

## BAB II LANDASAN TEORI

Bab ini membahas teori-teori penunjang yang diperlukan untuk memahami permasalahan yang akan dibahas, serta penelitian terdahulu yang berhubungan dengan permasalahan pengolahan citra digital.

### A. STUDI PUSTAKA

Sebagai acuan dalam perancangan sistem pengolahan citra digital deteksi tepi untuk penaksiran bobot sapi, maka penulis mencari referensi terhadap beberapa penelitian sejenis yang berkaitan dengan penelitian ini. Berikut ini adalah beberapa penelitian yang dilakukan oleh peneliti terdahulu antara lain :

**Tabel 2.1** penelitian sejenis

No	Peneliti	Judul	Hasil penelitian
1	(Kurniawan Alhamal, Dr.Ir. Jangkung Raharjo., 2021)	Estimasi bobot sapi berdasarkan citra digital dengan metode fraktal dan klasifikasi decision tree	Hasil dari penelitian ini yaitu program yang dibuat dapat mengestimasi bobot sapi dengan metode fraktal dan klasifikasi decision tree. Sistem ini dapat dimanfaatkan menjadi alternatif dalam mencari bobot sapi. Implementasi sistem dapat menghasilkan tingkat akurasi estimasi sistem sebesar 81%, nilai root mean squared error perhitungan sekolah winter 72,56277013, denmark 75,0014818, 69,1126758. Waktu komputasi rata-rata 0,3329 detik. Akurasi dan waktu komputasi diperoleh dengan jumlah data latih

			sebanyak 47 citra dan jumlah data uji 21 citra, data diolah dengan parameter matriks dimensi fraktal sejumlah 5 dan menggunakan tree sejumlah 7.
2	(Supiyani, Haerul, and Padilah, n.d. 2021)	Pengolahan citra digital prediksi bobot sapi menggunakan ekstraksi fitur canny dan klasifikasi k-nearest neighbor	Hasil dari penelitian ini adalah implementasi sistem dapat menghasilkan tingkat akurasi yang sangat tinggi dalam proses prediksi bobot sapi potong mencapai 97,46%, dan perhitungan rata-rata adalah 9161s, apabila nilai setiap parameter yang digunakan dalam preprocessing adalah sebagai berikut, maka diperoleh akurasi dan waktu perhitungan sistem : rasio=0,3 dan operator cerdik=0,08. pada proses klasifikasi 10 citra uji, tingkat akurasi sistem mencapai 90%, nilai parameter k=1, dan perhitungan jarak KNN menggunakan jarak Euclidean.
3	(Latif et al., 2019)	Pengolahan citra digital untuk menentukan bobot sapi menggunakan metode canny edge detection	Hasil dari penelitian ini yaitu aplikasi pengolahan citra digital dapat diketahui bobot sapi menggunakan canny edge detection dengan menggunakan variabel lingkaran pada obyek sapi. Penentuan bobot sapi menggunakan rumus

			schoorl diperoleh rata-rata eror yang dihasilkan 7,176 kg, dan rata-rata berat 10 ekor sapi tersebut adalah 185 kg, sehingga tingkat akurasi hasil penelitian adalah 50,258%.
4	(Raharjo, Bambang Hidayat, and Fatah, n.d. 2019)	Estimasi bobot ternak sapi dengan metode deformable template dan klasifikasi support vector machine multiclass.	Berdasarkan hasil analisis terhadap sekenario pengujian yang telah dilakukan pada tugas akhir ini dapat ditarik kesimpulan sebagai berikut : segmentasi menggunakan deformable template mampu mensegmen citra sapi dan klasifikasi SVM mampu mengelaskan berat sapi. Hasil performansi terbaik diperoleh dari pengujian adalah ketika kondisi parameter rasio scaling resize sebesar 0,2, nilai threshold sebesar 80, dan jenis kernel SVM yang digunakan gaussian. Penerapan metode deformable template dan klasifikasi SVM dalam aplikasi estimasi bobot sapi mencapai tingkat akurasi terbaik sebesar 76,1905% dan waktu komputasi 0,701821 detik.
5	(Mustafid et al., 2018)	Sistem pengolahan citra digital untuk menentukan bobot sapi menggunakan	Aplikasi pengolahan citra yang dibangun dapat mengetahui bobot sapi dengan menggunakan variabel panjang badan dan

		metode titik berat	<p>lingkar dada. Hasil penentuan bobot sapi menggunakan rumus schoorl dan rumus modifikasi/lambourn memiliki nilai deviasi bobot badan sebesar 16,87% untuk rumus schoorl dan nilai penyimpangan bobot sebesar 10,58% untuk rumus modifikasi/lambourn. Nilai varian yang besar dari rumus schoorl dan rumus modifikasi/lambourn menunjukkan bahwa masih diperlukan adanya penelitian selanjutnya yang dapat membuat rumus baru untuk menentukan berat badan sapi secara lebih tepat dengan nilai varian yang lebih kecil.</p>
--	--	--------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Dari uraian penelitian diatas maka relevansi penelitian ini dengan penelitian-penelitian diatas adalah penerapan pengolahan citra digital dalam penaksiran bobot sapi. Perbedaan penelitian ini dengan penelitian-penelitian pengolahan citra diatas adalah terdapat pada metode deteksi tepi yang digunakan oleh penulis yaitu deteksi tepi dengan menggunakan metode sobel.

## B. Citra



**Gambar 2. 1** Citra sapi

Citra merupakan suatu representasi (gambaran), kemiripan, atau imitasi dari suatu objek. Menurut arti secara harfiah, citra (*image*) adalah gambar pada bidang dua dimensi. Ditinjau dari sudut pandang matematis citra yang terlihat merupakan cahaya yang direfleksikan dari sebuah objek. Sumber cahaya menerangi objek, objek memantulkan kembali sebagian dari berkas cahaya tersebut dan pantulan cahaya ditangkap oleh alat-alat optik, seperti mata manusia kamera, scanner, sensor satelit, dan lain-lain sehingga bayangan objek dalam bentuk citra dapat direkam. Citra sebagai keluaran dari suatu sistem perekaman data dapat bersifat optik berupa foto, citra analog berupa sinyal video seperti gambar monotor televisi, dan citra digital yang dapat langsung disimpan pada media penyimpanan magnetik. (“FITRI SANI NAJIHA-FST,” n.d.)

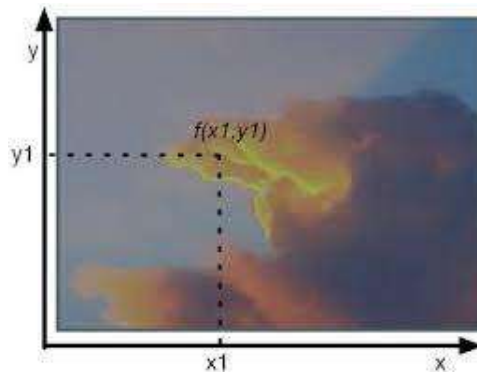
### 1. Citra analog

Citra analog adalah citra yang bersifat continue, seperti gambar pada monitor televisi, foto sinar X, foto yang tercetak di kertas foto, lukisan, hasil CT scan, gambar yang terekam dalam pita kaset, dan lain sebagainya. Citra analog tidak bisa direpresentasikan kedalam perangkat komputer, sehingga tidak dapat diproses secara langsung di komputer. Oleh sebab itu, agar citra dapat diproses oleh komputer proses konversi citra analog ke citra digital harus dilakukan terlebih dahulu. Citra analog dihasilkan dari alat-alat analog seperti kamera analog, CT scan, sensor rontgen, sensor gelombang pendek

pada sinar radar, sensor ultrasound pada sistem UDG dan lain sebagainya. (“Richard E. Wood. 2004,” n.d.)

## 2. Citra digital

Citra digital adalah citra yang dapat diolah oleh komputer, Secara umum citra digital menunjuk pada gambar dua dimensi yang dapat ditampilkan pada layar komputer sebagai himpunan berhingga (diskrit).



**Gambar 2. 2** Citra digital

Terlihat sebuah citra digital dengan kordinat  $(x,y)$ , citra digital memiliki angka-angka dengan demikian, citra digital dapat digambarkan sebagai suatu matriks, dimana indeks baris dan indeks kolom dari matriks menyatakan posisi suatu titik didalam citra dan harga dari elemen matriks menyatakan warna citra pada titik tersebut elemen-elemen matriks tersebut disebut juga dengan istilah pixel yang berasal dari kata *picture element*. (“FITRI SANI NAJIHA-FST,” n.d.)

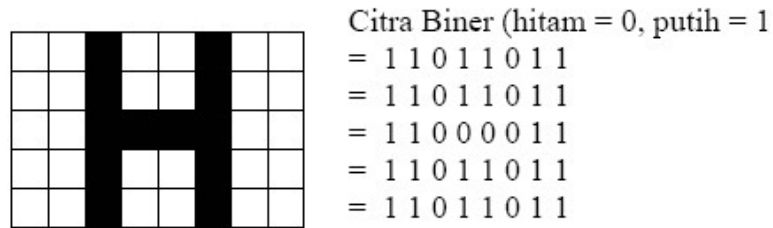
## 3. Jenis citra digital

Ada tiga jenis citra yang umum digunakan pada pengaplikasian pengolahan citra yaitu, color image atau true color, black and white image, dan binary image. Berikut ini penjelasan lebih lengkap mengenai jenis-jenis citra yang digunakan pada pengaplikasian pengolahan citra :

### a. Citra Biner

Citra jenis ini hanya mempunyai dua jenis warna yaitu hitam dan putih. Piksel-piksel objek bernilai 0 dan piksel-piksel latar belakang bernilai 1. Pada waktu menampilkan gambar, 0 adalah putih

dan 1 adalah hitam.



**Gambar 2. 3** Citra Biner

b. Citra Skala Keabuan

Skala keabuan memberikan kemungkinan warna yang lebih banya dari pada citra biner. Banyaknya kemungkinan nilai dan nilai maksimumnya bergantung pada jumlah bit yang digunakan. Misalnya untuk skala keabuan 8 bit, maka maka jumlah kemungkinan nilainya adalah  $2^8 = 256$ , dan nilai maksimumnya adalah  $2^8 - 1 = 255$ . Format citra ini disebut skala keabuan karena pada umumnya warna yang dipakai adalah antara hitam sebagai warna minimal dan putih sebagai warna maksimalnya, sehingga warna antaranya adalah abu-abu.(Puspandari, n.d.)

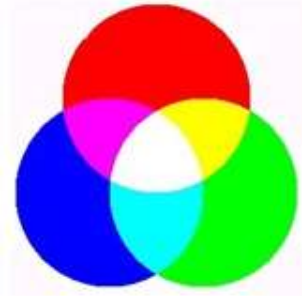


**Gambar 2. 4** Citra skala keabuan

c. *Color image* atau citra RGB (*True color*)

Pada *color image* tiap pixel memiliki warna tertentu, warna tersebut adalah merah (*red*), hijau (*green*), dan biru (*blue*). Pada umumnya komputer menampilkan 8, 16, dan 24 bit untuk setiap piksel. Banyaknya bit setiap piksel menentukan banyak warna berbeda yang dapat ditampilkan. Setiap warna dasar menggunakan penyimpanan 8 bit = 1 byte, yang berarti setiap warna mempunyai range 0-255 dan setiap piksel mempunyai kombinasi warna sebanyak  $2^8 \cdot 2^8 \cdot 2^8 = 2^{24} = 16.777.216$  warna. Itulah sebabnya format ini dinamakan *true color* karena mempunyai jumlah warna yang cukup

besar sehingga bisa dikatakan hampir mencakup semua warna di alam.  
(“FITRI SANI NAJIHA-FST,” n.d.)



**Gambar 2. 5** Citra RGB

#### **4. Format file citra**

Format file citra standar yang digunakan saat ini terdiri dari beberapa jenis. Format-format ini digunakan untuk menyimpan citra dalam sebuah file. Setiap format memiliki karakteristik masing-masing. Berikut adalah beberapa format umum yang digunakan saat ini.

##### *a. Bitmap (.bmp)*

Format .bmp adalah format penyimpanan standar tanpa kompresi yang umum dapat digunakan untuk menyimpan citra biner hingga citra warna. Format ini terdiri dari beberapa jenis yang setiap jenisnya ditentukan dengan jumlah bit yang digunakan untuk menyimpan sebuah nilai pixel.

##### *b. Tagged image format (.tif,.tiff)*

Format .tif merupakan format penyimpanan citra yang dapat digunakan untuk menyimpan citra bitmap hingga citra dengan warna palet terkompresi. Format ini dapat digunakan untuk menyimpan citra yang tidak terkompresi dan juga citra terkompresi.

##### *c. Portable network graphics (.png)*

Format .png adalah format penyimpanan citra terkompresi. Format ini dapat digunakan pada citra grayscale, citra dengan palet warna, dan juga citra fullcolor. Format .png juga mampu menyimpan informasi hingga kanal alpha dengan penyimpanan sebesar 1 hingga 16 bit per



kanal.

d. JPEG (*.jpg*)

*.jpg* adalah format yang sangat umum digunakan saat ini khususnya untuk transmisi citra. Format ini digunakan untuk menyimpan citra hasil kompresi dengan metode JPEG.

e. MPEG (*.mpg*)

Format ini digunakan di dunia internet dan diperuntukan sebagai format penyimpanan citra bergerak (video). Format ini mendukung video dengan kompresi ber-rugi.

f. Graphics interchange format (*.gif*)

Format ini dapat digunakan pada citra warna dengan palet 8 bit. Penggunaan umumnya pada aplikasi web. Kualitas yang rendah menyebabkan format ini tidak terlalu populer dikalangan peneliti pengolahan citra digital.

g. RGB (*.rgb*)

Format ini merupakan format penyimpanan yang dibuat oleh silicon graphics untuk menyimpan citra berwarna.

h. RAS (*.ras*)

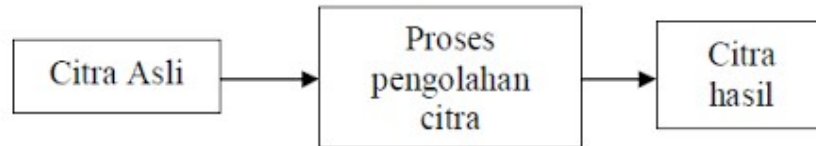
Format *.ras* digunakan untuk menyimpan citra dengan format RGB tanpa kompresi.

i. Portable image file format

Format ini memiliki beberapa bagian diantaranya adalah portable bitmap, portable graymap, portable pixmap, dan portable network map dengan format berturut-turut adalah *.pbm*, *.pgm*, *.ppm*, dan *.pnm*. format ini baik digunakan untuk menyimpan dan membaca kembali data citra.(FITRI SANI NAJIHA-FST, n.d.)

## C. Pengolahan citra

### 1. Definisi pengolahan citra



**Gambar 2. 6** Proses pengolahan citra digital

Pengolahan citra merupakan suatu proses perbaikan citra untuk meningkatkan kualitas citra menjadi lebih baik, agar citra yang mengalami gangguan mudah diinterpretasi oleh manusia maupun mesin/komputer. Teknik-teknik pengolahan citra mentransformasikan citra menjadi sebuah masukan dan keluaranya juga berupa citra, namun dengan kualitas yang lebih baik daripada citra masukan.

Meskipun sebuah citra kaya akan informasi, namun seringkali mengalami penurunan mutu (degradasi), misalnya mengandung cacat atau derau (noise), warna yang terlalu kontras, kurang tajam, kabur, dan sebagainya. Citra semacam ini tentu sulit diinterpretasikan karena informasi yang disampaikan oleh citra tersebut menjadi kurang. Maka, proses pengolahan citra sangat diperlukan. Tujuan dari pengolahan citra digital adalah sebagai berikut :

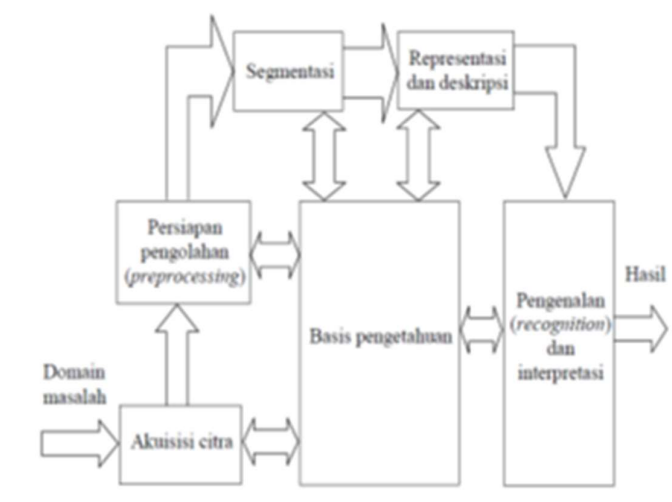
- a. Memperbaiki kualitas gambar dilihat dari aspek radiometrik (peningkatan kontras, transformasi warna, restorasi citra) dan dari aspek geometrik (rotasi, translasi, skala, transformasi geometrik).
- b. Melakukan proses penarikan informasi atau deskripsi atau pengenalan objek yang terkandung pada citra.
- c. Melakukan kompresi atau reduksi data untuk tujuan penyimpanan data, transmisi data, dan waktu proses data.

Proses pengolahan citra, khususnya dengan menggunakan komputer akan menghasilkan hasil yang lebih baik dari sebelumnya.

Berikut adalah alur dari pengolahan citra. (“FITRI SANI NAJIHA-FST,” n.d.)

## 2. Langkah-langkah pengolahan citra

Dalam pengolahan citra terdapat langkah-langkah penting. Langkah-langkah yang dilakukan dalam pengolahan citra dapat dilakukan seperti.



**Gambar 2.7** Langkah pengolahan citra

### a. Akuisisi citra

Akuisisi citra adalah tahap awal untuk mendapatkan citra digital. Tujuan dari proses ini adalah untuk menentukan data yang diperlukan dan memilih metode perekam digital. Tahap ini dimulai dari objek yang akan diambil gambarnya, persiapan alat-alat, sampai pada pencitraan. Dimana pencitraan adalah kegiatan transformasi dari citra tampak (foto, gambar, lukisan, dan lain sebagainya) menjadi citra digital.

### b. Preprocessing

Preprocessing memerlukan tahapan untuk menjamin kelancaran pada proses berikutnya, antara lain :

1. Meningkatkan kualitas citra (kontras, kecerahan, dan lain

sebagainya).

2. Menghilangkan noise.
3. Perbaiki citra (image restoration).
4. Transformasi (image transformation).
5. Menentukan citra yang akan diobservasi.

c. Segmentasi

Segmentasi bertujuan untuk mempartisi citra menjadi bagian-bagian pokok yang mengandung informasi penting, misalnya pada pemisahan objek dan latar belakang.

d. Representasi dan deskripsi

Representasi adalah suatu proses untuk mempresentasikan suatu wilayah sebagai suatu daftar titik-titik kordinat dalam kurva yang tertutup, dengan deskripsi luasan perimeternya. Proses selanjutnya dilakukan deskripsi citra dengan cara seleksi citra dan ekstraksi citra (feature extraction and selection). Dimana seleksi citra bertujuan untuk memilih informasi kuantitatif dari citra yang ada, dan dapat membedakan kelas-kelas objek dengan baik, sedangkan ekstraksi citra mempunyai tujuan untuk mengukur besaran kuantitatif citra setiap piksel, misalnya rata-rata, standar deviasi, dan lain-lain.

e. Pengenalan dan interpretasi

Tahap pengenalan bertujuan untuk memberi label pada sebuah objek yang informasinya disediakan oleh descriptor, berbeda dengan tahap interpretasi yang bertujuan untuk memberi arti atau makna kepada kelompok objek-objek yang dikenali.

f. Basis pengetahuan

Basis pengetahuan bertujuan untuk memandu operasi dari masing-masing modul proses mengontrol interaksi antara modul-modul tersebut, dan dapat sebagai referensi pada proses pengenalan

pola (template matcing). (“FITRI SANI NAJIHA-FST,” n.d.)

### **3. Oprasi pengolahan citra**

Oprasi-oprasi yang dilakukan di dalam pengolahan citra banyak ragamnya. Namun secara umum, oprasi pengolahan citra dapat diklasifikasikan dalam beberapa jenis sebagai berikut :

#### **a. Perbaikan kualitas citra**

Jenis operasi ini bertujuan untuk memperbaiki kualitas citra dengan cara memanipulasi parameter-parameter citra. Dengan operasi ini citra-citra khusus yang terdapat didalam citra lebih ditonjolkan.

#### **b. Pemugaran citra**

Operasi ini bertujuan untuk menghilangkan atau meminimumkan cacat pada citra. Tujuan pemugaran citra hampir sama dengan operasi perbaikan citra. Bedanya pada pemugaran citra penyebab degradasi gambar diketahui.

#### **c. Pemampatan citra**

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 pemampatan adalah citra yang telah dimampatkan harus tetap mempunyai kualitas gambar yang bagus.

#### **d. Segmentasi citra**

Jenis operasi ini bertujuan untuk memecah citra kedalam beberapa segmen dengan suatu kriteria tertentu.

#### **e. Analisis citra**

Jenis operasi ini bertujuan untuk menghitung besar kuantitatif dari citra untuk menghasilkan deskripsinya. Teknik analisis citra mengekstraksi ciri-ciri tertentu yang membantu dalam mengidentifikasi objek. Proses segmentasi kadangkala diperlukan

untuk melokalisasi objek yang diinginkan dari sekelilingnya.

f. Rekonstruksi citra

jenis operasi ini bertujuan untuk membentuk ulang objek dari beberapa citra hasil proyeksi. Operasi rekonstruksi citra banyak digunakan dalam bidang medis. ("Muhammad Nurullah 2014" n.d.)

#### D. Bobot sapi

untuk cara pengukuran berat badan sapi secara konvensional ada dua cara yaitu :

##### 1. Pengukuran berat sapi dengan timbangan

pengukuran berat badan sapi menggunakan timbangan sapi adalah cara yang paling akurat dan tepat, tidak jarang sering diterapkan di peternakan-peternakan moderen baik skala menengah maupun besar. Namun cara ini jarang digunakan di peternakan kecil dikarenakan harga timbangan sapi yang lumayan mahal.

##### 2. Pengukuran berat sapi dengan rumus

Pengukuran berat badan sapi bisa dilakukan dengan menggunakan beberapa rumus, yaitu:

a. Rumus Schoorl Denmark

Rumus schoorl denmark yaitu:

$$BB = \frac{(LD + 22)^2}{100}$$

Keterangan :

BB : berat badan (Kg)

LD : lingkar dada (cm)

Faktor 22 : faktor penambah untuk lingkar dada pada sapi.

b. Rumus Schoorl indonesia

Rumus Schoorl indonesia yaitu:

$$BB = \frac{(LD + 18)^2}{100}$$

Keterangan :

BB : berat badan (Kg)

LD : lingkar dada (cm)

Faktor 18 : faktor penambah untuk lingkar dada pada sapi

c. Rumus Winter

Rumus Winter yaitu:

$$BB = \frac{(LD^2 + PB)^2}{10815,15}$$

Keterangan :

BB : berat badan (Kg)

LD : lingkar dada (cm)

PB : panjang badan (cm).

(Raharjo, Bambang Hidayat, and Fatah, n.d.)

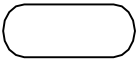
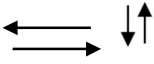
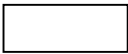
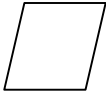
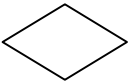


## E. *Unified Modeling Language (UML)*

UML (*Unified Modeling Language*) adalah bahasa untuk membangun dan mendokumentasikan *artifacts* (bagian dari informasi yang digunakan atau dihasilkan oleh proses pembuatan perangkat lunak, *artifact* tersebut dapat berupa model, deskripsi, atau perangkat lunak) dari sistem perangkat lunak. Selain itu UML adalah bahasa pemodelan yang menggunakan konsep orientasi *object*. UML menyediakan notasi-notasi yang membantu memodelkan sistem dari berbagai perspektif. UML tidak hanya digunakan dalam pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan (Ipo Novianto 2017)

## 1. Flowchart

Flowchart adalah bagan-bagan yang mempunyai arus yang menggambarkan langkah-langkah penyelesaian suatu masalah. Flowchart merupakan cara penyajian dari suatu algoritma (Ipo Novianto 2017) . Simbol flowchart dapat dilihat pada Tabel

**Tabel 2.2** Simbol-simbol flowchart

No	Simbol	Nama Simbol dan Keterangan
1.		<b>Simbol Terminal</b> adalah untuk menyatakan permulaan atau akhir suatu pemrograman.
2.		<b>Simbol Arus / Flow</b> yaitu untuk menyatakan jalannya arus suatu proses (arah aliran program).
3.		<b>Simbol Proses</b> yaitu untuk menyatakan proses perhitungan / proses pengolahan data.
4.		<b>Simbol Input - Output</b> yaitu untuk memasukan data (proses <i>input</i> ) maupun menunjukkan hasil ( <i>output</i> data) dari suatu proses.
5.		<b>Simbol Decision / Logika</b> yaitu untuk menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban ya / tidak.
6.		<b>Simbol Document</b> yaitu untuk mencetak laporan ke printer
7.		<b>Simbol Manual Operation</b> adalah simbol berfungsi untuk menunjukkan pengolahan yang




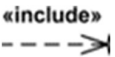


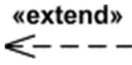

		tidak dilakukan oleh komputer.
--	--	--------------------------------

## 2. Use Case Diagram

Use case adalah abstraksi dari interaksi antara sistem dan aktor. Use case bekerja dengan cara mendeskripsikan tipe interaksi antara user sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Use case merupakan konstruksi untuk mendeskripsikan bagaimana sistem akan terlihat di mata user. Sedangkan use case diagram memfasilitasi komunikasi diantara analis dan pengguna serta antara analis dan client (Ipo Novianto 2017).

**Tabel 2.3** *Use Case Diagram*

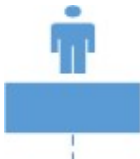

No	Simbol	Nama Simbol dan Keterangan
1.		<i>Actor</i> adalah menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2.		<i>Use Case</i> menjelaskan urutan kegiatan yang dilakukan <i>actor</i> dan sistem untuk mencapai suatu tujuan tertentu.
3.		<i>Association</i> penghubung antara <i>Actor</i> dan <i>Use Case</i>
4.		<i>Include</i> yaitu perilaku <i>use case</i> merupakan bagian dari <i>use case</i> lain.





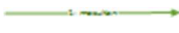

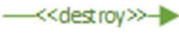
5.		<p><i>Extend</i> yaitu perilaku suatu <i>use case</i> memperluas perilaku <i>use case</i> lain. <i>Use case</i> boleh dilakukan boleh tidak (optional)</p>
6.		<p><i>Sistem Boundary</i> yaitu batasan dari sebuah sistem.</p>

### 3. Sequence Diagram

Diagram sequence menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima oleh objek. Oleh karena itu untuk menggambar diagram sequence maka harus diketahui objek-objek yang terlibat dalam sebuah use case beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sequence juga dibutuhkan untuk melihat skenario yang ada pada use case. Berikut adalah simbol-simbol yang ada pada diagram sequence (Ipo Novianto 2017)

**Tabel 2.4** *Sequence Diagram*

No	Simbol	Keterangan
1.		<p><i>aktor</i>, orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri</p>
2.		<p>Garis hidup/<i>lifeline</i>, menyatakan kehidupan suatu objek</p>

3.		Objek, menyatakan objek yang berinteraksi pesan
4.		Waktu aktif, menyatakan objek dalam keadaan aktif dan berinteraksi
5.		Pesan tipe <i>create</i> , menyatakan suatu objek membuat objek lain arah panah mengarah pada objek yang dibuat
6.		Pesan tipe <i>call</i> , menyatakan suatu objek memanggil operasi atau metode yang ada pada objek lain atau dirinya sendiri
7.		Pesan tipe <i>send</i> , menyatakan bahwa suatu objek mengirim data atau masukan atau informasi ke objek lainnya arah panah mengarah pada objek yang dikirim
8.		Pesan tipe <i>return</i> , menyatakan bahwa suatu objek yang telah menjalankan operasi atau metode menghasilkan suatu kembalian ke objek tertentu arah panah mengarah pada objek yang menerima kembalian
9.		Pesan tipe <i>destroy</i> , menyatakan suatu objek mengakhiri hidup objek yang lain arah panah mengarah pada objek yang diakhiri sebaiknya jika ada <i>create</i> maka ada <i>destroy</i>

#### 4. Class Diagram




Class diagram adalah diagram yang digunakan untuk menampilkan beberapa kelas serta paket-paket yang ada dalam sistem/perangkat lunak yang sedang kita gunakan. Class diagram memberi kita gambaran tentang perangkat lunak dan relas-relasi yang ada didalamnya. Menurut (Whitten L. Jeffery et al 2004:432) Class Diagram adalah diagram yang menunjukkan class-class yang ada dari sebuah sistem dan hubungannya secara logika. Class diagram menggambarkan struktur statis dari sebuah sistem. Karena itu class diagram merupakan tulang punggung atau kekuatan dasar dari hampir setiap metode berorientasi objek termasuk UML (Ipo Novianto 2017).

Atribut dan operation dapat memiliki salah satu sifat berikut :

- a. *Private* hanya bisa dipanggil dari dalam kelas itu sendiri. Atribut diawali “-”.
- b. *Protected* hanya dapat dipanggil oleh class yang bersangkutan dan class turunannya. Atribut diawali dengan tanda “#”.
- c. *Public* dapat dipanggil dari semua objek atribut diawali tanda “+”

Ada beberapa simbol *relationships* antar class yg digunakan pada diagram class. Berikut adalah beberapa simbol-simbol yang ada pada class diagram dapat dilihat pada Tabel :

**Tabel 2.5 Class Diagram**

No	Simbol	Nama dan Keterangan
1.		<i>Asosiasi</i> adalah hubungan antar kelas.
2.		<i>Generalization</i> adalah relasi antar kelas dengan makna generalisasi – spesialisasi (umum-khusus).
3.		<i>Composition</i> adalah bentuk khusus dari agregasi dimana kelas yang menjadi bagian diciptakan setelah kelas menjadi <i>whole</i> dibuat. Misal kelas <i>whole</i> dihapus, maka kelas yg menjadi part ikut musnah.

Multiplicity adalah jumlah banyaknya objek sebuah kelas yang berelasi dengan sebuah objek dari kelas lain yang berasosiasi dengan kelas tersebut (Ipo Novianto 2017). Macam-macam multiplicity disajikan dalam Tabel 2.5

**Tabel 2.6 Multiplicity**

Nilai Kardinalitas	Arti
0..1	Nol atau satu
1	Hanya satu
0..*	Nol atau lebih
1..*	Satu atau lebih
N	Hanya n (dengan $n > 1$ )
0..n	Nol sampai n (dengan $n > 1$ )
1..n	Satu sampai n (dengan $n > 1$ )

## F. Open Cv



**Gambar 2. 8** Logo OpenCv

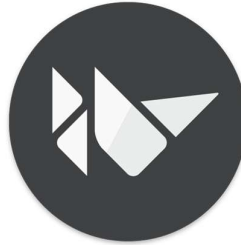
OpenCV (Open Source Computer Vision) adalah sebuah perpustakaan (library) komputer yang berfokus pada pengolahan citra dan analisis komputer. Tujuan utama dari OpenCV adalah untuk menyediakan alat dan fungsi yang kuat untuk memanipulasi, menganalisis, dan memahami citra dan video dalam berbagai konteks, termasuk pengenalan wajah, deteksi objek, kalibrasi kamera, visi mesin, dan sebagainya.

OpenCV ditulis dalam bahasa pemrograman C++, tetapi juga menyediakan antarmuka pemrograman untuk bahasa pemrograman seperti Python dan Java, sehingga memungkinkan pengguna dengan berbagai latar belakang untuk menggunakan dan memanfaatkan fungsionalitasnya. Perpustakaan ini bersifat open source, yang berarti kode sumbernya dapat diakses dan dimodifikasi oleh siapa saja sesuai dengan lisensi BSD yang melindungi hak cipta dan penggunaannya.

OpenCV menawarkan berbagai algoritma dan fungsi yang siap pakai untuk mempermudah pengolahan citra, termasuk transformasi citra, deteksi tepi, segmentasi, ekstraksi fitur, dan pemrosesan video. Hal ini memungkinkan para pengembang dan peneliti untuk mengembangkan aplikasi visi komputer dengan lebih cepat dan efisien. OpenCV juga kompatibel dengan berbagai perangkat keras dan sistem operasi, sehingga dapat diintegrasikan dengan berbagai platform.

Dengan OpenCV, pengguna dapat melakukan berbagai tugas pengolahan citra dan analisis komputer, seperti mendeteksi objek dalam gambar, melacak objek bergerak dalam video, mengukur jarak dan sudut antara objek, mengenali wajah dan fitur wajah, serta memperbaiki citra dengan menghilangkan noise atau mengubah kontras, kecerahan, dan lain sebagainya.

## G. Kivy



**Gambar 2.9** Logo kivy

Kivy adalah sebuah framework Python yang digunakan untuk mengembangkan aplikasi dengan antarmuka pengguna (UI) berbasis NUI (Natural User Interface) seperti multitouch apps, aplikasi desktop, dan mobile. Kivy dikembangkan dengan tujuan membuat pengembangan aplikasi UI menjadi lebih mudah, efisien, dan konsisten antar platform, serta dapat berjalan pada berbagai platform seperti Windows, macOS, Linux, Android, iOS, dan Raspberry Pi. Beberapa fitur yang dimiliki oleh Kivy antara lain:

1. Menggunakan bahasa pemrograman Python, yang mudah dipelajari dan memiliki banyak dukungan komunitas.
2. Mendukung input dan output berbasis NUI seperti multi-touch, gestur, dan sensor-sensor pada perangkat mobile.
3. Mendukung animasi, transisi, dan efek visual pada antarmuka pengguna.
4. Memiliki struktur pengembangan aplikasi yang modular dan fleksibel, sehingga memudahkan pengembangan aplikasi yang kompleks.
5. Dapat berjalan pada berbagai platform seperti Windows, macOS, Linux, Android, iOS, dan Raspberry Pi.
6. Open-source dengan lisensi MIT.

Kivy juga dilengkapi dengan berbagai tools dan library seperti Kivy Designer (untuk membuat antarmuka pengguna secara visual), KivyMD (library untuk membuat aplikasi dengan Material Design), dan KivyPie (operating system berbasis Raspberry Pi yang telah dikonfigurasi untuk menjalankan Kivy secara langsung).