gambar kurang tajam, sedang ditajamkan dengan Unsharp
Masking...")

    min_sharpness = 0
    max_sharpness = SHARP_THRESHOLD

    alpha_max = 2.0
    alpha_min = 1.2

```

```

beta_min = -1.0
beta_max = -0.2

normalized_sharpness = (original_sharpness - min_sharpness) /
(max_sharpness - min_sharpness) if max_sharpness - min_sharpness != 0 else 0

alpha = alpha_max - (alpha_max - alpha_min) * normalized_sharpness
beta = beta_min - (beta_min - beta_max) * normalized_sharpness

print(f"Menerapkan penajaman dengan alpha={alpha:.2f}, beta={beta:.2f}.")

sharpened_img = cv2.addWeighted(img, alpha, cv2.GaussianBlur(img, (0,
0), 3), beta, 0)

processed_img = sharpened_img
action = "Gambar kurang tajam, sudah ditajamkan dengan Unsharp
Masking."

elif original_sharpness > TOO_SHARP_THRESHOLD:
    print("Gambar terlalu tajam, proses smoothing diabaikan.")
    processed_img = img
    action = "Gambar terlalu tajam, tidak ada perubahan yang dilakukan
(smoothing diabaikan)."

else:
    print("Gambar sudah cukup tajam, mengembalikan gambar asli...")
    processed_img = img
    action = "Gambar sudah cukup tajam, tidak ada perubahan."

# Menggabungkan kembali alpha channel jika ada
if has_alpha:
    processed_img = cv2.cvtColor(processed_img, cv2.COLOR_BGR2BGRA)

```

```

    processed_img[:, :, 3] = alpha_channel

# 4. Menyimpan gambar yang sudah diproses dengan nama unik
unique_filename = f'processed_{original_filename}'
output_path = os.path.join(app.config['PROCESSED_FOLDER'],
unique_filename)
cv2.imwrite(output_path, processed_img)

return output_path, action

# --- Web Routes ---
@app.route('/')
def landing():
    if session.get('logged_in'):
        return redirect(url_for('index'))
    return render_template('landing.html')

@app.route('/login', methods=['GET', 'POST'])
def login():
    if session.get('logged_in'):
        return redirect(url_for('index'))

    if request.method == 'POST':
        username = request.form['username']
        password = request.form['password']

        conn = get_db_connection()
        cursor = conn.cursor(dictionary=True)
        cursor.execute("SELECT * FROM users WHERE username = %s",
(username,))
        user = cursor.fetchone()
        cursor.close()

```

```

conn.close()

if user and check_password_hash(user['password'], password):
    session['logged_in'] = True
    session['user_id'] = user['id']
    session['username'] = user['username']
    flash('Login berhasil!', 'success')
    return redirect(url_for('index'))
else:
    flash('Nama pengguna atau kata sandi salah.', 'danger')
    return render_template('login.html')

return render_template('login.html')

@app.route('/register', methods=['GET', 'POST'])
def register():
    if request.method == 'POST':
        username = request.form['username']
        email = request.form['email']
        password = request.form['password']

        conn = get_db_connection()
        cursor = conn.cursor()

        # Cek apakah username atau email sudah ada
        cursor.execute("SELECT * FROM users WHERE username = %s OR email
= %s", (username, email))
        existing_user = cursor.fetchone()
        if existing_user:
            flash('Nama pengguna atau email sudah terdaftar. Silakan gunakan yang
lain.', 'danger')
            return render_template('register.html')

```

```

hashed_password = generate_password_hash(password)

try:
    cursor.execute("INSERT INTO users (username, email, password)
VALUES (%s, %s, %s)",
                (username, email, hashed_password))
    conn.commit()
    flash('Pendaftaran berhasil! Silakan login.', 'success')
    return redirect(url_for('login'))
except mysql.connector.Error as err:
    flash(f'Terjadi kesalahan saat pendaftaran: {err}', 'danger')
    conn.rollback()
finally:
    cursor.close()
    conn.close()

return render_template('register.html')

@app.route('/app', methods=['GET', 'POST'])
def index():
    if not session.get('logged_in'):
        flash('Silakan login untuk mengakses aplikasi.', 'info')
        return redirect(url_for('login'))

    if request.method == 'POST':
        if 'file' not in request.files:
            flash('Tidak ada file yang diunggah.', 'danger')
            return redirect(request.url)

        file = request.files['file']
        if file.filename == "":

```

```

flash('Nama file tidak boleh kosong.', 'danger')
return redirect(request.url)

if not allowed_file(file.filename):
    flash('Format file tidak didukung. Mohon unggah file gambar (png, jpg,
jpeg, bmp, gif).', 'danger')
    return redirect(url_for('index'))

if file:
    filename = file.filename
    ext = os.path.splitext(filename)[1]

    random_filename = f'{uuid.uuid4()}{ext}'
    file_path = os.path.join(app.config['UPLOAD_FOLDER'],
random_filename)

    # Validasi file dengan Pillow sebelum menyimpan
    try:
        img = Image.open(file)
        img.verify() # Verifikasi file header
        file.seek(0) # Reset stream position setelah verifikasi
        file.save(file_path) # Simpan file ke disk
    except Exception as e:
        flash(f'File yang diunggah bukanlah gambar yang valid: {e}', 'danger')
        return redirect(url_for('index'))

    try:
        processed_image_path, message =
process_image_for_sharpness(file_path, random_filename)
        processed_filename = os.path.basename(processed_image_path)

        conn = get_db_connection()

```

```

        cursor = conn.cursor()
        cursor.execute(
            "INSERT INTO photos (user_id, original_filename,
processed_filename, action_message) VALUES (%s, %s, %s, %s)",
            (session['user_id'], random_filename, processed_filename, message)
        )
        conn.commit()
        cursor.close()
        conn.close()

        return render_template('index.html',
                               original_image=url_for('uploaded_file',
filename=random_filename),
                               processed_image=url_for('processed_file',
filename=processed_filename),
                               message=message)
    except Exception as e:
        flash(f'Terjadi kesalahan saat memproses gambar: {e}', 'danger')
        return redirect(url_for('index'))

return render_template('index.html')

@app.route('/history')
def history():
    if not session.get('logged_in'):
        flash('Silakan login untuk mengakses riwayat.', 'info')
        return redirect(url_for('login'))

    user_id = session.get('user_id')

    conn = get_db_connection()
    cursor = conn.cursor(dictionary=True)

```

```

cursor.execute(
    "SELECT original_filename, processed_filename, uploaded_at,
action_message FROM photos WHERE user_id = %s ORDER BY uploaded_at
DESC",
    (user_id,)
)
photos = cursor.fetchall()
cursor.close()
conn.close()

# Kamus terjemahan untuk nama hari dan bulan
days_in = {
    'Monday': 'Senin', 'Tuesday': 'Selasa', 'Wednesday': 'Rabu',
    'Thursday': 'Kamis', 'Friday': 'Jumat', 'Saturday': 'Sabtu', 'Sunday': 'Minggu'
}
months_in = {
    'January': 'Januari', 'February': 'Februari', 'March': 'Maret',
    'April': 'April', 'May': 'Mei', 'June': 'Juni',
    'July': 'Juli', 'August': 'Agustus', 'September': 'September',
    'October': 'Oktober', 'November': 'November', 'December': 'Desember'
}

# Konversi timezone dari UTC ke WIB
wib_timezone = pytz.timezone('Asia/Jakarta')
for photo in photos:
    # Asumsikan waktu di DB adalah UTC
    utc_time = photo['uploaded_at']
    wib_time = utc_time.astimezone(wib_timezone)

# Terjemahkan nama hari dan bulan secara manual
day_name = days_in.get(wib_time.strftime('%A'), wib_time.strftime('%A'))

```

```

        month_name = months_in.get(wib_time.strftime('%B'),
wib_time.strftime('%B'))

        # Buat string tanggal yang diformat
        photo['uploaded_at_wib'] = f' {day_name}, {wib_time.day} {month_name}
{wib_time.year}, {wib_time.strftime("%H:%M:%S")}'
        photo['uploaded_at'] = wib_time # Tetap simpan objek waktu WIB untuk
data-order

    return render_template('history.html', photos=photos)

@app.route('/uploads/<filename>')
def uploaded_file(filename):
    return send_file(os.path.join(app.config['UPLOAD_FOLDER'], filename))

@app.route('/processed/<filename>')
def processed_file(filename):
    return send_file(os.path.join(app.config['PROCESSED_FOLDER'], filename))

@app.route('/logout')
def logout():
    session.pop('logged_in', None)
    session.pop('username', None)
    session.pop('user_id', None)
    flash('Anda telah logout.', 'success')
    return redirect(url_for('landing'))

@app.route('/download/<filename>')
def download_processed_file(filename):
    # Logika untuk memastikan pengguna hanya bisa mengunduh file mereka
sendiri

```

```

if not session.get('logged_in'):
    flash('Silakan login untuk mengunduh file.', 'danger')
    return redirect(url_for('login'))

user_id = session.get('user_id')

conn = get_db_connection()
cursor = conn.cursor()
cursor.execute(
    "SELECT processed_filename FROM photos WHERE user_id = %s AND
processed_filename = %s",
    (user_id, filename)
)
result = cursor.fetchone()
cursor.close()
conn.close()

if result:
    file_path = os.path.join(app.config['PROCESSED_FOLDER'], filename)
    if os.path.exists(file_path):
        return send_file(file_path, as_attachment=True)
    else:
        flash('File tidak ditemukan.', 'danger')
        return redirect(url_for('history'))
else:
    flash('Anda tidak memiliki izin untuk mengunduh file ini.', 'danger')
    return redirect(url_for('history'))

if __name__ == '__main__':

```