**PRAKTIKUM PENGOLAHAN CITRA DIGITAL**



Oleh:

Kelompok I

**Dicky Umardhani (2005551138)**

**PROGRAM STUDI TEKNOLOGI INFORMASI**

**FAKULTAS TEKNIK**

**UNIVERSITAS UDAYANA**

**2022**

# KATA PENGANTAR

Puji syukur penyusun panjatkan kepada Tuhan Yang Maha Esa, karena atas rahmat-Nya, penyusun diberikan kelancaran dalam Praktikum Pengolahan Citra Digital. Selama pembuatan laporan ini, penyusun tidak lepas dari banyak kesulitan dalam pembuatannya. Penyusun mendapat banyak masukan dan bimbingan dari berbagai pihak sehingga tidak lupa penyusun mengucapkan terima kasih kepada dosen pengampu, asisten praktikum selaku pembimbing, orang tua dan keluarga yang selalu memberi motivasi dan dukungan, serta teman-teman mahasiswa Program Studi Teknologi Informasi yang juga mengikuti mata kuliah Praktikum Pengolahan Citra Digital.

Penyusun menyadari bahwa di dalam Laporan Praktikum Pengolahan Citra Digital ini masih terdapat banyak kesalahan dan jauh dari kata sempurna. Oleh karena itu, penyusun memohon maaf apabila terdapat kesalahan kata maupun penulisan di dalam Praktikum Pengolahan Citra Digital ini.

Denpasar, Mei 2022

Penyusun

# ABSTRAK

# DAFTAR ISI

# DAFTAR GAMBAR

# DAFTAR KODE PROGRAM

# DAFTAR TABEL

# BAB I

# PENDAHULUAN

Bab I Pendahulaun membahas tentang latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah, dan sistematika penulisan dari penyusunan laporan Praktikum Pengolahan Citra Digital. Masing-masing topik dibahas sebagai berikut.

## 1.1 Latar Belakang

Perkembangan komputer yang semakin pesat seperti saat ini membuktikan bahwa munculnya berbagai inovasi baru yang bertujuan untuk mempermudah kinerja manusia. Teknologi yang tenagh gencar mengalami perkembangan adalah pada bidang pengolahan citra digital atau *digital image processing.*

Citra digital merupakan representasi dari fungsi intensitas cahaya dalam bentuk diskrit pada bidang dua dimensi. Citra tersusun oleh sekumpulan piksel (picture element) yang memiliki koordinat (x,y) dan amplitudo f(x,y). Koordinat (x,y) menunjukkan letak/posisi piksel dalam suatu citra, sedangkan amplitudo f(x,y) menunjukkan nilai intensitas warna citra.

Pengolahan citra digital (*digital image processing*) merupakan bidang ilmu yang mempelajari tentang bagaimana suatu citra itu dibentuk, diolah, dan dianalisis sehingga menghasilkan informasi yang dapat dipahami oleh manusia. Pengolahan citra digital ini bisa digunakan untuk proses identifikasi dan dalam peningkatan kualitas citra untuk meningatkan informasi yang terkandung dalam citra tersebit agar lebih mudah dipahami oleh manusia. Citra memiliki informasi yang kaya, sering kali citra mengalami *degradasi* atau penurunan kualitas citra itu sendiri, seperti terdapat derau (*noise*) kontras warna tidak sesuai, terlalu tajam, kabur (*blurring*) dan lain sebagainya. Hal tersebut akan mengakibatkan citra sulit untuk di intrepertasikan karena informasi atau pesan yang terkadang di dalamnya tidak dapat disampaikan dengan maksimal.

Berdasarkan latar belakang tersebut, laporan Praktikum Pengolahan Citra Digital dibuat untuk menangani masalah pada proses inteoretasi citra dengan melakukan beberapa percobaan pengolahan citra digital untuk mendapatkan citra dengan kulaitas yang lebih baik sehingga pesan yang terkandung dapat disampaikan dengan baik.

## 1.2. Rumusan Masalah

Berdasarkan latar belakang di atas, maka dapat diketahui rumusan masalah dari laporan Praktikum Pengolahan Citra Digital adalah sebagai berikut.

1. Bagaimana cara membaca dan menyalin sebuah file citra (*image*)?

2. Bagimana cara mengolah citra ke dalam format ruang warna R, G, B, C, M, Y?

3. Bagaimana cara mengimplementasikan algoritma dalam pengolahan citra berbasis titik seperti mengubah dari citra berwarna ke *grayscale,* mengubah citra berwarna ke biner, mengatur kecerahan (*brightness*), dan mengubah ke citra negasi?

4. Bagimana cara melakukan proses *filtering* dengan menggunakan *Low Pass Filter, High Pass Filter*, dan *Median Filter*?

5. Bagaimana cara melakukan tranformasi citra (*image*) dengan menggunakan transformasi *fourier* dan tranformasi wavelet?

6. Bagaimana cara menganalisa bentuk citra menggunakan operasi matematika translasi dan reflikasi?

## 1.4 Manfaat

Manfaat yang dapat diambil dari laporan Praktikum Pengolahan Citra Digital ini berdasarkan rumusan masalah yang dipaparkan di atas antara lain adalah sebagai berikut.

1. Untuk mengetahui cara membaca dan menyalin sebuah file citra (*image*).

2. Untuk mengetahui cara mengolah citra ke dalam format ruang warna R, G, B, C, M, Y.

3. Untuk mengetahui cara mengimplementasikan algoritma dalam pengolahan citra berbasis titik seperti mengubah dari citra berwarna ke *grayscale,* mengubah citra berwarna ke biner, mengatur kecerahan (*brightness*), dan mengubah ke citra negasi.

4. Untuk mengetahui cara melakukan proses *filtering* dengan menggunakan *Low Pass Filter, High Pass Filter*, dan *Median Filter*.

5. Untuk mengetahui cara melakukan tranformasi citra (*image*) dengan menggunakan transformasi *fourier* dan tranformasi wavelet.

6. Untuk mengetahui cara menganalisa bentuk citra menggunakan operasi matematika translasi dan reflikasi.

## 1.5 Batasan Masalah

Adapun batasan masalah dalam penulisan laporan Praktikum Pengolahan Citra Digital yaitu agar pembahasan yang disampaikan tidak melenceng dari masalah yang dibahas. Laporan Praktikum Pengolahan Citra Digital akan membahas mengenai implementasi dalam membaca dan mengcopy file gambar kemudian mengubahnya menjadi ruang warna R, G, B, C, M, Y, mengubah citra berwarna menjadi *grayscale,* biner, mengatur kecerahan citra (*brightness*), mengubah citra negasi, *filtering* dengan mengunakan *Low Pass Filter, high Pass Filter* dan *Median Filter.*

## 1.6 Sistematika Penulisan

Sistematika penulisan yang digunakan pada laporan Praktikum Pengolahan Citra Digital ini terdiri dari pembahasan umum secara teori disertai dengan pemaparan contoh dari penjelasan tersebut. Pengambilan teori menggunakan metode pengumpulan data seperti artikel maupun jurnal yang nantinya akan dirangkum dari beberapa sumber.

### 1.6.1 Bab I Pendahuluan

Bab I merupakan pendahuluan dari laporan Praktikum Pengolahan Citra Digital. Laporan pada Bab I membahas tentang latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, dan sistematika penulisan yang diguanakn dalam proses penulisan laporan Praktikum Pengolahan Citra Digital.

### 1.6.2 Bab II Tinjauan Pustaka

Bab II merupakan tinjauan Pustaka dari laporan Praktikum Pengolahan Citra Digital. Pada bab ini menjelaskan mengenai landasan teori yang digunakan pada laporan Praktikum Pengolahan Citra Digital.

### 1.6.3 Bab III Metode Penelitian

Bab III merupakan metode penelitian dari laporan Praktikum Pengolahan Citra Digital. Pada bab ini akan menjelaskan mengenai tempat dan waktu penelitian, sumber data, serta alat dan bahan yang digunakan untuk mendukung pembuatan penelitian atau Praktikum Pengolahan Citra Digital.

### 1.6.4 Bab IV Hasil dan Pembahasan

Bab IV merupakan pembahasan dan hasil dari laporan Praktikum Pengolahan Citra Digital. Pada bab ini akan menjelaskan mengenai modul-modul yang sudah dipraktikan pada mata kuliah Praktikum Pengolahan Citra Digital.

### 1.6.5 Bab V Penutup

Bab V merupakan penutup dari laporan Praktikum Pengolahan Citra Digital. Pada bab ini akan menjelaskan mengenai kesimpulan-kesimpulan dari keseluruhan modul-modul Praktikum Pengolahan Citra Digital serta saran yang bisa pengempu berikan pada mata kuliah Praktikum Pengolahan Citra Digital.

# BAB II

# TINJAUAN PUSTAKA

Bab II merupakan bagian yang menjelaskan mengenai tinjauan Pustaka yang digunakan dalam laporan Praktikum Pengolahan Citra Digital. Tinjuan Pustaka meliputi materi-materi yang digunakan dalam mengerjalan setiap modul dalam Praktikum Pengolahan Citra Digital.

## 2.1 Dasar Teori

Pengolahan citra digital (*digitak image processing*) merupakan ilmu yang mempelajari hal-hal berkaitan dengan perbaikan kualitas terhadap suatu gambar (meningkatkan kontras, perubahan warna, restorasi citra), transformasi gambar (translasi, rotasi transformasi, skala, geometrik), melakukan pemilihan citra ciri (feature images) yang optimal untuk tujuan analisis, melakukan penyimpanan data yang sebelumnya dilakukan reduksi dan kompresi, transmisi data, dan waktu proses data. Selain itu, pengolahan citra digital merupakan suatu proses dengan memasukkan citra dan menghasilkan keluaran berupa citra seperti yang dikehendaki. Secara umum tahapan pengolahan citra digital meliputi akusisi citra, peningkatan kualitas citra, segmentasi citra, representasi dan uraian, pengenalan dan interpretasi.

## 2.2 Akuisisi Citra

Akuisisi citra merupakan proses pembentukan citra sebelum dilakukannya tahapan porses perbaikan citra. Pada pengolahan citra, akuisisi citra sangatlah penting. Akuisisi memiliki tujuan untuk menentukan data yang diperlukan dan memilih metode perekaman citra digital. Tahal ini dimulai dari objek yang akan diambil gambarnya, persiapa alata-alat, sampai pada pencitraan. Proses akuisisi citra adalah pemetaan suatu pandangan (*scene*) menjadi citra kontinu dengan menggunakan sensor.

### 2.2.1 Akuisisi Aktif

Sensor aktif merupakan jenis sensor yang mengirimkan dan menerima pantulan dari energi yang dikirim kea rah objek. Energi yag dikirmkan dapat berupa gelombang. Sensor aktif membutuhkan sumber daya eksternal untuk bisa beroperasi. Sifat fisik dari sensor artif bervariadi sehubungan dengan efek eksternal yang diberikan. Adapun contoh dari sensor aktif yaitu sensor *Rontgen* untuk foto *thorox,* sensor gelombang pendek pada sistem radar, sensor *ultrasound* pada sistem USG.

### 2.2.2 Akuisisi Pasif

Akuisis pasif merupakan sistem sensor yang merekam data objek tanpa mengirimkan energi, sumber energi bisa dalam bentuk sinar matahari, sinar lampu, dan lain sebagainya. Adapun contoh dari sensor pasif yaitu seperti sensor optic dari kamera foto, sensor optic pada sistem inderajaya.

## 2.3 Sampling

Sampling merupakan proses transformasi citra lanjutan ke citra digital atau bisa disingkat porses digitasi pada bidang. Selain itu sampling dapat dikatakan sebagai suatu proses untuk menentukan warna piksel dari gambar dua dimensi yang kontinyu menjadi gambar diskrit. Contoh pada gambar atau titik diskrit pada baris n dan kolom m disebut dengan piksel [n,m].

### 2.3.1 Sampling Uniform

Sampling *uniform* memiliki spasi (interval) baris dan kolom yang sama pada seluruh area sebuah citra. Proses sampling melalui celah yang berukuran sama. Kuantisasi Uniform mempunyai interval pengelompokan tingkat keabuan yang sama (misal: intensitas 1 s/d 10 diberi nilai 1, intensitas 11 s/d 20 diberi nilai 2).

### 2.3.2 Sampling Non-uniform

Sampling Non-uniform bersifat adaptif tergantung karakteristik citra dan bertujuan untuk menghindari adanya informasi yang hilang. Daerah citra yang mengandung detail yang tinggi di-sampling secara lebih halus, sedangkan daerah yang homogen dapat di-sampling lebih kasar. Kerugian sistem sampling Non-uniform adalah diperlukannya data ukuran spasi atau tanda batas akhir suatu spasi. Proses sampling melalui celah yg bervariasi. Kuantisasi Non-Uniform yang lebih halus diperlukan terutama pada bagian citra yang menggambarkan detail atau tekstur atau batas suatu wilayah obyek, dan kuantisasi yang lebih kasar diberlakukan pada wilayah yang sama pada bagian obyek.

## 2.4 Kuantisasi

Kuantisasi citra merupakan teknik pada pengelompokkan nilai tingkat keabuan suatu citra kontinyu ke dalam beberapa level. Selain itu, kuantitas merupakan sebuah proses pada skala keabuan [0,L] menjadi G buah level yang dinyatakan dengan suatu harga bilangan bulat. Kuantisasi dibagi menjadi tiga bagian yaitu sebagai berikut

### 2.4.1 Kuantisasi Uniform

Kuantisasi *uniform* merupakan suatu proses penyesuaian amplitude sinyal PAM ke dalam amplitude standar pengkodean. Dengan kata lain sinyal analog di transformasikan menjadi sinyal diskrit (digital). Kuantisasi *uniform* memiliki interval pengelompokan tingkat keabuan yang sama. Sebagai contoh yaitu intensitas 1 s/d 10 diberi nilai 1, intensitas 11 s/d 20 diberi nilai 2, dstnya.

### 2.4.2 Kuantisasi Non-uniform

Kuantitas *non-uniform* merupakan kuantisasi yang memberikan kode sinyal ke dalam level-level tertentu dengan jarak antar level yang tidak sama. Pada kuantisasi ini kesalahan sapat diminimalisir karena konsep dari metode ini merupakan jika amplitude kecil, maka pembagian jarak antar level juga semakin kecil.

### 2.4.3 Kuantisasi Tapered

Kuantisasi tempered merupakan jika ada daerah tingkat keabuan yang sering muncul, sebaiknya kuantisasi secara lebih halus dan diluar batas daerah tersebut dapat dikuantisasi secara lebih kasar (*local stretching*).

## 2.5 Format Ruang Warna

Warna merupakan hasil persepsi dari cahaya dalam spektrum wilayah yang terlihat oleh retina mata serta memiliki panjang gelombang antara 400 nm sampai 700nm. Ruang warna atau yang sering disebut dengan model warna merupakan sebuah cara atau metode untuk mengatur,membuat, dan memvisualisasikan warna. Ruang warna akan berbeda untuk aplikasi yang berbeda. Hal tersebut karena beberapa peralatan tertentu memang membatasi secara ketat ukuran dan jenis ruang warna yang bisa digunakan. Beberapa ruang warna tersebut seperti berikut.

### 2.5.1 RGB

RGB merupakan model warna yang berdasarkan pada konsep penambahan kuat cahaya primer yaitu *Red, Green,* dan *Blue*. Jika pada ruangan gelap, maka gelmbang cahaya yang diserap oleh mata atau RGB (0,0,0). Namun jika ruangan gelap tersebut ditambahkan cahaya merah, maka ruangan tersebut akan berubah warna menjadi merah misalnya RGB (255,0,0), semua benda dalam ruangan tersebut hanya dapat terlihat berwarna merah. Begitu juga ketika cahaya tersebut diganti dengan hijau atau biru.

Jika diberikan dua macam cahaya primer dalam sebuah ruangan seperti merah dan hijau, atau merah dan biru, atau hijau dan biru, maka ruangan akan berubah warna masing-masing menjadi kuning, atau magenta, daatu cyan. Warna-warna yang dibentuk o,eh kombinasi dua macam cahaya tersebut disebut warna sekunder. Warna tersier merupakan warna yang hanya dapat terlihat apabila ada tiga cahaya primer. Hal tersebut berarti jika dinonaktifkan salah satu warna, maka benda tersebut berubah warna. Contoh dari warna tersier yaitu abu-abu, putih. Jika RGB dikombinasikan maka terciptalah warna putoh inilah mengapa RGB disebut sebagai *additive color* atau warna pencahayaan. Warna RGB merupakan prinsip warna elektronik seperti televisi, monitor komputer, dan juka *scanner.* Maka dari itu, warna yang ditampilkan RGB selalu terang dan menyenangkan, karena di setiing untuk display monitor.

|  |
| --- |
|  |

**Gambar 2. 1** Model Warna RGB

Gambar 2.1 dalam RGB (r,g,b) masing-masing nilai antara 0 hingga 255 sesuai dengan urutan masing-masing yaitu yang pertama *Red,* yang kedua yaitu *Green*, dan ketiga yaitu *Blue*, dengan demikian masing-masing komponen ada 256 tingkat. Jika digabungkan maka ada 256 x 256 x 256 atau 16.777.216 kombinasi warna RGB yang dapat dibentuk.

### 2.5.2 HSV

Model warna HSV mencerminkan warna dalam terminology *Hue*, *saturation*, dan *value.* Gambar di bawah ini merupakan model warna dari HSV yang dapat dilihat pada gambar di bawah ini.

|  |
| --- |
| Ruang Warna Hue Saturation Value (HSV) serta Proses Konversinya - Kita  Informatika |

**Gambar 2. 2** Model Warna HSV

Gambar 2.2 menyatakan bahwa *Hue* menyatakan warna sebenarnya, seperti merah, violet, dan kuning yang digunakan untuk menentukan kemerahan (*redness*), kehijauan (greeness), dsb. *Saturation* menyatakan tingkat kemurnian suatu warna, yaitu mengindikasikan seberapa banyak warna putih diberikan pada warna. *Value* adalah kecefahan dari warna. Nilai dari *value* berkisar antara 0-100%. Jika nilainya 0 maka warna akan menjaid hitam. Maka semakin besar nilai maka semakin cerah dan muncul variasi-variasi baru dari warna tersebut.

.

### 2.5.3 CMYK

CMYK merupakan singkatan dari *cyan, magenta, yellow*, dan warna utamanya yaitu *black*-hitam, dan sering kali dijadikan referensi sebagai suatu proses pewarnaan dengan emmpergunakan empat warna yang merupakan bagian dari model pewarnaan yang sering dipergunakan dalam pencetakan berwarna.

|  |
| --- |
| Warna - Color: Model Warna CMYK |

**Gambar 2. 3** Model Warna CMYK

Gambar 2.3 merupakan empat kombinasi warna atau warna prose. Keempat warn aini lebih diutamakan pada proses percetakkan karena warna CMYK bisa tampil dengan seimbang pada latar belakang putih seperti kerta maupun bahan yang sering digunakan *printing* lainnya.

Warna hitam memiliki fungsi untuk mengatur kontras atau kecerahan suatu warna. *Cyan* merupakan *secondary* dari warna hijau dan biru. Magenta merupakan *secondary* dari warna merah dan biru. Sedangkan kuning merupakan *secondary* dari warna merah dan hijau.

### 2.5.4 YIQ

YIQ merupakan ruang warna yang digunakanoleh TV berwarna NTSC, digunakan Sebagian besar di Amerika Utara dan Jepang. Komponen Y (luma) merupakan satu-satunya komponen yang digunakan oleh televisi hitam-putih. I dan O mewakili informasi *chrominance* (sinyal yang digunakan dalam video sistem untuk menyempaikan informasi warna dari gambar).

|  |
| --- |
| Model Warna YIQ |

**Gambar 2. 4** Model Warna YIQ

Gambar 2.4 merupakan ruangan warna YIQ. Masing-masing dari unsur Y, I, dan Q nilainya diukur antara 0-255. Unsur oada Y berupa nilai *Iuminance,* sedangkan I dan Q antara 0 yang mewakili warna hijau dan 255 yang mewakili unsur warna yang mirip dengan magenta.

## 2.6 Perbaikan Kualitas Citra

### 2.6.1 Operasi Titik

2.6.1.1 Mengubah Citra Berwarna ke *Grayscale*

2.6.1.2 Thresholding

2.6.1.3 Brightness Adjustment

2.6.1.4 Operasi Negasi

### 2.6.2 Operasi Spasial

2.6.2.1 Neighborhood Averaging

2.6.2.2 Median Filtering

2.6.2.3 High-Pass Filtering

### 2.6.3 Filtering

2.6.3.1 Low-Pass Filter

2.6.3.2 High-Pass Filter

2.6.3.3 Band-Pass Filter

## 2.7 Transformasi Citra

### 2.7.1 Transformasi Fourier

2.7.1.1 DFT (*Discrete Fourier Transform*)

2.7.1.2 DCT (*Discrete Cosine Transform*)

2.7.1.3 FFT (*Fast Fourier Transform*)

### 2.7.2 Transformasi Wavelet

2.7.2.1 *Discrete Wavelet Transform*

# BAB III

# METODOLOGI PENELITIAN

Bab III membahas mengenai metodologi penelitian dari laporan Praktikum Pengolahan Citra Digital. Pada bab ini akan membahas mengenai tempat dan waktu penelitian, sumber data, serta alat dan bahan yang akan mendulung pembuatan penelitian atau Praktikum Pengolahan Citra Digital ini,

## 3.1 Tempat dan Waktu Penelitian

Penelitian pada Praktikum Pengolahan Citra Digital ini dilaksanakan secara online menggunakan platform Google Meet serta di sekitar lingkungan Fakultas Teknik Universitas Udayana, khususnya di jurusan Teknologi Informasi dengan kurun waktu pengerjaan selama 4 bulan.

## 3.2 Gambar Umum Sistem

Gambaran umum sistem terdiri atas penjelasan umum dari seluruh program di setiap modul serta metode yang digunakan dalam menjalankan program dengan benar sesuai dengan fungsinya

### 3.2.1 Dasar Pengolahan Citra

Pengolahan citra digital merupakan pemrosesan pada suatu citra dua dimensi mengguakan komputer. Modul 1 melakukan penelitian terhadap Dasar Pengolahan Citra Digital, sehingga dibuatkan program menggunakan bahasa Python dan Matlab untuk menjalankan fungsi membaca dan meng-*copy* citra, lalu citra tersebut diubah ke dalam ruangan warna R,G,B,C,M,Y. Gambar di bawah ini akan menjelasakn diagram yang menunjukkan aktivitas *user* dalam modul I Dasar Pengolahan Citra Digital.

|  |
| --- |
|  |

**Gambar 3. 1** Diagram Dasar Pengolahan Citra Digital

Gambar 3.1 merupakan diagram dasar pada pengolahan citra digital. Pada gambar, *user* pertama akan membuka *file* citra, lalu menyalin citra tersebut lalu memilih pilihan *filter* (*red, green, blue, cyan, magenta,* dan *yellow*) yang ingin di aplikasikan pada citra. Kemudian jika ingin keluar dari program dapat meneken *button exit.*

### 3.2.2 Perbaikan Kualitas Citra

Perbaikan kualitas citra merupakan proses memperbaiki kualitas citra yang mengalami degradasi sehingga memiliki kualitas yang kurang baik. Perbaikan kualitas citra memiliki diagram yang akan menggambarkan bagaimana seorang *user* menggunakan program aplikasi perbaikan kualitas citra. Gambar di bawah ini akan menjelasakn diagram yang menunjukkan aktivitas penggunaan program oleh *user*.

|  |
| --- |
|  |

**Gambar 3. 2** Diagram Dasar Perbaikan Kualitas Citra

Gambar 3.2 merupakan diagram *use case* dari perbaikan kualitas citra. Pada gambar, *user* pertama-tama akan meng-*input*kan gambar, lalu mengubahnya menjadi citra *grayscale*, mengatur *brightness*, menambahkan *noise*, dan mengubahnya menjadi citra biner setelah diubah citra biner maupun citra yang sudah ditambahkan *noise* akan dibelikan *filter*,dimana *filter* dapat dipilih diantaranya LPH, HPF, dan *Median Filter* sesuai ukurab kernel yang diinginkan. Setelah itu, jika ingin keluar dari program, dapat menekan *button exit.*

### 3.2.3 Transformasi Citra

Transformasi citra merupakan proses pengubahan suatu citra dari suatu Kawasan ke Kawasan lainnya. Transformasi citra dapat dijadikan alternatif dalam representasi citra serta banyak jenis tranfromasi yang dapat digunakan untuk mengubah dari kawaban ruang ke kawaban frekuensi. Berikut merupakan aktivitas penggunaan program oleh *user.*

|  |
| --- |
|  |

**Gambar 3. 3** Diagram Transformasi Citra

Gambar 3.3 merupakan diagram dari *use case* dari program transformasi citra. Pada gambar, *user* pertama akan mengunggah gambar, lalu diuah ke citra *grayscale*. Setelah itu memilih transformasi yang diinginkan seperti Wavelet, DCT, DFT, DDT, invers DFT, invers FFT, invers DCT, dan invers wavelet.

### 3.2.4 Morfologi dan Segmentasi Citra

Proses segmentasi citra adalah tahapan mengelompokkan *region*-*region* di dalam suatu citra. Pengelompokkan tersebut dilakukan agar bisa dilakukan analisis pada *region* tertentu saja. Sementara jika morfologi merupakan teknik pengolahan citra digital dengan menggunakan bentuk objek sebagai pedoman dalam pengolahan. Morfologi dan segmentasi citra memiliki diagram yang akan menggambarkan bagaimana *user* menggunakan program aplikasi morfologi dan segmentasi citra. Berikut merupakan gambar aktivitas penggunaan program oleh *user.*

|  |
| --- |
|  |

**Gambar 3. 4** Diagram Morfologi dan Segmentasi Citra

Gambar 3.4 merupakan diagram dari *use case* program morfologi dan segmentasi citra. Pada gambar *user* pertama kali akan mengunggah gambar yang sudah diubah citranya menjadi citra biner. Lalu *user* akan memilih proses yang akan dijalankan, bisa proses morfologi atau segmentasi. Setelah itu jika *user* ingin ekluar dari program, maka dapat menekan *button exit.*

# BAB IV

# HASIL DAN PEMBAHASAN

Bab IV berisi hasil dan pembahasan pada laporan Praktikum Pengolahan Citra Digital yang terdiri dari dasar pengolahan citra, perbaikan kualitas citra, transformasi citra, serta morfologi dan segmentasi citra. Masing-masing percobaan dibuat sesuai dengan hasil yang di dapat selama praktikum berlangsung. Hasil dan pembahasan dari masing-masing percobaan praktikum adalah sebagai berikut.

## 4.1 Dasar Pengolahan Citra

Modul pertama pada praktikum Pengolahan Citra Digital membahas mengenai dasar dari pengolahan citra. Tujuannya yaitu untuk mempelajari serta mengimplementasikan bagaimana cara membaca dan meng-*copy* file gambar yang kemudian diubah ke ruang warna R,G,B,C,M, dan Y. Berikut merupakan implementasi dari modul pertama menggunakan bahasa pemrograman Python.

### 4.1.1. Source Code

*Source code* dari modul 1 berisi kode program untuk membaca dan meng-*copy* file gambar yang kemudian diubah ke ruang warna R,G,B,C,M, dan Y. Berikut merupakan kode program dari modul 1.

4.1.1.1 Python

Implementasi modul pertama dengan Python berupa form yang terdiri dari tiga *picture box,* 3 *button* (*Open, Copy,* dan *Exit*), dan enam *radio button* (R,G,B,C,M, dan Y). penjelasan dari implementasi modul pertama adalah sebagai berikut.

|  |
| --- |
| from ctypes.wintypes import RGB  from PyQt5 import QtCore, QtGui, QtWidgets  from PyQt5.QtGui import QPixmap, QImage, QColor, QPixelFormat  from PyQt5.QtWidgets import QFileDialog, QMessageBox, QAction, QMainWindow, QSlider, QPushButton, QToolTip, QApplication  from PIL import Image, ImageOps, ImageEnhance, ImageColor  from PIL.ImageQt import ImageQt  import cv2  import numpy as np  class Ui\_MainModul1(object):  def setupUi(self, MainModul1):  MainModul1.setObjectName("MainModul1")  MainModul1.resize(662, 595)  self.centralwidget = QtWidgets.QWidget(MainModul1)  self.centralwidget.setObjectName("centralwidget")  self.btnCopy = QtWidgets.QPushButton(self.centralwidget)  self.btnCopy.setGeometry(QtCore.QRect(350, 230, 75, 23))  self.btnCopy.setObjectName("btnCopy")  self.btnGreen = QtWidgets.QRadioButton(self.centralwidget)  self.btnGreen.setGeometry(QtCore.QRect(130, 510, 80, 18))  self.btnGreen.setObjectName("btnGreen")  self.btnOpen = QtWidgets.QPushButton(self.centralwidget)  self.btnOpen.setGeometry(QtCore.QRect(230, 230, 75, 23))  self.btnOpen.setObjectName("btnOpen")  self.btnCyan = QtWidgets.QRadioButton(self.centralwidget)  self.btnCyan.setGeometry(QtCore.QRect(400, 510, 80, 18))  self.btnCyan.setObjectName("btnCyan")  self.gambarRGB = QtWidgets.QLabel(self.centralwidget)  self.gambarRGB.setGeometry(QtCore.QRect(70, 350, 191, 151))  self.gambarRGB.setAutoFillBackground(False)  self.gambarRGB.setFrameShape(QtWidgets.QFrame.Panel)  self.gambarRGB.setFrameShadow(QtWidgets.QFrame.Raised)  self.gambarRGB.setText("")  self.gambarRGB.setScaledContents(True)  self.gambarRGB.setObjectName("gambarRGB")  self.btnYellow = QtWidgets.QRadioButton(self.centralwidget)  self.btnYellow.setGeometry(QtCore.QRect(540, 510, 80, 18))  self.btnYellow.setObjectName("btnYellow")  self.gambarAsli = QtWidgets.QLabel(self.centralwidget)  self.gambarAsli.setGeometry(QtCore.QRect(200, 70, 261, 151))  self.gambarAsli.setAutoFillBackground(False)  self.gambarAsli.setFrameShape(QtWidgets.QFrame.Panel)  self.gambarAsli.setFrameShadow(QtWidgets.QFrame.Raised)  self.gambarAsli.setText("")  self.gambarAsli.setScaledContents(True)  self.gambarAsli.setObjectName("gambarAsli")  self.gambarCMY = QtWidgets.QLabel(self.centralwidget)  self.gambarCMY.setGeometry(QtCore.QRect(400, 350, 191, 151))  self.gambarCMY.setAutoFillBackground(False)  self.gambarCMY.setFrameShape(QtWidgets.QFrame.Panel)  self.gambarCMY.setFrameShadow(QtWidgets.QFrame.Raised)  self.gambarCMY.setText("")  self.gambarCMY.setScaledContents(True)  self.gambarCMY.setObjectName("gambarCMY")  self.btnBlue = QtWidgets.QRadioButton(self.centralwidget)  self.btnBlue.setGeometry(QtCore.QRect(190, 510, 80, 18))  self.btnBlue.setObjectName("btnBlue")  self.btnMagenta = QtWidgets.QRadioButton(self.centralwidget)  self.btnMagenta.setGeometry(QtCore.QRect(460, 510, 80, 18))  self.btnMagenta.setObjectName("btnMagenta")  self.labelCitraAsli = QtWidgets.QLabel(self.centralwidget)  self.labelCitraAsli.setGeometry(QtCore.QRect(290, 30, 81, 21))  self.labelCitraAsli.setFrameShape(QtWidgets.QFrame.StyledPanel)  self.labelCitraAsli.setAlignment(QtCore.Qt.AlignCenter)  self.labelCitraAsli.setObjectName("labelCitraAsli")  self.btnRed = QtWidgets.QRadioButton(self.centralwidget)  self.btnRed.setGeometry(QtCore.QRect(80, 510, 80, 18))  self.btnRed.setObjectName("btnRed")  self.btnExit = QtWidgets.QPushButton(self.centralwidget)  self.btnExit.setGeometry(QtCore.QRect(290, 260, 75, 23))  self.btnExit.setObjectName("btnExit")  self.labelCitraRGB = QtWidgets.QLabel(self.centralwidget)  self.labelCitraRGB.setGeometry(QtCore.QRect(120, 310, 81, 21))  self.labelCitraRGB.setFrameShape(QtWidgets.QFrame.StyledPanel)  self.labelCitraRGB.setAlignment(QtCore.Qt.AlignCenter)  self.labelCitraRGB.setObjectName("labelCitraRGB")  self.labelCitraCMY = QtWidgets.QLabel(self.centralwidget)  self.labelCitraCMY.setGeometry(QtCore.QRect(460, 310, 81, 21))  self.labelCitraCMY.setFrameShape(QtWidgets.QFrame.StyledPanel)  self.labelCitraCMY.setAlignment(QtCore.Qt.AlignCenter)  self.labelCitraCMY.setObjectName("labelCitraCMY")  MainModul1.setCentralWidget(self.centralwidget)  self.menubar = QtWidgets.QMenuBar(MainModul1)  self.menubar.setGeometry(QtCore.QRect(0, 0, 662, 21))  self.menubar.setDefaultUp(True)  self.menubar.setObjectName("menubar")  self.menuModul1 = QtWidgets.QMenu(self.menubar)  self.menuModul1.setObjectName("menuModul1")  MainModul1.setMenuBar(self.menubar)  self.statusbar = QtWidgets.QStatusBar(MainModul1)  self.statusbar.setObjectName("statusbar")  MainModul1.setStatusBar(self.statusbar)  self.actionNew = QtWidgets.QAction(MainModul1)  self.actionNew.setObjectName("actionNew")  self.menuModul1.addAction(self.actionNew)  self.menuModul1.addSeparator()  self.menubar.addAction(self.menuModul1.menuAction())  self.retranslateUi(MainModul1)  self.btnOpen.clicked.connect(self.loadImage)  self.btnCopy.clicked.connect(self.copyImage)  # self.exitButton.clicked.connect(lambda:self.close())  self.btnRed.clicked.connect(self.convertRed)  self.btnGreen.clicked.connect(self.convertGreen)  self.btnBlue.clicked.connect(self.convertBlue)  self.btnCyan.clicked.connect(self.convertCyan)  self.btnMagenta.clicked.connect(self.convertMagenta)  self.btnYellow.clicked.connect(self.convertYellow)  QtCore.QMetaObject.connectSlotsByName(MainModul1)  def retranslateUi(self, MainModul1):  \_translate = QtCore.QCoreApplication.translate  MainModul1.setWindowTitle(\_translate("MainModul1", "Praktikum Pengolahan Citra Digital"))  self.btnCopy.setText(\_translate("MainModul1", "Copy"))  self.btnGreen.setText(\_translate("MainModul1", "Green"))  self.btnOpen.setText(\_translate("MainModul1", "Open"))  self.btnCyan.setText(\_translate("MainModul1", "Cyan"))  self.btnYellow.setText(\_translate("MainModul1", "Yellow"))  self.btnBlue.setText(\_translate("MainModul1", "Blue"))  self.btnMagenta.setText(\_translate("MainModul1", "Magenta"))  self.labelCitraAsli.setText(\_translate("MainModul1", "Citra Asli"))  self.btnRed.setText(\_translate("MainModul1", "Red"))  self.btnExit.setText(\_translate("MainModul1", "Exit"))  self.labelCitraRGB.setText(\_translate("MainModul1", "Citra RGB"))  self.labelCitraCMY.setText(\_translate("MainModul1", "Citra CMY"))  self.menubar.setStatusTip(\_translate("MainModul1", "Modul1"))  self.menuModul1.setTitle(\_translate("MainModul1", "File"))  self.actionNew.setText(\_translate("MainModul1", "New")) |

**Kode Program 4. 1** Source Code untuk Tampilan Awal

Kode program 4.1 merupakan kode program yang menjelaskan sintaks tampilan awal program. Program tersebut menampilkan sebanyak tiga buah *frame.* Pada tampilan awal, terdapat opsi untuk *file*, opsi salin citra, opsi konversi citra, dan opsi menu.

|  |
| --- |
| def loadImage(self):  self.namaImg, typeimg= QFileDialog.getOpenFileName(self.centralwidget, "Open Image", "", ".png;.jpg;;.bmp;;All Files()")  jpg = QPixmap(self.namaImg)  self.gambarAsli.setPixmap(jpg) |

**Kode Program 4. 2** Source Code untuk Load Image

Kode program 4.2 merupakan kode program yang menjelaskan sintaks untuk otombol *open*. Ketika tombol ditekan, maka program akan menampilkan *dialog box* untuk memilih *file* gambar. *File* gambar yang dapat dipilih merupakan gambar dengan ekstensi .jpg, .bmp, dll. Ketika *file* gambar sudah dipilih, maka program akan menampilkan gambar pada *picture box* 1.

|  |
| --- |
| def copyImage(self):  img = Image.open(self.namaImg)  width, height= img.size  for row in range(height):  for col in range(width):  img.getpixel((col,row))  img.save("copyImage.jpg")  jpg=QPixmap("copyImage.jpg")  self.gambarRGB.setPixmap(jpg) |

**Kode Program 4. 3** Source Code unuk Tombol Copy

Kode program 4.3 menjelaskan mengenai sintaks tombol *copy.* Ketika tombol ditekan, maka program akan mengambil nilai R, G, dan B dari tiap pixel gambar pada *picture box* 1. Lalu program akna menampilkan gambar yang telah disalin ke *picture box*  2.

|  |
| --- |
| def convertRed(self):  img = cv2.imread(self.namaImg, cv2.IMREAD\_UNCHANGED)  print(img.shape)  red\_channel = img[:,:,2]  red\_img = np.zeros(img.shape)  red\_img[:,:,2] = red\_channel  cv2.imwrite("convertResult.jpg", red\_img)  jpg=QPixmap("convertResult.jpg")  self.gambarRGB.setPixmap(jpg) |

**Kode Program 4. 4** Source Code untuk Radio Button R

Kode program 4.4 menjelaskan sintaks untuk *radio button* R. ketika *radio button* R ditekan, maka program akna mengambil tiap pixel gambar pada *picture box* 2 dan mengubah nilai alfa R menjadi maksimum (255) dan nilai G dan B menjadi minimum (0). Kemudian program akan menampilkan gambar yang memiliki warna dominan merah dan ditampilkan pada *picture box* 3.

|  |
| --- |
| def convertGreen(self):  img = cv2.imread(self.namaImg, cv2.IMREAD\_UNCHANGED)  print(img.shape)  #BGR, 0=blue, 1=Greed, 2=Blue  green\_channel = img[:,:,1]  green\_img = np.zeros(img.shape)  green\_img[:,:,1] = green\_channel  cv2.imwrite("convertResult.jpg", green\_img)  jpg=QPixmap("convertResult.jpg")  self.gambarRGB.setPixmap(jpg) |

**Kode Program 4. 5** Source Code untuk Radio Button G

Kode program 4.5 menjelaskan sintaks untuk *radio button* G. ketika menekan tombol, maka program akan mengambil tiap pixel gambar pada *picture box* 2 dan mengubah nilai alfa G menjadi maksimum (255) dan nilai R dan B menjadi minimum (0). Lalu program akan menampilkan gambar yang memiliki warna dominan hijau dan ditampilkan pada *picture box* 3.

|  |
| --- |
| def convertBlue(self):  img = cv2.imread(self.namaImg, cv2.IMREAD\_UNCHANGED)  print(img.shape)  blue\_channel = img[:,:,0]  blue\_img = np.zeros(img.shape)  blue\_img[:,:,0] = blue\_channel  #save image  cv2.imwrite("convertResult.jpg", blue\_img)  jpg=QPixmap("convertResult.jpg")  self.gambarRGB.setPixmap(jpg) |

**Kode Program 4. 6** Source Code untuk Radio Button B

Kode program 4.6 menjelaskan sintaks untuk *radio button* B. ketika menekan tombol, maka program akan mengambil tiap pixel gambar pada *picture box* 2 dan mengubah nilai alfa B menjadi maksimum (255) dan nilai R dan G menjadi minimum (0). Lalu program akan menampilkan gambar yang memiliki warna dominan biru dan ditampilkan pada *picture box* 3.

|  |
| --- |
| def convertCyan(self):  img = cv2.imread(self.namaImg, cv2.IMREAD\_UNCHANGED)  print(img.shape)    img[:,:,2] = np.zeros([img.shape[0], img.shape[1]])  #save image  cv2.imwrite("convertResult.jpg", img)  jpg=QPixmap("convertResult.jpg")  self.gambarCMY.setPixmap(jpg) |

**Kode Program 4. 7** Source Code untuk Radio Button C

Kode program 4.7 menjelaskan sintaks untuk *radio button* C. Ketika menekan tombol, maka program akan mengambil tiap pixel gambar pada *picture box* 2 dan mengubah nilai alfa G dan B menjadi maksimum (255) dan nilai R menjadi minimum (0). Lalu program akan menampilkan gambar yang memiliki warna dominan cyan dan ditampilkan pada *picture box* 3.

|  |
| --- |
| def convertMagenta(self):  img = cv2.imread(self.namaImg, cv2.IMREAD\_UNCHANGED)  print(img.shape)  img[:,:,1] = np.zeros([img.shape[0], img.shape[1]])  #save image  cv2.imwrite("convertResult.jpg", img)  jpg=QPixmap("convertResult.jpg")  self.gambarCMY.setPixmap(jpg) |

**Kode Program 4. 8** Source Code untuk Radio Button M

Kode program 4.8 menjelaskan sintaks untuk *radio button* M. Ketika menekan tombol, maka program akan mengambil tiap pixel gambar pada *picture box* 2 dan mengubah nilai alfa R dan B menjadi maksimum (255) dan nilai G menjadi minimum (0). Lalu program akan menampilkan gambar yang memiliki warna dominan Magenta (ungu) dan ditampilkan pada *picture box* 3.

### 4.1.2 Langkah-Langkah Percobaan

Langkah-langkah percobaan sistem diperlukan untuk mengetahui bagaimana sistem suatu aplikasi dijalanlan. Maka dari itu, perlu diadakan uji coba sistem pada dasar pengolahan citra.

4.1.2.1 MATLAB

Langkah-langkah percobaan dibagi menjadi dua yaitu tampilan awal dan proses utama program. Langkah-langkah percobaan pada program MATLAB yaitu sebagai berikut.

a. Tampilan Awal Program Dasar Pengolahan Citra

Tampilan awal program dasar pengolahan citra memiliki sebiah form yang terdiri dari tiga *Picture Box* dimana *Picture Box* satu digunakan untuk menampikkan *image* asli, *Picture Box* dua untuk menampilkan hasil salin dan *Picture Box* tiga untuk menampilkan *image* ruang warna.

|  |
| --- |
|  |

**Gambar 4. 1** Tampilan Awal Dasar Pengolahan Citra

Gambar 4.1 merupakan tampilan awal program Dasar Pengolahan Citra menggunakan MATLAB. Pada gambar terdapat tiga *Picture Box,* tiga tombol, dan enam *radio button*. Citra nantinya akan ditampilkan pada ketiga *picture box* tersebut ketika tombol tertentu ditekan.

b. Proses Utama

Proses utama program diawali dengan langkah pertama yaitu menginputkan gambar ke dalam program dengan cara menekan tombol *open.* Ketika menekan tombol *open* ditekan, maka program akan menampilkan *dialog box* untuk memilih *file* gambar, proses utama tersebut dimulai pada gambar 4.2.

|  |
| --- |
|  |

**Gambar 4. 2** Memilih File Gambar

Gambar 4.2 merupakan tampilan dari *dialog box* untuk memilih gambar uang akan digunakan. *File* gambar yang dapat dipilih adalah *file* gambar dengan ekstensi *file* .jpg, .png, dan .bmo. proses selanjutnya dapat dilihat pada gambar 4.3

|  |
| --- |
|  |

**Gambar 4. 3** Proses Menampilkan Gambar pada Picture Box 1

Gambar 4.3 merupakan proses menampilkan gambar pada *picture box 1.* Ketika *file* gambar terlah dipilih, maka gambar (citra) asli akan ditampilkan pada *picture box* 1. Berikut merupakan proses ketika gambar di *copy.*

|  |
| --- |
|  |

**Gambar 4. 4** Gambar di copy pada Picture Box 2

Gambar 4.4 merupakan proses ketika gambar di-*copy* pada *picture box* 2. Proses ini terjadi ketika *user* menekan tombol *copy.* Proses selanjunya adalah mengubah gambar menjadi ruangan warna sesuai dengan *radio button* yang ditekan.

|  |
| --- |
|  |

**Gambar 4. 5** Ruang Warna Red

Gambar 4.5 merupakan hasil gambar yang diberi ruang warna *red*. Gambar yang diberi ruang warna *red* akan terjadi perubahan warna dominan merah. Tahapan selanjutnya gambar akan diberi ruang warna *green* yang dapat dilihat pada gambar 4.6.

|  |
| --- |
|  |

**Gambar 4. 6** Ruang Warna Green

Gambar 4.6 merupakan hasil gambar yang diberi ruang warna *green*. Gambar yang diberi ruang warna *green* akan terjadi perubahan warna dominan hijau. Tahapan selanjutnya gambar akan diberi ruang warna *blue* yang dapat dilihat pada gambar 4.7.

|  |
| --- |
|  |

**Gambar 4. 7** Ruang Warna Blue

Gambar 4.7 merupakan hasil gambar yang diberi ruang warna *blue*. Gambar yang diberi ruang warna *blue* akan terjadi perubahan warna dominan biru. Tahapan selanjutnya gambar akan diberi ruang warna *cyan* yang dapat dilihat pada gambar 4.8.

|  |
| --- |
|  |

**Gambar 4. 8** Ruang Warna Cyan

Gambar 4.8 merupakan hasil gambar yang diberi ruang warna *cyan*. Gambar yang diberi ruang warna *cyan* akan terjadi perubahan warna dominan *cyan*. Tahapan selanjutnya gambar akan diberi ruang warna *magenta* yang dapat dilihat pada gambar 4.9.

|  |
| --- |
|  |

**Gambar 4. 9** Ruang Warna Magenta

Gambar 4.9 merupakan hasil gambar yang diberi ruang warna *magenta*. Gambar yang diberi ruang warna *magenta*akan terjadi perubahan warna dominan *magenta*. Tahapan selanjutnya gambar akan diberi ruang warna *yellow* yang dapat dilihat pada gambar 4.10.

|  |
| --- |
|  |

**Gambar 4. 10** Ruang Warna Yellow

Gambar 4.10 merupakan hasil gambar yang diberi ruang warna *yellow*. Gambar yang diberi ruang warna *yellow*akan terjadi perubahan warna dominan *yellow*. Tahapan selanjutnya merupakan menekan tombol *exit* dan keluar dari program yang dapat dilihat pada gambar 4.11

### 4.1.3 Hasil Uji Program Dasar Pengolahan Citra

# BAB V

# PENUTUP

## 5.1 Simpulan

Simpulan yang dapat diambil berdasarkan pembuatan laporan praktikum pemgolahan citra digital adalah sebagai berikut.

1. Pada Python, hasil analisis menenai perbaikan kualitas citra seperti pengujian *grayscale* dan biner bahwa penggunaan pemrograman pada Python memiliki hasil kerja yang lebih baik dari pada bahasa pemrograman lainnya.

2.

## 5.2 Saran

Saran yang dapat diberikan dari pembuatan laporan praktikum Pengolahan Citra Digital yaitu:

1. praktikum citra digital yang dilakukan telah berjalan dengan lancar, meskipun masih terdapat beberapa kendala dalam pembuatan aplikasi.

2. Praktikan lanjutan sebaiknya lebih memperhatikan penggunaan aplikasi Python, Virtual Basic, dan Matlab agar lebih mempermudah dalam pembuatan program san mendapatkan hasil yang maksimal.

3. Praktikan lanjutan sebaiknya menggunakan bahasa pemrograman selain Python agar mendapatkan hasil yang lebih beragam.