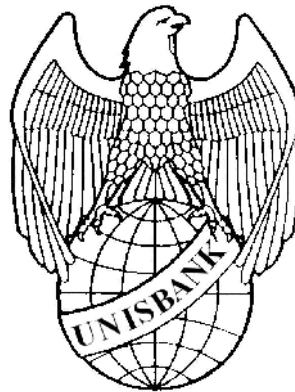


**IMPLEMENTASI METODE IMAGE SUBTRACTING UNTUK
MENDETEKSI GERAKKAN OBJEK DENGAN WARNA PADA FILE
VIDEO**

Tugas Akhir disusun untuk memenuhi syarat
Mencapai gelar Kesarjanaan Komputer pada
Program Studi Teknik Informatika
Jenjang Program Strata-1



oleh: SITO

RESMI

08.01.53.0086
10376

**FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS STIKUBANK (UNISBANK)**

SEMARANG

2013

PERNYATAAN KESIAPAN UJIAN SKRIPSI

Saya, Sito resmi , dengan ini menyatakan bahwa Laporan Skripsi yang berjudul:

Implementasi Metode Image Subtracting Untuk Mendeteksi Gerakan Objek Dengan Warna Pada File Video

Adalah benar hasil karya saya dan belum pernah diajukan sebagai karya ilmiah, sebagian atau seluruhnya, atas nama saya atau pihak lain.

(Sito Resmi) NIM :
08.01.53.0086

Disetujui oleh Pembimbing
Kami setuju Laporan tersebut diajukan untuk Ujian Skripsi

Semarang : 12 Pebruari 2013

(Eka Ardianto, S.kom, M.Cs)
Pembimbing

HALAMAN PENGESAHAN

Telah dipertahankan di depan tim dosen penguji Tugas Akhir Fakultas Teknologi Informasi, Universitas STIKUBANK (UNISBANK) Semarang dan diterima sebagai salah satu syarat guna menyelesaikan jenjang Program Strata 1, Program Studi Teknik Informatika.

Semarang :

Ketua

Sekretaris

Anggota

MENGETAHUI :
UNIVERSITAS STIKUBANK (UNISBANK) SEMARANG

Fakultas Teknologi Informasi
Dekan

Dwi Agus Diartono, S.Kom, M.Kom

MOTTO DAN PERSEMBAHAN

MOTTO

- ✎ Hidup adalah perjuangan
- ✎ Sabar menghadapi masalah dan bersyukur merupakan salah satu pedoman hidup manusia
- ✎ Orang yang bahagia adalah orang yang dijauhkan dari fitnah dan bila dtimpa ujian serta cobaan ia selalu bersabar
- ✎ Lupakan kesempurnaan, dan cobalah mengejar kesempurnaan
- ✎ Sesungguhnya setelah mengalami kesulitan selalu ada jalan kemudahan
- ✎ Tiada hari tanpa instropeksi diri
- ✎ buku merupakan jendela informasi dunia

PERSEMBAHAN

1. Allah S.W.T
2. Orang tua dan keluarga tercinta
3. Teman-teman Teknik Informatika.
4. Sahabat–sahabat dan saudara –saudara yang memberi semangat.

**FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS STIKUBANK (UNISBANK) SEMARANG**

Program Studi : Teknik Informatika
Tugas Akhir Sarjana Komputer
Semester Ganjil Tahun 2013

**Aplikasi Informasi Cuaca Lokasi Pariwisata di Wilayah Jawa Tengah
Berkas SMS Gateway**

**Alhuda Robby I
NIM : 08.01.53.0151**

Abstrak

Minimnya informasi kepariwisataan di Jawa Tengah yang bisa diakses dengan SMS dan *up-to-date* membuat potensi pariwisata di Jawa Tengah kurang begitu dikenal oleh wisatawan domestik maupun mancanegara. Penyediaan informasi mengenai tempat pariwisata dan cuaca pariwisata pada website pariwisata pun sangat sedikit sehingga calon wisatawan harus mencari informasi mengenai tempat pariwisata pada tempat lain.

Adapun tujuan yang akan dicapai adalah untuk membangun aplikasi informasi cuaca lokasi pariwisata di wilayah Jawa Tengah berbasis SMS *gateway* yang dapat memberikan informasi dan jawaban tentang informasi tempat pariwisata dan cuaca pariwisata di Jawa Tengah kepada calon wisatawan secara cepat, tepat dan akurat.

Hasil dalam penelitian ini adalah aplikasi informasi cuaca lokasi pariwisata di wilayah Jawa Tengah berbasis SMS Gateway dapat memudahkan calon wisatawan untuk mengetahui informasi mengenai tempat pariwisata dan cuaca pariwisata yang dibutuhkan hanya dengan mengirimkan sms sesuai dengan format yang ditentukan.

Kata Kunci

Cuaca, Jawa Tengah, SMS Gateway

Semarang : 12 Pebruari 2013

Pembimbing

(Dwi Agus Diartono, S.Kom, M.Kom)

KATA PENGANTAR

Dengan memanjatkan puji syukur kehadiran Allah SWT. yang telah melimpahkan segala rahmat, hidayah dan inayah-Nya kepada penulis sehingga laporan tugas akhir dengan judul **“Aplikasi Informasi Cuaca Lokasi Pariwisata di Wilayah Jawa Tengah Berbasis SMS Gateway”** dapat penulis selesaikan sesuai dengan rencana karena dukungan dari berbagai pihak yang tidak ternilai besarnya. Oleh karena itu penulis menyampaikan terimakasih kepada:

1. Dr. Bambang Suko Priyono, MM selaku Rektor Universitas Stikubank Semarang.
2. Dwi Agus Diartono, S.Kom, M.Kom, selaku Dekan Fakultas Teknologi Informasi.
3. Dewi Handayani UN, S.Kom, M.Kom selaku Ka. Progdi Teknik Informatika.
4. Dwi Agus Diartono, S.Kom, M.Kom selaku Pembimbing yang telah membantu dan memberikan bimbingan serta pengarahan dalam penulisan Laporan Tugas Akhir ini.
5. Dosen-dosen pengampu di Program Studi Teknik Informatika Fakultas Teknologi Informasi Universitas Stikubank Semarang yang telah memberikan ilmu dan pengalamannya masing-masing, sehingga penulis dapat mengimplementasikan ilmu yang telah disampaikan.

Semoga Allah SWT memberikan balasan yang lebih besar kepada beliau-beliau, dan pada akhirnya penulis berharap bahwa penulisan laporan tugas akhir ini dapat bermanfaat dan berguna sebagaimana fungsinya.

Semarang, 12 Pebruari 2013

Penulis

DAFTAR ISI

| | |
|--|-----|
| HALAMAN JUDUL | i |
| HALAMAN PERSETUJUAN | ii |
| HALAMAN PENGESAHAN | iii |
| MOTTO DAN PERSEMBAHAN..... | iv |
| ABSTRAKSI..... | v |
| KATA PENGANTAR | vi |
| DAFTAR ISI | vii |
| DAFTAR TABEL | x |
| DAFTAR GAMBAR..... | xi |
| | |
| BAB I PENDAHULUAN | |
| 1.1 Latar Belakang Masalah..... | 1 |
| 1.2 Perumusan Masalah | 2 |
| 1.3 Tujuan dan Manfaat Penelitian..... | 3 |
| 1.4 Metodologi Penelitian | 4 |
| 1.4.1. Metode Pengumpulan Data..... | 4 |
| 1.4.2. Metode Pengembangan Sistem | 4 |
| 1.5 Sistematika Penulisan | 6 |
| | |
| BAB II TINJAUAN PUSTAKA | |
| 2.1 Pustaka Yang Terkait Dengan Penelitian..... | 8 |
| 2.2 Perbedaan Penelitian Yang Dilakukan Dengan Penelitian Terdahulu | 11 |
| | |
| BAB III LANDASAN TEORI | |
| 3.1 Cuaca..... | 12 |
| 3.2. SMS Gateway | 12 |
| 3.2.1. SMS (Short Message Service) | 12 |

| | | |
|--------|--|----|
| 3.2.2. | SMS Gateway | 13 |
| 3.2.3. | Komponen Pendukung SMS Gateway | 14 |
| 3.2.4. | Keuntungan SMS Gateway..... | 14 |
| 3.2.5. | Kekurangan SMS Gateway..... | 15 |
| 3.2.6. | Model SMS Gateway | 15 |
| 3.3. | Desain Berorintasi Objek | 16 |
| 3.3.1. | Use Case Diagram..... | 16 |
| 3.3.2. | Class Diagram..... | 18 |
| 3.3.3. | Activity Diagram..... | 20 |
| 3.4. | Database | 22 |
| 3.4.1. | Konsep Dasar Database | 22 |
| 3.4.2. | DBMS dan RDBMS | 23 |
| 3.5. | Delphi..... | 25 |
| 3.5.1. | Integrated Development Environment | 25 |
| 3.5.2. | Tipe Data Pada Delphi..... | 26 |
| 3.5.3. | Konversi Tipe Data | 26 |
| 3.6. | MySQL..... | 27 |
| 3.6.1. | Sejarah Singkat MySQL..... | 28 |
| 3.6.2. | Keistimewaan MySQL | 28 |

BAB IV ANALISA DAN PERANCANGAN SISTEM

| | | |
|--------|--------------------------------|----|
| 4.1 | Analisa Sistem | 32 |
| 4.1.1. | Analisa Permasalahan..... | 32 |
| 4.2 | Perancangan Sistem | 34 |
| 4.2.1. | Use Case Diagram..... | 34 |
| 4.2.2. | Class Diagram..... | 35 |
| 4.2.3. | Activity Diagram..... | 36 |
| 4.2.4. | Perancangan Database | 37 |
| 4.2.5. | Desain Input Output | 40 |
| 4.2.6. | Format SMS | 44 |
| 4.2.7. | Kebutuhan Perangkat Lunak..... | 44 |

| | | |
|--------|---------------------------------------|----|
| | 4.2.8. Kebutuhan Perangkat Keras..... | 45 |
| BAB V | IMPLEMENTASI SISTEM | |
| | 5.1 Form Login..... | 46 |
| | 5.2 Form Utama..... | 47 |
| | 5.3 Form Lokasi..... | 48 |
| | 5.4 Form Cuaca | 49 |
| | 5.5 Form Inbox | 50 |
| | 5.6 Form Outbox | 51 |
| | 5.7 Form SMS | 52 |
| BAB VI | PENUTUP | |
| | 6.1 Kesimpulan..... | 53 |
| | 6.2. Saran..... | 54 |

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR TABEL

| | | |
|------------|------------------------------|----|
| Tabel 3.1. | Simbol Use Case | 17 |
| Tabel 3.2. | Simbol Activity Diagram..... | 22 |
| Tabel 4.1. | Tabel Lokasi..... | 37 |
| Tabel 4.2. | Tabel Cuaca..... | 38 |
| Tabel 4.3. | Tabel Inbox | 38 |
| Tabel 4.4. | Tabel Outbox..... | 39 |

DAFTAR GAMBAR

| | | |
|--------------|-------------------------------|----|
| Gambar 3.1. | Alur SMS Gateway..... | 6 |
| Gambar 3.2. | Use Case Diagram | 9 |
| Gambar 3.3. | Class Diagram | 11 |
| Gambar 3.4. | Tampilan IDE Delphi..... | 16 |
| Gambar 4.1. | Use Case Diagram | 34 |
| Gambar 4.2. | Class Diagram | 35 |
| Gambar 4.3. | Activity Diagram Admin | 36 |
| Gambar 4.4. | Activity Diagram User..... | 37 |
| Gambar 4.5. | Form Login..... | 40 |
| Gambar 4.6. | Perancangan Form Utama | 40 |
| Gambar 4.7. | Perancangan Form Lokasi | 41 |
| Gambar 4.8. | Perancangan Form Cuaca..... | 42 |
| Gambar 4.9. | Perancangan Form Inbox | 42 |
| Gambar 4.10. | Perancangan Form Outbox..... | 43 |
| Gambar 4.11. | Perancangan SMS..... | 43 |

BAB I

PENDAHULUAN

1.1 Latar Belakang

Minat terhadap bidang pengolahan citra secara *digital* dimulai pada awal tahun 1921, yaitu pertama kalinya sebuah foto berhasil ditransmisikan secara *digital* melalui kabel laut dari kota New York ke kota London (*Bartlane Cable Picture Transmission System*).

Keuntungan utama yang dirasakan saat itu adalah pengurangan waktu pengiriman foto dari sekitar 1 minggu menjadi kurang dari 3 jam. Foto tersebut dikirim dalam bentuk kode *digital* dan kemudian diubah kembali oleh *printer telegraph*.

Sekitar tahun 1960 baru tercatat suatu perkembangan pesat seiring dengan munculnya teknologi komputer yang sanggup memenuhi suatu kecepatan proses dan kapasitas memori yang dibutuhkan oleh berbagai algoritma pengolahan citra. Sejak itu berbagai aplikasi mulai dikembangkan, yang secara umum dapat dikelompokkan kedalam dua kegiatan :

1. Memperbaiki kualitas suatu gambar (citra) sehingga dapat lebih mudah diinterpretasikan oleh mata manusia.
2. Mengolah informasi yang terdapat pada gambar (citra) untuk keperluan pengenalan objek secara otomatis oleh suatu mesin.

Seiring dengan pesatnya perkembangan teknologi komputer maka semakin banyak pula manfaat yang diperoleh dari pengolahan citra secara digital yang dapat dirasakan diberbagai bidang. Karena alasan inilah sehingga kemampuan dasar sebuah komputer dari generasi terakhir ini selalu dikaitkan dengan fasilitas dalam bidang grafika desain dan juga multimedia yang tidak hanya dapat mengolah suatu citra atau gambar tetapi juga dalam bentuk video.

Salah satu pengolahan citra dalam bentuk video adalah untuk melakukan proses pengamatan suatu objek atau benda yang tidak perlu dilakukan pengamatan secara langsung atau terus menerus pada suatu objek, namun cukup meletakkan suatu kamera yang mengarah pada objek yang diinginkan lalu mengamatinya dari layar monitor. Dengan mengamati citra yang terekam kamera dapat diketahui kondisi dari objek tersebut.

Masalah timbul karena selama ini kamera hanya dapat menangkap suatu objek tetapi tidak dapat memberikan informasi tentang gerakan dari objek tersebut. Penentuan gerakan objek tersebut biasanya dilakukan dengan mengamati citra yang terekam oleh kamera. Hal ini menjadi tidak efektif bila terlalu banyak citra yang akan diamati, dan hal ini memungkinkan terjadinya kesalahan pengamatan.

Berdasarkan latar belakang masalah diatas, penulis mengemukakan gagasan membuat perangkat lunak untuk mendeteksi gerakan objek dengan judul “ **Implementasi Metode Image Subtracting Untuk Mendeteksi Gerakan Objek Dengan Warna Pada File Video**”.

1.2 Perumusan Masalah

Permasalahan yang akan dibahas dalam penelitian ini adalah mendeteksi gerakan objek dengan warna pada file video dengan mengimplementasikan metode image subtracting untuk menggunakan bahasa pemrograman Matlab.

1.3 Batasan Masalah

Agar tidak menyimpang jauh dari permasalahan, maka penelitian ini membatasi masalah sebagai berikut :

1. Metode deteksi gerakan menggunakan metode image subtracting.
2. Bahasa pemrograman yang dipakai adalah bahasa pemrograman Matlab.
3. Objek yang dideteksi pada file video adalah objek yang berwarna merah.
4. Webcam tidak bergerak kesegala arah dan hanya mengarah pada suatu objek yang akan diamati.

5. Objek yang dijadikan percobaan adalah objek tunggal dalam file video.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah :

1. Menghasilkan aplikasi yang dapat mendeteksi objek bergerak dalam file video streaming dengan metode image subtracting.
2. Menghasilkan aplikasi yang dapat mendeteksi objek dengan warna tertentu pada file video.
3. Menghasilkan aplikasi yang dapat merecord pergerakan objek.

1.5 Manfaat Penelitian

Setiap kegiatan yang dilakukan tentunya diyakini dan diharapkan memiliki manfaat, walaupun manfaat yang diperoleh tidaklah terlalu besar akan tetapi paling tidak akan membawa perubahan-perubahan menuju ke arah yang lebih baik.

Adapun Manfaat yang dapat diambil dari penulisan skripsi ini adalah sebagai salah satu ide pembuatan aplikasi yang memanfaatkan *image processing* dalam mendeteksi objek bergerak dan merecord pergerakan objek tersebut.

1.6 Metodologi Penelitian

Metode yang digunakan pada penelitian ini adalah :

1.6.1 Studi Pustaka Yang Dilakukan

yaitu pengumpulan data dengan mempelajari literatur-literatur atau buku – buku yang berkaitan dengan masalah diatas sehingga mendapatkan pengetahuan intuk merepresentasikan ide gagasan diatas.

1.6.2 Pengembangan Aplikasi

Metode yang digunakan untuk membuat atau mengembangkan aplikasi perangkat lunak pada penelitian ini adalah metode *prototype*. Metode ini merupakan metode pengembangan sistem dimana hasil analisa per bagian langsung diterapkan kedalam sebuah model tanpa harus menunggu seluruh sistem selesai dianalisa (Pressman, 2002).



Gambar 1.1 Metodologi Ptototipe

(<http://jejakjari007.blogspot.com/2011/04/metodologi-pengembangan-sistem.html>)

Seperti yang terlihat pada gambar 1.1 tahapan-tahapan dalam metode *prototype* antara lain :

1. Pengumpulan Kebutuhan dan perbaikan

Menetapkan segala kebutuhan untuk pembangunan perangkat lunak

2. Desain cepat

Tahap penerjemahan dari keperluan atau data yang telah dianalisis ke dalam bentuk yang mudah dimengerti oleh user.

3. Bentuk *Prototype*

Menerjemahkan data yang telah dirancang ke dalam bahasa pemrograman (Program contoh atau setengah jadi)

4. Evaluasi Pelanggan Terhadap *Prototype*

Program yang sudah jadi diuji oleh pelanggan, dan bila ada kekurangan pada program bisa ditambahkan.

5. Perbaikan *Prototype*

Perbaikan program yang sudah jadi, sesuai dengan kebutuhan konsumen. Kemudian dibuat program kembali dan di evaluasi oleh konsumen sampai semua kebutuhan user terpenuhi.

6. Produk Rekayasa

Program yang sudah jadi dan seluruh kebutuhan user sudah terpenuhi

1.6.3 Penyusunan Laporan

Penyusunan laporan dilakukan untuk mendokumentasikan penelitian ini.

1.7 Sistematika Penulisan

Sistematika penulisan yang digunakan dalam penyusunan laporan skripsi adalah sebagai berikut :

BAB I PENDAHULUAN

Bab ini merupakan pembahasan tentang latar belakang permasalahan, perumusan masalah, batasan masalah, tujuan dan manfaat penelitian, metodologi penelitian serta sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Bab ini menjelaskan uraian sistematis tentang informasi hasil penelitian yang telah dilakukan sebelumnya dan menghubungkannya dengan masalah penelitian yang sedang diteliti.

BAB III LANDASAN TEORI

Bab ini menjelaskan mengenai teori-teori dasar dan teori pendukung untuk menyusun skripsi, mulai dari definisi,

gambaran umum dan penjelasan tentang perangkat lunak pendukung.

BAB IV ANALISIS DAN PERANCANGAN SISTEM

Bab ini menjelaskan tentang perancangan sistem yang akan digunakan dalam mendesain program.

BAB V IMPLEMENTASI DAN PENGUJIAN SISTEM

Bab ini penulis menyajikan tentang hasil-hasil dari tahapan penelitian teknik implementasi serta pengujian sistem yang sudah selesai. Dalam bab ini juga menguraikan tentang pemilihan perangkat keras (*hardware*) dan piranti lunak (*software*) yang dibutuhkan.

BAB VI PENUTUP

Berisi kesimpulan dan saran dari uraian - uraian bab yang telah dibahas sebelumnya serta saran – saran.

BAB II TINJAUAN

PUSTAKA

2.1 Merujuk dari penelitian **Benedictus Yoga Budi Putranto, Widi Hapsari, Katon Wijana (Universitas Kristen Duta Wacana Yogyakarta)** yang berjudul **“Segmentasi Warna Citra Dengan Deteksi Warna HSV Untuk Mendeteksi Objek“** menghasilkan penelitian sebagai berikut :

Penelitian ini memaparkan penerapan metode segmentasi warna dengan detel di warna HSV oleh Giannakopolous untuk menghasilkan objek segmen citra berupa *blob* sehingga dapat terdeteksi komputer. Berdasarkan hasil pengujian dan analisa pada penelitian ini diperoleh kesimpulan kontrol pengguna dalam hal penentuan sampel warna dan toleransi warna berperan penting dalam proses segmentasi, sampel warna alam menghasilkan nilai acuan warna sebagai acuan segmentasi dan toleransi warna digunakan sebagai jangkauan filter dalam proses segmentasi. Proses deteksi objek akan mengolah segmen warna yang dihasilkan oleh proses segmentasi sehingga dapat diketahui banyaknya objek terdeteksi, luas area dan titik pusat tiap objek.

Berdasarkan penelitian ini disimpulkan bahwa segmentasi warna berdasarkan deteksi warna HSV merupakan proses segmentasi terkontrol dengan filter warna HSV. Filter warna HSV akan memisahkan warna tertentu sesuai dengan warna acuan dan nilai toleransi tiap elemen warna HSV. Kontrol pengguna melalui sampel warna dan toleransi warna yang menjadi acuan filter sehingga

dapat diperoleh segmen warna yang sesuai. Hasil segmentasi warna berdasarkan deteksi warna HSV sangat dipengaruhi sampel warna dan toleransi warna yang menjadi acuan proses segmentasi. Pencahayaan, letak, tekstur dan kontur benda atau latar belakang citra akan sangat mempengaruhi hasil segmentasi dan deteksi objek.

2.2 Merujuk dari penelitian **Putri Mahanani R, Fernando Ardilla S, ST, Setiawardhana S.kom (Institut Teknologi Sepuluh Nopember)** yang berjudul **“Tracking Obyek Menggunakan Metode Counterpropagation“** menghasilkan penelitian sebagai berikut :

Dalam penelitian ini dibuat sebuah sistem yang mampu melakukan penjejakan terhadap sebuah obyek dengan memanfaatkan teknik *Image Processing*. Sistem menggunakan kamera *wireless* yang digunakan untuk mendapatkan gambar obyek. Untuk penjejakan, metode yang digunakan adalah metode Jaringan Syaraf Tiruan *Counter Propagation Network (CPN)*, karena metode ini dapat melakukan proses iterasi, pelatihan dan respon lebih cepat. Sehingga diharapkan sistem yang dibuat menjadi sebuah aplikasi yang mampu melakukan pengontrolan terhadap suatu obyek dengan baik.

Berdasarkan Penelitian ini disimpulkan bahwa deteksi objek untuk pendeteksian obyek, berdasarkan hasil uji coba yang telah dilakukan, didapatkan bahwa objek yang terdeteksi adalah obyek yang berwarna biru. Untuk dapat melakukan pendeteksian dengan baik, disekitar obyek yang dideteksi harus tidak

mengandung warna biru yang sesuai dengan obyek yang diamati. Untuk memudahkan sistem mengamati obyek. Terdapat *delay* pada saat obyek ditampilkan di dalam *picture box*, sehingga gambar yang ditampilkan terlihat lambat.

2.3 Merujuk dari penelitian **Ibnu Catur Mustofa (Universitas Islam Negeri Malang)** yang berjudul **“Monitoring Gerakan pada Ruang Menggunakan Webcam dan Motor Stepper “** menghasilkan penelitian sebagai berikut :

Dalam penelitian ini menggunakan teknik pengolahan citra atau sering di sebut *image processing*. Metode yang digunakan adalah deteksi gerakan. Metode ini mampu mendeteksi perbedaan nilai RGB (*red, green, blue*) di setiap titik *pixel* pada suatu citra atau gambar digital. Keadaan inilah yang dimanfaatkan untuk di aplikasikan menjadi suatu sistem keamanan. Untuk mendapatkan nilai RGB yang lebih besar maka obyek yang ditangkap oleh kamera harus bercahaya. Dengan nilai RGB yang semakin besar maka pendeteksian gerakan akan lebih mudah. Pendeteksian gerakan dinyatakan aktif jika alarm telah berbunyi. Bunyi alarm disertai dengan putaran motor stepper yang menggerakkan kamera untuk mengikuti letak obyek yang bergerak. Putaran motor stepper dikontrol oleh PC (*personal computer*) melalui port paralel. Proses pengikutan letak obyek akan berhenti jika obyek telah tidak terdeteksi oleh kamera atau aplikasi dimatikan.

Berdasarkan penelitian ini dapat disimpulkan bahwa *webcam* dapat dimanfaatkan sebagai sistem keamanan dengan menggunakan metode deteksi

gerakan. Metode ini digunakan untuk mendeteksi adanya perbedaan gerakan pada obyek. Dengan menambahkan motor stepper dapat di perintah melalui PC diperlukan rangkaian driver yang berfungsi untuk menggerakkan motor stepper dengan perintah software tertentu. Pada rangkaian driver terdapat sebuah IC, dimana input dari IC tersebut dihubungkan dengan port paralel yang terdapat pada PC dan output dari IC dihubungkan dengan motor stepper. Webcam dapat mendeteksi letak obyek dengan cara menganalisis nilai RGB pada koordinat yang ditentukan terhadap citra yang tersampling, dengan parameter nilai RGB lebih besar dari 230.

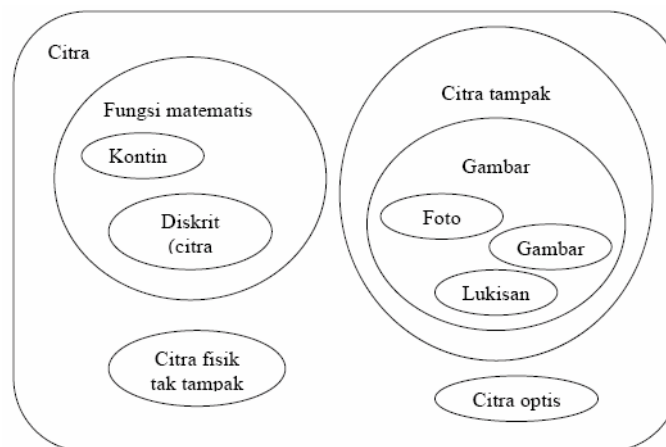
Hasil dari pengujian program deteksi gerak ini adalah program dapat berjalan sebagaimana mestinya akan tetapi program ini kurang maksimal apabila obyek yang ditangkap jarak antara kamera dengan obyek tersebut terlalu jauh, yakni motor stepper tidak dapat bergerak sebagaimana mestinya, hal ini dikarenakan intensitas cahaya yang ditangkap oleh kamera kurang maksimal, karena prosedur untuk menggerakkan motor stepper menggunakan nilai RGB.

BAB III LANDASAN

TEORI

3.1 Definisi Citra (Gambar diam)

Citra (*image*) sebagai salah satu komponen multimedia memegang peranan sangat penting sebagai bentuk informasi visual. Citra mempunyai karakteristik yang tidak dimiliki oleh data teks, yaitu citra kaya dengan informasi. Ada sebuah peribahasa yang berbunyi “sebuah gambar bermakna lebih dari seribu kata” (*a picture is more than a thousand words*). Maksudnya tentu sebuah gambar dapat memberikan informasi yang lebih banyak daripada informasi tersebut disajikan dalam bentuk kata-kata (tekstual). Citra dapat dikelompokkan menjadi citra tampak dan citra tak tampak, sebagaimana disajikan pada gambar 3.1 di bawah ini :



Gambar 3.1 Pengelompokan jenis-jenis citra (Castleman, 1996)

Contoh citra tampak dalam kehidupan sehari-hari adalah foto keluarga, gambar yang nampak pada layar monitor dan televisi, serta hologram (citra optis). Sedangkan contoh citra tak tampak adalah data gambar dalam file (citra digital) dan citra yang merepresentasikan menjadi fungsi matematis. Di samping itu ada juga citra fisik tak tampak, misalnya citra distribusi panas di kulit manusia serta peta densitas dalam suatu material. Untuk dapat dilihat mata manusia, citra tak tampak ini harus diubah menjadi citra tampak, misalnya dengan menampilkannya di monitor, dicetak di atas kertas, dan sebagainya.

3.1.1 Definisi Citra Analog

Citra analog adalah citra yang bersifat kontinu, 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 computer secara langsung. Oleh sebab itu, agar citra ini dapat diproses di computer, proses konversi analog ke digital harus dilakukan terlebih dahulu. Citra analog dihasilkan dari alat – alat analog, seperti video kamera analog, kamera foto analog, *WebCam*, *CT scan*, sensor *roentgen* untuk foto *thorax*, sensor gelombang pendek pada sistem radar, sensor *ultrasound* pada sistem USG, dan lain – lain.

3.1.2 Definisi Citra Digital

Menurut kamus Webster (dalam Hestiningsih, 2009) citra adalah suatu representasi, kemiripan, atau imitasi dari suatu objek atau benda. Citra dapat dikatakan sebagai citra digital jika citra tersebut disimpan dalam format digital (dalam bentuk *file*). Hanya citra digital yang dapat diolah menggunakan komputer. Jenis citra lain jika akan diolah dengan komputer harus diubah dulu menjadi citra digital.

Sebuah citra digital dapat mewakili oleh sebuah matriks yang terdiri dari M kolom N baris, dimana perpotongan antara kolom dan baris disebut piksel (*pixel* = picture element), yaitu elemen terkecil dari sebuah citra. Piksel mempunyai dua parameter, yaitu koordinat dan intensitas atau warna. Nilai yang terdapat pada koordinat (x,y) adalah f(x,y), yaitu besar intensitas atau warna dari piksel di titik itu. Oleh sebab itu, sebuah citra digital dapat ditulis dalam bentuk matriks pada gambar 3.2 .

$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0, M - 1) \\ f(1,0) & \dots & \dots & f(1, M - 1) \\ \dots & \dots & \dots & \dots \\ f(N - 1, 0) & f(N - 1, 1) & \dots & f(N - 1, M - 1) \end{bmatrix}$$

Gambar 3.2 Matriks citra digital N x M

Berdasarkan gambar 3.2 matriks citra digital, Suatu citra $f(x,y)$ dalam fungsi matematis dapat dituliskan persamaan (2.1) sebagai berikut :

$$\begin{array}{l}
 0 \quad x \quad M-1 \\
 0 \quad y \quad N-1 \\
 0 \quad f(x,y) \quad G-1
 \end{array}
 \left. \vphantom{\begin{array}{l} 0 \quad x \quad M-1 \\ 0 \quad y \quad N-1 \\ 0 \quad f(x,y) \quad G-1 \end{array}} \right\} \dots\dots\dots (2.1)$$

dimana : M = jumlah piksel baris (*row*) pada array citra

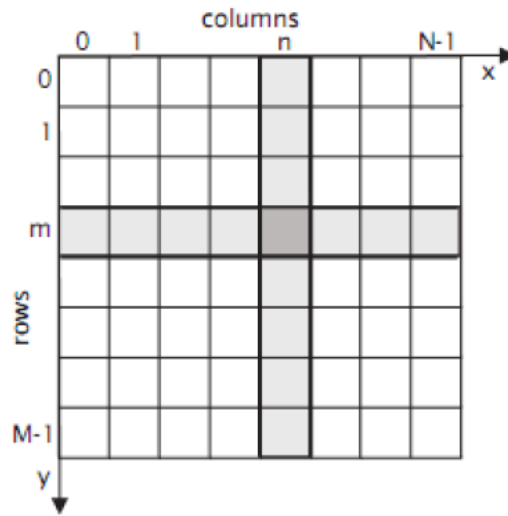
N = jumlah piksel kolom (*column*) pada array citra

G = nilai skala keabuan (*graylevel*)

Besarnya nilai M , N dan G pada umumnya merupakan perpangkatan dari dua seperti yang terlihat pada persamaan (2.2).

$$M = 2^m ; N = 2^n ; G = 2^k \dots\dots\dots(2.2)$$

Dimana nilai m , n dan k adalah bilangan bulat positif. Interval (0,G) disebut skala keabuan (grayscale). Besar G tergantung pada proses digitalisasinya. Biasanya keabuan 0 (nol) menyatakan intensitas hitam dan 1 (satu) menyatakan intensitas putih. Untuk citra 8 bit, nilai G sama dengan $2^8 = 256$ warna (derajat keabuan). (RD. Kusumanto : 2011)



Gambar 3.3 Representasi citra digital dalam 2 dimensi

3.2 Video (Citra Bergerak)

3.2.1 Animasi

Animasi adalah proses penciptaan efek gerak atau efek perubahan bentuk yang terjadi selama beberapa waktu.

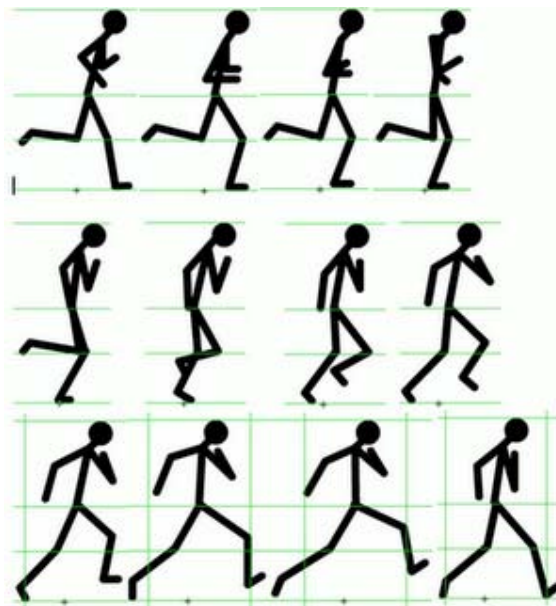
Animasi merupakan suatu teknik menampilkan gambar berurut sedemikian rupa sehingga penonton merasakan adanya ilusi gerakan (motion) pada gambar yang ditampilkan. Secara umum ilusi gerakan merupakan perubahan yang dideteksi secara visual oleh mata penonton. Perubahan seperti perubahan warna pun dapat dikatakan sebuah animasi.

Berdasarkan kedua definisi diatas dapat diketahui bahwa animasi tidak semata - mata hanyalah menggerakkan saja, tetapi juga memberikan suatu karakter pada obyek - obyek yang akan dianimasikan, sehingga obyek animasinya tidak bersifat perubahan gerak, tetapi lebih

daripada itu yaitu adanya mood, emosi, watak tak jarang dimasukkan sebagai suatu pengembangan karakterisasi. Sebuah animasi disusun oleh himpunan gambar yang ditampilkan secara berurut, maka animasi dapat dikatakan sebuah fungsi terhadap waktu.

Animasi merupakan kumpulan gambar yang ditampilkan secara bergantian sehingga akan terlihat bergerak. Pergerakan dari animasi akan lebih mudah dicerna oleh pemakai daripada gambar diam, akan tetapi gambar diam memang lebih komunikatif dibanding animasi dalam hal-hal tertentu, sedangkan animasi dibuat khusus untuk mendukung konsep ilustrasi yang mengharuskan adegan gambar yang bergerak.

(skp.unair.ac.id/.../web_ANIMASI_PUTRI_NI_MATUL_LILLAH.pdf)



Gambar 3.4 Gambar Animasi

3.2.2 Video

Video adalah gabungan dari banyak citra digital yang diperlihatkan sesuai urutan dengan jangka waktu tertentu sehingga citra tersebut tampak bergerak. Untuk dapat mengolah video, maka harus mendapatkan data warna pada frame – frame yang ada pada video, karena frame – frame tersebut berupa sebuah citra digital, maka dalam pengolahan video tidak dapat terlepas dari pengolahan citra digital.

3.3 Jenis – Jenis Citra Digital

Berdasarkan warna-warna penyusunnya, citra digital dapat dibagi menjadi tiga macam (Marvin Chandra Wijaya,2007) yaitu:

3.3.1 Citra Biner

Citra biner, yaitu citra yang hanya terdiri atas dua warna, yaitu hitam dan putih. Oleh karena itu, setiap *pixel* pada citra biner cukup direpresentasikan dengan 1 bit yang dapat dilihat dalam gambar 3.5 citra biner yang memiliki piksel dengan 2 kemungkinan 0 dan 1.



Gambar. 3.5 Citra biner yang memiliki piksel dengan 2 kemungkinan 0 dan 1

Meskipun saat ini citra berwarna lebih disukai karena member kesan yang lebih kaya dari citra biner, namun tidak membuat citra biner mati. Pada beberapa aplikasi citra biner masih tetap di butuhkan, misalkan citra logo instansi (yang hanya terdiri dari warna hitam dan putih), citra kode barang (*bar code*) yang tertera pada label barang, citra hasil pemindaian dokumen teks, dan sebagainya. Seperti yang sudah disebutkan diatas, citra biner hanya mempunyai dua nilai derajat keabuan hitam dan putih. Piksel – piksel objek bernilai 1 dan piksel – piksel latar belakang bernilai 0. Pada waktu menampilkan gambar, 0 adalah putih dan 1 adalah hitam.

Jadi pada citra biner, latar belakang berwarna hitam sedangkan objek berwarna putih seperti tampak pada gambar 3.5 diatas. Meskipun komputer saat ini dapat memproses citra hitam-putih (*grayscale*) maupun

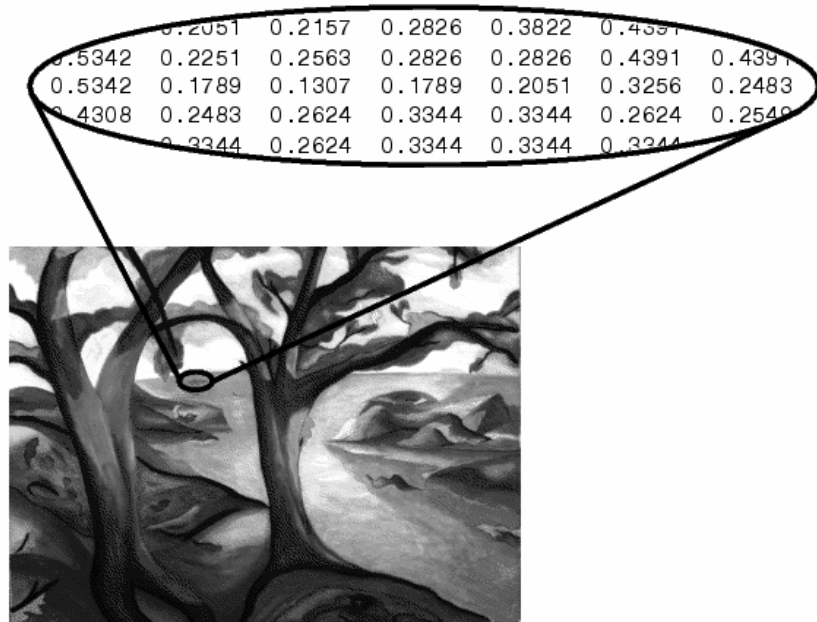
citra berwarna, namun citra biner masih tetap di pertahankan keberadaannya.

Alasan penggunaan citra biner adalah karena citra biner memiliki sejumlah keuntungan sebagai berikut:

- a. Kebutuhan memori kecil karena nilai derajat keabuan hanya membutuhkan representasi 1 bit.
- b. Waktu pemrosesan lebih cepat di bandingkan dengan citra hitam putih ataupun warna.

3.3.2 Citra Grayscale

Citra *grayscale*, yaitu citra yang nilai *pixel*-nya merepresentasikan derajat keabuan atau intensitas warna putih. Nilai intensitas paling rendah merepresentasikan warna hitam dan nilai intensitas paling tinggi merepresentasikan warna putih. Pada umumnya citra *grayscale* memiliki kedalaman *pixel* 8 bit (256 derajat keabuan), tetapi ada juga citra *grayscale* yang kedalaman *pixel*-nya bukan 8 bit, misalnya 16 bit untuk penggunaan yang memerlukan ketelitian tinggi.



Gambar 3.6 Citra grayscale (abu-abu)

Citra grayscale merupakan citra satu kanal, dimana citra $f(x,y)$ merupakan fungsi tingkat keabuan dari hitam ke putih, x menyatakan variable kolom atau posisi pixel di garis jelajah dan y menyatakan variable baris atau posisi pixel di garis jelajah. Intensitas f dari gambar hitam putih pada titik (x,y) disebut derajat keabuan (grey level), yang dalam hal ini derajat keabuannya bergerak dari hitam keputih. Derajat keabuan memiliki rentang nilai dari I_{min} sampai I_{max} , atau $I_{min} < f < I_{max}$, selang (I_{min}, I_{max}) disebut skala keabuan.

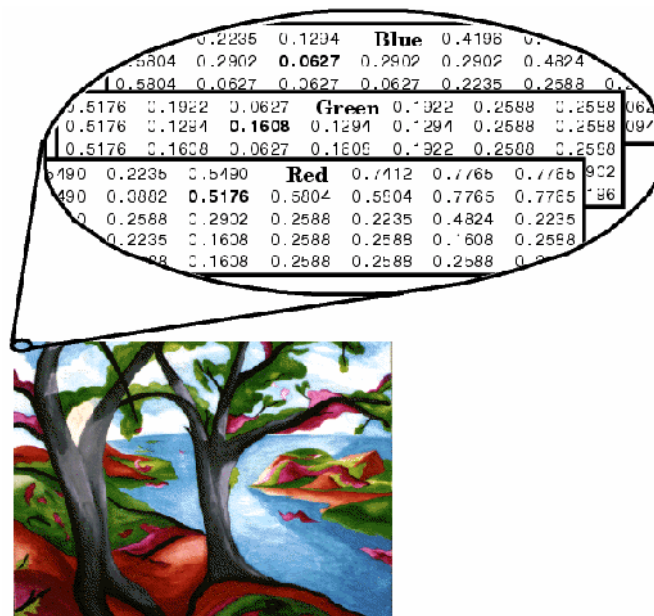
Biasanya selang (I_{min}, I_{max}) sering digeser untuk alasan-alasan praktis menjadi selang $[0,L]$, yang dalam hal ini nilai intensitas 0 menyatakan hitam, nilai intensitas L menyatakan putih, sedangkan nilai intensitas antara 0 sampai L bergeser dari hitam ke putih. Sebagai contoh

citra grayscale dengan 256 level artinya mempunyai skala abu dari 0 sampai 255 atau [0,255], yang dalam hal ini intensitas 0 menyatakan hitam, intensitas 255 menyatakan putih, dan nilai antara 0 sampai 255 menyatakan warna keabuan yang terletak antara hitam dan putih.

3.3.3 Citra RGB (Citra Warna atau truecolor)

Citra berwarna, yaitu citra yang nilai *pixel*-nya merepresentasikan warna tertentu Banyaknya warna yang mungkin digunakan bergantung kepada kedalaman *pixel* citra yang bersangkutan.

Citra berwarna direpresentasikan dalam beberapa kanal (*channel*) yang menyatakan komponen-komponen warna penyusunnya. Banyaknya kanal yang digunakan bergantung pada model warna yang digunakan pada citra tersebut.

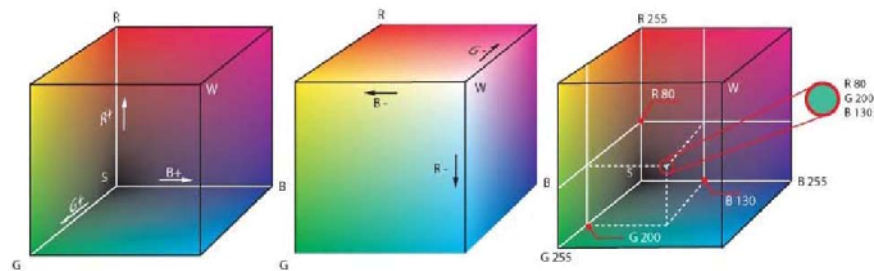


Gambar 3.7 Citra RGB (True Color)

Intensitas suatu pada titik pada citra berwarna merupakan kombinasi dari tiga intensitas :

1. derajat keabuan merah ($f_{\text{merah}}(x,y)$)
2. hijau ($f_{\text{hijau}}(x,y)$)
3. biru ($f_{\text{biru}}(x,y)$)

Persepsi visual citra berwarna umumnya lebih kaya di bandingkan dengan citra hitam putih. Citra berwarna menampilkan objek seperti warna aslinya (meskipun tidak selalu tepat demikian). Warna-warna yang diterima oleh mata manusia merupakan hasil kombinasi cahaya dengan panjang gelombang berbeda. Pada citra RGB setiap piksel diwakilkan oleh gabungan dari ketiga warna ini, yang mana masing-masing bernilai antara 0 sampai dengan 255. Sebagai contoh, warna ungu dinyatakan dengan nilai 255 untuk merah(R), 0 untuk hijau(G), dan 255 untuk biru(B).



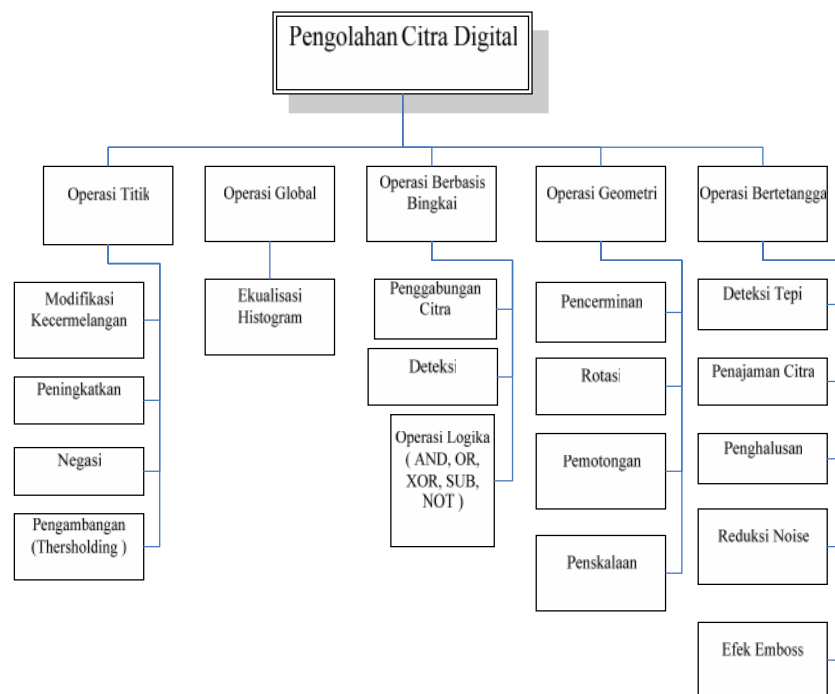
Gambar 3.8 RGB Color Space

3.4 *Image Processing (Pengolahan Citra)*

Kegiatan untuk mengubah informasi citra fisik non digital menjadi digital disebut sebagai pencitraan (*imaging*). Citra digital dapat

diolah dengan komputer karena berbentuk data numeris. Suatu citra digital melalui pengolahan citra digital (*digital image processing*) menghasilkan citra digital yang baru termasuk di dalamnya adalah perbaikan citra (*image restoration*) dan peningkatan kualitas citra (*image enhancement*).

Seperti yang terlihat dalam gambar 3.8 operasi – operasi dasar pada pengolahan citra digital.



Gambar 3.9 Operasi-operasi dasar pada Pengolahan Citra Digital

Operasi – operasi dasar pengolahan citra digital antara lain:

1. Operasi Titik

Operasi titik adalah operasi yang hanya dilakukan pada *pixel* tunggal di dalam citra. Operasi titik dikenal juga dengan nama operasi *pointwise*.

http://informatika.stei.ac.id/~rinaldi.munir/Buku/Pengolahan%20citra%20Digital/Bab-4_Operasi-operasi%20Dasar%20Pengolahan%20Citra%20Dijital.pdf

Beberapa operasi pengolahan citra, terkait operasi titik :

- a. Modifikasi kecermerlangan (*brightness modification*)
- b. Peningkatan Kontras (*contrast enhancement*)
- c. Negasi (*negation*)
- d. Pengambangan (*thresholding*)

2. Operasi global

Operasi global adalah operasi untuk menghasilkan citra keluaran yang intensitas suatu *pixel* bergantung pada intensitas keseluruhan *pixel*. Contoh operasi global adalah operasi penyetaraan histogram untuk meningkatkan kualitas citra.

3. Operasi berbasis bingkai (*frame*)

Operasi berbasis bingkai (*frame*) adalah operasi yang melibatkan 2 buah citra atau lebih dan menghasilkan sebuah citra keluaran yang merupakan hasil operasi matematis. Operasi ini dilakukan titik per titik dengan lokasi yang bersesuaian pada citra-citra masukan tersebut. Untuk operasi berbasis bingkai antara 2 buah citra, misalnya antara citra A dan citra B yang menghasilkan citra C, secara umum dapat dituliskan sebagai :

$$C(x,y) = A(x,y) \text{ op } B(x,y) \dots\dots\dots(2.3)$$

Dimana op adalah operator yang akan diterapkan terhadap kedua citra tersebut. Untuk operasi yang melibatkan N buah citra A_1 sampai dengan A_N , maka rumus diatas menjadi :

$$C(x,y) = A_1(x,y) \text{ op } A_2(x,y) \text{ op } A_3(x,y) \dots \text{ op } A_N(x,y) \dots \dots \dots (2.4)$$

(http://sorisoga.blogspot.com/2010/10/pengolahan-citra_20.html).

Operasi berbasis bingkai dalam pengolahan citra antara lain :

- a. Penggabungan citra
- b. Deteksi
- c. Operasi logika (AND, OR, XOR, SUB, NOT)

4. Operasi geometri

Operasi geometri pada pengolahan citra ditujukan untuk memodifikasi koordinat piksel dalam suatu citra dengan pendekatan tertentu, tetapi dalam perkembangannya dimungkinkan juga memodifikasi nilai skala keabuan (<http://edywinarnosemarang.wordpress.com/materi-kuliah>).

Operasi Geometri berhubungan dengan perubahan bentuk geometri citra, antara lain :

- a. Pencerminan (*flipping*)
- b. Rotasi/pemutaran (*Rotating*)
- c. Pemotongan (*Cropping*)
- d. Penskalaan (*Scaling/Zooming*)

5. Operasi bertetangga

Operasi bertetangga adalah operasi yang memodifikasi nilai keabuan sebuah titik berdasarkan nilai-nilai keabuan dari titik-titik yang ada di

sekitarnya (bertetangga) yang masing-masing mempunyai bobot tersendiri.

Beberapa operasi pengolahan citra yang berkaitan dengan operasi bertetangga antara lain :

- a. Deteksi Tepi (*Edge Detection*)
- b. Penghalusan Citra (*Smoothing*)
- c. Penajaman Citra (*Sharping*)
- d. Reduksi *Noise*
- e. Efek *Emboss*

3.5 Threshold

Thresholding merupakan salah satu teknik segmentasi yang baik digunakan untuk citra dengan perbedaan nilai intensitas yang signifikan antara latar belakang dan objek utama (Katz,2000). Dalam pelaksanaannya *Thresholding* membutuhkan suatu nilai yang digunakan sebagai nilai pembatas antara objek utama dengan latar belakang, dan nilai tersebut dinamakan dengan *threshold*.

Thresholding digunakan untuk mempartisi citra dengan mengatur nilai intensitas semua piksel yang lebih besar dari nilai *threshold T* sebagai latar depan dan yang lebih kecil dari nilai *threshold T* sebagai latar belakang. Biasanya pengaturan nilai *threshold* dilakukan berdasarkan histogram *grayscale* (Gonzales dan Woods, 2002; Fisher, dkk, 2003; Xiaoyi dan Mojon, 2003).

Thresholding adalah metode yang paling sederhana dari segmentasi. Setiap individu piksel didalam *grayscale* ditandai sebagai “objek” piksel jika nilai mereka lebih besar dari nilai *Thresholding* (dijadikan sebuah objek yang lebih terang dari *backgroundnya*) dan sebagai “*background*” piksel sebaliknya. Biasanya, sebuah objek piksel diberikan nilai “1” dan piksel *background* diberikan nilai “0”.

Parameter kunci di dalam *Thresholding* merupakan pilihan dalam melakukan *Threshold* . Terdapat berbagai metode dalam memilih *Threshold*. Metode paling sederhana dilakukan dengan cara memilih nilai *mean* atau *median* Pada dasarnya jika piksel objek lebih terang dibandingkan dengan *background* maka piksel objek tersebut juga lebih terang dari rata-ratanya. Pada gambar yang masih memiliki *noise* dengan *background* dan nilai objek, *mean* dan *median* akan bekerja maksimal dalam *Threshold*. Dalam pendekatan yang lebih dalam, dapat pula dilakukan dengan cara membuat sebuah histogram dari intensitas citra piksel dan menggunakan *valley point* sebagai nilai *threshold*. Dengan melakukan pendekatan histogram memungkinkan adanya beberapa nilai rata – rata pada piksel *background* dan objek, tetapi nilai piksel tersebut mempunyai beberapa variasi nilai yang masih berada pada sekitar nilai rata – rata itu. Akan tetapi biasanya tidak selalu sesederhana itu dan banyak histogram dari citra yang mempunyai *valley point* yang tidak jelas.

Pencarian metode threshold yang sederhana tidak memerlukan pengetahuan yang lebih tentang citra dan *thresholding* pun bisa bekerja pada citra yang memiliki *noise* , Metode iterative merupakan pendekatan yang baik untuk dilakukan seperti :

1. Memilih *initial* dari *threshold* (T). Dapat dilakukan secara random atau menurut metoda yang diinginkan
2. Citra ini disegmentasikan ke dalam piksel objek dan piksel *background* seperti di bawah ini.

- $G_1 = \{f(m,n) : f(m,n) > T\}$(2.5)

- $G_2 = \{f(m,n) : f(m,n) \leq T\}$(2.6)

di mana :

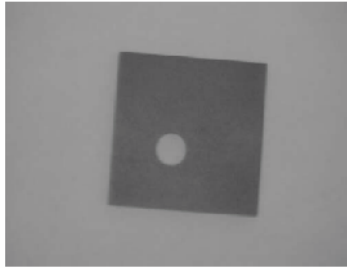
- G_1 adalah nilai piksel objek.
- G_2 adalah nilai piksel *background*.
- $f(m,n)$ adalah nilai dari piksel yang terletak pada m^{th} kolom dan n^{th} baris

3. Hitung nilai rata – rata *gray value* μ_1 dan μ_2 pada piksel dalam G_1 dan G_2

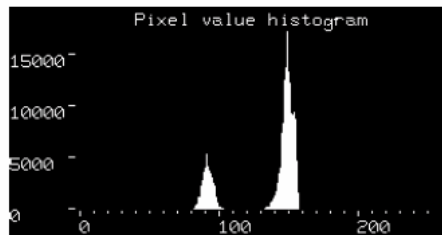
4. Hitung nilai *threshold* baru :

$$T = \frac{1}{2} (\mu_1 + \mu_2) \dots\dots\dots(2.7)$$

5. Ulangi langkah ke 2 sampai dengan langkah ke 4 dengan nilai T yang berbeda sampai nilai *threshold* yang baru sama dengan nilai yang sebelumnya.



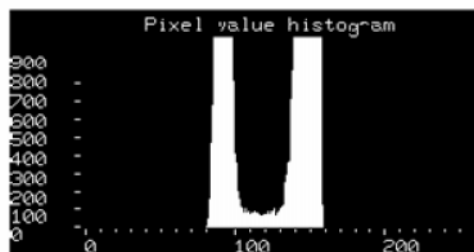
Gambar 3.10 Objek dengan warna gelap dan diletakkan pada *background* terang



Gambar 3.11 Histogram dari Gambar 3.10



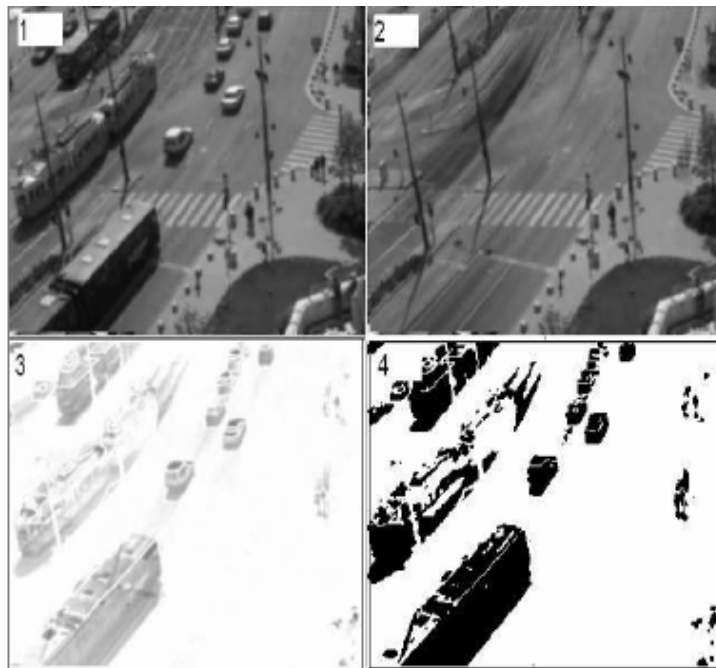
Gambar 3.12 Objek yang telah *dithreshold* dengan nilai *threshold* sekitar 120



Gambar 3.13 Histogram dari gambar 3.12

3.6 Background subtraction

Background subtraction adalah proses untuk menemukan objek pada gambar dengan cara membandingkan gambar yang ada dengan sebuah model latar belakang. Prosedur *background subtraction* terdiri dari 3 tahap, yaitu *pre-processing*, *background modeling*, dan *foreground detection*.



Gambar 3.14 *Background subtraction*

1- gambar sekarang, 2- *background model*, 3- hasil *background subtraction*,
4- hasil *background subtraction* setelah *threshold*

Tahapan dalam *Background subtraction* :

a. **Pre-processing**

Pada tahap ini data mentah dari kamera (atau input lainnya) diproses menjadi bentuk yang dapat dimengerti oleh bagian program lain. Pada tahapan awal ini dilakukan *noise removal* dan eliminasi objek kecil

pada gambar agar menjadi lebih informatif. Eliminasi objek kecil dilakukan dengan menggunakan *mathematical morphology* yaitu transformasi Opening.

b. Background modeling

Tahap ini bertujuan untuk membentuk model *background* yang konsisten, namun tetap dapat beradaptasi dengan perubahan lingkungan yang ada. Model harus dapat mentoleransi tingkat perubahan lingkungan, namun tetap sensitif dalam mendeteksi pergerakan dari objek yang relevan. Algoritma *background modeling* sendiri sangat banyak, namun pada skripsi ini akan dipakai *Approximated Median Filter*, karena proses komputasinya cepat dan hasilnya cukup memuaskan.

c. Foreground detection

Pada tahap ini, dilakukan proses ekstraksi *foreground* dari *background*. Secara sederhana hal ini dilakukan dengan persamaan (2.8).

$$R_{r,c} = I_{r,c} - B_{r,c} \dots\dots\dots(2.8)$$

- R = hasil
- I = gambar saat ini
- B = *background model*
- r = baris, c = kolom

Nilai R lalu dibandingkan dengan nilai *threshold* yang telah ditentukan, jika lebih besar dari nilai *threshold* maka piksel di I(r,c) dapat

dianggap berbeda dengan piksel di $B(r,c)$. Nilai *threshold* adalah semacam nilai untuk menolerasi error yang mungkin terjadi, *threshold* sendiri dipakai untuk mengurangi error yang disebabkan *noise* pada gambar digital.

3.7 *Median filter*

Median filter adalah salah satu *filtering* non-linear yang mengurutkan nilai intensitas sekelompok *pixel*, kemudian mengganti nilai *pixel* yang diproses dengan nilai mediannya. *Median filter* telah digunakan secara luas untuk memperhalus dan mengembalikan bagian dari citra yang mengandung *noise* yang berbentuk bintik putih.

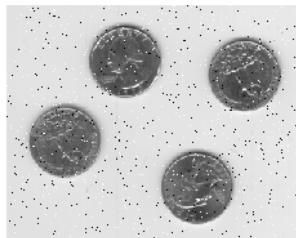
Metode median filter merupakan filter non - linear yang dikembangkan Tukey, yang berfungsi untuk menghaluskan dan mengurangi *noise* atau gangguan pada citra. Dikatakan nonlinear karena cara kerja penapis ini tidak termasuk kedalam kategori operasi konvolusi. Operasi nonlinear dihitung dengan mengurutkan nilai intensitas sekelompok *pixel*, kemudian menggantikan nilai *pixel* yang diproses dengan nilai tertentu.

Pada *median filter* suatu *window* atau penapis yang memuat sejumlah *pixel* ganjil digeser titik per titik pada seluruh daerah citra. Nilai-nilai yang berada pada *window* diurutkan secara *ascending* untuk kemudian dihitung nilai mediannya. Nilai tersebut akan menggantikan nilai yang berada pada pusat bidang *window*.

Jika suatu *window* ditempatkan pada suatu bidang citra, maka nilai *pixel* pada pusat bidang *window* dapat dihitung dengan mencari nilai median dari nilai intensitas sekelompok *pixel* yang telah diurutkan. Secara matematis dapat dari persamaan (2.9) sebagai berikut:

$$g(x,y) = \text{Median} \{ f(x-I, y-j), (i,j) \in W \} \dots \dots \dots (2.9)$$

Dimana $g(x,y)$ merupakan citra yang dihasilkan dari citra $f(x,y)$ dengan w sebagai *window* yang ditempatkan pada bidang citra dan (i,j) elemen dari *window* tersebut.



Gambar 3.15 Gambar awal sebelum difilter

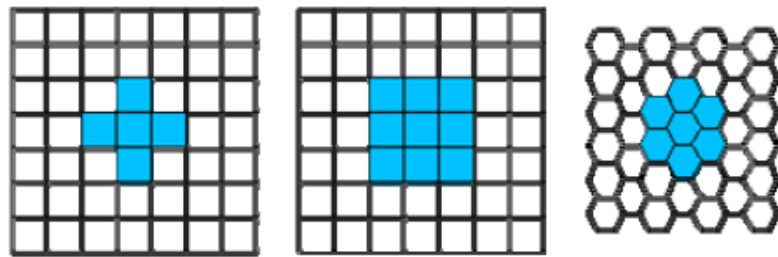


Gambar 3.16 Gambar hasil median filter

3.8 Connected component labeling

Algoritma *Connected Component Labeling* digunakan untuk melabeli tiap objek pada gambar binary dengan suatu label unik. Pengelompokan piksel sebagai satu objek ditentukan dari status

ketetanggaan mereka. Sebuah piksel disebut bertetangga dengan piksel lain apabila piksel tersebut bertetangga langsung dengan piksel lain itu ataupun piksel lain itu merupakan tetangga dari tetangga piksel tersebut. Kriteria sebuah piksel merupakan tetangga dari piksel lain dapat berupa *4-connectivity*, *6-connectivity* ataupun *8-connectivity*.



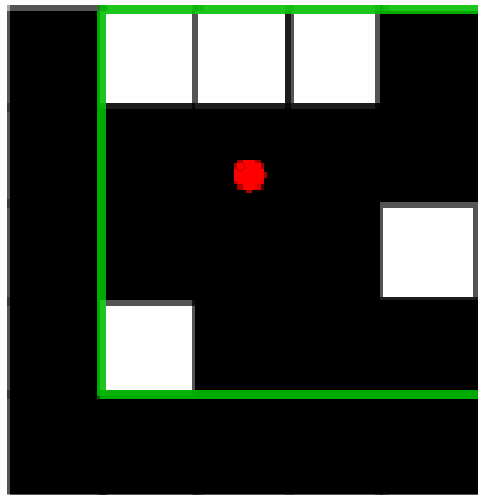
Gambar 3.17 4, 8 dan 6-connectivity

Algoritma *connected component labeling* berdasarkan banyaknya penelusuran gambar yang dilakukan dapat dibagi menjadi 3, yaitu one pass, two pass, dan multi pass. Sedangkan berdasarkan tipe perulangannya dapat dibagi menjadi 2 yaitu, *recursive* dan *sequential*. Algoritma yang digunakan pada program ini berjenis *sequential algorithm* dan merupakan two pass. Sedangkan kriteria ketetanggaan yang dipakai adalah *8-connectivity*.

3.9 *Regionprops*

Region properties (regionprops) adalah sebuah fungsi yang dimiliki MATLAB untuk mengukur sekumpulan properti-properti dari setiap *region* yang telah dilabeli dalam matriks label L . Bilangan integer positif yang merupakan elemen dari L berkorespondensi dengan *region* yang bersesuaian. *Area*, panjang *major axis*, dan panjang *minor axis* yang digunakan dalam tugas akhir ini merupakan sebagian dari properti yang dihasilkan fungsi *regionprops*.

Dalam fungsi *regionprops* sebuah obyek direpresentasikan sebagai sebuah *region* dengan pendekatan bentuk persegi panjang. Gambar 3.18 menunjukkan sebuah *region* dari kumpulan piksel berwarna putih yang direpresentasikan dengan pendekatan bentuk persegi panjang.



Gambar 3.18 Representasi region dengan pendekatan bentuk persegi panjang

Pada penelitian ini, fitur dari pendekatan letak atau posisi objek menggunakan properti BoundingBox dan properti Centroid dari fungsi *regionprops*.

Dimana Properti BoundingBox adalah persegi panjang yang memuat semua region. Sedangkan properti Centroid didefinisikan sebagai titik tengah atau pusat dari *region*. Elemen pertama dari Centroid adalah koordinat horizontal ("X" koordinat) dari objek , dan elemen kedua dari Centroid adalah koordinat vertical ("Y" koordinat) dari objek.

3.10 MATLAB (*Matrix Laboratory*)

Bahasa pemrograman sebagai media untuk berinteraksi antara manusia dan computer saat dibuat semakin mudah dan cepat. Sebagai contoh, dapat dilihat dari perkembangan bahasa pemrograman Pascal yang terus memunculkan varian baru sehingga akhirnya menjadi Delphi, demikian pula dengan Basic dengan Visual Basicnya serta C dengan C ++ Buildernya. Pada akhirnya semua bahasa pemrograman akan semakin memberikan kemudahan pemakainya (*programmer*) dengan penambahan fungsi-fungsi baru yang sangat mudah digunakan bahkan oleh pemakai tingkat pemula.

MATLAB muncul di dunia bahasa pemrograman yang cenderung dikuasai oleh bahasa yang telah mapan. Tentu saja sebagai bahasa pemrograman yang baru MATLAB akan sukar mendapat hati dari pemakai. Namun MATLAB hadir tidak dengan fungsi dan karakteristik

yang umumnya ditawarkan bahasa pemrograman lain yang biasanya hampir seragam. MATLAB dikembangkan sebagai bahasa pemrograman sekaligus alat visualisasi, yang menawarkan banyak kemampuan untuk menyelesaikan berbagai kasus yang berhubungan langsung dengan disiplin keilmuan matematika. MATLAB memiliki kemampuan mengintegrasikan komputasi, visualisasi, dan pemrograman dalam sebuah lingkungan yang tunggal dan mudah digunakan. MATLAB menyediakan beberapa pilihan untuk dipelajari, mempelajari metode visualisasi saja, pemrograman saja, atau kedua-duanya.

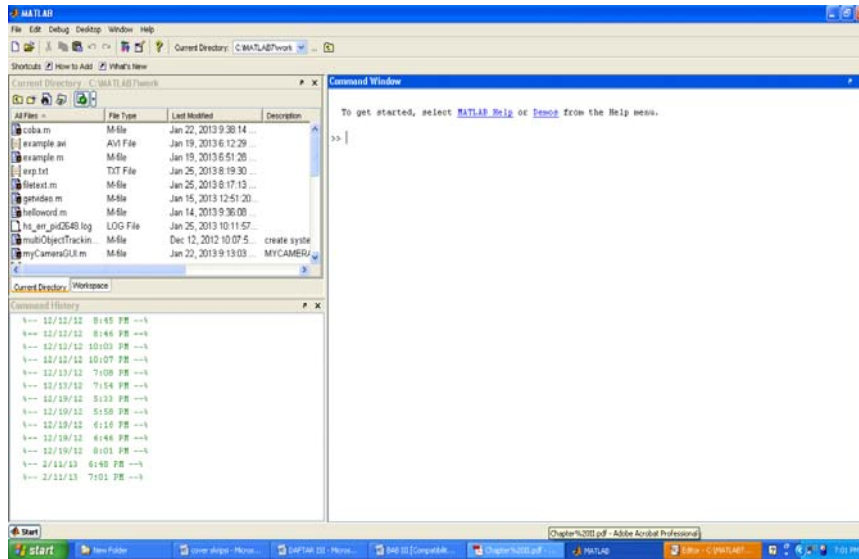
MATLAB adalah bahasa pemrograman level tinggi yang dikhususkan untuk komputasi teknis. Bahasa ini mengintegrasikan kemampuan komputasi, visualisasi, dan pemrograman dalam sebuah lingkungan yang tunggal dan mudah digunakan. MATLAB memberikan sistem interaktif yang menggunakan konsep *array* sebagai standar variabel elemennya tanpa membutuhkan pendeklarasian *array* seperti pada bahasa pemrograman lain.

Kehadiran MATLAB sebagai bahasa pemrograman memberikan jawaban sekaligus tantangan. MATLAB menyediakan beberapa pilihan untuk dipelajari, mempelajari metoda visualisasi saja, pemrograman saja, atau kedua-duanya. MATLAB memang dihadirkan bagi orang-orang yang tidak ingin disibukkan dengan rumitnya sintaks dan alur logika pemrograman, sementara pada saat yang sama membutuhkan hasil komputasi dan visualisasi yang maksimal untuk mendukung

pekerjaannya. Selain itu, MATLAB juga memberikan kemudahan bagi *programmer/developer* program yaitu untuk menjadi pembanding yang sangat handal, hal tersebut dapat dilakukan karena kekayaannya akan fungsi matematika, fisika, statistika, dan visualisasi.

3.10.1 Lingkungan Kerja MATLAB

Sebagaimana bahasa pemrograman lainnya, MATLAB juga menyediakan lingkungan kerja terpadu yang sangat mendukung dalam membangun sebuah aplikasi. Pada setiap versi MATLAB terbaru, lingkungan terpadunya akan semakin dilengkapi. Lingkungan terpadu ini terdiri dari beberapa *form* yang memiliki kegunaan masing-masing. Setiap pertama kali membuka aplikasi MATLAB, maka akan diperoleh beberapa *form*. MATLAB akan menyimpan *mode/setting* terakhir lingkungan kerja yang digunakan sebagai *mode/setting* lingkungan kerja pada saat membuka aplikasi MATLAB diwaktu berikutnya.



Gambar 3.19 Window Utama Matlab

3.11 Flowchart

Flowchart atau *Data flow diagram* adalah gambaran grafis yang memperlihatkan aliran data dari sumbernya dalam objek kemudian melewati suatu proses yang mentransformasikannya ke tujuan yang lain, yang ada pada objek lain. (Adi Nugroho, 2002:60).

Didalam pembuatan program, mendesain alur program sangat berperan penting. flowchart sangat membantu seorang programmer untuk memecahkan masalah kedalam bagian-bagian yang lebih kecil dan menganalisis alternatif – alternatif lain dalam pengoperasian program. Sehingga memudahkan para programmer dalam menentukan algoritma atau proses dari aplikasi / program yang dibuat.

Flowchart adalah sebuah alat bantu yang berguna dalam pembuatan logika / alur dalam pembuatan program karena Logika atau

proses bisa bersumber dari kegiatan penanganan informasi dari langkah-langkah dan urutan prosedur dari suatu program.

System flowchart adalah urutan proses dalam system dengan menunjukkan alat media input, output serta jenis media penyimpanan dalam proses pengolahan data.

Program flowchart adalah suatu bagan dengan simbol-simbol tertentu yang menggambarkan urutan proses secara mendetail dan hubungan antara suatu proses (instruksi) dengan proses lainnya dalam suatu program.

3.11.1 Pedoman Dalam Pemakaian *Flowchart*

Hal-hal yang perlu diketahui dalam pembuatan Flowchart, yaitu:


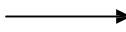



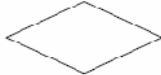

1. Flowchart digambarkan dari halaman atas ke bawah dan dari kiri ke kanan.
2. Aktivitas yang digambarkan harus didefinisikan secara hati-hati dan definisi tersebut harus dapat dimengerti oleh pembacanya.
3. Kapan aktivitas dimulai dan berakhir harus ditentukan secara jelas.
4. Setiap langkah dari aktivitas harus diuraikan dengan menggunakan deskripsi kata kerja.
5. Setiap langkah dari aktivitas harus berada pada urutan yang benar.
6. Lingkup dan range dari aktifitas yang sedang digambarkan harus ditelusuri dengan hati-hati. Percabangan-percabangan yang memotong aktivitas yang sedang digambarkan tidak perlu digambarkan pada flowchart yang sama. Simbol konektor harus digunakan dan

percabangannya diletakkan pada halaman yang terpisah atau hilangkan seluruhnya bila percabangannya tidak berkaitan dengan sistem.

7. Gunakan simbol-simbol flowchart yang standar.

3.11.2 Simbol – Simbol *Flowchart*

Simbol – simbol flowchart yang biasa dipakai adalah simbol standar yang dikeluarkan oleh ANSI dan ISO.

| SIMBOL | NAMA | KEGUNAAN |
|---|------------------------|---|
|  | TERMINATOR | Permulaan / akhir program |
|  | GARIS ALUR / FLOW LINE | Arah aliran program |
|  | PREPARATION | Proses inisialisasi / Pemberian Harga Awal |
|  | PROSES | Proses perhitungan / pengolahan data |
|  | INPUT / OUTPUT | Proses Input / Output Data |
|  | DECISION | Perbandingan pernyataan |
| | ON PAGE CONNECTOR | Penghubung Bagian – bagian flowchart yang ada dalam satu halaman |
| | OFF PAGE CONNECTOR | Penghubung Bagian – bagian flowchart yang ada dalam halaman berbeda |
| | DOKUMEN | I/O dalam format yang dicetak |
|  | COMUNICACION LINK | Tranmisi data melalui Channel komunikasi |

Tabel 3.1 Simbol – simbol Flowchart

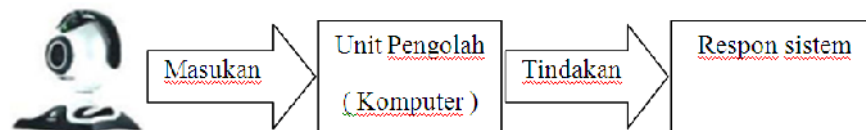
BAB IV

ANALIS DAN PERANCANGAN SISTEM

4.1 Analisis Sistem

Sistem yang akan dibuat dalam penelitian kali ini adalah suatu sistem yang mampu mendeteksi gerakan objek berwarna merah yang tertangkap oleh webcam.

Dalam sistem ini, setiap gerakan dari objek berwarna merah dapat terdeteksi secara langsung (*real time*). Objek yang terdeteksi akan ditandai dengan *boundingbox* dan posisi objek koordinat “X” dan “Y” yang akan muncul secara otomatis mengikuti setiap gerakan objek. Dan seluruh gerakan objek yang terdeteksi dalam sistem ini akan disimpan dalam file format .txt .

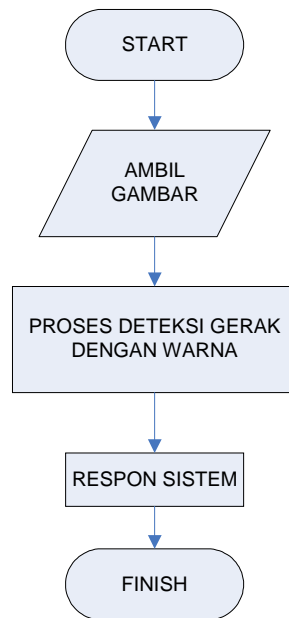


Gambar 4.1 Diagram blok sistem pendeteksi gerakan

4.2 Perancangan Sistem

Sistem ini menggunakan kamera *webcam* yang sudah tersedia atau *built in* pada *notebook Lenovo G470*, karena lebih praktis. Pada dasarnya citra bergerak atau video merupakan gabungan dari beberapa citra diam

yang berubah ubah dengan kecepatan tertentu, sehingga mata manusia menangkapnya sebagai citra yang bergerak.



Gambar 4.2 diagram flowchart sistem deteksi gerak

Seperti yang terlihat pada gambar 4.2 diagram flowchart sistem deteksi gerak suatu video agar objek yang berada di dalam dapat terdeteksi memerlukan beberapa proses sebagai berikut :

1. Ambil gambar

Saat sistem pertama kali dijalankan langkah pertama sistem akan mengambil gambar yang akan dijadikan sebagai inputan yang akan diolah dalam proses pengolahan citra. Citra inputan yang akan diproses dalam proses pengolahan citra ini berupa citra RGB

2. Proses deteksi gerak dengan warna

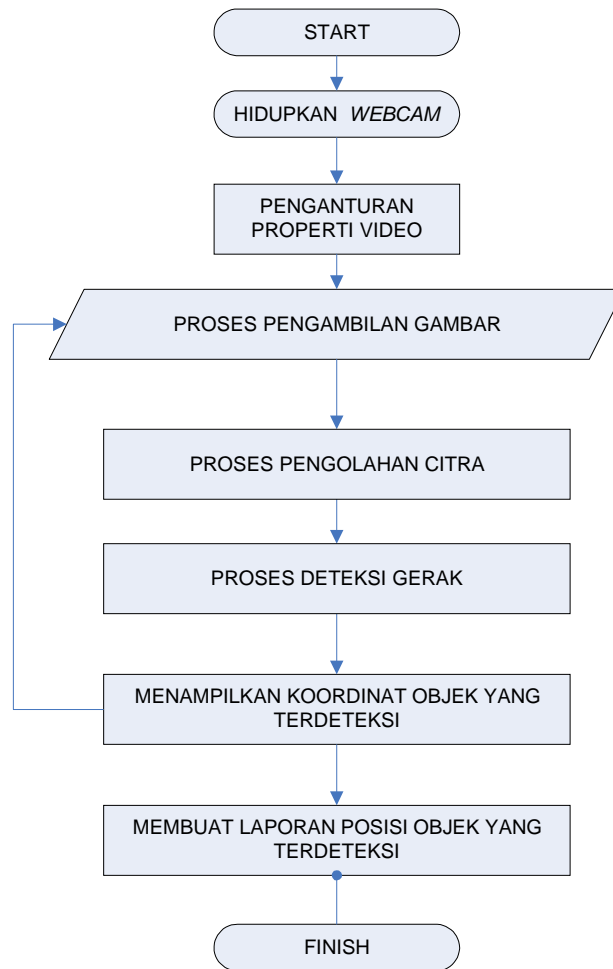
Citra inputan yang berupa citra RGB kemudian diolah dalam proses pengolahan citra agar dapat di deteksi sebagai objek yang berwarna merah dengan metode image subtracting.

3. Respon sistem

Citra hasil pengolahan citra dengan metode image subtracting kemudian akan diolah agar dapat sistem dapat merecord semua perubahan posisi pada citra yang telah terdeteksi.

4.3 Perancangan Sistem Deteksi Gerak Berdasar Warna

Perancangan sistem pada penelitian ini menggunakan perangkat lunak Matlab 7 untuk analisis citra sehingga dapat mendeteksi gerakan objek yang berwarna merah. Dalam sistem ini pertama – tama program akan mengambil menginisialisasi *webcam* pada *notebook Lenovo G470* . Setelah itu dilakukan pemanggilan terhadap webcam tersebut dan perintah untuk merekam setiap citra yang ada.



Gambar 4.3 diagram flowchart program pendeteksi gerakan secara keseluruhan

Berdasarkan gambar 4.3 diagram flowchart program pendeteksi gerakan secara keseluruhan ini terdiri dari beberapa bagian sebagai berikut :

1. Pengaktifan Webcam

Setelah webcam diinisialisasi, maka langkah selanjutnya adalah membuka atau mengaktifkan webcam agar dapat digunakan untuk mengakuisisi citra yang akan dipakai sebagai input sistem.

2. Mengatur Properti Video

Setelah mengaktifkan *webcam* maka properti video akan diatur untuk mendapatkan video sesuai spesifikasi yang dibutuhkan oleh sistem.

3. Proses Pengambilan Gambar

Video terdiri dari frame – frame yang terus berganti setiap saat sehingga terlihat seperti bergerak, maka dalam sistem ini frame atau citra yang dihasilkan akan diproses secara terpisah dan berulang – ulang pada setiap frame atau citra yang tertangkap kamera hingga sistem berhenti. Maka dalam proses pengambilan gambar ini sistem akan memisahkan frame – frame yang dihasilkan oleh *webcam* yang akan dijadikan inputan dalam proses pengolahan citra dimana citra yang dihasilkan dalam proses ini adalah citra RGB.

4. Proses pengolahan citra

Setelah dilakukan proses pengambilan citra yang menghasilkan citra RGB , maka proses selanjutnya adalah proses pengolahan citra, proses pengolah citra dalam sistem ini menggunakan metode image subtracting untuk mendeteksi objek berwarna merah.

5. Proses deteksi gerak

Setelah sistem dapat mendeteksi objek berwarna merah dengan metode image subtracting maka sistem dapat mendeteksi gerakan objek berwarna merah tersebut dengan fungsi *Region properties* (*regionprops*) untuk mendeteksi posisi dari objek berwarna merah tersebut.

6. Menampilkan koordinat objek

Setelah posisi objek terdeteksi maka sistem dapat menampilkan koordinat atau posisi dari objek berwarna merah yang terdeteksi dan mengikuti serta menunjukkan pergerakan objek tersebut dengan menampilkan perubahan koordinat objek yang terdeteksi.

7. Membuat laporan posisi yang terdeteksi

Setelah sistem menampilkan koordinat atau posisi dari objek berwarna merah yang terdeteksi maka sistem akan merecord semua perubahan posisi objek yang terdeteksi dengan menyimpan semua perubahan koordinat objek kedalam file .txt

4.4 Pengolahan Citra Pada Sistem Deteksi Gerak Berdasar Warna

Untuk mendeteksi gerakan pada sistem deteksi gerak berdasarkan warna pada file video dibutuhkan proses pengolahan citra. Dimana inputan citra dalam proses pengolahan citra dalam sistem ini didapat dari

proses pengambilan gambar atau akuisisi citra yang outputnya berupa citra yang memiliki nilai RGB.

Dalam proses pengolahan citra untuk mendeteksi gerak berdasarkan warna dari input citra RGB secara garis besar prosesnya dapat dibagi menjadi 2 tahapan antara lain :

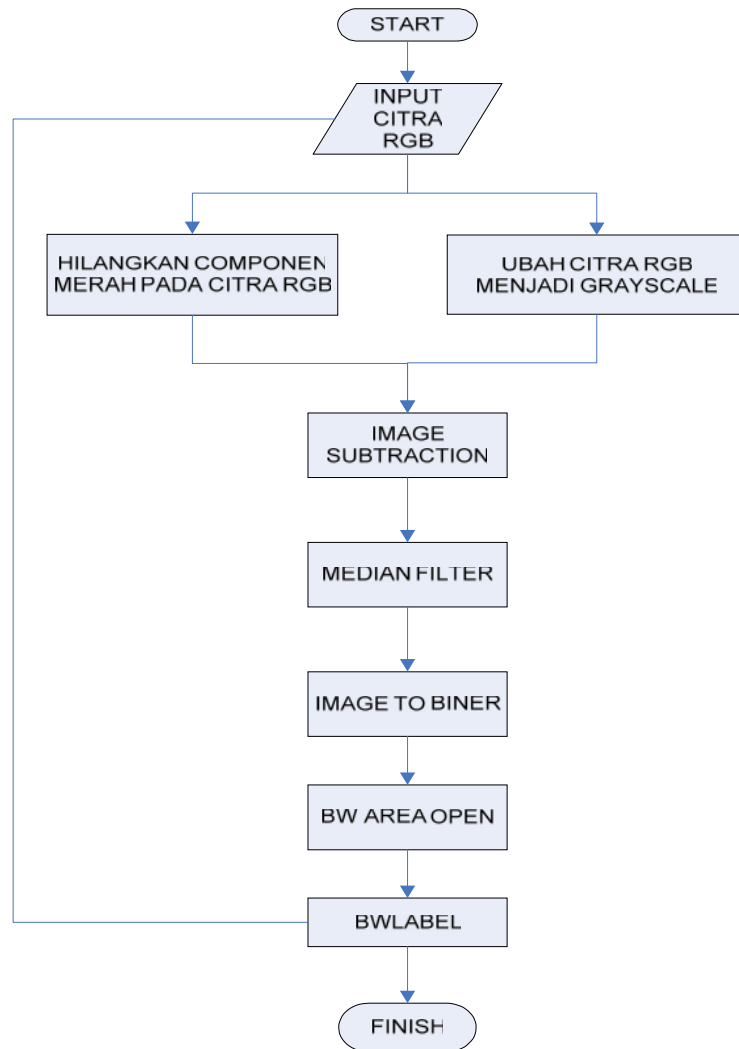
1. Mendeteksi warna dengan metode *image subtracting*
2. Mendeteksi gerakan objek dengan fungsi *regionprop*

Proses – proses dari ke dua tahapan diatas akan terus berulang pada setiap frame dari file video yang diakuisisi oleh sistem dan akan berhenti setelah sistem dihentikan.

4.4.1 Mendeteksi Warna Dengan Metode Image Subtracting

Input citra RGB yang didapat dari proses pengambilan gambar kemudian akan diproses menggunakan metode image subtracting untuk mendeteksi objek berwarna merah pada input citra RGB.

Dimana untuk mendeteksi objek berwarna merah dengan metode image subtracting dibutuhkan beberapa proses yang dapat dilihat pada gambar 4.4 diagram flowchart pengolahan citra pada sistem deteksi gerak :



Gambar 4.4 diagram flowchart pengolahan citra pada sistem deteksi gerak

Berikut adalah proses pengolahan citra pada sistem deteksi gerak :

1. Input Citra RGB

Input citra RGB yang diperoleh dari proses akuisisi citra. Setelah sistem mendapatkan input citra RGB maka langkah

pertama adalah sistem akan mengubah citra inputan tersebut menjadi 2 citra yang berbeda.

2. Mengambil Komponen Merah Pada Citra RGB

Citra inputan yang berupa citra RGB akan diambil komponen warna merahnya (red), sehingga menghasilkan citra baru yang hanya memiliki komponen warna hijau (green) dan biru (blue).

3. Mengubah Citra RGB menjadi Grayscale

Kemudian masih dari citra inputan yang berupa citra RGB yang kemudian akan diubah menjadi citra grayscale menggunakan fungsi RGB to gray yang ada pada matlab 7.

4. Image subtraction

Hasil dari langkah ke 3 atau citra grayscale akan *subtract* (dikurangi) dengan hasil dari langkah ke 2 yaitu citra RGB yang telah diambil komponen warna merahnya. Sehingga akan menghasilkan sebuah citra yang memiliki nilai grayscale yang menunjukkan area berwarna merah dari input citra RGB.

5. Median Filter

Hasil dari image subtracting kemudian di proses menggunakan fungsi median filter untuk memperjelas hasil subtracting dan menghilangkan *salt and pepper noise* (derau)

6. Image to Black White (biner)

Proses median filter akan menghasilkan citra yang telah dibersihkan deraunya dan memiliki nilai grayscale yang kemudian dari citra grayscale ini kemudian akan diubah menjadi citra hitam putih atau black white atau biner menggunakan fungsi `im2bw` pada matlab 7.

7. Black white area open

Citra biner dari proses sebelum akan kembali diproses dengan menghilangkan piksel – piksel yang berukuran kecil.

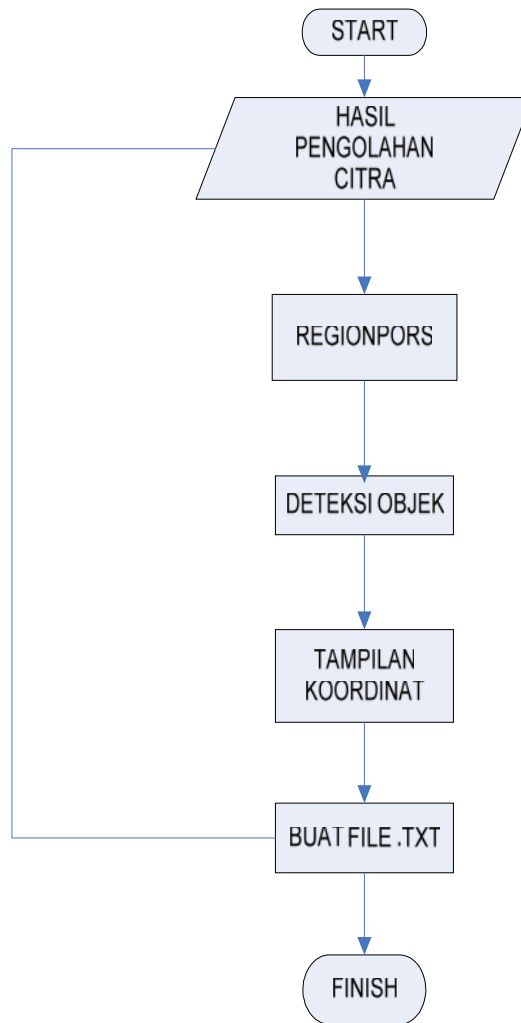
8. Black white label

Untuk melabeli semua yang piksel dari citra biner yang dihasilkan dari tahapan proses sebelumnya.

4.4.2 Mendeteksi Gerakan Objek Dengan Fungsi Regionprop

Proses pengolahan citra yang menggunakan metode image subtracting akan menghasilkan citra biner yang menunjukkan objek berwarna merah dari input citra RGB yang dihasilkan dari proses pengambilan gambar.

Kemudian untuk mendeteksi gerakan pada objek yang berwarna merah yang telah terdeteksi dalam sistem ini menggunakan fungsi `regionprop` dimana proses pendeteksian gerakan pada objek berwarna merah dapat dilihat dalam flowchart gambar 4.4 diagram flowchart pendeteksi gerakan .



Gambar 4.4 diagram flowchart pendeteksi gerakan

Berikut adalah adalah proses pengolahan citra pada sistem deteksi gerak :

1. Input citra

Output dari proses pengolahan citra yang menghasilkan citra biner yang telah dilabeli tiap pixelnya yang menunjukkan objek berwarna merah.

2. Regionprop

Input citra biner yang telah diberi label pada tiap – tiap pixelnya kemudian akan diproses dengan menggunakan 2 properti fungsi *regionprop*, antara lain :

a. *BoundingBox*

Properti *BoundingBox* dalam tahap ini berfungsi untuk memuat dan mengukur semua area citra biner yang telah diberi label.

b. *Centroid*

Properti *Centroid* dalam tahap ini berfungsi untuk menunjukkan titik tengah area citra biner yang telah diberi label.

3. Deteksi posisi objek

Dengan Properti *BoundingBox* dari fungsi *regionprop* yang ada dalam matlab 7, maka posisi dari objek berwarna merah

ini dapat ditunjukkan dengan membuat kotak berwarna merah disekitar area *regionprop*.

4. Koordinat objek

Setelah objek berwarna merah terdeteksi dengan munculnya kotak warna merah disekitar area objek yang terdeteksi, maka sistem akan menunjukkan posisi dari objek yang terdeteksi dengan memanfaatkan Properti *Centroid* dari fungsi *regionprop* untuk menunjukkan titik tengah dari objek yang terdeteksi dengan menampilkan koordinat “X” dan “Y” dari titik tersebut.

5. Membuat file .txt

Setelah posisi objek atau koordinat pada objek berwarna merah terdeteksi maka sistem akan *record* setiap posisi atau koordinat objek yang terdeteksi dari setiap frame yang dihasilkan dari proses pengambilan gambar.

BAB V

IMPLEMENTASI DAN PENGUJIAN SISTEM

5.1 Implementasi Sistem

Tahapan ini dilakukan setelah perancangan sistem selesai dilakukan dan selanjutnya diimplementasikan pada bahasa pemrograman yang digunakan. Tujuan implementasi sistem adalah untuk menerapkan perancangan yang telah dilakukan terhadap perangkat lunak sehingga nantinya maksud dan tujuan pembangunan perangkat lunak dapat tercapai.

5.1.1 Persiapan Kebutuhan Dan Instalasi Program

Program yang telah dibuat melakukan proses deteksi dan pembelajaran. Proses deteksi pada gambar tidak bergerak pada umumnya tidak memerlukan proses yang berat karena sifatnya hanya mensimulasikan data input terhadap model matematik yang telah disusun. Proses pada suatu *video* sifatnya berat dan penting, karena deteksi dilakukan terus menerus sehingga akan menyebabkan *video* terasa lebih lambat apabila spesifikasi prosesor pengguna tidak memadai, oleh karena itu sebaiknya program dijalankan pada komputer yang memiliki kemampuan komputasi yang cukup cepat. Adapun pelaksanaan implementasi ini dilakukan pada komputer dengan spesifikasi seperti pada tabel 5.1 Spesifikasi perangkat keras:

| Spesifikasi Perangkat Keras | |
|-----------------------------|-------------------------------------|
| Processor | Intel Celeron B820 @1.7 GHz |
| Memori | 2GB DDR3 |
| Harddisk | 320GB SATA |
| VGA | Intel Graphics Media Accelerator HD |

Tabel 5.1 Spesifikasi perangkat keras

Spesifikasi perangkat lunak yang digunakan mempunyai spesifikasi seperti pada table 5.2 Spesifikasi perangkat keras:

| Spesifikasi Perangkat Lunak | |
|-----------------------------|-------------------------------|
| Operating system | Windows Xp, Vista, atau Seven |
| Bahasa pemrograman | Matlab 7 |

Tabel 5.2 Spesifikasi perangkat lunak

5.1.1.1 Instalasi Matlab 7

Dalam instalasi Matlab Jika menggunakan Operating System Windows Xp dan Vista program dapat langsung di instal dengan lancar tanpa ada masalah, tetapi jika menggunakan OS Windows 7 terkadang mengalami kegagalan dalam penggunaan. Berikut adalah proses instalasi Matlab pada windows 7 sebagai berikut :

1. Ubah tema windows 7 kamu menjadi windows Clasic, caranya klik kanan pada desktop > Personalize > pilih Windows Clasic, seperti yang terlihat pada gambar 5.1 Desktop Personalize

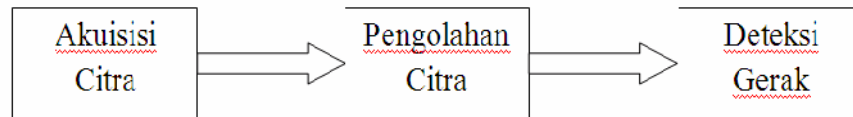


Gambar 5.1 Desktop Personalize

2. Buka setup.exe Dari master matlab dan lakukan penginstalan seperti biasa dan ikut langkah – langkah pada matlab atau lihat pada tautan <http://www.mathworks.com/videos/getting-started-with-matlab-68985.html>
3. Setelah sukses terinstal pada windows 7, sekarang klik kanan Icon Matlab pada desktop > Properties > Compatibility >Pilih Run Compatibility this Program for : Windows 2000 > Centang Disable Viual Themes.
4. Matlab sudah dapat digunakan.

5.1.2 Proses Sistem Deteksi Gerak Berdasarkan Warna

Proses pendeteksian gerak pada sistem deteksi gerak berdasarkan warna ini secara garis besar dapat dilihat pada gambar 5.2 diagram blok system deteksi gerak.



Gambar 5.2 diagram blok system deteksi gerak

5.1.3 Akuisisi Citra

Akuisisi citra adalah pengambilan gambar dalam penelitian ini akuisisi citra menggunakan webcam yang *built in* pada *notebook Lenovo G470*.

a. Mengaktifkan Webcam

Untuk dapat mengakuisisi citra dalam pengimplementasiannya digunakan fungsi video input yang terdapat pada matlab 7. Bentuk source code yang digunakan dapat dilihat pada gambar 5.3 source code video input.

```
1 % Mengambil frame video dengan menggunakan fungsi videoinput  
2 - vid = videoinput('winvideo');
```

Gambar 5.3 source code video input

Dengan source code gambar 5.3 sistem dapat mengakuisisi gambar dengan menggunakan webcam secara langsung karena adaptor 'winvideo' pada matlab7 ini berfungsi menghubungkan hardware (webcam) dengan sistem deteksi gerak. Sehingga sistem dapat mengakuisisi gambar yang ditangkap oleh webcam untuk dijadikan sebagai input video dalam sistem.

b. Pengaturan Properti Video

Pengaturan properti video dilakukan untuk mendapatkan video input sesuai dengan kebutuhan sistem. Seperti yang terlihat pada gambar 5.4 source code property video objek.

```
4 % Mengatur properti dari video objek
5 - set(vid, 'FramesPerTrigger', Inf);
6 - set(vid, 'ReturnedColorspace', 'rgb')
7 - vid.FrameGrabInterval = 2;
```

Gambar 5.4 source code property video objek

Dengan *source code* diatas maka sistem akan mendapatkan video input yang terdiri dari frame – frame atau citra yang memiliki nilai RGB dan frame dari video input diambil secara terus menerus dan akan mengakuisisi 2 frame per detik.

c. Pengambilan Gambar

Setelah mengaktifkan dan mengatur properti video input maka sistem akan memulai video streaming dengan menggunakan source code yang pada gambar 5.5 source code memulai video.

```
9 % Memulai video
10 - start(vid,
11
12 % Mengatur sebuah perulangan yang akan berhenti setelah mengakuisisi 200 frame
13 - while(vid.FramesAcquired<=200)
14
15 % Mendapat snapshot dari frame yang diambil
16 - data = getsnapshot(vid);
```

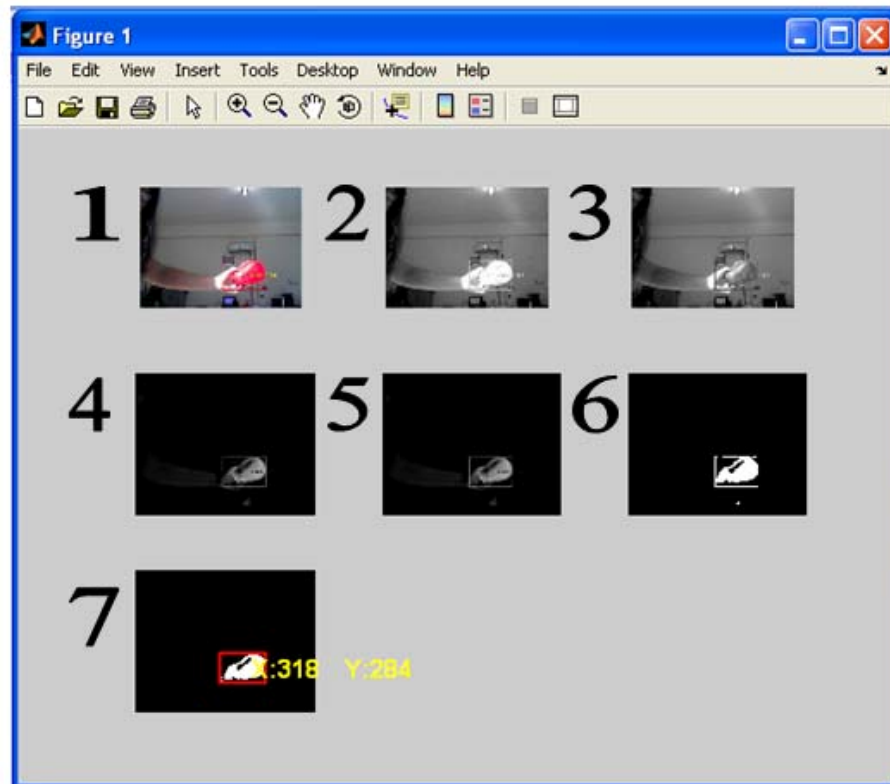
Gambar 5.5 source code memulai video

Dengan source code pada gambar 5.5 maka proses pengambilan gambar akan dimulai secara terus menerus hingga sistem mengakuisisi 200 frame yang kemudian akan diolah secara terpisah pada proses pengolahan citra seperti terlihat pada gambar 5.6 tahapan proses pengolahan citra pada sistem deteksi gerak berdasarkan warna pada file citra RGB.

5.1.4 Proses Proses Pengolahan Citra

Video adalah suatu citra atau frame yang putar secara berurutan dengan kecepatan tertentu sehingga terlihat seperti bergerak. Maka dalam sistem ini video akan diolah berdasarkan citra atau frame yang diakuisisi oleh sistem secara berurutan. Adapun tahapan proses pengolahan citra yang harus dilalui suatu citra berwarna merah agar dapat terdeteksi oleh sistem deteksi gerak berdasarkan warna dapat dilihat pada gambar 5.6

tahapan proses pengolahan citra pada sistem deteksi gerak berdasarkan warna pada file citra RGB.



Gambar 5.6 tahapan proses pengolahan citra pada sistem deteksi gerak berdasarkan warna pada file citra RGB

Seperti yang terlihat pada gambar 5.6, adapun pengolahan yang harus dilalui agar dapat mendeteksi objek berwarna merah adalah sebagai berikut :

1. Citra yang pertama adalah Inputan awal yang berupa citra RGB yang didapat dari salah satu frame video input sistem deteksi objek yang didapat dengan menggunakan source code gambar 5.7 source code untuk menampilkan gambar.

```
5 - data = imread('warnal.jpg');
```

Gambar 5.6 source code untuk menampilkan gambar

- Setelah input citra RGB dapat ditampilkan maka, langkah berikutnya dapat lihat pada gambar 5.6 yaitu Citra kedua dimana citra kedua ini adalah citra RGB yang dihilangkan komponen merahnya (*Red*) yang diperoleh dengan menggunakan *source code* gambar 5.7 *source code* untuk menghilangkan komponen merah (*red*).

```
6 - a = data(:,:,1);
```

Gambar 5.7 source code untuk menghilangkan komponen merah (*red*)

- Citra ketiga adalah hasil konversi input citra yang memiliki nilai RGB diubah menjadi citra grayscale yang diubah menggunakan *source code* gambar 5.8 *source code* image to grayscale.

```
7 - b = rgb2gray(data);
```

Gambar 5.8 source code image to grayscale

- Langkah berikutnya adalah mengurangi citra 3 dengan citra 2 dengan menggunakan metode image subtracting, dimana citra ketiga adalah citra RGB yang telah diubah menjadi grayscale, dan citra kedua adalah citra RGB yang telah dihilangkan komponen warna merahnya. Sehingga mendapatkan hasil seperti gambar 5.7 pada gambar ke 4 dimana langkah ke 4 ini menggunakan *source code* seperti gambar 5.9 *source code* image subtracting

```
8 - diff_im = imsubtract(a, b);
```

Gambar 5.9 source code image subtracting

5. Setelah didapat citra hasil image subtracting pada sistem, maka langkah selanjutnya adalah dengan memfilter gambar dengan menggunakan fungsi medfilter dengan source code pada gambar 5.10 source code median filter.

```
9 - diff_im_a = medfilt2(diff_im, [3 3]);
```

Gambar 5.10 source code median filter

6. Langkah selanjutnya adalah pengolahan citra dengan mengubah citra hasil median filter yang berupa citra grayscale diubah menjadi black white atau citra biner. Kemudian citra yang berukuran kurang dari 300x dihilangkan dengan menggunakan source code pada gambar 5.11 source code image to bw dan bwareaopen.

```
10 - diff_im_b = im2bw(diff_im_a, 0.18);  
11 - diff_im_c = bwareaopen(diff_im_b, 300);
```

Gambar 5.11 source code image to bw dan bwareaopen

7. Langkah berikutnya adalah labeling dimana pada proses ini system akan menghubungkan pixel – pixel yang terdapat dalam citra hasil pengolahan citra dengan source code gambar 5.11 source code image to bw dan bwareaopen kemudian citra hasil source code gambar 5.11 diolah kembali menggunakan source code gambar 5.12 source code labeling image.

```
12 - bw = bwlabel(diff_im_c, 8);
```

gambar 5.12 source code labeling image

8. Langkah 1 sampai 7 adalah tahap – tahap pengolahan citra yang harus dilalui suatu frame dalam video input agar dapat mendeteksi objek berwarna merah yang ada dalam frame tersebut. Setelah objek berwarna terdeteksi maka objek yang terdeteksi tersebut akan diberi tanda dan diberi koordinat untuk menunjukkan posisi objek. Seperti yang terdapat pada gambar 5.13 source code regionprops.

```
23 - stats = regionprops(bw, 'BoundingBox', 'Centroid');
24 - for object = 1:length(stats)
25 -     bb = stats(object).BoundingBox;
26 -     bc = stats(object).Centroid;
27 -     rectangle('Position',bb,'EdgeColor','r','LineWidth',2)
28 -     a=text(bc(1)+15,bc(2), sprintf(' X: ', num2str(round(bc(1))), ' Y: ', num2str(round(bc(2)))));
29 -     set(a, 'FontName', 'Arial', 'FontWeight', 'bold', 'FontSize', 12, 'Color', 'yellow');
30
31 - end
32 - clear all
```

Gambar 5.13 source code regionprops

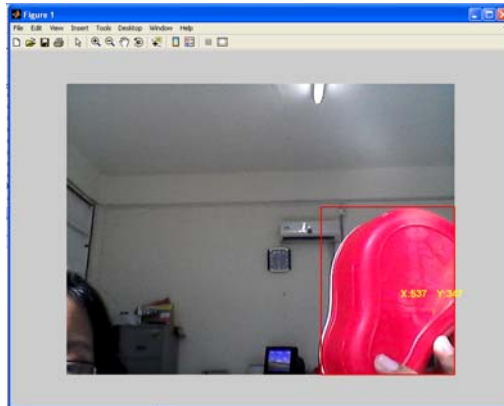
5.2 Pengujian Sistem

Proses pengujian suatu program bertujuan untuk memastikan bahwa program yang telah dibuat dapat di implementasikan secara akurat dan tepat. Sehingga pengguna dapat menggunakan program ini dengan baik sesuai dengan fungsi program tersebut dibuat.

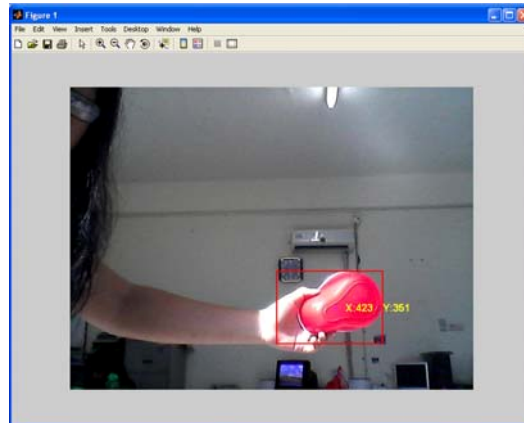
Sesuai dengan tujuan dari sistem ini yaitu menghasilkan aplikasi yang dapat mendeteksi objek berwarna merah pada file video, maka pengujian sistem deteksi gerak berdasarkan file video menggunakan video yang terekam secara langsung menggunakan kamera *webcam* yang sudah tersedia atau *built in* pada *notebook Lenovo G470*.

5.2.1 Pengujian 1 (Kondisi Terang)

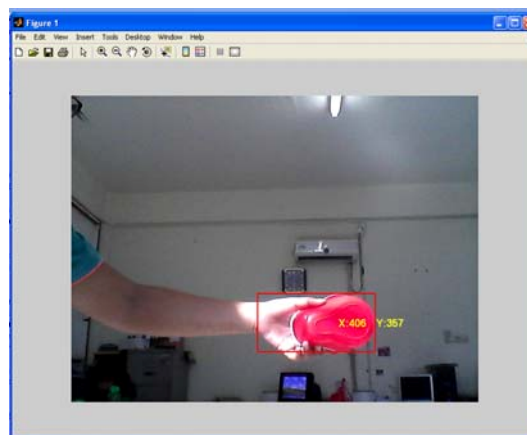
Pada pengujian pertama dilakukan pengujian deteksi objek yang dilakukan pada ruangan yang terdapat dilantai 2 gedung perkantoran yang dilengkapi dengan jendela kaca. Pengujian pertama ini dilakukan pada siang hari sehingga intensitas cahaya pada pada ruang pengujian cukup terang, saat sistem mulai dijalankan objek yang akan dideteksi berada cukup dekat dengan kamera yakni 10 cm, pada pengujian pertama ini dapat menghasilkan kesimpulan bahwa objek dapat diterima dengan baik oleh kamera, dan gambar dapat di proses dengan baik sehingga objek akan dideteksi dapat terdeteksi sesuai dengan harapan yakni munculnya boundingbox yang mengikuti gerakan objek yang terdeteksi dan koordinat objek yang menunjukkan perubahan posisi objek selama sistem mendeteksi objek tersebut. Seperti yang terlihat dalam gambar 5.4, gambar 5.5 dan gambar 5.6.



Gambar 5.4 Objek pertama kali dideteksi



Gambar 5.5 objek bergerak kekanan

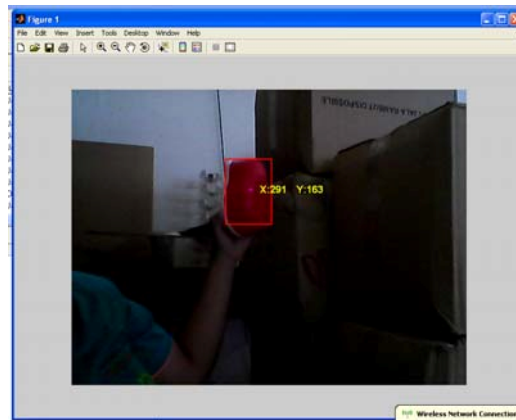


Gambar 5.6 objek bergerak menjauh dari kamera

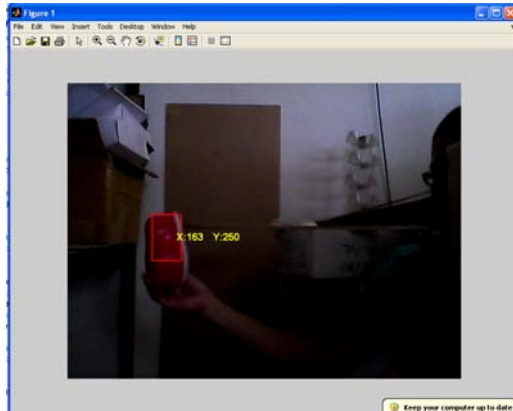
5.2.2 Pengujian 2 (Kondisi Redup)

Pada pengujian kedua dilakukan pengujian deteksi objek pada ruangan yang tertutup dan tidak dilengkapi oleh penerangan yang cukup. Saat system mulai dijalankan objek yang akan dideteksi berada cukup dekat dengan kamera, namun hasil pengujian pada pengujian pertama ini berbeda dengan hasil pengujian kedua.

Pada pengujian yang kedua ini objek dapat terdeteksi dengan baik, seperti yang terlihat pada gambar 5.7. Namun saat objek semakin menjauh maka area yang terdeteksi yang ditandai dengan boundingboxpun berkurang tidak mencakup seluruh area objek yang terdeteksi serti yang terlihat pada gambar 5.8



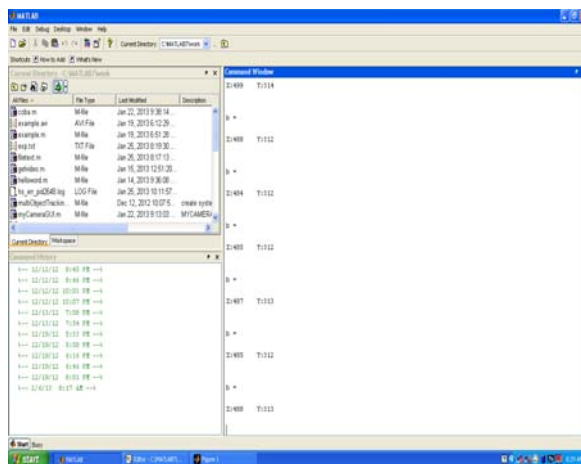
Gambar 5.8 Objek pertama kali dideteksi pada pengujian kedua



Gambar 5.9 Objek menjauh dari kamera

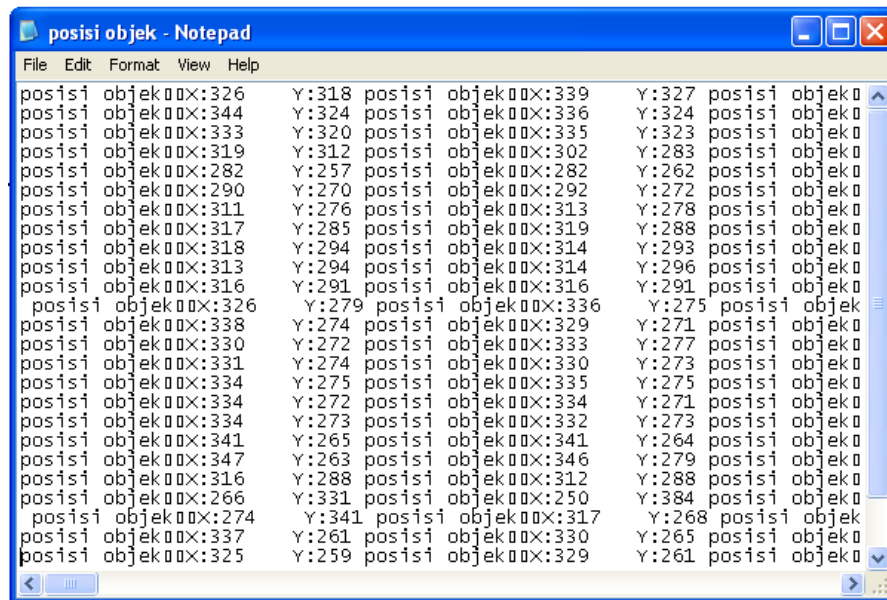
5.2.3 Pengujian Hasil *Report* Dari Sistem Deteksi Gerak Berdasarkan Warna Pada File Video Berupa File .txt

Saat sistem mulai mendeteksi objek yang berwarna merah maka akan muncul boundingbox dan koordinat pada objek yang terdeteksi. Koordinat objek yang terdeteksi selain muncul pada tampilan utama sistem juga akan muncul pada windows utama matlab seperti yang terlihat pada gambar 5.9



Gambar 5.9 Windows utama matlab 7 saat sistem berjalan

Dari windows utama matlab inilah setiap perubahan posisi objek yang terdeteksi dapat terekam kemudian dapat tersimpan dalam file .txt dengan nama posisi objek.txt yang terdapat pada folder work seperti yang terlihat dalam gambar 5.10



Gambar 5.6 Laporan perubahan posisi objek dalam file .txt

Pada gambar 5.6 Laporan perubahan posisi objek dalam file .txt dapat dilihat system berhasil merecord perubahan posisi objek yang terdeteksi dengan menampilkan koordinat “X” dan “Y” dari objek tersebut.

BAB VI

PENUTUP

6.1 Kesimpulan

Dari hasil pengujian dan analisis maka dapat disimpulkan hal – hal sebagai berikut :

1. Metode *image subtracting* dapat diimplementasikan untuk mendeteksi gerakan objek pada file video.
2. Metode *image subtracting* ini dapat dipergunakan untuk jika dari hasil image subtracting dilakukan proses pengolahan citra sebagai berikut :
 - a. *Median filter*
 - b. *Image to Black white*
 - c. *Bw area open*
 - d. *Bw label*
 - e. *Regionprops*
3. Dari pengujian yang dilakukan dengan webcam pada ruangan dengan intensitas cahaya yang cukup terang didapatkan hasil pengujian yang sempurna sesuai dengan harapan. Dimana objek berwarna merah dapat terdeteksi dan pergerakannya dapat terekam dengan sempurna. Dan dari pengujian yang dilakukan dengan webcam pada ruangan dengan intensitas cahaya yang kurang, sistem dapat mendeteksi objek berwarna merah dan pergerakannya dapat terekam, namun hasil pendeteksian kurang sempurna.

4. Penentuan posisi *webcam* dan pencahayaan memiliki pengaruh besar dalam pengenalan objek, untuk itu dibutuhkan penyesetan *webcam* dan cahaya yang cukup agar sistem dapat berjalan sesuai yang diharapkan.
5. Tingkat kemampuan program pendeteksi gerakan akan cenderung menurun, dikarenakan *webcam* tidak dapat merekam gambar secara sempurna. Sensitivitas *webcam* untuk mendeteksi suatu objek tergantung pada intensitas cahaya yang ada disekitar objek. Apabila pada cahaya semakin terang, tingkat sensitivitas *webcam* semakin bagus. Namun apabila pada cahaya semakin gelap, tingkat sensitivitas *webcam* semakin menurun.
6. Kualitas *webcam* juga berpengaruh untuk mengolah citra yang dihasilkan dari kecepatan pergerakan objek tersebut. Apabila benda bergerak terlalu cepat dengan kualitas *webcam* yang bagus, maka akan memperoleh hasil pengolahan citra yang maksimal.
7. Hasil dari pengujian program deteksi gerak ini adalah program dapat berjalan sebagaimana mestinya akan tetapi program ini kurang maksimal apabila obyek yang ditangkap jarak antara kamera dengan obyek tersebut terlalu jauh, atau intensitas cahaya disekitar objek akan di deteksi kurang sehingga sistem tidak dapat mendeteksi objek berwarna merah. Atau sistem dapat mendeteksi objek berwarna merah tersebut namun hanya sebagian dari objek tersebut.

8. Implementasi image subtracting pada sistem pendeteksi gerak berdasar objek ini dapat mendeteksi lebih dari satu objek berwarna merah dan hasil pengujian dapat terecord dengan baik, namun hasil record dari sistem ini tidak dapat menunjukkan berapa objek yang terdeteksi saat system dijalankan namun semua koordinat dari objek yang terdeteksi tetap ter-*record* .

6.2 Saran

Untuk penelitian berikutnya penulis meyarankan adanya perbaikan – perbaikan sebagai berikut :

1. Diharapkan pada penelitian selanjutnya sistem yang dihasilkan tidak hanya mendeteksi satu warna saja tetapi terdapat banyak pilihan warna sesuai dengan kebutuhan .
2. Diharapkan pada penelitian selanjutnya ditambah interface atau antarmuka program sehingga memudahkan *user* dalam mengakses sistem.
3. Agar user dapat menentukan lamanya proses pengambilan gambar diharapkan pada penelitian selanjutnya dapat ditambahkan timer yang dapat diset secara otomatis sesuai dengan keinginan atau kebutuhan user.

4. Record atau laporan sebagai output dari sistem tidak hanya menunjukkan perubahan posisi dari objek tunggal tetapi dapat menunjukkan berapa banyak objek yang terdeteksi dan perubahan posisi dari objek – objek tersebut.

DAFTAR PUSTAKA

Castleman K.R. (1996). *Digital image processing*. New Jersey : Prentice Hall.

Catur, Ibnu Mustofa. (2008) Monitoring Gerakan pada Ruangan Menggunakan Webcam dan Motor Stepper, *Skripsi*, Jurusan Teknik Informatika Fakultas Sains Dan Teknologi Universitas Islam Negeri Malang, Malang

<http://asanisembiring.files.wordpress.com/2012/02/operasi-operasi-dasar-pengolahan-citra-dijital.pdf>. Akses tanggal 28 Januari 2013

http://informatika.stei.itb.ac.id/~rinaldi.munir/Buku/Pengolahan%20Citra%20Digital/Bab4_Operasi-operasi%20Dasar%20Pengolahan%20Citra%20Dijital.pdf.

Akses tanggal 28 Januari 2013

<http://library.binus.ac.id/eColls/eThesis/Bab2/2008-1-0278mtif%20Bab%202.pdf>. Akses tanggal 28 Januari 2013

Mahanani, Putri R. (2010) Tracking Obyek Menggunakan Metode Counterpropagation, Institut Teknologi Sepuluh Nopember, Surabaya

Pressman, Roger S.(2002) Rekayasa Perangkat Lunak : Pendekatan Praktisi (Buku 1), Andi, Yogyakarta

Putra, Darma. 2010. Pengolahan Citra Digital, Andi, Yogyakarta

Wijaya, Marvin CH & Agus Prijono, (2007) Pengolahan Citra Digital Menggunakan Matlab. Penerbit : Infomatika, Bandung

Yoga, Benedictus Budi Putranto, Widi Hapsari dan Katon Wijana.(2010)
Segmentasi Warna Citra Dengan Detekdi Warna HSV Untuk Mendeteksi Objek,
Universitas Kristen Duta Wacana Yogyakarta, Yogyakarta

```

% Mengambil frame video dengan menggunakan fungsi videoinput
vid = videoinput('winvideo');

% Mengatur properti dari video objek
set(vid, 'FramesPerTrigger', Inf);
set(vid, 'ReturnedColorspace', 'rgb')
vid.FrameGrabInterval = 2;

% Memulai video start(vid)
aviobj=avifile('123rgb.avi','compression','None');

% Mengatur sebuah perulangan yang akan berhenti setelah mengakuisisi 200 frame
while(vid.FramesAcquired<=50)
    for k=1:50

        % Mendapat snapshot dari frame yang diambil
        data = getsnapshot(vid);

        % Memulai mendeteksi objek berwarna merah secara real time
        % Kita harus mengurangi komponen warna merah
        % Dari image grayscale untuk diambil komponen warna merah dari image tersebut

        diff_imb = imsubtract(data(:,:,3),rgb2gray(data));

```

```
diff_img = imsubtract(data(:,:,2),rgb2gray(data));
diff_im = imsubtract(data(:,:,1), rgb2gray(data));

%Menggunakan median filter untuk menghilangkan noise
diff_imb = medfilt2(diff_imb, [3 3]);
diff_img = medfilt2(diff_img, [3 3]);
diff_im = medfilt2(diff_im, [3 3]);

%Ubah hasil image grayscale menjadi binary image

diff_imb = im2bw(diff_imb,0.18);
diff_img = im2bw(diff_img,0.18);
diff_im = im2bw(diff_im,0.18);

% Hapus semua pixel yang kurang dari 300px

diff_imb = bwareaopen(diff_imb,300);
diff_img = bwareaopen(diff_img,300);
diff_im = bwareaopen(diff_im,300);

% melabeli semua pixel yang terhubung pada image

bwb = bwlabel(diff_imb, 8);
bwg = bwlabel(diff_img, 8);
bw = bwlabel(diff_im, 8);
```

```

% menganalisis hasil pengolahan citra

% mengatur properti dari setiap label dalam wilayah region

statsb = regionprops(bwb, 'BoundingBox', 'Centroid');

statsg = regionprops(bwg, 'BoundingBox', 'Centroid');

stats = regionprops(bw, 'BoundingBox', 'Centroid');

% Menampilkan image

f=subplot(1,1,1);imshow(data);

hold on

% perulangan untuk menandai objek berwarna merah dengan rectangular box
for object = 1:length(stats)

    bb = stats(object).BoundingBox; bc =
    stats(object).Centroid;

    rectangle('Position',bb,'EdgeColor','r','LineWidth',2)

    plot(bc(1),bc(2), '-m+')

    a=text(bc(1)+15,bc(2), strcat('X: ', num2str(round(bc(1))), ' Y: ', num2str(round(bc(2)))));

    set(a, 'FontName', 'Arial', 'FontWeight', 'bold', 'FontSize', 12, 'Color', 'yellow');

    b=strcat('X: ', num2str(round(bc(1))), ' Y: ', num2str(round(bc(2))))

    % Menbuat report ke dalam file notepad posisi objek.txt

    TheText = b;

```

```

TheFile = fullfile(tempdir, 'posisi objek red.txt');
fid = fopen('posisi objek red.txt', 'A');
if fid == -1; error('Cannot open file: %s', TheFile); end
fprintf(fid,'posisi objek\n\n');
fprintf(fid,'%s %6.2f %12.8f\n', TheText);
fclose(fid);

end

for objectg = 1:length(statsg)
    bbg = statsg(objectg).BoundingBox; bcg =
    statsg(objectg).Centroid;
    rectangle('Position',bbg,'EdgeColor','g','LineWidth',2)
    plot(bcg(1),bcg(2), '-m+')
    ag=text(bcg(1)+15,bcg(2), strcat('X: ', num2str(round(bcg(1))), ' Y: ', num2str(round(bcg(2)))));
    set(ag, 'FontName', 'Arial', 'FontWeight', 'bold', 'FontSize', 12, 'Color', 'yellow');
    bg=strcat('X: ', num2str(round(bcg(1))), ' Y: ', num2str(round(bcg(2))))
    % Menbuat report ke dalam file notepad posisi objek.txt

    TheText = bg;

    TheFile = fullfile(tempdir, 'posisi objek green.txt');
    fidg = fopen('posisi objek green.txt', 'A');
    if fidg == -1; error('Cannot open file: %s', TheFile); end
    fprintf(fidg,'posisi objek\n\n');
    fprintf(fidg,'%s %6.2f %12.8f\n', TheText);
    fclose(fidg);

end

```

```

for objectb = 1:length(statsb)
    bbb = statsb(objectb).BoundingBox; bcb =
    statsb(objectb).Centroid;
    rectangle('Position',bbb,'EdgeColor','b','LineWidth',2)
    plot(bcb(1),bcb(2), '-m+')
    ab=text(bcb(1)+15,bcb(2), strcat('X: ', num2str(round(bcb(1))), ' Y: ', num2str(round(bcb(2)))));
    set(ab, 'FontName', 'Arial', 'FontWeight', 'bold', 'FontSize', 12, 'Color', 'yellow');
    bblue=strcat('X: ', num2str(round(bcb(1))), ' Y: ', num2str(round(bcb(2))))
    % Menbuat report ke dalam file notepad posisi objek.txt
    TheText = bblue;
    TheFile = fullfile(tempdir, 'posisi objek blue.txt');
    fidb = fopen('posisi objek blue.txt', 'A');
    if fidb == -1; error('Cannot open file: %s', TheFile); end
    fprintf(fidb,'posisi objek\n\n');
    fprintf(fidb,'%s %6.2f %12.8f\n', TheText);
    fclose(fidb);
end

hold off F=getframe(f);
aviobj=addframe(aviobj,f);
end
end
% kedua perulangan berakhir disini

```

```
% Stop pengambilan gambar.
```

```
stop(vid);
```

```
% menghapus semua data yang tersimpan dalam memori.
```

```
flushdata(vid);
```

```
% Hapus semua variabel
```

```
clear all
```

```
sprintf('%s', 'Posisi objek dapat dilihat dalam folder work ')
```