

## **BAB II**

### **LANDASAN TEORI**

#### **II.1. Tanaman Kopi**

Tanaman kopi (*Coffea* spp.) merupakan komoditas ekspor unggulan yang dikembangkan di Indonesia karena mempunyai nilai ekonomis yang relatif tinggi di pasaran dunia. Permintaan kopi Indonesia dari waktu ke waktu terus meningkat karena seperti kopi Robusta mempunyai keunggulan bentuk yang cukup kuat serta kopi Arabika mempunyai karakteristik cita rasa (acidity, aroma, flavour) yang unik dan ekselen (Main Sese, Nurariaty, Annie, 2012).

Dalam hal perkopian di Indonesia , kopi rakyat memegang peranan yang penting, mengingat sebagian besar (93 %) produksi kopi merupakan kopi rakyat. Namun demikian kondisi pengelolaan usaha tani pada kopi rakyat relatif masih kurang baik dibanding kondisi perkebunan besar Negara (PBN). Ada dua permasalahan utama yang diidentifikasi pada perkebunan kopi rakyat, yaitu rendahnya produktivitas dan mutu hasil yang kurang memenuhi syarat untuk diekspor. Di Sulawesi Selatan berdasarkan data Statistik Dinas Perkebunan Prov. Sul Sel tahun 2008, luas areal pertanaman kopi Arabika sebesar 47.181,46 ha yang melibatkan 65.178 KK petani dengan total produksi hanya sebesar 19.384,69 ton, karena produktivitasnya yang masih sangat rendah yaitu hanya sebesar 636,24 kg/ha/tahun, sementara potensi produksinya dapat mencapai 1.500 kg/ha/tahun. Demikian halnya dengan Kabupaten Enrekang yang merupakan salah

satu daerah penghasil kopi Arabika di Sulawesi Selatan dari luas areal sebesar 11.384 ha dengan jumlah petani sebanyak 16.632 KK produksinya pada tahun 2008 hanya sebesar 5.350 ton karena produktivitas hanya mencapai 648,48 kg/ha/tahunnya (Main Sese, Nurariaty, Annie, 2012).

Rendahnya produktivitas kopi di antaranya disebabkan adanya serangan Organisme Pengganggu Tumbuhan (OPT). Beberapa jenis OPT yang menyerang tanaman kopi di Sulawesi Selatan adalah hama penggerek buah kopi (*Hypothenemus hampei* Ferr.), penggerek batang, (*Zeuzera* sp.), Penggerek cabang (*Xylosandrus* spp.), kutu hijau (*Coccus viridis*), kutu putih (*Ferrisia virgata*), penyakit karat daun (*Hemileia vastatrix*), *Cercospora* (Main Sese, Nurariaty, Annie, 2012).

Tanaman kopi harus dihindarkan dari serangan hama dan penyakit. Hal ini dikarenakan faktor tersebut dapat menurunkan produksi dan mutu kopi yang dihasilkan. Bahkan, akibat serangan hama dan penyakit dapat menyebabkan tanaman tidak mau berbuah sama sekali, atau bahkan sering menyebabkan kematian (Sonya, 2016).

### **II.1.1. Hama**

Hama adalah organisme perusak tanaman pada akar, batang, daun atau bagian tanaman lainnya sehingga tanaman tidak dapat tumbuh dengan sempurna atau mati. Serangga dikatakan hama apabila serangga tersebut mengurangi kualitas dan kuantitas bahan makanan, pakan ternak, tanaman serat, hasil pertanian atau panen, pengolahan dan dalam penggunaannya serta dapat bertindak sebagai vektor penyakit pada tanaman, binatang dan manusia, dapat merusak

tanaman hias , bunga serta merusak bahan bangunan dan milik pribadi lainnya (Sonya, 2016).

#### 1. Hama pada Akar (Nematoda)

Hama nematoda merusak akar-akar halus dan kecil. Hama ini umumnya menyerang tanaman kopi yang masih muda (berumur 1-2 tahun). Berikut beberapa hama nematoda yang sering dijumpai di tanaman kopi adalah *Pratylenchus coffeae*, *Meloidogyne* spp., *Radopholus similis*, dan *Heterodera marioni*.

#### 2. Hama pada Batang dan Dahan

Hama penggerek batang berupa larva dari beberapa jenis kumbang, seperti penggerek merah (*Zeuzera coffeae*). Jenis penggerek merah memiliki ciri fisik khusus yaitu adanya bulatan warna kuning di bagian dada dan perut. Ukuran hama ini dapat mencapai 5 mm. Hama penggerek betina bisa bertelur hingga 1.000 butir. Ciri utama telur hama ini diantaranya berwarna kuning, berkelompok kecil, dan biasanya terletak di bagian dalam kulit cabang. Perubahan telur menjadi larva terjadi cukup cepat. Setelah menjadi larva, hama ini menggerek menuju batang cabang dan berkembang biak menjadi kepompong.

#### 3. Hama pada Ranting

Hama penggerek ranting di antaranya *Xylosandrus morstati* dan *Xylosandrus morigerus*. Keduanya sering disebut bibik dahan. Cara merusak dan berkembang biak hama ini dengan menggerek kulit dan membuat lubang kevil di dalam kayu. Lubang tersebut merupakan tempat bertelurnya hama

betina. Perkembangan telur menjadi mago sekitar tiga minggu. Makanannya berupa cendawan *Ambrosia* yang ada di dalam lubang. Kumbang atau hama penggerek ini biasanya tidak ada pada musim kemarau.

#### 4. Hama pada Buah

Jenis hama pada buah yang paling sulit diberantas diantaranya *Hypothenemus hampei* dan *Stephanoderes hampei*. Kerugian akibat hama tersebut dapat mencapai 80%. Hama tersebut pertama kali dijumpai di Bogor pada tahun 1909. Awalnya, induk hama menggerek buah kopi, lalu masuk ke dalam buah dan bertelur. Satu indukan dapat menggerek 20 buah dalam waktu 2-6 hari. Setelah 5-10 hari, telur menetas menjadi larva. Setelah itu, larva berkembang menjadi kepompong (pupa) selama 6-16 hari. Selesai perkembangan itu, terjadilah perkawinan antara jantan dan betina di dalam biji kopi. Selanjutnya, betina akan pindah dan kembali bertelur di dalam lubang buah. Siklus ini terjadi selama lebih dari dua bulan.

#### 5. Hama pada Daun, Tunas, dan Bunga

Hama yang sering menyerang daun, tunas, dan bunga tanaman kopi biasanya berupa kutu, seperti *Pseudococcus citri*, *Coccus viridis*, *Ferrissia virgata*. Berbagai kutu ini merusak tanaman kopi dengan menghisap cairan zat makanan di tunas muda, daun muda, ranting, dompolan bunga, dan buah menggunakan paruhnya.

### **II.1.2. Penyakit**

Penyakit tanaman dapat diartikan gangguan terhadap tanaman yang disebabkan oleh pathogen dan non pathogen yang menyebabkan terganggunya

proses pertumbuhan pada bagian-bagian tertentu dari tanaman yang tidak dapat berjalan sesuai fungsinya dengan normal dan dengan baik sehingga menghambat pertumbuhan pada tanaman (Sonya, 2016).

#### 1. Penyakit Karat Daun

CLR (*Coffea Leaf Rust*) merupakan penyakit karat daun yang disebabkan oleh *Hemileia vastatrix* melalui penyebaran angin atau udara. Sporanya berbentuk seperti kutil. Penyakit ini umumnya menyerang daun muda dengan kondisi kelembaban lingkungan yang tinggi. Penyakit ini sangat terkenal keganasannya, khususnya untuk perkebunan kopi arabika. Penyakit ini jarang menyerang tanaman kopi robusta. Pasalnya, perkebunan robusta umumnya berada di dataran rendah yang memiliki kelembaban lingkungan yang rendah.

#### 2. Penyakit Bercak Daun

Penyakit bercak daun disebabkan oleh *Cercospora coffeicola*. Penyakit ini bisa menyerang kopi jenis arabika dan robusta. Serangannya dimulai dari daun, lalu merusak bagian buah. Penyakit ini biasanya menyerang saat pembibitan atau persemaian. Terdapat bercak berwarna coklat dan coklat tua berbentuk bulat.

#### 3. Penyakit Daun Hangu

Penyakit daun hangu disebabkan oleh *Root dauw*. Ciri-cirinya adalah adanya lapisan berwarna hitam dipermukaan daun, lalu terdapat kumpulan semut di bagian daun. Suhu daun yang terserang mengalami kenaikan dan menyebabkan kelayuan.

#### 4. Penyakit pada Bunga dan Buah

Penyakit pada bunga biasanya disebabkan oleh iklim yang ekstrem atau kondisi tanah yang sangat lembab dan basah. Salah satu penyakit pada bunga yang banyak dijumpai adalah bunga bintang. Penyakit ini umumnya menyerang tanaman kopi jenis arabika. Sementara itu, penyakit yang sering terjadi di bagian buah adalah penyakit rontok buah. Penyakit ini dapat disebabkan oleh siklus tumbuh yang abnormal dan iklim ekstrem.

#### 5. Penyakit Bercak Hitam

Penyakit bercak hitam pada buah disebabkan oleh *Cephaleuros coffea*. Mikroorganisme ini pada umumnya menyerang buah kopi yang belum matang. Ciri-cirinya terdapat bercak-bercak hitam pada buah kopi muda (berwarna hijau), dan terdapat rambut-rambut halus dan butiran-butiran berwarna merah di buah kopi (Edy Panggabean, 2011).

## II.2. Teknologi Komputer

Teknologi komputer berkorelasi dengan teknologi informasi dan digunakan untuk mempelajari jaringan komputer, dan lain-lain, selain itu juga mencakup daftar perangkat lunak dan perangkat lunak yang beragam. Termasuk pemrograman, jaringan, desain database dan pengembangan untuk memastikan komputer bekerja dengan baik. (Rio, 2014)

Secara definisi **Teknologi komputer** juga bisa diartikan kombinasi perangkat keras atau perangkat lunak yang terkait dengan komputer seperti komputer multimedia, berbagai jenis media (grafik, audio dan video) dan alat

komunikasi berjejaring (email, bulletin board, Blackboard atau WebCT, dan listserves), bersama dalam setting pendidikan untuk Meningkatkan pembelajaran bahasa program.

### **II.2.1. Kecerdasan Buatan**

Kecerdasan Buatan (Artificial Intelligence) adalah salah satu cabang Ilmu pengetahuan berhubungan dengan pemanfaatan mesin untuk memecahkan persoalan yang rumit dengan cara yang lebih manusiawi. Hal Ini biasanya dilakukan dengan mengikuti/mencontoh karakteristik dan analogi berpikir dari kecerdasan/ intelegensi manusia, dan menerapkannya sebagai algoritma yang dikenal oleh komputer. Dengan suatu pendekatan yang kurang lebih fleksibel dan efisien dapat diambil tergantung dari keperluan, yang mempengaruhi bagaimana wujud dari perilaku kecerdasan buatan. biasanya dihubungkan dengan Ilmu Komputer, akan tetapi juga terkait erat dengan bidang lain seperti Matematika, Psikologi, Pengamatan, Biologi, Filosofi, dan yang lainnya. Kemampuan untuk mengkombinasikan pengetahuan dari semua bidang ini pada akhirnya akan bermanfaat bagi kemajuan dalam upaya menciptakan suatu kecerdasan buatan. (Sutojo et al, 2010).

### **II.2.2. Jaringan Syaraf Tiruan**

*Artificial Neural Network* atau Jaringan Syaraf Tiruan (JST) adalah salah satu cabang dari *Artificial Intelligence*. JST merupakan suatu sistem pemrosesan informasi yang memiliki karakteristik menyerupai jaringan syaraf biologi (Kusumadewi, 2003). Demikian juga Fachrul dan Hani (2012) menyatakan bahwa JST adalah sebuah mesin yang dirancang untuk memodelkan cara otak manusia

mengerjakan fungsi atau tugas-tugas tertentu. Mesin ini memiliki kemampuan menyimpan pengetahuan berdasarkan pengalaman dan menjadikan pengetahuan yang dimiliki menjadi bermanfaat.

Menurut Sutojo *et al* (2010), Jaringan Syaraf Tiruan merupakan sistem pemrosesan informasi yang memiliki karakteristik yang mirip dengan jaringan syaraf manusia. Jaringan syaraf tiruan ini juga dapat diterapkan untuk mengenali pola atau memetakan suatu masukan menjadi keluaran yang dilatih melalui suatu proses pelatihan dan dikembangkan menjadi pemodelan matematis dari syaraf manusia yang berdasar pada asumsi bahwa :

- a. Proses informasi terjadi pada beberapa elemen sederhana yaitu *neuron*.
- b. Sinyal terhubung diantara *neuron* menciptakan jaringan koneksi.
- c. Setiap jaringan koneksi penghubung memiliki bobot yang terhubung, yang dalam jaringan saraf tertentu mengalikan sinyal yang ditransmisikan.
- d. Setiap *neuron* mempunyai fungsi aktivasi (biasanya tidak linier) pada jaringan masukannya (jumlah dari bobot sinyal keluaran) untuk menentukan sinyal keluarannya.

Karakteristik dari jaringan saraf tiruan adalah :

- a. Pola terhubung diantara *neuron* (yang menjadi arsitekturnya).
- b. Metode penentuan bobot dalam koneksi (disebut sebagai proses latihan, pembelajaran, atau Algoritma).
- c. Fungsi aktivasi.

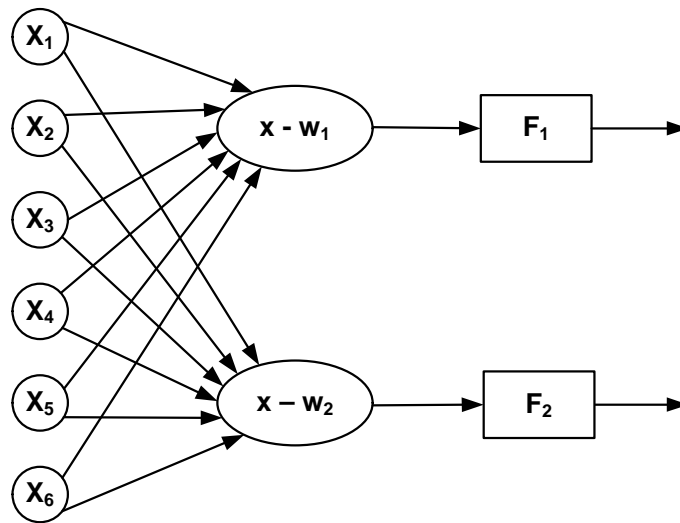
### **II.2.2.1. Learning Vector Quantization**

*LVQ* merupakan metode pelatihan pada lapisan kompetitif terawasi yang

akan belajar secara otomatis untuk mengklasifikasikan vektor-vektor input ke dalam kelas-kelas tertentu. Kelas yang dihasilkan tergantung pada jarak antara vektor-vektor input. Jika ada 2 vektor input yang nilainya hampir sama maka lapisan kompetitif akan mengklasifikasikan kedua vektor input tersebut ke dalam kelas yang sama (Sutojo et al, 2010).

Arsitektur *LVQ* juga terdiri dari 2 lapisan, input (X) dan output (Y), dimana antara lapisannya dihubungkan oleh bobot tertentu yang disebut sebagai vektor perwakilan (W), sama halnya dengan metode Self Organizing Map (SOM) yang sebelumnya juga dikembangkan oleh Teuvo Kohonen. Pada saat pembelajaran, informasi yang diberikan ke jaringan bukan hanya vektor data saja melainkan informasi kelas/target dari data juga ikut dimasukkan (Ginting, 2015).

Pada *LVQ* standar, umumnya cara menentukan vektor bobot awal tersebut biasanya menggunakan vektor yang diambil dari data yaitu data pertama dan data kedua dengan kelas yang berbeda apabila hanya terdapat dua kelas dalam sekumpulan data kasus. Namun apabila dua vektor bobot tersebut memiliki nilai yang hampir sama, maka akan mengalami proses pembelajaran yang lama untuk mengenali data pada setiap kelas. Untuk contoh arsitektur *JST LVQ* dapat dilihat pada Gambar 2.1.



Gambar 2.1 Arsitektur JST LVQ (Kusumadewi, 2003)

Secara umum, algoritma LVQ adalah sebagai berikut (Safwandi, 2016) :

1. Langkah pertama adalah menentukan bobot dari data setiap kelas, menetapkan *learning rate* ( ) dan penurunan *learning rate*.
2. Bandingkan masing-masing input dengan masing-masing bobot yang telah ditetapkan dengan melakukan pengukuran jarak antara masing-masing bobot  $w_0$  dan input  $x_p$ , persamaannya sebagai berikut :

$$\| x_p - w_0 \|$$

3. Nilai minimum dari hasil perbandingan itu akan menentukan kelas dari vektor input dan perubahan bobot dari kelas tersebut. Perubahan untuk bobot baru ( $w_0'$ ) dapat dihitung dengan persamaan berikut :

- Untuk input dan bobot yang memiliki kelas yang sama :

$$w_0 = w_0 + \alpha (x - w_0)$$

- Untuk input dan bobot yang memiliki kelas yang berbeda :

$$w_0 = w_0 - \alpha (x - w_0)$$

Pada dasarnya perhitungan diatas akan dilakukan terus-menerus sampai nilai bobot tidak berubah jika ada input baru. Hal ini tentu saja membutuhkan keperluan memori yang sangat besar untuk perhitungan. Untuk itu, dalam melakukan perhitungan *LVQ* bisa ditentukan maksimal perulangan (*epoch*). Setelah dilakukan pelatihan, akan diperoleh bobot akhir (*W*). Bobot ini nantinya akan digunakan untuk melakukan simulasi atau pengujian terhadap data yang lain.

### **II.3. Pemodelan Sistem**

Pemodelan secara umum, dimengerti sebagai proses merepresentasikan objek nyata atau realita sebagai seperangkat persamaan matematika, grafis ataupun bagan agar mudah dipahami oleh pihak yang berkepentingan. Lebih khusus lagi, istilah ini sering digunakan untuk proses menggambarkan konsep yang mewakili obyek-obyek dalam pengembangan sistem informasi. Pemodelan dalam pengembangan sistem informasi, berevolusi selaras perkembangan teknologi dan metodologi pengembangannya. Kesamaan pandang diperlukan dalam pemodelan, yang akhirnya menuntut adanya keseragaman atau standar. (Aliq Zuhdi, 2017)

Teknologi obyek, membutuhkan pemodelan yang relevan, telah mendorong Jacobson dkk., menciptakan standar pemodelan untuk pengembangan sistem, dengan pendekatan obyek yang dikenal dengan UML (Unified Modeling Language). UML memiliki keuntungan yang menghasilkan representasi yang dapat diverifikasi melalui penalaran logis, pengujian, atau bahkan simulasi. Sebagai contoh, diagram proses bisnis dapat diuji dengan mental melewati kasus

imajinasi , melalui pemikiran logis yang mungkin berbeda untuk melihat apakah itu merupakan upaya menghasilkan *output* yang diperlukan.

### **II.3.1. Data Flow Diagram (DFD)**

Data Flow Diagram (DFD) disebut juga dengan Diagram Arus Data (DAD). DFD adalah: suatu model logika data atau proses yang dibuat untuk menggambarkan: darimana asal data, dan kemana tujuan data yang keluar dari sistem, dimana data disimpan, proses apa yang menghasilkan data tersebut, dan interaksi antara data yang tersimpan, dan proses yang dikenakan pada data tersebut (Rita, 2014)

Ada empat buah simbol pada DFD, yang masing-masingnya digunakan untuk mewakili antara lain (Rita, 2014):

- a. *External entity* (kesatuan luar) atau boundary (batas sistem), digunakan untuk menyatakan: suatu kantor, departemen atau divisi dalam perusahaan tetapi di luar sistem yang dikembangkan; orang atau sekelompok orang di organisasi tetapi di luar sistem yang sedang dikembangkan; suatu organisasi atau orang yang berada di luar organisasi misal: langganan, pemasok; sistem informasi yang lain di luar sistem yang sedang dikembangkan; sumber asli suatu transaksi; penerima akhir dari suatu laporan yang dihasilkan oleh sistem.
- b. *Data flow* (arus data), digunakan untuk menunjukkan arus dari data yang dapat berupa: masukan untuk sistem ataupun hasil dari proses system. Arus data sebaiknya diberi nama yang jelas dan mempunyai arti.

- c. *Process* (proses), digunakan untuk menunjukkan kegiatan atau kerja yang dilakukan oleh orang, mesin atau komputer dari hasil suatu arus data yang masuk ke dalam proses untuk dihasilkan arus data yang keluar dari proses. Suatu proses harus menerima arus data dan menghasilkan arus data.
- d. *Data store* (simpanan data) ), digunakan untuk menunjukkan simpanan dari data yang dapat berupa: suatu file atau database di sistem komputer; suatu arsip atau catatan manual; suatu kotak tempat data di meja seseorang; suatu tabel acuan manual; suatu agenda atau buku.

### **II.3.2. Unified Modelling Language (UML)**

Menurut Munawar (2005), UML (*Unified Modeling Language*) adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi obyek. Hal ini disebabkan karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembangan sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan mereka dengan yang lain.

UML sendiri terdiri atas pengelompokkan diagram-diagram sistem menurut aspek atau sudut pandang tertentu. Diagram adalah yang menggambarkan permasalahan maupun solusi dari permasalahan suatu model. Semakin kompleks bentuk sistem yang akan dibuat, maka semakin sulit komunikasi antara orang-orang yang saling terkait dalam pembuatan dan pengembangan software yang akan dibuat. Berikut ini adalah beberapa diagram UML (Rosa A. S, M. Shalahuddin; 2014) :

1. Use Case Diagram
2. Class Diagram
3. Activity Diagram

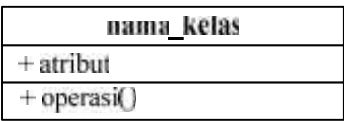
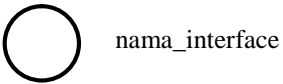

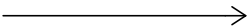
### II.3.2.1. Class Diagram

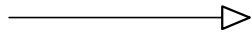
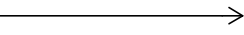
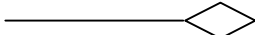
Diagram kelas atau *Class* diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. (Sonya, 2016)

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

Berikut adalah simbol-simbol yang ada pada diagram kelas :

**Tabel II.1. Simbol-Simbol Diagram Kelas**

Simbol	Deskripsi
<p>Kelas</p> 	Kelas pada struktur sistem
<p>Antarmuka / <i>interface</i></p> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
<p>Asosiasi / <i>association</i></p> 	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
<p>Asosiasi berarah / <i>directed association</i></p> 	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>

Generalisasi / <i>Generalization</i> 	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum-khusus)
Kebergantungan / <i>dependency</i> 	Relasi antarkelas dengan makna kebergantungan antarkelas
Agregasi / <i>aggregation</i> 	Relasi antarkelas dengan makna semua bagian ( <i>whole-part</i> )
1	Satu dan hanya satu
0.....*	Boleh tidak ada atau 1 atau lebih
1.....*	1 atau lebih
0.....1	Boleh tidak ada, maksimal 1
n.....n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

(Sumber : Sonya M, 2016)

### II.3.2.2. Use Case Diagram

*Use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (behaviour) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. (Sonya, 2016)

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 *use case*.

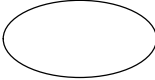
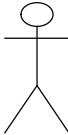

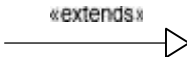

- Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan

dibuat itu sendiri, jadi walaupun simbol aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.

- *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

Berikut adalah simbol-simbol yang ada pada diagram *use case* :

**Tabel II.2. Simbol-simbol Diagram *Use Case***

Simbol	Keterangan
<p><i>Use Case</i></p> 	<p>Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i></p>
<p>Aktor / Actor</p> 	<p>Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda diawal frase nama actor</p>
<p>Asosiasi / Association</p> 	<p>Komuniikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan actor</p>
<p>Ekstensi / Extends</p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan</p>
<p>Generalisasi / Generalization</p> 	<p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.</p>
<p>Menggunakan / Include / Uses</p>	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> di mana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini</p>








(Sumber : Sonya M, 2016)

### II.3.2.3. Activity Diagram

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Berikut adalah simbol-simbol yang ada pada diagram aktivitas :

**Tabel II.3. Simbol-simbol Diagram Aktivitas**

Simbol	Keterangan
Status Awal 	Status awal aktivitas sistem sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan / <i>Decision</i> 	<i>Decision</i> , atau pilihan untuk mengambil keputusan.
Penggabungan / <i>Join</i> 	Arah tanda panah alur proses.
Status Akhir 	Titik akhir atau akhir dari aktivitas.

(Sumber : Sonya M, 2016)

## **II.4. Pemrograman**

Pemrograman adalah proses menulis, menguji dan memperbaiki (*debug*), dan memelihara kode yang membangun suatu program komputer. Kode ini ditulis dalam berbagai bahasa pemrograman. Tujuan dari pemrograman adalah untuk memuat suatu program yang dapat melakukan suatu perhitungan atau 'pekerjaan' sesuai dengan keinginan si pemrogram. Untuk melakukan pemrograman, diperlukan keterampilan dalam algoritma, logika, bahasa pemrograman, dan pada banyak kasus, pengetahuan-pengetahuan lain seperti matematika. (Ritayani, 2016)

Pemrograman adalah suatu seni dalam menggunakan satu atau lebih algoritma yang saling berhubungan dengan menggunakan suatu bahasa pemrograman tertentu sehingga menjadi suatu program komputer. Bahasa pemrograman yang berbeda mendukung gaya pemrograman yang berbeda pula. Gaya pemrograman ini biasa disebut paradigma pemrograman. (Saniman & Fathoni, 2008)

Dalam mengimplementasikan suatu bahasa pemrograman dibutuhkan suatu wadah aplikasi yang biasa disebut dengan Integrated Development Environment (IDE). IDE sendiri berfungsi sebagai tempat para programmer mengedit, meng-*compile*, membuat *link* hingga menjalankan kode program sehingga menghasilkan aplikasi yang diharapkan oleh programmer.

### **II.4.1. IDE (Integrated Development Environment)**

IDE (Integrated Development Environment) adalah program komputer yang memiliki beberapa fasilitas yang diperlukan dalam pembangunan perangkat

lunak. Tujuan dari IDE adalah untuk menyediakan semua utilitas yang diperlukan dalam membangun perangkat lunak.

Sebuah IDE, atau secara bebas dapat diterjemahkan sebagai Lingkungan Pengembangan Terpadu, setidaknya memiliki fasilitas:

- *Editor*, yaitu fasilitas untuk menuliskan kode sumber dari perangkat lunak.
- *Compiler*, yaitu fasilitas untuk mengecek sintaks dari kode sumber kemudian mengubah dalam bentuk binari yang sesuai dengan bahasa mesin.
- *Linker*, yaitu fasilitas untuk menyatukan data binari yang beberapa kode sumber yang dihasilkan compiler sehingga data-data binari tersebut menjadi satu kesatuan dan menjadi suatu program komputer yang siap dieksekusi.
- *Debugger*, yaitu fasilitas untuk mengetes jalannya program, untuk mencari *bug*/kesalahan yang terdapat dalam program.

Sampai tahap tertentu IDE modern dapat membantu memberikan saran yang mempercepat penulisan. Pada saat penulisan kode, IDE juga dapat menunjukkan bagian-bagian yang jelas mengandung kesalahan atau keraguan.

#### **II.4.1.1. Netbeans**

NetBeans adalah sebuah open source lingkungan pengembangan terpadu (IDE) untuk mengembangkan dengan Java, PHP, C ++, dan bahasa pemrograman lainnya. NetBeans adalah juga disebut sebagai platform komponen modular digunakan untuk mengembangkan aplikasi Java desktop (Angga Wijaya, 2013).

NetBeans mengelola fitur platform berikut dan komponen:

- Pengaturan pengguna
- Jendela (penempatan, penampilan, dll)
- NetBeans Visual Perpustakaan
- Penyimpanan
- Terpadu alat pengembangan
- Kerangka Wizard

NetBeans menggunakan komponen, juga dikenal sebagai modul, untuk memungkinkan pengembangan perangkat lunak. NetBeans menginstal modul secara dinamis dan memungkinkan pengguna untuk men-download fitur update dan upgrade digital dikonfirmasi.

#### **II.4.1.2. Visual Studio**

Visual Studio merupakan sebuah aplikasi untuk bahasa pemrograman Visual Basic yang menawarkan Integrated Development Environment (IDE) visual untuk membuat program perangkat lunak/ aplikasi berbasis sistem operasi Microsoft Windows yang berbasis GUI (Graphical User Interface).

##### **1. Kemampuan VB**

Beberapa kemampuan atau manfaat dari Visual Studio, diantaranya:

1. Untuk membuat program aplikasi berbasis Windows.
2. Untuk membuat *ActiveX*, Aplikasi Internet dan lain sebagainya.
3. Menguji program (debugging) dan menghasilkan program akhir berakhiran EXE yang bersifat executable atau dapat langsung dijalankan.

## 2. Control Menu

Control menu adalah menu yang digunakan terutama untuk memanipulasi jendela Microsoft Visual Studio. Dari menu kita bisa mengubah ukuran, memindahkan, atau menutup jendela Microsoft Visual Studio atau jendela windows lainnya. Control menu tersebut terdiri dari Restore, Move, Size, Minimize, Maximize, dan Close. Restore berfungsi untuk mengatur ukuran jendela visual basic, Move untuk memindahkan jendela pada visual basic, Size untuk merubah ukuran pada visual basic, Minimize untuk mengubah ukuran jendela visual basic ke ukuran minimal, Maximize untuk memaksimalkan ukuran jendela visual basic sehingga tampilannya memenuhi layar.

## 3. Menu

Menu adalah batang menu yang terletak di bawah title bar yang berfungsi untuk menampilkan pilihan menu / perintah untuk mengoperasikan program visual basic.

## 4. Toolbar

Toolbar merupakan batang yang berisi kumpulan menu yang terletak dibagian bawah menu berfungsi untuk menjalankan suatu perintah pada kondisi default. Cara menggunakannya klik kanan pada toolbar dan pilih salah satu menu yang ingin digunakan

## 5. Form Window

Form window atau jendela window adalah daerah kerja utama, dimana kita dapat membuat program-program aplikasi visual basic. Pada form ini kita

merancang teks, gambar, tombol-tombol perintah, scrollbar, dan sebagainya. Jendela form ini pada awalnya kelihatan kecil, tetapi ukurannya bisa diubah-ubah sesuai dengan kebutuhan aplikasi yang kita buat.

#### 6. Toolbox

Toolbox adalah sebuah kotak piranti yang mengandung semua objek atau control yang dibutuhkan untuk membuat sebuah program aplikasi.

#### 7. Project Explorer

Project Explorer berfungsi berbagai saran pengakses bagian-bagian pembentuk project. Pada windows ini terdapat tiga tombol pengaktif untuk Windows Code, Windows Object dan Toggle Folder. Juga terdapat diagram yang menampilkan susunan folder penyimpanan file-file project

#### 8. Jendela Properties

Jendela properties adalah jendela yang mengandung semua informasi mengenai objek yang terdapat pada aplikasi microsoft visual basic. Properti adalah sifat sebuah objek, misalnya seperti namanya, warna, ukuran, posisi, dan sebagainya

#### 9. Form Layout Window

Form layout window adalah jendela yang menggambarkan posisi dari form yang ditampilkan pada layar monitor. Posisi form pada form layout window inilah yang merupakan petunjuk dimana aplikasi yang dibuat akan ditampilkan

#### 10. Jendela Code

Jendela code adalah salah satu jendela yang penting di dalam microsoft visual basic. Jendela ini berisi kode-kode program yang merupakan instruksi-instruksi untuk aplikasi visual basic yang dibuat. Setiap objek pada visual basic dapat ditambahkan dengan kode-kode program untuk melakukan tugas-tugas tertentu seperti menutup aplikasi, membatalkan perintah, dan sebagainya.

## **II.4.2. Bahasa Pemrograman**

Bahasa pemrograman, atau sering diistilahkan juga dengan bahasa komputer atau bahasa pemrograman komputer, adalah instruksi standar untuk memerintah komputer. Bahasa pemrograman ini merupakan suatu himpunan dari aturan sintaks dan semantik yang dipakai untuk mendefinisikan program komputer. Bahasa ini memungkinkan seorang programmer dapat menentukan secara persis data mana yang akan diolah oleh komputer, bagaimana data ini akan disimpan/diteruskan, dan jenis langkah apa secara persis yang akan diambil dalam berbagai situasi (Rita, 2014).

### **II.4.2.1. Java**

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer termasuk telepon genggam. Bahasa ini awalnya dibuat oleh James Gosling saat masih bergabung di Sun Microsystems saat ini merupakan bagian dari Oracle dan dirilis tahun 1995. Bahasa ini banyak mengadopsi sintaksis yang terdapat pada C dan C++ namun dengan sintaksis model objek yang lebih sederhana serta dukungan rutin-rutin aras bawah yang minimal. Aplikasi-aplikasi berbasis java umumnya dikompilasi ke dalam p-code (*bytecode*) dan dapat

dijalankan pada berbagai Mesin Virtual Java (JVM). Java merupakan bahasa pemrograman yang bersifat umum/non-spesifik (*general purpose*), dan secara khusus didisain untuk memanfaatkan dependensi implementasi seminimal mungkin. Karena fungsionalitasnya yang memungkinkan aplikasi java mampu berjalan di beberapa platform sistem operasi yang berbeda, java dikenal pula dengan slogannya, "*Tulis sekali, jalankan di mana pun*". Saat ini java merupakan bahasa pemrograman yang paling populer digunakan, dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi ataupun aplikasi.

#### **II.4.2.1. Visual Basic**

Visual Basic adalah salah satu bahasa pemrograman komputer. Bahasa pemrograman adalah perintah yang dimengerti oleh komputer untuk melakukan tugas-tugas tertentu. Bahasa pemrograman Visual Basic, yang dikembangkan oleh Microsoft sejak tahun 1991, merupakan pengembangan dari pendahulunya yaitu bahasa pemrograman BASIC (Beginner's All-purpose Symbolic Instruction Code) yang dikembangkan pada era 1950-an. Visual Basic merupakan salah satu Development Tool yaitu alat bantu untuk membuat berbagai macam program komputer, khususnya yang menggunakan sistem operasi Windows. Visual Basic merupakan salah satu bahasa pemrograman komputer yang mendukung object (Object Oriented Programming = OOP) (Riduwan, 2015).

Secara historis, aplikasi adalah software yang dikembangkan oleh sebuah perusahaan. Bahasa pemrograman Visual Basic dapat digunakan untuk menyusun dan membuat program aplikasi pada sistem operasi windows. Program aplikasi

dapat berupa program database, program grafis dan lain sebagainya. Didalam Visual Basic terdapat komponen-komponen yang sangat membantu dalam pembuatan program aplikasi. Dalam pembuatan program aplikasi pada Visual Basic dapat didukung oleh software seperti Microsoft Access, Microsoft Exel, Seagate Crystal Report, dan lain sebagainya. Untuk dapat menyusun dan membuat suatu program aplikasi dari VB, tentunya harus mengetahui fasilitas–fasilitas yang disediakan agar proses penyusunan dan pembuatan program tersebut berjalan dengan baik. (Djoni H, 2013)

## **II.5. Database**

Banyak sekali definisi tentang database yang diberikan oleh para pakar dibidang ini. Database terdiri dari dua penggalan kata yaitu data dan base, yang artinya berbasiskan pada data. Tetapi secara konseptual, database diartikan sebuah koleksi atau kumpulan data yang saling berhubungan (relation), disusun menurut aturan tertentu secara logis, sehingga menghasilkan informasi. Sebuah informasi yang berdiri sendiri tidaklah dikatakan database. (Sonya, 2016)

Sebagai contoh, nomor telepon seorang pelanggan, disimpan dalam banyak tempat apakah itu di file pelanggan, di file alamat dan di lokasi yang lain. Antara file yang satu dengan file yang lainnya tidak saling berhubungan, sehingga apabila salah seorang pelanggan berganti nomor telepon dan anda hanya mengganti di file pelanggan saja, akibatnya akan terjadi ketidakcocokan data, karena di lokasi yang lain masih tersimpan data telepon yang lama. (Sonya, 2016)

Dalam sistem database hal ini tidak boleh dan tidak bisa terjadi, karena antara file yang satu dengan file yang lain saling berhubungan. Jika suatu data yang sama anda ubah, data tersebut di file yang lain akan otomatis berubah juga. Sehingga mampu menjadi informasi yang diinginkan dan dapat dilakukan proses pengambilan, penghapusan, pengeditan, terhadap data secara mudah dan cepat (Efektif, Efisien dan Akurat).

Data adalah fakta, baik berupa sebuah objek, orang dan lain – lain yang dapat dinyatakan dengan suatu nilai tertentu (angka, simbol, karakter tertentu, dan lain – lain). Sedangkan informasi adalah data yang telah diolah sehingga bernilai guna dan dapat dijadikan bahan dalam pengambilan keputusan, (Yuhefizard , 2010).

Hubungan data dan informasi dapat digambarkan sebagai berikut :

**Data** —————> **Proses** —————> **Informasi**

**Gambar II.1. Data dan Informasi.**

**(Sumber : Yuhefizard , 2010)**

### **II.5.1. My SQL**

MySQL adalah sebuah implementasi dari sistem manajemen basisdata relasional (RDBMS) yang didistribusikan secara gratis. Setiap pengguna dapat secara bebas menggunakan MySQL, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam basisdata yang

telah ada sebelumnya; SQL (Structured Query Language). SQL adalah sebuah konsep pengoperasian basisdata, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. (Rio, 2014)

### **II.5.2. SQL Server**

SQL Server 2008 adalah sebuah RDBMS (*Relational Database Management System*) yang dibangun oleh *Microsoft*, yang digunakan untuk menyimpan dan mengolah data. Pada SQL Server 2008, kita bisa melakukan pengambilan dan modifikasi data yang ada dengan cepat dan efisien. Pada SQL Server 2008, kita bisa membuat *object – object* yang sering digunakan pada aplikasi bisnis, seperti membuat *database, table, fuction, stored procedure, trigger* dan *view*. Selain *object*, kita juga menjalankan perintah SQL (*Structured Query Language*) untuk mengambil data (Riduan, 2015)

Adapun beberapa dari kelebihan dan kekurangan dari SQL Server sebagai berikut :

#### **1. Kelebihan Microsoft SQL Server**

- Dengan kemampuannya untuk mengolah data yang besar maka DBMS ini sangat cocok untuk perusahaan mikro, menengah hingga perusahaan besar sekalipun.
- DBMS jenis memiliki kelebihan memanage user data serta masing-masing user dapat diatur hak aksesnya terhadap pengaksesan data base oleh DBA.
- Mempunyai tingkat keamanan data yang sangat baik.

- Dapat melakukan back up, recovery dan rollback data dengan mudah.
- Mempunyai kelebihan untuk membuat data base mirroring dan clustering.

## 2. Kekurangan Microsoft SQL Server

- DBMS jenis ini hanya dapat berjalan pada sistem operasi / platform windows saja.
- Software ini mempunyai lisensi dari microsoft sehingga pemakaiannya membutuhkan biaya yang cukup mahal.