BAB 2
LANDASAN TEORI

2.1 Intelligent Transportation Sistem

Intelligent Transportation Sistem (ITS) adalah kesatuan teknologi yang dibangun untuk mengatasi masalah-masalah yang terjadi pada transportasi [4]. Lebih jelasnya, *intelligent transportation sistem* adalah sistem yang mengintegrasikan transportasi dengan waktu yang akurat dan efisien dan di dalamnya diterapkan teknologi informasi, komputer, sensor, kontrol elektron, kontrol teori, sertai kecerdasan buatan [5]. Menurut Wewengkang, *Intelligent Transportation Sistem* umumnya membutuhkan teknologi semisal Automatic Visual Traffic Surveillance yang menggunakan kamera sebagai alat utama sebagai penerima sensor. Zhou di dalam penelitiannya yang berjudul, "*Bus Passenger Recognitionand Track of Video Sequence*", mengatakan bahwa pengenalan deteksi danpelacakan dari obyek-obyek yang bergerak menjadi salah satu fokus dalam penelitian dalam dunia komputer yang yang berhubungan dengan *computer vision*. Pada saat ini, penelitian mengenai pelacakan obyek menggunakan *computer vision* tersebut, umumnya digunakan pada *traffic surveillance*, pelayanan perbankan, *homing guidance* dan penerbangan militer. Dari sinilah, studi mengenai deteksi dan *tracking* pergerakan obyek menjadi hal yang realistik dan signifikan [6].

2.2 Pengertian Lampu Lalu Lintas

Lampu lalu lintas (*traffic lights*) menurut UU no. 22/2009 adalah lampu yang mengendalikan arus lalu lintas yang terpasang di persimpangan jalan, tempat penyeberangan pejalan kaki (*zebra cross*), dan tempat arus lalu lintas lainnya. Lampu ini yang menandakan kapan kendaraan harus berjalan dan berhenti secara bergantian dari berbagai arah. Saat ini pengaturan nyala lampu lalu lintas yang diterapkan adalah dengan penetapan durasi nyala lampu merah, kuning maupun

hijau. Pengaturan ini berlaku sepanjang hari secara otomatis. Contoh lampu lalu lintas dapat dilihat pada gambar 2.1.



Gambar 2.1 Lampu Lalu Lintas

2.3 Pelanggaran

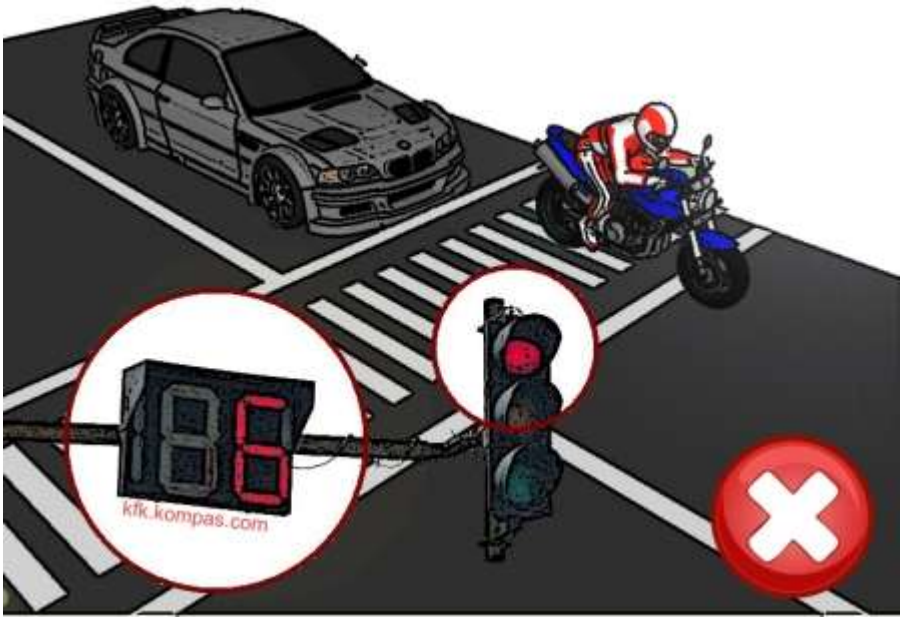
Pelanggaran adalah perbuatan pidana yang tergolong tidak seberat Kejahatan [7]. Pelanggaran adalah perbuatan yang oleh umum baru disadari sebagai tindak pidana, karena undang-undang menyebutnya sebagai delik, jadi karena ada undang-undang mengancam dengan pidana misalnya memparkir motor di sebelah kanan jalanan [8].

2.4 Pengemudi

Menurut Peraturan Pemerintah Republik Indonesia Nomor 44 Tahun 1993 tentang "Kendaraan dan Pengemudi", Pengemudi adalah orang yang mengemudikan kendaraan bermotor atau orang yang secara langsung mengawasi calon pengemudi yang sedang belajar mengemudikan kendaraan bermotor.

2.5 Pelanggaran Lampu Lalu Lintas

Setiap pengendara yang **melanggar rambu lalu lintas** dipidana dengan pidana kurungan paling lama 2 bulan atau denda paling banyak Rp 500 ribu (Pasal 287 ayat 1) [9]. Contoh pelanggaran menerobos pada saat lampu merah dapat dilihat pada gambar 2.2.



Gambar 2.2 Menerobos Saat Lampu Merah

Hal yang paling sering terjadi dan seringkali ditemukan adalah pengendara motor yang mencuri waktu di area lampu lalu lintas. Pengendara motor sering mencuri waktu dengan cara langsung berjalan dan buru-buru melaju padahal lampu hijau belum menyala. Pencurian waktu ini juga semakin terlihat jelas di area lampu lalu lintas yang dipasang *Countdown Digital Display*. Biasanya pencurian waktu terjadi 3-5 detik sebelum lampu hijau menyala, pengendara motor langsung melajukan kendaraannya [10]. Bahkan ada pula sebaliknya pengendara yang menerobos lampu lalu lintas padahal lampu merah sudah menyala, ini yang menyebabkan terjadinya kecelakaan lalu lintas.

2.6 Kendaraan Bermotor

Menurut Peraturan Pemerintah Republik Indonesia Nomor 44 Tahun 1993 tentang "Kendaraan dan Pengemudi", Kendaraan Bermotor adalah kendaraan yang digerakkan oleh peralatan teknik yang berada pada kendaraan itu. Umumnya kendaraan bermotor menggunakan mesin pembakaran dalam (perkakas atau alat untuk menggerakkan atau membuat sesuatu yg dijalankan dengan roda, digerakkan oleh tenaga manusia atau motor penggerak, menggunakan bahan bakar minyak atau

tenaga alam). Kendaraan bermotor memiliki roda, dan biasanya berjalan di atas jalanan.

Berdasarkan UU No. 14 tahun 1992 , yang dimaksud dengan peralatan teknik dapat berupa motor atau peralatan lainnya yang berfungsi untuk mengubah suatu sumber daya energi tertentu menjadi tenaga gerak kendaraan bermotor yang bersangkutan. Ada beberapa jenis kendaraan bermotor diantaranya mobil dan sepeda motor.

Mobil adalah kendaraan darat yang digerakkan oleh tenaga mesin, beroda empat atau lebih (selalu genap), dengan menggunakan bahan bakar minyak seperti bensin atau solar untuk menghidupkan mesinnya. Contoh mobil dapat dilihat pada gambar 2.3.



Gambar 2.3 Mobil

Sepeda Motor adalah kendaraan darat yang digerakkan oleh tenaga mesin, beroda dua, dengan menggunakan bahan bakar minyak seperti bensin untuk menghidupkan mesinnya. Contoh sepeda motor dapat dilihat pada gambar 2.4



Gambar 2.4 Sepeda Motor

2.7 Kecerdasan Buatan (*Artificial Intelligent*)

Sistem pakar atau expert sistem merupakan suatu cabang dari ilmu komputer yang berdasarkan kecerdasan buatan (*artificial intelligence*). *Artificial intelligence* adalah sebuah rancangan program yang memungkinkan komputer melakukan suatu tugas atau mengambil keputusan dengan meniru suatu cara berpikir dan penalaran manusia.

Adapun beberapa definisi *artificial intelligence* adalah sebagai berikut :

a. Definisi artificial intelligence menurut Jog [11] adalah : "*Artificial Intelligence* didefinisikan sebagai suatu mesin atau alat pintar (biasanya adalah komputer) yang dapat melakukan suatu tugas yang bilamana tugas tersebut dilakukan oleh manusia akan dibutuhkan suatu kepintaran untuk melakukannya".
b. Definisi artificial intelligence menurut Kus [12] adalah : "Kecerdasan buatan atau *artificial intelligence* merupakan salah satu bagian ilmu komputer yang membuat agar mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia".
c. Definisi *artificial intelligence* menurut Sar [13] adalah : "Kecerdasan buatan adalah salah satu bidang ilmu komputer yang mendayagunakan komputer sehingga dapat berperilaku cerdas seperti manusia".

Dengan perancangan *artificial intelligence* yang baik, maka diharapkam peran manusia dapat diminimalkan dan dapat meringankan beban kerja manusia.

Cara kerja artificial intelligence adalah menerima input, kemudian diproses dan kemudian mengeluarkan output berupa suatu keputusan.

Implementasi dari artificial intelligence saat ini umum ditemui dalam bidang-bidang seperti berikut :

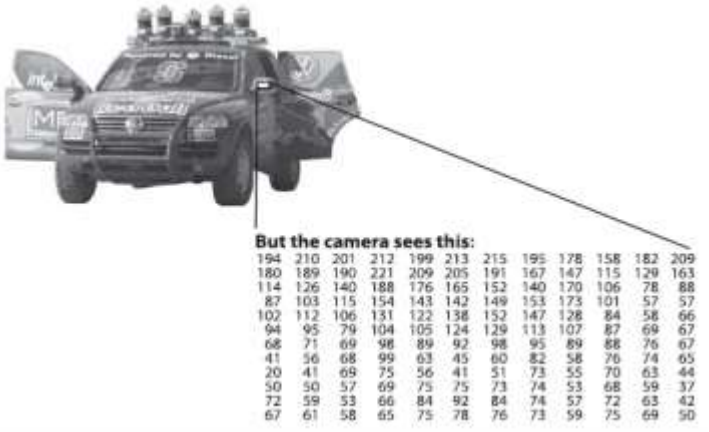
a. *Computer vision* yaitu suatu metode *artificial intelligence* yang memungkinkan sebuah sistem komputer mengenali gambar sebagai inputnya. Contohnya adalah mengenali dan membaca tulisan yang ada didalam gambar.
b. *Fuzzy Logic* yaitu suatu metode artificial intelligence yang banyak terdapat pada alat-alat elektronik dan robotika. Dimana alat-alat elektronik atau robotika tersebut mampu berpikir dan bertingkah laku sebagaimana layaknya manusia.
c. Game yaitu suatu metode artificial intelligence yang berguna untuk meniru cara berpikir seorang manusia dalam bermain game. Contohnya adalah program Perfect Chessmate yang mampu berpikir setara dengan seorang grandmaster catur.

2.8 *Computer vision*

Computer vision adalah suatu proses transformasi atau perubahan dari data yang berasal dari kamera video maupun foto/gambar kedalam sebuah hasil keputusan ataupun sebuah presentasi yang baru, dimana hasil dari kegiatan transformasi tersebut memiliki kepentingan untuk mencapai suatu tujuan. Data yang dimasukan kedalam kegiatan transformasi tersebut memungkinkan untuk memiliki beberapa informasi yang terkonstektual seperti halnya sebuah foto/gambar yang didalamnya terdapat berbagai objek. Dengan demikian akan didapatkan keputusan-keputusan yang akan diambil pada gambar, misalnya berupa “adakah wajah seseorang pada gambar tersebut?” atau “siapa sajakah orang yang terdapat pada foto tersebut?”. Adapun perubahan kedalam presentasi yang baru seperti perubahan gambar menjadi *grayscale* atau juga pemotongan objek pada gambar.

Berbeda halnya dengan manusia yang memiliki pemikiran untuk meneliti, memahami, dan membandingkan informasi pada objek secara langsung dengan

informasi dari pengalaman-pengalaman yang didapatkan selama bertahun-tahun hidup di dunia. Dalam sistem penglihatan mesin (*machine vision*) komputer hanya mampu mendapatkan informasi berupa kumpulan angka-angka dari media input data seperti kamera atau disket. Berikut contoh gambar 2.5 untuk mengilustrasikannya.



Gambar 2.5 Ilustrasi penglihatan komputer

Dari gambar 2.5 terlihat bahwa kaca samping pada gambar mobil hanyalah berupa kumpulan angka-angka jika dilihat oleh computer. Tentunya angka-angka ini masih memiliki banyak noise atau gangguan sehingga informasi yang terkandung sangatlah sedikit. Hal ini disebabkan dari penglihatan komputer yang terbatas hanya 2 dimensi (2D) namun objek yang diproses ialah objek 3 dimensi (3D) dunia nyata, sehingga informasi yang didapat mengalami banyak sekali gangguan dan kendala dari berbagai fenomena di kehidupan nyata (cuaca, cahaya, bayangan, refleksi cahaya, dan gerakan). Dengan demikian hal tersebut menjadi permasalahan yang terus-menerus diteliti untuk ditemukan cara atau tehnik penyelesaiannya.

2.9 Citra Digital

Image atau citra adalah gambar pada bidang dua dimensi. Dari sudut pandang matematis, citra adalah fungsi kontinyu dari intensitas radiasi pada bidang dua dimensi. Ketika sumber radiasi mengeluarkan radiasi dan radiasi tersebut mengenai obyek, obyek memantulkan kembali sebagian dari radiasi tersebut,

pantulan dari radiasi inilah yang diterima oleh sensor berupa alat optik, misalnya mata dan kamera.

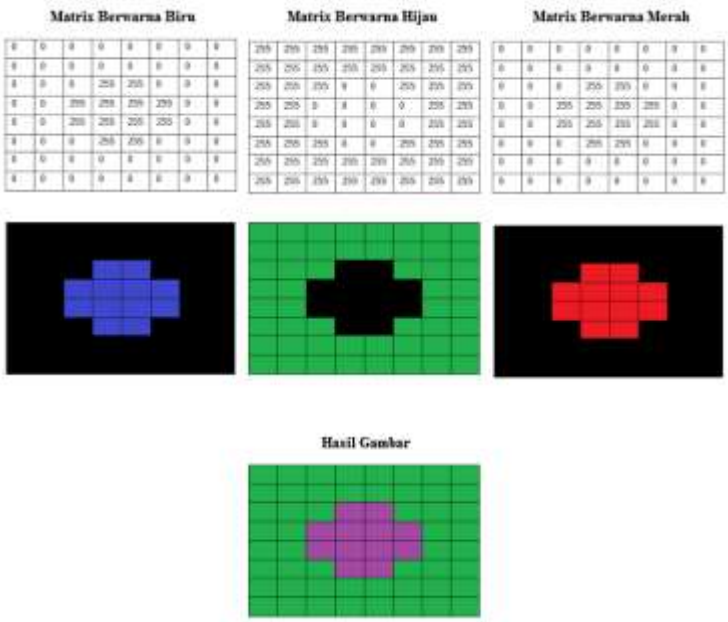
Citra dapat berupa foto hitam-putih atau berwarna, sinyal-sinyal video seperti gambar pada televisi, atau bersifat digital yang berarti dapat langsung disimpan pada pita magnetik. Citra dihasilkan dari gambar analog dua dimensi yang kontinyu menjadi gambar diskrit melalui proses pembuatan kisi-kisi ke arah vertikal dan horizontal, atau dikenal dengan istilah digitisasi atau sampling. Setiap kisi yang menjadi elemen baris vertikal dan horizontal pada citra disebut dengan piksel.

Hasil keluaran dari citra bisa berupa 2 jenis yakni:

1. Citra statis yang artinya citra tunggal yang tidak bergerak.
2. Citra bergerak yang merupakan rangkaian citra diam yang ditampilkan secara beruntun (sekuensial) sehingga memberikan kesan pada mata sebagai gambar yang bergerak, contohnya adalah video.

2.10 Citra RGB (Red, Green, Blue)

OpenCV biasanya menangkap gambar dan video dengan ukuran 8 - bit , bertipe integer dengan format BGR. Dengan kata lain, gambar yang diambil dapat dianggap sebagai 3 matriks yaitu BIRU, HIJAU dan MERAH (BGR) dengan nilai berkisar antara 0 sampai 255. Berikut contoh gambar yang menunjukkan bagaimana gambar warna direpresentasikan dengan menggunakan 3 matriks, dapat dilihat pada gambar 2.6:



Gambar 2.6 Gambar BGR dibentuk

Pada gambar di atas, masing-masing kotak kecil mewakili pixel gambar. Dalam gambar sebenarnya, piksel ini sangat kecil yang mata manusia tidak bisa membedakan.

2.11 *Citra Grayscale*

Grayscale merupakan range dengan warna yang abu-abu yang hanya memiliki satu nilai kanal pada setiap pixelnya, dengan kata lain nilai bagian *RED = GREEN = BLUE*. Nilai tersebut digunakan untuk menunjukan tingkat intensitas. Warna yang dimiliki adalah warna dari hitam, keabuan, dan putih. Tingkatan keabuan disini merupakan warna abu dengan berbagai dari hitam hingga mendekati putih. Banyaknya warna tergantung pada jumlah bit yang disediakan di memori untuk menampung kebutuhan warna. Semakin besar jumlah bit warna yang disediakan di memori, maka semakin halus gradasi warna yang terbentuk [14].

Sebagai contoh, skala keabuan 4 bit. Jumlah kemungkinan 2^4 = l6 warna. Jadi, kemungkinan warna 0 (minimal/hitam) sampai l5 (rnaksimal/putih). Untuk penelitian ini menggunakan *grayscale* dengan kedalaman warna 8 bit (256 kombinasi warna keabuan). Rentang nilai warna – warna pada RGB untuk Red

(merah) rentangnya (255,0,0), Green (hijau) rentangnya (0,255,0), Blue (biru) rentangnya (0,0,255), dan Mediun Gray (abu) rentangnya (127,127,127) Untuk mendapatkan nilai keabuan digunakan persamaan sebagai berikut:

$$Grayscale = 0.299 \text{ x } R + 0.587 \text{ x } G + 0.114 \text{ x } B \quad (2.1)$$

Keterangan:
R = nilai piksel warna merah
G = nilai piksel warna hijau
B = nilai piksel warna biru

Grayscaling merupakan teknik mengubah citra warna menjadi citra abu-abu. Teknik ini merupakan proses awal dalam pengolahan citra. Dalam pemrograman MATLAB teknik *grayscaling* dapat digunakan dengan menggunakan fungsi *rgb2gray* sehingga yang awalnya citra warna dikonversi menjadi citra yang terdiri dari hitam dan putih atau biasa disebut citra *grayscale*. Teknik ini merupukan proses awal dalam pengolahan citra agar dapat melanjutkan pada proses berikutnya. Berikut adalah teknik grayscaling dapat dilihat pada gambar 2.7: [14]



Gambar 2.7 Teknik Grayscaling

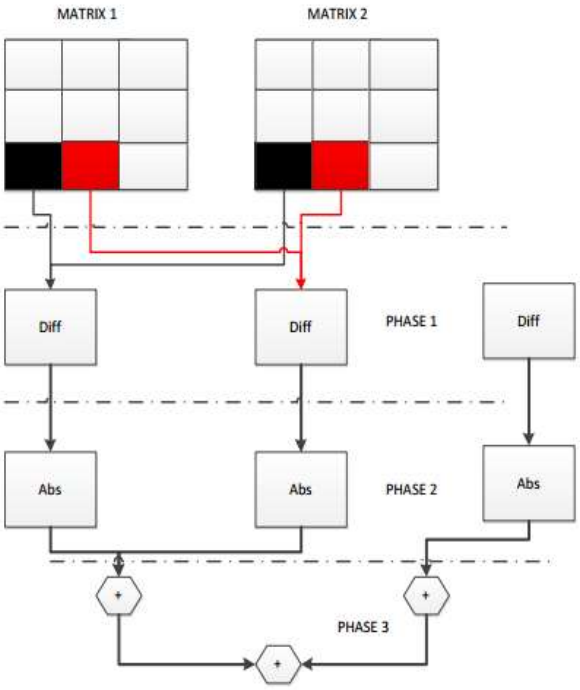
2.12 Pendeteksian Gerak

Pendeteksian gerak digunakan untuk melihat atau mendeteksi adanya gerakan dari suatu objek pada daerah tangkapan kamera. Beberapa metode deteksi gerak (*motion detection*), diantaranya :

1. *Background substraction*, metode yang terdapat proses membandingkan setiap warna pada setiap piksel.
2. *Phase Only Correlation*, metode yang digunakan untuk membandingkan antara citra referensi dengan citra saat ini.
3. Metode *Sum of absolute dfference*, merupakan salah satu cara untuk pencocokan blok gambar antara gambar acuan dengan gambar aktual. Dari pencocokan blok gambar tersebut akan dicari perbedaan antara kedua gambar.

2.13 Sum of Absolute Diference

Sum of absolute difference (SAD) adalah salah satu metode yang digunakan untuk mencari pencocokan blok gambar. Pada metode pencocokan blok, suatu blok citra yang berpusat pada satu titik di tengahnya di *frame* sekarang dibandingkan dengan blok kandidat di *frame* referensi berdasarkan perbedaan atau persaaan tertentu untuk menemukan blok cocok yang terbaik pada area yang dicari. Berikut adalah ilustrasi menghitung pixel antar *frame*, dapat dilihat pada gambar 2.8:



Gambar 2.8 Ilustrasi menghitung pixel antar *frame*

Berikut penjelasan dari gambar 2.9:

1. Bandingkan selisih *pixel* 1 di matrix 1 dan *pixel* 1 di matrix 2, *pixel* 2 pada matrix 1 dan *pixel* 2 pada matrix 2 dan begitu selanjutnya. Matrix disini dianalogikan sebagai *frame*.
2. *Absolute*-kan nilai hasil selisih *pixel* antar *frame*.
3. Jumlahkan seluruh selisih *pixel* tersebut sehingga mendapatkan nilai SAD nya.

Frame dinyatakan berbasis blok maka gerak diperoleh menggunakan metode blok matching dengan nilai SAD yang dinyatakan dengan persamaan berikut:

$$\text{SAD} = \sum_{k=1}^{n} \sum_{l=1}^{n} | B_{i,j}(k,l) - B_{i-u,j-v}(k,l) | \quad (2.2)$$

Bi,j(k,l) adalah pixel ke (k,l) dari macroblok n pixel *frame* sekarang dan Bi-u, j-v (k,l) menyatakan pixel ke (k,l) pada macroblok kandidat dari *frame* referensi di lokasi spatial (i,j) di pindahkan dengan vector (u,v).

2.14 Citra *Threshold*

Threshold merupakan tahapan untuk membuat warna citra menjadi hanya berwarna hitam atau putih. Untuk melakukan *threshold*ing terlebih dahulu harus dicari suatu nilai yang digunakan untuk batas toleransi suatu warna keabuan dapat dijadikan warna hitam atau warna putih, nilai tersebut biasa disebut nilai ambang. Setelah semua piksel diubah menjadi nilai keabuan, pencarian nilai ambangpun dilakukan. Ada beberapa cara menentukan nilai ambang yaitu pertama, dengan menentukan langsung berdasarkan nilai tengah dari level keabuan yaitu 128. Kedua, menentukan suatu nilai tertentu sebagai nilai ambang. Ketiga, menentukan nilai rata – rata dari total piksel keabuan dalam citra dan menjadikannya sebagai nilai ambang rata – rata. Kemudian setelah didapat nilai ambang tersebut maka akan dilakukan perbandingan setiap nilai piksel dari citra keabuan tersebut dengan nilai ambangnya. Jika nilai piksel lebih besar dari nilai ambangnya maka nilai piksel tersebut akan diubah menjadi 0 (piksel berwarna hitam). Sedangkan jika yang ditemukan sebaliknya yaitu nilai pikselnya lebih kecil dari nilai ambang yang telah ditentukan maka nilai piksel tersebut diubah menjadi 1 (piksel berwarna putih) [15].

Adapun rumus untuk menentukan nilai threshold bisa didapatkan dari persamaan 2.3:

```
g(x,y)={1 if f(x,y)>=T}
g(x,y)={0 if f(x,y)<=T}                                    (2.3)
```

Sebagai contoh dari citra biner (*threshold*), dapat dilihat pada gambar 2.9.

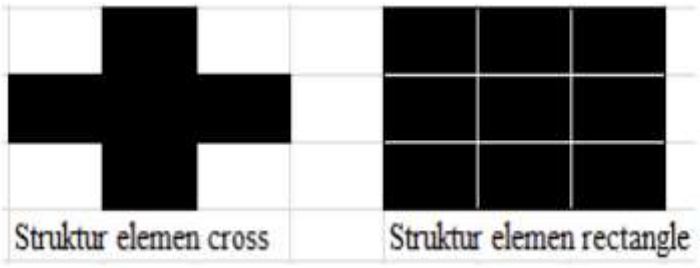


Gambar 2.9 Citra *Grayscale* ke Citra *Threshold*

2.15 Operasi Morphologi

Morphologi adalah teknik pengolahan citra digital dengan menggunakan bentuk (shape) sebagai pedoman dalam pengolahan. Nilai dari setiap pixel dalam citra digital hasil diperoleh melalui proses perbandingan antara pixel yang bersesuaian pada citra digital masukan dengan pixel tetangganya. Operasi morphologi bergantung pada urutan kemunculan dari pixel, tidak memperhatikan nilai numeric dari pixel sehingga teknik morphologi sesuai apabila digunakan untuk melakukan pengolahan binary image dan *grayscale* image. Dengan mengatur atau memilih ukuran dan bentuk dari matrik kernel (structuring element) yang digunakan, maka dapat mengatur sensitivitas operasi morphologi terhadap bentuk tertentu (spesifik) pada citra digital masukan. Operasi morphologi standar yang dilakukan adalah proses erosi dan dilasi. Dilasi adalah proses penambahan pixel pada batas dari suatu objek pada citra digital masukan, sedangkan erosi adalah proses pemindahan/pengurangan pixel pada batas dari suatu objek. Jumlah pixel

yang ditambahkan atau yang dihilangkan dari batas objek pada citra digital masukan tergantung pada ukuran dan bentuk dari structuring element yang digunakan. 19 Berikut adalah struktur element operasi yang dapat dipakai untuk proses operasi morphologi, dapat dilihat pada gambar 2.10: [16]



Gambar 2.10 Struktur Elemen

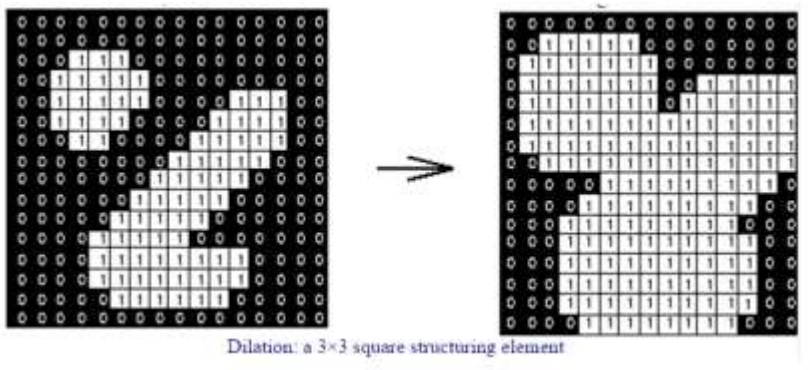
2.15.1 Dilasi

Dilasi adalah operasi morphologi yang akan menambahkan pixel pada batas antar objek dalam suatu citra digital. Atau secara rinci Dilasi merupakan suatu proses menambahkan piksel pada batasan dari objek dalam suatu image sehingga nantinya apabila dilakukan operasi ini maka image hasilnya lebih besar ukurannya dibandingkan dengan image aslinya. Dalam operasi dilasi, citra asli akan mengalami pelebaran dengan mengikuti bentuk *Structuring Element* yang digunakan. Pada gambar 2.10 dan gambar 2.11 merupakan hasil image setelah dilakukan proses dilasi dari original image dengan stucturing element yang telah ditentukan [17]:



Gambar 2.11 (a) Gambar Biner (b) Gambar Hasil Dilasi

Berikut gambar 2.12 menjelaskan tentang proses dilasi cara membuka letak sktruktur elemen terbuka yang dipisahkan oleh wilayah tertutup.



Gambar 2.12 Proses Dilasi

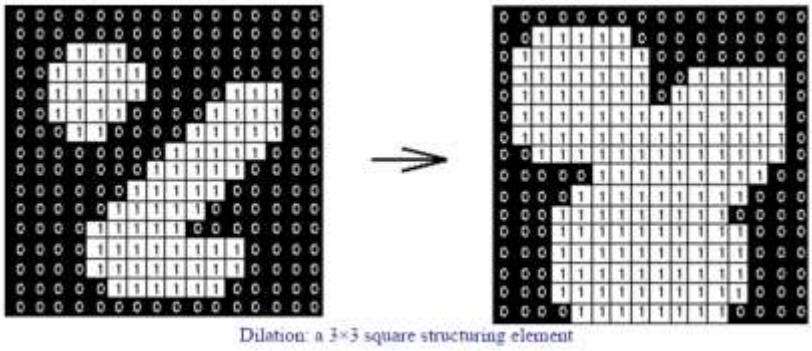
2.15.2 Erosi

Erosi merupakan kebalikkan dari Dilasi. Proses ini akan membuat ukuran sebuah citra menjadi lebih kecil. Berbeda dengan dilatasi, apabila erosi dilakukan maka yang dikerjakan adalah memindahkan piksel pada batasan-batasan objek yang akan di erosi. Jumlah dari piksel yang ditambah atau dihilangkan bergantung pada ukuran dan bentuk dari structuring element yang digunakan untuk memproses image tersebut. Berikut adalah proses hasil erosi, dapat dilihat pada gambar 2.13 [17]:



Gambar 2.13 (a) Gambar *Grayscale* (b) Gambar *Threshold* (c) Hasil Erosi

Berikut adalah gambar 2.14 tentang proses erosi memilih bagian yang tidak perlukan pada struktur elemen:



Gambar 2.14 Proses Erosi

2.15.3 Opening

Opening umumnya digunakan untuk menghaluskan kontur dari sebuah objek, memutuskan garis tipis yang menghubungkan dua *region* besar, dan menghilangkan tonjolan tipis. Operasi *opening* dari himpunan A oleh *Structuring Element* B, yang dinotasikan dengan A ∘ B, didefinisikan pada persamaan berikut [17]:

$$(A \ominus B) \oplus B \tag{2.4}$$

Keterangan:

A = Matriks Awal

B = Struktur Element

$\ominus$ = Operasi “dan”

$\oplus$ = Operasi “atau”

2.15.4 Closing

Closing digunakan untuk menghaluskan bagian dari kontur tetapi berbeda dengan *opening*, *closing* biasanya menyatukan bagian kecil yang terputus dan menyatukan cekungan yang panjang dan tipis. Operasi *closing* dari himpunan A oleh *Structuring Element* B, yang dinotasikan dengan $A \cdot B$, didefinisikan pada persamaan berikut [17]:

$$(A \oplus B) \ominus B \tag{2.5}$$

Keterangan:

A = Matriks Awal

B = Struktur Element

$\oplus$ = Operasi “dan”

$\ominus$ = Operasi “atau”

2.16 *Blob tracking*

Blob tracking adalah proses memisahkan tiap-tiap gumpalan yang terdapat dalam suatu gambar dan kemudian melabelinya sehingga antara gumpalan yang satu dengan gumpalan yang lainnya dapat dibedakan dengan identitas yang berbeda.

2.17 Pemograman Berorientasi Objek

Analisis dan desain berorientasi objek adalah cara baru dalam memikirkan suatu masalah dengan menggunakan model yang dibuat menurut konsep sekitar dunia nyata. Dasar pembuatan adalah objek, yang merupakan kombinasi antara struktur data dab perilaku dalam suatu entitas. Model berorientasi objek bermanfaat untuk memahami masalah, komunikasi dengan ahli aplikasi, pemodelan suatu organisasi, meyiapkan dokumentasi serta perancangan program dan basis data. Pertama-tama suatu model analisis dibuat untuk menggambarkan aspek dasar dari domain aplikasi, dimana model tersebut berisi objek yang terdapat dalam domain aplikasi termasuk deskripsi dari keterangan objek dan perilakunya. Secara spesifik, pengertian berorientai objek berarti bahwa mengorganisasi perangkat lunak sebagai kumpulan dari objek tertentu yang memiliki struktur data dan perilakunya. Hal ini yang membedakan dengan pemograman konvensional dimana struktur data dan perilaku hanya berhubungan secara terpisah. Terdapat beberapa cara untuk menentukan karateristik dalam pendekatan berorientasi objek, tetapi secara umum mencakup empat hal, yaitu identifikasi, klasifikasi, *polymorphism* (polimorfisme)

dan *inheritance* (pewarisan). Metodologi pengembangan sistem berorientasi objek mempunyai tiga karateristik utama, yaitu [18]:

a. *Encapsulation*

Encapsulation (pengkapsulan) merupakan dasar untuk pembatasan ruang lingkup program terhadap data yang diproses. Data dan prosedur atau fungsi dikemas dalam bersama-sama dalam suatu objek, sehingga prosedur atau fungsi lain dari luar tidak dapat mengaksesnya. Data terlindung dari prosedur atau objek lain kecuali prosedur yang berada dalam objek itu sendiri.

b. *Inheritance*

Inheritance (pewarisan) adalah teknik yang menyatakan bahwa anak dari objek akan mewarisi atribut dan metoda dari induknya langsung. Atribut dan metoda dari objek induk diturunkan kepada anak objek, demikian seterusnya. Pendefinisian objek dipergunakan untuk membangun suatu hirarki dari objek turunannya, sehingga tidak perlu membuat atribut dan metoda lagi pada anaknya, karena telah mewarisi sifat induknya.

c. Polymorphism

Polymorphism (polimorfisme) yaitu konsep yang menyatakan bahwa sesuatu yang sama dapat mempunyai bentuk dan perilaku berbeda. Polimorfisme mempunyai arti bahwa operasi yang sama mungkin mempunyai perbedaan dalam kelas yang berbeda.

2.18 *Unified Modelling Language* (UML)

Unified Modelling Language (UML) adalah sebuah "bahasa" yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. Dengan menggunakan UML dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan *class* dan operation dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti

lunak dalam bahasa-bahasa berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C. Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan *syntax*/semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai Diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML syntax mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari tiga notasi yang telah ada sebelumnya: Grady Booch OOD (*Object-Oriented Design*), Jim Rumbaugh OMT (*Object Modeling Technique*), dan Ivar Jacobson OOSE (*Object Oriented Software Engineering*) [18].

2.19 Diagram UML

Menggunakan berbagai macam Diagram dengan fungsi masing-masing untuk menggambarkan setiap proses dari sistem berorientasi objek. Berikut meruapakan beberapa Diagram UML diantaranya [18]:

d. *Use case* Diagram

Use case atau Diagram *use case* merupakan pemodelan yang digunakan untuk menggambarkan kelakuan (*behavior*) dari sistem yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *usecase*.

1. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem yang akan dibuat diluar sistem yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
2. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit yang saling bertukar pesan antar unit atau aktor.

e. *Activity* Diagram

Diagram aktivitas atau *Activity* Diagram adalah sebuah Diagram yang menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Dalam Diagram aktivitas yang perlu diperhatikan adalah bahwa Diagram aktivitas menggambarkan aktivitas sistem, bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut:

1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
2. Urutan atau pengelompokkan tampilan dari sistem/*user interface* dimana setiap aktivitas dianggap memiliki seuah rancangan antarmuka tampilan.
3. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.
4. Rancangan menu yang ditampilkan pada perangkat lunak.

f. *Class* Diagram

Diagram kelas atau *class* Diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

1. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas.
2. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

g. *Sequence* Diagram

Diagram sekuen adalah Diagram yang menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan Diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Banyaknya Diagram sekuen yang harus digambarkan adalah sebanyak pendefinisian *use case* yang memiliki proses itu sendiri atau yang penting semua

use case yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada Diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka Diagram sekuen yang harus dibuat juga semakin banyak.

2.20 Visual Studio 2013

Visual Studio 2013 adalah alat untuk para pengembang guna memberi kemudahan dalam merancang aplikasi modern agar dapat terhubung dengan semua *platform* Microsoft; Mulai dari Windows Azure dan Windows Server 2012 hingga Windows 8.1 dan Office 365. aplikasi Visual Studio 2013 telah dilengkapi dengan beragam alat yang memudahkan penggunanya membangun sebuah aplikasi yang lebih menarik. Untuk pengembang individu, Visual Studio 2013 telah dilengkapi dengan fitur yang membantu produktivitas penggunanya, seperti fitur Peek Definition dan Code Lens bisa menyajikan beragam informasi penting tentang lokasi *source code* yang sedang Anda olah dalam editor Visual Studio. Fitur lainnya adalah Browser Link, Office 365 Cloud Business Apps, Performance and Diagnostics Hub, Agile Portfolio Management, dan tentunya Editor XAML yang telah ditingkatkan. Microsoft juga mengembangkan Visual Studio Online yang merupakan layanan bagi *developer* dan menjadi bagian dari Microsoft Cloud OS. Alat ini akan memberi pengalaman baru bagi *developer*guna membuat aplikasi yang lebih mudah, cepat, dan tentunya bebas masalah. Saat ini, Microsoft telah memasukkan Visual Studio Online ke dalam bagian MSDN [19].

2.21 Pemrograman C++

Pemrograman **C++** adalah bahasa pemrograman komputer yang di buat oleh Bjarne Stroustrup, yang merupakan perkembangan dari bahasa C dikembangkan di Bong Labs (Dennis Ritchie) pada awal tahun 1970-an, Bahasa itu diturunkan dari bahasa sebelumnya, yaitu B, Pada awalnya, bahasa tersebut dirancang sebagai bahasa pemrograman yang dijalankan pada sistem Unix, Pada perkembangannya, versi ANSI (American National Standart Institute) Bahasa pemrograman C menjadi versi dominan, Meskipun versi tersebut sekarang jarang dipakai dalam pengembangan sistem dan jaringan maupun untuk sistem embedded,

Bjarne Stroustrup pada Bel labs pertama kali mengembangkan C++ pada awal 1980-an. Untuk mendukung fitur-fitur pada C++, dibangun efisiensi dan sistem support untuk pemrograman tingkat rendah (low level coding). Pada C++ ditambahkan konsep-konsep baru seperti *class* dengan sifat-sifatnya seperti inheritance dan over*load*ing. Salah satu perbedaan yang paling mendasar dengan bahasa C adalah dukungan terhadap konsep pemrograman berorientasi objek (Object Oriented Programming) [20].

2.22 OpenCV

OpenCV (*Open Source Computer vision Library*) adalah sebuah pustaka perangkat lunak yang ditujukan untuk pengolahan citra dinamis secara real-time, yang dibuat oleh Intel, dan sekarang didukung oleh Willow Garage dan Itseez. Program ini bebas dan berada dalam naungan sumber terbuka dari lisensi BSD. 32 Pustaka ini merupakan pustaka lintas platform. Program ini didedikasikan sebagaian besar untuk pengolahan citra secara real-time. Jika pustaka ini menemukan pustaka Integrated Performance Primitives dari intel dalam sistem komputer, maka program ini akan menggunakan rutin ini untuk mempercepat proses kerja program ini secara otomatis. [21]

Berikut ini adalah fitur-fitur pada *library* OpenCV:

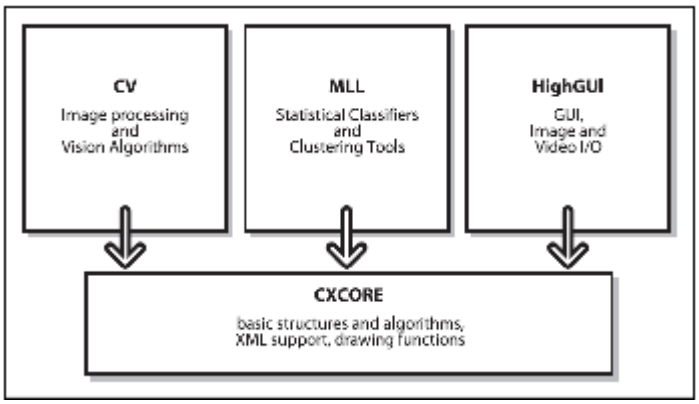
1. Manipulasi data gambar (alokasi memori, melepaskan memori, kopi gambar, setting serta konversi gambar)
2. *Image*/*Video* I/O (Bisa menggunakan camera yang sudah didukung oleh *library* ini)
3. Manipulasi matrix dan vektor serta terdapat juga routines linear algebra (*products*, *solvers*, *eigenvalues*, SVD)
4. *Image processing* dasar (filtering, *edge detection*, pendeteksian tepi, sampling dan interpolasi, konversi warna, operasi morfologi, histograms, *image pyramids*)
5. Analisis struktural
6. Kalibrasi kamera
7. Pendeteksian grerak

8. Pengenalan objek
9. Basic GUI (Dis*play* gambar/*video*, *mouse/keyboard* kontrol, *scrollbar*) *Image Labelling* (*line, conic, polygon, text drawing*)

OpenCV sendiri terdiri dari 5 *library*, yaitu :

1. CV : untuk algoritma Image processing dan Vision.
2. ML : untuk machine learning *library.*
3. Highgui : untuk GUI, Image dan Video I/O.
4. CXCORE : untuk struktur data, support XML dan fungsi-fungsi grafis.
5. CvAux

Berikut adalah struktur dan Konten OpenCV, dapat dilihat pada gambar 2.15.



Gambar 2.15 Struktur dan Konten OpenCV

OpenCV yang digunakan di dalam sistem yang akan dibangun yaitu OpenCV v.2.4.10 yaitu yang berisikan tentang *method image processing*, yang didalamnya mempunyai banyak *method*. Namun yang akan dipakai dan dipanggil dalam sistem antara lain: *grayscale, threshold* dan operasi morphologi. Fungsi pemanggilan *grayscale* pada opencv adalah *cvtColor* dengan bentuk fungsi transformasi dari RGB ke bentuk gray dengan `BgrToGray`, adapun parameter-parameter didalamnya seperti pada Tabel 2.1:

Tabel 2.1 Parameter cvtColor OpenCV

Nama Parameter	Deskripsi
Src	Source gambar asli hasil inputan berisi nilai piksel RGB 8 bit, 16 bit atau nilai single-presisi
Dst	Destinasi keluaran gambar dengan ketentuan sama dengan parameter src.
Code	Kode konversi ruang warna
dstCn	Jumlah channel pada gambar tujuan, jumlah chanel berasal secara otomatis dari src dan code

Fungsi pemanggilan cv*threshold* pada opencv adalah *threshold*, adapun parameter yang ada didalam fungsi cv*threshold* seperti pada tabel 2.2:

Tabel 2.2 Parameter *threshold* OpenCV

Nama Parameter	Deskripsi
Src	Input array (single-chanel, 8 bit atau 32bit)
Dst	Keluaran array yang mempunyai ukuran yang sama pada parameter src.
Thresh	Nilai *threshold*
maxval	Maksimum nilai yang untuk yang digunakan pada tipe *threshold*
Type	Memilih tipe binary

Fungsi pemanggilan operasi morphologi pada opencv adalah cv*morphologiEx*, adapun parameter yang ada didalam fungsi cv*morphologiEx* seperti pada Tabel 2.3:

Tabel 2.3 Parameter cvmorphologiEx OpenCV

Nama Parameter	Deskripsi
Src.	Input array (single-chanel, 8 bit atau 32bit)
dst	Keluaran array yang mempunyai ukuran yang sama pada parameter src.
NULL	NULL
morphKernel	Struktur element.
op	Tipe dari operasi morphologi adalah sebagai berikut: CV_MOP_OPEN opening CV_MOP_CLOSE closing”
value	Nilai proses opening atau closing

Fungsi pemanggilan segmentasi *blob* pada opencv adalah *cvLabel*, adapun parameter yang ada didalam fungsi *cvLabel* seperti pada Tabel 2.4:

Tabel 2.4 Parameter cvLabel OpenCV

Nama Parameter	Deskripsi
src	Input array (single-chanel, 8 bit atau 32bit)
labelImg	Proses pemetaan objek
label	Keluaran array yang mempunyai ukuran yang sama pada parameter src.

Fungsi pemanggilan segmentasi *blob* kedua pada opencv adalah *cvFilterByArea*, adapun parameter yang ada didalam fungsi *cvFilterByArea* seperti pada Tabel 2.5:

Tabel 2.5 Parameter cvFilterByArea Opencv

Nama Parameter	Deskripsi
src	Input array (single-chanel, 8 bit atau 32bit)
minArea	Proses minimum area *blob*
maxArea	Proses maximum are *blob*

Fungsi pemanggilan segmentasi *blob* kedua pada opencv adalah *cvRenderblobs*, adapun parameter yang ada didalam fungsi *cvRenderblobs* seperti pada Tabel 2.6:

Tabel 2.6 Parameter cv*Renderblob*s Opencvcv*Renderblob*s

Nama Parameter	Deskripsi
labelImg	Proses Pemetaan Objek
label	input hasil pelabelan
frame	Proses *frame* awal atau inputan
frame	Proses *frame* referensi

Fungsi pemanggilan deteksi garis pada opencv adalah *cvLine*, adapun parameter yang ada didalam fungsi *cvLine* seperti pada Tabel 2.7:

Tabel 2.7 Parameter cvLine OpenCV

Nama Parameter	Deskripsi
src	Input array (single-chanel, 8 bit atau 32bit)
Pt1	titik awal pembuatan garis (x,y)
Pt2	titik akhir pembuatan garis (x,y).
color	warna dari garis yang dibuat