

BAB II TINJAUAN PUSTAKA

2.1 Studi Literature

Pada tahun 2019, Herdianto H., melakukan perbandingan antara metode *template matching* dengan metode *background subtraction* untuk mendeteksi objek orang. Penelitian ini menggunakan data video yang berdurasi 15 detik berdasarkan empat kondisi, yaitu pagi terang, pagi hujan, malam terang dan malam hujan. *File* video yang digunakan berformatkan .mp4 dengan resolusi video 1920 x 1080.pixel dengan frame rate 10 *frames/second*. Hasil dari pengujian penelitian ini, dalam mendeteksi objek orang menggunakan metode *template matching* pada kondisi pagi, siang dan malam hari ketepatannya mencapai akurasi 100%, sedangkan untuk pengujian metode *background subtraction* hasil ketepatannya belum akurat.

Pada tahun 2018, Yolinda F., dkk. melakukan perbandingan algoritma *histogram of oriented gradient* (HOG) dan *gaussian mixture model* (GMM) untuk mendeteksi orang. Penelitian ini melakukan uji coba dengan memasukkan dua *file* video dengan dua kondisi, yaitu kondisi pertama yang dimana objek orang saling berhimpitan dengan objek orang lainnya (kondisi orang padat) dan kondisi kedua yang dimana objek orang berdekatan dengan objek yang lain (orang dengan kursi). Hasil uji coba kedua kondisi diatas menghasilkan bahwa algoritma *histogram oriented gradient* memiliki akurasi sebesar 87,5%, sedangkan *Gaussian mixture model* memiliki akurasi sebesar 0-81,7%.

Pada tahun 2016, Khairul U., dkk. menerapkan metode *background subtraction* dan operasi morfologi untuk mendeteksi orang. Penelitian ini menggunakan data video yang berdurasi 10 detik berdasarkan beberapa kondisi, seperti dalam ruangan, malam hari, luar ruangan, dan *background* yang berbeda. *File* video yang digunakan berformatkan .mp4 dengan

resolusi video 320x240.*pixel*. Hasil dari pengujian penelitian ini, dalam mendeteksi objek orang memiliki akurasi sebesar 30%.

Berdasarkan penelitian terdahulu yang telah dijelaskan diatas menunjukkan bahwa mendeteksi orang memiliki kekurangan, seperti keterbatasan dalam durasi tiap *file* video, hasil akurasi yang relatif kecil, dan berfokus pada tipe file .mp4. Untuk mengatasi kekurangan tersebut penelitian ini menggunakan metode *background subtraction* dengan beberapa teknik tambahan, seperti *threshold*, *median filter*, dan *connected component labeling*. *File* video yang digunakan memiliki durasi waktu 15 detik dengan format video .mp4.

2.2 Definisi Citra Digital

(Indrawati, 2013) Citra atau bisa disebut dengan gambar merupakan salah satu komponen dari multimedia yang memegang peranan penting karena mengandung informasi dalam bentuk visual. Citra memiliki lebih banyak informasi yang dapat disampaikan dibandingkan dalam bentuk teks. Sedangkan menurut (Fauzi, 2007) Citra merupakan kumpulan elemen-elemen gambar (*pixel*) yang secara keseluruhan merekam suatu adegan (*scene*) melalui pengindera visual (*camera*). Untuk kebutuhan pengolahan dengan bantuan *computer*, citra disajikan dalam bentuk diskrit yang disebut citra digital.

(Sutoyo et. al. 2009), menyatakan “Citra adalah suatu representasi (gambaran), kemiripan, atau imitasi dari suatu objek”. Citra sebagai keluaran suatu sistem perekaman data dapat bersifat optik berupa foto, bersifat analog berupa sinyal-sinyal video seperti gambar pada monitor televisi, atau bersifat digital yang dapat langsung disimpan pada suatu media penyimpanan. Perekaman data citra dapat dibagi menjadi dua, yaitu:

a. Citra Analog

Citra analog terdiri dari sinyal-sinyal elektromagnetik yang tidak dapat dibedakan sehingga pada umumnya tidak dapat ditentukan ukurannya.

Citra analog mempunyai fungsi yang kontinu. Hasil perekaman dapat bersifat optik yakni berupa *foto* (film *foto* konvensional) dan bersifat sinyal video seperti gambar pada monitor televisi.

b. Citra Digital

Citra digital terdiri dari sinyal-sinyal yang dapat dibedakan dan mempunyai fungsi yang tidak kontinu yakni berupa titik-titik warna pembentuk citra. Hasil perekaman citra digital dapat disimpan pada suatu media penyimpanan.

Representasi citra dari fungsi malar (kontinu) menjadi nilai-nilai diskrit disebut *digitalisasi*. Citra yang dihasilkan inilah yang disebut citra digital (*Digital Image*). Pada umumnya citra digital berbentuk empat persegi panjang, dan dimensi ukurannya dinyatakan sebagai Tinggi x Lebar (Lebar x Panjang). Citra digital yang tingginya N, lebarnya M, memiliki L derajat keabuan dapat dianggap sebagai fungsi (Munir, 2004).

2.3 Elemen-Elemen Dasar Citra

(Sutoyo et al. 2009), menjelaskan elemen-elemen dasar citra, yaitu:

a. Kecerahan (*Brightness*)

Kecerahan (*Brightness*) adalah intensitas cahaya yang dipancarkan oleh *piksel* dari citra yang dapat ditangkap oleh sistem penglihatan. Kecerahan pada sebuah titik (*piksel*) didalam citra merupakan intensitas rata-rata dari suatu area yang melingkupinya.

b. Kontras (*Contrast*)

Kontras (*Contrast*) adalah sebaran terang dan gelap dalam sebuah citra. Pada citra yang baik, komposisi gelap dan terang tersebar secara merata.

c. Kontur (*Contour*)

Kontur (*Contour*) adalah keadaan yang ditimbulkan oleh perubahan intensitas pada *piksel-piksel* yang bertetangga.

d. Warna (*Color*)

Warna (*Color*) adalah persepsi yang ditangkap sistem visual terhadap perubahan panjang gelombang cahaya yang dipantulkan oleh objek. Setiap warna memiliki panjang gelombang yang berbeda-beda. Warna merah memiliki panjang gelombang (λ) yang paling tinggi, sedangkan warna violet mempunyai panjang gelombang (λ) yang paling rendah.

e. Bentuk (*Shape*)

Bentuk adalah properti *intrinsic* dari objek tiga dimensi, dengan pengertian bahwa bentuk merupakan properti *intrinsic* utama untuk sistem visual orang.

f. Tekstur (*Texture*)

Tekstur (*Texture*) adalah sifat-sifat atau karakteristik yang dimiliki oleh suatu daerah yang cukup besar sehingga secara alami sifat-sifat tadi dapat berulang dalam daerah tersebut.

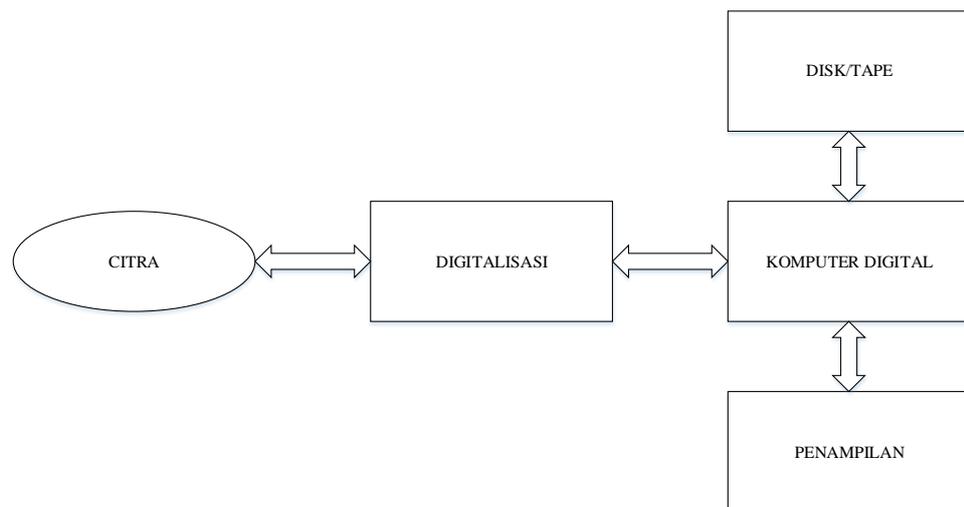
2.4 Definisi Pengolahan Citra Digital

(Yudistira, 2010), pengolahan citra adalah pemrosesan citra, khususnya dengan menggunakan *computer*, menjadi citra yang kualitasnya lebih baik. Tujuan utama pengolahan citra adalah agar citra yang mengalami gangguan mudah diinterpretasi oleh orang maupun mesin (*computer*). Teknik pengolahan citra digital adalah mentransformasikan citra dua dimensi menjadi citra lain dengan menggunakan *computer*.

Proses ini mempunyai ciri data masukan dan informasi keluaran yang berbentuk citra. Jadi masukannya berupa citra dan keluarannya juga berbentuk citra, dengan kualitas yang lebih baik dari citra masukan. Beberapa contoh operasi pengolahan citra adalah pengubahan kontras citra, penghilangan derau (*noise*) dengan operasi penapisan (*filtering*), penajaman (*sharpening*), pemberian warna semu (*pseudocoloring*), dan sebagainya. Operasi-operasi tersebut akan diterapkan pada pengolahan citra apabila:

1. Digunakan untuk meningkatkan kualitas penampakan atau menonjolkan beberapa aspek informasi yang terkandung dalam citra dengan perbaikan atau modifikasi citra.
2. Perlu pengelompokkan, pencocokan atau pengukuran elemen pada citra.
3. Perlu penggabungan sebagian citra dengan bagian citra lainnya.

Berikut ini adalah blok sistem pengolahan citra digital:



Gambar 2.1 Diagram Blok Sistem Pengolahan Citra Digital

Proses pengolahan data dapat dilakukan oleh vb, baik berupa mikrokomputer sederhana (*microprocessor based computer*) atau komputer besar (*mainframe computer*), tergantung jumlah data dan jenis pengolahan. Proses penampil data merupakan salah satu segi yang penting karena bagaimanapun juga citra digital hasil olahan harus dapat dinilai oleh mata orang melalui suatu penampil (*display*). Penampil yang digunakan biasanya berupa suatu *graphic* monitor atau suatu *graphic* printer/plotter.

2.5 Operasi Pengolahan Citra

(Wijaya dan Prijono, 2007), menjelaskan operasi-operasi yang dilakukan dalam pengolahan citra banyak ragamnya. Namun, secara umum pada pengolahan citra terdapat enam jenis operasi pengolahan, yaitu:

1. Peningkatan kualitas citra (*image enhancement*)

Jenis operasi ini bertujuan untuk memperbaiki kualitas citra dengan cara memanipulasi parameter-parameter citra. Dengan operasi-operasi ini, ciri-ciri khusus yang terdapat didalam citra lebih ditonjolkan. Berikut contoh operasi peningkatan kualitas citra:

- a. Perbaikan kontras gelap/terang
- b. Perbaikan tepian objek (*Edge enhancement*)
- c. Penajaman (*Sharpening*)
- d. Pemberian warna semu (*Pseudocoloring*)
- e. Penapisan derau (*Noise filtering*)

2. Restorasi citra (*image restoration*)

Operasi ini bertujuan menghilangkan atau meminimumkan cacat pada citra. Tujuan restorasi citra hampir sama dengan operasi peningkatan kualitas citra. Bedanya, pada restorasi citra penyebab degradasi gambar diketahui. Berikut contoh operasi restorasi citra:

- a. Penghilangan derau (*noise*)
- b. Penghilangan kesamaran (*deblurring*)

3. Kompresi citra (*image compression*)

Jenis operasi ini dilakukan agar citra dapat direpresentasikan dalam bentuk yang lebih kompak sehingga memerlukan memori yang lebih sedikit. Hal penting yang harus diperhatikan dalam kompresi citra adalah citra yang telah dikompresikan harus tetap mempunyai kualitas gambar yang bagus. Contoh metode kompresi citra adalah metode JPEG. Misalkan ada citra kapal yang berukuran 258 KB. Hasil kompresi citra dengan metode JPEG dapat mereduksi ukuran citra semula sehingga menjadi 49 KB.

4. Segmentasi citra (*image segmentation*)

Operasi ini adalah suatu tahap pada proses analisis citra yang bertujuan untuk memperoleh informasi yang ada dalam citra tersebut dengan membagi citra ke dalam daerah-daerah terpisah dimana setiap daerah adalah homogeny dan mengacu pada sebuah kriteria keseragaman yang jelas. Segmentasi yang dilakukan pada

citra harus tepat agar informasi yang terkandung di dalamnya dapat diterjemahkan dengan baik.

5. Analisis citra (*image analysis*)

Jenis operasi ini bertujuan menghitung besaran kuantitatif dari citra untuk menghasilkan deskripsinya. Teknik analisis citra mengekstraksi ciri-ciri tertentu yang membantu dalam identifikasi objek. Proses segmentasi kadangkala diperlukan untuk melokalisasi objek yang diinginkan dari sekelilingnya. Berikut contoh operasi analisis citra:

- a. Pendeteksian tepi objek (*edge detection*)
- b. Ekstraksi batas (*boundary*)
- c. Representasi daerah (*region*)

6. Rekonstruksi citra (*image reconstruction*)

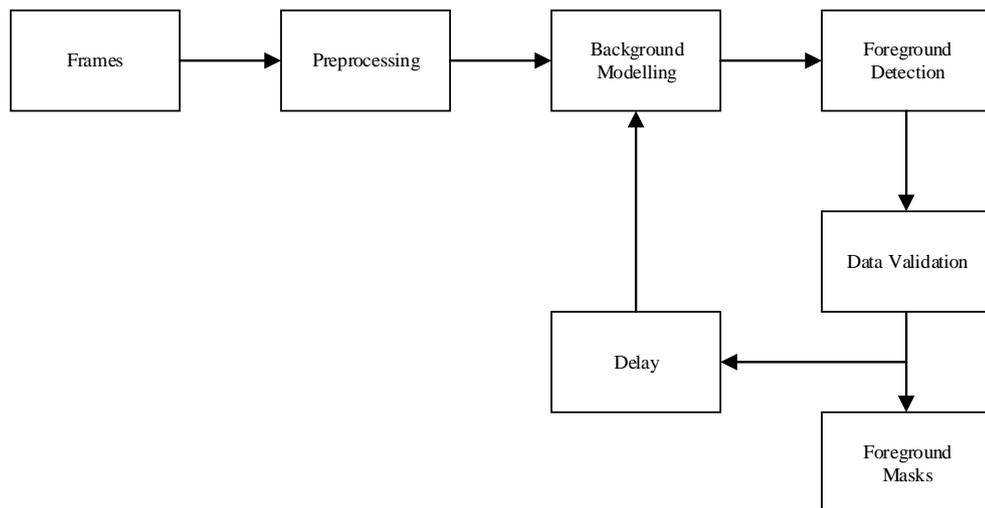
Jenis operasi ini bertujuan untuk membentuk ulang objek dari beberapa citra hasil proyeksi. Operasi rekonstruksi citra banyak digunakan dalam bidang medis. Misalnya beberapa *foto* rontgen dengan sinar X digunakan untuk membentuk ulang gambar organ tubuh.

2.6 Background subtraction

Background subtraction merupakan salah satu tugas penting yang pertama kali di kerjakan pada aplikasi *computer vision*. *Output* dari *background subtraction* biasanya adalah masukkan yang akan diproses pada tingkat yang lebih lanjut lagi seperti mentracking objek yang teridentifikasi. Kualitas *background subtraction* umumnya tergantung pada teknik pemodelan *background* yang digunakan untuk mengambil *background* dari suatu layar kamera. *Background subtraction* biasanya digunakan pada teknik segmentasi objek yang dikehendaki dari suatu layar, dan sering diaplikasikan untuk sistem pengawasan. (Puspita, 2011) Tahapan pada *background subtraction* ada 4:

1. Preprocessing,

2. Background modeling,
3. Foreground detection, dan
4. Data validation.

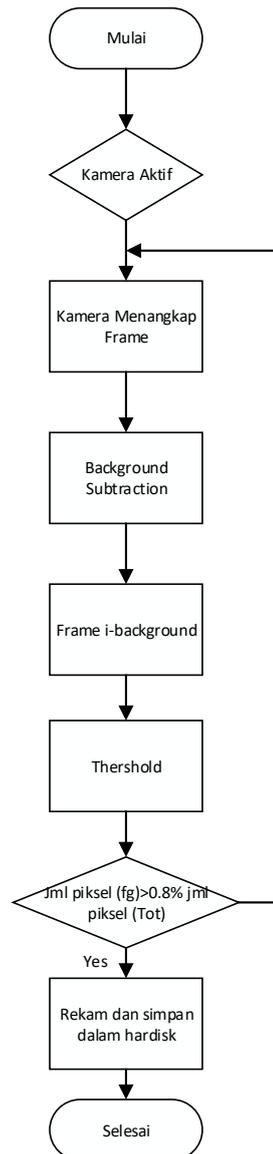


Gambar 2.2 Flow Diagram *Background Subtraction*

Flow diagram diatas merupakan alur metode *Background Subtraction*. Flow diagram tersebut menggambarkan alur ketika *frame* awal yang di ambil oleh *camera* yang dijadikan acuan sebagai tahapan awal proses yang dilakukan, setelah memiliki gambar acuan maka gambar akan diproses untuk memodelkan *background* yang ada pada *frame* yang telah diambil. Tahapan selanjutnya dilakukan *foreground detection* yakni mendeteksi objek yang ada pada *frame* yang telah diambil oleh *camera*. Selanjutnya data yang sudah dideteksi akan divalidasi apakah memang sudah benar ada gerakan atau tidak. Jika tidak ada gerakan maka *camera* akan mengambil *frame* saat ini juga untuk mendeteksi ulang *frame* selanjutnya. Dan jika terdeteksi objek maka akan ditandai objek tersebut pada tahapan *foreground masks*. Tahapan *foreground masks* merupakan tahapan yang memiliki gambar biner yang berisi *piksel* milik objek bergerak dalam suatu pemandangan atau *background*.

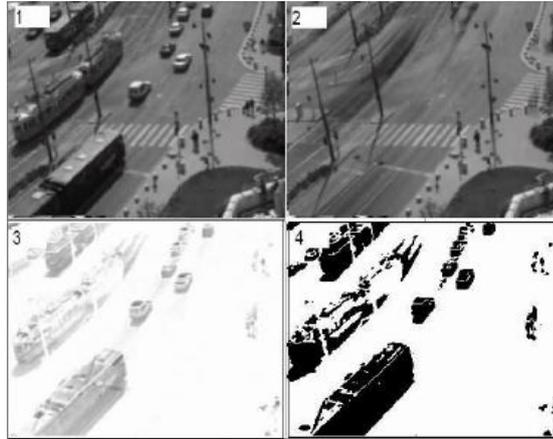
(Cheung dan Kamath, 2008) mendefinisikan *data validation* sebagai proses

perkembangan kandidat *foreground mask* berdasarkan informasi yang berisi dari sisi luar model *background*. Tahap *data validation* terkadang berdasarkan tahap proses *post* dari *foreground mask (pixel)*. Tujuan *data validation* untuk memperkecil angka kesalahan klasifikasi tanpa penurunan kecepatan klasifikasi yang cukup besar.



Gambar 2.3 Sistem Perancangan Metode *Background Subtraction*. (Irianto, K. D. & G. 2009)

Prosedur *background subtraction* terdiri dari 3 tahap, yaitu pre-processing, *background modeling*, dan *foreground detection*.



Gambar 2.4 *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.

$$R_{r,c} = I_{r,c} - B_{r,c}$$

(2.1)

Keterangan:

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.

2.7 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.

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 metode yang diinginkan.

2. Citra ini disegmentasikan ke dalam *piksel* objek dan *piksel* background seperti di bawah ini.

$$G1 = \{f(m,n) : f(m,n) > T\}$$

.....

(2.2)

$$G2 = \{f(m,n) : f(m,n) < T\}$$

.....

(2.3)

keterangan:

- G1 adalah nilai *piksel* objek.
- G2 adalah nilai *piksel background*.
- f(m,n) adalah nilai dari *piksel* yang terletak pada mth kolom dan nth baris.

1. Hitung nilai rata-rata *gray value* μ_1 dan μ_2 pada *piksel* dalam G1 dan G2.

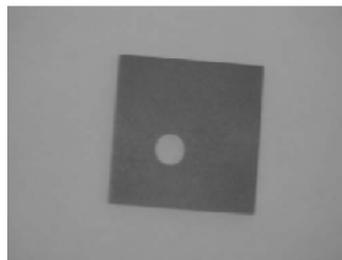
2. Hitung nilai *threshold* baru:

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

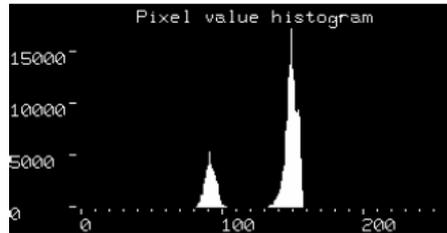
.....

(2.4)

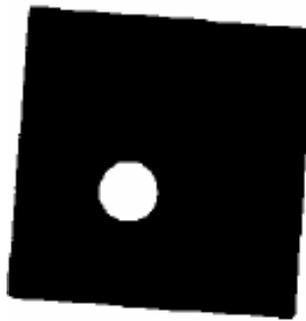
3. Ulangi langkah ke 2 sampai dengan langkah ke 4 dengan nilai T yang berbeda sampai nilai *thereshold* yang baru sama dengan nilai yang sebelumnya.



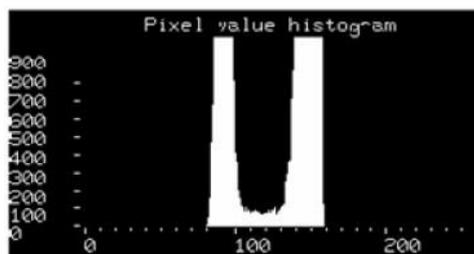
Gambar 2.5 Objek dengan warna gelap dan diletakkan pada *background* terang



Gambar 2.6 Histogram dari Gambar 2.5



Gambar 2.7 Objek yang telah *dithreshold* dengan nilai *threshold* sekitar 120



Gambar 2.8 Histogram dari gambar 2.7

2.8 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 *non-linear* dihitung dengan mengurutkan nilai intensitas sekelompok *pixel*, kemudian menggantikan nilai *pixel* yang diproses dengan nilai tertentu.

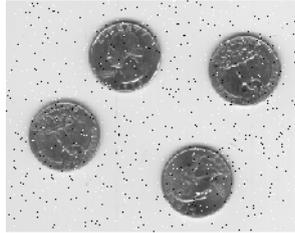
Pada median *filter* suatu *window* atau penapis yang memuat sejumlah *pixel* ganjil digeser titik per titik pada seluruh daerah citra. Nilai-nilai yang berada pada *window* diurutkan secara *ascending* untuk kemudian dihitung nilai mediannya. Nilai tersebut akan menggantikan nilai yang berada pada pusat 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 (2.5) sebagai berikut:

$$g(x,y) = \text{Median} \{f(x-i, y-j), (i,j) \in w\}$$

.....
(2.5)

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.



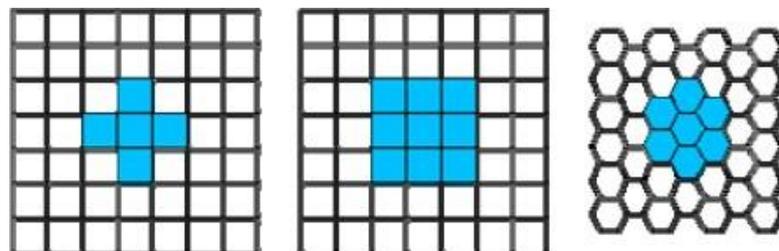
Gambar 2.9 Gambar awal sebelum difilter



Gambar 2.10 Gambar hasil median filter

2.9 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.



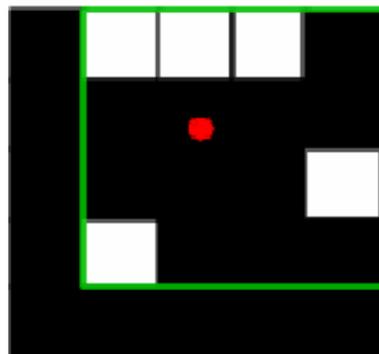
Gambar 2.11 4, 8 dan 6-connectivity

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.

2.10 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 objek direpresentasikan sebagai sebuah region dengan pendekatan bentuk persegi panjang. Gambar 2.11 menunjukkan sebuah region dari kumpulan *pixel* berwarna putih yang direpresentasikan dengan pendekatan bentuk persegi panjang.



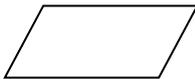
Gambar 2.12 Representasi region dengan pendekatan bentuk persegi panjang

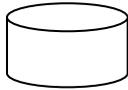
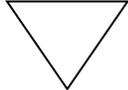
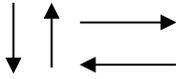
Pada penelitian ini, fitur dari pendekatan letak atau posisi objek menggunakan properti BoundingBox dan properti Centroid dari fungsi regionprops. Dimana Properti BoundingBox adalah persegi panjang yang memuat semua region. Sedangkan properti Centroid didefinisikan sebagai titik tengah atau pusat dari region. Elemen pertama dari Centroid adalah koordinat horizontal (“X” koordinat) dari objek dan elemen kedua dari Centroid adalah koordinat vertical (“Y” koordinat) dari objek.

2.11 Bagan Alir Program (*Flowchart*)

(Menurut Jogiyanto, 1999, p.46) bagan alir program (*flowchart*) adalah suatu bagan yang menjelaskan secara rinci langkah-langkah dari proses program. Simbol-simbol yang digunakan dalam bagan alir program dapat dilihat pada tabel 2.1.

Tabel 2.1 Simbol-Simbol Bagan Alir Program (*Flowchart*)

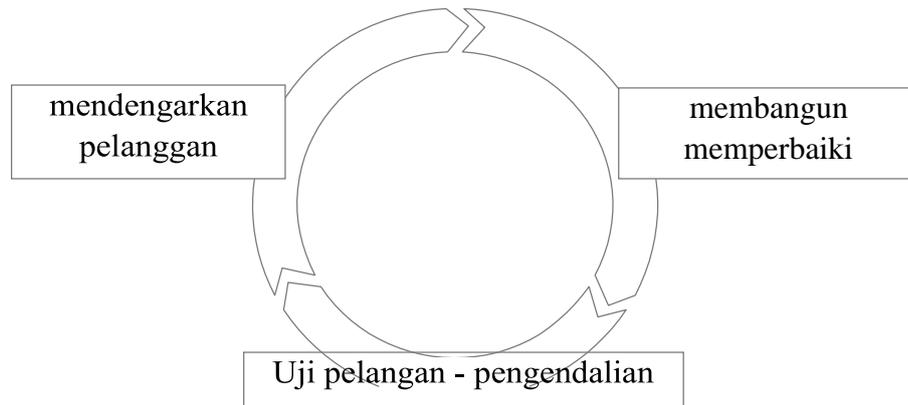
Simbol	Keterangan
	Terminator Menunjukkan awal dan akhir suatu proses.
	Input / Output Menunjukkan data masukan (<i>input</i>) atau data keluaran (<i>output</i>).
	Prosedur Menunjukkan proses data.
	Process Menunjukkan kegiatan proses dari operasi program
	Decision Menunjukkan penjelasan suatu proses <i>computer</i>
	Keyboard Memasukkan atau input data melalui keyboard

	<p>Harddisk</p> <p>Tempat penyimpanan data pada <i>computer</i></p>
	<p>Simpan Offline</p> <p>Penyimpanan arsip manual</p>
	<p>Connector</p> <p>Menunjukkan penghubung ke halaman yang sama</p>
	<p>Off-page Connector</p> <p>Menunjukkan penghubung ke halaman yang berbeda</p>
	<p>Garis Alir</p> <p>Menunjukkan simbol dari alur system atau proses</p>

2.12 Metode Pengembangan Perangkat Lunak

Pengembangan perangkat lunak menggunakan *prototyping model* dalam berbagai situasi dapat menawarkan pendekatan yang terbaik, hal ini didasari oleh pendapat (Pressman, 2002, p.39) yang menyatakan bahwa *prototyping model* merupakan metode yang efektif dalam merancang perangkat lunak. (Pressman, 2002, p.40) menjelaskan bahwa *prototyping paradigm* dimulai dengan mengumpulkan kebutuhan.

Pengembang bertemu dan pelanggan bertemu dan mendefinisikan objek keseluruhan dari perangkat lunak, mengidentifikasi segala kebutuhan yang diketahui dan kemudian melakukan “perancangan kilat”. Perancangan kilat berfokus pada penyajian dari aspek-aspek perangkat lunak tersebut yang akan nampak bagi pelanggan atau pemakai (contohnya pendekatan *input* dan *format output*). Perancangan kilat membawa kepada konstruksi sebuah *prototype*. *Prototype* tersebut dievaluasi oleh pelanggan atau pelanggan dan dipakai untuk menyaring kebutuhan pengembangan perangkat lunak.



Gambar 2.13 Prototype Paradigma

Prototype model juga dapat didefinisikan sebagai proses pengembangan pada *prototype* secara cepat untuk penggunaan pertama dan perbaikan terus menerus sehingga memperoleh sistem yang lengkap. *Prototype model* adalah proses yang dapat digunakan untuk membantu mengembangkan perangkat lunak dalam membentuk *prototype* dari perangkat lunak yang harus dibuat. Proses pada *prototyping model* dapat dijelaskan sebagai berikut :

1. Pengumpulan kebutuhan: *developer* dan klien bertemu, kemudian menentukan tujuan umum, kebutuhan yang diketahui dan seluruh gambaran bagian yang akan dibutuhkan selanjutnya. Detail kebutuhan mungkin tidak dapat dibicarakan disini, pada awal pengumpulan kebutuhan. Data yang digunakan dalam penelitian ini adalah data dari IIB Darmajaya.
2. Perancangan: perancangan dilakukan cepat dan rancangan mewakili semua aspek perangkat lunak yang diketahui, dan rancangan ini menjadi dasar pembuatan *prototype*.
3. Evaluasi *prototype*: klien mengevaluasi *prototype* yang dibuat dan digunakan untuk memperjelas kebutuhan perangkat lunak.

2.13 Video

Video adalah teknologi untuk menangkap, merekam, memproses, mentransmisikan dan menata ulang gambar bergerak. Biasanya menggunakan film seluloid, sinyal elektronik, atau media digital. Video juga bisa dikatakan sebagai gabungan gambar-gambar mati yang dibaca berurutan dalam suatu waktu dengan kecepatan tertentu. Gambar-gambar yang digabung tersebut dinamakan *frame* dan kecepatan pembacaan gambar disebut dengan *frame rate*, dengan satuan fps.

2.13.1 Jenis-Jenis Video:

Pada dasarnya terdapat dua jenis video dalam layer *computer*, yaitu: analog dan digital video.

1. Video Analog

Meskipun banyak video yang diproduksi hanya untuk platform display digital (untuk *Web*, CD-ROM, atau sebagai presentasi HDTV DVD), video analog (kebanyakan masih digunakan untuk penyiaran *televisi*) masih merupakan platform yang paling banyak diinstal untuk mengirim dan melihat video. Standar Penyiaran Video Analog: Tiga standar penyiaran video analog yang paling banyak digunakan di dunia adalah NTSC, PAL, dan SECAM.

A. NTSC

Amerika Serikat, Kanada, Meksiko, Jepang, dan banyak Negara lain menggunakan sistem penyiaran dan pemutaran video berdasarkan spesifikasi yang dibuat pada tahun 1952, National Television Standar Comitee. Standar ini mendefinisikan sebuah metode untuk mengkode informasi kedalam sinyal video terbuat dari 525 garis Horizontal yang di-scan dan digambar ke dalam wajah dalam tabung gambar berfosfor setiap 1/30 detik dengan electron yang bergerak cepat.

B. PAL

Sistem Phase Alternate Line (PAL) digunakan di Inggris, Eropa Barat, Australia, Afrika Selatan, Cina, dan Amerika Selatan. PAL meningkatkan resolusi layer menjadi 625 garis Horizontal, namun memperlambat kecepatan scan menjadi 25 *frame* per detik. Sama seperti saat penggunaan NTSC, garis genap dan ganjil digabungkan, setiap field memerlukan 1/50 detik untuk menggambar (50Hz).

C. SECAM

System Sequential Color and Memory (digunakan di Perancis, Eropa timur, USSR (sekarang Rusia), dan beberapa Negara lain. Meskipun SECAM merupakan system dengan 625 garis, 50 Hz, namun berbeda jauh dari *system* warna NTSC dan PAL dalam hal dasar teknologi dan metode penyiaran. Terkadang TV yang dijual di Eropa memanfaatkan dual komponen dan dapat menggunakan system PAL dan SECAM.

2. Video Digital

Integrasi Penuh dari video digital dalam *camera* dan *computer* mengurangi nemtuk televisi analog dari video dari produksi multimedia dan platform pengiriman, jika kamera video anda menggerakkan sinyal output digital, Anda dapat merekam video, Anda langsung ke disk, yang siap untuk diedit. Jika sebuah video klip disimpan sebagai data pada hard disk, CD-ROM atau perangkat penyimpanan massal lain, Dunia video kini telah mengalami perubahan dari analog ke digital. Pada konsumen rumahan dan perkantoran kita dapat menikmati kualitas video digital yang prima lewat hadirnya teknologi VCD dan DVD (*Digital Versatile Disc*), sedangkan dunia broadcasting kini juga lambat laun mengalihkan teknologinya kearah DTV (*Digital Television*).

1. Arsitektur Video Digital

Arsitektur Video Digital tersusun atas sebuah format untuk mengkode dan memainkan kembali *file* video dengan *computer* dan menyertakan sebuah *player* yang dapat mengenali dan membuka *file* yang dibuat untuk format tersebut. Arsitektur video digital yang utama adalah *AppleQuicktime*, *Microsoft Windows Media Format*, dan *Real Network RealMedia*. Format *file* video yang terkait adalah *QuickTime movie* (.mov), *Audio Video Interleaved* (.AVI), *Windows Media Video* (.wmv), dan *RealMedia* (.rm). Beberapa *player* mengenali dan memainkan lebih dari satu format *file* video. Video, seperti halnya *audio* juga mengalami proses yang serupa yaitu biasanya direkam dan dimainkan sebagai sinyal analog. Untuk itulah harus dikonversi menjadi digital terlebih dahulu agar dapat diproses menjadi sebuah multimedia title.

2.14 Pengertian Video (Citra Bergerak)

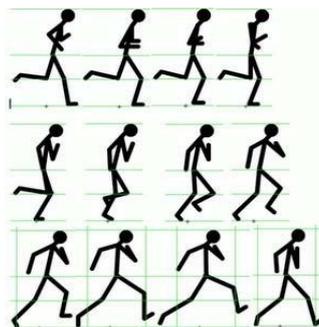
2.14.1 Animasi

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 objek-objek yang akan dianimasikan, sehingga objek 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_LILLA H.pdf)



Gambar 2.14 Gambar Animasi