

Makalah
Pengolahan Citra Digital

“Rekap Materi”



Dosen Pengampu:

Melri Deswina S.Pd., M.Pd.T

Oleh:

Najwa Alawiyah Siregar (22346040)

PROGRAM STUDI INFORMATIKA
DEPARTEMEN TEKNIK ELEKRONIKA
FAKULTAS TEKNIK
UNIVERSITAS NEGERI PADANG
2024

KATA PENGANTAR

Puji syukur saya panjatkan ke hadirat Allah SWT yang telah melimpahkan rahmat dan karunia-Nya, sehingga saya dapat menyelesaikan tugas ini dengan baik. Tugas ini disusun sebagai salah satu bentuk tanggung jawab akademik dalam merekap materi yang telah diajarkan oleh Ibu Melri Deswina, S.Pd., M.Pd.T., selama proses pembelajaran.

Dalam penyusunan tugas ini, saya berusaha memberikan yang terbaik sesuai dengan pemahaman dan materi yang telah disampaikan. Saya juga menyadari bahwa hasil ini masih jauh dari sempurna, sehingga dengan lapang hati saya menerima kritik dan saran yang membangun untuk perbaikan di masa mendatang.

Semoga tugas ini dapat bermanfaat bagi saya pribadi dalam menambah pemahaman dan juga bagi pihak lain yang mungkin membutuhkan referensi terkait materi yang disampaikan.

Sago, 02 Desember 2024

Hormat Saya

Penulis

DAFTAR ISI

KATA PENGANTAR.....	ii
DAFTAR ISI.....	iii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	1
1.3 Tujuan.....	2
BAB II PEMBAHASAN	3
2.1 Memahami Konsep Pembentukan Citra Dan Mampu Menganalisa Dan Menjelaskan Elemen – Elemen Citra Atau pun Pemrosesan Citra Digital	3
2.1.1 Pembentukan Citra	3
2.1.2 Elemen System Pemrosesan citra Digital	9
2.2 Operasi Pixel Pada Citra Digital	12
2.2.1 Jenis Citra.....	12
2.2.2 Sytem Koordinat Warna	15
2.2.3 Format File Citra	22
2.3 Analisis Operasi Pixel Dan Penerapan Metode Histogram Dalam Pengolahan Citra Digital.....	23
2.3.1 Meningkatkan Kecerahan Pada Operasi Pixel	23
2.4 Operasi Ketetanggaan Pixel.....	26
2.4.1 Pengertian Operasi Ketetanggaan Pixel.....	26
2.4.2 Pengertian Ketetanggaan Pixel	27
2.4.3 Aplikasi Ketetanggaan Pixel pada Filter.....	28
2.4.4 Pengertian Konvolusi	33
2.5 Operasi Aritmatika Citra.....	38
2.5.1 Operasi Aritmatika Citra	38
2.5.2 Operasi Logika Citra	44
2.5.3 Aplikasi Operasi Aritmatika dan Logika pada Citra	47
2.6 Operasi Geometri Citra	48
2.6.1 Pengertian Operasi Geometri pada Citra.....	48
2.6.2 Translasi	49
2.6.3 Rotasi	50

2.6.4 Refleksi	53
2.7 Operasi Morfologi.....	54
2.7.1 Pengertian Operasi Morfologi.....	54
2.7.2 Jenis-Jenis Operasi Morfologi.....	55
2.8 Operasi Pada Citra Biner.....	56
2.8.1 Ekstraksi Tepi Objek	56
2.8.2 Representasi Bentuk.....	61
2.8.3 Kontur Objek.....	62
2.8.4 Rantai Kode.....	65
2.8.5 Perimeter	68
2.9 Operasi Citra Warna	70
2.9.1 Dasar Warna	70
2.9.2 Ruang Warna	72
2.9.3 Memperoleh Statistika Warna	77
2.9.4 Mengatur Kecerahan dan Kontras.....	79
2.9.5 Menghitung Jumlah Warna	83
BAB III PENUTUP	84
3.1 Kesimpulan	84
DAFTAR PUSTAKA	90

BAB I

PENDAHULUAN

1.1 Latar Belakang

Di era digital yang terus berkembang, citra digital memainkan peranan penting dalam berbagai aspek kehidupan, mulai dari hiburan, komunikasi, pendidikan, hingga penelitian ilmiah. Pemrosesan citra digital menjadi dasar untuk memahami, mengolah, dan menganalisis informasi visual secara efektif. Oleh karena itu, penguasaan konsep dasar pembentukan citra dan elemen-elemen yang terkait merupakan langkah awal yang sangat penting dalam mendalami bidang ini.

Pemrosesan citra digital melibatkan berbagai operasi, seperti analisis piksel, transformasi geometri, morfologi, hingga pengolahan citra biner dan warna. Operasi piksel memungkinkan manipulasi detail terkecil pada citra, sedangkan metode histogram membantu dalam menganalisis distribusi intensitas piksel untuk meningkatkan kualitas visual. Operasi ketetanggaan piksel dan operasi aritmatika berkontribusi pada pengolahan citra dengan berbagai tingkat kompleksitas.

Selain itu, operasi geometri dan morfologi menyediakan teknik untuk memodifikasi bentuk atau struktur citra. Sementara itu, pengolahan citra biner dan warna memperluas aplikasi pemrosesan citra pada bidang-bidang seperti pengenalan pola, analisis medis, dan pengolahan multimedia.

Pentingnya pemahaman terhadap berbagai jenis operasi tersebut menjadi dasar untuk memecahkan berbagai permasalahan teknis dan aplikasi nyata. Oleh karena itu, tugas ini disusun untuk merangkum dan memahami:

1. Konsep pembentukan citra dan analisis elemen-elemen citra.
2. Operasi piksel pada citra digital.
3. Penerapan metode histogram dalam pengolahan citra.
4. Operasi ketetanggaan piksel, aritmatika, geometri, morfologi, serta pengolahan citra biner dan warna.

Melalui pemahaman dan analisis yang mendalam terhadap materi ini, diharapkan dapat meningkatkan kemampuan dalam menerapkan pemrosesan citra digital untuk menyelesaikan berbagai permasalahan di dunia nyata.

1.2 Rumusan Masalah

1. Memahami Konsep Pembentukan Citra Dan Mampu Menganalisa Dan Menjelaskan Elemen – Elemen Citra Atau pun Pemrosesan Citra Digital
2. Operasi Pixel Pada Citra Digital

3. Analisis Operasi Pixel Dan Penerapan Metode Histogram Dalam Pengolahan Citra Digital
4. Operasi Ketetanggaan Pixel
5. Operasi Aritmatika Citra
6. Operasi Geometri Citra
7. Operasi Morfologi
8. Operasi Pada Citra Biner
9. Operasi Citra Warna

1.3 Tujuan

1. Memahami Konsep Pembentukan Citra Digital
Mengidentifikasi, menganalisis, dan menjelaskan elemen-elemen dasar pembentukan citra serta proses yang terlibat dalam pemrosesan citra digital.
2. Menguasai Operasi Piksel pada Citra Digital
Memahami bagaimana operasi pada tingkat piksel dapat memengaruhi hasil pengolahan citra, serta menganalisis efek yang dihasilkan.
3. Menerapkan Metode Histogram dalam Pengolahan Citra
Menganalisis distribusi intensitas piksel pada citra melalui histogram, dan memanfaatkan metode ini untuk meningkatkan kualitas visual dan melakukan pengolahan lebih lanjut.
4. Mengaplikasikan Operasi Ketetanggaan Piksel
Memahami konsep ketetanggaan piksel dan menerapkannya untuk operasi pengolahan citra yang memanfaatkan hubungan antar-piksel.
5. Memahami Operasi Aritmatika pada Citra
Mempelajari dan mengaplikasikan operasi matematis sederhana seperti penjumlahan, pengurangan, perkalian, dan pembagian pada citra untuk tujuan pengolahan tertentu.
6. Menerapkan Operasi Geometri pada Citra
Menganalisis dan melakukan operasi geometri seperti translasi, rotasi, dan skala untuk mengubah struktur citra sesuai kebutuhan.
7. Menguasai Operasi Morfologi pada Citra
Memahami dan menerapkan operasi morfologi untuk manipulasi struktur dan bentuk dalam citra digital, terutama pada citra biner.
8. Melakukan Operasi pada Citra Biner
Memahami teknik pengolahan khusus untuk citra biner, termasuk thresholding dan operasi logika.
9. Mengenal Operasi pada Citra Warna
Mempelajari dan menganalisis teknik pengolahan citra warna untuk berbagai aplikasi seperti pengenalan pola, segmentasi, dan peningkatan kualitas citra.

Tujuan utama dari rekap ini adalah memberikan pemahaman yang menyeluruh terhadap setiap topik, sehingga dapat menjadi landasan kuat untuk menerapkan pemrosesan citra digital dalam berbagai konteks dan aplikasi.

BAB II

PEMBAHASAN

2.1 Memahami Konsep Pembentukan Citra Dan Mampu Menganalisa Dan Menjelaskan Elemen – Elemen Citra Atau Proses Citra Digital

2.1.1 Pembentukan Citra

Pengertian umum citra adalah kombinasi antara titik, garis, bidang, dan warna untuk menciptakan suatu imitasi dari suatu objek—biasanya objek fisik atau manusia. Adapun pengertian lain, citra adalah suatu representasi (gambaran), kemiripan, atau imitasi dari suatu objek. Citra terbagi 2 yaitu citra yang bersifat analog dan ada citra yang bersifat digital. Citra analog adalah citra yang bersifat continue seperti gambar pada monitor televisi, foto sinar X, dan lain-lain. Sedangkan pada citra digital adalah citra yang dapat diolah oleh komputer. Citra dapat didefinisikan sebagai fungsi $f(x,y)$ berukuran M baris dan N kolom, dengan x dan y adalah koordinat spasial, dan amplitudo f di titik koordinat (x,y) dinamakan intensitas atau tingkat keabuan dari citra pada citra tersebut.

Perbedaan Citra Analog dan Citra Digital

- Citra analog adalah citra yang bersifat continue, seperti gambar pada monitor televisi, foto sinar X, foto yang tercetak di kertas foto, lukisan, pemandangan alam, hasil CT scan, gambar-gambar yang terekam pada pita kaset, dan lain sebagainya. Citra analog tidak dapat direpresentasikan dalam komputer, sehingga tidak bisa diproses di komputer secara langsung. Oleh sebab itu, agar ini dapat diproses di komputer, proses konversi analog ke digital harus dilakukan terlebih dahulu. Citra analog dihasilkan dari alat-alat analog, seperti video kamera analog, kamera foto analog, cam, CT scan, sensor rontgen untuk foto thorax, sensor gelombang pendek pada sistem radar, sensor ultrasound pada sistem USG, dan lain-lain
- Citra digital merupakan representatif dari citra yang diambil oleh mesin dengan bentuk pendekatan berdasarkan sampling dan kuantisasi. Sampling menyatakan besarnya kotak-kotak yang disusun dalam baris dan kolom. Dengan kata lain, sampling pada citra menyatakan besar kecilnya ukuran pixel (titik) pada citra, dan kuantisasi menyatakan besarnya nilai tingkat kecerahan yang dinyatakan dalam nilai Tingkat keabuan (grayscale) sesuai dengan jumlah bit biner yang digunakan oleh mesin, dengan kata lain kuantisasi pada citra menyatakan jumlah warna yang ada pada citra.

a) Model Citra

Representasi dan deskripsi hampir selalu mengikuti keluaran dari tahap segmentasi, yang biasanya berupa data piksel mentah, yang

merupakan batas suatu wilayah atau semua titik di wilayah itu sendiri. Memilih representasi hanyalah sebagian dari solusi untuk mengubah data mentah menjadi bentuk yang sesuai untuk pemrosesan komputer selanjutnya. Deskripsi berkaitan dengan penggalian atribut yang menghasilkan beberapa informasi kuantitatif yang menarik atau dasar untuk membedakan satu kelas objek dari yang lain.

Representasi Citra digital adalah sebuah aktivitas dalam kegiatan pengolahan citra digital, dimana kegiatan ini merupakan proses menampilkan kembali suatu citra yang telah melalui tahap digitizing (proses pengubahan bentuk citra analog ke dalam format digital agar mampu dilakukan proses manipulasi oleh komputer) dengan cara mencacah gambar tersebut dalam bentuk titik – titik warna yang ditandai dengan angka yang menunjukkan tingkat kecerahan warna tersebut, kemudian dipetakan dengan menggunakan sistem koordinat.

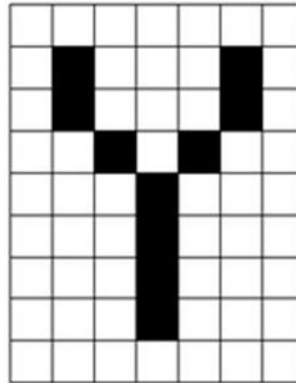
Koordinat Matriks dan koordinat piksel tidak memiliki perbedaan dalam operasi matematisnya. Perbedaan dari koordinat matriks dan koordinat piksel adalah koordinat matriks menunjukkan letak suatu titik pada citra asli, dan koordinat piksel menunjukkan letak suatu titik pada citra di layer monitor. Berdasarkan format representasi citra diatas, maka dapat disimpulkan bahwa suatu citra dinyatakan dalam bentuk lebar x tinggi. Citra digital memiliki satuan berupa piksel, ataupun dalam satuan panjang (mm atau inci).

Adapun mekanisme representasi citra digital adalah suatu elemen matriks diisi oleh angka–angka yang mewakilkan warna–warna yang tampak pada mata. Kumpulan angka yang mewakilkan warna pada matriks tersebut kemudian disimpan dalam komputer dengan berbagai format citra yang ada, dan memerlukan program khusus untuk membukanya (seperti Ms.Paint, Photoshop, dll). Sehingga dapat disimpulkan bahwa sebuah data citra digital menyimpan informasi berupa kumpulan angka yang mewakilkan warna yang ada. Suatu citra digital dapat ditampilkan dalam tiga format tampilan, diantaranya:

1. Citra Biner

Citra biner merupakan salah satu cara dalam merepresentasikan citra digital dimana citra ini menggunakan dua jenis warna saja, yakni hitam dan putih. Kedua warna ini masing – masing diwakili oleh angka – angka biner (0 dan 1). Dalam mewakili warna hitam dan putih, angka biner memiliki ketentuan sebagai berikut:

- a. Model citra cahaya : angka 1 mewakili warna putih, dan angka 0 mewakili warna hitam (warna putih menyatakan adanya cahaya, warna hitam menyatakan tidak ada cahaya).
- b. Model citra tinta / cat : angka 1 mewakili warna hitam, dan angka 0 mewakili warna putih (warna hitam menandakan adanya cat, warna putih menandakan tidak ada cat).



1	1	1	1	1	1	1
1	0	1	1	1	0	1
1	0	1	1	1	0	1
1	1	0	1	0	1	1
1	1	1	0	1	1	1
1	1	1	0	1	1	1
1	1	1	0	1	1	1
1	1	1	0	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1

Contoh: Citra biner dengan model citra Cahaya

2. Citra Grayscale

Citra grayscale merupakan suatu cara dalam merepresentasikan citra digital dengan menggunakan skala derajat keabuan, dimana derajat keabuan yang ada merupakan hasil pemangkatan nilai bit yang ada terhadap angka 2 (2^n).

Misalkan skala keabuan 4 bit memiliki rentang skala keabuan sebanyak 24 warna = 16 warna, yang diwakili dengan angka 0 hingga 15 (angka 0 / minimal mewakili warna hitam, dan angka 15 / maksimal mewakili warna putih). Adapun angka diantara 0 hingga 15 merepresentasikan warna abu dalam skala kecerahan yang berbeda.

3. Citra Warna

Citra warna merupakan metode dalam merepresentasikan suatu citra secara digital, dimana metode ini menggunakan kombinasi dari tiga warna primer (merah, hijau dan biru = RGB) untuk membentuk suatu citra. Adapun setiap titik pada citra mewakili kombinasi dari ketiga warna ini. Setiap warna ini masing masing memiliki intensitas tersendiri dengan rentang nilai 0 hingga 255 (8 bit).

- Red : warna minimal putih, warna maksimal merah
- Green : warna minimal putih, warna maksimal hijau
- Blue : warna minimal putih, warna maksimal biru

Misalkan warna ungu = merupakan kombinasi warna merah dan biru, sehingga nilai RGBnya: 255 0 255. Jika ketiga warna pada suatu piksel memiliki angka minimal, maka warna yang ditunjukkan pada piksel tersebut adalah warna hitam. Jika ketiga warna pada suatu piksel menunjukkan angka maksimal, maka warna yang ditunjukkan adalah warna putih. Jika salah satu dari ketiga angka pada piksel memiliki nilai minimal, maka warna tersebut tidak terkandung pada warna yang ditampilkan.

Contoh: pada kombinasi warna ungu diatas, dapat disimpulkan bahwa warna ungu tidak mengandung warna hijau, karena nilai skala warna hijau pada warna tersebut adalah 0.

Mengingat bahwa setiap piksel merupakan kombinasi dari ketiga warna ini, maka satu piksel memerlukan memori sebanyak 3 bit. Adapun jumlah total dari kombinasi warna yang mungkin adalah sebagai berikut:

- a. Warna dasar terdiri atas 3 warna
- b. Masing – masing warna dasar memiliki nilai maksimum 8 bit
- c. Sehingga: kemungkinan jumlah warna yang ada = $28 \times 3 = 224 = 16.777.216$ warna.

Ditunjukkan adalah warna putih. Jika salah satu dari ketiga angka pada piksel memiliki nilai minimal, maka warna tersebut tidak terkandung pada warna yang ditampilkan.

Contoh: pada kombinasi warna ungu diatas, dapat disimpulkan bahwa warna ungu tidak mengandung warna hijau, karena nilai skala warna hijau pada warna tersebut adalah 0. Mengingat bahwa setiap piksel merupakan kombinasi dari ketiga warna ini, maka satu piksel memerlukan memori sebanyak 3 bit.

Adapun jumlah

total dari kombinasi warna yang mungkin adalah sebagai berikut:

- a. Warna dasar terdiri atas 3 warna
- b. Masing – masing warna dasar memiliki nilai maksimum 8 bit
- c. Sehingga: kemungkinan jumlah warna yang ada = $28 \times 3 = 224 = 16.777.216$ warna.

Catatan : warna dasar pada konteks ini memiliki perbedaan, yakni warna dasar untuk cahaya / diplay pada monitor dan warna dasar untuk cat atau tinta / display cetakan diatas kertas.

1. Citra cahaya menggunakan warna dasar RGB (Red, Green, Blue)
2. Citra cat menggunakan warna dasar CMY (Cyan, Magenta, Yellow)

Berikut merupakan perbandingan ketiga citra yang dibahas pada bahasan ini:



RGB



GRAY SCALE



BINARY

b) Digitalisasi Citra

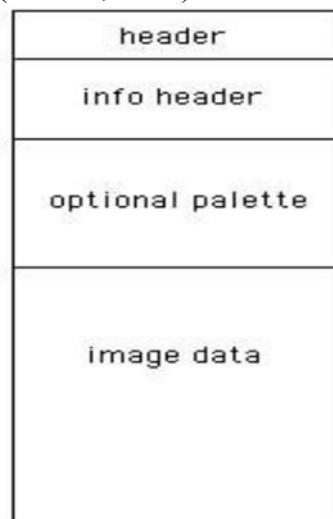
Sebuah format file citra harus dapat menyatukan kualitas citra, ukuran file dan kompatibilitas dengan berbagai aplikasi. Format file citra

standar yang digunakan saat ini terdiri dari beberapa jenis. Format-format ini digunakan untuk menyimpan citra dalam sebuah file. Setiap format memiliki karakteristik masing-masing. Ini adalah contoh format umum, yaitu : Bitmap (.bmp), Graphics Interchange Format (.gif), Portable Network Graphics (.png), JPEG (.jpg) (Putra, 2010).

1. Bitmap (.bmp)

Format .bmp adalah format penyimpanan standar tanpa kompresi yang umum dapat digunakan untuk menyimpan citra biner hingga citra warna. Format ini terdiri dari beberapa jenis yang setiap jenisnya ditentukan dengan jumlah bit yang digunakan untuk menyimpan sebuah nilai pixel (Putra, 2010). File BMP adalah format file bersejarah (tapi masih umum digunakan) untuk sistem operasi yang disebut "Windows". Gambar BMP dapat berkisar dari hitam dan putih (1 bit per pixel) hingga 24 bit warna (16,7 juta warna). Sementara gambar dapat dikompresi (Bourke, 1998).

Sebuah file BMP terdiri dari 3 atau 4 bagian seperti yang ditunjukkan dalam diagram di bawah. Bagian pertama adalah header, ini diikuti oleh bagian informasi, jika gambar diindeks warna kemudian palet berikut, dan terakhir dari semua adalah data pixel. Posisi data gambar sehubungan dengan start file yang terkandung dalam header. Informasi seperti lebar gambar dan tinggi, jenis kompresi, jumlah warna yang terkandung dalam header informasi (Bourke, 1998).



2. Portable Network Graphics (.png)

Format .png adalah format penyimpanan citra terkompresi. Format ini dapat digunakan pada citra grayscale, citra dengan palet warna, dan juga citra fullcolor. Format .png juga mampu menyimpan informasi hingga kanal alpha dengan penyimpanan sebesar 1 hingga 16 bit per kanal (Putra, 2010). PNG singkatan dari Portable Network Graphics. PNG adalah format citra bitmap yang

menggunakan kompresi data lossless. PNG diciptakan untuk meningkatkan dan menggantikan format GIF.

Format file PNG dianggap, dan dibuat sebagai penerus gratis dan open source untuk format file GIF. Format file PNG mendukung true color (16 juta warna), sedangkan format file GIF hanya memungkinkan 256 warna. PNG unggul ketika citra memiliki area besar warna seragam. The lossless PNG paling cocok untuk mengedit citra, sedangkan format lossy seperti JPG yang terbaik untuk distribusi final fotografi-jenis citra karena ukuran file yang lebih kecil.

Namun banyak browser sebelumnya tidak mendukung format file PNG; Namun dengan rilis Internet Explorer 7 browser modern semua populer sepenuhnya mendukung PNG. Fitur khusus dari file PNG termasuk dukungan untuk hingga 48 bit informasi warna.

Format PNG memanfaatkan jalinan skema 2D, yang semakin menampilkan citra lebih cepat daripada sebuah file citra GIF. Gamma koreksi memungkinkan nilai-nilai yang ditampilkan pada platform apapun menjadi sama dengan yang asli (Jayaraman, et al., 2009).

3. JPEG (.jpg)

Format .jpg adalah format yang sangat umum digunakan saat ini khususnya untuk transmisi citra. Format ini digunakan untuk menyimpan citra hasil kompresi dengan metode JPEG (Putra, 2010). JPEG sebenarnya bukan jenis file. JPEG adalah standar saat yang paling penting untuk kompresi citra. Standar JPEG diciptakan oleh kelompok kerja Organisasi Internasional untuk Standardisasi (ISO). Format ini menyediakan pilihan kompresi paling dramatis untuk citra fotografi. Kompresi JPEG digunakan dalam format file JFIF yang menggunakan ekstensi file (.jpg). Format ini berguna ketika ruang penyimpanan adalah pada premium. JPEG menyimpan citra raster tunggal dalam warna 24-bit. JPEG adalah format platform-independen yang mendukung tingkat tertinggi kompresi; Namun, kompresi ini lossy. File JPEG progresif mendukung interlacing (Jayaraman, et al., 2009). Fitur penting dari format file JPEG adalah sebagai berikut (Jayaraman, et al., 2009):

1. JPEG menggunakan skema kompresi lossy.
2. citra JPEG tidak interlaced; Namun, citra JPEG progresif dapat interlaced.

File Format Kekuatan format file JPEG adalah kemampuannya untuk kompres file citra yang lebih besar. Karena kompresi ini, data citra dapat disimpan secara efektif dan efisien menular dari satu tempat ke tempat lain (Jayaraman, et al., 2009).

File Format JPEG dalam versi basis tidak mendukung beberapa lapisan, kisaran dinamis tinggi. Oleh karena itu JPEG tidak

akan menjadi pilihan bijak jika ada yang tertarik untuk mempertahankan citra berkualitas tinggi (Jayaraman, et al., 2009).

c) Elemen Elemen Citra Digital

Berikut adalah elemen-elemen yang terdapat pada citra digital:

1. **Kecerahan (Brightness)**
Brightness merupakan intensitas Cahaya yang dipancarkan piksel dari citra yang dapat ditangkap oleh sistem penglihatan. Kecerahan pada sebuah titik (piksel) di dalam citra merupakan intensitas rata-rata dari suatu area yang melingkupinya.
2. **Kontras (Contrast)**
Kontras menyatakan sebaran terang dan gelap dalam sebuah citra. Pada citra yang baik, komposisi gelap dan terang tersebar secara merata.
3. **Kontur (Contour)**
Kontur adalah keadaan yang ditimbulkan oleh perubahan intensitas pada piksel-piksel yang bertetangga. Karena adanya perubahan intensitas inilah mata mampu mendeteksi tepi-tepi objek di dalam citra.
4. **Warna**
Warna sebagai persepsi yang ditangkap sistem visual terhadap panjang gelombang cahaya yang dipantulkan oleh objek.
5. **Bentuk (Shape)**
Shape adalah properti intrinsik dari objek 3 dimensi, dengan pengertian bahwa bentuk merupakan property intrinsik utama untuk sistem visual manusia.
6. **Tekstur (Texture)**
Texture dicirikan sebagai distribusi spasial dari derajat keabuan di dalam sekumpulan piksel-piksel yang bertetangga. Tekstur adalah sifat-sifat atau karakteristik yang dimiliki oleh suatu daerah yang cukup besar, sehingga secara alami sifat-sifat tadi dapat berulang dalam daerah tersebut. Tekstur adalah keteraturan pola-pola tertentu yang terbentuk dari susunan piksel- piksel dalam citra digital. Informasi tekstur dapat digunakan untuk membedakan sifat-sifat permukaan suatu benda dalam citra yang berhubungan dengan kasar dan halus, juga sifat-sifat spesifik dari kekasaran dan kehalusan permukaan tadi, yang sama sekali terlepas dari warna permukaan tersebut.

2.1.2 Elemen System Pemrosesan citra Digital

Pengolahan citra digital adalah sebuah disiplin ilmu yang mempelajari hal- hal yang berkaitan dengan perbaikan kualitas gambar (peningkatan kontras, transformasi warna, restorasi citra), transformasi gambar (rotasi, translasi, skala, transformasi geometrik), melakukan

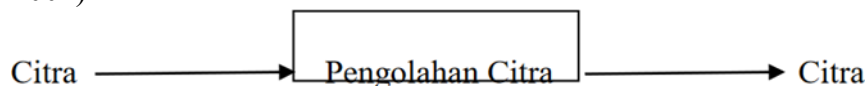
pemilihan citra ciri (feature images) yang optimal untuk tujuan analisis, melakukan proses penarikan informasi atau deskripsi objek atau pengenalan objek yang terkandung pada citra, melakukan

kompresi atau reduksi data untuk tujuan penyimpanan data, transmisi data, dan waktu proses data. Input dari pengolahan citra adalah citra, sedangkan outputnya adalah citra hasil pengolahan (Sutoyo, et al., 2009)

Pengolahan citra pada dasarnya mencakup tiga langkah berikut (Munir, 2004):

- a. Mengimpor gambar melalui alat akuisisi gambar.
- b. Menganalisis dan memanipulasi gambar.
- c. Output yang hasilnya dapat diubah gambar atau laporan yang didasarkan pada analisis citra.

Pengolahan Citra bertujuan memperbaiki kualitas citra agar mudah diinterpretasi oleh manusia atau mesin (dalam hal ini komputer). Teknik-teknik pengolahan citra mentransformasikan citra menjadi citra lain. Jadi, masukannya adalah citra dan keluarannya juga citra, namun citra keluaran mempunyai kualitas lebih baik daripada citra masukan. Termasuk ke dalam bidang ini juga adalah pemampatan citra (image compression)(Munir, 2004).



Perbaikan citra bertujuan meningkatkan kualitas tampilan citra untuk pandangan manusia atau untuk mengkonversi suatu citra agar memiliki format yang lebih baik sehingga citra tersebut menjadi lebih mudah diolah dengan mesin (komputer). Perbaikan terhadap suatu citra dapat dilakukan dengan operasi titik (point operation), operasi spasial (spatial operation), operasi geometri (geometric operation), dan operasi aritmatik (arithmetic operation) (Putra, 2010).

Perubahan bentuk suatu citra dapat berupa perubahan geometri pixel seperti perputaran (rotasi), pergeseran (translasi), penskalaan, dan lain sebagainya atau dapat juga berupa perubahan ruang (domain) citra ke domain lainnya, seperti transformasi Fourier yang mengubah suatu citra dari domain spasial menjadi domain frekuensi(Putra, 2010).

Melalui proses transformasi, suatu citra dapat dinyatakan sebagai kombinasi linier dari sinyal dasar (basis signals) yang sering disebut dengan fungsi basis (basis function). Suatu citra yang telah mengalami transformasi dapat diperoleh kembali dengan menggunakan transformasi balik (invers transformation). Tujuan diterapkannya transformasi citra adalah untuk memperoleh informasi (feature extraction) yang lebih jelas yang terkandung dalam suatu citra(Putra, 2010). Analisis citra digunakan untuk menghitung besaran kuantitatif dari citra untuk menghasilkan deskripsinya. Teknik analisis citra mengekstraksi ciri-ciri tertentu yang membantu dalam identifikasi objek, contoh operasi dalam analisis citra misalnya pendeteksian tepi objek (edge detection)(Munir, 2004).

Tepian citra dapat merepresentasikan objek-objek yang terkandung dalam citra tersebut, bentuk, dan ukurannya serta terkadang juga informasi tentang teksturnya. Tepian citra adalah posisi di mana intensitas pixel dari citra berubah dari nilai rendah ke nilai tinggi atau sebaliknya. Deteksi tepi umumnya adalah langkah awal melakukan segmentasi citra (Putra, 2010). Tepian citra dapat dilihat melalui perubahan intensitas pixel pada suatu area. Berdasarkan perbedaan perubahan intensitas tersebut, tepian dapat dibagi menjadi 4 jenis, antara lain adalah (Putra, 2010):

1. Step

Tepian jenis step merupakan tepian citra yang terbentuk dari perubahan intensitas citra secara signifikan dari tinggi ke rendah ataupun sebaliknya.



2. Ramp

Tepian jenis ini terbentuk dari perubahan intensitas citra secara perlahan. Perubahan secara perlahan dapat dilihat pada bentuk kurva yang semakin tinggi dengan perubahan kontinu.



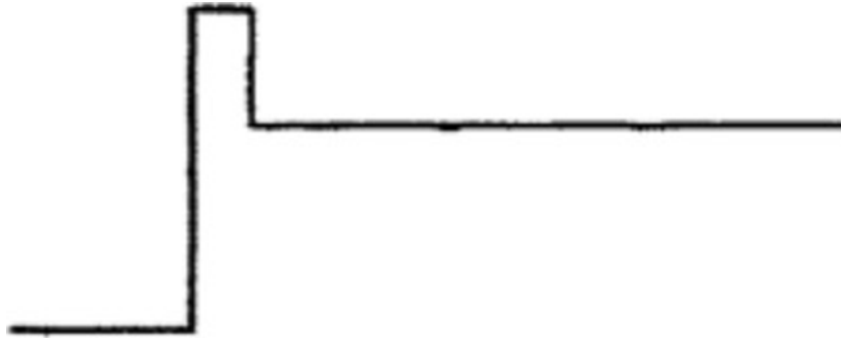
3. Line

Tepian jenis ini ditandai dengan perubahan intensitas secara drastis dari intensitas rendah-tinggi-rendah atau sebaliknya.



4. Step-line

Tepian step-line merupakan gabungan dari tepian jenis step dan line. Tepian jenis ini ditandai dengan peningkatan intensitas yang tajam dalam interval tertentu dan kemudian ditandai dengan penurunan yang tidak signifikan, sehingga perubahan intensitas selanjutnya berlangsung stabil.



Segmentasi merupakan teknik untuk membagi suatu citra menjadi beberapa daerah (region) di mana setiap daerah memiliki kemiripan atribut. Menurut (Jain, 1989), segmentasi citra dapat dibagi dalam beberapa jenis yaitu dividing image space dan clustering feature space. Jenis yang pertama adalah teknik segmentasi dengan membagi image menjadi beberapa bagian untuk mengetahui batasannya, sedangkan teknik yang kedua dilakukan dengan caramemberi index warna pada tiap piksel yang menunjukkan keanggotaan dalam suatu segmentasi.

2.2 Operasi Pixel Pada Citra Digital

2.2.1 Jenis Citra

Citra digital merupakan fungsi dua variabel $f(x,y)$, dimana x dan y adalah koordinat dan nilai $f(x,y)$ yang merupakan intensitas citra pada koordinat tersebut. Sedangkan pengertian citra digital secara harafiah adalah citra pada bidang dwimatra (dua dimensi). Ditinjau dari sudut pandang matematis, citra merupakan fungsi terus menerus (continue) dari intensitas cahaya pada bidang dwimatra. Sumber cahaya menerangi objek, kemudian objek memantulkan kembali sebagian dari berkas cahaya tersebut. Pantulan cahaya ini ditangkap oleh alat-alat optik seperti mata manusia, kamera, dan pemindai (scanner). Citra sebagai keluaran dari suatu sistem perekaman dapat bersifat optik berupa foto (Munir, 2004).

Citra digital terbagi 2, yaitu citra bitmap dan citra vector. Citra bitmap merupakan sebuah struktur data yang mewakili susunan piksel warna yang ditampilkan pada layar monitor ataupun kertas. Citra bitmap sangat bergantung pada resolusi. Citra bitmap Jika diperbesar maka citra akan tampak kurang halus sehingga mengurangi detailnya dan mempunyai ukuran file yang lebih besar. kelebihan citra bitmap mampu menangkap nuansa warna dan bentuk secara natural dan lebih sesuai dan lebih cepat ditampilkan ke layar monitor. Format citra bitmap yang banyak digunakan adalah GIF, JPEG, BMP dan PNG.

Citra vektor adalah citra yang dibuat dari unsur garis dan kurva yang disebut vektor. Citra Vektor terdiri dari penggabungan koordinat-koordinat titik menjadi garis atau kurva untuk kemudian menjadi sebuah objek, sehingga citra tidak menjadi pecah walaupun diperbesar atau diperkecil. Citra vektor umumnya memiliki ukuran yang lebih kecil bila dibandingkan

dengan citra bitmap. Citra vektor dapat diperbesar atau diperkecil ukurannya tanpa kehilangan detail citraa dan mempunyai ukuran file yang lebih kecil. Citra vector digunakan untuk citra logo dan kartun.

Jenis-jenis citra dikelompokkan menjadi tiga (Kaswidjanti, 2011) yaitu citra biner (monokrom), citra skala keabuan (grayscale) dan citra warna (true color).

a) Citra Biner

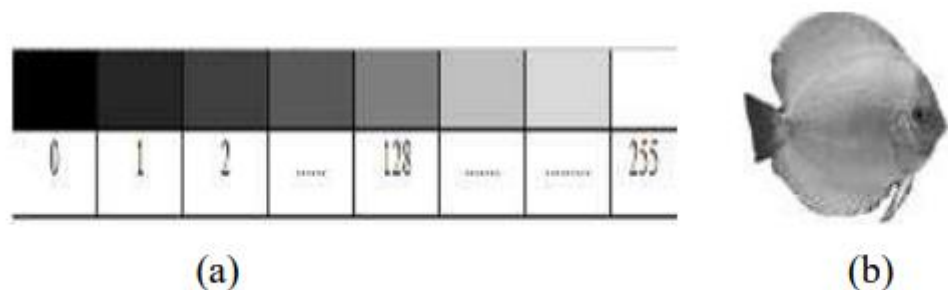
Citra biner (binary image) adalah citra yang hanya mempunyai dua nilai warna yaitu, hitam dan putih. Meskipun saat ini citra berwarna lebih disukai karena memberi kesan yang lebih kaya daripada citra biner, namun tidak membuat citra biner mati (Munir,2004). Citra biner merepresentasikan citra hanya dengan satu warna. Salah satu contoh dari citra biner dapat dilihat pada Citra 2.1.



Gambar 2.1 (a) Gradasi Warna Citra Biner (b) Citra Biner

b) Citra Grayscale

Citra grayscale merupakan citra yang menggunakan tingkatan warna abu-abu. Citra grayscale menyatakan nilai intensitas tiap piksel dengan nilai tunggal, dimana setiap piksel komponen merah, hijau, dan biru mempunyai intensitas yang sama. Citra grayscale lebih mudah diproses oleh karena hanya memiliki satu nilai intensitas pada setiap pixel. Intensitas nilai citra grayscale berkisar antara 0 sampai dengan 255. Nilai 0 menyatakan hitam dan nilai 255 menyatakan putih (kadir, 2013) sehingga warna antaranya adalah abu-abu, seperti ditunjukkan pada citra 2.2.



Gambar 2.2 (a) Gradasi Warna Citra Grayscale (b) Citra Grayscale

Derajat keabuan pada citra grayscale memiliki beberapa skala nilai, tidak hanya skala 0 sampai 255. Hal ini tergantung nilai

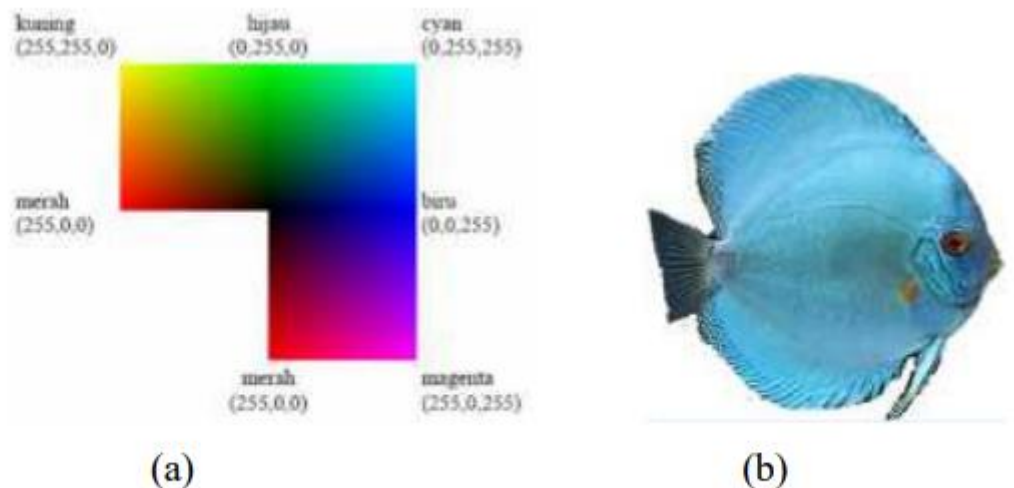
kedalaman pixel citra. Contoh pembagian nilai derajat keabuan berdasarkan kedalaman pixel ditunjukkan pada tabel 1.

Tabel 2.1 Pembagian Nilai Derajat Keabuan

<i>Grayscale</i>	<i>Scale (O,L)</i>	<i>Pixel depth</i>
2^1 (2)	0 sampai 1	1 bit
2^2 (4)	0 sampai 3	2 bit
2^4 (16)	0 sampai 15	4 bit
2^8 (256)	0 sampai 255	8 bit

c) Citra Warna

Citra warna merupakan citra dengan campuran warna, dimana setiap piksel yang terdapat pada citra merupakan kombinasi dari tiga warna dasar RGB (Red Green Blue) (kadir, 2013). Setiap warna dasar menggunakan penyimpanan 8 bit = 1 byte, yang berarti setiap warna memiliki gradasi sebanyak 255 warna. Berarti setiap piksel memiliki kombinasi warna sebanyak lebih dari 16 juta warna. Itulah sebabnya format ini dinamakan true color karena memiliki jumlah warna yang cukup besar. Gradasi citra warna dan contoh citra berwarna dapat dilihat pada gambar 2.3



Gambar 2.3 (a) Gradasi Warna RGB (b) Citra Warna

Warna adalah persepsi yang dirasakan oleh sistem visual manusia terhadap panjang gelombang cahaya yang dipantulkan oleh objek. Setiap warna mempunyai panjang gelombang yang berbeda (Fatmawati, 2011). Warna merah mempunyai panjang gelombang paling tinggi, sedangkan warna ungu mempunyai panjang gelombang paling

rendah. Citra warna adalah citra dengan sistem grafik yang memiliki satu set nilai tersusun yang menyatakan berbagai tingkat warna (Siregar, 2009). Setiap piksel pada citra warna mewakili warna yang merupakan kombinasi dari tiga warna dasar (RGB = Red Green Blue). Manusia punya 5-7 juta sel yang dibagi menjadi tiga kategori sensor (merah, hijau, dan biru). 65% sel kerucut sensitif pada warna merah, 33% pada warna hijau, dan 2% pada warna biru. Pada karakteristik mata manusia warna terlihat seperti kombinasi warna variabel yang disebut primer, yaitu merah (red), hijau (green), dan biru (blue). Warna primer dapat digunakan untuk menghasilkan warna sekunder. 1. Magenta = merah + biru 2. Cyan = hijau + biru 3. Kuning = merah + hijau Warna-warna yang diterima oleh mata merupakan hasil kombinasi cahaya dengan panjang gelombang berbeda. Kombinasi warna yang memberikan rentang warna yang paling lebar adalah red(R), green(G) dan blue(B) dan warna bukan merupakan besaran fisik tetapi warna merupakan suatu sensasi yang dihubungkan dengan sistem saraf kita, seperti halnya rasa maupun bau. Sensasi warna diperoleh dengan adanya interaksi antara warna dengan sistem saraf sensitive warna kita.

Sistem RGB digabungkan untuk memperoleh warna tertentu. Misalnya warna putih diperoleh dari hasil gabungan warna merah = 255, hijau = 255, dan biru = 255. Dalam sistem RGB, warna putih cerah dinyatakan dengan RGB (255, 255, 255). Nilai dari setiap primer adalah 0 sampai 255. Sehingga kemungkinan warna yang didapat adalah $256 \times 256 \times 256$ yakni kurang lebih 16.7 juta warna. Pada tabel 2.1 akan diperlihatkan beberapa hasil penggabungan kode warna RGB.

Tabel 2.3 Kode Warna RGB

Colour	Red	Green	Blue
Black	0	0	0
Blue	0	0	255
Green	0	255	0
Cyan (Blue + Green)	0	255	255
Red	255	0	0
Magenta (Red+Blue)	255	0	255
Yellow (Red+Green)	255	255	0
White (Red+Green+Blue)	255	255	255
Gray	128	128	128

2.2.2 Sytem Koordinat Warna

Sistem koordinat warna adalah cara untuk mengidentifikasi dan menggambarkan berbagai warna dalam ruang tiga dimensi. Berikut adalah beberapa sistem koordinat warna yang paling umum digunakan dalam berbagai aplikasi pengolahan citra dan teknologi:

a) **System koordinat Warna CIE**

Komisi Internasional untuk Penerangan (CIE) telah mengembangkan sistem koordinat warna yang sangat penting dalam pengolahan citra dan aplikasi percetakan, yaitu CIELab (Lab) dan CIE XYZ.

CIE XYZ:

- **Sumber Utama:** CIE XYZ menggunakan cahaya monokromatis dengan panjang gelombang 700 nm (merah), 546 nm (hijau), dan 435 nm (biru) sebagai sumber utama. Sistem warna CIE dimodelkan atas nilai rasio kontribusi ketiga komponen tersebut.
 - **X:** Mewakili kontribusi cahaya dengan panjang gelombang 700 nm.
 - **Y:** Mewakili kontribusi cahaya dengan panjang gelombang 546 nm.
 - **Z:** Mewakili kontribusi cahaya dengan panjang gelombang 435 nm.
- **Transformasi:** Transformasi dari basis CIE RGB ke CIE XYZ dan sebaliknya dapat dilakukan untuk acuan platform perangkat keras tertentu. Formula transformasi biasanya menggunakan perumusan matematika untuk mengkonversi salah satu model ke model lainnya. Contoh transformasi dari CIE RGB ke CIE XYZ adalah sebagai berikut:

$$X=0.412453R+0.357580G+0.180423B \quad X=0.412453R+0.357580G+0.180423B$$

$$Y=0.212671R+0.715160G+0.072169B \quad Y=0.212671R+0.715160G+0.072169B$$

$$Z=0.019334R+0.119193G+0.950227B \quad Z=0.019334R+0.119193G+0.950227B$$

CIE Lab:

- **Definisi:** CIELab (Lab) dimodelkan setelah CIE XYZ dengan menambahkan dimensi ringan (Lightness) untuk membedakan level terang-gelap. Koordinatnya melibatkan L sebagai nilai kecerahan, a dan b sebagai kroma (koordinat merah-hijau dan kuning-biru).
 - **L:** Nilai kecerahan yang berada di antara 0 (hitam) dan 100 (putih).
 - **a:** Nilai kroma merah-hijau, di mana nilai positif menunjukkan arah merah dan nilai negatif menunjukkan arah hijau.

- *b*: Nilai kroma kuning-biru, di mana nilai positif menunjukkan arah kuning dan nilai negatif menunjukkan arah biru.
- **Perbedaan Warna:** Perbedaan warna dapat diidentifikasi dengan menggunakan delta (Δ) nilai antara koordinat sampel dan standar. Perhitungan seperti ΔL , Δa , dan Δb memberikan nilai numerik perbedaan koordinat, membantu mengontrol konsistensi warna produk.

Penggunaan CIE XYZ dan CIELab:

- **Pengolahan Citra:** Dalam pengolahan citra, sistem CIE XYZ dan CIELab digunakan untuk mengkonversi citra dari basis RGB ke basis yang lebih objektif dan tidak bergantung pada perangkat keras. Ini sangat penting dalam aplikasi seperti pengolahan gambar, percetakan, dan desain grafis.
- **Manajemen Warna:** Sistem ini juga digunakan dalam manajemen warna untuk memastikan bahwa produk yang dihasilkan memiliki konsistensi warna yang tinggi. Perusahaan percetakan dan industri desain grafis sering menggunakan CIELab untuk mengukur perbedaan warna dan memastikan bahwa produk mereka memenuhi standar kualitas.

b) System koordinat Warna RGB

Ruang Warna RGB merupakan salah satu ruang warna yang paling umum digunakan, terutama dalam sistem tampilan gambar di komputer dan televisi.

Penentuan Warna:

- **Nilai Warna:** Warna ditentukan oleh nilai Red (R), Green (G), dan Blue (B) yang dinormalisasi dari 0.0 hingga 1.0. Setiap tuple (R, G, B) dapat mewakili satu titik dalam ruang warna, dengan (0,0,0) merepresentasikan hitam dan (1,1,1) merepresentasikan putih.
- **R:** Nilai merah yang berada di antara 0 dan 1.
- **G:** Nilai hijau yang berada di antara 0 dan 1.
- **B:** Nilai biru yang berada di antara 0 dan 1.

Ketergantungan:

- **Perangkat Keras:** Ruang Warna RGB dapat bergantung pada perangkat, sehingga nilai warna dapat berbeda kecil tetapi signifikan antara perangkat penggunaan yang berbeda. Misalnya, warna yang ditampilkan pada monitor komputer mungkin berbeda sedikit jika dilihat pada layar televisi.

Penggunaan RGB:

- **Pengolahan Citra:** Dalam pengolahan citra digital, sistem RGB digunakan sebagai basis utama karena kebanyakan perangkat keras komputer dapat menampilkan citra dalam format ini. Namun, untuk aplikasi yang memerlukan akurasi tinggi dan ketepatan warna, konversi ke sistem CIE XYZ atau CIELab sering dilakukan.
- **Desain Grafis:** Sistem RGB juga digunakan dalam desain grafis karena kemampuan menampilkan warna yang luas dan fleksibel. Desainer grafis sering menggunakan aplikasi seperti Adobe Photoshop yang menggunakan basis RGB untuk mengolah citra dan membuat desain.

c) System Koordinat Warna Berdasarkan Persepsi Warna

Ruang Warna yang berbasis pada persepsi warna seperti HSL (Hue, Saturation, Lightness) dan HSV (Hue, Saturation, Value) dirancang untuk meningkatkan kemudahan dalam mengidentifikasi dan memanipulasi warna berdasarkan komponen yang lebih intuitif.

HSV:

Definisi: Model HSV menggunakan sistem koordinat yang lebih mudah dipahami karena membagi komponen warna menjadi Hue, Saturation, dan Value.

- **Hue (H):** Menentukan warna dasar (hijau, merah, biru).
- **Saturation (S):** Menentukan intensitas atau ketajaman warna.
- **Value (V):** Menentukan terang-gelap dari warna.

HSL:

Definisi: Model HSL juga membagi komponen warna menjadi Hue, Saturation (Kroma), dan Lightness (Terang-gelap), tetapi dengan perbedaan kecil dalam penggunaan istilah-istilah tersebut.

- **Hue (H):** Sama seperti HSV.
- **Saturation (S):** Sama seperti HSV.
- **Lightness (L):** Menentukan terang-gelap dari warna.

Konversi Warna:

Kedua model warna tersebut dapat dikonversi ke model RGB atau CIE XYZ melalui algoritma matematika tertentu untuk memenuhi kebutuhan aplikasi yang berkaitan dengan pengolahan citra dan tampilan gambar.

Penggunaan HSV dan HSL:

- **Desain Grafis:** Sistem HSV dan HSL digunakan dalam desain grafis karena memungkinkan desainer untuk mengidentifikasi dan memanipulasi warna dengan lebih mudah. Aplikasi seperti Adobe Photoshop sering menggunakan basis HSV atau HSL untuk mengolah citra dan membuat desain.
- **Pengembangan Perangkat Lunak:** Dalam pengembangan perangkat lunak, sistem ini digunakan untuk mengolah citra dan mengatur warna dalam aplikasi grafis. Ini membantu meningkatkan kemampuan pengguna untuk mengatur dan memanipulasi warna dalam aplikasi.

Kelebihan dan Kekurangan:

a. Kelebihan:

- **Intuitif:** Sistem HSV dan HSL lebih intuitif karena membagi komponen warna menjadi bagian yang lebih mudah dipahami.
- **Mudah Digunakan:** Sistem ini lebih mudah digunakan dalam aplikasi desain grafis dan pengolahan citra.

b. Kekurangan:

- **Ketergantungan pada Perangkat:** Sistem ini masih bergantung pada perangkat keras yang digunakan, sehingga nilai warna dapat berbeda sedikit antara perangkat yang berbeda.
- **Kompleksitas Perhitungan:** Meskipun lebih intuitif, perhitungan untuk mengkonversi antara sistem ini dan sistem lain seperti CIE XYZ atau RGB masih memerlukan perhitungan matematika yang kompleks.

Dalam keseluruhan, pemahaman tentang sistem koordinat warna sangat penting untuk aplikasi pengolahan citra dan manajemen warna yang akurat dalam berbagai bidang teknologi. Masing-masing model memiliki kelebihan dan kelemahan, tetapi semua dirancang untuk merepresentasikan kompleksitas warna dalam cara yang efektif bagi berbagai konteks penggunaan.

d) System ruang warna HSV

Sistem ruang warna HSV (Hue, Saturation, Value) merupakan model warna yang dikembangkan oleh ahli grafis untuk lebih dekat dengan cara manusia secara alami memandang warna. Sementara sistem warna RGB didasarkan pada pencampuran warna-warna primer cahaya (merah, hijau, dan biru), HSV menggambarkan warna berdasarkan karakteristik visual yang lebih mudah dimengerti dan digunakan dalam konteks kreatif dan teknik manipulasi gambar.

HSV merupakan salah satu model warna yang memetakan warna dalam bentuk kerucut atau silinder, yang mana posisi warna ditentukan berdasarkan tiga parameter utama: Hue, Saturation, dan Value.

Komponen Utama HSV

1. Hue (Nada Warna):

- Hue mewakili "jenis" warna yang kita lihat, seperti merah, kuning, biru, atau hijau. Hue diukur dalam derajat pada lingkaran warna, mulai dari 0 hingga 360 derajat.
- 0° untuk warna merah.
- 60° untuk warna kuning.
- 120° untuk warna hijau.
- 180° untuk warna cyan.
- 240° untuk warna biru.
- 300° untuk warna magenta.
- Hue tidak menunjukkan kecerahan atau kejenuhan warna, hanya menunjukkan posisinya pada spektrum warna. Misalnya, merah di Hue 0° adalah warna yang sama baik itu merah terang atau merah gelap, tergantung pada nilai Saturation dan Value.

2. Saturation (Kejenuhan):

- Saturation mengukur seberapa murni warna yang ditampilkan, atau seberapa "jenuh" warna tersebut. Saturasi menunjukkan seberapa banyak warna yang bercampur dengan putih.
- Saturation 100% berarti warna paling murni atau penuh.
- Saturation 0% berarti warna menjadi abu-abu, yaitu tidak ada kejenuhan sama sekali.
- Warna yang jenuh sepenuhnya (100%) akan terlihat sangat kuat dan murni, sedangkan warna dengan kejenuhan rendah akan terlihat lebih pudar atau campuran dari warna dan abu-abu.

3. Value (Nilai atau Kecerahan):

- Value atau Kecerahan merujuk pada seberapa terang atau gelap suatu warna. Ini berkisar dari 0% hingga 100%. Nilai yang lebih rendah (dekat 0%) mendekati hitam, sementara nilai yang lebih tinggi (dekat 100%) mendekati warna yang sangat terang.
- Value (V) berfungsi untuk mengukur intensitas cahaya suatu warna. Misalnya, dengan Value yang lebih rendah, meskipun Hue dan Saturation tetap sama, warna akan tampak lebih gelap hingga mendekati hitam.

Bentuk Geometris Ruang Warna HSV

Ruang warna HSV seringkali digambarkan sebagai kerucut atau silinder:

- Kerucut HSV:
 - Hue diatur di sepanjang tepi melingkar kerucut.

- Saturation meningkat dari pusat kerucut ke tepi.
- Value meningkat dari dasar (gelap/hitam) ke puncak kerucut (terang/putih).
- Silinder HSV:
 - Pada silinder HSV, Hue ditampilkan pada sumbu melingkar horizontal, sedangkan Saturation disebar dari pusat silinder ke luar, dan Value diukur dari bawah (hitam) ke atas (putih).

Kegunaan Ruang Warna HSV

1. Manipulasi Warna:

HSV sangat berguna dalam pengeditan gambar karena lebih intuitif untuk manusia. Misalnya, dengan menggunakan model HSV, kita bisa mengubah warna (Hue) tanpa mempengaruhi kecerahan (Value) atau kejenuhan (Saturation), dan sebaliknya. Ini mempermudah pengeditan gambar dan video.

2. Perbaikan Gambar Otomatis:

Pengolahan gambar sering menggunakan HSV untuk memperbaiki warna otomatis pada gambar, karena dalam HSV, memanipulasi kecerahan atau kontras lebih mudah tanpa mengubah warna dasar gambar.

3. Segmentasi Warna dalam Visi Komputer:

Dalam banyak aplikasi visi komputer, seperti deteksi objek atau pengenalan pola, ruang warna HSV digunakan untuk segmentasi warna. Karena Hue lebih tahan terhadap perubahan pencahayaan dibandingkan dengan RGB, HSV memudahkan untuk mengidentifikasi objek berdasarkan warna di lingkungan dengan pencahayaan yang berubah-ubah.

4. Desain Grafis dan Seni Digital:

Banyak alat desain grafis menggunakan HSV untuk memberi kontrol lebih baik kepada pengguna atas penyesuaian warna. Desainer dapat dengan mudah menyesuaikan nada warna sambil mempertahankan tingkat kecerahan dan kejenuhan yang diinginkan.

Kelebihan Dan Kekurangan Hsv

a. Kelebihan:

- Intuitif: Representasi HSV lebih sesuai dengan cara manusia memandang warna, menjadikannya lebih mudah digunakan dalam manipulasi warna.
- Pisahkan Komponen: Memisahkan antara warna, kejenuhan, dan kecerahan memberikan fleksibilitas dalam mengubah salah satu komponen tanpa mempengaruhi yang lain.
- Efektif untuk Pengeditan Warna: Lebih mudah untuk melakukan perubahan halus pada satu aspek warna tanpa mempengaruhi aspek lain seperti dalam ruang warna RGB.

b. Kekurangan:

- Tidak Selalu Presisi: Untuk beberapa aplikasi teknis atau ilmiah, HSV mungkin tidak memberikan representasi warna yang presisi seperti model warna lain, seperti CIELAB.
- Kejenuhan Tidak Selalu Berarti: Dalam kondisi pencahayaan tertentu, kejenuhan warna dapat menjadi tidak relevan atau sulit diukur dengan benar.

2.2.3 Format File Citra

Citra Digital memiliki beberapa format umum diantaranya Bitmap (BMP), Joint Photographic Group Expert (JPEG/JPG), Graphics Interchange Format (GIF), dan Purchase Ordentable Network Graphics (PNG).

1. **Bitmap (BMP)**

Format baku citra pada sistem operasi windows dan IBM OS/2 adalah bitmap. Citra berformat Bitmap (BMP) merupakan citra yang tidak terkompresi, sehingga pada umumnya citra berformat BMP memiliki ukuran relative lebih besar dibandingkan dengan format citra yang lainnya. Intensitas pada pixel dari citra berformat BMP dipetakan ke sejumlah bit tertentu. Panjang setiap pixel pada bitmap adalah 4 bit, 8 bit, hingga 24 bit yang merepresentasikan nilai intensitas pixel. Maka dari itu ada sebanyak $2^8=256$ derajat keabuan, mulai dari 0 sampai dengan 255.

2. **Joint Photographic Group Expert (JPEG/JPG)**

Joint Photographic Group Expert atau yang biasa kita kenal JPEG/ JPG merupakan format standar kompresi file yang dikembangkan oleh Group Joint Photographic Expert dengan menggunakan kombinasi DCT dengan pengkodean Huffman untuk mengkompresikan citra. Citra dengan format JPEG merupakan citra terkompresi yang bersifat lossy, artinya dimana citra tidak dapat dikembalikan ke bentuk aslinya. Citra ini memiliki ukuran yang relative lebih kecil dibandingkan dengan citra berformat BMP karena telah terkompresi.

3. **Graphics Interchange Format (GIF)**

Graphics Interchange Format (GIF) merupakan format citra terkompresi. Ukuran dari citra berformat GIF adalah yang terkecil dari semua format citra digital. Kombinasi warna yang tersedia adalah sebanyak 256 warna. Citra pada jenis ini biasanya banyak digunakan untuk keperluan website, untuk membuat sebuah gambar, icon, untuk membuat animasi dan keperluan lainnya yang mengharuskan menggunakan citra dengan ukuran yang sangat kecil.

4. **Purchase Ordentable Network Graphics (PNG)**

Purchase Ordentable Network Graphics atau lebih dikenal dengan PNG merupakan salah satu format penyimpanan citra yang menggunakan metode kompresi yang tidak menghilangkan bagian dari citra tersebut (lossless compression). Biasanya citra berformat PNG merupakan salah

satu format yang baik untuk digunakan pengolahan citra karena format ini tidak menghilangkan bagian dari citra yang sedang diolah.

2.3 Analisis Operasi Pixel Dan Penerapan Metode Histogram Dalam Pengolahan Citra Digital

2.3.1 Meningkatkan Kecerahan Pada Operasi Pixel

Meningkatkan kecerahan pada citra digital adalah salah satu operasi dasar dalam pengolahan citra, yang bertujuan untuk memperbaiki gambar yang diambil dalam kondisi pencahayaan yang kurang. Kecerahan diukur dari intensitas setiap piksel dalam gambar, di mana intensitas lebih tinggi menghasilkan piksel yang lebih terang. Dalam gambar grayscale, intensitas berkisar antara 0 (hitam) hingga 255 (putih). Ketika nilai intensitas ini ditambah, gambar menjadi lebih terang secara keseluruhan. Ini membantu dalam meningkatkan visibilitas detail yang mungkin tersembunyi pada gambar asli, terutama jika gambar diambil dalam kondisi kurang cahaya.

Namun, meningkatkan kecerahan tidak selalu memberikan hasil yang diinginkan. Jika dilakukan secara berlebihan, gambar bisa mengalami overexposure, di mana bagian-bagian gambar yang sangat terang kehilangan detail karena nilainya mendekati 255 (putih). Oleh karena itu, operasi ini harus dilakukan dengan hati-hati untuk menjaga keseimbangan antara area terang dan gelap dalam gambar. Dalam beberapa kasus, peningkatan kecerahan dikombinasikan dengan teknik lain, seperti peregangan kontras atau penajaman, untuk mendapatkan hasil yang lebih optimal.

A. Negasi

Negasi dalam pengolahan citra adalah proses pembalikan nilai intensitas piksel

dalam sebuah gambar. Dalam gambar grayscale, piksel dengan nilai intensitas tinggi diubah menjadi rendah dan sebaliknya. Sebagai contoh, sebuah piksel dengan intensitas 250 yang hampir putih akan diubah menjadi 5 yang hampir hitam. Hasil dari operasi negasi adalah gambar yang menyerupai negatif foto film, di mana area yang terang menjadi gelap dan area yang gelap menjadi terang. Teknik ini sering digunakan untuk meningkatkan visibilitas bagian tertentu dari gambar yang mungkin tersembunyi dalam tampilan normal.

Negasi juga sangat bermanfaat dalam bidang tertentu seperti pencitraan medis dan astronomi, di mana kontras antara objek dan latar belakang sangat penting. Dengan menggunakan negasi, fitur-fitur halus yang sulit terlihat pada gambar asli dapat lebih mudah diidentifikasi. Selain itu, negasi dapat digunakan sebagai langkah awal dalam serangkaian operasi pengolahan

citra yang lebih kompleks, seperti deteksi tepi atau segmentasi citra, karena negasi mempermudah pemisahan objek dari latar belakang.

B. Peregangan Kontras

Peregangan kontras adalah teknik yang bertujuan untuk memperbaiki citra dengan memperluas rentang dinamis dari intensitas piksel dalam gambar. Pada gambar dengan kontras rendah, nilai intensitas piksel terkonsentrasi dalam rentang yang sempit, menghasilkan gambar yang tampak kusam dan tidak memiliki perbedaan yang jelas antara area terang dan gelap. Peregangan kontras bekerja dengan memetakan nilai intensitas awal menjadi nilai yang lebih besar, sehingga area gelap menjadi lebih gelap dan area terang menjadi lebih terang. Hal ini memperluas rentang dinamis citra, sehingga detail pada gambar lebih mudah terlihat.

Proses peregangan kontras dapat dilakukan secara linear atau non-linear tergantung pada kebutuhan. Peregangan linear biasanya lebih sederhana dan mudah diterapkan, tetapi dalam beberapa kasus, peregangan non-linear lebih efektif untuk meningkatkan kontras pada bagian-bagian tertentu dari citra tanpa memperburuk bagian lain. Peregangan kontras banyak digunakan dalam aplikasi seperti pengolahan citra medis, pencitraan satelit, dan fotografi, di mana kontras yang baik sangat penting untuk mengungkap detail kritis dalam gambar.

C. Pengirisan Intensitas

Pengirisan intensitas adalah teknik pengolahan citra yang melibatkan pemisahan nilai intensitas dalam citra menjadi beberapa kisaran atau "iris." Teknik ini memungkinkan kita untuk menyoroti rentang nilai intensitas tertentu yang relevan dengan tujuan analisis. Misalnya, dalam citra medis, pengirisan intensitas dapat digunakan untuk mengekstraksi jaringan atau struktur tertentu berdasarkan intensitas cahayanya dalam hasil pemindaian, seperti MRI atau CT Scan. Dengan mengisolasi intensitas ini, dokter atau analis dapat lebih fokus pada area yang relevan.

Pengirisan intensitas juga digunakan dalam pencitraan multispektral, seperti pencitraan satelit, di mana rentang intensitas tertentu mewakili tipe lahan, vegetasi, atau struktur. Dengan cara ini, teknik pengirisan intensitas dapat membantu dalam klasifikasi dan interpretasi data citra secara lebih efisien. Selain itu, teknik ini juga sering digunakan dalam aplikasi visualisasi citra, di mana berbagai intensitas dapat diberikan warna yang berbeda untuk mempermudah interpretasi visual.

D. Penajaman Lokal

Penajaman lokal bertujuan untuk meningkatkan detail atau tepi pada bagian tertentu dari citra tanpa mempengaruhi keseluruhan gambar. Teknik ini bekerja dengan menekankan perbedaan intensitas antara piksel yang

berdekatan, khususnya di daerah di mana terdapat perubahan besar dalam intensitas, seperti tepi objek. Dengan penajaman lokal, area tepi objek menjadi lebih jelas dan tajam, sehingga detail halus yang mungkin kabur pada gambar asli menjadi lebih terlihat. Penajaman ini biasanya diterapkan dengan metode filter, seperti filter laplacian atau unsharp masking.

Dalam dunia fotografi digital dan pengolahan citra medis, penajaman lokal sangat penting untuk meningkatkan kualitas gambar tanpa menambahkan noise atau artefak di seluruh citra. Sebagai contoh, dalam pengolahan citra medis, penajaman lokal dapat digunakan untuk menonjolkan perbedaan antara jaringan yang normal dan yang abnormal. Demikian pula, dalam fotografi, teknik ini dapat digunakan untuk menonjolkan detail seperti tekstur kulit, rambut, atau objek kecil lainnya dalam sebuah gambar, memberikan kesan visual yang lebih menarik.

E. Algoritma Penghitungan Histogram

Algoritma penghitungan histogram digunakan untuk menghitung distribusi nilai intensitas piksel dalam sebuah gambar. Histogram ini memberikan gambaran grafis tentang bagaimana intensitas piksel tersebar di seluruh gambar, apakah gambar memiliki lebih banyak piksel terang, gelap, atau seimbang. Misalnya, dalam sebuah gambar dengan kontras rendah, histogramnya akan menunjukkan bahwa sebagian besar nilai piksel terkonsentrasi dalam kisaran sempit, sedangkan gambar dengan kontras tinggi akan menunjukkan distribusi yang lebih luas. Algoritma ini sangat berguna untuk analisis citra karena memberikan informasi tentang kualitas gambar secara keseluruhan.

Dengan mengetahui histogram sebuah gambar, kita dapat menentukan tindakan apa yang harus diambil untuk memperbaikinya, seperti meningkatkan kontras, ekualisasi histogram, atau penerapan filter lain. Histogram juga sering digunakan dalam deteksi tepi dan segmentasi gambar, di mana distribusi nilai intensitas dapat membantu memisahkan objek dari latar belakang. Dalam dunia fotografi digital, histogram juga sering digunakan sebagai alat untuk membantu fotografer mengevaluasi eksposur dan kualitas gambar yang diambil.

F. Ekualisasi Histogram

Ekualisasi histogram adalah metode yang bertujuan untuk meningkatkan kontras dalam gambar dengan mendistribusikan ulang nilai intensitas piksel sehingga histogram gambar lebih merata. Pada gambar dengan kontras rendah, nilai intensitas piksel cenderung terkonsentrasi dalam rentang sempit, sehingga gambar tampak kusam dan detail sulit terlihat. Dengan ekualisasi histogram, nilai intensitas ini diubah sehingga piksel tersebar

lebih merata di seluruh rentang intensitas, menghasilkan gambar dengan kontras yang lebih tinggi dan detail yang lebih jelas.

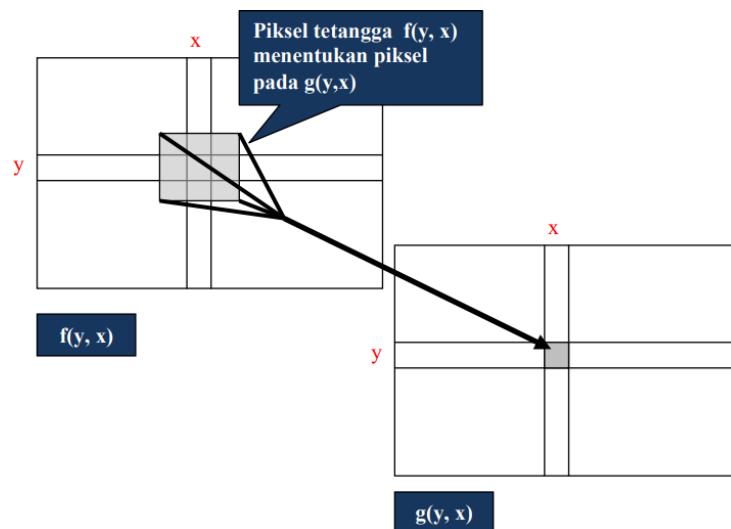
Metode ekualisasi histogram sering digunakan dalam aplikasi yang membutuhkan peningkatan visibilitas detail, seperti pengolahan citra medis dan penglihatan mesin. Sebagai contoh, dalam citra medis seperti sinar X atau CT scan, ekualisasi histogram dapat membantu meningkatkan visibilitas jaringan atau struktur yang mungkin tersembunyi dalam gambar asli. Teknik ini juga banyak digunakan dalam aplikasi pemrosesan citra otomatis, di mana peningkatan kontras sangat penting untuk deteksi objek atau fitur dalam gambar.

2.4 Operasi Ketetanggaan Pixel

2.4.1 Pengertian Operasi Ketetanggaan Pixel

Operasi ketetanggaan piksel adalah operasi pengolahan citra untuk mendapatkan nilai suatu piksel yang melibatkan nilai piksel-piksel tetangganya. Hal ini didasarkan kenyataan bahwa setiap piksel pada umumnya tidak berdiri sendiri, melainkan terkait dengan piksel tetangga, karena merupakan bagian suatu objek tertentu di dalam citra. Sifat inilah yang kemudian mendasari timbulnya algoritma untuk mengolah setiap piksel citra melalui piksel-piksel tetangga. Sebagai contoh, suatu citra yang berderau dapat dihaluskan melalui pererataan atas piksel-piksel tetangga.

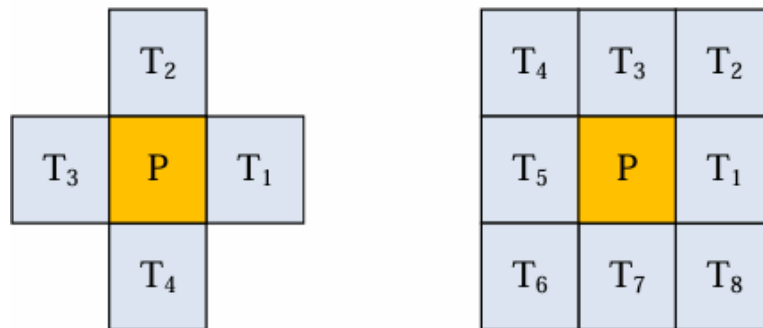
Gambar 2.1 memberikan ilustrasi operasi ketetanggaan piksel. Delapan piksel tetangga terdekat dengan piksel $f(y,x)$ digunakan untuk memperbaikinya menjadi $g(y,x)$ di tempat yang sama



*Gambar 2.1 Operasi ketetanggaan piksel.
Sejumlah tetangga menentukan nilai sebuah piksel*

2.4.2 Pengertian Ketetanggaan Pixel

Pada pengolahan citra, ketetanggaan piksel banyak dipakai terutama pada analisis bentuk objek. Ketetanggaan piksel yang umum dipakai adalah 4- ketetanggaan dan 8-ketetanggaan. Untuk memahami dua jenis ketetanggaan piksel, lihat Gambar 2.2.



Gambar 2.2 Dua macam ketetanggaan piksel.

Pada 4-ketetanggaan, T1, T2, T3, dan T4 merupakan tetangga terdekat piksel P. Pada 8-ketetanggaan, tetangga piksel P yaitu piksel-piksel yang berada di sekitar P. Totalnya sebanyak 8 buah. Bila P mempunyai koordinat (b, k) dengan b baris dan k kolom, hubungan piksel tetangga terhadap P sebagai berikut.

- Pada 4-ketetanggaan

$$T1 = (b, k + 1), T2 = (b - 1, k), T3 = (b, k - 1), T4 = (b + 1, k) \quad (2.1)$$

- Pada 8-ketetanggaan

$$\begin{aligned} T1 &= (b, k + 1), T2 = (b - 1, k - 1), \\ T3 &= (b, k - 1), T4 = (b - 1, k - 1) \end{aligned} \quad (2.2) \quad T5$$

$$= (b, k-1), T6 = (b+1, k-1),$$

$$T7 = (b+1, k-1), T8 = (b+1, k+1)$$

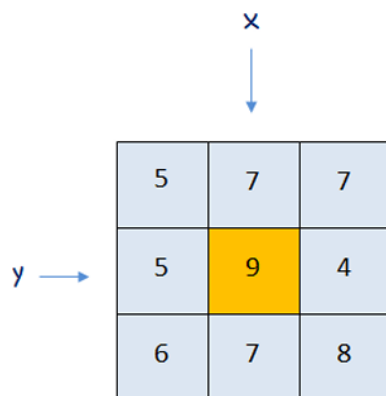
2.4.3 Aplikasi Ketetanggaan Pixel pada Filter

Ada tiga jenis filter yang menggunakan operasi ketetanggaan piksel yang akan dibahas sebagai pengantar pada bab ini. Ketiga filter tersebut adalah filter batas, filter pererataan, dan filter median. Sebagai filter atau tapis, operasi ketetanggaan piksel berfungsi untuk menyaring atau paling tidak mengurangi gangguan atau penyimpangan pada citra.

a). Filter Bebas

Filter batas adalah filter yang dikemukakan dalam Davies (1990). Idennya adalah mencegah piksel yang intensitasnya di luar intensitas piksel- piksel tetangga. Algoritma yang digunakan untuk keperluan ini dapat dilihat berikut ini.

Sebagai contoh, terdapat piksel seperti terlihat pada Gambar 2.3.



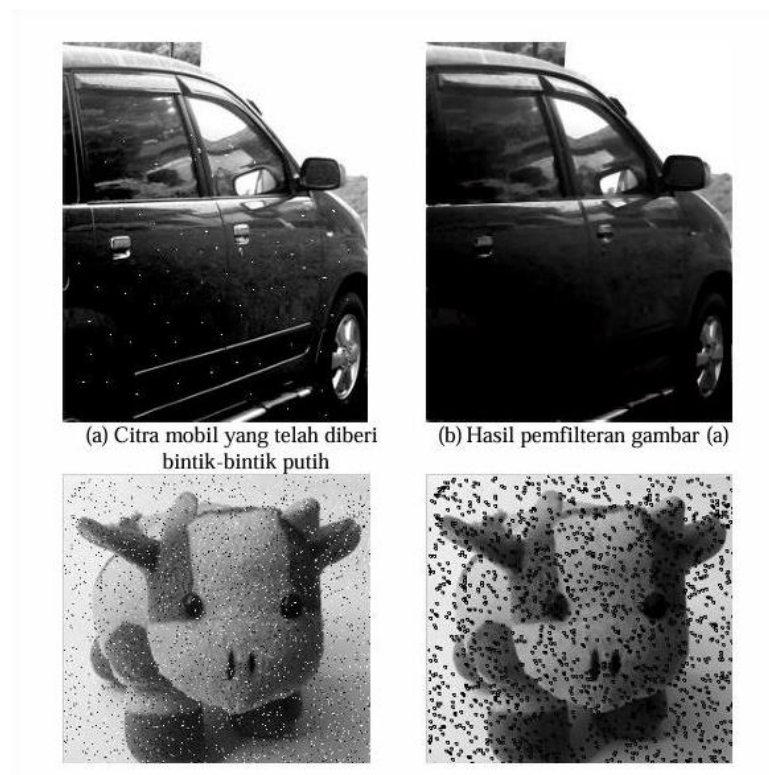
Gambar 2.3 Contoh piksel dan tetangga

Berdasarkan keadaan tersebut,

- $\text{minInt} = \text{minimum}(5, 7, 7, 5, 4, 6, 7, 8) = 4;$
- $\text{maksInt} = \text{maksimum}(5, 7, 7, 5, 4, 6, 7, 8) = 8;$

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

seandainya $f(y, x)$ pada keadaan di atas bernilai 2 (bukan 9), $g(y, x)$ akan bernilai 4. Untuk melihat efek filter batas, cobalah program berikut. Perlu diketahui, pemrosesan hanya dilakukan selain baris pertama, baris terakhir, kolom pertama, dan kolom terakhir. Keempat area tersebut tidak diproses karena tidak mempunyai tetangga yang lengkap (sebanyak 8).



Gambar 2.4 Efek filter batas terhadap citra yang mengandung derau

Terlihat bahwa bintang-bintang putih pada citra mobil.png dapat dihilangkan. Namun, kalau diperhatikan dengan saksama, operasi tersebut juga mengaburkan citra. Pada citra boneka2.png, derau malah diperkuat. Artinya, filter itu tidak cocok

digunakan untuk menghilangkan jenis derau yang terdapat pada citra tersebut.

b). Filter Pererataan

Filter pererataan (Costa dan Cesar, 2001) dilakukan dengan menggunakan rumus:

$$g(y, x) = \frac{1}{9} \sum_{p=-1}^1 \sum_{q=-1}^1 f(y+p, x+q) \quad (2.3)$$

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

65	50	55
78	68	60
60	60	62

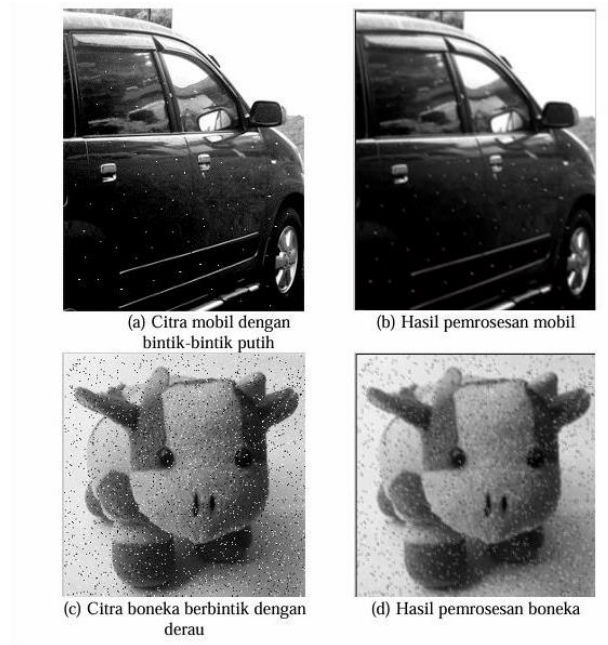
Pada contoh di atas, yang diarsir (yaitu yang bernilai 68) merupakan nilai pada $f(y, x)$. Nilai rerata pengganti untuk $g(y, x)$ dihitung dengan cara seperti berikut:

$$g(y, x) = \frac{1}{9} \times (65+50+55+76+68+60+60+60+62) = 61,7778$$

$$\cong 62$$

Jadi, nilai 68 pada $f(y, x)$ diubah menjadi 62 pada $g(y, x)$.

Gambar 2.5 menunjukkan efek pemrosesan dengan filter pererataan. Dibandingkan dengan filter batas, hasil pemrosesan filter pererataan tidak menghilangkan bintik-bintik putih pada citra mobil, tetapi hanya agak menyamarkan. Pada citra boneka2.png, derau lebih dihaluskan.



Gambar 2.5 Contoh penerapan filter pererataan

CATATAN:

Perhatikan hasil pemrosesan filter pada Gambar 4.5(b) dan Gambar 4.5(d). Terlihat keberadaan garis pada kolom pertama dan baris pertama. Untuk menghindari efek seperti itu, baris pertama, kolom pertama, baris terakhir, dan kolom terakhir perlu dihilangkan. Jadi, efek “bingkai” dihilangkan dengan memperkecil ukuran citra menjadi $(N-2) \times (N-2)$ jika ukuran citra semula adalah $N \times N$.

c). Filter Median

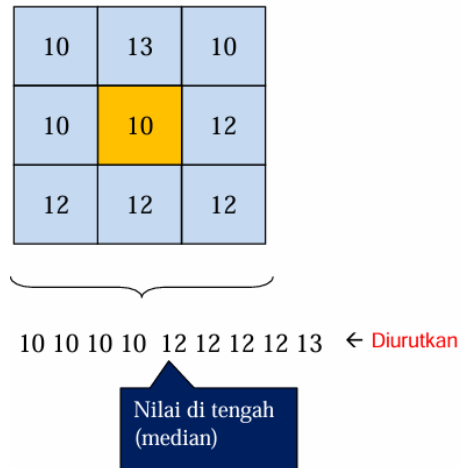
Filter median sangat populer dalam pengolahan citra. Filter ini dapat dipakai untuk menghilangkan derau bintik-bintik. Nilai yang lebih baik digunakan untuk suatu piksel ditentukan oleh nilai median dari setiap piksel dan kedelapan piksel tetangga pada 8-ketetanggaan. Secara matematis, filter dapat dinotasikan seperti berikut:

$$g(y,x) = \text{median} ($$

$$f(y-1,x-1), f(y-1,x), f(y-1,x+1), 2.4)$$

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

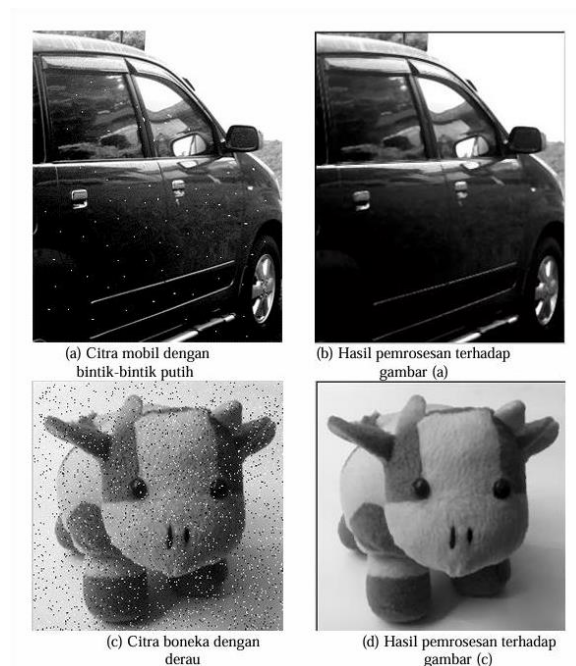
Contoh untuk satu piksel ditunjukkan pada Gambar 2.6.



Gambar 2.6 Gambaran operasi penggunaan filter median

Pada contoh di atas terlihat bahwa untuk mendapatkan median, diperlukan pengurutan (sorting) terlebih dulu.

Contoh hasil penggunaan filter median dapat dilihat pada Gambar 4.7.

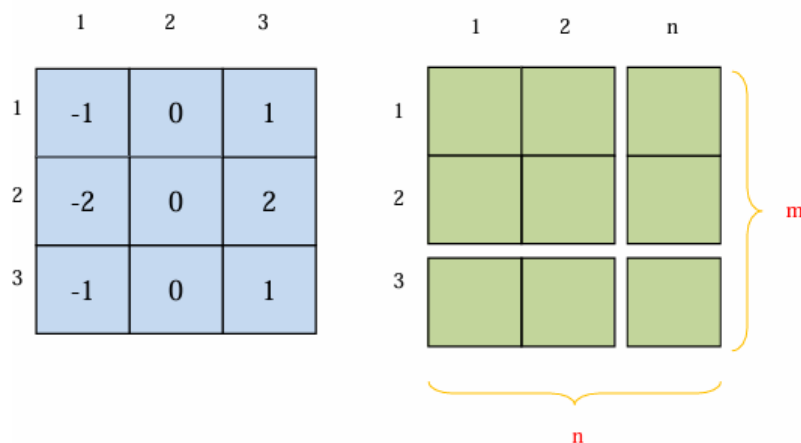


Gambar 2.7 Contoh penerapan filter median

Hasilnya terlihat bahwa derau dapat dihilangkan, tetapi detail pada citra tetap dipertahankan. Namun, hal ini tentu saja didapat dengan tambahan beban komputasi “pengurutan”.

2.4.4 Pengertian Konvolusi

Konvolusi seringkali dilibatkan dalam operasi ketetanggaan piksel. Konvolusi pada citra sering disebut sebagai konvolusi dua-dimensi (konvolusi 2D). Konvolusi 2D didefinisikan sebagai proses untuk memperoleh suatu piksel didasarkan pada nilai piksel itu sendiri dan tetangganya, dengan melibatkan suatu matriks yang disebut kernel yang merepresentasikan pembobotan. Wujud kernel umumnya bujur sangkar, tetapi dapat pula berbentuk persegi panjang. Gambar 2.8 menunjukkan contoh kernel untuk konvolusi.



Gambar 2.8 Contoh kernel untuk konvolusi berukuran 3 x 3 dan m x n

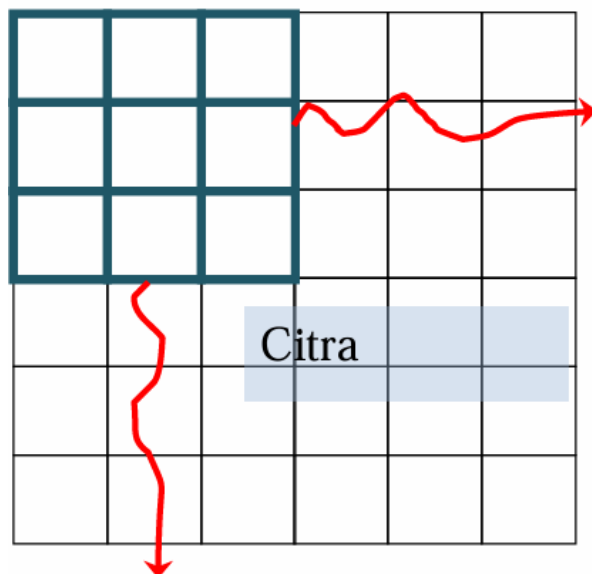
Kernel konvolusi terkadang disebut dengan istilah cadar, cadar konvolusi, atau cadar spasial.

Secara umum, proses penapisan di kawasan ruang (space domain), sebagai alternatif di kawasan frekuensi, dilaksanakan melalui operasi konvolusi. Operasi ini dilakukan dengan menumpangkan suatu jendela (kernel) yang berisi angka angka pengali pada setiap piksel yang ditimpali. Kemudian, nilai rerata diambil dari hasil- hasil kali tersebut. Khusus bila angka-angka pengali tersebut semua adalah 1, hasil yang

didapat sama saja dengan filter pererataan.

Pada pelaksanaan konvolusi, kernel digeser sepanjang baris dan kolom dalam citra (lihat Gambar 2.9) sehingga diperoleh nilai yang baru pada citra keluaran.

Kernel digerakkan di sepanjang baris dan kolom



Gambar 4.9 Konvolusi dilakukan dengan melakukan proses di sepanjang kolom dan baris pada citra

Bagaimana konvolusi dilakukan? Prosesnya dirumuskan sebagai berikut:

$$g(y, x) = \sum_{p=-m_2}^{m_2} \sum_{q=-n_2}^{n_2} h(p + m_2 + 1, q + n_2 + 1) f(y - p, x - q)$$

Dalam hal ini,

- m_2 adalah separuh dari tinggi kernel ($m_2 = \text{floor}(m/2)$),
- n_2 adalah separuh dari lebar kernel ($n_2 = \text{floor}(n/2)$),
- floor menyatakan pembulatan ke bawah, dan h menyatakan kernel, dengan indeks dimulai dari 1.

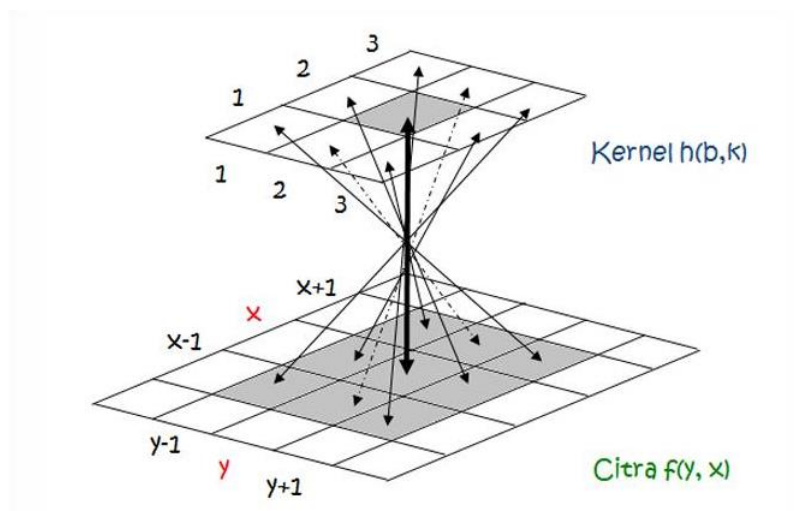
Catatan



Apabila kernel h diputar sebesar 180° (dapat dilaksanakan dengan perintah $k = \text{rot90}(h)$), perhitungan $g(y,x)$ dapat diperoleh melalui:

$$g(y,x) = \sum_{p=-m2}^{m2} \sum_{q=-n2}^{n2} k(p,q)f(y+p,x+q)$$

Ilustrasi konvolusi dijelaskan melalui contoh pada Gambar 2.10.



Gambar 2.10 Contoh konvolusi

Berdasarkan Gambar 2.10, apabila citra yang berada pada jendela kernel berupa

65	50	55
78	68	60
60	60	62

Dan kernel berupa

-1	0	1
-2	0	2
-1	0	1

maka nilai piksel hasil konvolusi berupa:

$$\begin{aligned}
 g(x, y) &= -1 \times 62 + 0 \times 60 + 1 \times 60 + \\
 &\quad -2 \times 60 + 0 \times 68 + 2 \times 78 + \\
 &\quad -1 \times 55 + 0 \times 50 + 1 \times 65 \\
 &= -62 + 0 + 60 - 120 + 0 + 152 - 55 + 0 + 65 \\
 &= 40
 \end{aligned}$$

Dengan demikian, nilai 68 akan diubah menjadi 40 pada citra keluaran.

Catatan



- Apabila kernel h diputar sebesar 180° ($k = \text{rot}180(h)$), perhitungan $g(y,x)$ dapat diperoleh melalui

$$g(y, x) = \sum_{p=-m2}^{m2} \sum_{q=-n2}^{n2} h(p + m2 + 1, q + n2 + 1) f(y + p, x + q)$$

- Untuk kernel yang simetrik ($k(y,x)=k(x,y)$), proses konvolusi sama dengan tidak melalui konvolusi (yaitu korelasi).

Berikut adalah salah satu algoritma yang dipakai untuk mengimplementasikan konvolusi pada citra, dengan asumsi kernel mempunyai jumlah baris dan kolom bernilai ganjil.

Contoh pemakaian fungsi konvolusi ditunjukkan berikut ini.


```
>> H = [-1 0 -1; 0 4 0; -1 0 -1]; □
```

```
>> F = imread('C:\Image\gedung.png'); □
```

```
>> K = konvolusi (F, H); □
```

Pertama-tama, kernel H

ditentukan melalui $H = \begin{bmatrix} -1 & 0 & -1 \\ 0 & 4 & 0 \\ -1 & 0 & -1 \end{bmatrix}$

1; 0 4 0; -1 0 -1];

Kernel di atas dinamakan “Quick Mask” (Phillips, 2000) dan berguna untuk deteksi tepi.

Selanjutnya, citra gedung.png dibaca dan diletakkan di F. Lalu, konvolusi dilaksanakan dengan memanggil fungsi konvolusi. Dengan cara seperti itu, K berisi hasil konvolusi citra F dan kernel H. Nilai K dapat dilihat secara sekilas dengan mengetikkan

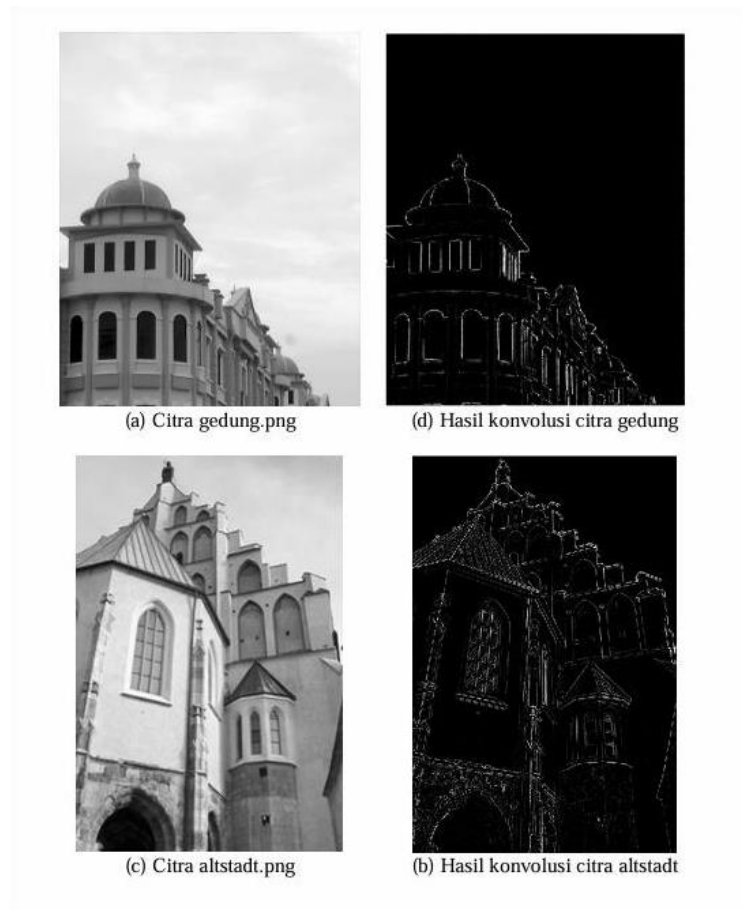
```
>> K
```

Nilai yang dihasilkan dengan konvolusi dapat bernilai negatif dan bahkan dapat melebihi nilai 255. Oleh karena itu, pemrosesan konvolusi harus dilaksanakan dengan menggunakan presisi ganda (bukan bilangan bulat). Lalu, setelah semua citra diproses dengan konvolusi, perlu dilakukan pengaturan nilai piksel agar berada pada jangkauan [0, 255]. Nilai yang kurang dari 0 diubah menjadi 0 dan yang melebihi 255 diubah menjadi 255. Fungsi uint8 dapat digunakan untuk kepentingan tersebut Contoh:

```
>> K2 = uint8(K); □
```

Dengan cara seperti itu, nilai pada K2 berada pada jangkauan [0, 255]. Citra K2 dapat ditampilkan dengan menggunakan **imshow**.

Gambar 2.11 memperlihatkan contoh citra asli (tersimpan dalam F) dan citra yang telah mengalami konvolusi dan telah diatur agar bernilai dalam jangkauan [0, 255] (tersimpan dalam K2).



Gambar 2.11 Contoh penerapan konvolusi

Contoh di atas menunjukkan aplikasi konvolusi yang dapat digunakan untuk mendapatkan tepi objek. Namun, tentu saja aplikasi konvolusi tidak hanya untuk kepentingan seperti itu. Untuk memperlihatkan hasil deteksi tepi objek, nilai-nilai piksel hasil perlu dinaikkan yaitu ditambah 127.

2.5 Operasi Aritmatika Citra

2.5.1 Operasi Aritmatika Citra

Operasi aritmatika pada citra digital melibatkan manipulasi nilai intensitas piksel dalam citra, di mana setiap piksel direpresentasikan sebagai nilai numerik. Operasi ini mencakup penjumlahan, pengurangan, perkalian, pembagian, dan pencampuran citra. Setiap operasi memiliki tujuan dan fungsi spesifik dalam pengolahan citra, yang dapat diterapkan untuk meningkatkan kualitas citra, menggabungkan citra, hingga mengurangi noise.

1. Penjumlahan Citra

Operasi penjumlahan citra (image addition) dilakukan dengan menjumlahkan nilai piksel dari dua citra. Hasilnya akan menghasilkan citra baru dengan intensitas yang lebih tinggi atau cerah. Operasi ini sering digunakan dalam penggabungan citra atau untuk memperbaiki kecerahan citra. Misalnya, citra yang memiliki kekurangan cahaya dapat diperbaiki dengan menambahkan citra lain yang lebih terang.

Contoh

Citra 1:

```
100 150 200
50 100 150
0 50 100
```

Citra 2:

```
50 50 50
100 100 100
150 150 150
```

Hasil Penjumlahan:

```
150 200 250
150 200 250
150 200 250
```

Penjelasan:

Setiap piksel dari citra pertama dijumlahkan dengan piksel dari citra kedua yang berada di posisi yang sama. Jika hasil penjumlahan lebih besar dari 255, biasanya dipotong (clipping) menjadi 255 (karena rentang nilai piksel adalah 0-255 untuk citra 8-bit).

Clipping Diperlukan: Saat menjumlahkan dua citra, nilai piksel hasil penjumlahan bisa melebihi batas maksimum (biasanya 255 untuk citra 8-bit). Jika hasil penjumlahan suatu piksel adalah 300, nilai tersebut perlu di-"clip" menjadi 255.

2. Pengurangan Citra

Operasi pengurangan citra (image subtraction) berguna untuk mendeteksi perbedaan antara dua citra, seperti mendeteksi perubahan pada dua gambar yang diambil pada waktu yang berbeda. Dalam aplikasi medis, ini dapat digunakan untuk mendeteksi perbedaan pada gambar radiologi

sebelum dan sesudah perawatan.

Contoh

Citra 1:

```
100 150 200
50 100 150
0 50 100
```

Citra 2:

```
50 50 50
100 100 100
150 150 150
```

Hasil Pengurangan:

```
50 100 150
0 0 50
0 0 0
```

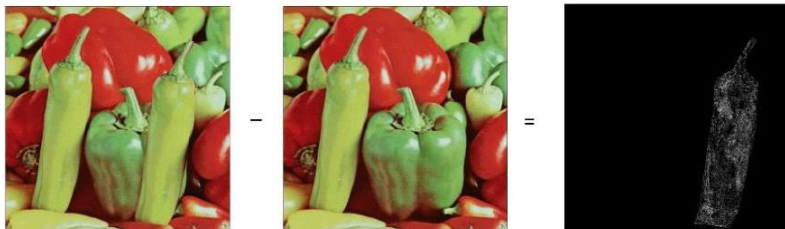
Penjelasan:

Setiap piksel dari citra pertama dikurangi dengan piksel dari citra kedua. Jika hasil pengurangan lebih kecil dari 0, hasilnya dipotong menjadi 0, karena piksel tidak bisa bernilai negatif.

Clipping Diperlukan: Saat mengurangi nilai piksel, hasilnya bisa menjadi negatif.

Oleh karena itu, hasil yang kurang dari 0 harus di-"clip" ke 0.

Contoh gambar:



3. Perkalian Citra

Operasi perkalian citra (image multiplication) biasanya digunakan untuk mengubah kontras atau menggabungkan dua citra. Perkalian antara dua citra dapat meningkatkan intensitas piksel tertentu sambil mempertahankan

nilai yang lebih rendah pada bagian lainnya.

Contoh

Citra 1:

```
100 150 200
50 100 150
0 50 100
```

Citra 2:

```
2 2 2
2 2 2
2 2 2
```

Hasil Perkalian:

```
200 255 255
100 200 255
0 100 200
```

Penjelasan:

Setiap nilai piksel pada citra pertama dikalikan dengan nilai piksel pada citra kedua. Clipping Dapat Dihindari (Tapi Perlu Diperhatikan): Perkalian tidak secara otomatis memerlukan clipping, tetapi hasil perkalian dapat melampaui batas maksimum. Misalnya, jika piksel bernilai 200 dikalikan dengan 2, hasilnya 400, yang harus di-"clip" menjadi 255.



Citra semula



Citra setelah setiap pixel dikali dengan 3

4. Pembagian Citra

Operasi pembagian citra (image division) umumnya digunakan dalam proses normalisasi, yaitu untuk menyesuaikan intensitas piksel dalam citra

sehingga berada dalam rentang tertentu. Pembagian juga berguna dalam proses penggabungan citra dengan mempertahankan proporsi piksel dari citra asli.

Contoh

Citra 1:

100 150 200
50 100 150
10 50 100

Citra 2:

2 3 4
2 2 2
5 5 5

Hasil Pembagian:

50 50 50
25 50 75
2 10 20

Penjelasan:

Setiap nilai piksel dari citra pertama dibagi dengan nilai piksel pada citra kedua. Seperti dalam operasi lain, hasil pembagian yang lebih tinggi dari batas maksimal (255) akan di-"clip" ke 255.

Clipping Diperlukan: Saat melakukan pembagian, kita harus menghindari pembagian dengan nol dan juga perlu memastikan hasil pembagian tidak melebihi batas maksimum (meskipun dalam pembagian, biasanya hasil tidak akan lebih dari batas maksimum jika pembagi tidak nol).

5. Pencampuran Citra (Blending)

Operasi pencampuran citra (blending) melibatkan penggabungan dua citra dengan cara memberikan bobot tertentu pada setiap citra. Proses ini sering digunakan untuk membuat efek transisi halus antara dua citra, atau untuk menggabungkan dua gambar dengan properti visual yang berbeda. Dalam aplikasi desain grafis, pencampuran citra digunakan untuk membuat hasil visual yang lebih artistik.

Pencampuran citra dilakukan dengan menggunakan rumus:

Citra hasil = $\alpha \times \text{Citra 1} + (1-\alpha) \times \text{Citra 2}$ Di mana α

adalah koefisien pencampuran yang menentukan kontribusi dari masing-masing citra.

Penjelasan Nilai α Rentang Nilai α :

- $\alpha=0$: Hasil akhir sepenuhnya akan berasal dari Citra 2. Dalam hal ini, citra hasil hanya akan menampilkan informasi dari Citra 2.
- $\alpha=1$: Hasil akhir sepenuhnya akan berasal dari Citra 1. Citra hasil hanya akan menampilkan informasi dari Citra 1.
- $0<\alpha<1$: Citra hasil adalah kombinasi dari kedua citra. Nilai α yang lebih tinggi memberikan lebih banyak bobot kepada Citra 1, sedangkan nilai $1-\alpha$ memberikan bobot lebih besar kepada Citra 2.

Contoh:

Jika $\alpha=0.5$, maka kontribusi kedua citra adalah 50% masing-masing.

Citra 1:

100	150	200
50	100	150
0	50	100

Citra 2:

200	200	200
100	100	100
50	50	50

Langkah Pencampuran:

Posisi (1,1): $0.5 \times 100 + 0.5 \times 200 = 150$
asil akhir:

50	175	200
75	100	125
25	50	75

Penjelasan:

- Nilai α Harus Ditetapkan: Koefisien α harus dipilih dengan

tepat tergantung pada seberapa besar proporsi masing-masing citra yang ingin dicampurkan.

- Normalisasi Piksel: Seperti operasi lainnya, jika hasil pencampuran melebihi batas maksimal (255 untuk citra 8-bit), hasil tersebut akan dipotong (clipped) ke 255.
- Clipping Diperlukan: Pada operasi pencampuran, hasil juga dapat melebihi batas maksimum atau menjadi kurang dari 0, tergantung pada nilai α .
- Contoh: Hasil pencampuran dari dua piksel (misalnya 200 dan 100) dengan α yang tepat harus di-clip jika hasilnya melebihi 255.

2.5.2 Operasi Logika Citra

Operasi logika pada citra melibatkan manipulasi citra berdasarkan aturan logika biner yang diterapkan pada nilai piksel. Operasi logika biasanya digunakan dalam segmentasi citra, deteksi tepi, atau penggabungan objek dari dua gambar. Beberapa operasi logika yang umum digunakan adalah AND, OR, XOR, dan NOT.

1. Logika AND

Operasi logika AND pada citra hanya mempertahankan piksel yang memiliki nilai sama pada kedua citra yang dioperasikan. Hasil dari operasi ini akan menghasilkan citra dengan piksel yang intensitasnya lebih rendah, kecuali pada area yang sama-sama terang pada kedua citra. Operasi ini sering digunakan untuk menemukan area yang tumpang tindih dari dua citra.

Contoh

Citra Biner 1:

```
1 0 1
0 1 0
1 1 0
```

Citra Biner 2:

```
1 1 0
0 1 1
```


1 0 0

Hasil

Operas

i AND:

1 0 0

0 1 0

1 0 0

Penjelasan:

Operasi AND akan menghasilkan 1 hanya jika kedua piksel sama-sama bernilai 1.

2. Logika OR

Operasi OR menggabungkan dua citra dengan mempertahankan piksel dengan nilai intensitas tertinggi dari kedua citra. Hasil dari operasi OR adalah citra yang lebih terang. Operasi ini sering digunakan dalam penggabungan objek dari dua citra yang berbeda.

Contoh

Citra Biner 1:

1 0 1

0 1 0

1 1 0

Citra Biner 2:

1 1 0

0 1 1

1 0 0

Hasil Operasi OR:

1 1 1

0 1 1

1 1 0

Penjelasan:

Operasi OR akan menghasilkan 1 jika salah satu atau kedua piksel bernilai 1.

3. Logika XOR

Operasi XOR menghasilkan citra yang menyoroti perbedaan antara dua citra. Piksel yang memiliki nilai berbeda pada kedua citra akan diwakili dalam citra hasil. Ini berguna dalam deteksi perbedaan atau perubahan pada dua citra yang diambil pada waktu berbeda.

Contoh

Citra Biner 1:

1 0 1
0 1 0
1 1 0

Citra Biner 2:

1 1 0
0 1 1
1 0 0

Hasil Operasi XOR:

0 1 1
0 0 1
0 1 0

Penjelasan:

Operasi XOR akan menghasilkan 1 jika hanya salah satu piksel bernilai 1, bukan keduanya.

4. Logika NOT

Operasi NOT membalikkan nilai intensitas piksel pada citra, menghasilkan efek negatif. Semua piksel terang akan menjadi gelap dan sebaliknya. Operasi ini berguna dalam proses analisis kontras citra atau untuk efek visual tertentu.

Contoh

Citra Biner 1:

1 0 1
0 1 0
1 1 0

Hasil Operasi NOT:

0 1 0
1 0 1
0 0 1

Penjelasan:

Operasi NOT akan membalik nilai piksel, 1 menjadi 0 dan 0 menjadi 1.

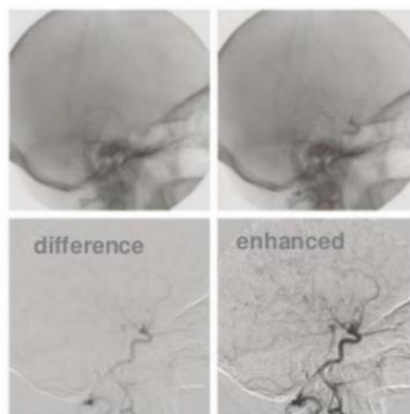
2.5.3 Aplikasi Operasi Aritmatika dan Logika pada Citra

Penggabungan antara operasi aritmatika dan logika dalam pengolahan citra memberikan berbagai solusi yang inovatif dan aplikatif dalam kehidupan sehari-hari. Beberapa contoh aplikasi dari kedua operasi ini meliputi:

1. Pengolahan Citra Medis

Dalam dunia kedokteran, operasi aritmatika dan logika digunakan untuk menganalisis gambar hasil radiologi, seperti CT scan dan MRI. Misalnya, operasi pengurangan dapat digunakan untuk membandingkan hasil sebelum dan sesudah perawatan pasien, membantu dalam mendeteksi perubahan seperti pertumbuhan tumor. Operasi AND dan XOR berguna untuk memisahkan objek tertentu dalam citra medis, misalnya jaringan yang terdeteksi dalam pemindaian.

Pada aplikasi medis, pengurangan citra berguna untuk mendeteksi kelainan pada sebuah citra medis.



2. Pengenalan Wajah

Dalam aplikasi keamanan dan identifikasi, penggabungan operasi logika seperti XOR dan AND dapat digunakan untuk mencocokkan citra wajah dengan database wajah yang tersimpan. Proses ini memungkinkan deteksi yang lebih akurat dengan menyoroti fitur unik dari setiap wajah.

3. Deteksi Cacat Produk di Industri

Operasi pengurangan dan XOR sangat penting dalam proses kontrol kualitas di pabrik, di mana dua citra dari produk yang sama dibandingkan untuk menemukan cacat atau perbedaan kecil. Sistem ini dapat digunakan untuk secara otomatis menyoroti bagian produk yang memiliki kerusakan atau kesalahan dalam proses manufaktur.

4. Sistem Pengawasan Video

Operasi logika seperti XOR sering diterapkan dalam sistem pengawasan untuk mendeteksi gerakan atau perubahan pada video. Dengan membandingkan frame video secara berurutan, sistem dapat dengan mudah mendeteksi gerakan, memicu alarm, atau merekam aktivitas mencurigakan.

2.6 Operasi Geometri Citra

2.6.1 Pengertian Operasi Geometri pada Citra

Salah satu cabang utama pengolahan gambar digital adalah operasi geometri, yang berfokus pada mengubah bentuk, orientasi, dan ukuran objek di dalam gambar. Dalam operasi ini, posisi piksel di dalam gambar dimanipulasi **tanpa mengubah konten** atau **intensitas piksel**. Operasi geometri sangat penting dalam berbagai bidang, mulai dari analisis visual hingga pengenalan pola, serta dalam aplikasi pengolahan gambar dalam waktu nyata seperti robotika dan augmented reality.

Operasi geometri umumnya melibatkan beberapa transformasi dasar: translasi, rotasi, refleksi, dan penskalaan. Tujuan utama dari semua operasi ini adalah mengubah tata letak atau orientasi gambar agar lebih sesuai dengan kebutuhan pengguna atau aplikasi yang sedang digunakan. Selain itu, berbagai bidang seperti **penglihatan komputer** (computer vision), **grafis komputer**, **fotografi digital**, dan **pengenalan objek** sering menggunakan operasi geometri.

Geometri dapat digunakan bersama dengan operasi lainnya, seperti morfologi atau pemfilteran, untuk menghasilkan transformasi tambahan dalam sistem pengolahan gambar yang lebih kompleks. Misalnya, dalam pengenalan wajah, setelah translasi dan rotasi untuk memusatkan wajah, operasi tambahan seperti segmentasi atau deteksi tepi dapat digunakan untuk analisis yang lebih dalam.

2.6.2 Translasi

Translasi adalah salah satu operasi dasar dalam pengolahan citra yang memainkan peran penting dalam memanipulasi gambar dengan cara memindahkan semua piksel dalam citra ke lokasi baru sesuai dengan jarak tertentu dalam arah yang diinginkan.

Proses translasi bertujuan untuk menggeser objek dalam citra tanpa mengubah orientasi, bentuk, atau ukuran objek tersebut. Pergeseran citra ke arah mendatar atau vertikal dapat dilakukan dengan persamaan:

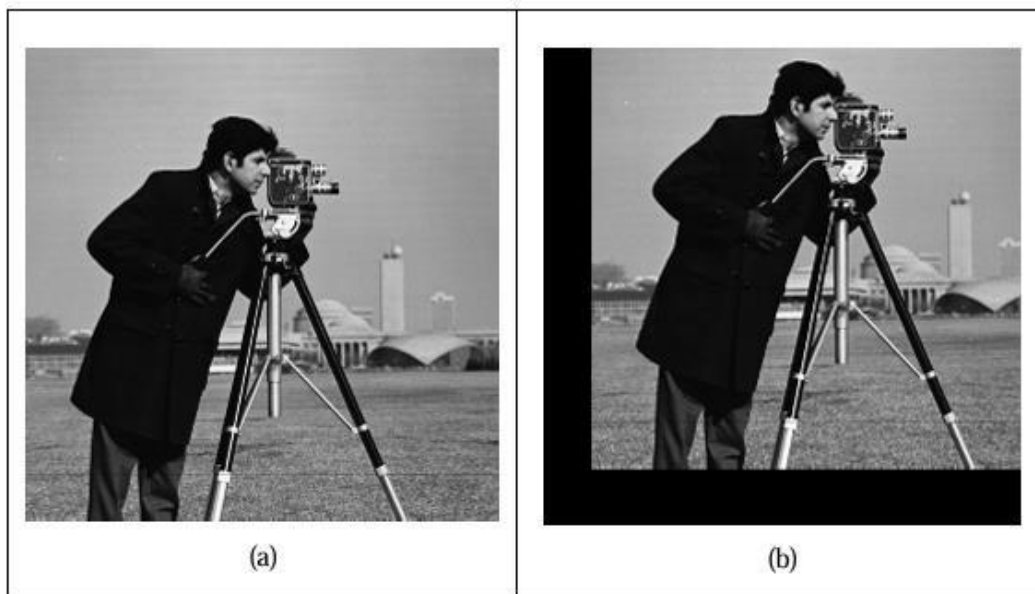
$$\begin{aligned}x_{baru} &= x_{lama} + S_x \\ y_{baru} &= y_{lama} + S_y\end{aligned}$$

Catatan:

Rumus tersebut digunakan untuk **menggeser posisi** sebuah titik atau piksel pada gambar:

- $x_{baru} = x_{lama} + S_x \rightarrow$ artinya kita menggeser posisi **horizontal** (sumbu-x) titik lama dengan menambahkan jarak geser S_x .
- $y_{baru} = y_{lama} + S_y \rightarrow$ artinya kita menggeser posisi **vertikal** (sumbu-y) titik lama dengan menambahkan jarak geser S_y .
- Jika S_x positif, titik bergeser ke kanan, kalau negatif, bergeser ke kiri.
- Jika S_y positif, titik bergeser ke atas, kalau negatif, bergeser ke bawah.

Jadi, translasi adalah seperti **memindahkan** suatu objek ke tempat baru tanpa merubah bentuk atau ukurannya, hanya dengan menggesernya ke arah tertentu.



Pada contoh (a) merupakan Citra Asli, sedangkan (b) merupakan hasil Translasi

Meskipun translasi adalah operasi yang relatif sederhana, ada beberapa tantangan yang perlu diperhatikan, terutama dalam hal penanganan piksel yang bergerak keluar dari batas citra. Berikut adalah beberapa tantangan yang sering dihadapi:

1. **Piksel yang Jatuh di Luar Batas Citra:** Salah satu masalah utama dalam translasi adalah ketika piksel-piksel yang dipindahkan keluar dari batas citra asli. Ketika objek digeser terlalu jauh, beberapa piksel mungkin berada di luar bingkai gambar, yang berarti bahwa informasi tersebut hilang dari citra yang ditampilkan.
2. **Pembuangan Piksel:** Piksel yang tidak termasuk dalam bingkai gambar dapat dibuang, yang dapat menghilangkan informasi penting, terutama jika objek tidak terlihat sepenuhnya dalam gambar.
3. **Mengisi Ruang Kosong:** dapat mengisi ruang kosong yang ditinggalkan oleh piksel yang dipindahkan dengan menggunakan nilai latar belakang atau teknik interpolasi dari piksel di sekitarnya. Misalnya, Anda dapat menggunakan warna putih, hitam, atau warna yang sesuai dengan objek di sekitarnya untuk memberi kesan yang lebih halus dan konsisten.

2.6.3 Rotasi

Rotasi adalah transformasi geometri yang mengubah orientasi objek dalam citra dengan cara memutar piksel-pikselnya di sekitar titik pusat (atau titik acuan lainnya) dengan sudut tertentu. Operasi rotasi umum digunakan dalam berbagai aplikasi seperti **fotografi digital**, **pemetaan**, dan **pengolahan citra medis**. Dalam rotasi, posisi relatif antar piksel tetap

terjaga, sementara sudut orientasi dari objek dalam citra berubah.
Operasi rotasi citra dapat dilakukan dengan fungsi cosinus-sinus, sebagai berikut:

$$x_{baru} = x * \cos(\theta) + y * \sin(\theta)$$

$$y_{baru} = y * \cos(\theta) - x * \sin(\theta)$$

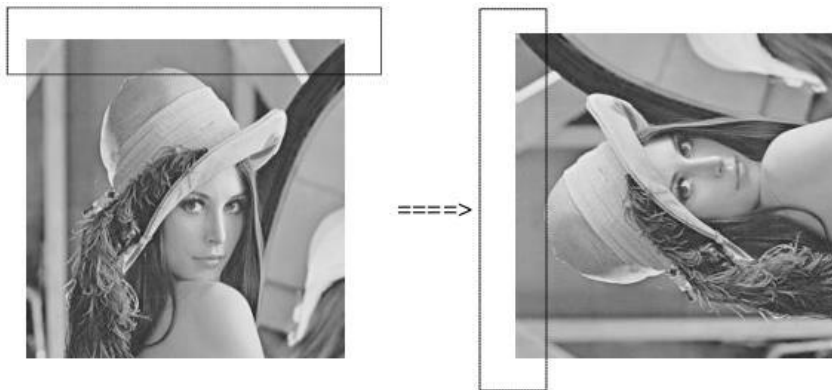
Catatan:

Rumus tersebut digunakan untuk memutar sebuah titik dalam gambar, seperti ketika kita memutar gambar searah atau berlawanan dengan jarum jam.

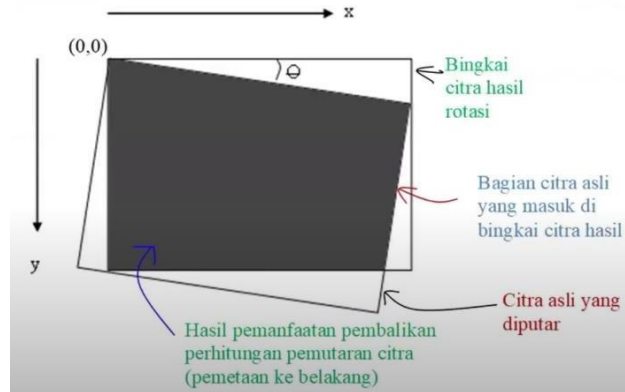
- **x baru** adalah posisi baru pada sumbu x setelah diputar.
- **y baru** adalah posisi baru pada sumbu y setelah diputar.
- **Θ (theta)** adalah sudut rotasi, yaitu seberapa besar gambar diputar. Berikut adalah langkah-langkah dasar:

1. **Sumbu x:** Untuk menghitung posisi baru di sumbu x, kita kalikan posisi awal x dengan $\cos(\theta)$ lalu tambahkan posisi awal y yang dikalikan dengan $\sin(\theta)$.
2. **Sumbu y:** Untuk posisi baru di sumbu y, kita kalikan posisi awal y dengan $\cos(\theta)$, lalu kurangi posisi awal x yang dikalikan dengan $\sin(\theta)$.

Jika sudutnya positif, rotasinya berlawanan dengan arah jarum jam. Jika sudutnya negatif, rotasinya searah jarum jam. Contohnya, jika kita memutar gambar 90 derajat, posisi tiap titik akan berubah mengikuti rumus ini, seperti saat kita memutar kertas untuk mengubah orientasi gambar.



Rotasi citra lena sejauh 90° berlawanan arah jarum jam



ini adalah rotasi dengan nilai atau dengan titik pusat 0,0 artinya dipojok kiri atas, kemudian diputar sebanyak Θ (theta), artinya akan ada bagian-bagian citra yang hilang atau diluar dari area pada citra. Terdapat dua pemetaan dalam rotasi citra seperti: rotasi pemetaan kebelakang (Bacward Mapping) dan kedepan (Forward Mapping).

Rotasi Pemetaan Kebelakang (Backward Mapping)

Dalam **rotasi pemetaan kebelakang**, proses rotasi dimulai dengan citra hasil (output) dan mencari asal dari setiap piksel di citra asli (input) yang akan mengisi piksel tersebut.

Dengan kata lain, untuk setiap piksel di citra yang sudah diputar, kita menghitung dari mana asalnya di citra asli sebelum rotasi dilakukan. Misalnya, saat kita ingin memutar sebuah gambar sebesar 30 derajat, untuk setiap piksel yang kita temukan di citra hasil, kita harus melakukan perhitungan ke belakang, yaitu mencari koordinat asal di citra asli. Dalam kasus ini, kita memetakan citra hasil kembali ke citra asli, sehingga disebut rotasi pemetaan ke belakang.

Rotasi Pemetaan Kedepan (Forward Mapping)

Dalam **rotasi pemetaan kedepan**, proses rotasi dimulai dari citra asli (input), dan setiap piksel di citra asli dipetakan atau dipindahkan ke posisi barunya di citra hasil (output) setelah rotasi. Berbeda dengan pemetaan kebelakang, dalam metode ini kita menggerakkan piksel dari citra asli dan meletakkannya di lokasi barunya di citra hasil sesuai dengan perhitungan rotasi.

Perbandingan Antara Pemetaan Kebelakang Dan Kedepan

Faktor	Pemetaan Kebelakang (Backward Mapping)	Pemetaan Kedepan (Forward Mapping)
Proses	Memulai dari citra hasil dan mencari asal piksel di citra asli	Memulai dari citra asli dan memetakan piksel ke posisi baru di citra hasil

Kekosongan/Celah (Gaps)	Tidak ada celah karena setiap piksel di citra hasil terisi	Bisa ada celah atau lubang di citra hasil
Interpolasi	Diperlukan jika hasil perhitungan jatuh di antara piksel	Tidak diperlukan jika koordinatnya tepat
Kompleksitas	Lebih kompleks dan membutuhkan waktu lebih lama	Lebih cepat dalam beberapa kasus, tetapi bisa kehilangan detail
Keakuratan	Hasil lebih akurat dan detail	Bisa terjadi overlapping atau kehilangan detail

Namun, tantangan dari operasi rotasi adalah penanganan piksel yang jatuh di luar batas citra. Piksel yang diputar keluar dari bingkai citra sering kali diabaikan atau diberi nilai latar belakang. Selain itu, karena koordinat piksel yang baru setelah rotasi mungkin tidak seluruhnya berupa bilangan bulat, interpolasi sering diperlukan untuk mengisi nilai piksel baru.

Interpolasi piksel adalah teknik yang digunakan dalam pengolahan citra digital untuk memperkirakan atau menghitung nilai intensitas piksel baru ketika citra dimodifikasi, misalnya ketika citra di-*resize* (diubah ukurannya), dirotasi, atau di-*warp* (dibengkokkan). Teknik interpolasi piksel sangat penting dalam berbagai aplikasi seperti memperbesar gambar (*upscaling*), rotasi, transformasi geometrik, serta memperbaiki kualitas gambar yang buram. Metode interpolasi yang umum digunakan dalam rotasi adalah **nearest-neighbor interpolation**, **bilinear interpolation**, dan **bicubic interpolation**.

Nearest-Neighbor Interpolation adalah metode yang memilih piksel tetangga terdekat sebagai nilai piksel baru. Metode tersebut cepat tetapi sering menghasilkan citra dengan tepi yang kasar atau pixelated.

Bilinear Interpolation adalah metode yang menghitung nilai piksel baru berdasarkan rata-rata tertimbang dari piksel-piksel di sekitarnya. Bilinear interpolation menghasilkan citra yang lebih halus dibandingkan nearest-neighbor, tetapi membutuhkan waktu komputasi yang lebih lama.

Bicubic Interpolation adalah metode yang menggunakan rata-rata piksel di sekitar titik rotasi dengan menghitung faktor kubik. Meskipun lebih rumit dan membutuhkan lebih banyak waktu komputasi, bicubic interpolation menghasilkan gambar yang lebih halus dengan tepi yang lebih lembut.

2.6.4 Refleksi

Refleksi atau pencerminan adalah operasi geometri yang mencerminkan citra terhadap sumbu tertentu, seperti sumbu-x, sumbu-y, atau sumbu

diagonal. Refleksi dapat digunakan untuk membalikkan orientasi objek dalam citra, misalnya, membalik citra horizontal atau vertikal. Refleksi berguna dalam berbagai aplikasi seperti **desain grafis**, **fotografi digital**, dan **pengenalan pola**, di mana simetri dan pencerminan sering diperlukan. Secara matematis, refleksi dapat digambarkan dengan persamaan berikut:

Refleksi terhadap sumbu-x:

$$X' = x \text{ (nilai } x \text{ tidak berubah)}$$

$$Y' = -y \text{ (nilai } y \text{ berubah tanda, menjadi kebalikannya)}$$

Catatan: Pada refleksi terhadap sumbu-x, hanya nilai y yang berubah. Nilai x tetap sama. Objek akan dibalik secara vertikal (atas-bawah) terhadap sumbu-x. Bayangkan kita membalik gambar seperti cermin yang diletakkan di bagian bawah gambar. Poin di atas akan pindah ke bawah, dan sebaliknya.

2.7 Operasi Morfologi

2.7.1 Pengertian Operasi Morfologi

Operasi morfologi adalah teknik pengolahan citra yang didasarkan pada bentuk segmen atau region dalam citra. Karena difokuskan pada bentuk objek, maka operasi ini biasanya diterapkan pada citra biner. Biasanya segmen tadi didasarkan pada objek yang menjadi perhatian. Segmentasi dilakukan dengan membedakan antara objek dan latar, antara lain dengan memanfaatkan operasi pengambangan yang mengubah citra warna dan skala keabuan menjadi citra biner. Nilai biner dari citra hasil merepresentasikan 2 keadaan: objek dan bukan objek (latar). Meskipun lebih banyak dipakai pada citra biner, operasi morfologi sering pula digunakan pada citra skala keabuan dan warna. Hasil operasi morfologi dapat dimanfaatkan untuk pengambilan keputusan dengan analisis lebih lanjut.

Operasi morfologi menerapkan elemen penataan ke gambar masukan, menciptakan hasil keluaran gambar dengan ukuran yang sama. Pada operasi morfologi, nilai setiap piksel pada gambar keluarannya didasarkan pada perbandingan piksel yang bersesuaian pada citra masukan dengan tetangganya.

Secara umum pemrosesan citra secara morfologi dilakukan dengan cara mempassing sebuah Structuring Element terhadap sebuah citra dengan cara yang hampir sama dengan konklusi. Structuring Element dapat diibaratkan dengan mask pada pemrosesan citra yang lain selain pemrosesan citra secara morfologi. Structuring Element dapat berukuran sembarang, Structuring Element juga memiliki titik poros.

Perbedaan antara pemrosesan citra secara morfologis dengan pemrosesan biasa yaitu terletak pada sudut pandang dari sebuah citra, pemrosesan biasa memandang sebuah citra sebagai suatu fungsi intensitas terhadap posisi (x,y) , sedangkan dengan pendekatan morfologi memandang suatu citra sebagai himpunan. Pemrosesan citra secara morfologi biasanya dilakukan terhadap citra biner (hanya terdiri dari 0 dan 1), walaupun tidak menutup kemungkinan dilakukan terhadap citra dengan skala keabuan 0-255.

2.7.2 Jenis-Jenis Operasi Morfologi

a). Erosi

Operasi erosi adalah operasi dimana objek diperkecil dengan mengikis sekeliling objek. Cara yang dapat dilakukan juga ada 2. Cara pertama yaitu dengan mengubah semua titik batas menjadi titik latar dan cara kedua dengan menset semua titik di sekeliling titik latar menjadi titik latar.

b). Dilasi

Operasi dilasi dilakukan untuk memperbesar ukuran segmen objek dengan menambah lapisan di sekeliling objek. Terdapat 2 cara untuk melakukan operasi ini, yaitu dengan cara mengubah semua titik latar yang bertetangga dengan titik batas menjadi titik objek, atau lebih mudahnya set setiap titik yang tetangganya adalah titik objek menjadi titik objek. Cara kedua yaitu dengan mengubah semua titik di sekeliling titik batas menjadi titik objek, atau lebih mudahnya set semua titik tetangga sebuah titik objek menjadi titik objek.

c). Opening (Pembukaan)

Operasi opening merupakan kombinasi antara operasi erosi dan dilasi yang dilakukan secara berurutan, tetapi citra asli dierosi terlebih dahulu baru kemudian hasilnya didilasi. Operasi ini digunakan untuk memutus bagian-bagian dari objek yang hanya terhubung dengan 1 atau 2 buah titik saja, atau menghilangkan objek-objek kecil dan secara umum mensmoothkan batas dari objek besar tanpa mengubah area objek secara signifikan. Opening adalah idempotent yaitu apabila operasi opening diulang-ulang tidak memberikan dampak yang berkelanjutan.

d). Closing (Penutupan)

Operasi Closing adalah kombinasi antara operasi dilasi dan erosi yang dilakukan secara berurutan. Citra asli didilasi terlebih dahulu, kemudian hasilnya dierosi. Operasi ini digunakan untuk menutup atau menghilangkan lubang-lubang kecil yang ada dalam segmen objek, menggabungkan objek yang berdekatan dan secara umum mensmoothkan batas dari objek besar tanpa mengubah objek secara signifikan.

e). Hit and Miss Transform

Transformasi hit-and-miss adalah operasi morfologi biner umum yang dapat digunakan untuk mencari pola piksel latar depan dan latar belakang tertentu dalam gambar. Ini sebenarnya adalah operasi dasar morfologi biner karena hampir semua operator morfologi biner lainnya dapat diturunkan darinya. Seperti halnya operator morfologi biner lainnya, ia mengambil sebagai input gambar biner dan elemen penataan, dan menghasilkan gambar biner lain sebagai output.

Elemen penataan dalam metode hit-and-miss merupakan pengembangan dari elemen yang digunakan dalam proses erosi dan pelebaran, karena elemen ini bisa memiliki piksel latar depan maupun latar belakang, bukan hanya piksel latar depan saja, yaitu nilai satu dan nol. Jenis elemen penataan yang lebih sederhana yang digunakan pada erosi dan pelebaran juga sering digambarkan dengan nilai satu dan nol, namun dalam kasus ini, nol hanya berarti "tidak diperhitungkan" dan berfungsi untuk membantu mengatur elemen penataan dalam bentuk yang lebih praktis, biasanya persegi. Pada ilustrasi, nilai "tidak diperhitungkan" ini ditampilkan sebagai area kosong di dalam kernel agar lebih mudah dipahami.

2.8 Operasi Pada Citra Biner

2.8.1 Ekstraksi Tepi Objek

Ekstraksi tepi objek pada citra biner adalah proses untuk mendeteksi batas atau tepi objek dalam gambar biner, menggunakan algoritma yang dapat menyoroti piksel tepi. Salah satu algoritma yang digunakan untuk ekstraksi tepi adalah algoritma 8-ketetanggaan, seperti yang dibahas oleh

Davis pada tahun 1990. Lihatlah pada gambar dibawah ini dilakukan dengan menggunakan 8 ketetanggan, piksel P mempunyai 8 tetangga yang dinyatakan dengan P0 hingga P7.

P ₃	P ₂	P ₁
P ₄	P	P ₀
P ₅	P ₆	P ₇

ALGORITMA – Memperoleh tepi objek

Masukan:

- $f(m,n)$: Citra masukan berupa citra biner berukuran m baris dan n kolom

Keluaran:

- $g(m,n)$: Hasil citra yang berisi tepi objek

```

FOR q ← 2 to m-1
  FOR p ← 2 to n-1
    p0 ← f(q, p+1)
    p1 ← f(q-1, p+1)
    p2 ← f(q-1, p)
    p3 ← f(q-1, p-1)
    p4 ← f(q, p-1)
    p5 ← f(q+1, p-1)
    p6 ← f(q+1, p)
    p7 ← f(q+1, p+1)
    sigma ← p0 + p1 + p2 + p3 + p4 + p5 + p6 + p7
    IF sigma = 8
      g(q, p) ← 0
    ELSE
      g(q, p) ← f(q, p)
    END-IF
  END-FOR
END-FOR
END-FOR

```

Algoritma ini mengasumsikan bahwa semua piksel pada kolom pertama, kolom terakhir, baris pertama, dan baris terakhir tidak ada yang bernilai 1. Apabila ada kemungkinan bahwa piksel pada posisi tersebut ada yang bernilai satu, perlu dibentuk larik baru yang berukuran $(m+2) \times (n+2)$, yang mencakup seluruh nilai f dan dengan bagian tepi larik berisi 0.

Dalam penelitian ini, kami menerapkan algoritma deteksi tepi pada citra biner menggunakan bahasa pemrograman Python. Proses ini memanfaatkan pustaka OpenCV untuk manipulasi citra dan Matplotlib untuk visualisasi hasil.

```

import cv2
import numpy as np
import matplotlib.pyplot as plt

def tepibiner(F):
    G =
    np.zeros_like(F)
    rows, cols
    = F.shape
    for q in range(1,
        rows - 1): for p
        in range(1, cols
        - 1):
        p0 =
        F[q,
        p+1]
        p1 =
        F[q-
        1,
        p+1]
        p2 = F[q-1, p]
        p3 = F[q-1, p-1]
        p4 = F[q, p-1]
        p5 =
        F[q+1
        , p-1]
        p6 =
        F[q+1
        , p]
        p7 =
        F[q+1
        , p+1]
        sigma = p0 + p1 + p2 + p3 + p4 + p5
        + p6 + p7 if sigma == 8:
            G[q, p] = 0
        else:
            G[q, p] =
            F[q, p] return G

# Baca gambar sebagai citra biner
Img = cv2.imread('foto1.jpg', cv2.IMREAD_GRAYSCALE)
_, BW = cv2.threshold(Img, 127, 1, cv2.THRESH_BINARY)

# Jalankan fungsi
tepibiner G =
tepibiner(BW)

```

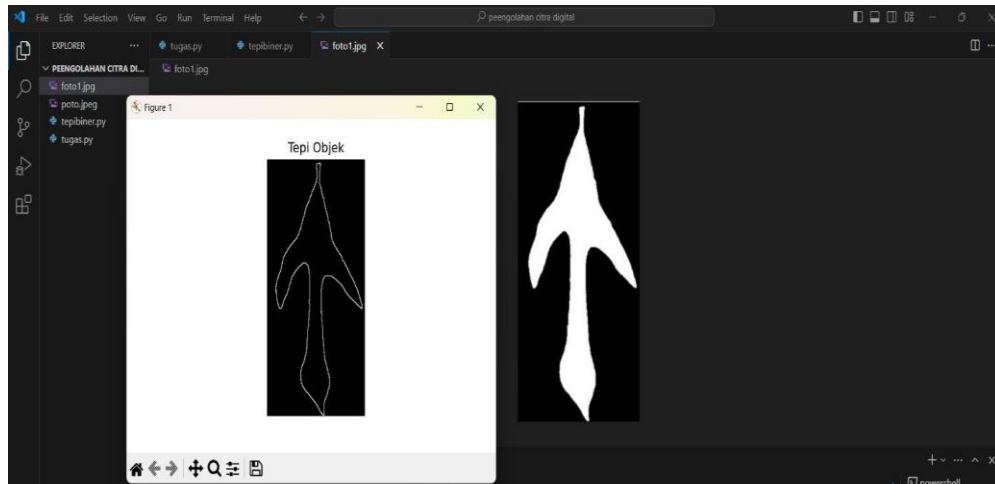
```
# Tampilkan hasil menggunakan
matplotlib plt.imshow(G * 255,
cmap='gray') plt.title("Tepi Objek")
plt.axis("off") # Opsional: Sembunyikan
sumbu plt.show()

# Simpan hasil ke file
cv2.imwrite("hasil_tepi.jpg", G *
255)
# Simpan hasil ke file
cv2.imwrite("hasil_tepi.jpg", G *
255)
print("Gambar hasil disimpan sebagai hasil_tepi.jpg")
```

Proses deteksi tepi objek dapat dijelaskan dalam beberapa langkah berikut:

1. Citra dibaca dalam format grayscale dari file menggunakan fungsi `cv2.imread`.
2. Konversi Citra ke Biner, citra grayscale diubah menjadi citra biner dengan menerapkan metode `thresholding`. Metode ini menghasilkan array dua dimensi yang hanya berisi nilai 0 dan 1, di mana 1 merepresentasikan objek dan 0 merepresentasikan latar belakang.
3. Implementasi Fungsi Deteksi Tepi
 - Fungsi `tepibiner` dikembangkan untuk melakukan deteksi tepi. Fungsi ini mengevaluasi setiap piksel dalam citra biner, kecuali yang terletak di tepi citra, dengan memeriksa nilai delapan piksel tetangga.
 - Jika semua delapan piksel di sekeliling piksel yang sedang diproses bernilai 1, maka piksel tersebut tidak dianggap sebagai tepi, dan nilainya diatur menjadi 0. Sebaliknya, jika terdapat piksel tetangga yang bernilai 0, nilai piksel tersebut akan tetap sama dengan nilai pada citra asli.
4. Setelah proses deteksi tepi selesai, hasilnya ditampilkan menggunakan `Matplotlib`, dan citra yang telah diproses disimpan ke dalam file dengan format `JPEG` untuk analisis lebih lanjut.

Hasil :



2.8.2 Representasi Bentuk

Representasi bentuk adalah konsep penting dalam pengolahan citra, analisis objek, dan pengenalan pola. Ini merujuk pada cara untuk menggambarkan karakteristik geometris dari suatu objek dalam citra, sehingga dapat digunakan untuk identifikasi, klasifikasi, dan analisis lebih lanjut.

1. Representasi bentuk umumnya melibatkan beberapa komponen utama:
 - a. Kontur: Garis yang membentuk batas objek. Kontur adalah elemen kunci dalam representasi bentuk karena memberikan informasi dasar tentang bentuk luar objek.
 - b. Area: Luas dari objek yang didefinisikan oleh kontur. Ini memberikan informasi tentang ukuran objek dan sering digunakan dalam analisis perbandingan.
 - c. Dimensi: Ini mencakup panjang, lebar, tinggi, dan sifat geometris lainnya yang membantu menggambarkan bentuk objek.
 - d. Titik Pusat: Titik yang mewakili lokasi pusat dari objek, sering digunakan dalam analisis simetri dan penempatan objek dalam ruang.
2. Teknik Representasi Bentuk
 - a. Polygonal Approximation (Hampiran Poligon): Metode ini

menyederhanakan kontur objek dengan mengganti kurva yang kompleks dengan poligon yang lebih sederhana, sehingga mengurangi jumlah titik dan informasi yang diperlukan untuk merepresentasikan objek.

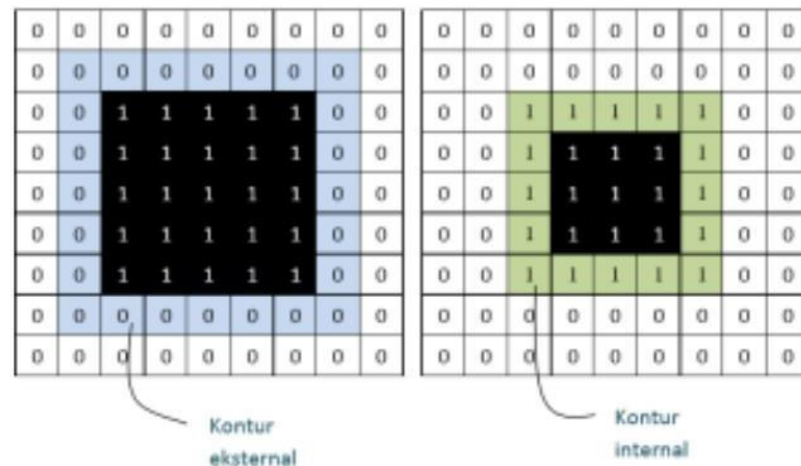
- b. Chain Code: Teknik ini merepresentasikan kontur objek sebagai urutan kode yang menunjukkan arah gerakan di sepanjang kontur. Kode ini memudahkan analisis dan penyimpanan informasi tentang bentuk.
- c. Moments: Penggunaan momen (moments) untuk menggambarkan bentuk dalam hal statistik, seperti momen pusat yang dapat memberikan informasi tentang posisi dan orientasi objek.
- d. Fourier Descriptors: Metode ini menggunakan transformasi Fourier untuk menggambarkan bentuk objek dalam domain frekuensi.

2.8.3 Kontur Objek

Kontur adalah garis atau kurva yang mengelilingi area yang memiliki warna atau intensitas yang sama dalam citra. Kontur digunakan untuk menggambarkan batas objek, dan dapat diekstrak dari citra digital dengan menggunakan berbagai teknik pengolahan citra. Kontur objek adalah representasi visual dari batas objek yang terdeteksi dalam citra. Dalam pengolahan citra, kontur memainkan peran penting karena mereka membantu dalam analisis bentuk, ukuran, dan struktur objek. Kontur terbagi menjadi 2, yaitu:

- a). Kontur internal adalah batasan yang terletak di dalam objek. Biasanya, kontur ini muncul di objek yang memiliki beberapa bagian, seperti lingkaran dalam pada bentuk oval atau bagian yang terpisah dalam objek kompleks. Kontur internal dapat digunakan untuk menganalisis fitur-fitur yang terdapat di dalam objek, seperti ruang kosong atau bagian yang lebih terang/gelap di dalam objek.
- b). Kontur eksternal adalah batas luar dari objek yang memisahkannya

dari latar belakang. Kontur ini menggambarkan bentuk keseluruhan objek dan merupakan kontur yang paling umum digunakan dalam pengenalan pola dan analisis objek. Kontur eksternal memberikan informasi penting mengenai ukuran, bentuk, dan posisi objek dalam citra, serta digunakan dalam aplikasi seperti pelacakan objek dan pengenalan wajah.



Implementasi kontur objek:

```
import cv2
import numpy as np
import matplotlib.pyplot as plt

# Langkah 1: Membaca
Citra image = cv2.imread('foto1.jpg')

# Langkah 2: Mengonversi Citra ke Grayscale
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

# Langkah 3: Menerapkan Gaussian Blur untuk mengurangi noise
blurred = cv2.GaussianBlur(gray, (5, 5), 0)

# Langkah 4: Menerapkan Canny Edge Detection
edges = cv2.Canny(blurred, 50, 150)

# Langkah 5: Menemukan Kontur
contours, hierarchy = cv2.findContours(edges, cv2.RETR_EXTERNAL,
```

```

cv2.CHAIN_APPROX_SIMPLE)

# Langkah 6: Menggambar Kontur pada Citra
# -1 untuk menggambar semua kontur, warna hijau (0, 255, 0), ketebalan
3 piksel
cv2.drawContours(image, contours, -1, (0, 255, 0), 3)

# Langkah 7: Menampilkan Citra dengan Kontur yang Digambar
menggunakan matplotlib
# Mengonversi citra dari BGR ke RGB untuk menampilkan
dengan benar image_rgb = cv2.cvtColor(image,
cv2.COLOR_BGR2RGB)

# Menampilkan
citra
plt.imshow(image_rgb)
plt.title("Kontur
Objek")
plt.axis('off') #
Menonaktifkan
axis plt.show()

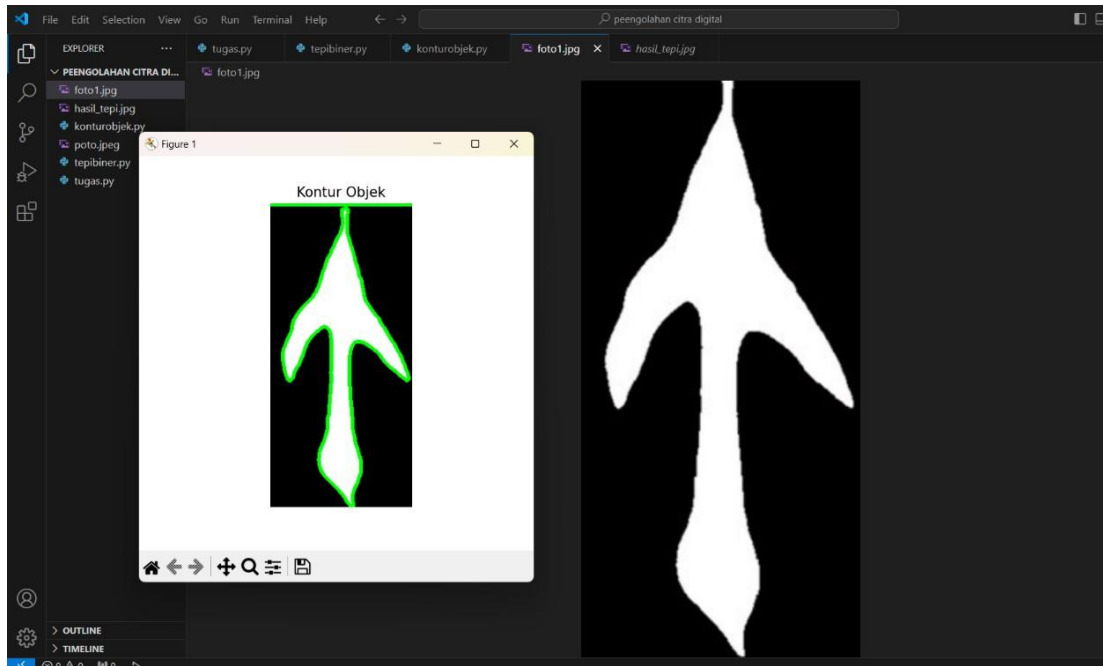
```

Proses kontur dapat dijelaskan dalam beberapa langkah berikut:

1. Menggunakan `cv2.imread()` untuk membaca citra dari file. Gantilah 'image.jpg' dengan jalur citra yang sesuai.
2. Menggunakan `cv2.cvtColor()` untuk mengonversi citra berwarna menjadi grayscale, yang akan mempermudah proses deteksi kontur.
3. Menggunakan `cv2.GaussianBlur()` untuk mengurangi noise dengan aplikasi blur pada citra, agar deteksi tepi menjadi lebih akurat.
4. Canny Edge Detectio, `cv2.Canny()` digunakan untuk mendeteksi tepi objek di dalam citra, menghasilkan citra tepi yang lebih jelas.
5. `cv2.findContours()` digunakan untuk menemukan kontur berdasarkan citra hasil deteksi tepi. Fungsi ini mengembalikan daftar kontur dan hierarki kontur yang ditemukan.
6. Menggunakan `cv2.drawContours()` untuk menggambar semua kontur yang ditemukan pada citra asli. Kontur digambar dengan

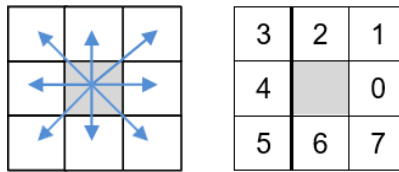
warna hijau dan ketebalan garis 3 piksel.

7. Menampilkan Citra: `cv2.imshow()` digunakan untuk menampilkan citra dengan kontur yang digambar. `cv2.waitKey(0)` menunggu hingga tombol ditekan untuk menutup jendela citra.



2.8.4 Rantai Kode

Rantai kode (*code chain*) merupakan contoh representasi kontur yang mula-mula diperkenalkan oleh Freeman pada tahun 1961. Representasi bentuk dilakukan dengan menggunakan pendekatan 8-ketetanggaan. Pada pendekatan ini, setiap piksel yang berada di tepi objek akan "dirantai" ke piksel tetangganya melalui kode angka yang mewakili arah relatif antara piksel yang bertetangga. Misalnya, jika sebuah piksel di kiri atas tetangganya, akan diberi angka tertentu untuk menunjukkan arah tersebut. Kode rantai setiap tetangga piksel dinyatakan dengan sebuah angka sebagaimana terlihat pada gambar dibawah ini:



Untuk mempermudah perolehan kode rantai piksel yang menjadi tetangga suatu piksel, perlu pembuatan indeks yang dapat dihitung melalui rumus berikut:

$$indeks = 3 \Delta y + \Delta x + 5$$

Dalam rumus ini:

- Δx adalah selisih kolom antara dua piksel yang bertetangga.
- Δy adalah selisih baris antara dua piksel yang bertetangga.

Dalam hal ini, Δx menyatakan selisih nilai kolom dua piksel yang bertetangga dan Δy menyatakan selisih nilai baris dua piksel yang bertetangga. Hubungan kode rantai dan indeks pada persamaan tersebut tersaji pada Tabel berikut.

Δx	Δy	Kode Rantai	Indeks = $3 \Delta y + \Delta x + 5$
0	+1	6	8
0	-1	2	2
-1	+1	5	7
-1	-1	3	1
+1	+1	7	9
+1	-1	1	3
-1	0	4	4

1. Inisialisasi Arah Pergerakan:
 1. Dx dan Dy adalah daftar yang masing-masing mewakili perubahan posisi x dan y. Ini berfungsi sebagai panduan pergerakan untuk kontur.
 2. rantai_kode adalah urutan kode yang digunakan untuk menentukan arah dari titik ke titik dalam gambar kontur. Nilai rantai_kode sesuai dengan indeks pada Dx dan Dy.
2. Membuat Matriks Koordinat Awal:
 1. U adalah array dua dimensi yang menyimpan posisi koordinat untuk setiap titik pada kontur.
 2. $U[0] = [10, 10]$ menentukan posisi awal kontur di koordinat (10,10). Anda bisa menyesuaikan koordinat awal ini sesuai kebutuhan.
3. Fungsi show_contour untuk Menghitung Koordinat Kontur:
 1. Fungsi show_contour menggunakan perulangan untuk menghitung koordinat dari setiap titik pada kontur berdasarkan kode rantai.
 2. rantai_kode[p] mengakses kode untuk setiap titik yang mengarahkan pergerakan ke titik berikutnya.
 3. Dx[bilangan] dan Dy[bilangan] memberikan nilai pergerakan pada sumbu x dan y sesuai dengan arah yang ditentukan oleh rantai_kode.
 4. $U[p, 0]$ dan $U[p, 1]$ diperbarui dengan koordinat baru pada titik tersebut.
4. Menentukan Ukuran Matriks Gambar D:
 1. Matriks D akan digunakan untuk menampilkan kontur. maks_x dan maks_y menentukan ukuran maksimal dari gambar berdasarkan koordinat terbesar dari semua titik kontur.
 2. $\text{np.max}(U[:, 1])$ mengambil nilai maksimum pada kolom x (posisi horizontal), dan $\text{np.max}(U[:, 0])$ untuk kolom y (posisi

vertikal).

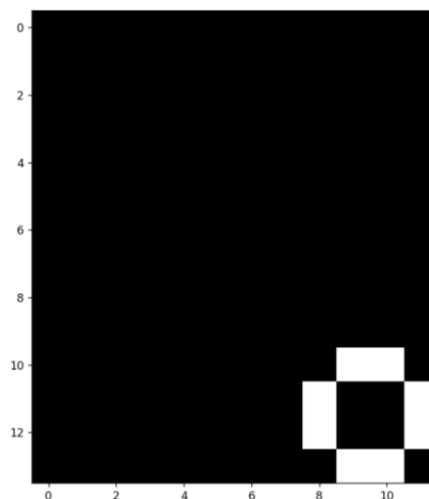
3. D diinisialisasi dengan matriks kosong (bernilai 0) dengan ukuran sesuai maks_x dan maks_y.

5. Menandai Piksel yang Berada di Kontur:

1. Perulangan ini menandai setiap titik yang berada di sepanjang kontur dalam matriks D dengan nilai 1.
2. $D[U[p, 0], U[p, 1]] = 1$ menandakan bahwa titik $(U[p, 0], U[p, 1])$ adalah bagian dari kontur, sehingga memudahkan visualisasi.

6. Menampilkan Gambar Kontur:

1. `plt.imshow(D, cmap='gray')` menampilkan matriks D sebagai gambar, di mana piksel dengan nilai 1 akan muncul sebagai bagian dari kontur, dan lainnya akan tetap hitam (0).
2. `plt.show()` menampilkan gambar di jendela output.



2.8.5 Perimeter

Perimeter atau keliling adalah panjang total dari tepi suatu objek. Pada pengolahan citra, perimeter sering kali dihitung untuk objek yang telah diekstraksi atau dibatasi dalam gambar biner. Perimeter ini dapat digunakan dalam berbagai aplikasi, seperti pengenalan pola, analisis

bentuk objek, dan pengukuran geometris objek dalam citra. Pada gambar di bawah ini, kita dapat melihat ilustrasi sederhana dari sebuah objek daun, yang memiliki perimeter atau keliling yang akan dihitung.



ALGORITMA 8.6 – Estimasi perimeter

Masukan:

- $f(M,N)$: Citra masukan berukuran M baris dan N kolom

Keluaran:

- perimeter

1. Peroleh citra biner.
2. Kenakan algoritma deteksi tepi.
3. Perimeter \leftarrow jumlah piksel pada tepi objek hasil langkah 2.

Implementasi parimeter

1. Pra-pemrosesan Gambar: Gambar pertama kali dimuat dalam mode grayscale (abu-abu) dan kemudian dibinarisasi, yaitu mengubah gambar menjadi bentuk biner di mana objek yang ingin dianalisis dipisahkan dari latar belakang (hitam dan putih).
2. Inbound Tracing: Proses ini digunakan untuk mendeteksi kontur atau batas objek dalam gambar biner. Fungsi `inbound_tracing` menggunakan `cv2.findContours` dari OpenCV untuk mendeteksi kontur-kontur tersebut dan mengembalikan kontur pertama yang ditemukan.
3. Perhitungan Keliling: Setelah kontur ditemukan, keliling objek dihitung dengan menghitung jumlah titik dalam kontur. Karena kontur adalah sebuah loop tertutup, titik pertama dan terakhir adalah sama, sehingga kode mengurangi 1 dari panjang kontur untuk menghindari penghitungan ganda.
4. Hasil: Keliling yang dihitung, yaitu jumlah titik pada kontur, kemudian dicetak sebagai output.

2.9 Operasi Citra Warna

2.9.1 Dasar Warna

a) RGB

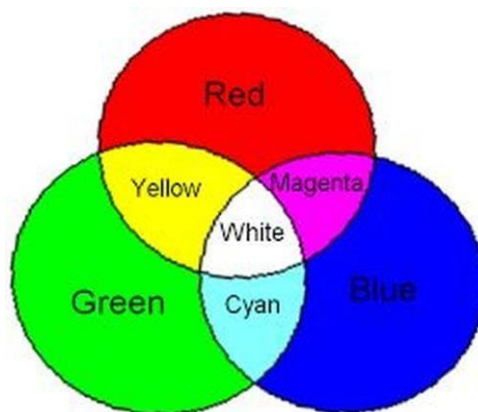
RGB adalah singkatan dari warna merah, hijau, dan biru.

Dalam bahasa Inggris, ketiga warna itu menjadi Red, Green, Blue yang kalau disingkat menjadi RGB. RGB sangat penting karena diantara jutaan warna berbeda ini, ada tiga warna yang sangat peka di mata manusia, dan ketiga warna itu adalah warna merah, hijau, dan biru alias red, green, blue dalam bahasa Inggris. RGB digunakan untuk menampilkan gambar pada layar elektronik. Selain itu, warna merah, hijau, dan biru juga berfungsi untuk menampilkan pancaran cahaya pada layar elektronik.

Kelebihan lain dari model warna RGB ini adalah, gambar bisa dengan mudah disalin dan dipindahkan dari satu software ke software lain sehingga lebih mudah diedit. Ini karena hampir semua software di komputer bisa menerima format gambar RGB, baik itu formatnya JPEG atau PNG. Kekurangan dari model warna ini adalah, ketika gambar digital dicetak ke dalam bentuk kertas dengan menggunakan printer, hasil yang didapatkan cenderung kurang bagus. Ini karena, berbeda dengan gambar digital yang tampil di layar, printer tidak menggunakan model warna RGB. Alih-alih menggunakan RGB, printer menggunakan mode warna CMY untuk hasil yang lebih sempurna.

b) CMY

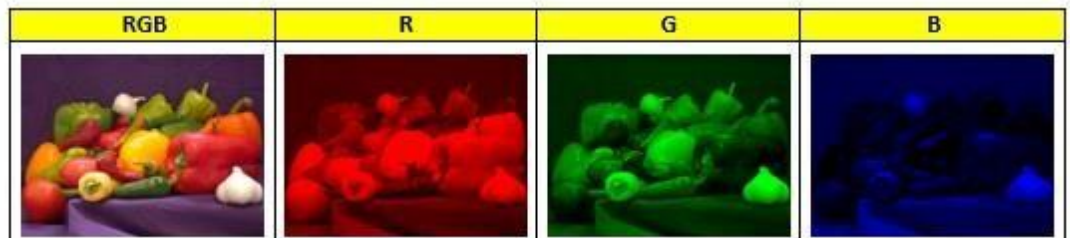
CMY juga merupakan singkatan dari empat warna berbeda dalam Bahasa Inggris yaitu Cyan alias biru kehijauan, Magenta, dan Yellow alias kuning. Dalam desain grafis, jika RGB disebut sebagai warna adaptif, maka CMY dikenal dengan warna subtraktif. Sama seperti warna RGB, warna-warna CMY juga bisa menghasilkan warna lain. Berbeda dengan warna RGB yang banyak digunakan untuk kepentingan gambar pada dunia elektronik, warna CMY banyak digunakan untuk dunia digital printing.



2.9.2 Ruang Warna

a) Ruang Warna RGB

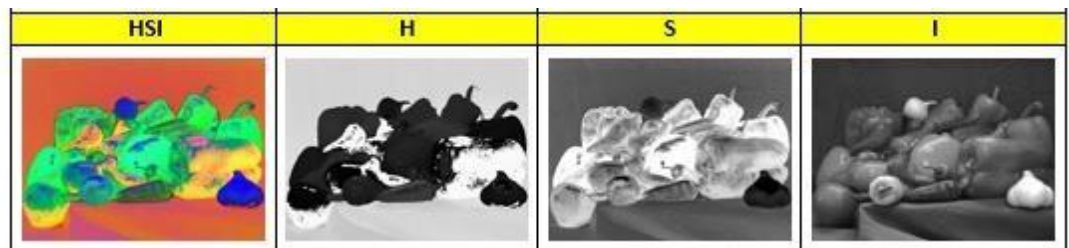
Ruang warna RGB adalah sistem warna yang menggunakan tiga warna dasar: Merah (Red), Hijau (Green), dan Biru (Blue) untuk merepresentasikan setiap warna dalam citra digital. Dalam konteks pengolahan citra, setiap piksel dalam citra direpresentasikan oleh tiga komponen warna ini. Masing-masing komponen warna memiliki nilai intensitas yang berkisar antara 0 hingga 255, di mana 0 berarti tidak ada intensitas warna dan 255 adalah intensitas penuh warna. Ruang warna RGB (Red, Green, Blue) merupakan ruang warna yang paling umum digunakan pada sistem tampilan gambar baik di komputer, televisi, dll.



b) Ruang Warna HSI

Model HSI sering digunakan dalam pengenalan warna dan pengolahan citra medis, terutama ketika penting untuk membedakan warna berdasarkan sifat-sifat visualnya.

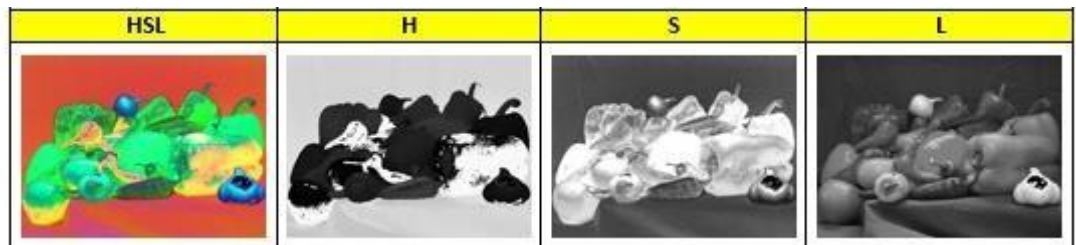
- Hue (H): Hue mengukur tipe warna atau tonenya. Ini adalah komponen yang mewakili warna aktual dalam spektrum warna. Contohnya, merah, biru, atau hijau memiliki nilai hue yang berbeda.
- Saturation (S): Saturation mengukur sejauh mana warna itu jenuh atau pucat. Nilai tinggi menunjukkan warna yang kaya dan jenuh, sedangkan nilai rendah akan lebih mendekati warna abu-abu atau putih.
- Intensity (I): Intensity (juga disebut Value dalam model HSV) mengukur sejauh mana warna itu terang atau gelap. Nilai tinggi menunjukkan warna yang sangat terang, sedangkan nilai rendah adalah warna yang sangat gelap atau bahkan hitam.



c) Ruang Warna HSL

Model HSL sering digunakan dalam desain grafis dan pengolahan citra yang lebih kreatif, seperti dalam pengeditan foto atau penyesuaian tampilan warna.

- Hue (H): Seperti pada model HSI, komponen Hue dalam HSL mengukur tipe warna atau tonenya.
- Saturation (S): Saturation dalam HSL mengukur tingkat kejenuhan warna, mirip dengan HSI.
- Lightness (L): Lightness mengukur tingkat kecerahan warna. Ini memungkinkan untuk mengontrol sejauh mana warna tampak gelap atau terang tanpa mengubah hue atau saturasi.

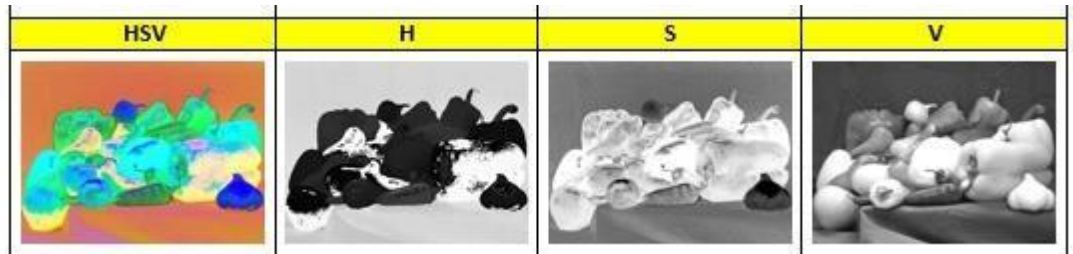


d) Ruang Warna HSV

Model HSV sering digunakan dalam berbagai aplikasi, termasuk pemrosesan gambar medis, pelacakan objek, dan penyesuaian warna dalam pengolahan citra.

- Hue (H): Seperti pada model-model sebelumnya, komponen Hue dalam HSV mengukur tipe warna atau tonenya.
- Saturation (S): Sama seperti HSL dan HSI, komponen Saturation dalam HSV mengukur tingkat kejenuhan warna.
- Value (V): Value mengukur tingkat kecerahan warna,

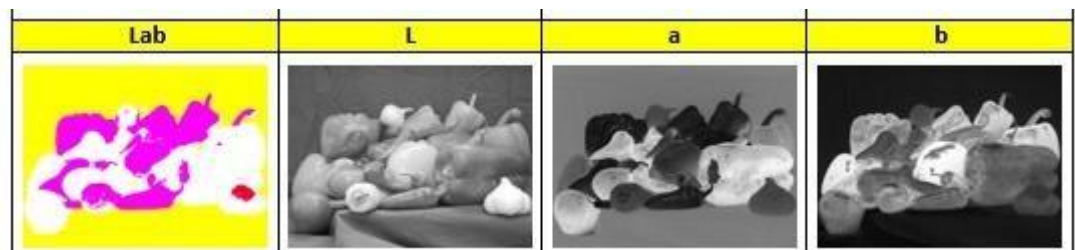
mirip dengan Intensity dalam model HSI. Ini memungkinkan kontrol terhadap kecerahan atau kegelapan warna.



e) Ruang Warna LAB

Model LAB adalah salah satu model ruang warna yang paling serbaguna dalam pengolahan citra karena memisahkan kecerahan dari informasi warna, sehingga lebih mudah untuk mengidentifikasi, memanipulasi, dan menganalisis citra

- L (Lightness): Ini mengukur kecerahan atau kegelapan piksel dan berjalan dari 0(hitam) hingga 100 (putih).
- A (Green to Magenta): Komponen ini mengukur perbedaan antara hijau (positif) dan magenta (negatif). Ini sering digunakan dalam pengenalan warna.
- B (Blue to Yellow): Komponen B mengukur perbedaan antara biru (positif) dan kuning (negatif). Seperti komponen A, ini juga berguna dalam pengenalan warna.



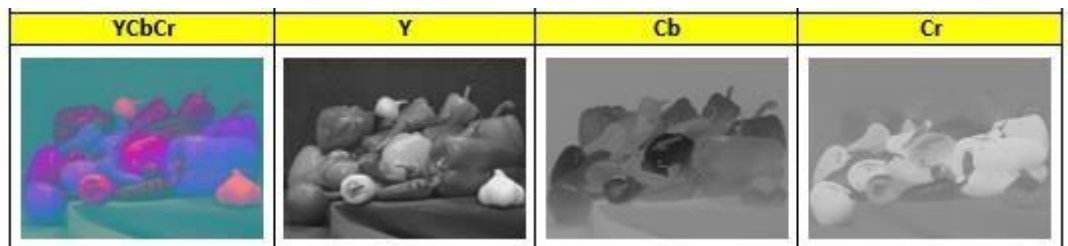
f) Ruang Warna YcbCr

YCbCr adalah model ruang warna yang umum digunakan dalam pemrosesan citra, terutama dalam aplikasi yang melibatkan kompresi video dan transmisi data digital. Model ini memiliki tiga komponen utama:

- Y (Luminance): Ini adalah komponen kecerahan, yang mencerminkan tingkat keabuan atau kecerahan piksel. Y berkisar dari 0 hingga 255.

- Cb (Chrominance Blue): Ini mengukur perbedaan antara biru dan kecerahan piksel. Ini membantu untuk merepresentasikan warna dalam citra.
- Cr (Chrominance Red): Cr mengukur perbedaan antara merah dan kecerahan piksel. Seperti Cb, ini membantu dalam merepresentasikan warna.

Model YCbCr sering digunakan dalam kompresi video, transmisi, dan penyimpanan citra digital karena memungkinkan kompresi efisien dan pemisahan informasi warna dan kecerahan.

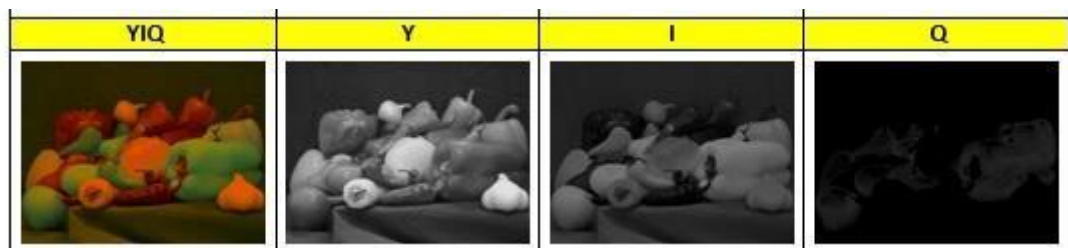


g) Ruang Warna YIQ

YIQ adalah model ruang warna yang awalnya dikembangkan untuk televisi berwarna NTSC (National Television System Committee). Model ini memiliki tiga komponen utama:

- Y (Luminance): Mirip dengan dalam model YCbCr, ini mengukur kecerahan piksel.
- I (In-Phase): Komponen I mengukur perbedaan antara warna biru dan kecerahan.
- Q (Quadrature): Komponen Q mengukur perbedaan antara warna merah dan kecerahan.

YIQ masih digunakan dalam beberapa aplikasi televisi analog, meskipun dalam pengolahan citra digital saat ini, model YCbCr lebih umum digunakan.



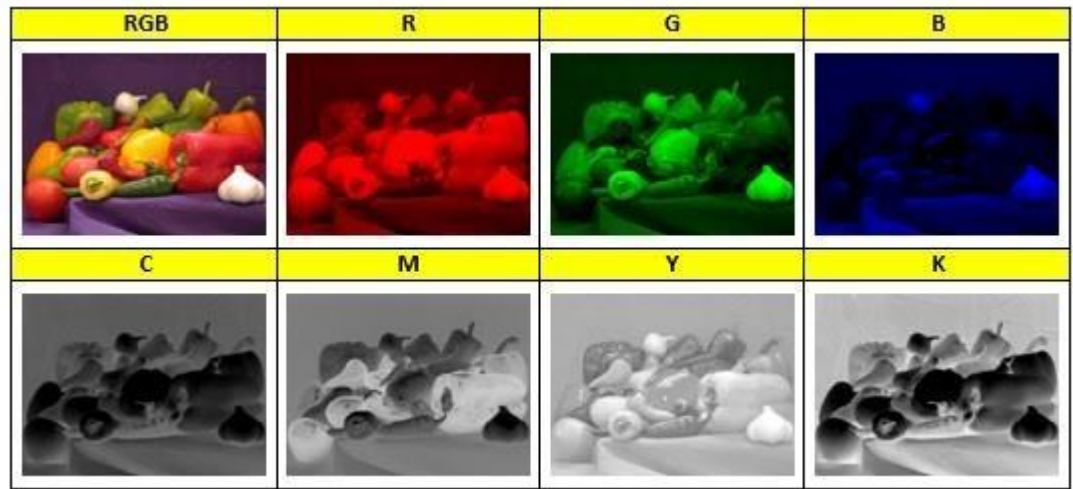
h) Ruang Warna CMYK

Ruang warna CMYK adalah sistem warna yang digunakan

dalam percetakan untuk mereproduksi gambar dan teks berwarna. Ini berbeda dengan model warna RGB (Red, Green, Blue), yang biasanya digunakan dalam tampilan layar dan monitor. Model CMYK menggunakan empat warna dasar: Cyan, Magenta, Yellow, dan Black (Key), dan merupakan fondasi dari hampir semua proses pencetakan warna.

Prinsip utama ruang warna CMYK adalah subtractive color mixing, yang berarti warna-warna dasar ini dicampur dengan mengurangi atau menghilangkan cahaya. Berikut adalah cara kerjanya:

- Cyan: Warna cyan adalah warna yang mirip dengan biru-hijau. Ketika menggabungkan cyan dengan area berwarna putih di atas kertas, area tersebut akan tetap putih karena tidak ada warna yang diserap (0% cyan). Semakin banyak cyan yang ditambahkan, semakin hijau menjadi warna tersebut.
- Magenta: Magenta adalah warna merah ungu. Ketika dicampurkan dengan area putih di atas kertas, ia juga tidak menyerap warna (0% magenta). Semakin banyak magenta yang ditambahkan, semakin merah ungu warna tersebut.
- Yellow: Yellow adalah warna kuning. Ketika menggabungkan yellow dengan area putih di atas kertas, area tersebut tetap putih (0% yellow). Semakin banyak yellow yang ditambahkan, semakin kuning menjadi warna tersebut.
- Black (Key): Warna black (atau key) digunakan untuk meningkatkan kegelapan dan kedalaman warna. Kunci ini digunakan karena campuran murni dari cyan, magenta, dan yellow tidak selalu menghasilkan warna hitam yang murni. Semakin banyak kunci yang ditambahkan, semakin gelap menjadi warna tersebut.



2.9.3 Memperoleh Statistika Warna

a) Nilai Rata-Rata (Mean)

Nilai rata-rata (mean) adalah ukuran yang menunjukkan "pusat" atau nilai tengah dari distribusi warna. Dalam pengolahan citra, nilai rata-rata menghitung intensitas rata-rata untuk setiap saluran warna (misalnya, merah, hijau, biru untuk model RGB). Nilai rata-rata memberi gambaran tentang warna dominan dalam gambar. Sebagai contoh, jika rata-rata saluran merah (R) tinggi dan saluran hijau (G) serta biru (B) rendah, maka gambar cenderung memiliki warna kemerahan. Untuk saluran warna (misalnya, RGB), nilai rata-rata dihitung dengan menjumlahkan semua nilai piksel pada saluran tersebut, kemudian membaginya dengan jumlah piksel.

b) Varians

Varians adalah ukuran yang menggambarkan sebaran atau keragaman nilai warna dari nilai rata-rata. Varians yang tinggi menunjukkan bahwa warna dalam citra sangat bervariasi, sementara varians rendah menunjukkan bahwa warna dalam citra cenderung seragam. Varians digunakan untuk mengukur seberapa besar perbedaan warna antarpiksel. Jika variansnya besar, berarti ada banyak variasi dalam warna citra, sedangkan varians kecil menunjukkan warna yang lebih homogen. Varians dihitung dengan mengukur rata-rata kuadrat selisih antara setiap nilai piksel dan nilai rata-ratanya.

c) Nilai Minimum

Nilai maksimum adalah nilai terbesar yang ditemukan dalam saluran warna tertentu. Dalam gambar RGB, nilai maksimum untuk setiap saluran (merah, hijau, biru) akan menunjukkan warna paling terang atau paling intens pada saluran tersebut. Nilai maksimum digunakan untuk mengetahui seberapa terang atau kontras warna dalam gambar. Sebagai contoh, nilai maksimum 255 pada saluran merah menunjukkan intensitas tertinggi warna merah dalam gambar. Nilai maksimum adalah nilai terbesar yang ditemukan dalam semua piksel di saluran tersebut.

d) Nilai Maksimum

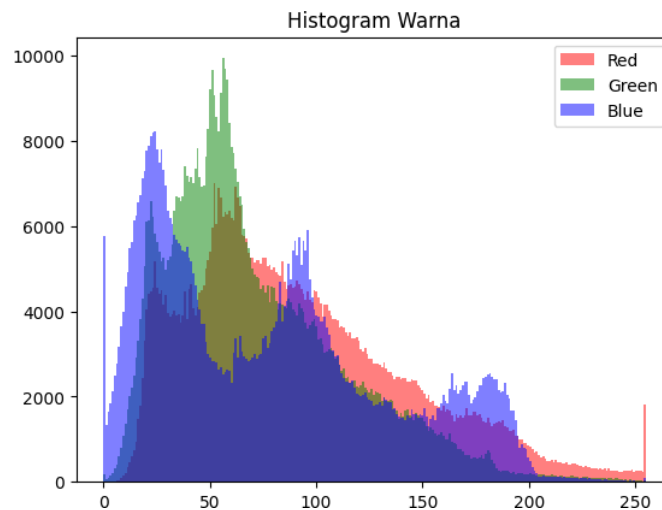
Nilai minimum adalah nilai terkecil yang ditemukan dalam saluran warna tertentu. Ini menunjukkan warna paling gelap atau paling redup pada saluran tersebut. Nilai minimum digunakan untuk mengetahui seberapa gelap warna dalam gambar. Jika nilai minimum pada saluran hijau adalah 0, ini berarti ada area di gambar yang tidak memiliki intensitas hijau sama sekali. Nilai minimum adalah nilai terkecil yang ditemukan dalam semua piksel di saluran tersebut.

e) Contoh



1. Statistik warna Merah:
 - Rata-rata (Mean): 92.29
 - Varians: 2637.16

- Nilai Minimum: 0
 - Nilai Maksimum: 255
2. Statistik warna Hijau:
- Rata-rata (Mean): 71.99
 - Varians: 1778.20
 - Nilai Minimum: 0
 - Nilai Maksimum: 255
3. Statistik warna Biru:
- Rata-rata (Mean): 78.98
 - Varians: 3076.95
 - Nilai Minimum: 0
 - Nilai Maksimum: 255



2.9.4 Mengatur Kecerahan dan Kontras

Kontras dan kecerahan adalah dua konsep kunci dalam dunia pemrosesan citra yang memengaruhi tampilan dan interpretasi suatu gambar. Dalam artikel ini, kita akan menggali lebih dalam tentang apa itu kontras dan kecerahan dalam citra, mengapa keduanya penting, serta bagaimana pengaturannya dapat memengaruhi tampilan akhir suatu gambar

a) Kecerahan (Brightness)

Kecerahan adalah ukuran seberapa terang atau gelap suatu citra keseluruhannya. Ini mencerminkan pencahayaan global pada citra. Kecerahan yang sesuai akan memastikan bahwa citra terlihat nyaman bagi mata manusia tanpa terlalu terang atau terlalu gelap.

Kecerahan dipengaruhi oleh nilai piksel dalam citra, di

mana nilai intensitas piksel yang lebih tinggi menunjukkan bagian gambar yang lebih terang, dan nilai yang lebih rendah menunjukkan bagian yang lebih gelap. Pada model warna RGB, kecerahan

dapat dihitung dengan mempertimbangkan rata-rata atau gabungan dari ketiga saluran (R, G, B).

Jika kecerahan gambar bernilai positif, citra akan lebih terang. Sebaliknya, jika mengurangi nilai piksel dengan nilai negatif, citra akan menjadi lebih gelap. Kecerahan mengontrol keseluruhan tingkat intensitas cahaya dalam gambar. Secara umum, ini mengubah keseluruhan suasana gambar (lebih terang atau lebih gelap)

b) Kontras (Contrast)

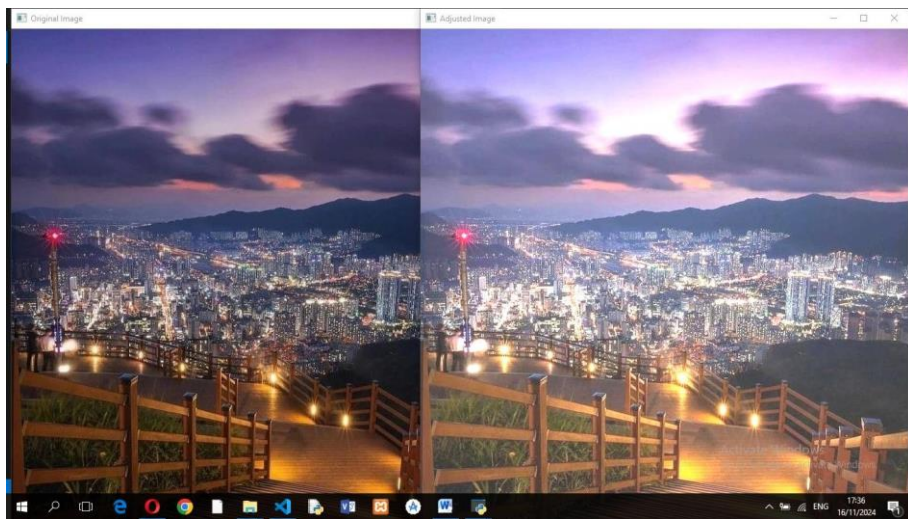
Kontras merujuk pada perbedaan kecerahan antara objek dalam suatu citra. Ini menggambarkan tingkat perbedaan antara piksel yang memiliki intensitas tinggi dan rendah dalam gambar. Kontras yang baik akan membuat objek lebih terlihat jelas dan mempermudah identifikasi detail dalam citra.

Kontras dipengaruhi oleh perbedaan antara nilai piksel dalam citra. Semakin besar perbedaan antara nilai piksel terang dan gelap, semakin tinggi kontrasnya. Pada gambar yang memiliki kontras rendah, nilai piksel di sekitar bagian terang dan gelap cenderung lebih mirip, sementara pada gambar dengan kontras tinggi, perbedaan antara bagian terang dan gelap menjadi lebih tajam.

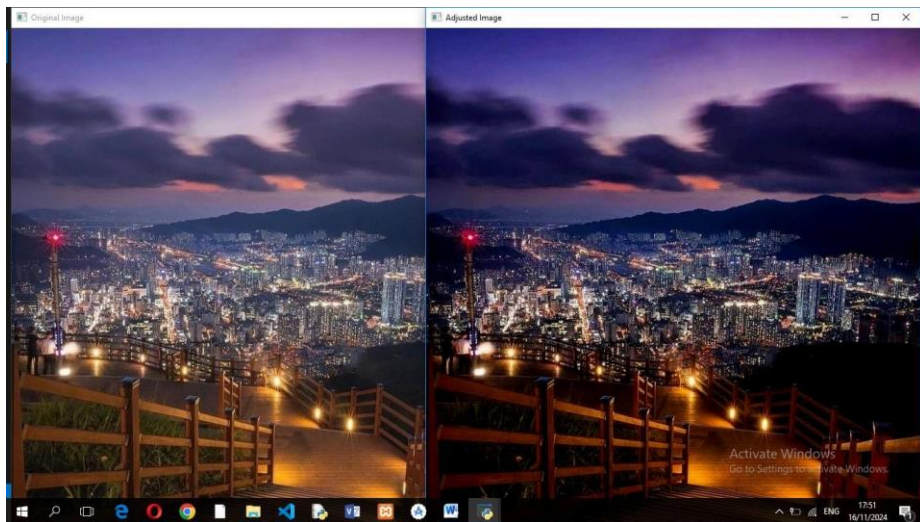
Jika kontras gambar ditingkatkan (misalnya, dengan mengalikan nilai piksel dengan faktor lebih besar dari 1), bagian terang akan semakin terang dan bagian gelap akan semakin gelap, menghasilkan perbedaan yang lebih tajam antara keduanya. Sebaliknya, jika kontras dikurangi (dengan nilai kurang dari 1), gambar akan tampak lebih rata dan kurang jelas perbedaannya antara bagian terang dan gelap.

Kontras mengontrol perbedaan intensitas antara bagian terang dan gelap dalam gambar. Ini mempengaruhi seberapa jelas dan tajam perbedaan antara objek atau elemen dalam citra.

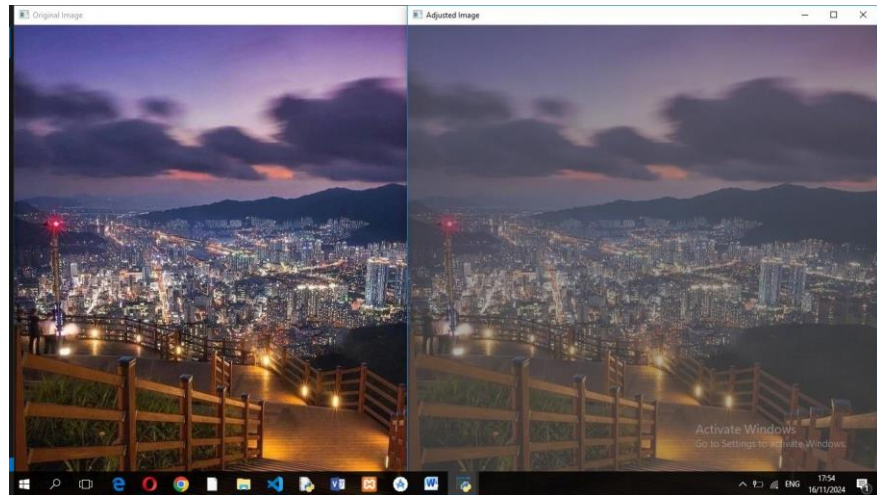
c) Contoh



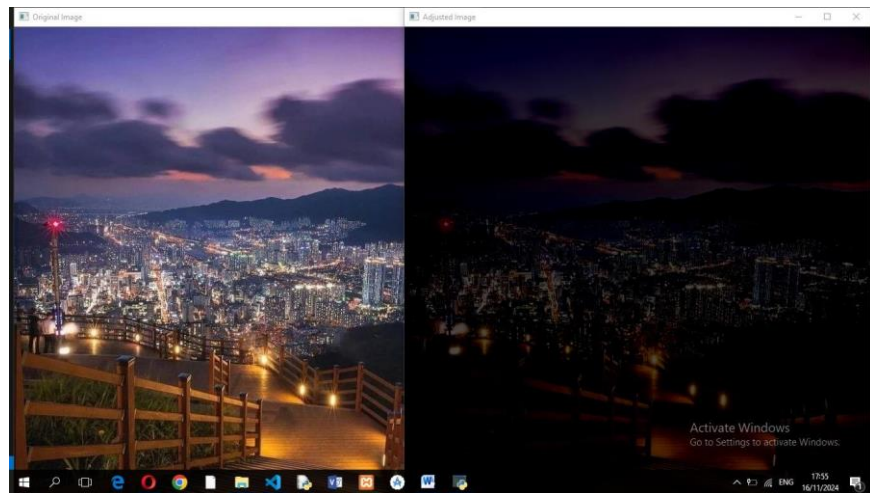
Kecerahan: 50
Kontras: 1.2



Kecerahan: -50
Kontras: 1.2



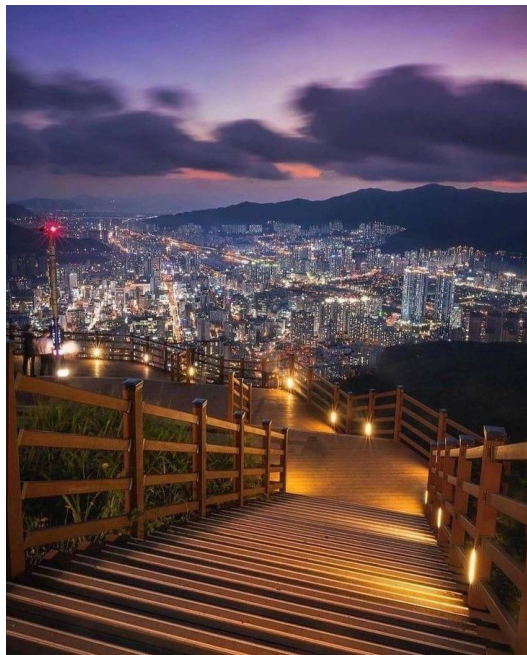
Kecerahan: 50
Kontras: 0.5



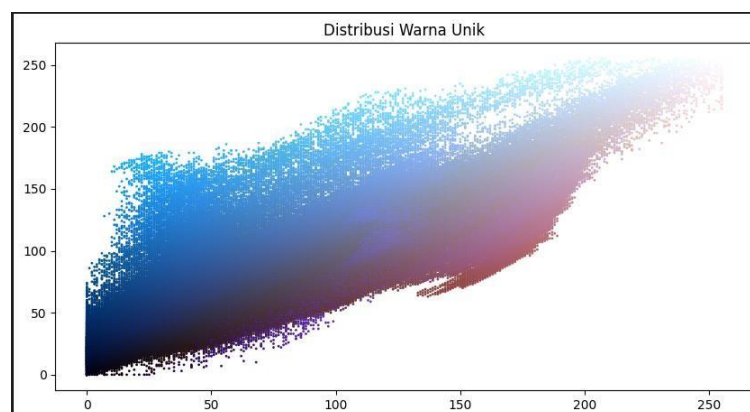
Kecerahan: -50
Kontras: 0.5

2.9.5 Menghitung Jumlah Warna

Warna unik pada citra mengacu pada warna-warna yang muncul dalam gambar yang tidak terulang atau hanya muncul dalam jumlah terbatas. Dalam konteks pemrosesan citra, warna unik merujuk pada kombinasi nilai-nilai intensitas yang berbeda dari saluran warna (seperti RGB) yang membentuk warna tertentu pada piksel dalam gambar. Setiap piksel dalam gambar memiliki nilai warna yang dapat dipetakan sebagai kombinasi dari komponen warna (merah, hijau, biru dalam model RGB, atau komponen lain dalam model warna yang berbeda).



Jumlah warna unik dalam gambar: 135500



BAB III

PENUTUP

3.1 Kesimpulan

Makalah ini membahas berbagai aspek terkait citra digital, mulai dari definisi, perbedaan antara citra analog dan digital, hingga teknik pengolahan citra dan representasi format file. Berikut adalah poin-poin utama yang dapat disimpulkan:

Citra merupakan representasi visual dari objek yang dapat berupa citra analog atau digital. Citra analog bersifat kontinu dan mencakup gambar pada televisi, foto sinar-X, dan lukisan. Sebaliknya, citra digital merupakan representasi diskret yang dapat diproses oleh komputer dan diwakili dalam bentuk fungsi $f(x,y)$ yang menggambarkan intensitas pada koordinat tertentu.

Citra analog tidak dapat diproses langsung oleh komputer dan memerlukan konversi menjadi citra digital. Citra digital dihasilkan melalui proses sampling dan kuantisasi, yang mempengaruhi ukuran pixel dan tingkat kecerahan yang dapat ditangkap oleh komputer.

Representasi citra digital melibatkan proses digitizing yang mengubah citra analog menjadi format digital dengan mengukur warna dan kecerahan setiap pixel. Citra digital dapat ditampilkan dalam beberapa format, termasuk citra biner (hitam-putih), grayscale (skala abu-abu), dan warna (RGB).

Format file citra memiliki peran penting dalam penyimpanan dan distribusi citra. Format umum termasuk Bitmap (.bmp), Portable Network Graphics (.png), dan JPEG (.jpg). Masing-masing format memiliki kelebihan dan kekurangan terkait kompresi, kualitas, dan kompatibilitas.

Elemen penting dalam citra digital meliputi kecerahan, kontras, kontur, warna, bentuk, dan tekstur. Kecerahan dan kontras mempengaruhi kualitas visual citra, sedangkan kontur, warna, dan tekstur memberikan informasi lebih lanjut tentang objek dalam citra.

Pengolahan citra digital mencakup perbaikan kualitas citra,

transformasi geometris, analisis citra, dan kompresi. Teknik-teknik ini meliputi operasi titik, spasial, geometrik, dan aritmatik untuk meningkatkan kualitas citra dan memudahkan interpretasi.

Segmentasi citra adalah teknik untuk membagi citra menjadi daerah-daerah yang memiliki kemiripan atribut. Teknik ini penting untuk analisis lebih lanjut dan identifikasi objek dalam citra. Dengan memahami berbagai aspek ini, kita dapat lebih baik dalam memproses, menganalisis, dan menyimpan citra digital untuk berbagai aplikasi, baik dalam konteks visualisasi maupun analisis data.

Pengolahan citra digital merupakan salah satu bidang yang berkembang pesat seiring dengan kemajuan teknologi, dan memiliki banyak aplikasi di berbagai sektor, seperti medis, keamanan, desain grafis, dan komunikasi. Citra digital dapat dikelompokkan menjadi dua jenis utama, yaitu citra bitmap dan citra vektor. Citra bitmap terdiri dari piksel yang bergantung pada resolusi, sedangkan citra vektor menggunakan garis dan kurva yang dapat diperbesar tanpa kehilangan kualitas. Kedua jenis citra ini memiliki karakteristik yang berbeda, dengan citra bitmap lebih cocok untuk fotografi dan citra vektor lebih ideal untuk ilustrasi dan logo.

Selain klasifikasi citra berdasarkan strukturnya, citra digital juga dibedakan berdasarkan kedalaman warnanya menjadi citra biner, grayscale, dan citra warna (true color). Citra biner hanya memiliki dua nilai warna (hitam dan putih), citra grayscale memiliki variasi warna abu-abu, sementara citra warna mencakup kombinasi tiga warna dasar (RGB) yang menghasilkan jutaan warna berbeda. Setiap jenis citra ini memiliki kegunaan tersendiri, tergantung pada aplikasi dan kebutuhan pemrosesan yang diinginkan.

Sistem koordinat warna, seperti RGB, CIE XYZ, CIELab, HSV, dan HSL, berperan penting dalam merepresentasikan dan mengidentifikasi warna secara akurat dalam citra digital. Setiap sistem koordinat warna menawarkan metode yang berbeda dalam penggambaran warna, dengan RGB sebagai sistem yang paling umum digunakan dalam tampilan layar, sedangkan CIE XYZ dan CIELab digunakan dalam manajemen warna yang lebih presisi,

misalnya dalam industri percetakan. Sistem HSV dan HSL lebih dekat dengan cara manusia secara alami memandang warna, sehingga lebih intuitif digunakan dalam desain grafis dan manipulasi gambar.

Pemahaman tentang jenis-jenis citra digital dan sistem koordinat warna sangat penting dalam aplikasi pengolahan citra. Ini membantu pengguna untuk memilih teknik yang tepat untuk berbagai tujuan, dari desain kreatif hingga analisis medis dan pengenalan objek. Pemahaman ini juga mendukung perancangan aplikasi dan perangkat lunak yang lebih efektif dalam menangani berbagai jenis citra digital dan kebutuhan pengolahan warna.

Pengolahan citra digital melibatkan berbagai teknik untuk meningkatkan kualitas visual dan mempermudah analisis gambar. Setiap teknik memiliki fungsinya masing-masing, seperti meningkatkan kecerahan untuk menonjolkan detail dalam gambar yang kurang terang, atau negasi yang mengubah citra menjadi versi terbaliknya untuk meningkatkan kontras antara objek dan latar belakang. Teknik peregangan kontras berguna untuk memperluas rentang intensitas dalam gambar, sementara pengirisan intensitas membantu menyoroti rentang intensitas tertentu yang relevan untuk analisis.

Penajaman lokal memungkinkan peningkatan ketajaman hanya pada bagian-bagian gambar tertentu tanpa mempengaruhi keseluruhan citra, membantu dalam memfokuskan pada detail-detail penting. Algoritma penghitungan histogram memberikan gambaran tentang distribusi intensitas piksel dalam sebuah citra, sehingga kita dapat menganalisis kualitas dan kontras gambar. Ekualisasi histogram, di sisi lain, menjadi metode penting untuk meratakan distribusi intensitas piksel dan meningkatkan kontras secara keseluruhan, sehingga gambar tampak lebih tajam dan detailnya lebih jelas.

Secara keseluruhan, teknik-teknik ini tidak hanya memperbaiki kualitas visual gambar, tetapi juga memfasilitasi berbagai aplikasi praktis di bidang medis, astronomi, penglihatan mesin, dan fotografi digital. Pemahaman dan penerapan yang tepat dari setiap metode memungkinkan hasil yang lebih optimal dalam berbagai situasi, baik untuk keperluan

visualisasi maupun analisis data citra yang lebih mendalam.

Makalah ini membahas tentang operasi ketetanggaan piksel dalam pengolahan citra digital, yang merupakan elemen penting dalam banyak algoritma pemrosesan citra. Konsep ketetanggaan piksel mencakup hubungan antara piksel dengan tetangganya, yang dapat digunakan untuk memperbaiki kualitas gambar, mendeteksi tepi, dan menghilangkan noise. Terdapat dua jenis utama ketetanggaan piksel, yaitu 4-ketetanggaan dan 8-ketetanggaan, yang diaplikasikan dalam berbagai filter seperti filter batas, filter pererataan, dan filter median.

Masing-masing filter memiliki peran khusus dalam pengolahan citra; filter batas mengidentifikasi tepi, filter pererataan menghaluskan citra, dan filter median efektif dalam menghilangkan noise tanpa mengorbankan detail citra. Selain itu, makalah ini juga memperkenalkan konsep konvolusi yang merupakan proses penting dalam memanipulasi citra melalui operasi matematis pada piksel dan tetangganya. Konvolusi memungkinkan deteksi tepi, penghalusan, dan berbagai transformasi citra lainnya dengan penggunaan kernel matriks.

Operasi aritmatika dan logika pada citra digital memainkan peranan penting dalam pengolahan citra dengan cara memanipulasi nilai intensitas piksel. Setiap operasi, seperti penjumlahan, pengurangan, perkalian, dan pembagian, memiliki tujuan spesifik yang dapat meningkatkan kualitas citra, menggabungkan citra, dan mendeteksi perubahan atau perbedaan antara citra. Operasi logika, termasuk AND, OR, XOR, dan NOT, membantu dalam segmentasi dan analisis citra, serta menyoroti perbedaan dan kesamaan antar citra. Kombinasi antara kedua jenis operasi ini juga memberikan berbagai aplikasi praktis, terutama dalam bidang kedokteran, keamanan, dan industri.

Geometri adalah komponen penting dari pengolahan citra digital, memungkinkan pengguna untuk mengubah posisi, orientasi, dan skala objek dalam gambar dengan melakukan operasi dasar seperti translasi, rotasi, refleksi, dan penskalaan. Setiap operasi memiliki fitur dan masalah tertentu, seperti masalah piksel yang jatuh di luar batas citra, yang memerlukan pendekatan khusus seperti interpolasi untuk menghasilkan hasil yang optimal.

Dalam berbagai aplikasi, seperti analisis visual, pengenalan pola, grafika komputer, dan augmented reality, transformasi geometris ini sangat penting. Seringkali digunakan bersama dengan operasi lain, seperti morfologi atau

pemfilteran, yang memungkinkan pengolahan gambar yang lebih kompleks dan detail.

Operasi morfologi adalah teknik pengolahan citra yang fokus pada bentuk dan struktur objek dalam gambar. Teknik ini umumnya diterapkan pada citra biner tetapi juga dapat diterapkan pada citra skala keabuan dan warna. Operasi ini menggunakan elemen penataan untuk mengubah citra masukan dan menghasilkan citra keluaran yang mempermudah analisis lebih lanjut.

Jenis-jenis Operasi Morfologi:

- Erosi
- Dilasi
- Opening (Pembukaan)
- Closing (Penutupan)
- Hit and Miss Transform
- Thinning
- Thickening

Secara keseluruhan, operasi morfologi memberikan alat yang sangat berguna untuk modifikasi dan analisis citra, memungkinkan pemrosesan gambar yang lebih efektif dan efisien. Teknik – Teknik ini memfasilitasi pembersihan gambar, segmentasi objek, dan ekstraksi fitur, yang sangat penting dalam berbagai aplikasi seperti pengolahan gambar medis dan pengenalan pola.

Citra biner merupakan bentuk penyederhanaan gambar digital yang efektif untuk analisis objek, karena hanya memiliki dua tingkat kecerahan, yaitu hitam dan putih. Penggunaan citra biner mempermudah proses analisis tanpa dipengaruhi oleh variasi warna dan detail yang kompleks, sehingga sering diaplikasikan dalam berbagai bidang seperti deteksi objek, pengenalan pola, dan analisis citra medis. Berbagai teknik dasar pengolahan citra biner, seperti ekstraksi tepi, deteksi kontur, serta perhitungan perimeter, luas, dan diameter objek, memainkan peran penting dalam mengidentifikasi dan mengukur karakteristik geometris objek. Implementasi algoritma-algoritma ini membantu memperoleh informasi bentuk dan ukuran objek dengan akurat. Penerapan teknik-teknik ini dalam berbagai aplikasi nyata, seperti pengukuran morfologi sel dalam medis atau analisis kualitas produk di industri, menunjukkan potensi besar dari pengolahan citra biner untuk mendukung berbagai kebutuhan analisis dan pengenalan pola secara efisien.

Operasi citra warna merupakan aspek penting dalam pengolahan

citra digital, karena memungkinkan manipulasi dan analisis gambar secara lebih mendalam. Model warna seperti RGB, CMY, dan HSI menawarkan pendekatan yang berbeda dalam representasi warna, yang masing-masing memiliki kelebihan dan kekurangan tergantung pada aplikasinya. Teknik pengaturan kecerahan dan kontras sangat mempengaruhi kualitas visual dari citra yang ditampilkan. Ruang warna memainkan peranan kunci dalam menginterpretasikan dan memproses citra, terutama dalam aplikasi yang membutuhkan akurasi tinggi. Dengan memahami statistik warna, kita dapat mengidentifikasi dominasi warna dan variasi yang ada dalam suatu gambar, yang berguna untuk berbagai analisis lebih lanjut.

DAFTAR PUSTAKA

- Budi, U. P. (n.d.). Citra Digital. *Bab II Landasan Teori*, 6-10.
- Kusumo, D. (2017). BAB 1 Pendahuluan 1.1. Latar Belakang Citra Digital . *E-Journal University Atma Jaya Yogyakarta*, 1-3.
- Mangaras, Y. (2022). Dasar Pengolahan Citra Digital Edisi 2022. *Repository UPN Veteran Yogyakarta*, 1-4 dan 11-33.
- Fatmawati, A. (2011). Warna dalam Sistem Visual Manusia. *Jurnal Teknologi Visual*, 9(2), 45-56.
- Kadir, A. (2013). Dasar-Dasar Pengolahan Citra Digital. Yogyakarta: Andi Offset.
- Kaswidjanti, S. (2011). Pengantar Pengolahan Citra Digital. Jakarta: PT RajaGrafindo Persada.
- Munir, R. (2004). Pengolahan Citra Digital dengan Pendekatan Praktis. Bandung: Informatika Bandung.
- Siregar, R. (2009). Teori dan Aplikasi Pengolahan Citra Digital. Surabaya: Graha Ilmu.
- <https://sonoku.com/operasi-inversi-negasi-citra-dengan-delphi/>
<https://putrinursandini.wordpress.com/2016/12/25/bab-3-operasi-piksel-dan-histogram/>
- Riza Fennisya. 2017. *Operasi Ketetanggaan Piksel*. Diakses pada 20 September 2024 <https://rizafennisya.files.wordpress.com/2017/01/pcd-4.pdf>
- Putri Nursadini. 2016. *BAB 4 - Operasi Ketetanggaan Piksel*. Diakses pada 20 September 2024 [Bab 4 – Operasi Ketetanggaan Piksel – Putri Nursadini \(wordpress.com\)](#)
- Hotma Pangaribuan. 2019. *Optimalisasi Kualitas Citra Digital Dengan Metode Ketetanggaan Piksel*. Batam : Universitas Putera Batam)
<https://ejournal.upbatam.ac.id/index.php/jif/article/view/864/672>
- Munir, R. (2021). *Operasi-operasi dasar pengolahan citra*. IF4073 Interpretasi dan Pengolahan Citra. Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung.
- Munir, Rinaldi. "Operasi Dasar Pengolahan Citra." Informatika STEI ITB, 2020-2021. Diakses pada 13 Oktober 2024.

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2020-2021/05-Operasi-dasar-pengolahan-citra-2021.pdf>.

Munir, Rinaldi. Bab 4: Operasi-operasi Dasar Pengolahan Citra Dijital. Informatika STEI ITB. Diakses pada 13 Oktober 2024.
https://informatika.stei.itb.ac.id/~rinaldi.munir/Buku/Pengolahan%20Citra%20Digital/pdf/Bab-4_Operasi-operasi%20Dasar%20Pengolahan%20Citra%20Dijital.pdf.

"Pengolahan Citra Digital Dasar." YouTube, diunggah oleh Dwi Koetin, 7 Februari 2021. Diakses pada 13 Oktober 2024. <https://youtu.be/wTZ5ksO3JzM>.

"Basic Image Processing with OpenCV and Python." YouTube, diunggah oleh OpenCV.org, 12 Oktober 2022. Diakses pada 13 Oktober 2024.
<https://youtu.be/k2x6rX1octg>.

HIPR. (n.d.). *Hit and Miss Transform*. Retrieved from
homepages.inf:<https://homepages.inf.ed.ac.uk/rbf/HIPR2/hitmiss.html>
Gusti Ngurah Winanda Wijaksana, I. G. (n.d.). *Morphologi - Thining Dan Thickening*. Retrieved from SCRIBD:
<https://id.scribd.com/document/51643630/Morphologi-Thinning-dan-Thickening>

Pratama, B. Y. (2007). *Operasi Morfologi Pada Citra Biner*. Retrieved from
IlmuKomputer.Com: <https://ilmukomputer.org/wp-content/uploads/2014/02/Batra-Operasi-Morfologi-Pada-Citra-Biner.pdf>

repository.unikom. (n.d.). *Morfologi Citra*. Retrieved from
repository.unikom:
<https://repository.unikom.ac.id/62706/1/CV-3-%20Morfologi%20Citra.pdf>

TheAILearner. (2019, July 31). *Hit-or-Miss Transform*. Retrieved from
TheAILearner: <https://theailearner.com/2019/07/31/hit-or-miss-transform/>

Trivusi. (2022, November 29). *Pengertian Dan Jenis - Jenis Operasi Morfologi Pada Citra*.

Retrieved from Trivusi: <https://www.trivusi.web.id/2022/10/operasi-morfologi.html>

Scribd, Inc. (2024). *Operasi pada Citra Biner*. Retrieved on

October 30, 2024, from

<https://www.scribd.com/presentation/651092243/Operasi-Pada-Citra-Biner>

Nugroho, S. (2019). BAB 8: Operasi pada Citra Biner: Pengolahan Citra.

Retrieved on October 30, 2024, from

https://www.academia.edu/42968486/BAB_8_Operasi_pada_Citra_Biner_Pengolahan_Citra

Nursandini, P. (2016, December 25). BAB 8: Operasi pada Citra Biner. Retrieved on October 30, 2024, from <https://putrinursandini.wordpress.com/2016/12/25/bab-8-operasi-pada-citra-biner/>

A Andrew. *RGB Adalah: Pengertian Turunan Warna, dan Fungsi dalam Desain*. Diakses pada 16 November, 2024, dari <https://www.gramedia.com/literasi/rgb-adalah/>

Adi Pamungkas. *Ruang Warna dalam Pengolahan Citra*. Diakses pada 16 November, 2024, dari <https://pemrogramanmatlab.com/2023/09/30/ruang-warna-dalam-pengolahan-citra/>

Trivusi. (2022). *Mengenal Model Ruang Warna pada Pengolaha Citra*. Diakses pada 16 November, 2024, dari <https://www.trivusi.web.id/2022/11/model-ruang-warna.html>

Adi Pamungkas. *Kontras dan Kecarahan dalam Citra Digital*. Diakses pada 16 November, 2024, dari <https://pemrogramanmatlab.com/2023/09/15/kontras-dan-kecerahan-dalam-citra-digital/>