

BUKU AJAR



PENGOLAHAN CITRA DIGITAL UNTUK SISTEM PENGENALAN UCAPAN AUDIO-VISUAL MENGUNAKAN PEMROGRAMAN PYTHON



Penulis :

**Muhaimin Hasanudin, M.Kom.
Dedy Prasetya Kristiadi, M.Kom.
Deni Kuswoyo, M.Kom.
Ridlan Ahmad, M.Kom.
Prof. Dr. Indrianto. Skom., M.T.**

**BUKU AJAR PENGOLAHAN CITRA
DIGITAL UNTUK SISTEM PENGENALAN
UCAPAN AUDIO-VISUAL
MENGUNAKAN PEMROGRAMAN
PYTHON**

Disusun oleh:

**Muhaimin Hasanudin, M.Kom.
Dedy Prasetya Kristiadi, M.Kom.
Deni Kuswoyo, M.Kom.
Ridlan Ahmad, M.Kom.
Prof. Dr. Indrianto. Skom., M.T.**

PENERBIT:



UU No 28 tahun 2014 tentang Hak Cipta
Pasal 113

- 1) Setiap Orang yang dengan tanpa hak melakukan pelanggaran hak ekonomi sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf i untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 1 (satu) tahun dan/atau pidana denda paling banyak Rp 100.000.000 (seratus juta rupiah).
- 2) Setiap Orang yang dengan tanpa hak dan/atau tanpa izin Pencipta atau pemegang Hak Cipta melakukan pelanggaran hak ekonomi Pencipta sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf c, huruf d, huruf f, dan/atau huruf h untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 3 (tiga) tahun dan/atau pidana denda paling banyak Rp. 500.000.000,00 (lima ratus juta rupiah).
- 3) Setiap Orang yang dengan tanpa hak dan/atau tanpa izin Pencipta atau pemegang Hak Cipta melakukan pelanggaran hak ekonomi Pencipta sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf a, huruf b, huruf e, dan/atau huruf g untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 4 (empat) tahun dan/atau pidana denda paling banyak Rp1.000.000.000,00 (satu miliar rupiah).
- 4) Setiap Orang yang memenuhi unsur sebagaimana dimaksud pada ayat (3) yang dilakukan dalam bentuk pembajakan, dipidana dengan pidana penjara paling lama 10 (sepuluh) tahun dan/atau pidana denda paling banyak Rp. 4.000.000.000,00 (empat miliar rupiah).

**BUKU AJAR PENGOLAHAN CITRA DIGITAL
UNTUK SISTEM PENGENALAN UCAPAN AUDIO-VISUAL
MENGUNAKAN PEMROGRAMAN PYTHON**

Penulis:

**Muhaimin Hasanudin, M.Kom.
Dedy Prasetya Kristiadi, M.Kom.
Deni Kuswoyo, M.Kom.
Ridlan Ahmad, M.Kom.
Prof. Dr. Indrianto. Skom., M.T.**

Editor:

Nurhadi

Desain Cover:

Sulaiman

Tata Letak:

Sulaiman

ISBN:

Cetakan Pertama:

Maret, 2026

Hak Cipta 2026, Pada Penulis

Hak Cipta Dilindungi Oleh Undang-Undang

Copyright © 2026

by HADLA Media Informasi

All Right Reserved

Dilarang keras menerjemahkan, memfotokopi, atau memperbanyak sebagian atau seluruh isi buku ini tanpa izin tertulis dari Penerbit

PENERBIT:



Website: www.media.hadlacorp.com

KATA PENGANTAR

Puji syukur ke hadirat Tuhan Yang Maha Esa atas tersusunnya Buku ajar mata kuliah Pengelolaan Citra Digital ini. Buku ajar ini dirancang sebagai panduan akademis bagi mahasiswa S1 untuk memahami konsep fundamental hingga aplikasi lanjut visi komputer secara runtut dan sistematis. Dengan memanfaatkan bahasa pemrograman Python dan pustaka OpenCV, mahasiswa diajak mengeksplorasi berbagai teknik manipulasi citra, peningkatan kualitas visual hingga sistem seperti Pengenalan Ucapan Audio-Visual (AVSR).

Penyusunan modul ini menitikberatkan pada pengalaman langsung melalui serangkaian tugas terstruktur, mulai dari representasi matriks hingga pengenalan pola. Fokus utamanya adalah membekali mahasiswa dengan keterampilan teoritis dan praktis yang relevan terhadap tantangan teknologi modern di bidang kecerdasan buatan. Semoga Buku ajar ini menjadi landasan kokoh bagi mahasiswa dalam mengembangkan solusi inovatif berbasis data visual di masa depan. Selamat bereksperimen

DAFTAR ISI

Kata Pengantar	v
Daftar Isi	vii
Bab 1 Pengolahan Citra Digital	1
A. Deskripsi Pembelajaran	1
B. Kompetensi Pembelajaran	1
1.1 Pengolahan Citra Digital	2
1.1.1 Definisi	2
1.1.2 Ruang Lingkup	3
1.1.3 Pendekatan Dalam Pengolahan Citra Digital.....	3
1.1.4 Tantangan Dalam Pengolahan Citra Digital	4
1.2 Pengolahan Citra Dalam Pengenalan Ucapan	4
1.2.1 Integrasi Pengolahan Citra Dan Pengenalan Ucapan ...	5
1.2.2 Aplikasi Pengolahan Citra Dalam Pengenalan Ucapan ...	5
1.2.3 Tantangan Dan Solusi	6
1.3 Sistem Pengenalan Ucapan Audio-Visual.	6
1.3.1 Komponen Utama Sistem AVSR	7
1.3.2 Aplikasi Sistem AVSR	7
1.3.3 Tantangan Dalam Sistem AVSR	8
1.4 Tujuan.....	8
1.5 Manfaat.....	9
1.6 Tantangan Dan Peluang.....	10
1.6.1 Peluang Dalam Pengolahan Citra	11
1.7 Tujuan Praktikum.....	12
1.7.1 Alat dan Bahan.....	12
1.7.2 Langkah Praktikum.....	13
1.8 Tugas Praktikum.....	18
1.9 Rangkuman.....	19
Daftar Pustaka.....	20
Bab 2 Pemrograman Python Untuk Pengolahan Citra	21
A. Deskripsi Pembelajaran	21
B. Kompetensi Pembelajaran	21
2.1 Pengantar Dan Lingkungan Pemrograman Python.....	22
2.1.1 Pustaka Fundamental Untuk Pengolahan Citra	23
2.1.2 Operasi Input/Output Dan Representasi Citra	24
2.1.3 Manipulasi Pixel Dan Array Dasar	25

2.1.4	Transformasi Ruang Warna	25
2.1.5	Operasi Geometri	26
2.1.6	Operasi Dasar Dalam Domain Spasial	27
2.2	Tujuan Praktikum	28
2.2.1	Alat Dan Bahan	28
2.2.2	Langkah Praktikum	29
2.3	Tugas Praktikum.....	38
2.4	Rangkuman.....	40
Daftar Pustaka.....		41
Bab 3 Operasi Dasar Dan Manipulasi Citra		42
A.	Deskripsi Pembelajaran	42
B.	Kompetensi Pembelajaran	42
3.1	Konsep Dasar Operasi Citra.....	43
3.1.1	Operasi Berbasis Titik (Point Operations)	44
3.1.2	Operasi Aritmetika Dan Logika	45
3.1.3	Transformasi Geometri	46
3.1.4	Analisis Histogram Dan Peningkatan Kontras	47
3.1.5	Operasi Ketetangaan Dan Filtering Spasial	48
3.1.6	Implementasi Praktis Dengan Python	49
3.2	Tujuan Praktikum	50
3.2.1	Alat Dan Bahan	51
3.2.2	Langkah Praktikum	51
3.3	Tugas Praktikum.....	62
3.4	Rangkuman.....	64
Daftar Pustaka.....		65
Bab 4 Histogram Dan Perbaikan Kualitas Citra		66
A.	Deskripsi Pembelajaran	66
B.	Kompetensi Pembelajaran	66
4.1	Histogram Dan Perbaikan Kualitas Citra.....	67
4.1.1	Konsep Dasar Histogram Citra	67
4.1.2	Interpretasi Histogram Untuk Diagnosis Citra.....	68
4.1.3	Teknik Perbaikan Kualitas Berbasis Histogram	69
4.1.4	Histogram Global Dan Solusi CLAHE.....	70
4.1.5	Pendekatan Retinex Dan Filtering	71
4.1.6	Evaluasi Kualitas Citra Hasil Perbaikan	72
4.1.7	Implementasi Praktis Dengan Python	73
4.2	Tujuan Praktikum	74

4.2.1 Alat Dan Bahan	74
4.2.2 Langkah Praktikum	75
4.3 Tugas Praktikum.....	89
4.4 Rangkuman.....	90
Daftar Pustaka.....	91
Bab 5 Filtering Dan Peningkatan Citra.....	92
A. Deskripsi Pembelajaran	92
B. Kompetensi Pembelajaran	92
5.1 Konsep Dasar Filtering Dan Konvolusi.....	93
5.1.1 Klasifikasi Filter Berdasarkan Sifat Matematis.....	94
5.1.2 Filter Penghalusan Untuk Reduksi Derau	95
5.1.3 Filter Penajaman Untuk Memperkuat Detail	97
5.1.4 Filtering Dalam Domain Frekuensi	98
5.1.5 Implementasi Praktis Dan Evaluasi Kinerja	100
5.2 Tujuan Praktikum	101
5.2.1 Alat Dan Bahan	101
5.2.2 Langkah Praktikum	102
5.3 Tugas Praktikum.....	114
5.4 Rangkuman.....	115
Daftar Pustaka.....	117
Bab 6 Deteksi Tepi.....	118
A. Deskripsi Pembelajaran	118
B. Kompetensi Pembelajaran	118
6.1 Konsep Dasar Deteksi Tepi.....	119
6.1.1 Operator Gradien Berbasis Turunan Pertama.....	120
6.1.2 Operator Berbasis Turunan Kedua: Laplacian	122
6.1.3 Algoritma Canny: Pendekatan Optimal	123
6.1.4 Perbandingan Kinerja Dan Aplikasii	124
6.1.5 Implementasi Praktis Dengan Python	126
6.2 Tujuan Praktikum	127
6.2.1 Alat Dan Bahan	127
6.2.2 Langkah Praktikum	127
6.3 Tugas Praktikum.....	138
6.4 Rangkuman.....	140
Daftar Pustaka.....	141

Bab 7 Segmentasi Citra	142
A. Deskripsi Pembelajaran	142
B. Kompetensi Pembelajaran	142
7.1 Konsep Dasar Dan Definisi Segmentasi.....	143
7.1.1 Segmentasi Berbasis Thresholding.....	144
7.1.2 Segmentasi Berbasis Region	146
7.1.3 Segmentasi Berbasis Tepi	148
7.1.4 Segmentasi Berbasis Active Contour	149
7.1.5 Implementasi Praktis Dengan Python	150
7.1.6 Evaluasi Hasil Segmentasi	151
7.2 Tujuan Praktikum	152
7.2.1 Alat Dan Bahan	152
7.2.2 Langkah Praktikum	152
7.3 Tugas Praktikum.....	164
7.4 Rangkuman.....	166
Daftar Pustaka.....	167
Bab 8 Kontur Dan Representasi	168
A. Deskripsi Pembelajaran	168
B. Kompetensi Pembelajaran	168
8.1 Konsep Kontur Dalam Citra Digital.....	169
8.1.1 Ekstraksi Kontur	171
8.1.2 Representasi Kontur	172
8.1.3 Chain Code.....	172
8.1.4 Polygonal Approximation	172
8.1.5 Skeleton Representation	173
8.1.6 Signature Representation	173
8.1.7 Descriptor Bentuk	174
8.2 Tujuan Praktikum	174
8.2.1 Alat Dan Bahan	174
8.2.2 Langkah Praktikum	175
8.3 Tugas Praktikum.....	186
8.4 Rangkuman.....	187
Daftar Pustaka.....	188
Bab 9 Kontur Dan Representasi	189
A. Deskripsi Pembelajaran	189
B. Kompetensi Pembelajaran	189
9.1 Konsep Dasar Pemampatan Citra.....	190

9.1.1 Pemampatan Lossless	192
9.1.2 Pemampatan Lossy	193
9.1.3 JPEG Compression	194
9.1.4 Parameter Evaluasi Pemampatan	195
9.2 Tujuan Praktikum	195
9.2.1 Alat Dan Bahan	195
9.2.2 Langkah Praktikum	196
9.3 Tugas Praktikum.....	206
9.4 Rangkuman.....	207
Daftar Pustaka.....	207
Bab 10 Citra Biner.....	208
A. Deskripsi Pembelajaran	208
B. Kompetensi Pembelajaran	208
10.1 Konsep Dasar Citra Biner.....	209
10.1.1 Proses Konversi Citra Grayscale ke Citra Biner	210
10.1.2 Representasi Matriks Citra Biner	211
10.1.3 Operasi Morfologi Pada Citra Biner	212
10.1.4 Analisis Komponen Terhubung	212
10.1.5 Implementasi Python Untuk Citra Biner	213
10.2 Tujuan Praktikum	214
10.2.1 Alat Dan Bahan	214
10.2.2 Langkah Praktikum	214
10.3 Tugas Praktikum.....	224
10.4 Rangkuman.....	225
Daftar Pustaka.....	226
Bab 11 Warna.....	227
A. Deskripsi Pembelajaran	227
B. Kompetensi Pembelajaran	227
11.1 Konsep Dasar Warna Dalam Citra Digital.....	228
11.1.1 Representasi Matriks Warna Digital	229
11.1.2 Model Warna RGB	230
11.1.3 Model Warna HSV	231
11.1.4 Transformasi Ruang Warna	232
11.1.5 Implementasi Python Untuk Pengolahan Warna	232
11.2 Tujuan Praktikum	233
11.2.1 Alat Dan Bahan	233
11.2.2 Langkah Praktikum	233

11.3 Tugas Praktikum.....	242
11.4 Rangkuman.....	243
Daftar Pustaka.....	243

Bab 12 Steganografi Dan Watermarking.....	244
A. Deskripsi Pembelajaran	244
B. Kompetensi Pembelajaran	244
12.1 Konsep Dasar Steganografi.....	245
12.1.1 Metode Least Significant Bit (LSB)	246
12.1.2 Konsep Dasar Watermarking	247
12.1.3 Domain Spasial Dan Domain Transformasi.....	248
12.1.4 Evaluasi Kualitas Citra	248
12.1.5 Implementasi Python Untuk Steganografi	248
12.2 Tujuan Praktikum	249
12.2.1 Alat Dan Bahan	249
12.2.2 Langkah Praktikum	249
12.3 Tugas Praktikum.....	258
12.4 Rangkuman.....	259
Daftar Pustaka.....	260

Bab 13 Pemampatan Citra Fraktal.....	261
A. Deskripsi Pembelajaran	261
B. Kompetensi Pembelajaran	261
13.1 Konsep Dasar Fraktal Dalam Citra Digital.....	262
13.1.1 Prinsip Pemampatan Citra Fraktal	263
13.1.2 Domain Block Dan Range Block	264
13.1.3 Proses Encoding Citra Fraktal.....	264
13.1.4 Proses Decoding Citra Fraktal.....	264
13.1.5 Evaluasi Kualitas Pemampatan	265
13.2 Tujuan Praktikum	265
13.2.1 Alat Dan Bahan	265
13.2.2 Langkah Praktikum	265
13.3 Tugas Praktikum.....	277
13.4 Rangkuman.....	278
Daftar Pustaka.....	278

Bab 14 Pengenalan Pola.....	279
A. Deskripsi Pembelajaran	279

B. Kompetensi Pembelajaran	279
14.1 Konsep Dasar Pengenalan Pola.....	280
14.1.1 Representasi Fitur Dalam Citra Digital	281
14.1.2 Ekstraksi Fitur	281
14.1.3 Klasifikasi Pola.....	282
14.1.4 Evaluasi Sistem Pengenalan Pola.....	283
14.2 Tujuan Praktikum	283
13.2.1 Alat Dan Bahan	283
13.2.2 Langkah Praktikum	283
13.3 Tugas Praktikum.....	293
13.4 Rangkuman.....	294
Daftar Pustaka.....	294
Profile Penulis.....	295

DAFTAR GAMBAR

Gambar 2.1	Model Ruang Warna Pengolahan Citra.....	26
Gambar 4.1	Visualisasi Histogram Sebelum Dan Sesudah Perataan	74
Gambar 5.1	Ilustrasi Efek Filter Penghalusan Pada Citra Berderau	97
Gambar 5.2	Perbandingan Filtering Domain Spasial Dan Domain Frekuensi	100
Gambar 6.1	Diagram Alir Proses Deteksi Tepi Canny	124
Gambar 7.1	Proses Segmentasi Watershed Pada Citra Koin	148
Gambar 8.1	Kontur Objek Sebagai Batas Region Dalam Citra Digital	171
Gambar 8.2	Representasi Skeleton Objek	173
Gambar 9.1	Proses Pemampatan Citra Digital Menggunakan Encoder Dan Decoder	191
Gambar 9.2	Proses Pemampatan JPEG Menggunakan Transformasi DCT	194
Gambar 10.1	Representasi Citra Biner Dari Citra Grayscale...	210
Gambar 10.2	Operasi Morfologi Pada Citra Biner	212
Gambar 11.1	Representasi Warna Menggunakan Model RGB.	229
Gambar 11.2	Model Warna RGB Dan Kombinasi Warna Aditif.	230
Gambar 11.3	Model Warna HSV	231
Gambar 12.1	Proses Penyisipan Dan Ekstraksi Pesan Dalam Steganografi Citra Digital	246
Gambar 12.2	Proses Watermarking Dalam Citra Digital.....	247
Gambar 13.1	Struktur Fraktal Dan Self-Similarity Dalam Citra Digital.....	263
Gambar 14.1	Tahapan Sistem Pengenalan Pola Dalam Pengolahan Citra Digital	280
Gambar 14.2	Ekstraksi Fitur Pada Citra Digital	282

DAFTAR TABEL

Tabel	Perbandingan Karakteristik Metode Perbaikan	
4.1	Citra Berbasis Histogram.....	71
Tabel	Perbandingan Karakteristik Filter Linear Dan Non-	
5.1	Linear	95
Tabel	Operator Gradien Deteksi Tepi	121
6.1		
Tabel	Perbandingan Kinerja Deteksi Tepi Pada Citra	
6.2	Medis	125
Tabel	Perbandingan Karakteristik Metode Thresholding	
7.1	145
Tabel	Kinerja Metode Segmentasi Pada Citra Medis	
7.2	150
Tabel	Representasi Arah Chain Code	172
8.1		
Tabel	Representasi Huffman Coding	192
9.1		
Tabel	Representasi Matriks Citra Biner	211
10.1		
Tabel	Representasi Matriks Citra Warna RGB	229
11.1		
Tabel	Penyisipan Pesan Menggunakan Metode LSB	247
12.1		
Tabel	Representasi Domain Block Dan Range Block	264
13.1		
Tabel	Contoh Representasi Vektor Fitur	281
14.1		

BAB 1 PENGELOLAAN CITRA DIGITAL

A. Deskripsi Pembelajaran

Mata kuliah ini memberikan fondasi komprehensif tentang konsep, teori, dan praktik pengolahan citra digital. Pembelajaran dimulai dengan pengenalan elemen dasar citra, representasi matriks, serta teknik peningkatan dan pemulihan citra. Mahasiswa akan mengeksplorasi berbagai pendekatan pemrosesan, termasuk metode spasial dan transformasi frekuensi. Fokus utama diarahkan pada aplikasi nyata, khususnya peran krusial pengolahan citra dalam sistem canggih seperti Pengenalan Ucapan Audio-Visual (AVSR). Melalui serangkaian praktikum terstruktur, mahasiswa akan mendapatkan pengalaman langsung dalam memanipulasi citra, menganalisis histogram, melakukan sampling dan kuantisasi, hingga mengimplementasikan deteksi wajah sederhana sebagai langkah awal dalam sistem AVSR. Kuliah ini dirancang untuk membekali mahasiswa dengan keterampilan teoritis dan praktis yang relevan dengan tantangan teknologi terkini.

B. Kompetensi Pembelajaran

Setelah menyelesaikan mata kuliah ini, mahasiswa diharapkan mampu:

- 1) Memahami konsep fundamental pengolahan citra digital, termasuk definisi citra, piksel, ruang warna, serta kaitannya dengan sistem pengenalan ucapan audio-visual.
- 2) Mengaplikasikan teknik dasar pengolahan citra seperti pembacaan, penulisan, manipulasi nilai piksel (misalnya, negatif citra), serta konversi ruang warna menggunakan pustaka pemrosesan citra.
- 3) Menganalisis karakteristik citra melalui histogram dan statistik sederhana (rata-rata, standar deviasi) untuk memahami distribusi intensitas dan kualitas visual dari berbagai jenis citra (alam, wajah, teks).
- 4) Mengevaluasi dampak perubahan resolusi (sampling) dan jumlah tingkat keabuan (kuantisasi) terhadap persepsi visual dan kualitas informasi dalam citra.

- 5) Membangun program sederhana yang mengintegrasikan pengolahan citra untuk aplikasi dasar, seperti deteksi dan ekstraksi region of interest (wajah) dari video real-time, sebagai landasan untuk sistem audio-visual yang lebih kompleks.

1.1 Pengolahan Citra Digital

Pengolahan citra digital (digital image processing) adalah bidang ilmu yang berkaitan dengan manipulasi dan analisis citra menggunakan komputer digital. Citra digital didefinisikan sebagai fungsi dua dimensi, $f(x,y)$, di mana x dan y adalah koordinat spasial, dan amplitudo f pada setiap pasangan koordinat disebut intensitas atau tingkat keabuan citra pada titik tersebut. Elemen-elemen citra digital ini disebut piksel (pixel), yang merupakan unit terkecil dalam citra digital (Gonzalez & Woods, 2018).

1.1.1 DEFINISI

Pengolahan citra digital mencakup serangkaian teknik dan metode yang digunakan untuk memproses citra dengan tujuan meningkatkan kualitas visual, mengekstrak informasi, atau mempersiapkan citra untuk analisis lebih lanjut. Proses ini melibatkan operasi matematis dan algoritma yang diterapkan pada data citra untuk mencapai tujuan tertentu, seperti peningkatan kontras, penghilangan noise, atau segmentasi objek (Rafael C. Gonzalez, 2017). Pengolahan citra digital dapat dibagi menjadi beberapa kategori utama:

- a) Peningkatan Citra (Image Enhancement): Bertujuan untuk meningkatkan kualitas visual citra, seperti meningkatkan kontras atau mengurangi noise.
- b) Pemulihan Citra (Image Restoration): Memperbaiki citra yang terdegradasi, seperti menghilangkan blur atau noise.
- c) Pengolahan Citra Berwarna: Menggunakan model warna untuk memproses citra berwarna, seperti konversi warna atau ekstraksi fitur warna.
- d) Kompresi Citra: Mengurangi ukuran citra untuk penyimpanan atau transmisi tanpa kehilangan informasi penting.
- e) Segmentasi Citra: Memisahkan citra menjadi bagian-bagian atau objek-objek yang berbeda.

- f) Ekstraksi Fitur: Mengidentifikasi dan mendeskripsikan fitur-fitur penting dalam citra, seperti tepi atau tekstur.
- g) Klasifikasi Pola: Memberikan label pada objek berdasarkan fitur yang diekstraksi.

1.1.2 Ruang Lingkup

Ruang lingkup pengolahan citra digital sangat luas dan mencakup berbagai aplikasi di berbagai bidang, termasuk:

- a) **Kedokteran:** Pengolahan citra digunakan dalam diagnosis medis, seperti analisis citra MRI, CT scan, dan X-ray. Contohnya, pengolahan citra digunakan untuk mendeteksi tumor atau mengukur volume organ (Smith & Smith, 2020).
- b) **Astronomi:** Citra dari teleskop diproses untuk mengamati objek langit, seperti bintang, galaksi, dan planet. Pengolahan citra membantu dalam analisis data astronomi dan penemuan objek baru (Johnson et al., 2019).
- c) **Penginderaan Jauh:** Citra satelit diproses untuk memantau kondisi lingkungan, seperti perubahan iklim, deforestasi, dan polusi. Pengolahan citra juga digunakan dalam pemetaan dan perencanaan kota (Zhang & Li, 2021).
- d) **Industri:** Pengolahan citra digunakan dalam inspeksi otomatis dan pengawasan kualitas produk, seperti deteksi cacat pada produk manufaktur atau pengukuran dimensi objek (Wang et al., 2022).
- e) **Hukum dan Forensik:** Pengolahan citra digunakan dalam analisis sidik jari, pengenalan wajah, dan rekonstruksi kejadian kriminal (Brown & Davis, 2020).

1.1.3 Pendekatan Dalam Pengolahan Citra Digital

Pengolahan citra digital dapat dilakukan dengan berbagai pendekatan, termasuk:

1. **Pendekatan Spasial:** Operasi dilakukan langsung pada piksel citra, seperti filtering atau konvolusi.
2. **Pendekatan Frekuensi:** Citra diubah ke domain frekuensi menggunakan transformasi Fourier, dan operasi dilakukan pada koefisien frekuensi.

3. **Pendekatan Berbasis Model:** Menggunakan model matematis atau statistik untuk memproses citra, seperti model Markov Random Field (MRF) atau jaringan saraf tiruan (ANN).

1.1.4 Tantangan Dalam Pengolahan Citra Digital

Meskipun pengolahan citra digital telah mencapai kemajuan yang signifikan, masih ada beberapa tantangan yang perlu diatasi, seperti:

- **Kompleksitas Komputasi:** Pengolahan citra dengan resolusi tinggi membutuhkan daya komputasi yang besar.
- **Noise dan Artefak:** Citra sering kali mengandung noise atau artefak yang dapat mengganggu proses pengolahan.
- **Variabilitas Data:** Citra dari sumber yang berbeda mungkin memiliki karakteristik yang berbeda, seperti resolusi, pencahayaan, atau sudut pandang.

1.2 Pengolahan Citra Dalam Pengenalan Ucapan.

Pengolahan citra telah menjadi fondasi bagi banyak teknologi modern, termasuk pengenalan wajah, augmented reality, robotika, dan tentu saja, sistem pengenalan ucapan audio-visual. Dalam sistem pengenalan ucapan, pengolahan citra digunakan untuk menganalisis gerakan bibir dan ekspresi wajah, yang kemudian diintegrasikan dengan data audio untuk meningkatkan akurasi pengenalan (Sonka et al., 2021). Contoh aplikasi nyata dari teknologi ini dapat ditemukan dalam sistem asistensi virtual seperti Siri, Google Assistant, atau Alexa, di mana pengenalan ucapan tidak hanya mengandalkan suara, tetapi juga visual gerakan bibir pengguna. Selain itu, teknologi ini juga digunakan dalam bidang medis, seperti membantu pasien dengan gangguan pendengaran melalui visualisasi gerakan bibir (Wang et al., 2019).

Pengolahan citra digital telah menjadi tulang punggung dalam berbagai teknologi modern, termasuk sistem pengenalan ucapan (speech recognition). Meskipun pengenalan ucapan secara tradisional lebih terkait dengan pengolahan sinyal audio, pengolahan citra memainkan peran penting dalam meningkatkan

BAB 2 DASAR PEMROGRAMAN PYTHON UNTUK PENGOLAHAN CITRA

A. Deskripsi Mata Kuliah

Mata Kuliah Pengelolaan Citra Digital Dirancang Untuk Memberikan Pemahaman Fundamental Dan Keterampilan Praktis Kepada Mahasiswa Dalam Mengolah Dan Menganalisis Citra Digital Menggunakan Bahasa Pemrograman Python. Citra Digital Telah Menjadi Bagian Integral Dari Berbagai Aspek Kehidupan Modern, Mulai Dari Dokumentasi Medis, Sistem Keamanan, Hiburan, Hingga Riset Ilmiah. Kemampuan Untuk Memanipulasi Dan Mengekstrak Informasi Dari Citra Menjadi Kompetensi Yang Sangat Berharga Di Era Digital Ini. Perkembangan Pesat Dalam Bidang Kecerdasan Buatan Dan Visi Komputer Semakin Mengukuhkan Pentingnya Penguasaan Teknik Pengolahan Citra Sebagai Fondasi Utama. Oleh Karena Itu, Mata Kuliah Ini Tidak Hanya Membahas Teori Dasar Tentang Representasi Dan Sifat-Sifat Citra Digital, Tetapi Juga Secara Intensif Mempraktikkan Implementasi Algoritma Menggunakan Pustaka-Pustaka Canggih Seperti Opencv, Numpy, Dan Matplotlib. Mahasiswa Akan Diajak Untuk Memahami Bagaimana Citra Direpresentasikan Dalam Bentuk Matriks Numerik, Bagaimana Melakukan Operasi Dasar Seperti Peningkatan Kualitas Citra, Restorasi, Segmentasi, Hingga Transformasi Ruang Warna. Dengan Pendekatan Pembelajaran Berbasis Proyek Dan Studi Kasus, Mata Kuliah Ini Bertujuan Membekali Mahasiswa Dengan Kemampuan Analitis Dan Teknis Untuk Mengembangkan Solusi Inovatif Dalam Berbagai Aplikasi Dunia Nyata Yang Melibatkan Data Visual.

B. Kompetensi Pembelajaran

Setelah Menyelesaikan Mata Kuliah Ini, Mahasiswa Diharapkan Mampu Menguasai Dan Mengaplikasikan Konsep Serta Teknik Pengelolaan Citra Digital Secara Komprehensif. Kompetensi Tersebut Dirinci Sebagai Berikut.

1. Mahasiswa Mampu Menjelaskan Konsep Fundamental Citra Digital, Termasuk Proses Pembentukan Citra, Representasi Warna, Dan Karakteristik Ruang Warna Seperti RGB, Grayscale, HSV, Serta CMYK.

2. Mahasiswa Terampil Menggunakan Bahasa Pemrograman Python Dan Pustaka Pendukungnya, Terutama Numpy Untuk Manipulasi Array Multidimensi, Matplotlib Untuk Visualisasi, Serta Opencv Dan PIL Untuk Operasi Input/Output Dan Pemrosesan Citra.
3. Mahasiswa Mampu Mengimplementasikan Berbagai Teknik Pemrosesan Citra, Dimulai Dari Operasi Dasar Titik Dan Ketetangaan, Peningkatan Kualitas Citra Melalui Histogram Equalization Dan Filtering Spasial, Hingga Teknik Restorasi Citra Untuk Mengatasi Derau.
4. Mahasiswa Dapat Melakukan Analisis Citra Lebih Lanjut Seperti Segmentasi Untuk Memisahkan Objek Dari Latar Belakang, Ekstraksi Fitur Untuk Mendeskripsikan Bentuk Dan Tekstur Objek, Serta Memahami Dasar-Dasar Kompresi Citra.
5. Mahasiswa Mampu Mengintegrasikan Pengetahuan Dan Keterampilan Yang Diperoleh Untuk Mengembangkan Sebuah Proyek Sederhana Namun Fungsional, Seperti Sistem Deteksi Objek Dasar Atau Aplikasi Perbaikan Kualitas Citra Otomatis, Yang Mencerminkan Kemampuan Pemecahan Masalah Dan Inovasi Dalam Konteks Pengolahan Data Visual

2.1 Pengantar Dan Lingkungan Pemrograman Python

Python Telah Menjadi Salah Satu Bahasa Pemrograman Paling Populer Untuk Pengolahan Citra Digital Dan Visi Komputer Karena Sintaksisnya Yang Sederhana, Mudah Dipelajari, Serta Didukung Oleh Ekosistem Pustaka Ilmiah Yang Sangat Kaya Dan Kuat. Dalam Konteks Pengolahan Citra, Python Menyediakan Berbagai Pustaka Yang Dirancang Khusus Untuk Menangani Data Numerik Multidimensi, Melakukan Operasi Aljabar Linear, Dan Memvisualisasikan Hasil Pemrosesan Secara Interaktif. Seorang Praktisi Pengolahan Citra Perlu Memahami Bahwa Citra Digital Pada Hakikatnya Adalah Kumpulan Angka Yang Merepresentasikan Intensitas Cahaya Pada Titik-Titik Diskrit Yang Disebut Piksel. Kemampuan Untuk Memanipulasi Angka-Angka Ini Secara Efisien Adalah Inti Dari Pemrosesan Citra, Dan Python Dengan Pustaka Numpy-Nya Menawarkan Performa Yang Mendekati Bahasa Tingkat Rendah Seperti C Atau Fortran Untuk Operasi Matriks Skala Besar (Van Der Walt, Colbert, & Varoquaux, 2011). Untuk Memulai, Seorang Mahasiswa Perlu Menginstal

Python Beserta Pustaka-Pustaka Utama Yang Akan Digunakan Secara Konsisten Sepanjang Perkuliahan. Instalasi Dapat Dilakukan Secara Individual Menggunakan Manajer Paket Pip, Atau Yang Lebih Direkomendasikan, Menggunakan Distribusi Terintegrasi Seperti Anaconda Yang Secara Otomatis Menyertakan Python, Numpy, Matplotlib, Dan Banyak Pustaka Ilmiah Lainnya Dalam Satu Paket Yang Terkelola Dengan Baik (Bradski & Kaehler, 2008). Lingkungan Pengembangan Terintegrasi Seperti Jupyter Notebook Atau Google Colab Juga Sangat Dianjurkan Karena Memungkinkan Penulisan Kode Secara Interaktif Per Sel, Memudahkan Eksperimen, Visualisasi Langsung, Serta Dokumentasi Yang Terselip Di Antara Kode Sehingga Proses Belajar Menjadi Lebih Intuitif Dan Efisien.

2.1.1 Pustaka Fundamental Untuk Pengolahan Citra

Pemrograman Pengolahan Citra Di Python Bertumpu Pada Beberapa Pustaka Fundamental Yang Masing-Masing Memiliki Peran Spesifik Dan Saling Melengkapi. Pustaka Yang Paling Utama Adalah Numpy, Yang Menyediakan Struktur Data Dasar Berupa Array Multidimensi Yang Efisien Dan Kaya Akan Fungsi-Fungsi Untuk Manipulasi Numerik. Setiap Citra Digital Yang Dibaca Akan Direpresentasikan Sebagai Array Numpy, Dengan Dimensi Yang Sesuai: Dua Dimensi Untuk Citra Grayscale (Baris Dan Kolom), Dan Tiga Dimensi Untuk Citra Berwarna (Baris, Kolom, Dan Kanal Warna). Operasi Seperti Memotong (Cropping) Citra, Mengakses Nilai Piksel Tertentu, Atau Melakukan Operasi Aritmetika Pada Seluruh Citra Menjadi Sangat Sederhana Berkat Kemampuan Broadcasting Dan Vektorisasi Numpy (Bradski & Kaehler, 2008). Selanjutnya, Pustaka Opencv (Open Source Computer Vision Library) Merupakan Pustaka Yang Sangat Komprehensif, Menyediakan Ratusan Algoritma Di Bidang Visi Komputer Dan Pengolahan Citra, Mulai Dari Operasi Dasar Seperti Membaca Dan Menulis Citra, Mengubah Ukuran, Memutar, Hingga Algoritma Yang Lebih Kompleks Seperti Deteksi Tepi Canny, Segmentasi, Dan Pengenalan Objek. Opencv Dirancang Untuk Efisiensi Komputasi Dan Memiliki Antarmuka Yang Mudah Digunakan Di Python (Pulli, Baksheev, Korniyakov, & Eruhimov, 2012). Selain Opencv, Pustaka PIL (Python Imaging Library) Atau Penerusnya, Pillow, Juga Sering Digunakan Untuk Operasi Dasar Pembukaan, Manipulasi, Dan

Penyimpanan Berbagai Format File Citra Dengan Cara Yang Lebih Ringan. Terakhir, Matplotlib, Khususnya Modul Pyplot, Menjadi Alat Utama Untuk Visualisasi. Kemampuan Untuk Menampilkan Citra Di Dalam Notebook, Membuat Plot Histogram, Atau Memvisualisasikan Hasil Transformasi Secara Berdampingan Sangat Membantu Dalam Memahami Efek Dari Setiap Operasi Pemrosesan Yang Dilakukan (Hunter, 2007).

2.1.2 Operasi Input/Output Dan Representasi Citra

Langkah Pertama Dalam Setiap Proyek Pengolahan Citra Adalah Membaca Citra Dari Media Penyimpanan Ke Dalam Memori Sebagai Struktur Data Yang Dapat Diproses. Dengan Opencv, Fungsi `Cv2.imread()` Digunakan Untuk Tujuan Ini. Fungsi Ini Akan Mengembalikan Citra Sebagai Array Numpy, Dengan Nilai Piksel Berkisar Antara 0 Hingga 255 Untuk Citra 8-Bit. Penting Untuk Dicatat Bahwa Opencv Membaca Citra Berwarna Dalam Urutan Kanal BGR (Blue, Green, Red) Secara Default, Berbeda Dengan Urutan RGB Yang Umum Digunakan Oleh Pustaka Lain Seperti Matplotlib Untuk Ditampilkan. Oleh Karena Itu, Setelah Membaca Citra Dengan Opencv, Seringkali Diperlukan Konversi Urutan Kanal Menggunakan `Cv2.cvtColor()` Sebelum Menampilkannya Dengan Matplotlib Agar Warna Yang Ditampilkan Sesuai Dengan Aslinya (Szeliski, 2022). Sementara Itu, Pustaka Pillow Dengan Fungsi `Image.open()` Membaca Citra Dalam Format RGB Dan Mengembalikan Objek Image, Yang Kemudian Dapat Dikonversi Menjadi Array Numpy Untuk Pemrosesan Lebih Lanjut. Untuk Menyimpan Citra Hasil Pemrosesan, Opencv Menyediakan Fungsi `Cv2.imwrite()` Yang Secara Otomatis Menentukan Format File Berdasarkan Ekstensi Yang Diberikan. Setelah Citra Dibaca Dan Direpresentasikan Sebagai Array Numpy, Mahasiswa Dapat Mulai Mengeksplorasi Atribut-Atribut Dasarnya. Atribut Shape Dari Array Akan Mengembalikan Tupel Yang Berisi Tinggi (Jumlah Baris), Lebar (Jumlah Kolom), Dan Jumlah Kanal (Jika Ada). Atribut Dtype Akan Menunjukkan Tipe Data Piksel, Misalnya `uint8` Untuk Citra 8-Bit Tanpa Tanda. Pemahaman Tentang Representasi Ini Sangat Fundamental, Karena Setiap Operasi Pemrosesan Pada Dasarnya Adalah Manipulasi Matematis Pada Elemen-Elemen Array Ini (Burger & Burge, 2022).

BAB 3 OPERASI DASAR DAN MANIPULASI CITRA

A. Deskripsi Mata Kuliah

Bab Ini Membekali Mahasiswa Dengan Pemahaman Teoretis Dan Keterampilan Praktis Dalam Mengolah Serta Menganalisis Citra Digital Menggunakan Bahasa Pemrograman Python. Citra Digital Telah Menjadi Komponen Esensial Dalam Berbagai Bidang Kehidupan Modern, Mulai Dari Diagnostik Medis, Sistem Keamanan, Dokumentasi Ilmiah, Hingga Aplikasi Industri Dan Hiburan. Kemampuan Memanipulasi Dan Mengekstrak Informasi Dari Data Visual Menjadi Kompetensi Yang Sangat Diperlukan Di Era Digital Ini. Perkembangan Pesat Teknologi Kecerdasan Buatan Dan Visi Komputer Semakin Mengukuhkan Pentingnya Penguasaan Teknik Pengolahan Citra Sebagai Fondasi Utama Pengembangan Sistem Cerdas. Oleh Karena Itu, Mata Kuliah Ini Tidak Hanya Membahas Konsep Fundamental Tentang Representasi Dan Karakteristik Citra Digital, Tetapi Juga Secara Intensif Mempraktikkan Implementasi Berbagai Algoritma Menggunakan Pustaka-Pustaka Mutakhir Seperti Opencv, Numpy, Dan Matplotlib. Mahasiswa Akan Mempelajari Bagaimana Citra Direpresentasikan Sebagai Matriks Numerik, Melakukan Operasi Dasar Peningkatan Kualitas Citra, Restorasi, Segmentasi, Transformasi Ruang Warna, Hingga Ekstraksi Fitur Untuk Analisis Lebih Lanjut. Dengan Pendekatan Pembelajaran Berbasis Proyek Dan Studi Kasus Kontekstual, Mata Kuliah Ini Bertujuan Menghasilkan Lulusan Yang Mampu Mengembangkan Solusi Inovatif Untuk Berbagai Permasalahan Nyata Yang Melibatkan Data Visual (Gonzalez & Woods, 2018; Szeliski, 2022).

B. Kompetensi Pembelajaran

Setelah Menyelesaikan Bab Ini, Mahasiswa Mampu Menguasai Dan Mengaplikasikan Konsep Serta Teknik Pengelolaan Citra Digital Secara Komprehensif.

1. Mahasiswa Mampu Menjelaskan Konsep Fundamental Citra Digital, Meliputi Proses Pembentukan Citra Melalui Sampling Dan Kuantisasi, Representasi Warna Dalam Berbagai Ruang Warna Seperti RGB, Grayscale, HSV, Dan CMYK, Serta Karakteristik Berbagai Format File Citra (Hidayatullah, 2017).

2. Mahasiswa Terampil Menggunakan Bahasa Pemrograman Python Dan Pustaka Pendukungnya, Terutama Numpy Untuk Manipulasi Array Multidimensi, Matplotlib Untuk Visualisasi Data, Serta Opencv Dan Pillow Untuk Operasi Input-Output Dan Pemrosesan Citra.
3. Mahasiswa Mampu Mengimplementasikan Berbagai Teknik Pemrosesan Citra, Dimulai Dari Operasi Dasar Aras Titik Dan Aras Ketetangaan, Peningkatan Kualitas Citra Melalui Peregangan Kontras, Perataan Histogram, Dan Filtering Spasial, Hingga Teknik Restorasi Citra Untuk Mengatasi Berbagai Jenis Derau.
4. Mahasiswa Dapat Melakukan Analisis Citra Lebih Lanjut Seperti Segmentasi Untuk Memisahkan Objek Dari Latar Belakang Menggunakan Metode Thresholding, Ekstraksi Fitur Morfologi Untuk Mendeskripsikan Bentuk Dan Tekstur Objek, Serta Memahami Prinsip Dasar Kompresi Citra. Kelima, Mahasiswa Mampu Mengintegrasikan Seluruh Pengetahuan Dan Keterampilan Yang Diperoleh Untuk Mengembangkan Sebuah Proyek Aplikasi Sederhana Namun Fungsional, Seperti Sistem Deteksi Objek Dasar Atau Aplikasi Perbaikan Kualitas Citra Otomatis, Yang Mencerminkan Kemampuan Pemecahan Masalah Dan Inovasi Dalam Konteks Pengolahan Data Visual (Burger & Burge, 2022; Nasuha Dkk., 2023).

3.1 Konsep Dasar Operasi Citra

Operasi Pengolahan Citra Digital Pada Dasarnya Merupakan Serangkaian Manipulasi Matematis Yang Diterapkan Pada Representasi Numerik Citra. Setiap Operasi Bertujuan Mengubah Atau Menganalisis Informasi Yang Terkandung Dalam Citra Untuk Memenuhi Kebutuhan Spesifik, Baik Untuk Perbaikan Kualitas Visual, Ekstraksi Informasi, Maupun Persiapan Data Untuk Analisis Lebih Lanjut. Para Praktisi Pengolahan Citra Mengklasifikasikan Operasi Ini Berdasarkan Cakupan Area Pemrosesan, Yaitu Operasi Aras Titik Yang Hanya Melibatkan Satu Piksel, Operasi Aras Lokal Yang Melibatkan Sekumpulan Piksel Tetangga, Dan Operasi Aras Global Yang Melibatkan Seluruh Piksel Dalam Citra. Pemahaman Tentang Klasifikasi Ini Membantu Mahasiswa Memilih Pendekatan Yang Tepat Untuk Menyelesaikan Permasalahan Tertentu. Sebagai Contoh, Koreksi Kecerahan Cukup Dilakukan Dengan Operasi Aras

Titik, Sementara Penghalusan Derau Memerlukan Operasi Aras Lokal Yang Mempertimbangkan Piksel-Piksel Di Sekitarnya (Hidayatullah, 2017). Dalam Implementasinya Menggunakan Python, Seluruh Operasi Tersebut Memanfaatkan Representasi Citra Sebagai Array Numpy, Sehingga Mahasiswa Perlu Memahami Dengan Baik Cara Mengakses, Memodifikasi, Dan Memanipulasi Struktur Data Array Multidimensi Yang Menjadi Fondasi Semua Pemrosesan.

3.1.1 Operasi Berbasis Titik (Point Operations)

Operasi Berbasis Titik Merupakan Bentuk Paling Sederhana Dalam Pengolahan Citra Digital. Pada Operasi Ini, Nilai Intensitas Piksel Keluaran Hanya Ditentukan Oleh Nilai Intensitas Piksel Masukan Pada Koordinat Yang Sama, Tanpa Melibatkan Piksel Tetangganya. Para Ahli Menyebut Operasi Ini Sebagai Transformasi Intensitas Karena Fungsinya Memetakan Rentang Intensitas Masukan Ke Rentang Intensitas Keluaran Yang Baru (Gonzalez & Woods, 2018). Dalam Pemrograman Python Menggunakan Pustaka Pillow Atau Opencv, Mahasiswa Dapat Mengimplementasikan Operasi Ini Dengan Mengakses Setiap Piksel Menggunakan Perulangan Bersarang Atau Memanfaatkan Kemampuan Vektorisasi Numpy Untuk Efisiensi Komputasi Yang Lebih Tinggi.

Salah Satu Operasi Titik Yang Paling Fundamental Adalah Operasi Negatif Atau Inversi. Operasi Ini Menghasilkan Citra Dengan Hubungan Intensitas Terbalik Dari Citra Asli, Di Mana Area Terang Menjadi Gelap Dan Sebaliknya. Rumus Yang Digunakan Sangat Sederhana, Yaitu Nilai Keluaran Sama Dengan 255 Dikurangi Nilai Masukan Untuk Citra 8-Bit. Implementasinya Dalam Python Dapat Dilakukan Dengan Mengurangkan Setiap Elemen Array Dari 255, Atau Menggunakan Fungsi Bawaan Dari Pustaka Pengolahan Citra. Operasi Negatif Bermanfaat Untuk Analisis Area Tertentu Dalam Citra Medis Atau Untuk Efek Artistik Dalam Fotografi Digital (Burger & Burge, 2022).

Operasi Pencerahan Dan Penggelapan Citra Juga Termasuk Dalam Kategori Operasi Titik. Mahasiswa Dapat Mencerahkan Citra Dengan Menambahkan Konstanta Positif Ke Setiap Nilai Piksel, Atau Menggelapkannya Dengan Mengurangi Konstanta. Namun Demikian, Penambahan Atau Pengurangan Ini Berisiko

Menghasilkan Nilai Piksel Di Luar Rentang Valid 0 Hingga 255. Oleh Karena Itu, Diperlukan Mekanisme Kliping Yang Memastikan Nilai Hasil Operasi Tetap Berada Dalam Rentang Yang Diizinkan. Dalam Praktik Pemrograman, Mahasiswa Perlu Membuat Fungsi Kliping Sendiri Atau Menggunakan Fungsi Bawaan Opencv Seperti `Cv2.Add()` Dan `Cv2.Subtract()` Yang Secara Otomatis Melakukan Penjenuhan Nilai (Nasuha Dkk., 2023). Implementasi Kliping Manual Dapat Dilakukan Dengan Memeriksa Setiap Nilai Hasil Operasi, Mengatur Nilai Di Bawah 0 Menjadi 0 Dan Nilai Di Atas 255 Menjadi 255.

Operasi Peregangan Kontras Merupakan Pengembangan Lebih Lanjut Dari Operasi Titik. Teknik Ini Bertujuan Memperluas Rentang Intensitas Citra Sehingga Memanfaatkan Seluruh Rentang Yang Tersedia, Misalnya Dari 0 Hingga 255. Citra Dengan Kontras Rendah Umumnya Memiliki Histogram Yang Mengumpul Pada Rentang Sempit, Sehingga Detail-Detail Penting Sulit Dibedakan Oleh Mata Manusia. Peregangan Kontras Melakukan Pemetaan Linear Yang Mendistribusikan Ulang Intensitas Piksel Sehingga Nilai Minimum Petanya Menjadi 0 Dan Nilai Maksimum Menjadi 255. Implementasi Teknik Ini Memerlukan Identifikasi Nilai Intensitas Minimum Dan Maksimum Dari Citra Asli, Kemudian Menerapkan Transformasi Linear Yang Sesuai (Hidayatullah, 2017).

3.1.2 Operasi Aritmetika Dan Logika

Operasi Aritmetika Dan Logika Melibatkan Dua Atau Lebih Citra Sebagai Masukan, Menghasilkan Citra Baru Berdasarkan Perhitungan Matematis Antar Piksel Pada Posisi Yang Bersesuaian. Operasi Aritmetika Meliputi Penjumlahan, Pengurangan, Perkalian, Dan Pembagian, Sementara Operasi Logika Mencakup AND, OR, XOR, Dan NOT. Kedua Jenis Operasi Ini Memiliki Aplikasi Luas Dalam Berbagai Skenario Pengolahan Citra (Szeliski, 2022). Operasi Penjumlahan Dua Citra Sering Digunakan Untuk Menciptakan Efek Superimposisi Atau Menggabungkan Informasi Dari Dua Sumber Berbeda. Dalam Python, Mahasiswa Dapat Menjumlahkan Dua Array Numpy Secara Langsung Menggunakan Operator `+`, Namun Perlu Memperhatikan Potensi Overflow. Pendekatan Yang Lebih Aman Menggunakan Fungsi `Cv2.Add()` Yang Melakukan Penjenuhan Nilai Secara

Otomatis. Operasi Pengurangan Sangat Berguna Untuk Mendeteksi Perubahan Atau Pergerakan Antar Dua Citra Yang Diambil Pada Waktu Berbeda. Teknik Ini Menjadi Dasar Dalam Sistem Pengawasan Video Untuk Mendeteksi Objek Bergerak. Ketika Mengurangkan Citra Latar Dari Citra Terkini, Area Yang Tidak Berubah Akan Mendekati Nol, Sementara Area Dengan Perubahan Akan Memiliki Nilai Positif Atau Negatif Yang Signifikan (Ramdani, 2006).

Operasi Perkalian Citra Dengan Skalar Atau Dengan Citra Lain Dapat Digunakan Untuk Mengatur Kontras Atau Menerapkan Efek Tertentu. Sementara Itu, Operasi Pembagian Bermanfaat Dalam Normalisasi Dan Koreksi Iluminasi. Untuk Operasi Logika, Penerapannya Umumnya Pada Citra Biner Hasil Segmentasi. Operasi AND Digunakan Untuk Menampilkan Irisan Dua Area, OR Untuk Menggabungkan Area, XOR Untuk Mendeteksi Perbedaan, Dan NOT Untuk Membalikkan Area Objek Menjadi Latar Atau Sebaliknya. Dalam Implementasinya, Opencv Menyediakan Fungsi-Fungsi Seperti `Cv2.Bitwise_And()`, `Cv2.Bitwise_Or()`, `Cv2.Bitwise_Xor()`, Dan `Cv2.Bitwise_Not()` Yang Bekerja Secara Efisien Pada Array Multidimensi (Bradski & Kaehler, 2008).

3.1.3 Transformasi Geometri

Transformasi Geometri Mengubah Tata Letak Spasial Pixel Dalam Citra, Berbeda Dengan Operasi Aras Titik Yang Hanya Mengubah Nilai Intensitas. Operasi Ini Meliputi Translasi, Rotasi, Penskalaan, Flipping, Dan Transformasi Affine Atau Perspektif. Pemahaman Tentang Transformasi Geometri Sangat Penting Dalam Berbagai Aplikasi Seperti Registrasi Citra, Koreksi Distorsi Kamera, Dan Augmentasi Data Untuk Pembelajaran Mesin (Burger & Burge, 2022). Translasi Merupakan Operasi Pergeseran Citra Sejauh Tertentu Pada Sumbu Horizontal Dan Vertikal. Implementasi Translasi Memerlukan Pembuatan Matriks Transformasi 2×3 Yang Mendefinisikan Besaran Pergeseran, Kemudian Menerapkannya Menggunakan Fungsi `Cv2.WarpAffine()`. Mahasiswa Perlu Memahami Bahwa Area Citra Yang Bergeser Keluar Dari Bingkai Akan Hilang, Sementara Area Kosong Yang Ditinggalkan Perlu Diisi Dengan Nilai Tertentu, Biasanya Hitam Atau Dengan Teknik Padding Lainnya. Untuk Translasi Yang Tidak Mengubah Ukuran Kanvas, Diperlukan

Alokasi Memori Yang Cukup Dan Penanganan Batas Yang Cermat (Nasuha Dkk., 2023).

Rotasi Citra Dapat Dilakukan Dengan Menentukan Pusat Rotasi, Sudut Putaran, Dan Faktor Skala. Opencv Menyediakan Fungsi `Cv2.Getrotationmatrix2d()` Untuk Menghasilkan Matriks Rotasi, Yang Kemudian Diterapkan Dengan `Cv2.Warpaffine()`. Rotasi Dengan Sudut Yang Bukan Kelipatan 90 Derajat Memerlukan Interpolasi Untuk Menghitung Nilai Piksel Pada Posisi Baru Yang Tidak Tepat Berada Pada Grid Asli. Metode Interpolasi Yang Tersedia Antara Lain `INTER_NEAREST` Untuk Kecepatan, `INTER_LINEAR` Untuk Keseimbangan Kecepatan Dan Kualitas, Serta `INTER_CUBIC` Untuk Kualitas Tertinggi. Pemilihan Metode Interpolasi Mempengaruhi Ketajaman Dan Kehalusan Citra Hasil Rotasi (Gonzalez & Woods, 2018).

Penskalaan Atau Pengubahan Ukuran Citra Dilakukan Dengan Fungsi `Cv2.Resize()`. Operasi Ini Dapat Memperbesar Atau Memperkecil Citra, Dengan Konsekuensi Yang Berbeda Pada Kualitas Visual. Saat Memperkecil, Terjadi Reduksi Informasi Yang Dapat Menyebabkan Hilangnya Detail. Saat Memperbesar, Diperlukan Interpolasi Untuk Mengisi Piksel-Piksel Baru. Mahasiswa Perlu Memahami Trade-Off Antara Berbagai Metode Interpolasi Dan Memilih Yang Sesuai Dengan Kebutuhan Aplikasi. Selain Itu, Penting Juga Memahami Konsep Aspek Rasio Dan Bagaimana Mempertahankannya Saat Mengubah Ukuran Agar Citra Tidak Terdistorsi (Szeliski, 2022).

3.1.4 Analisis Histogram Dan Peningkatan Kontras

Histogram Citra Merupakan Representasi Grafis Yang Menunjukkan Distribusi Frekuensi Intensitas Piksel Dalam Suatu Citra. Pada Sumbu Horizontal Tercantum Nilai Intensitas Dari 0 Hingga 255 Untuk Citra 8-Bit, Sementara Sumbu Vertikal Menunjukkan Jumlah Piksel Yang Memiliki Intensitas Tersebut. Histogram Memberikan Informasi Penting Tentang Karakteristik Citra, Seperti Kecerahan Rata-Rata, Kontras, Dan Kecenderungan Distribusi Intensitas. Mahasiswa Dapat Menghasilkan Histogram Menggunakan `Matplotlib` Dengan Fungsi `plt.hist()` Setelah Meratakan Array Citra Menjadi Satu Dimensi (Hunter, 2007).

Citra Dengan Kontras Baik Umumnya Memiliki Histogram Yang Merata Di Seluruh Rentang Intensitas, Sementara Citra Dengan

BAB 4 HISTOGRAM DAN PERBAIKAN KUALITAS CITRA

A. Deskripsi Mata Kuliah

Bab Ini Dirancang Untuk Membekali Mahasiswa Dengan Pemahaman Teoretis Dan Keterampilan Praktis Dalam Mengolah Serta Menganalisis Citra Digital Menggunakan Bahasa Pemrograman Python. Citra Digital Telah Menjadi Komponen Esensial Dalam Berbagai Bidang Kehidupan Modern, Mulai Dari Diagnostik Medis, Sistem Keamanan, Dokumentasi Ilmiah, Hingga Aplikasi Industri Dan Hiburan. Kemampuan Memanipulasi Dan Mengekstrak Informasi Dari Data Visual Menjadi Kompetensi Yang Sangat Diperlukan Di Era Digital Ini. Perkembangan Pesat Teknologi Kecerdasan Buatan Dan Visi Komputer Semakin Mengukuhkan Pentingnya Penguasaan Teknik Pengolahan Citra Sebagai Fondasi Utama Pengembangan Sistem Cerdas. Oleh Karena Itu, Mata Kuliah Ini Tidak Hanya Membahas Konsep Fundamental Tentang Representasi Dan Karakteristik Citra Digital, Tetapi Juga Secara Intensif mempraktikkan implementasi berbagai algoritma menggunakan pustaka-pustaka mutakhir seperti `Opencv`, `Numpy`, dan `Matplotlib`. Mahasiswa akan mempelajari bagaimana citra direpresentasikan sebagai matriks numerik, melakukan operasi dasar peningkatan kualitas citra, restorasi, segmentasi, transformasi ruang warna, hingga ekstraksi fitur untuk analisis lebih lanjut. Dengan pendekatan pembelajaran berbasis proyek dan studi kasus kontekstual, mata kuliah ini bertujuan menghasilkan lulusan yang mampu mengembangkan solusi inovatif untuk berbagai permasalahan nyata yang melibatkan data visual.

B. Kompetensi Pembelajaran

Setelah menyelesaikan mata kuliah ini, mahasiswa diharapkan mampu menguasai dan mengaplikasikan konsep serta teknik pengelolaan citra digital secara komprehensif. Pertama, mahasiswa mampu menjelaskan konsep fundamental citra digital, meliputi proses pembentukan citra melalui sampling dan kuantisasi, representasi warna dalam berbagai ruang warna seperti RGB, Grayscale, HSV, dan CMYK, serta karakteristik berbagai format file citra. Kedua, mahasiswa terampil

Menggunakan Bahasa Pemrograman Python Dan Pustaka Pendukungnya, Terutama Numpy Untuk Manipulasi Array Multidimensi, Matplotlib Untuk Visualisasi Data, Serta Opencv Dan Pillow Untuk Operasi Input-Output Dan Pemrosesan Citra. Ketiga, Mahasiswa Mampu Mengimplementasikan Berbagai Teknik Pemrosesan Citra, Dimulai Dari Operasi Dasar Aras Titik Dan Aras Ketetangaan, Peningkatan Kualitas Citra Melalui Peregangan Kontras, Perataan Histogram, Dan Filtering Spasial, Hingga Teknik Restorasi Citra Untuk Mengatasi Berbagai Jenis Derau. Keempat, Mahasiswa Dapat Melakukan Analisis Citra Lebih Lanjut Seperti Segmentasi Untuk Memisahkan Objek Dari Latar Belakang Menggunakan Metode Thresholding, Ekstraksi Fitur Morfologi Untuk Mendeskripsikan Bentuk Dan Tekstur Objek, Serta Memahami Prinsip Dasar Kompresi Citra. Kelima, Mahasiswa Mampu Mengintegrasikan Seluruh Pengetahuan Dan Keterampilan Yang Diperoleh Untuk Mengembangkan Sebuah Proyek Aplikasi Sederhana Namun Fungsional, Seperti Sistem Deteksi Objek Dasar Atau Aplikasi Perbaikan Kualitas Citra Otomatis, Yang Mencerminkan Kemampuan Pemecahan Masalah Dan Inovasi Dalam Konteks Pengolahan Data Visual.

4.1 Histogram Dan Perbaikan Kualitas Citra

4.1.1 Konsep Dasar Histogram Citra

Histogram Citra Merupakan Fondasi Penting Dalam Analisis Dan Perbaikan Kualitas Citra Digital. Para Praktisi Pengolahan Citra Mendefinisikan Histogram Sebagai Fungsi Yang Menunjukkan Distribusi Frekuensi Kemunculan^{4.1} Setiap Nilai Intensitas Piksel Dalam Suatu Citra. Pada Sumbu Horizontal Histogram Tercantum Nilai Intensitas Dari 0 Hingga 255 Untuk Citra 8-Bit, Sementara Sumbu Vertikal Menunjukkan Jumlah Piksel Yang Memiliki Intensitas Tersebut. Representasi Grafis Ini Memberikan Informasi Berharga Tentang Karakteristik Global Citra, Seperti Kecerahan Rata-Rata, Kontras, Dan Kecenderungan Distribusi Intensitas . Mahasiswa Perlu Memahami Bahwa Histogram Berfungsi Sebagai Alat Diagnostik Yang Memungkinkan Mereka Mengidentifikasi Masalah Kualitas Citra Secara Objektif Sebelum Menentukan Teknik Perbaikan Yang Tepat. Perhitungan Histogram Secara Matematis Dinyatakan Sebagai Fungsi $H(K)$ Yang Menghitung Jumlah Piksel Dengan Intensitas K . Untuk Citra Grayscale Dengan

L Tingkat Keabuan (Biasanya 256), Rumus Yang Digunakan Adalah $H(K) = \sum_i \sum_j I(F(I,J) = K)$, Di Mana $I(\cdot)$ Merupakan Fungsi Indikator Yang Bernilai 1 Jika Kondisi Terpenuhi Dan 0 Jika Tidak . Implementasi Perhitungan Histogram Dalam Python Dapat Dilakukan Secara Efisien Menggunakan Fungsi `Cv2.Calchist()` Dari Pustaka `Opencv` Atau `Np.Histogram()` Dari `Numpy`. Mahasiswa Juga Dapat Memanfaatkan `Matplotlib` Untuk Visualisasi Histogram Dengan Fungsi `Plt.Hist()` Yang Secara Otomatis Meratakan Array Citra Menjadi Satu Dimensi Dan Menampilkan Distribusi Frekuensinya. Pemahaman Tentang Representasi Matematis Ini Membantu Mahasiswa Mengapresiasi Bagaimana Histogram Mencerminkan Karakteristik Visual Citra Secara Kuantitatif.

4.1.2 Interpretasi Histogram Untuk Diagnosis Kualitas Citra

Histogram Memberikan Informasi Diagnostik Yang Sangat Berharga Tentang Kualitas Citra. Citra Dengan Kontras Baik Umumnya Memiliki Histogram Yang Terdistribusi Secara Merata Di Seluruh Rentang Intensitas 0 Hingga 255, Menunjukkan Bahwa Citra Memanfaatkan Seluruh Rentang Dinamis Yang Tersedia. Sebaliknya, Citra Dengan Kontras Rendah Memiliki Histogram Yang Mengumpul Pada Area Sempit, Sehingga Detail-Detail Penting Sulit Dibedakan Oleh Mata Manusia. Citra Yang Terlalu Gelap (Underexposed) Menunjukkan Histogram Yang Cenderung Mengelompok Di Daerah Nilai Rendah Dekat 0, Sementara Citra Yang Terlalu Terang (Overexposed) Memiliki Histogram Yang Mengelompok Di Daerah Nilai Tinggi Dekat 255. Kondisi Underexposed Dan Overexposed Ini Menyebabkan Hilangnya Detail Pada Area Bayangan Atau Area Sorot Karena Keterbatasan Rentang Dinamis .

Selain Itu, Analisis Histogram Juga Dapat Mengungkap Keberadaan Derau Dalam Citra. Derau Impulsif (Salt-And-Pepper Noise) Misalnya, Akan Tampak Sebagai Lonjakan-Lonjakan Kecil Pada Nilai Intensitas Ekstrem 0 Dan 255. Sementara Derau Gaussian Cenderung Membuat Histogram Menjadi Lebih Menyebar Dengan Variansi Yang Lebih Besar Dari Kondisi Normal. Dalam Konteks Pemampatan Citra, Analisis Histogram Membantu Peneliti Mengevaluasi Kualitas Citra Hasil Kompresi Dengan Membandingkan Bentuk Histogram Sebelum Dan Sesudah Pemampatan. Perubahan Distribusi Intensitas Akibat Kuantisasi

Yang Terlalu Kasar Akan Tampak Pada Histogram Sebagai Pita-Pita Diskontinu Atau Hilangnya Variasi Halus Antar Intensitas Yang Berdekatan . Kemampuan Membaca Dan Menginterpretasi Histogram Ini Menjadi Keterampilan Fundamental Yang Harus dikuasai Mahasiswa Sebelum Melangkah Ke Teknik Perbaikan Kualitas Citra Yang Lebih Kompleks.

4.1.3 Teknik Perbaikan Kualitas Berbasis Histogram

Perataan Histogram (Histogram Equalization) Merupakan Teknik Otomatis Yang Paling Populer Untuk Meningkatkan Kontras Citra. Teknik Ini Bekerja Dengan Mentransformasi Nilai Intensitas Sedemikian Rupa Sehingga Histogram Citra Keluaran Sedekat Mungkin Dengan Distribusi Uniform. Prosesnya Melibatkan Perhitungan Fungsi Distribusi Kumulatif Dari Histogram Asli, Kemudian Memetakan Setiap Intensitas Asli Ke Intensitas Baru Berdasarkan Fungsi Tersebut. Tujuan Utamanya Adalah Menghasilkan Citra Dengan Kontras Yang Seragam Di Seluruh Rentang Intensitas, Sehingga Detail-Detail Yang Sebelumnya Tersembunyi Di Area Gelap Atau Terang Menjadi Lebih Terlihat .

Implementasi Perataan Histogram Dalam Python Untuk Citra Grayscale Sangat Sederhana Menggunakan Opencv. Mahasiswa Cukup Memanggil Fungsi `Cv2.Equalizehist()` Dengan Parameter Citra Masukan, Dan Fungsi Tersebut Akan Mengembalikan Citra Hasil Perataan. Fungsi Ini Mengimplementasikan Algoritma Yang Telah Dioptimasi Untuk Kinerja Tinggi. Namun Demikian, Mahasiswa Perlu Memahami Proses Yang Terjadi Di Balik Fungsi Tersebut Agar Dapat Memodifikasinya Untuk Kebutuhan Khusus. Proses Tersebut Meliputi: Pertama, Menghitung Histogram Citra Asli; Kedua, Menghitung Distribusi Kumulatif Dengan Menjumlahkan Frekuensi Secara Berurutan; Ketiga, Menormalisasi Distribusi Kumulatif Ke Rentang 0-255; Dan Keempat, Memetakan Setiap Piksel Asli Ke Nilai Baru Berdasarkan Hasil Normalisasi .

Hasil Penelitian Menunjukkan Bahwa Penerapan Histogram Equalization Mampu Memperbaiki Citra Yang Diujikan Secara Signifikan. Dalam Pengujian Kuantitatif, Nilai Peak Signal-To-Noise Ratio (PSNR) Citra Hasil Perbaikan Mencapai Kisaran 25,98 Db Hingga 40,59 Db, Sementara Nilai Mean Squared Error (MSE) Berada Pada Rentang 74,63 Hingga 163,80. Penilaian Subjektif Dari Responden Menunjukkan Bahwa Citra Hasil Perbaikan Masuk

BAB 5 FILTERING DAN PENINGKATAN CITRA

A. Deskripsi

Bab Ini Dirancang Untuk Membekali Mahasiswa Dengan Pemahaman Teoretis Dan Keterampilan Praktis Dalam Mengolah Serta Menganalisis Citra Digital Menggunakan Bahasa Pemrograman Python. Citra Digital Telah Menjadi Komponen Esensial Dalam Berbagai Bidang Kehidupan Modern, Mulai Dari Diagnostik Medis, Sistem Keamanan, Dokumentasi Ilmiah, Hingga Aplikasi Industri Dan Hiburan. Kemampuan Memanipulasi Dan Mengekstrak Informasi Dari Data Visual Menjadi Kompetensi Yang Sangat Diperlukan Di Era Digital Ini. Perkembangan Pesat Teknologi Kecerdasan Buatan Dan Visi Komputer Semakin Mengukuhkan Pentingnya Penguasaan Teknik Pengolahan Citra Sebagai Fondasi Utama Pengembangan Sistem Cerdas. Oleh Karena Itu, Mata Kuliah Ini Tidak Hanya Membahas Konsep Fundamental Tentang Representasi Dan Karakteristik Citra Digital, Tetapi Juga Secara Intensif mempraktikkan implementasi berbagai algoritma menggunakan pustaka-pustaka mutakhir seperti `Opencv`, `Numpy`, dan `Matplotlib`. Mahasiswa akan mempelajari bagaimana citra direpresentasikan sebagai matriks numerik, melakukan operasi dasar peningkatan kualitas citra, restorasi, segmentasi, transformasi ruang warna, hingga ekstraksi fitur untuk analisis lebih lanjut. Dengan pendekatan pembelajaran berbasis proyek dan studi kasus kontekstual, mata kuliah ini bertujuan menghasilkan lulusan yang mampu mengembangkan solusi inovatif untuk berbagai permasalahan nyata yang melibatkan data visual.

B. Kompetensi Pembelajaran

Setelah menyelesaikan mata kuliah ini, mahasiswa diharapkan mampu menguasai dan mengaplikasikan konsep serta teknik pengelolaan citra digital secara komprehensif. Pertama, mahasiswa mampu menjelaskan konsep fundamental citra digital, meliputi proses pembentukan citra melalui sampling dan kuantisasi, representasi warna dalam berbagai ruang warna seperti RGB, Grayscale, HSV, dan CMYK, serta karakteristik berbagai format file citra. Kedua, mahasiswa terampil

Menggunakan Bahasa Pemrograman Python Dan Pustaka Pendukungnya, Terutama Numpy Untuk Manipulasi Array Multidimensi, Matplotlib Untuk Visualisasi Data, Serta Opencv Dan Pillow Untuk Operasi Input-Output Dan Pemrosesan Citra. Ketiga, Mahasiswa Mampu Mengimplementasikan Berbagai Teknik Pemrosesan Citra, Dimulai Dari Operasi Dasar Aras Titik Dan Aras Ketetangaan, Peningkatan Kualitas Citra Melalui Peregangan Kontras, Perataan Histogram, Dan Filtering Spasial, Hingga Teknik Restorasi Citra Untuk Mengatasi Berbagai Jenis Derau. Keempat, Mahasiswa Dapat Melakukan Analisis Citra Lebih Lanjut Seperti Segmentasi Untuk Memisahkan Objek Dari Latar Belakang Menggunakan Metode Thresholding, Ekstraksi Fitur Morfologi Untuk Mendeskripsikan Bentuk Dan Tekstur Objek, Serta Memahami Prinsip Dasar Kompresi Citra. Kelima, Mahasiswa Mampu Mengintegrasikan Seluruh Pengetahuan Dan Keterampilan Yang Diperoleh Untuk Mengembangkan Sebuah Proyek Aplikasi Sederhana Namun Fungsional, Seperti Sistem Deteksi Objek Dasar Atau Aplikasi Perbaikan Kualitas Citra Otomatis, Yang Mencerminkan Kemampuan Pemecahan Masalah Dan Inovasi Dalam Konteks Pengolahan Data Visual.

5.1 Konsep Dasar Filtering Dan Konvolusi

Filtering Atau Tapisan Merupakan Operasi Fundamental Dalam Pengolahan Citra Digital Yang Bertujuan Untuk Memodifikasi Atau Meningkatkan Kualitas Citra Dengan Cara Memanipulasi Nilai Piksel Berdasarkan Hubungan Ketetanggaannya. Operasi Ini Bekerja Dengan Menggunakan Suatu Kernel Atau Mask Yang Digeser Di Seluruh Piksel Citra, Kemudian Melakukan Perhitungan Matematis Antara Nilai Piksel Di Bawah Kernel Dengan Koefisien Kernel Tersebut. Proses Ini Dikenal Dengan Istilah Konvolusi, Yang Secara Matematis Didefinisikan Sebagai Integral Yang Mencerminkan Jumlah Tumpang Tindih Dari Dua Fungsi Setelah Salah Satu Fungsi Dibalik Dan Digeser . Dalam Implementasi Praktisnya, Konvolusi Diskrit Untuk Citra Dua Dimensi Dilakukan Dengan Menjumlahkan Hasil Perkalian Antara Elemen Kernel Dan Nilai Piksel Yang Bersesuaian Pada Setiap Posisi.

Konsep Konvolusi Memiliki Interpretasi Yang Beragam Dalam Berbagai Disiplin Ilmu. Dalam Teori Probabilitas, Konvolusi Menggambarkan Korelasi Silang Yang Menunjukkan Tingkat

Tumpang Tindih Atau Derajat Kesamaan Antara Dua Sekuens. Dalam Statistik, Konvolusi Merepresentasikan Rata-Rata Bergerak Terboboti Atas Sekuens Masukan Yang Ternormalisasi. Untuk Aplikasi Pengolahan Citra, Interpretasi Yang Paling Berguna Adalah Memandang Kernel Konvolusi Sebagai Detektor Fitur, Di Mana Keluaran Yang Besar Mengindikasikan Bahwa Fitur Tertentu Terdeteksi Dalam Citra . Pemahaman Tentang Konvolusi Ini Menjadi Fondasi Penting Karena Seluruh Operasi Filtering Spasial, Baik Untuk Penghalusan Maupun Penajaman, Menggunakan Mekanisme Konvolusi Sebagai Inti Pemrosesannya.

5.1.1 Klasifikasi Filter Berdasarkan Sifat Matematis

Para Peneliti Mengklasifikasikan Teknik Filtering Ke Dalam Dua Kategori Utama Berdasarkan Sifat Matematisnya, Yaitu Filter Linear Dan Filter Non-Linear . Filter Linear Bekerja Dengan Melakukan Operasi Linear Pada Piksel-Piksel Dalam Jendela Ketetangaan, Contohnya Adalah Penjumlahan Terboboti Seperti Pada Filter Rerata Dan Filter Gaussian. Filter Linear Umumnya Memiliki Kecepatan Komputasi Yang Relatif Tinggi Dan Implementasi Yang Sederhana, Namun Cenderung Menghasilkan Kualitas Yang Kurang Optimal Untuk Jenis Derau Tertentu. Data Eksperimental Menunjukkan Bahwa Filter Linear Relatif Cepat Tetapi Menghasilkan Hasil Yang Inferior, Sehingga Paling Baik Digunakan Sebagai Langkah Persiapan Awal.

Sementara Itu, Filter Non-Linear Bekerja Berdasarkan Operasi Yang Tidak Linear, Seperti Pengambilan Median Atau Pengurutan Nilai. Filter Non-Linear Telah Terbukti Lebih Robust Dan Dapat Diaplikasikan Pada Berbagai Tipe Derau, Meskipun Memerlukan Penyesuaian Parameter Yang Cermat . Contoh Filter Non-Linear Meliputi Median Filter, Filter Wiener, Dan Filter Anizotropik. Penelitian Menunjukkan Bahwa Filter Anizotropik Cocok Untuk Pemrosesan Citra Manual Maupun Aplikasi Real-Time Karena Menawarkan Keseimbangan Optimal Antara Kecepatan Pemrosesan Dan Kualitas Citra Yang Telah Dihilangkan Deraunya. Sementara Itu, Non-Local Means (NLM) Filter Menjadi Pilihan Optimal Untuk Pemrosesan Citra Tunggal Berkualitas Tinggi Karena Hasilnya Yang Superior, Meskipun Kecepatan Pemrosesannya Lebih Lambat

Tabel 5.1 Perbandingan Karakteristik Filter Linear Dan Non-Linear

Karakteristik	Filter Linear	Filter Non-Linear
Prinsip Kerja	Operasi Linear (Penjumlahan Terboboti)	Operasi Non-Linear (Median, Sorting)
Kecepatan	Relatif Cepat	Bervariasi, Umumnya Lebih Lambat
Kualitas Hasil	Inferior Untuk Derau Kompleks	Superior Untuk Berbagai Tipe Derau
Kebutuhan Parameter	Minimal	Penyesuaian Parameter
Contoh	Mean Filter, Gaussian Filter	Median Filter, Wiener Filter, NLM

5.1.2 Filter Penghalusan Untuk Reduksi Derau

Penghalusan Citra Atau Image Smoothing Merupakan Teknik Pengolahan Citra Yang Bertujuan Untuk Mengurangi Derau (Noise) Pada Sebuah Citra. Derau Tersebut Biasanya Muncul Sebagai Akibat Dari Hasil Penangkapan Citra Yang Tidak Sempurna, Baik Karena Keterbatasan Sensor, Gangguan Transmisi, Maupun Kondisi Lingkungan Saat Akuisisi . Dalam Penelitian Tentang Aplikasi Penghalusan Citra, Para Peneliti Menggunakan Tiga Metode Utama Yaitu Median Filtering, Wiener Filtering, Dan Gaussian Filtering. Citra Yang Diujikan Berupa Citra Warna 8 Bit, Yang Kemudian Ditambahkan Berbagai Jenis Derau Seperti Gaussian Noise, Salt And Pepper Noise, Rayleigh Noise, Uniform Noise, Dan Periodic Noise.

Filter Gaussian Merupakan Salah Satu Filter Linear Yang Paling Populer Untuk Penghalusan Citra. Filter Ini Menggunakan Kernel Dengan Bobot Yang Didasarkan Pada Distribusi Gaussian, Memberikan Bobot Lebih Besar Pada Piksel Yang Dekat Dengan Pusat Kernel Dan Bobot Semakin Kecil Untuk Piksel Yang Jauh. Karakteristik Ini Menghasilkan Penghalusan Yang Lebih Alami Dibandingkan Filter Rerata Biasa, Dengan Kemampuan Mempertahankan Tepi Yang Lebih Baik. Implementasi Filter Gaussian Dalam Python Menggunakan Opencv Dilakukan Dengan

BAB 6 DETEKSI TEPI

A. Deskripsi Mata Kuliah

Bab Ini Dirancang Untuk Membekali Mahasiswa Dengan Pemahaman Teoretis Dan Keterampilan Praktis Dalam Mengolah Serta Menganalisis Citra Digital Menggunakan Bahasa Pemrograman Python. Citra Digital Telah Menjadi Komponen Esensial Dalam Berbagai Bidang Kehidupan Modern, Mulai Dari Diagnostik Medis, Sistem Keamanan, Dokumentasi Ilmiah, Hingga Aplikasi Industri Dan Hiburan. Kemampuan Memanipulasi Dan Mengekstrak Informasi Dari Data Visual Menjadi Kompetensi Yang Sangat Diperlukan Di Era Digital Ini. Perkembangan Pesat Teknologi Kecerdasan Buatan Dan Visi Komputer Semakin Mengukuhkan Pentingnya Penguasaan Teknik Pengolahan Citra Sebagai Fondasi Utama Pengembangan Sistem Cerdas. Oleh Karena Itu, Mata Kuliah Ini Tidak Hanya Membahas Konsep Fundamental Tentang Representasi Dan Karakteristik Citra Digital, Tetapi Juga Secara Intensif mempraktikkan implementasi berbagai algoritma menggunakan pustaka-pustaka mutakhir seperti Opencv, Numpy, Dan Matplotlib. Mahasiswa Akan Mempelajari Bagaimana Citra Direpresentasikan Sebagai Matriks Numerik, Melakukan Operasi Dasar Peningkatan Kualitas Citra, Restorasi, Segmentasi, Transformasi Ruang Warna, Hingga Ekstraksi Fitur Untuk Analisis Lebih Lanjut. Dengan Pendekatan Pembelajaran Berbasis Proyek Dan Studi Kasus Kontekstual, Mata Kuliah Ini Bertujuan Menghasilkan Lulusan Yang Mampu Mengembangkan Solusi Inovatif Untuk Berbagai Permasalahan Nyata Yang Melibatkan Data Visual.

B. Kompetensi Pembelajaran

Setelah Menyelesaikan Mata Kuliah Ini, Mahasiswa Diharapkan Mampu Menguasai Dan Mengaplikasikan Konsep Serta Teknik Pengelolaan Citra Digital Secara Komprehensif. Pertama, Mahasiswa Mampu Menjelaskan Konsep Fundamental Citra Digital, Meliputi Proses Pembentukan Citra Melalui Sampling Dan Kuantisasi, Representasi Warna Dalam Berbagai Ruang Warna Seperti RGB, Grayscale, HSV, Dan CMYK, Serta Karakteristik Berbagai Format File Citra. Kedua, Mahasiswa Terampil

Menggunakan Bahasa Pemrograman Python Dan Pustaka Pendukungnya, Terutama Numpy Untuk Manipulasi Array Multidimensi, Matplotlib Untuk Visualisasi Data, Serta Opencv Dan Pillow Untuk Operasi Input-Output Dan Pemrosesan Citra. Ketiga, Mahasiswa Mampu Mengimplementasikan Berbagai Teknik Pemrosesan Citra, Dimulai Dari Operasi Dasar Aras Titik Dan Aras Ketetangaan, Peningkatan Kualitas Citra Melalui Peregangan Kontras, Perataan Histogram, Dan Filtering Spasial, Hingga Teknik Restorasi Citra Untuk Mengatasi Berbagai Jenis Derau. Keempat, Mahasiswa Dapat Melakukan Analisis Citra Lebih Lanjut Seperti Segmentasi Untuk Memisahkan Objek Dari Latar Belakang Menggunakan Metode Thresholding, Deteksi Tepi Untuk Mengidentifikasi Batas Objek, Ekstraksi Fitur Morfologi Untuk Mendeskripsikan Bentuk Dan Tekstur Objek, Serta Memahami Prinsip Dasar Kompresi Citra. Kelima, Mahasiswa Mampu Mengintegrasikan Seluruh Pengetahuan Dan Keterampilan Yang Diperoleh Untuk Mengembangkan Sebuah Proyek Aplikasi Sederhana Namun Fungsional, Seperti Sistem Deteksi Objek Dasar Atau Aplikasi Perbaikan Kualitas Citra Otomatis, Yang Mencerminkan Kemampuan Pemecahan Masalah Dan Inovasi Dalam Konteks Pengolahan Data Visual.

6.1 Konsep Dasar Deteksi Tepi

Deteksi Tepi Memegang Peranan Yang Cukup Penting Dalam Pengolahan Citra Terutama Dalam Aplikasi Pengenalan Objek . Para Praktisi Pengolahan Citra Mendefinisikan Deteksi Tepi Sebagai Proses Untuk Mencari Perbedaan Intensitas Yang Menyatakan Batas Suatu Objek Dalam Suatu Citra . Proses Deteksi Tepi Pada Suatu Citra Menghasilkan Tepi-Tepi Dari Suatu Citra Yang Merepresentasikan Garis Batas (Boundary) Dari Objek Yang Terdapat Pada Citra . Deteksi Tepi Pada Dasarnya Merupakan Salah Satu Teknik Analisis Mutu Citra Pada Domain Spasial Dan Juga Salah Satu Proses Pengolahan Citra Digital Yang Paling Awal Dilakukan Dan Paling Banyak Diteliti .

Tepi Dalam Citra Merepresentasikan Batas Antara Region Yang Berbeda. Beberapa Penyebab Munculnya Tepi Antara Lain Batas Antar Objek, Perubahan Orientasi Permukaan, Perubahan Tekstur, Dan Variasi Pencahayaan . Dalam Citra Grayscale, Tepi Merepresentasikan Perubahan Intensitas Yang Cepat, Misalnya

Pergeseran Dari Gelap Ke Terang Dalam Rentang Piksel Yang Sempit. Transisi Ini Secara Matematis Ditangkap Menggunakan Gradien, Yang Mengukur Perubahan Intensitas Piksel . Deteksi Tepi Berfungsi Untuk Mengidentifikasi Garis Batas Dari Suatu Objek Yang Terdapat Pada Citra Sehingga Dapat Dimanfaatkan Untuk Berbagai Keperluan Proses Pengolahan Citra Selanjutnya Seperti Pengenalan Nomor Mobil, Segmentasi Wajah, Identifikasi Wajah, Pencocokan Wajah, Dan Pengenalan Tulisan .

Dalam Implementasi Praktisnya, Mahasiswa Perlu Memahami Bahwa Citra Berwarna Umumnya Dikonversi Ke Grayscale Terlebih Dahulu Sebelum Menerapkan Teknik Deteksi Tepi Seperti Canny, Sobel, Atau Laplacian . Proses Konversi Ini Menyederhanakan Komputasi Dengan Mereduksi Citra Menjadi Satu Kanal Intensitas, Sehingga Algoritma Dapat Secara Efektif Mendeteksi Tepi Berdasarkan Perubahan Intensitas. Setelah Dikonversi Ke Grayscale, Algoritma-Algoritma Tersebut Dapat Secara Efektif Menonjolkan Tepi Dalam Citra .

6.1.1 Operator Gradien Berbasis Turunan Pertama

Operator Gradien Bekerja Dengan Menghitung Turunan Pertama Dari Fungsi Intensitas Citra. Pendekatan Ini Mendeteksi Tepi Pada Titik-Titik Di Mana Gradien Mencapai Nilai Maksimum. Beberapa Operator Gradien Yang Populer Meliputi Operator Roberts, Prewitt, Dan Sobel.

Operator Roberts Merupakan Operator Yang Paling Sederhana Dan Paling Sedikit Menemukan Pola Tepi Pada Citra Wajah Dibandingkan Dua Operator Lainnya Yaitu Prewitt Dan Sobel . Operator Ini Menggunakan Kernel 2x2 Untuk Menghitung Gradien Diagonal, Sehingga Responsnya Terhadap Tepi Diagonal Cukup Baik Namun Sensitivitasnya Terhadap Derau Relatif Tinggi Karena Ukuran Kernel Yang Kecil.

Operator Prewitt Menggunakan Kernel 3x3 Untuk Menghitung Gradien Horizontal Dan Vertikal. Penelitian Menunjukkan Bahwa Algoritma Prewitt Lebih Sensitif Terhadap Perubahan Intensitas Tetapi Memiliki Respons Yang Lebih Lemah Dibandingkan Sobel . Karakteristik Ini Membuat Operator Prewitt Lebih Sesuai Untuk Citra Dengan Gradasi Halus . Kernel Prewitt Untuk Arah Horizontal Dan Vertikal Dirancang Dengan Memberikan Bobot Yang Sama

Pada Semua Piksel Dalam Jendela Ketetangaan, Sehingga Implementasinya Relatif Sederhana.

Operator Sobel Merupakan Pengembangan Dari Operator Prewitt Dengan Memberikan Bobot Lebih Besar Pada Piksel Yang Lebih Dekat Dengan Pusat Kernel. Penelitian Perbandingan Menyimpulkan Bahwa Operator Sobel Menghasilkan Pola Tepi Yang Lebih Baik Kualitas Dan Kuantitasnya Daripada Menggunakan Operator Roberts Dan Prewitt . Algoritma Sobel Menghasilkan Tepi Yang Lebih Tajam Dengan Kontras Tinggi, Sehingga Lebih Disarankan Untuk Citra Dengan Kontras Tinggi . Kernel Sobel Untuk Arah Horizontal (Gx) Dan Vertikal (Gy) Mengimplementasikan Kombinasi Antara Penghalusan Gaussian Dan Diferensiasi, Membuatnya Kurang Sensitif Terhadap Derau Dibandingkan Operator Roberts.

Tabel 6.1 Operator Gradien Deteksi Tepi

Operator	Ukuran Kernel	Karakteristik Utama	Kelebihan	Kekurangan
Roberts	2x2	Gradien Diagonal	Sederhana, Cepat	Sensitif Terhadap Derau
Prewitt	3x3	Bobot Seragam	Sensitif Perubahan Intensitas	Respons Lebih Lemah
Sobel	3x3	Bobot Terpusat	Tepi Tajam, Kontras Tinggi	Kurang Sesuai Untuk Gradasi Halus

Dalam Implementasi Python Menggunakan Opencv, Mahasiswa Dapat Menggunakan Fungsi `Cv2.Sobel()` Untuk Menerapkan Operator Sobel. Fungsi Ini Menerima Parameter Berupa Citra Masukan, Kedalaman Output (`Cv2.CV_64F` Untuk Mempertahankan Nilai Negatif), Order Derivatif Arah X (`Dx`) Dan Arah Y (`Dy`), Serta Ukuran Kernel (`Ksize`) . Setelah Memperoleh Gradien Horizontal Dan Vertikal, Mahasiswa Dapat Menghitung Magnitude Gradien Menggunakan `Cv2.Magnitude()` Atau `Cv2.Addweighted()` Untuk Menggabungkan Kedua Komponen. Hasil Perhitungan Magnitude Ini Kemudian Dikonversi Ke Tipe

uint8_t menggunakan `Cv2.CvtColorScaleAbs()` agar dapat ditampilkan dengan benar .

6.1.2 Operator Berbasis Turunan Kedua: Laplacian

Operator Laplacian merupakan operator turunan kedua yang mendeteksi tepi berdasarkan persilangan nol (Zero-Crossing) dari turunan kedua. Berbeda dengan operator gradien yang bersifat directional, Laplacian bersifat non-directional dan mendeteksi tepi di semua arah secara simultan . Operator Laplacian didefinisikan sebagai jumlah turunan parsial kedua terhadap x dan y , sehingga menghitung laju perubahan dari gradien.

Kernel Laplacian yang umum digunakan meliputi kernel 4-tetangga yang hanya mempertimbangkan piksel atas, bawah, kiri, dan kanan, serta kernel 8-tetangga yang juga mempertimbangkan piksel diagonal. Kernel 4-tetangga memiliki koefisien -4 di pusat dan 1 di keempat tetangga orthogonal, sementara kernel 8-tetangga memiliki koefisien -8 di pusat dan 1 di kedelapan tetangga sekelilingnya . Operator Laplacian mengukur deviasi nilai piksel dari rata-rata piksel sekitarnya, sehingga responsnya tinggi di area dengan perubahan intensitas cepat.

Dalam implementasi `Opencv`, mahasiswa dapat menggunakan fungsi `Cv2.Laplacian()` yang menerima parameter citra masukan, kedalaman output, dan ukuran kernel . Fungsi ini mengaplikasikan kernel Laplacian pada citra dan mengembalikan hasil dalam tipe data yang ditentukan. Karena Laplacian merupakan turunan kedua, hasilnya sangat sensitif terhadap derau. Oleh karena itu, sebelum menerapkan Laplacian, mahasiswa perlu melakukan penghalusan citra terlebih dahulu menggunakan filter Gaussian untuk mengurangi amplifikasi derau.

Kombinasi antara filter Gaussian dan operator Laplacian dikenal dengan Laplacian of Gaussian (Log). Pendekatan ini pertama-tama menghaluskan citra dengan filter Gaussian untuk mereduksi derau, kemudian menerapkan operator Laplacian untuk mendeteksi tepi. Penelitian sistematis menunjukkan bahwa operator Log menawarkan akurasi tinggi pada citra MRI dengan waktu komputasi yang lebih efisien . Dalam aplikasi medis, Log

BAB 7 SEGMENTASI CITRA

A. Deskripsi Mata Kuliah

Bab Ini Membekali Mahasiswa Dengan Pemahaman Teoretis Dan Keterampilan Praktis Dalam Mengolah Serta Menganalisis Citra Digital Menggunakan Bahasa Pemrograman Python. Citra Digital Telah Menjadi Komponen Esensial Dalam Berbagai Bidang Kehidupan Modern, Mulai Dari Diagnostik Medis, Sistem Keamanan, Dokumentasi Ilmiah, Hingga Aplikasi Industri Dan Hiburan. Kemampuan Memanipulasi Dan Mengekstrak Informasi Dari Data Visual Menjadi Kompetensi Yang Sangat Diperlukan Di Era Digital Ini. Perkembangan Pesat Teknologi Kecerdasan Buatan Dan Visi Komputer Semakin Mengukuhkan Pentingnya Penguasaan Teknik Pengolahan Citra Sebagai Fondasi Utama Pengembangan Sistem Cerdas. Oleh Karena Itu, Mata Kuliah Ini Tidak Hanya Membahas Konsep Fundamental Tentang Representasi Dan Karakteristik Citra Digital, Tetapi Juga Secara Intensif Mempraktikkan Implementasi Berbagai Algoritma Menggunakan Pustaka-Pustaka Mutakhir Seperti Opencv, Numpy, Dan Matplotlib. Mahasiswa Akan Mempelajari Bagaimana Citra Direpresentasikan Sebagai Matriks Numerik, Melakukan Operasi Dasar Peningkatan Kualitas Citra, Restorasi, Segmentasi, Transformasi Ruang Warna, Hingga Ekstraksi Fitur Untuk Analisis Lebih Lanjut. Dengan Pendekatan Pembelajaran Berbasis Proyek Dan Studi Kasus Kontekstual, Mata Kuliah Ini Bertujuan Menghasilkan Lulusan Yang Mampu Mengembangkan Solusi Inovatif Untuk Berbagai Permasalahan Nyata Yang Melibatkan Data Visual.

B. Kompetensi Pembelajaran

Setelah Menyelesaikan Mata Kuliah Ini, Mahasiswa Diharapkan Mampu Menguasai Dan Mengaplikasikan Konsep Serta Teknik Pengelolaan Citra Digital Secara Komprehensif.

1. Mahasiswa Mampu Menjelaskan Konsep Fundamental Citra Digital, Meliputi Proses Pembentukan Citra Melalui Sampling Dan Kuantisasi, Representasi Warna Dalam Berbagai Ruang Warna Seperti RGB, Grayscale, HSV, Dan CMYK, Serta Karakteristik Berbagai Format File Citra.

2. Mahasiswa Terampil Menggunakan Bahasa Pemrograman Python Dan Pustaka Pendukungnya, Terutama Numpy Untuk Manipulasi Array Multidimensi, Matplotlib Untuk Visualisasi Data, Serta Opencv Dan Pillow Untuk Operasi Input-Output Dan Pemrosesan Citra.
3. Mahasiswa Mampu Mengimplementasikan Berbagai Teknik Pemrosesan Citra, Dimulai Dari Operasi Dasar Aras Titik Dan Aras Ketetangaan, Peningkatan Kualitas Citra Melalui Peregangan Kontras, Perataan Histogram, Dan Filtering Spasial, Hingga Teknik Restorasi Citra Untuk Mengatasi Berbagai Jenis Derau.
4. Mahasiswa Dapat Melakukan Analisis Citra Lebih Lanjut Seperti Segmentasi Untuk Memisahkan Objek Dari Latar Belakang Menggunakan Metode Thresholding, Deteksi Tepi Untuk Mengidentifikasi Batas Objek, Ekstraksi Fitur Morfologi Untuk Mendeskripsikan Bentuk Dan Tekstur Objek, Serta Memahami Prinsip Dasar Kompresi Citra.
5. Mahasiswa Mampu Mengintegrasikan Seluruh Pengetahuan Dan Keterampilan Yang Diperoleh Untuk Mengembangkan Sebuah Proyek Aplikasi Sederhana Namun Fungsional, Seperti Sistem Deteksi Objek Dasar Atau Aplikasi Perbaikan Kualitas Citra Otomatis, Yang Mencerminkan Kemampuan Pemecahan Masalah Dan Inovasi Dalam Konteks Pengolahan Data Visual.

7.1 Konsep Dasar Dan Definisi Segmentasi

Segmentasi Citra Merupakan Proses Penting Dalam Pengolahan Citra Digital Yang Bertujuan Untuk Mempartisi Citra Menjadi Wilayah-Wilayah Atau Region Yang Memiliki Makna Tertentu, Seperti Memisahkan Objek Dari Latar Belakangnya. Para Praktisi Pengolahan Citra Mendefinisikan Segmentasi Sebagai Langkah Awal Dalam Melakukan Pengolahan Citra, Pengenalan Pola, Dan Visi Komputer, Yang Menjadi Faktor Terpenting Karena Hampir Sebagian Besar Pengolahan Citra Bergantung Pada Hasil Segmentasi (Firdaus, 2013) . Segmentasi Yang Baik Akan Menyederhanakan Atau Mengubah Representasi Citra Menjadi Sesuatu Yang Lebih Bermakna Dan Lebih Mudah Dianalisis . Jika Segmentasi Dilakukan Dengan Baik, Maka Seluruh Tahapan Analisis Citra Selanjutnya Menjadi Lebih Sederhana. Dengan Demikian, Kualitas Dan Keandalan Segmentasi Menentukan

Keberhasilan Analisis Citra Secara Keseluruhan, Meskipun Mempartisi Citra Ke Dalam Segmen Yang Tepat Seringkali Merupakan Masalah Yang Sangat Menantang (Dey, 2018) .

Segmentasi Citra Dapat Diklasifikasikan Ke Dalam Beberapa Pendekatan Utama Berdasarkan Strategi Yang Digunakan. Tiga Pendekatan Utama Dalam Segmentasi Citra Meliputi Segmentasi Berbasis Tepi (Edge-Based Segmentation), Segmentasi Berbasis Wilayah (Region-Based Segmentation), Dan Segmentasi Berbasis Morfologi (Morphological Segmentation) . Pendekatan Berbasis Tepi Bekerja Dengan Mendeteksi Perubahan Intensitas Yang Tajam Sebagai Batas Antar Objek. Pendekatan Berbasis Wilayah Mengelompokkan Piksel-Piksel Yang Memiliki Kesamaan Atribut Ke Dalam Region Yang Sama. Sementara Pendekatan Berbasis Morfologi Memanfaatkan Operasi-Operasi Morfologi Untuk Memisahkan Objek Berdasarkan Bentuknya. Dalam Implementasi Praktisnya, Mahasiswa Perlu Memahami Bahwa Pemilihan Pendekatan Segmentasi Sangat Bergantung Pada Karakteristik Citra Dan Tujuan Analisis Yang Ingin Dicapai .

1.1.1 Segmentasi Berbasis Thresholding

Segmentasi Berbasis Thresholding Merupakan Teknik Yang Paling Sederhana Dan Banyak Digunakan Dalam Berbagai Aplikasi. Teknik Ini Bekerja Dengan Memisahkan Piksel-Piksel Berdasarkan Nilai Intensitasnya Menggunakan Satu Atau Lebih Nilai Ambang Batas (Threshold). Prinsip Dasarnya Adalah Membandingkan Setiap Piksel Dengan Nilai Threshold, Kemudian Mengelompokkannya Ke Dalam Kelas Foreground Atau Background. Thresholding Global Menggunakan Satu Nilai Threshold Yang Berlaku Untuk Seluruh Bagian Citra. Metode Otsu Merupakan Algoritma Thresholding Global Yang Paling Populer Karena Kemampuannya Menentukan Nilai Threshold Secara Otomatis Berdasarkan Distribusi Intensitas Piksel . Metode Ini Menghitung Nilai Threshold Optimal Dengan Cara Memaksimalkan Varians Antar Kelas (Between-Class Variance) Dan Meminimalkan Varians Dalam Kelas (Within-Class Variance) Dari Piksel-Piksel Yang Terpisah Oleh Threshold Tersebut .

Penelitian Perbandingan Antara Algoritma Otsu Thresholding Dan Algoritma Region Growing Menunjukkan Hasil Yang Menarik. Dari Segi Komputasi, Region Growing Unggul Dengan Nilai 0,03

BAB 8 KONTUR DAN REPRESENTASI

A. Deskripsi Mata Kuliah

Bab Ini Membahas Konsep, Metode, Dan Teknik Komputasi Untuk Memperoleh, Merepresentasikan, Menganalisis, Dan Memanipulasi Informasi Visual Dalam Bentuk Digital Secara Sistematis. Mata Kuliah Ini Menekankan Pemahaman Mahasiswa Terhadap Struktur Citra Digital, Termasuk Pikel, Region, Kontur, Dan Representasi Bentuk Sebagai Komponen Fundamental Dalam Analisis Citra. Sistem Pengolahan Citra Digital Menggunakan Pendekatan Matematis Dan Algoritmik Untuk Mengekstraksi Informasi Penting Dari Citra, Sehingga Mahasiswa Harus Memahami Bagaimana Sistem Komputer Merepresentasikan Bentuk Objek Secara Numerik Dan Geometrik Untuk Mendukung Proses Analisis Lebih Lanjut Seperti Klasifikasi, Segmentasi, Dan Pengenalan Pola.

Mahasiswa Mempelajari Konsep Kontur Sebagai Batas Antara Dua Region Yang Berbeda Dalam Citra Serta Mempelajari Berbagai Metode Representasi Bentuk Seperti Chain Code, Polygonal Approximation, Signature Dan Skeleton. Representasi Bentuk Memungkinkan Sistem Komputer Menyimpan Informasi Geometri Objek Secara Efisien Dan Akurat Untuk Keperluan Analisis Dan Pengenalan Objek (Sonka Et Al., 2014). Bab Ini Mengintegrasikan Pendekatan Praktikum Menggunakan Bahasa Pemrograman Python Untuk Mengimplementasikan Algoritma Ekstraksi Kontur Dan Representasi Bentuk Secara Langsung. Mahasiswa Memperoleh Pengalaman Praktis Dalam Menggunakan Pustaka Pengolahan Citra Seperti Opencv Dan Scikit-Image Untuk Memahami Hubungan Antara Teori Dan Implementasi. Mata Kuliah Ini Mendukung Capaian Pembelajaran Berbasis Outcome-Based Education (OBE) Dengan Menekankan Kemampuan Analisis, Implementasi Algoritma, Serta Evaluasi Representasi Bentuk Dalam Sistem Pengolahan Citra Digital.

B. Kompetensi Pembelajaran

Mahasiswa Menguasai Konsep Dasar Kontur Dan Representasi Bentuk Dalam Citra Digital Secara Teoritis Dan Komputasional. Mahasiswa Menjelaskan Definisi Kontur Sebagai Batas Objek Serta

Menjelaskan Pentingnya Kontur Dalam Analisis Bentuk Dan Segmentasi Citra Digital. Mahasiswa Menganalisis Struktur Geometrik Objek Berdasarkan Informasi Kontur Dan Mengevaluasi Metode Representasi Bentuk Berdasarkan Kebutuhan Sistem Pengolahan Citra Digital.

1. Mahasiswa Mengimplementasikan Algoritma Ekstraksi Kontur Menggunakan Metode Berbasis Gradien, Thresholding, Dan Edge Detection Dengan Menggunakan Bahasa Pemrograman Python.
2. Mahasiswa Menerapkan Metode Representasi Bentuk Seperti Chain Code, Polygon Approximation, Dan Skeletonization Untuk Menyederhanakan Bentuk Objek Secara Efisien.
3. Mahasiswa Mengevaluasi Kelebihan Dan Keterbatasan Setiap Metode Representasi Dalam Konteks Pengenalan Objek Dan Analisis Citra.
4. Mahasiswa Merancang Solusi Komputasi Yang Menggunakan Kontur Sebagai Fitur Utama Dalam Sistem Analisis Citra Digital.
5. Mahasiswa Mengembangkan Keterampilan Praktis Dalam Menggunakan Pustaka Opencv Dan Scikit-Image Untuk Ekstraksi Dan Representasi Kontur.
6. Mahasiswa Juga Menunjukkan Kemampuan Analitis Dalam Menginterpretasikan Hasil Representasi Bentuk Secara Matematis Dan Visual Untuk Mendukung Pengambilan Keputusan Dalam Sistem Berbasis Citra Digital.
7. Mahasiswa Mencapai Capaian Pembelajaran Yang Mencakup Kemampuan Kognitif, Afektif, Dan Psikomotorik Sesuai Standar SN-Dikti Dan Pendekatan Outcome-Based Education Melalui Kegiatan Pembelajaran Teori, Praktikum Dan Evaluasi Berbasis Proyek.

8.1 Konsep Kontur Dalam Citra Digital

Kontur Merupakan Batas Yang Memisahkan Objek Dari Latar Belakang Atau Region Lain Dalam Citra Digital. Sistem Pengolahan Citra Menggunakan Kontur Untuk Merepresentasikan Bentuk Objek Secara Geometrik. Kontur Menyediakan Informasi Penting Mengenai Struktur Dan Bentuk Objek Sehingga Sistem Komputer Dapat Mengidentifikasi, Mengklasifikasikan, Dan Menganalisis Objek Secara Efektif (Gonzalez & Woods, 2018). Kontur Terbentuk Dari Perubahan Intensitas Pikel Yang Signifikan Pada Citra.

Sistem Mendeteksi Perubahan Intensitas Menggunakan Operator Gradien Seperti Sobel, Prewitt, Dan Canny. Operator Gradien Menghitung Turunan Pertama Intensitas Citra Untuk Menentukan Lokasi Perubahan Intensitas Maksimum Yang Menunjukkan Keberadaan Tepi Objek (Sonka Et Al., 2014).

Secara Matematis, Gradien Citra Didefinisikan Sebagai:

$$\nabla f(x, y) = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

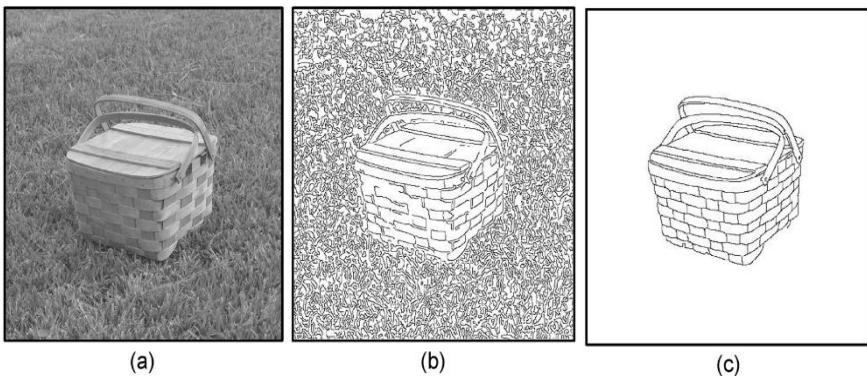
Magnitude Gradien Dihitung Menggunakan Persamaan:

$$|\nabla f| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

Magnitude Gradien Menunjukkan Kekuatan Tepi. Sistem Menggunakan Nilai Magnitude Gradien Yang Tinggi Untuk Menentukan Lokasi Kontur Objek.

Kontur Memiliki Karakteristik Berikut:

1. Kontur Merepresentasikan Batas Objek
2. Kontur Memiliki Struktur Geometrik
3. Kontur Menyimpan Informasi Bentuk Objek
4. Kontur Memungkinkan Analisis Bentuk Secara Efisien



Gambar 8.1 Kontur Objek Sebagai Batas Region Dalam Citra Digital

Gambar diatas Menunjukkan Bahwa Sistem Mengidentifikasi Kontur Sebagai Garis Batas Yang Mengelilingi Objek. Sistem Menggunakan Informasi Kontur Untuk Merepresentasikan Bentuk Objek Tanpa Menyimpan Seluruh Piksel Dalam Region Objek, Sehingga Meningkatkan Efisiensi Penyimpanan Dan Komputasi.

8.1.1 Ekstraksi Kontur

Sistem Mengekstraksi Kontur Menggunakan Metode Segmentasi Dan Deteksi Tepi. Metode Deteksi Tepi Seperti Canny Menghasilkan Citra Biner Yang Menunjukkan Lokasi Tepi. Sistem Kemudian Menggunakan Algoritma Contour Tracing Untuk Memperoleh Koordinat Piksel Yang Membentuk Kontur (Gonzalez & Woods, 2018).

Algoritma Ekstraksi Kontur Melibatkan Langkah Berikut:

1. Konversi Citra Ke Grayscale
2. Terapkan Smoothing Untuk Mengurangi Noise
3. Terapkan Edge Detection
4. Identifikasi Kontur Menggunakan Contour Tracing

Contoh Implementasi Python:

```
Import Cv2
```

```
Import Numpy As Np
```

```
# Membaca Citra
```

```
Image = Cv2.Imread('Object.Png')
```

```
Gray = Cv2.CvtColor(Image, Cv2.COLOR_BGR2GRAY)
```

```
# Edge Detection
```

```
Edges = Cv2.Canny(Gray, 100, 200)
```

```
# Ekstraksi Kontur
```

```
Contours, Hierarchy = Cv2.Findcontours(Edges,
```

```
    Cv2.RETR_EXTERNAL, Cv2.CHAIN_APPROX_NONE)
```

```
# Menampilkan Kontur
```

```
Output = Image.Copy()
```

```
Cv2.Drawcontours(Output, Contours, -1, (0,255,0), 2)
```

```
Cv2.Imshow("Kontur", Output)
Cv2.Waitkey(0)
Cv2.DestroyAllWindows()
```

Note :

Program Tersebut Menggunakan Fungsi FindContours Untuk Memperoleh Koordinat Kontur Objek. Fungsi Tersebut Mengembalikan Daftar Titik Yang Merepresentasikan Bentuk Objek.

8.1.2 Representasi Kontur

Representasi Kontur Memungkinkan Sistem Menyimpan Bentuk Objek Secara Efisien. Sistem Menggunakan Berbagai Metode Representasi Kontur Tergantung Pada Kebutuhan Aplikasi.

8.1.3 Chain Code

Chain Code Merepresentasikan Kontur Menggunakan Arah Antar Piksel Bertetangga. Metode Ini Menggunakan Sistem Kode Arah 4-Arah Atau 8-Arah (Freeman, 1974).

Tabel 8.1 Representasi Arah Chain Code

Kode	Arah
0	Kanan
1	Kanan Atas
2	Atas
3	Kiri Atas
4	Kiri
5	Kiri Bawah
6	Bawah
7	Kanan Bawah

Chain Code Memungkinkan Sistem Merepresentasikan Kontur Menggunakan Deretan Angka. Sistem Menggunakan Representasi Tersebut Untuk Analisis Bentuk Dan Pengenalan Objek.

8.1.4 Polygonal Approximation

Polygonal Approximation Menyederhanakan Kontur Dengan Mengurangi Jumlah Titik. Sistem Menggunakan Algoritma Douglas-Peucker Untuk Menghilangkan Titik Yang Tidak Signifikan (Douglas

& Peucker, 1973). Metode Ini Menghasilkan Representasi Bentuk Menggunakan Poligon Dengan Jumlah Titik Minimal.

Contoh Implementasi Python:

```
Epsilon = 0.01 * Cv2.Arclength(Contours[0], True)
```

```
Approx = Cv2.Approxpolydp(Contours[0], Epsilon, True)
```

Metode Tersebut Mengurangi Kompleksitas Bentuk Tanpa Menghilangkan Karakteristik Utama Objek.

8.1.5 Skeleton Representation

Skeleton Merupakan Representasi Bentuk Yang Mempertahankan Struktur Topologi Objek. Skeleton Merepresentasikan Objek Menggunakan Garis Tengah Objek (Zhang & Suen, 1984). Skeleton Membantu Sistem Memahami Struktur Objek Secara Efisien.



(a) (b) (c)

Gambar 8.2 Representasi Skeleton Objek

Skeleton Mempertahankan Bentuk Objek Dengan Jumlah Pixel Minimal Sehingga Meningkatkan Efisiensi Komputasi.

8.1.6 Signature Representation

Signature Merepresentasikan Kontur Menggunakan Fungsi Satu Dimensi. Sistem Menghitung Jarak Antara Centroid Dan Titik Kontur.

Persamaan:

$$r(\theta) = \sqrt{(x - x_c)^2 + (y - y_c)^2}$$

Representasi Signature Membantu Sistem Membandingkan Bentuk Objek Secara Matematis.

8.1.7 Descriptor Bentuk

Descriptor Bentuk Merupakan Fitur Numerik Yang Merepresentasikan Bentuk Objek. Descriptor Bentuk Meliputi:

1. Area
2. Perimeter
3. Circularity
4. Aspect Ratio

Circularity Dihitung Menggunakan:

$$C = \frac{4\pi A}{P^2}$$

Descriptor Bentuk Membantu Sistem Dalam Klasifikasi Objek.

8.2 Tujuan Praktikum

Setelah Menyelesaikan Praktikum Ini, Mahasiswa Diharapkan Mampu:

1. Memahami Konsep Kontur Dan Hubungannya Dengan Deteksi Tepi
2. Mengimplementasikan Deteksi Dan Ekstraksi Kontur Menggunakan Opencv
3. Menganalisis Properti Kontur Seperti Luas, Perimeter, Dan Bounding Box
4. Menerapkan Representasi Kontur Dengan Chain Code Dan Fourier Descriptor
5. Menggunakan Kontur Untuk Analisis Gerakan Bibir Dalam Sistem Audio-Visual

8.2.1 Alat Dan Bahan

- 1 **Perangkat Keras:**
 - Komputer/Laptop
- 2 **Perangkat Lunak:**
 - Python 3.X
 - Opencv (Opencv-Python)
 - Numpy
 - Matplotlib

- Scipy (Untuk Fourier Descriptor)

8.2.2 Langkah Praktikum

Langkah 8.1: Deteksi Tepi Canny Dan Menemukan Kontur

```
Import Cv2
```

```
Import Numpy As Np
```

```
Import Matplotlib.Pyplot As Plt
```

```
# Baca Citra Grayscale
```

```
Img = Cv2.Imread('Contoh.Jpg', Cv2.IMREAD_GRAYSCALE)
```

```
If Img Is None:
```

```
    Img = Np.Random.Randint(0, 256, (300, 400), Dtype=Np.Uint8)
```

```
# Deteksi Tepi Canny
```

```
Edges = Cv2.Canny(Img, Threshold1=50, Threshold2=150)
```

```
# Temukan Kontur
```

```
# RETR_EXTERNAL Hanya Kontur Luar, RETR_TREE Semua  
    Hierarki
```

```
Contours, Hierarchy = Cv2.Findcontours(Edges,  
    Cv2.RETR_TREE, Cv2.CHAIN_APPROX_SIMPLE)
```

```
# Buat Citra RGB Untuk Menggambar Kontur
```

```
Img_Contour = Cv2.CvtColor(Img, Cv2.COLOR_GRAY2BGR)
```

```
Cv2.Drawcontours(Img_Contour, Contours, -1, (0,255,0), 2) # -1 =  
    Semua Kontur
```

```
Print(F"Jumlah Kontur Ditemukan: {Len(Contours)}")
```

```
Plt.Figure(Figsize=(12,4))
```

```
Plt.Subplot(1,3,1)
```

```
Plt.Imshow(Img, Cmap='Gray')
```

```
Plt.Title('Citra Asli')
```

```
Plt.Axis('Off')
```

```
Plt.Subplot(1,3,2)
```

```
Plt.Imshow(Edges, Cmap='Gray')
```

```
Plt.Title('Tepi Canny')
```

```
Plt.Axis('Off')
```

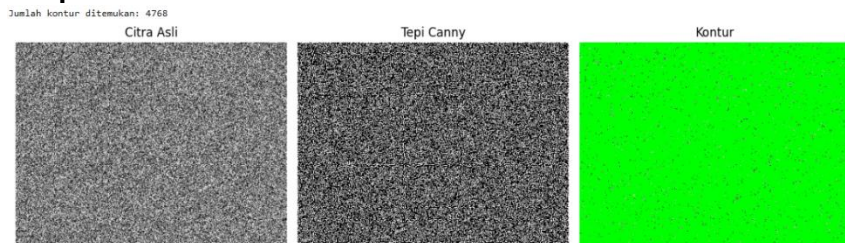
```

Plt.Subplot(1,3,3)
Plt.Imshow(Cv2.CvtColor(Img_Contour, Cv2.COLOR_BGR2RGB))
Plt.Title('Kontur')
Plt.Axis('Off')

Plt.Tight_Layout()
Plt.Show()

```

Output :



Langkah 8.2: Menyaring Kontur Berdasarkan Luas

Hitung Luas Setiap Kontur

```

Contour_Areas = [Cv2.Contourarea(Cnt) For Cnt In Contours]
Print(F"Luas Kontur (10 Pertama): {Contour_Areas[:10]}")

```

Pilih Kontur Dengan Luas > 500 Piksel

```

Min_Area = 500
Large_Contours = [Cnt For Cnt In Contours If Cv2.Contourarea(Cnt)
                  > Min_Area]

```

Gambar Kontur Besar

```

Img_Large = Cv2.CvtColor(Img, Cv2.COLOR_GRAY2BGR)
Cv2.Drawcontours(Img_Large, Large_Contours, -1, (0,255,0), 2)

```

Hitung Bounding Box Untuk Kontur Terbesar (Indeks 0)

```

If Large_Contours:
    Cnt = Large_Contours[0]
    X, Y, W, H = Cv2.Boundingrect(Cnt)
    Cv2.Rectangle(Img_Large, (X,Y), (X+W,Y+H), (255,0,0), 2)

```

```

Plt.Figure(Figsize=(12,4))

```

```

plt.subplot(1,2,1)
plt.imshow(Cv2.cvtColor(img_contour, Cv2.COLOR_BGR2RGB))
plt.title('Semua Kontur')
plt.axis('off')

```

```

plt.subplot(1,2,2)
plt.imshow(Cv2.cvtColor(img_large, Cv2.COLOR_BGR2RGB))
plt.title(f'Kontur Luas > {min_area} + Bounding Box')
plt.axis('off')

```

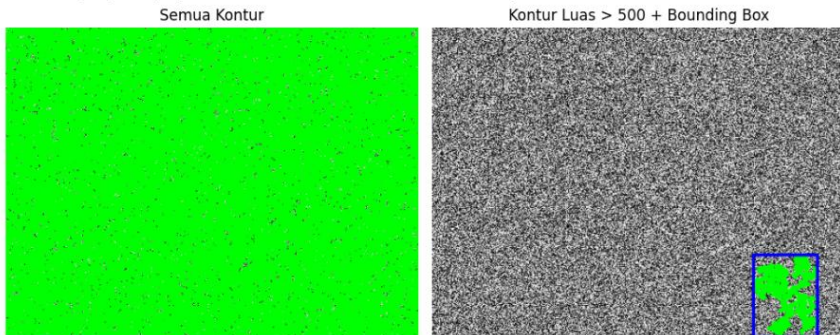
```

plt.tight_layout()
plt.show()

```

Output:

Luas kontur (10 pertama): [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]



Langkah 8.3: Menghitung Momen Dan Properti Kontur

If Large_Contours:

```

    cnt = Large_Contours[0] # Ambil Kontur Terbesar

```

```

    # Momen

```

```

    M = Cv2.Moments(cnt)

```

```

    If M['M00'] != 0:

```

```

        Cx = Int(M['M10'] / M['M00'])

```

```

        Cy = Int(M['M01'] / M['M00'])

```

```

        Print(F"Centroid: ({Cx}, {Cy})")

```

```

    Else:

```

```

        Print("Luas Nol")

```

```

    # Luas Dan Perimeter

```

```

Area = Cv2.Contourarea(Cnt)
Perimeter = Cv2.Arclength(Cnt, True)
Print(F"Luas: {Area:.2f}")
Print(F"Perimeter: {Perimeter:.2f}")

# Bounding Box Dan Rasio Aspek
X, Y, W, H = Cv2.Boundingrect(Cnt)
Aspect_Ratio = W / H
Print(F"Aspect Ratio: {Aspect_Ratio:.2f}")

# Elips Fit
If Len(Cnt) >= 5:
    Ellipse = Cv2.Fitellipse(Cnt)
    Print(F"Ellipse Center: {Ellipse[0]}, Axes: {Ellipse[1]}, Angle:
    {Ellipse[2]:.2f}")

# Gambar Centroid
Img_Props = Cv2.CvtColor(Img, Cv2.COLOR_GRAY2BGR)
Cv2.Drawcontours(Img_Props, [Cnt], -1, (0,255,0), 2)
If M['M00'] != 0:
    Cv2.Circle(Img_Props, (Cx,Cy), 5, (0,0,255), -1)

Plt.Imshow(Cv2.CvtColor(Img_Props, Cv2.COLOR_BGR2RGB))
Plt.Title('Kontur Dengan Centroid')
Plt.Axis('Off')
Plt.Show()

```

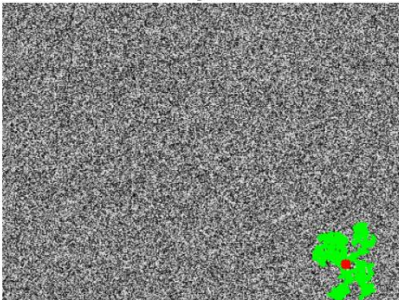
Output :

```

Centroid: (343, 261)
Luas: 503.50
Perimeter: 1639.36
Aspect Ratio: 0.78
Ellipse center: (344.59423828125, 257.7825012207031), axes: (53.304962158203125, 66.64143371582031), angle: 5.23

```

Kontur dengan Centroid



Langkah 8.4: Implementasi Chain Code Sederhana

Def Freeman_Chain_Code(Contour):

"""

Menghasilkan Chain Code 8-Arah Dari Kontur.

Contour: Array (N,1,2) Dari Opencv

"""

Pts = Contour[:,0,:].Aastype(Int)

Directions = []

For I In Range(1, Len(Pts)):

 Dx = Pts[I,0] - Pts[I-1,0]

 Dy = Pts[I,1] - Pts[I-1,1]

 # Kode 8-Arah: 0=Timur, 1=Tenggara, 2=Selatan,
 3=Baratdaya,

 # 4=Barat, 5=Baratlaut, 6=Utara, 7=Timurlaut

 If Dx == 1 And Dy == 0:

 Code = 0

 Elif Dx == 1 And Dy == 1:

 Code = 1

 Elif Dx == 0 And Dy == 1:

 Code = 2

 Elif Dx == -1 And Dy == 1:

 Code = 3

 Elif Dx == -1 And Dy == 0:

 Code = 4

 Elif Dx == -1 And Dy == -1:

 Code = 5

 Elif Dx == 0 And Dy == -1:

 Code = 6

 Elif Dx == 1 And Dy == -1:

 Code = 7

 Else:

 Code = -1 # Seharusnya Tidak Terjadi Jika Kontur Kontinu

 Directions.Append(Code)

Return Directions

If Large_Contours:

```

Cnt = Large_Contours[0]
Chain = Freeman_Chain_Code(Cnt)
Print("Chain Code (10 Pertama):", Chain[:10])
Print("Panjang Chain Code:", Len(Chain))

# Histogram Arah
Plt.Figure(Figsize=(10, 5))
Plt.Hist(Chain, Bins=8, Range=(-0.5, 7.5), Rwidth=0.8)
Plt.Title('Histogram Arah Chain Code')
Plt.Xlabel('Kode Arah')
Plt.Ylabel('Frekuensi')
Plt.Xticks(Range(8))
Plt.Grid(True, Alpha=0.3)
Plt.Show()

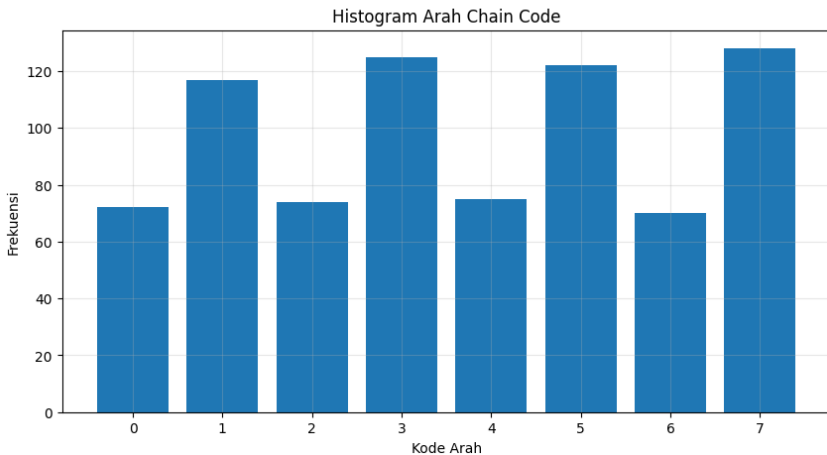
```

Output :

```

Chain code (10 pertama): [3, 1, 3, 4, 0, 1, 2, 3, -1, -1]
Panjang chain code: 1034

```



Langkah 8.5: Menghitung Fourier Descriptor

From Scipy.Fftpack Import Fft, Ifft

```

Def Fourier_Descriptor(Contour, N_Descriptors=10,
    Normalize=True):
    """

```

Menghitung Fourier Descriptor Dari Kontur.

BAB 9 PEMAMPATAN CITRA

A. Deskripsi Mata Kuliah

Mata Kuliah Pengelolaan Citra Digital Membahas Konsep, Metode, Dan Teknik Komputasi Untuk Merepresentasikan, Menyimpan, Memproses, Dan Mentransmisikan Citra Digital Secara Efisien. Salah Satu Komponen Utama Dalam Pengelolaan Citra Digital Adalah Pemampatan Citra Yang Bertujuan Untuk Mengurangi Ukuran Data Tanpa Menghilangkan Informasi Penting Yang Diperlukan Untuk Analisis Visual Maupun Komputasional. Sistem Pengolahan Citra Digital Menggunakan Teknik Pemampatan Untuk Mengoptimalkan Penggunaan Memori Penyimpanan Dan Bandwidth Transmisi, Terutama Dalam Aplikasi Multimedia, Sistem Medis, Sistem Pengawasan, Dan Sistem Komunikasi Digital.

Pemampatan Citra Bekerja Dengan Menghilangkan Redundansi Data Dan Merepresentasikan Informasi Citra Dalam Bentuk Yang Lebih Efisien Secara Matematis Dan Algoritmik. Sistem Menggunakan Pendekatan Pemampatan Lossless Untuk Mempertahankan Seluruh Informasi Citra Dan Menggunakan Pendekatan Lossy Untuk Mencapai Rasio Kompresi Yang Lebih Tinggi Dengan Toleransi Kehilangan Informasi Yang Minimal. Mahasiswa Mempelajari Prinsip Redundansi Spasial, Redundansi Statistik, Dan Redundansi Psikovisual Sebagai Dasar Pemampatan Citra Digital.

Mata Kuliah Ini Mengintegrasikan Pendekatan Teoritis Dan Praktis Menggunakan Bahasa Pemrograman Python Dan Pustaka Opencv Serta Pustaka Terkait Lainnya. Mahasiswa Memahami Bagaimana Algoritma Pemampatan Bekerja Dan Mengimplementasikan Metode Tersebut Secara Langsung. Mata Kuliah Ini Mendukung Capaian Pembelajaran Berbasis Outcome-Based Education (OBE) Dan Memenuhi Standar SN-Dikti Dengan Menekankan Kemampuan Analisis, Implementasi, Dan Evaluasi Teknik Pemampatan Citra Digital Secara Sistematis.

B. Kompetensi Pembelajaran

1. Mahasiswa Menguasai Konsep Dasar Pemampatan Citra Digital Secara Teoritis Dan Komputasional.

2. Mahasiswa Menjelaskan Prinsip Redundansi Data Dalam Citra Digital Dan Menjelaskan Bagaimana Sistem Menggunakan Redundansi Tersebut Untuk Mengurangi Ukuran Data Secara Efisien.
3. Mahasiswa Menganalisis Perbedaan Antara Pemampatan Lossless Dan Lossy Serta Mengevaluasi Kelebihan Dan Keterbatasan Masing-Masing Metode Dalam Konteks Aplikasi Pengolahan Citra Digital.
4. Mahasiswa Mengimplementasikan Algoritma Pemampatan Citra Menggunakan Bahasa Pemrograman Python Dan Pustaka Pengolahan Citra.
5. Mahasiswa Menerapkan Metode Pemampatan Lossless Seperti Run-Length Encoding Dan Huffman Coding Serta Menerapkan Metode Lossy Seperti Transform Coding Dan JPEG Compression.
6. Mahasiswa Mengevaluasi Kinerja Metode Pemampatan Menggunakan Parameter Rasio Kompresi, Distorsi, Dan Kualitas Citra Hasil Rekonstruksi.
7. Mahasiswa Merancang Sistem Komputasi Yang Menggunakan Pemampatan Citra Untuk Meningkatkan Efisiensi Penyimpanan Dan Transmisi Data. Mahasiswa Menganalisis Hasil Pemampatan Menggunakan Pendekatan Matematis Dan Visual. Mahasiswa Mengembangkan Kemampuan Analitis Dan Outcome-Based Education Melalui Kegiatan Praktikum, Analisis, Dan Evaluasi Berbasis Proyek.

9.1 Konsep Dasar Pemampatan Citra

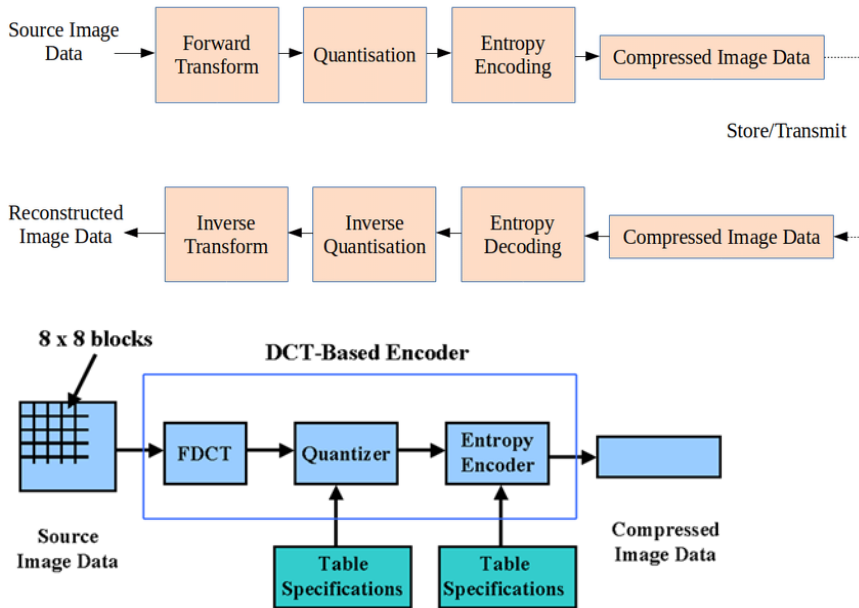
Pemampatan Citra Merupakan Proses Mengurangi Jumlah Bit Yang Diperlukan Untuk Merepresentasikan Citra Digital. Sistem Melakukan Pemampatan Dengan Menghilangkan Redundansi Data Dan Merepresentasikan Informasi Citra Secara Lebih Efisien (Gonzalez & Woods, 2018).

Citra Digital Memiliki Redundansi Berikut:

1. Redundansi Spasial
2. Redundansi Statistik
3. Redundansi Psikovisual

Redundansi Spasial Terjadi Karena Piksel Yang Berdekatan Memiliki Nilai Intensitas Yang Mirip. Redundansi Statistik Terjadi Karena Distribusi Probabilitas Piksel Tidak Seragam. Redundansi Psikovisual Terjadi Karena Sistem Visual Manusia Tidak Sensitif Terhadap Semua Informasi Visual (Sayood, 2017).

Gambar Berikut Menunjukkan Konsep Pemampatan Citra Digital.



Gambar 9.1 Proses Pemampatan Citra Digital Menggunakan Encoder Dan Decoder

Gambar diatas Menunjukkan Bahwa Sistem Encoder Mengubah Citra Asli Menjadi Representasi Terkompresi Dan Sistem Decoder Merekonstruksi Citra Dari Representasi Tersebut. Sistem Mengurangi Ukuran Data Selama Proses Encoding Dan Mempertahankan Informasi Penting Selama Proses Decoding.

Rasio Kompresi Dihitung Menggunakan Persamaan:

$$\text{Compression Ratio} = \frac{\text{Ukuran Data Terkompresi}}{\text{Ukuran Data Asli}}$$

Nilai Rasio Kompresi Yang Lebih Tinggi Menunjukkan Efisiensi Pemampatan Yang Lebih Baik.

9.1.1 Pemampatan Lossless

Pemampatan Lossless Mempertahankan Seluruh Informasi Citra Tanpa Kehilangan Data. Sistem Merekonstruksi Citra Hasil Pemampatan Secara Identik Dengan Citra Asli (Sayood, 2017).

Metode Lossless Meliputi:

1. Run-Length Encoding
2. Huffman Coding
3. Arithmetic Coding

Metode Run-Length Encoding Bekerja Dengan Menggantikan Deretan Pixel Yang Memiliki Nilai Sama Dengan Pasangan Nilai Dan Jumlah Kemunculan.

Contoh: AAAAA Direpresentasikan Sebagai A5

Metode Huffman Coding Menggunakan Probabilitas Kemunculan Simbol Untuk Menghasilkan Kode Biner Yang Optimal. Tabel Berikut Menunjukkan Contoh Huffman Coding.

Tabel 9.1 Representasi Huffman Coding

Simbol	Probabilitas	Kode
A	0.4	0
B	0.3	10
C	0.2	110
D	0.1	111

Sistem Memberikan Kode Yang Lebih Pendek Untuk Simbol Dengan Probabilitas Tinggi Dan Kode Yang Lebih Panjang Untuk Simbol Dengan Probabilitas Rendah. Pendekatan Ini Mengurangi Jumlah Bit Rata-Rata Yang Diperlukan Untuk Representasi Data.

Contoh Implementasi Python Run-Length Encoding:
Import Numpy As Np

Def Run_Length_Encoding(Data):

```

Encoding = []
Prev = Data[0]
Count = 1

For I In Range(1, Len(Data)):
    If Data[I] == Prev:
        Count += 1
    Else:
        Encoding.Append((Prev, Count))
        Prev = Data[I]
        Count = 1

Encoding.Append((Prev, Count))
Return Encoding

```

```

Data = [1,1,1,1,2,2,3,3,3,3]
Encoded = Run_Length_Encoding(Data)
Print(Encoded)

```

9.1.2 Pemampatan Lossy

Pemampatan Lossy Menghilangkan Sebagian Informasi Citra Untuk Mencapai Rasio Kompresi Yang Lebih Tinggi. Sistem Mempertahankan Informasi Yang Penting Secara Visual Dan Menghilangkan Informasi Yang Tidak Signifikan Secara Psikovisual (Gonzalez & Woods, 2018).

Metode Lossy Meliputi:

1. Transform Coding
2. Discrete Cosine Transform (DCT)
3. JPEG Compression

Discrete Cosine Transform Mengubah Citra Dari Domain Spasial Ke Domain Frekuensi.

Persamaan DCT:

$$F(u, v) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \cos \left[\frac{(2x+1)u\pi}{2N} \right] \cos \left[\frac{(2y+1)v\pi}{2N} \right]$$

Transformasi tersebut memusatkan energi citra pada koefisien frekuensi rendah.

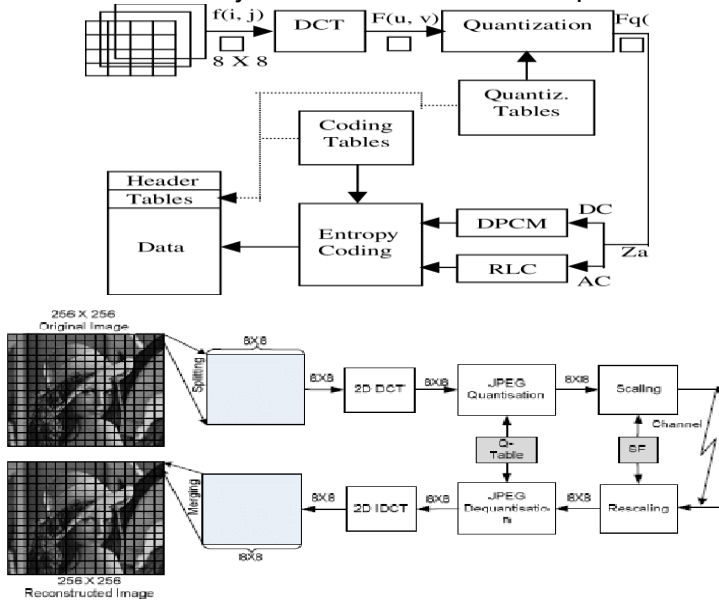
9.1.3 JPEG Compression

JPEG merupakan standar pemampatan citra yang menggunakan DCT dan quantization (Wallace, 1992).

Langkah JPEG meliputi:

1. Konversi Warna
2. Pembagian Blok 8x8
3. Transformasi DCT
4. Quantization
5. Entropy Coding

Gambar berikut menunjukkan proses JPEG compression.



Gambar 9.2 Proses Pemampatan JPEG Menggunakan Transformasi DCT

Quantization Mengurangi Presisi Koefisien Frekuensi Tinggi Sehingga Mengurangi Ukuran Data.

9.1.4 Parameter Evaluasi Pemampatan

Parameter Evaluasi Meliputi:

1. Compression Ratio
2. Mean Squared Error (MSE):

$$MSE = \frac{1}{MN} \sum (f(x, y) - g(x, y))^2$$

3. Peak Signal-To-Noise Ratio (PSNR):

$$PSNR = 10 \log_{10} \frac{MAX^2}{MSE}$$

4. Nilai PSNR Yang Tinggi Menunjukkan Kualitas Citra Yang Lebih Baik (Sayood, 2017).

9.2 Tujuan Praktikum

Setelah Menyelesaikan Praktikum Ini, Mahasiswa Diharapkan Mampu:

1. Memahami Konsep Pemampatan Citra Lossy Dan Lossless
2. Mengimplementasikan Penyimpanan Citra Dengan Berbagai Format Dan Parameter
3. Menganalisis Trade-Off Antara Ukuran File Dan Kualitas Citra
4. Menerapkan Teknik Pemampatan Pada Data Audio-Visual
5. Mengukur Kualitas Citra Terkompresi Dengan Metrik Objektif

9.2.1 Alat Dan Bahan

1. **Perangkat Keras:**
 - Komputer/Laptop
2. **Perangkat Lunak:**
 - Python 3.X
 - Opencv
 - Numpy
 - Matplotlib
 - PIL/Pillow (Opsional)

9.2.2 Langkah Praktikum

Langkah 9.1: Menyimpan Citra Dalam Format PNG Dengan Tingkat Kompresi Berbeda

```
Import Cv2
Import Os
Import Numpy As Np
Import Matplotlib.Pyplot As Plt

# Baca Citra Resolusi Tinggi
Img = Cv2.Imread('Highres_Image.Jpg')
If Img Is None:
    # Buat Citra Sintetis 1000x1000
    Img = Np.Random.Randint(0, 256, (1000,1000,3),
        Dtype=Np.Uint8)

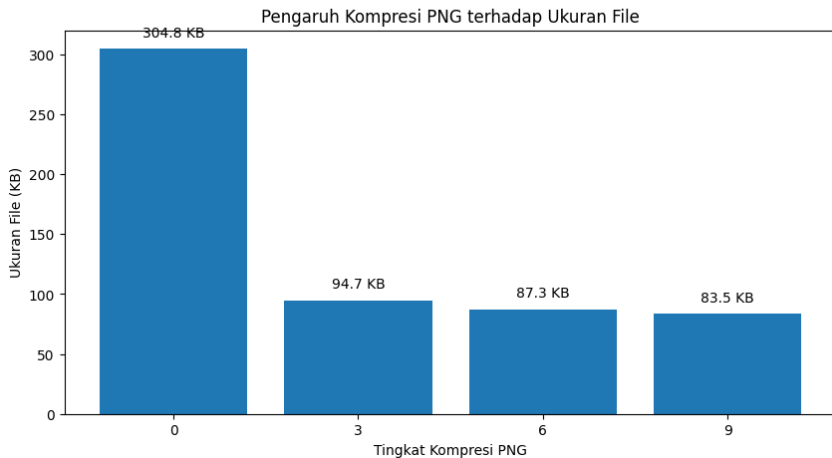
# Simpan PNG Dengan Berbagai Tingkat Kompresi
Compression_Levels = [0, 3, 6, 9]
File_Sizes = []

For Level In Compression_Levels:
    Filename = F'compressed_{Level}.Png'
    Cv2.Imwrite(Filename, Img,
        [Cv2.IMWRITE_PNG_COMPRESSION, Level])
    Size = Os.Path.Getsize(Filename) / 1024 # KB
    File_Sizes.Append(Size)
    Print(F"Level {Level}: {Size:.2f} KB")

Plt.Figure(Figsize=(10,5))
Plt.Bar([Str(L) For L In Compression_Levels], File_Sizes)
Plt.Xlabel('Tingkat Kompresi PNG')
Plt.Ylabel('Ukuran File (KB)')
Plt.Title('Pengaruh Kompresi PNG Terhadap Ukuran File')
For I, V In Enumerate(File_Sizes):
    Plt.Text(I, V+10, F"{V:.1f} KB", Ha='Center')
Plt.Show()
```

Output :

Level 0: 304.79 KB
Level 3: 94.72 KB
Level 6: 87.29 KB
Level 9: 83.54 KB



Langkah 9.2: Menyimpan Citra JPEG Dengan Kualitas Berbeda

```
Qualities = [10, 30, 50, 70, 90, 100]
```

```
Jpeg_Sizes = []
```

```
For Q In Qualities:
```

```
    Filename = F'jpeg_Q{Q}.Jpg'
```

```
    Cv2.Imwrite(Filename, Img, [Cv2.IMWRITE_JPEG_QUALITY,  
    Q])
```

```
    Size = Os.Path.Getsize(Filename) / 1024
```

```
    Jpeg_Sizes.Append(Size)
```

```
    Print(F"Kualitas {Q}: {Size:.2f} KB")
```

```
Plt.Figure(Figsz=(15,5))
```

```
Plt.Subplot(1,2,1)
```

```
Plt.Plot(Qualities, Jpeg_Sizes, Marker='O')
```

```
Plt.Xlabel('Kualitas JPEG')
```

```
Plt.Ylabel('Ukuran File (KB)')
```

```
Plt.Title('Ukuran Vs Kualitas')
```

```
Plt.Grid(True)
```

```
For I, (Q, S) In Enumerate(Zip(Qualities, Jpeg_Sizes)):
```

```
    Plt.Annotate(F'{S:.0f}KB', (Q, S), Textcoords="Offset Points",  
    Xytext=(0,10), Ha='Center')
```

BAB 10 CITRA BINER

A. Deskripsi Mata Kuliah

Bab Ini Membahas Konsep, Metode Dan Implementasi Komputasional Untuk Memperoleh, Memproses, Menganalisis, Dan Menginterpretasi Citra Digital Menggunakan Pendekatan Matematis Dan Algoritmik. Mata Kuliah Ini Memperkenalkan Mahasiswa Pada Representasi Citra, Transformasi Citra, Peningkatan Kualitas Citra, Segmentasi, Ekstraksi Fitur, Serta Analisis Citra Untuk Berbagai Aplikasi Seperti Visi Komputer, Pengenalan Pola, Dan Kecerdasan Buatan. Salah Satu Konsep Fundamental Yang Menjadi Dasar Dalam Analisis Citra Adalah Citra Biner, Yaitu Citra Yang Hanya Memiliki Dua Tingkat Intensitas Dan Digunakan Secara Luas Dalam Proses Segmentasi Dan Analisis Objek.

Mahasiswa Mempelajari Bagaimana Sistem Digital Merepresentasikan Citra Biner Dalam Bentuk Matriks Diskrit Serta Bagaimana Algoritma Melakukan Konversi Dari Citra Grayscale Ke Citra Biner Menggunakan Teknik Thresholding. Mata Kuliah Ini Juga Membahas Operasi Morfologi Seperti Erosi, Dilasi, Opening, Dan Closing Yang Digunakan Untuk Memproses Citra Biner Secara Struktural. Selain Itu, Mahasiswa Mengimplementasikan Berbagai Metode Menggunakan Bahasa Pemrograman Python Dan Pustaka Pengolahan Citra Seperti Opencv Dan Scikit-Image Untuk Memperkuat Pemahaman Konseptual Dan Keterampilan Praktis.

Bab Ini Mendukung Pencapaian Capaian Pembelajaran Lulusan Dalam Bidang Analisis Data Visual, Pengembangan Sistem Berbasis Citra, Serta Pemecahan Masalah Berbasis Komputasi Visual Secara Sistematis Dan Pendekatan Outcome Based Education (OBE).

B. Kompetensi Pembelajaran

1. Mahasiswa Mampu Menjelaskan Konsep Dasar Citra Digital Dan Representasi Citra Biner Secara Matematis Dan Komputasional.
2. Mahasiswa Mampu Menjelaskan Perbedaan Antara Citra Grayscale Dan Citra Biner Serta Menjelaskan Prinsip Thresholding Sebagai Metode Segmentasi Dasar.

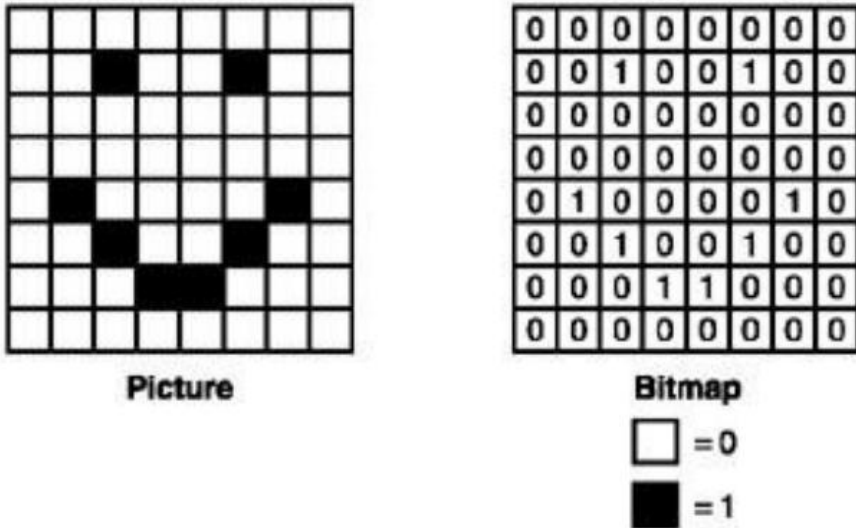
3. Mahasiswa Mampu Menganalisis Hubungan Antara Nilai Intensitas Piksel Dan Representasi Objek Dalam Citra Biner Menggunakan Pendekatan Matriks Diskrit.
4. Mahasiswa Mampu Menerapkan Algoritma Konversi Citra Grayscale Menjadi Citra Biner Menggunakan Metode Threshold Global, Threshold Adaptif, Dan Metode Otsu Menggunakan Bahasa Pemrograman Python.
5. Mahasiswa Mampu Mengimplementasikan Operasi Morfologi Dasar Seperti Erosi, Dilasi, Opening, Dan Closing Untuk Memperbaiki Kualitas Citra Biner Dan Menghilangkan Noise.
6. Mahasiswa Mampu Mengevaluasi Hasil Segmentasi Citra Biner Menggunakan Parameter Kuantitatif Seperti Luas Objek, Perimeter, Dan Jumlah Komponen Terhubung.
7. Mahasiswa Mampu Merancang Solusi Komputasional Untuk Memproses Citra Biner Dalam Berbagai Aplikasi Seperti Deteksi Objek, Pengenalan Karakter, Dan Analisis Bentuk.
8. Mahasiswa Mampu Mengintegrasikan Konsep Teoritis Dan Implementasi Praktis Untuk Memecahkan Masalah Pengolahan Citra Secara Sistematis, Efektif, Dan Sesuai Dengan Prinsip Ilmiah.

1.1 Konsep Dasar Citra Biner

Citra Biner Merupakan Citra Digital Yang Hanya Memiliki Dua Kemungkinan Nilai Intensitas Piksel, Yaitu 0 Dan 1, Atau Secara Umum Direpresentasikan Sebagai 0 Dan 255 Dalam Sistem 8-Bit Untuk Memudahkan Visualisasi Dan Kompatibilitas Sistem Tampilan (Gonzalez & Woods, 2018). Sistem Menggunakan Nilai 0 Untuk Merepresentasikan Latar Belakang (Background) Dan Nilai 1 Atau 255 Untuk Merepresentasikan Objek (Foreground). Representasi Biner Menyederhanakan Struktur Citra Sehingga Sistem Dapat Memproses Citra Dengan Kompleksitas Komputasi Yang Lebih Rendah Dibandingkan Citra Grayscale Atau Citra Berwarna (Jähne, 2005). Sistem Digital Merepresentasikan Citra Biner Sebagai Matriks Dua Dimensi Dengan Ukuran $M \times N$. Setiap Elemen Matriks Merepresentasikan Satu Piksel Dengan Nilai Diskrit. Persamaan Matematis Citra Biner Dapat Dituliskan Sebagai:

$$B(x,y) = \begin{cases} 1, & \text{Jika Piksel Termasuk Objek} \\ 0, & \text{Jika Piksel Termasuk Latar Belakang} \end{cases}$$

Representasi Ini Memungkinkan Sistem Melakukan Analisis Struktural Seperti Penghitungan Luas Objek, Perimeter, Dan Bentuk Secara Efisien (Sonka Et Al., 2014).



Gambar 10.1 Representasi Citra Biner Dari Citra Grayscale.

Gambar diatas Menunjukkan Proses Konversi Citra Grayscale Menjadi Citra Biner. Sistem Menggunakan Ambang Batas Tertentu Untuk Mengklasifikasikan Piksel Menjadi Objek Dan Latar Belakang. Proses Ini Disebut Segmentasi Citra Dan Menjadi Langkah Awal Dalam Analisis Citra Berbasis Objek (Russ, 2016).

Sistem Menggunakan Citra Biner Secara Luas Dalam Berbagai Aplikasi Seperti Pengenalan Karakter Optik (OCR), Deteksi Objek Medis, Analisis Bentuk, Dan Visi Robotik Karena Struktur Data Yang Sederhana Memungkinkan Algoritma Melakukan Analisis Secara Cepat Dan Efisien (Shapiro & Stockman, 2001).

1.1.1 Proses Konversi Citra Grayscale Menjadi Citra Biner

Sistem Melakukan Konversi Citra Grayscale Menjadi Citra Biner Menggunakan Teknik Thresholding. Thresholding Merupakan

Metode Segmentasi Yang Membandingkan Nilai Intensitas Piksel Dengan Nilai Ambang Batas Tertentu (Threshold) (Gonzalez & Woods, 2018).

Persamaan Thresholding Dapat Dituliskan Sebagai Berikut:

$$B(x, y) = \begin{cases} 1, & \text{Jika } I(x, y) \geq T \\ 0, & \text{Jika } I(x, y) < T \end{cases}$$

Keterangan:

$I(X, Y)$ = Nilai Intensitas Piksel Grayscale

T = Nilai Threshold

$B(X, Y)$ = Nilai Piksel Biner

Sistem Menggunakan Threshold Global Ketika Seluruh Citra Menggunakan Satu Nilai Threshold. Sistem Menggunakan Threshold Adaptif Ketika Nilai Threshold Berbeda Untuk Setiap Wilayah Citra (Sezgin & Sankur, 2004). Metode Otsu Menentukan Threshold Optimal Secara Otomatis Dengan Memaksimalkan Variansi Antar Kelas. Metode Ini Bekerja Dengan Meminimalkan Variansi Dalam Kelas Dan Memaksimalkan Variansi Antar Kelas (Otsu, 1979).

1.1.2 Representasi Matriks Citra Biner

Sistem Menyimpan Citra Biner Dalam Bentuk Matriks Diskrit. Contoh Representasi Matriks Citra Biner Ditunjukkan Pada Tabel 10.1.

Tabel 10.1 Representasi Matriks Citra Biner

Y\X	X=0	X=1	X=2	X=3	X=4
Y=0	0	0	0	0	0
Y=1	0	1	1	1	0
Y=2	0	1	1	1	0
Y=3	0	1	1	1	0
Y=4	0	0	0	0	0

Tabel 10.1 Menunjukkan Bahwa Piksel Dengan Nilai 1 Merepresentasikan Objek, Sedangkan Piksel Dengan Nilai 0 Merepresentasikan Latar Belakang. Sistem Menggunakan

Representasi Ini Untuk Melakukan Analisis Struktural Seperti Penghitungan Luas Objek Menggunakan Persamaan:

$$Area = \sum_{x,y} B(x,y)$$

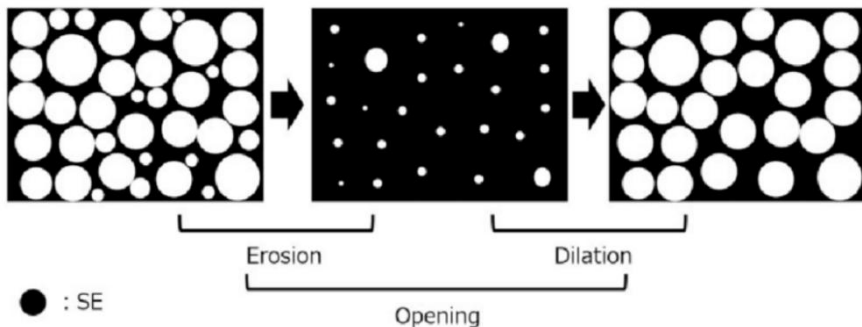
Persamaan Tersebut Menghitung Jumlah Pixel Objek Dalam Citra (Sonka Et Al., 2014).

1.1.3 Operasi Morfologi Pada Citra Biner

Operasi Morfologi Merupakan Teknik Pemrosesan Citra Yang Memodifikasi Struktur Objek Berdasarkan Bentuk (Structuring Element) (Serra, 1982).

Operasi Morfologi Utama Meliputi:

1. Erosi : Sistem Menghapus Pixel Objek Pada Batas Objek.
2. Dilasi : Sistem Menambahkan Pixel Pada Batas Objek.
3. Opening : Sistem Melakukan Erosi Diikuti Dilasi Untuk Menghilangkan Noise Kecil.
4. Closing : Sistem Melakukan Dilasi Diikuti Erosi Untuk Menutup Lubang Kecil.



Gambar 10.2 Operasi Morfologi Pada Citra Biner.

Gambar diatas Menunjukkan Bahwa Operasi Morfologi Memperbaiki Kualitas Citra Biner Dengan Menghilangkan Noise Dan Memperjelas Struktur Objek (Serra, 1982).

1.1.4 Analisis Komponen Terhubung

Sistem Menggunakan Analisis Komponen Terhubung Untuk Mengidentifikasi Objek Dalam Citra Biner. Sistem Mengelompokkan

BAB 11 WARNA

A. Deskripsi Mata Kuliah

Bab Ini Membahas Konsep, Prinsip, Dan Teknik Komputasional Untuk Merepresentasikan, Memproses, Dan Menganalisis Citra Digital Berbasis Intensitas Dan Warna. Sistem Pengolahan Citra Menggunakan Informasi Warna Untuk Meningkatkan Kemampuan Analisis Visual, Segmentasi Objek, Klasifikasi Citra, Dan Interpretasi Visual Secara Komputasional. Warna Menjadi Komponen Penting Karena Sistem Penglihatan Manusia Menggunakan Spektrum Warna Untuk Membedakan Objek, Tekstur, Dan Konteks Visual Secara Efektif.

Mahasiswa Mempelajari Representasi Warna Dalam Sistem Digital Menggunakan Model Warna Seperti RGB, HSV, Dan Grayscale. Sistem Komputer Merepresentasikan Warna Sebagai Kombinasi Nilai Numerik Diskrit Yang Membentuk Piksel Dalam Ruang Warna Tertentu. Mahasiswa Mempelajari Transformasi Ruang Warna, Konversi Antar Model Warna, Dan Penggunaan Ruang Warna Dalam Segmentasi Dan Analisis Citra. Sistem Menggunakan Model Warna Tertentu Untuk Meningkatkan Akurasi Analisis Objek Dan Mengurangi Sensitivitas Terhadap Perubahan Pencahayaan. Mahasiswa Juga Mengimplementasikan Algoritma Pengolahan Citra Warna Menggunakan Bahasa Pemrograman Python Dan Pustaka Opencv. Mahasiswa Memahami Hubungan Antara Representasi Matematis Warna Dan Implementasi Komputasional. Bab Ini Mendukung Pencapaian Capaian Pembelajaran Lulusan Dalam Bidang Analisis Citra Digital, Visi Komputer, Dan Kecerdasan Buatan Serta Outcome Based Education (OBE).

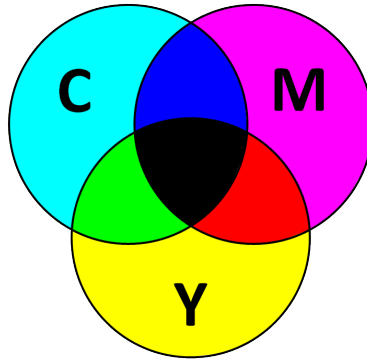
B. Kompetensi Pembelajaran

1. Mahasiswa Mampu Menjelaskan Konsep Dasar Warna Dalam Sistem Pengolahan Citra Digital Menggunakan Pendekatan Matematis Dan Komputasional.
2. Mahasiswa Mampu Menjelaskan Bagaimana Sistem Digital Merepresentasikan Warna Menggunakan Model RGB, HSV, Dan Model Warna Lainnya.

3. Mahasiswa Mampu Menjelaskan Hubungan Antara Sistem Penglihatan Manusia Dan Representasi Warna Digital Secara Ilmiah.
4. Mahasiswa Mampu Mengimplementasikan Representasi Warna Menggunakan Matriks Tiga Dimensi Dalam Bahasa Pemrograman Python.
5. Mahasiswa Mampu Melakukan Konversi Ruang Warna Menggunakan Algoritma Komputasional Dan Pustaka Opencv.
6. Mahasiswa Mampu Menganalisis Pengaruh Perubahan Ruang Warna Terhadap Hasil Segmentasi Dan Analisis Citra.
7. Mahasiswa Mampu Menggunakan Model Warna HSV Untuk Segmentasi Objek Berbasis Warna Secara Akurat.
8. Mahasiswa Mampu Mengevaluasi Hasil Pengolahan Citra Warna Menggunakan Parameter Numerik Dan Visual.
9. Mahasiswa Mampu Merancang Solusi Komputasional Berbasis Warna Untuk Aplikasi Visi Komputer Seperti Deteksi Objek, Pengenalan Pola, Dan Klasifikasi Citra.

11.1 Konsep Dasar Warna Dalam Citra Digital

Sistem Pengolahan Citra Digital Menggunakan Warna Sebagai Representasi Visual Dari Spektrum Elektromagnetik Yang Dapat Dideteksi Oleh Sistem Penglihatan Manusia. Sistem Penglihatan Manusia Menggunakan Sel Fotoreseptor Yang Disebut Cone Untuk Mendeteksi Panjang Gelombang Cahaya Yang Berbeda. Sistem Menggunakan Tiga Jenis Cone Untuk Mendeteksi Warna Merah, Hijau, Dan Biru (Russ, 2016). Sistem Digital Merepresentasikan Warna Menggunakan Model Matematis Yang Disebut Model Warna. Model Warna Merepresentasikan Warna Sebagai Kombinasi Numerik Dari Komponen Dasar Warna. Model Warna Memungkinkan Sistem Komputer Memproses Warna Secara Matematis Dan Komputasional (Gonzalez & Woods, 2018).



Gambar 11.1 Representasi Warna Menggunakan Model RGB.

Gambar diatas Menunjukkan Bahwa Sistem Digital Menggunakan Tiga Komponen Warna Dasar Yaitu Red (R), Green (G), Dan Blue (B). Sistem Merepresentasikan Setiap Piksel Sebagai Kombinasi Tiga Nilai Numerik. Setiap Komponen Memiliki Rentang Nilai 0 Sampai 255 Dalam Sistem 8-Bit (Gonzalez & Woods, 2018).

Persamaan Matematis Representasi Warna Dapat Dituliskan Sebagai:

$$C(x, y) = [R(x, y), G(x, y), B(x, y)]$$

Sistem Menggunakan Persamaan Tersebut Untuk Menyimpan Nilai Warna Setiap Piksel Dalam Matriks Tiga Dimensi.

11.1.1 Representasi Matriks Warna Digital

Sistem Merepresentasikan Citra Warna Sebagai Matriks Tiga Dimensi Dengan Ukuran $M \times N \times 3$. Dimensi Pertama Menunjukkan Baris, Dimensi Kedua Menunjukkan Kolom, Dan Dimensi Ketiga Menunjukkan Kanal Warna (Sonka Et Al., 2014).

Tabel 11.1 Representasi Matriks Citra Warna RGB

Y\X	X=0	X=1	X=2
Y=0	(255,0,0)	(0,255,0)	(0,0,255)
Y=1	(255,255,0)	(0,255,255)	(255,0,255)
Y=2	(0,0,0)	(128,128,128)	(255,255,255)

Tabel diatas Menunjukkan Bahwa Setiap Piksel Memiliki Tiga Komponen Warna. Sistem Menggunakan Nilai (255,0,0) Untuk

Merepresentasikan Warna Merah Penuh. Sistem Menggunakan Nilai (0,255,0) Untuk Merepresentasikan Warna Hijau Penuh. Sistem Menggunakan Nilai (0,0,255) Untuk Merepresentasikan Warna Biru Penuh.

Sistem Menyimpan Matriks Ini Dalam Memori Menggunakan Struktur Array Tiga Dimensi. Sistem Menggunakan Indeks Matriks:

$$C[y][x][channel]$$

Channel Memiliki Nilai:

0 = Red

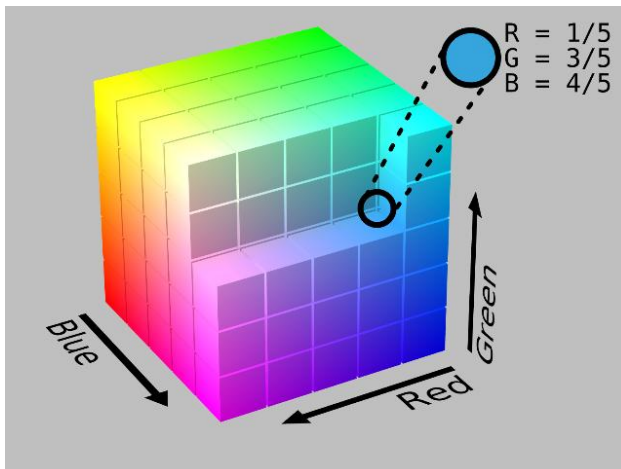
1 = Green

2 = Blue

Representasi Ini Memungkinkan Sistem Memproses Setiap Kanal Warna Secara Independen (Sonka Et Al., 2014).

11.1.2 Model Warna RGB

Model Warna RGB Merupakan Model Warna Aditif Yang Menggunakan Kombinasi Warna Merah, Hijau, Dan Biru Untuk Menghasilkan Warna Lain. Sistem Menghasilkan Warna Putih Ketika Nilai Ketiga Kanal Maksimum. Sistem Menghasilkan Warna Hitam Ketika Nilai Ketiga Kanal Minimum (Gonzalez & Woods, 2018).



Gambar 11.2 Model Warna RGB Dan Kombinasi Warna Aditif.

Gambar diatas Menunjukkan Bahwa Sistem Menghasilkan Warna Baru Dengan Mengkombinasikan Tiga Kanal Warna.

Contoh:

(255,255,255) = Putih

(0,0,0) = Hitam

(255,0,0) = Merah

Model RGB Digunakan Secara Luas Dalam Kamera Digital, Monitor, Dan Sistem Visi Komputer.

11.1.3 Model Warna HSV

Model Warna HSV Merepresentasikan Warna Menggunakan Tiga Komponen:

Hue = Jenis Warna

Saturation = Intensitas Warna

Value = Kecerahan Warna

Model HSV Lebih Sesuai Untuk Analisis Warna Karena Memisahkan Informasi Warna Dari Intensitas Cahaya (Russ, 2016).



Gambar 11.3 Model Warna HSV.

Model HSV Memungkinkan Sistem Melakukan Segmentasi Warna Dengan Lebih Akurat Dibandingkan RGB (Gonzalez & Woods, 2018).

11.1.4 Transformasi Ruang Warna

Sistem Melakukan Transformasi Ruang Warna Untuk Meningkatkan Analisis Citra. Sistem Mengubah Citra Dari RGB Ke HSV Menggunakan Transformasi Matematis (Sonka Et Al., 2014). Transformasi Ini Memungkinkan Sistem Memisahkan Informasi Warna Dari Pencahayaan.

11.1.5 Implementasi Python Untuk Pengolahan Warna

Mahasiswa Dapat Menggunakan Python Untuk Memproses Citra Warna.

Contoh Kode Python:

```
Import Cv2
```

```
Import Matplotlib.Pyplot As Plt
```

```
Img = Cv2.Imread('Gambar.Jpg')
```

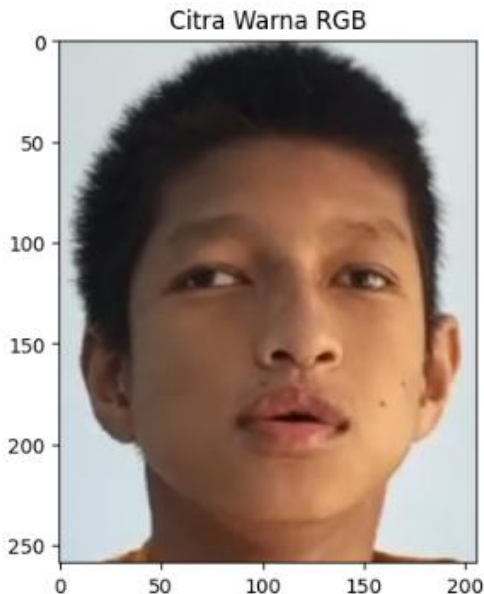
```
Img_Rgb = Cv2.CvtColor(Img, Cv2.COLOR_BGR2RGB)
```

```
Plt.Imshow(Img_Rgb)
```

```
Plt.Title("Citra Warna RGB")
```

```
Plt.Show()
```

Output :



11.2 Tujuan Praktikum

Setelah Menyelesaikan Praktikum Ini, Mahasiswa Diharapkan Mampu:

1. Memahami Berbagai Model Warna (RGB, HSV, Ycbcr, Lab)
2. Mengkonversi Antar Model Warna Menggunakan Opencv
3. Menerapkan Segmentasi Berbasis Warna Untuk Memisahkan Bibir
4. Mengekstraksi Fitur Warna Untuk Pengenalan Pola
5. Mengatasi Variasi Pencahayaan Dengan Ruang Warna Yang Sesuai

11.2.1 Alat Dan Bahan

1. **Perangkat Keras:**
 - Komputer/Laptop
2. **Perangkat Lunak:**
 - Python 3.X
 - Opencv
 - Numpy
 - Matplotlib

11.2.2 Langkah Praktikum

Langkah 11.1: Konversi Antar Model Warna Dan Visualisasi Channel

```
Import Cv2
Import Numpy As Np
Import Matplotlib.Pyplot As Plt
```

```
# Baca Citra
```

```
Img = Cv2.Imread('Contoh.Jpg')
```

```
If Img Is None:
```

```
    Img = Np.Random.Randint(0, 256, (200,300,3),
        Dtype=Np.Uint8)
```

```
# Konversi Ke Berbagai Ruang Warna
```

```
Img_Rgb = Cv2.CvtColor(Img, Cv2.COLOR_BGR2RGB)
```

```
Img_Hsv = Cv2.CvtColor(Img, Cv2.COLOR_BGR2HSV)
```

```
Img_Ycrcb = Cv2.CvtColor(Img, Cv2.COLOR_BGR2YCrCb)
```

BAB 13 PEMAMPATAN CITRA FRAKTAL

A. Deskripsi Mata Kuliah

Bab Ini Membahas Konsep, Metode, Dan Algoritma Untuk Merepresentasikan, Memproses, Dan Mengompresi Citra Digital Menggunakan Pendekatan Matematis Dan Komputasional. Sistem Pengolahan Citra Menggunakan Teknik Pemampatan Untuk Mengurangi Ukuran Data Citra Tanpa Menghilangkan Informasi Visual Secara Signifikan. Salah Satu Teknik Pemampatan Yang Penting Adalah Pemampatan Citra Fraktal, Yaitu Metode Pemampatan Yang Memanfaatkan Sifat Kemiripan Diri (Self-Similarity) Dalam Citra Digital Untuk Merepresentasikan Citra Menggunakan Transformasi Matematis Fraktal. Sistem Menggunakan Pemampatan Citra Fraktal Untuk Menghasilkan Rasio Kompresi Tinggi Dengan Mempertahankan Kualitas Visual Citra. Sistem Memodelkan Citra Menggunakan Transformasi Iteratif Yang Disebut Iterated Function System (IFS). Metode Ini Memungkinkan Sistem Merekonstruksi Citra Menggunakan Parameter Matematis Tanpa Menyimpan Semua Pikel Secara Langsung. Mahasiswa Mempelajari Konsep Dasar Fraktal, Representasi Matematis Fraktal, Algoritma Encoding Dan Decoding Fraktal, Serta Implementasi Komputasional Menggunakan Python. Mahasiswa Memahami Hubungan Antara Teori Matematika Fraktal Dan Implementasi Komputasional Dalam Sistem Pengolahan Citra Digital. Mata Kuliah Ini Mendukung Capaian Pembelajaran Lulusan Dalam Bidang Pengolahan Citra, Kompresi Data, Dan Pengembangan Sistem Visi Komputer Sesuai Dengan Standar SN-Dikti Dan Pendekatan Outcome Based Education (OBE).

B. Kompetensi Pembelajaran

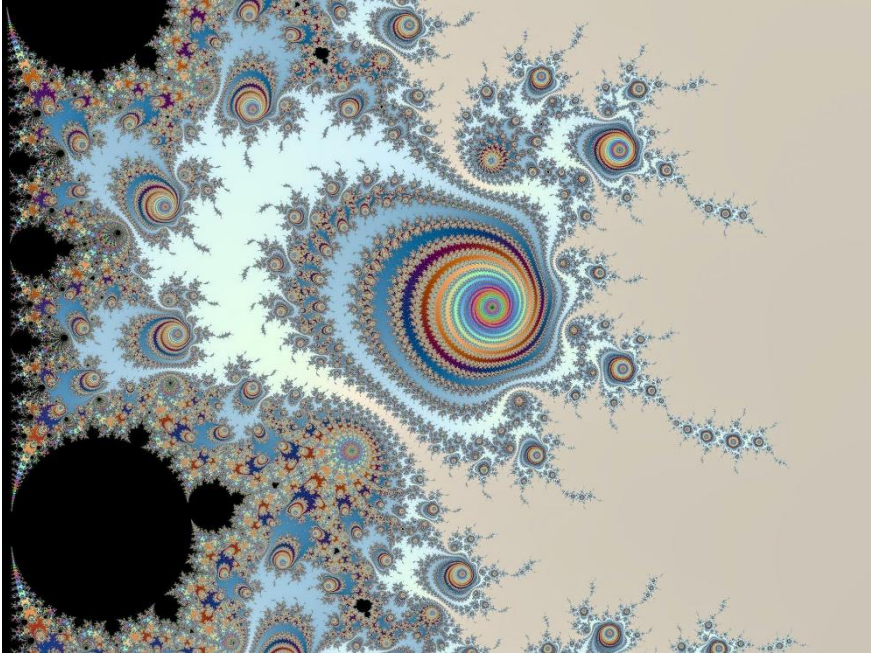
1. Mahasiswa Mampu Menjelaskan Konsep Dasar Pemampatan Citra Digital Dan Prinsip Pemampatan Citra Fraktal Menggunakan Pendekatan Matematis Dan Komputasional.
2. Mahasiswa Mampu Menjelaskan Konsep Self-Similarity Dan Bagaimana Sistem Menggunakan Konsep Tersebut Untuk Merepresentasikan Citra Digital Secara Efisien.
3. Mahasiswa Mampu Menjelaskan Prinsip Iterated Function System (IFS) Dalam Pemampatan Citra Fraktal.

4. Mahasiswa Mampu Menjelaskan Proses Encoding Dan Decoding Citra Fraktal Secara Sistematis.
5. Mahasiswa Mampu Menganalisis Perbedaan Antara Metode Pemampatan Fraktal Dan Metode Pemampatan Tradisional Seperti JPEG.
6. Mahasiswa Mampu Mengevaluasi Kualitas Citra Hasil Pemampatan Menggunakan Parameter Peak Signal-To-Noise Ratio (PSNR) Dan Mean Squared Error (MSE).
7. Mahasiswa Mampu Mengimplementasikan Algoritma Pemampatan Citra Fraktal Sederhana Menggunakan Bahasa Pemrograman Python.
8. Mahasiswa Mampu Menganalisis Efisiensi Kompresi Dan Kualitas Rekonstruksi Citra.
9. Mahasiswa Mampu Merancang Solusi Pemampatan Citra Berbasis Fraktal Untuk Aplikasi Pengolahan Citra Digital Secara Sistematis Dan Ilmiah.

13.1 Konsep Dasar Fraktal Dalam Citra Digital

Fraktal Merupakan Struktur Matematis Yang Memiliki Sifat Kemiripan Diri (Self-Similarity), Yaitu Bagian Dari Struktur Memiliki Bentuk Yang Mirip Dengan Keseluruhan Struktur. Sistem Menggunakan Konsep Fraktal Untuk Merepresentasikan Citra Digital Secara Efisien Karena Banyak Citra Alami Memiliki Pola Yang Berulang Pada Skala Yang Berbeda (Fisher, 1995).

Sistem Menggunakan Fraktal Untuk Memodelkan Citra Sebagai Kumpulan Transformasi Matematis Yang Dapat Merekonstruksi Citra Secara Iteratif. Sistem Tidak Menyimpan Nilai Pixel Secara Langsung, Tetapi Sistem Menyimpan Parameter Transformasi Yang Menghasilkan Citra Tersebut (Salomon, 2007).



Gambar 13.1 Struktur Fraktal Dan Self-Similarity Dalam Citra Digital.

Gambar Menunjukkan Bahwa Bagian Kecil Citra Memiliki Kemiripan Dengan Bagian Lain. Sistem Menggunakan Kemiripan Ini Untuk Mengurangi Jumlah Data Yang Harus Disimpan.

13.1.1 Prinsip Pemampatan Citra Fraktal

Pemampatan Citra Fraktal Menggunakan Konsep Iterated Function System (IFS). Sistem Menggunakan Transformasi Geometris Untuk Merepresentasikan Citra Digital (Barnsley, 1988).

Transformasi Tersebut Meliputi:

- Translasi
- Rotasi
- Skala
- Refleksi

Sistem Menggunakan Transformasi Affine Untuk Memetakan Bagian Citra Ke Bagian Lain.

Persamaan Transformasi Affine:

$$w(x, y) = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix}$$

Transformasi Ini Memungkinkan Sistem Memetakan Blok Citra Ke Blok Lain (Fisher, 1995).

13.1.2 Domain Block Dan Range Block

Sistem Membagi Citra Menjadi Dua Jenis Blok:

- Range Block
- Domain Block

Range Block Merupakan Blok Kecil Yang Akan Dikompresi. Domain Block Merupakan Blok Yang Digunakan Sebagai Referensi.

Tabel 13.1 Representasi Domain Block Dan Range Block

Range Block	Domain Block	Transformasi
Blok A	Blok C	Rotasi 90°
Blok B	Blok D	Skala 0.5

Istem Mencari Domain Block Yang Paling Mirip Dengan Range Block Menggunakan Transformasi Tertentu.

13.1.3 Proses Encoding Citra Fraktal

Proses Encoding Melibatkan Langkah Berikut:

- Sistem Membagi Citra Menjadi Range Block
- Sistem Mencari Domain Block Yang Mirip
- Sistem Menghitung Parameter Transformasi
- Sistem Menyimpan Parameter Transformasi

Parameter Ini Digunakan Untuk Merekonstruksi Citra (Salomon, 2007).

13.1.4 Proses Decoding Citra Fraktal

Sistem Merekonstruksi Citra Menggunakan Parameter Transformasi. Sistem Menggunakan Proses Iteratif Untuk Menghasilkan Citra Akhir (Fisher, 1995).

Proses Decoding Lebih Cepat Dibanding Encoding Karena Sistem Hanya Menerapkan Transformasi.

13.1.5 Evaluasi Kualitas Pemampatan

Sistem Menggunakan MSE Dan PSNR Untuk Mengevaluasi Kualitas Citra.

Persamaan MSE:

$$MSE = \frac{1}{MN} \sum (I - K)^2$$

Persamaan PSNR:

$$PSNR = 10 \text{Log}_{10} \left(\frac{255^2}{MSE} \right)$$

PSNR Tinggi Menunjukkan Kualitas Citra Lebih Baik (Gonzalez & Woods, 2018).

13.2 Tujuan Praktikum

Setelah Menyelesaikan Praktikum Ini, Mahasiswa Diharapkan Mampu:

1. Memahami Konsep Dasar Pemampatan Fraktal
2. Mengimplementasikan Algoritma Sederhana Pemampatan Fraktal
3. Menganalisis Rasio Kompresi Dan Kualitas Rekonstruksi
4. Membandingkan Dengan Metode Kompresi Konvensional

13.2.1 Alat Dan Bahan

1. Perangkat Lunak:

- o Python 3.X
- o Numpy
- o Matplotlib
- o Opencv

13.2.2 Langkah Praktikum

Langkah 13.1: Membagi Citra Menjadi Blok Range Dan Domain

```
Import Numpy As Np
```

```
Import Matplotlib.Pyplot As Plt
```

BAB 14 PENGENALAN POLA

A. Deskripsi Mata Kuliah

Mata Kuliah Pengelolaan Citra Digital Membahas Teknik Analisis Dan Interpretasi Citra Menggunakan Pendekatan Matematis Dan Komputasional. Sistem Menggunakan Pengenalan Pola Sebagai Metode Utama Untuk Mengidentifikasi, Mengklasifikasikan, Dan Menginterpretasi Objek Dalam Citra Digital Secara Otomatis. Pengenalan Pola Memungkinkan Sistem Mengenali Struktur Data Berdasarkan Karakteristik Tertentu Yang Disebut Fitur. Sistem Menggunakan Fitur Tersebut Untuk Membedakan Objek Yang Berbeda Secara Komputasional Dan Sistematis. Mahasiswa Mempelajari Konsep Dasar Pengenalan Pola, Proses Ekstraksi Fitur, Seleksi Fitur, Dan Klasifikasi Menggunakan Algoritma Statistik Dan Pembelajaran Mesin. Sistem Menggunakan Metode Pengenalan Pola Dalam Berbagai Aplikasi Seperti Pengenalan Wajah, Pengenalan Karakter Optik (OCR), Sistem Biometrik, Dan Visi Komputer (Jain Et Al., 2000). Mahasiswa Memahami Hubungan Antara Representasi Citra Digital Dan Analisis Fitur Untuk Menghasilkan Keputusan Klasifikasi Yang Akurat. Mahasiswa Juga Mengimplementasikan Algoritma Pengenalan Pola Menggunakan Bahasa Pemrograman Python Dan Pustaka Seperti Numpy Dan Opencv. Bab Ini Mendukung Capaian Pembelajaran Lulusan Dalam Bidang Analisis Data Visual, Kecerdasan Buatan Dan Outcome Based Education (OBE).

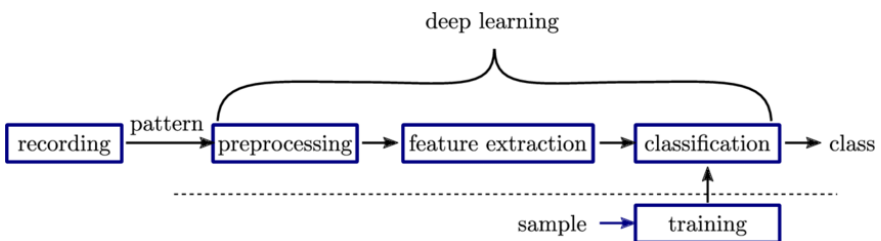
B. Kompetensi Pembelajaran

1. Mahasiswa Mampu Menjelaskan Konsep Dasar Pengenalan Pola Dalam Sistem Pengolahan Citra Digital Menggunakan Pendekatan Matematis Dan Komputasional.
2. Mahasiswa Mampu Menjelaskan Hubungan Antara Citra Digital, Fitur, Dan Proses Klasifikasi.
3. Mahasiswa Mampu Menjelaskan Peran Ekstraksi Fitur Dalam Meningkatkan Akurasi Pengenalan Pola.
4. Mahasiswa Mampu Mengimplementasikan Metode Ekstraksi Fitur Menggunakan Bahasa Pemrograman Python.

5. Mahasiswa Mampu Menggunakan Algoritma Klasifikasi Seperti Nearest Neighbor Dan Metode Berbasis Statistik Untuk Mengklasifikasikan Objek Dalam Citra.
6. Mahasiswa Mampu Menganalisis Performa Sistem Pengenalan Pola Menggunakan Parameter Akurasi, Presisi, Dan Recall.
7. Mahasiswa Mampu Merancang Sistem Pengenalan Pola Untuk Aplikasi Pengolahan Citra Digital.
8. Mahasiswa Mampu Mengevaluasi Performa Sistem Menggunakan Metode Validasi.
9. Mahasiswa Mampu Mengintegrasikan Konsep Teoritis Dan Implementasi Praktis Untuk Mengembangkan Sistem Pengenalan Pola Secara Sistematis Dan Ilmiah.

13 Konsep Dasar Pengenalan Pola

Pengenalan Pola Merupakan Proses Identifikasi Struktur Atau Pola Dalam Data Menggunakan Pendekatan Matematis Dan Komputasional. Sistem Menggunakan Pengenalan Pola Untuk Mengklasifikasikan Objek Berdasarkan Karakteristik Tertentu Yang Disebut Fitur (Duda Et Al., 2001). Sistem Menggunakan Fitur Sebagai Representasi Numerik Dari Objek Dalam Citra Digital. Sistem Pengenalan Pola Terdiri Dari Beberapa Tahap Utama, Yaitu Akuisisi Data, Ekstraksi Fitur, Dan Klasifikasi. Sistem Memperoleh Citra Digital Sebagai Input. Sistem Mengekstrak Fitur Dari Citra. Sistem Menggunakan Fitur Tersebut Untuk Menentukan Kelas Objek.



Gambar 14.1 Tahapan Sistem Pengenalan Pola Dalam Pengolahan Citra Digital(Aubreville, Marc. (2020)).

Gambar diatas Menunjukkan Bahwa Sistem Memproses Citra Melalui Tahap Ekstraksi Fitur Dan Klasifikasi. Sistem Menghasilkan Keputusan Klasifikasi Berdasarkan Fitur Yang Diekstrak.

14.1.1 Representasi Fitur Dalam Citra Digital

Fitur Merupakan Karakteristik Numerik Yang Merepresentasikan Objek Dalam Citra Digital. Sistem Menggunakan Fitur Untuk Membedakan Objek Yang Berbeda. Sistem Mengekstrak Fitur Seperti:

- Bentuk
- Tekstur
- Warna
- Ukuran

Sistem Merepresentasikan Fitur Sebagai Vektor Numerik (Jain Et Al., 2000).

Persamaan Vektor Fitur:

$$F = [f_1, f_2, f_3, \dots, f_n]$$

Sistem Menggunakan Vektor Fitur Untuk Klasifikasi.

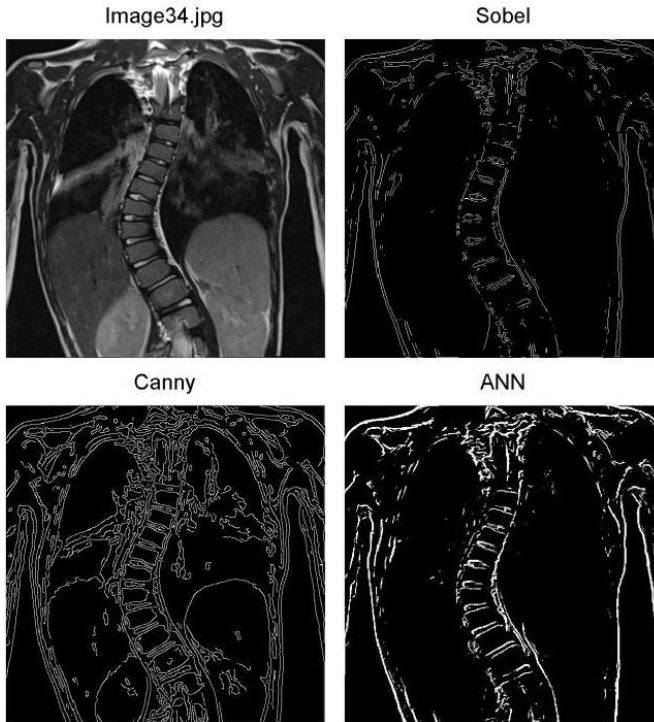
Tabel 14.1 Contoh Representasi Vektor Fitur

Objek	Luas	Perimeter	Intensitas
A	120	40	150
B	200	60	180

Tabel Tersebut Menunjukkan Bahwa Sistem Merepresentasikan Objek Sebagai Data Numerik.

14.1.2 Ekstraksi Fitur

Ekstraksi Fitur Merupakan Proses Mengubah Citra Digital Menjadi Representasi Numerik Yang Dapat Diproses Oleh Sistem Klasifikasi (Sonka Et Al., 2014). Sistem Menggunakan Teknik Ekstraksi Fitur Untuk Mengurangi Dimensi Data Dan Meningkatkan Efisiensi Komputasi.



Gambar 14.2 Ekstraksi Fitur Pada Citra Digital (Gunawan, Et All, 2017).

Sistem Menggunakan Fitur Seperti Tepi, Tekstur, Dan Bentuk Untuk Merepresentasikan Objek.

14.1.3 Klasifikasi Pola

Klasifikasi Merupakan Proses Menentukan Kelas Objek Berdasarkan Fitur. Sistem Menggunakan Algoritma Klasifikasi Seperti:

- Nearest Neighbor
- Bayesian Classifier
- Neural Network

Nearest Neighbor Menggunakan Jarak Euclidean:

$$d = \sqrt{\sum(x_i - y_i)^2}$$

PROFIL PENULIS



Muhaimin Hasanudin, S.T., M.Kom.

Penulis Lahir Di Tangerang - Banten Telah Lulus Strata-1 Pada Program Studi Teknik Informatika Di Sekolah Sains Dan Teknologi Bandung (St.Inten), Menyelesaikan Strata-2 Pada Program Studi Magister Ilmu Komputer Di Universitas Budi Luhur Jakarta Dan Lanjut Studi Doktoral Ilmu Komputer UGM Disemester Genap 2023 By Research Dengan Konsentasi Speech Recognition. Semenjak Bulan September 2018, Dosen Menjadi Pekerjaan Utama Dimana Sebelumnya Bekerja Sebagai Praktisi Konsultan Komputer Dengan Jabatan Terakhir Sebagai Project Manager – IT Risk Bank Danamon. Saat Ini Menjadi Dosen Di Prodi Informatika, Fakultas Ilmu Komputer, Universitas Mercu Buana Jakarta. Selain Mengajar Di Kampus, Penulis Juga Melakukan Penelitian Dan Pengabdian Kepada Masyarakat Baik Internal Kampus Dan Eksternal Dikti. Beberapa Hasil Penelitian Telah Di Publikasikan Di Jurnal Dan Seminar Baik Nasional Dan Internasional Terindex Scopus. Selain Publikasi Ilmiah Penulis Memuat Artikel Di Media Cetak Baik Online Maupun Offline Dan Buku Monograf. Penulis Dapat Dihubungi Di Alamat Email : muhammad.hasanudin@mercubuana.ac.id.



Dedy Prasetya Kristiadi, M.Kom.

Menyelesaikan Pendidikan S1 Di Universitas Budi Luhur Jakarta Dan S2 Di Universitas Nusa Mandiri. Saat Ini Menjadi Ketua Lembaga Penelitian Dan Pengabdian Masyarakat Pada STMIK Kuwera Dan Menjadi Managing Editor Pada Jurnal Sintek Kuwera

(<https://www.sintek.stmikku.ac.id/index.php/home/editor>)

Kegiatan Lainnya Adalah Sebagai Dosen Di Kampus Kuwera Dan Menulis Jurnal Nasional Maupun Internasional Terindex Scopus. Penelitian Yang Telah Dilakukan Saat Ini Meliputi Rekam Medis Elektronik (2018) Dan Penerapan Pengolahan Citra Untuk Menganalisis Respon Emosional Dalam Video Untuk Meningkatkan Efektivitas Iklan (2025). Selain Itu, Terdapat Pula Beberapa Buku Ajar Dan Buku Referensi Seperti Yang Telah Ditulis Dan Di Publish Sebagai Bentuk Pelaksanaan Tridharma Perguruan Tinggi. Semoga Kehadiran Buku Ini Dapat Bermanfaat Bagi Banyak Orang.

Email: Dedyprasetya.kuwera@gmail.com.



Deni Kuswoyo, M.Kom.

Penulis lahir di Palembang, bulan Maret 1988. Penulis menamatkan pendidikan dasar dan menengah di Jakarta, setelah lulus dari SMA Negeri 57 Jakarta melanjutkan kuliah S1 dan S2 di Universitas Budi Luhur Jakarta

Program Studi Sarjana Sistem Informasi dan Magister Ilmu Komputer. Memulai karir sebagai IT sejak tahun 2010 sebagai Network Engineer di PT. Telkom dan PT. Aero System Infonesia. Saat ini sebagai dosen di Politeknik Negeri Media Kreatif, mengajar pada Program Studi Teknologi Rekayasa Multimedia..



Ridlan Ahmad, M.Kom

Penulis lahir di Seram - Ambon telah lulus Strata-1 pada program studi Teknik Informatika di Sekolah Tinggi Ilmu Komputer Ambon, menyelesaikan Strata-2 pada program studi Universitas Amikom Yogyakarta dan lanjut Studi Doktoral Ilmu Komputer UGM di Semester Genap 2023 dengan konsentrasi Autonomous vehicles. Semenjak bulan Januari 2020, dosen menjadi pekerjaan Utama dimana sebelumnya bekerja sebagai praktisi konsultan komputer dengan jabatan terakhir sebagai FE (Field Enggineer), Menangani Solf Problem ATM Diebold Dan Wincor. Saat ini menjadi Dosen di Prodi Informatika, Fakultas Ilmu Komputer, Institut Teknologi dan Bisnis Stikom Ambon. Selain mengajar di kampus. Penulis dapat dihubungi di email: ahmadridlan@yahoo.co.id.



Prof. Dr. Indrianto. S.kom., M.T.

Penulis menyelesaikan gelar Magister dan Doktoral di Universitas Gunadarma dan lulus pada tahun 2025. Jabatan Fungsionalnya Profesor dibidang Rekayasa Sistem Komuter. Ia juga aktif melakukan beberapa penelitian yang pada tahun 2013 – 2025 dan mendapatkan hibah penelitian dari DIKTI. Saat ini, ia aktif menulis di beberapa jurnal yang terindeks Scopus. Ia menulis 21 jurnal terindeks Scopus dan 106 jurnal lokal dan Terakreditasi Sinta di bidang Teknik Informatika dengan indeks H- 6 dan indeks Google Scholar 10. Beliau juga aktif menulis buku, saat ini sudah ada 9 buku yang sudah ditulis. Saat ini menjabat sebagai dosen dan wakil dekan Fakultas Telematika dan Energi di Institut Teknologi PLN, Jakarta.