

## **BAB II**

### **LANDASAN TEORI**

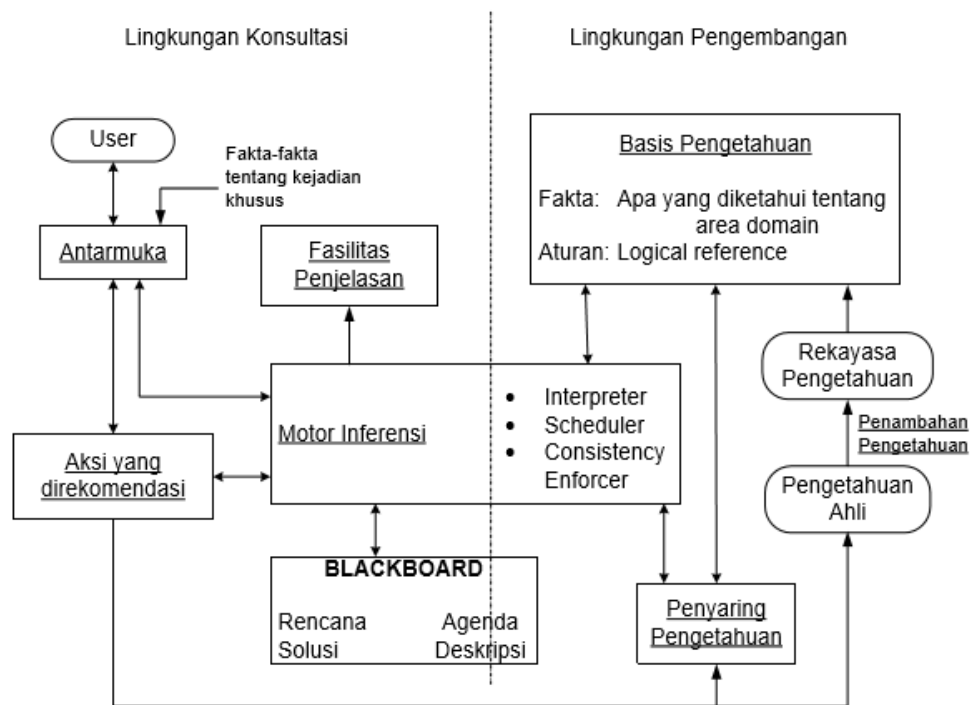
#### **2.1. Sistem Pakar**

Secara umum, sistem pakar adalah sistem yang mengadopsi pengetahuan manusia ke komputer agar dapat menyelesaikan masalah seperti yang biasa dilakukan oleh para ahli. Dengan sistem pakar ini, orang awampun dapat menyelesaikan masalah yang cukup rumit yang sebenarnya hanya dapat diselesaikan dengan bantuan para ahli. Bagi para ahli, sistem pakar ini juga akan membantu aktivitasnya sebagai asisten yang sangat berpengalaman (Kusumadewi, 2003).

Sistem pakar memiliki banyak manfaat yang dapat diambil yakni :

- a. Menyimpan pengetahuan dan keahlian para pakar.
- b. Dapat melakukan proses secara berulang secara otomatis.
- c. Meningkatkan *output* dan produktifitas.
- d. Meningkatkan kualitas.

Selain memiliki manfaat, sistem pakar juga memiliki kelemahan yakni biaya yang dibutuhkan untuk membuat dan memeliharanya sangat mahal dan sulit dikembangkan. Sistem pakar memiliki dua bagian pokok yakni lingkungan pengembangan (*development environment*) yakni untuk memasukkan pengetahuan pakar ke dalam lingkungan sistem pakar dan lingkungan konsultasi (*consultation environment*) yang digunakan oleh pengguna yang bukan pakar guna memperoleh pengetahuan pakar (Arhami, 2004).



Gambar 2.1. Arsitektur Sistem Pakar  
(Sumber: Arhami, 2004)

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

UML (*Unified Modeling Language*) adalah sebuah Bahasa yang berdasarkan grafik atau gambar untuk memvisualisasi, menspesifikasikan, membangun, dan pendokumentasian dari sebuah sistem pengembangan software berbasis OO (*Object-Oriented*).

### 2.2.1. Use Case Diagram

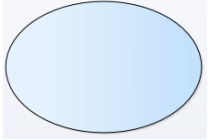
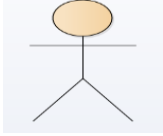
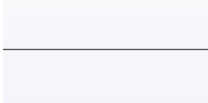
Merupakan model dalam diagram (*Unified Modeling Language UML*) yang digunakan untuk menggambarkan requirement fungsional yang diharapkan dari sebuah sistem.

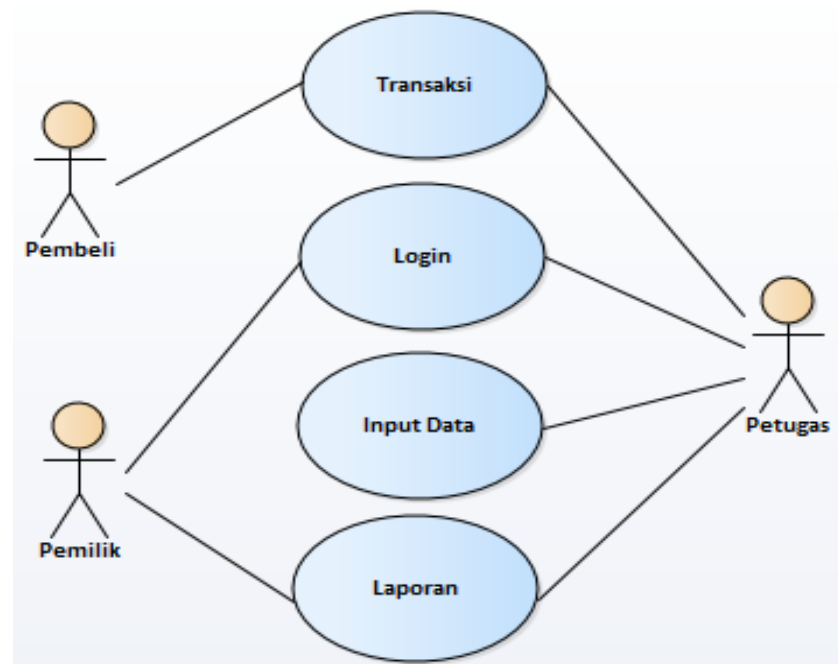
*Use case diagram* menekankan pada “siapa” melakukan “apa” dalam lingkungan sistem perangkat lunak yang akan dibangun.

Tujuan utama dari sebuah diagram *use case* adalah untuk

menunjukkan apa yang dilakukan fungsi sistem dan aktor. Peran aktor dalam sistem dapat digambarkan.

Tabel 2.1. Komponen *Use Case Diagram*

Komponen	Keterangan	Simbol
<i>Use Case</i>	<i>Use case</i>  Menunjukkan proses yang terjadi pada sistem baru	
<i>Actor</i>	<i>Actor</i>  Menunjukkan user yang akan menggunakan sistem baru	
<i>Association</i>	<i>Unidirectional Association</i>  Menunjukkan hubungan antara <i>actor</i> dengan <i>use case</i> atau antara <i>use case</i>	



Gambar 2.2. Contoh *Use Case Diagram*

Diagram *use case* di atas ada 3 Aktor yang masing-masing mempunyai hak sebagai berikut :

#### 1. Petugas

Login : untuk dapat melakukan proses-proses lainnya, maka terlebih dahulu, petugas harus masuk ke dalam sistem melalui validasi *user id* dan *password*.

Input data : pada proses ini, petugas dapat melakukan penginputan, pengeditan, dan penghapusan data.

Laporan : pada proses ini, petugas dapat melakukan pembuatan laporan data yang kemudian laporan akan di serahkan kepada pemilik.

Transaksi : pada proses ini, petugas dapat melakukan penginputan, dan penghapusan data transaksi yang akan tersimpan pada tabel penjualan serta dapat melakukan proses pembayaran dengan rincian data barang yang di beli customer yang kemudian akan di cetak menjadi sebuah bukti pembayaran.

#### 2. Pembeli

Di dalam sistem ini, pembeli tidak terlibat langsung di dalam sistem. Pembeli hanya dapat mengetahui informasi tentang barang yang akan dibeli dan transaksi atau bukti pembayaran.






#### 3. Pemilik

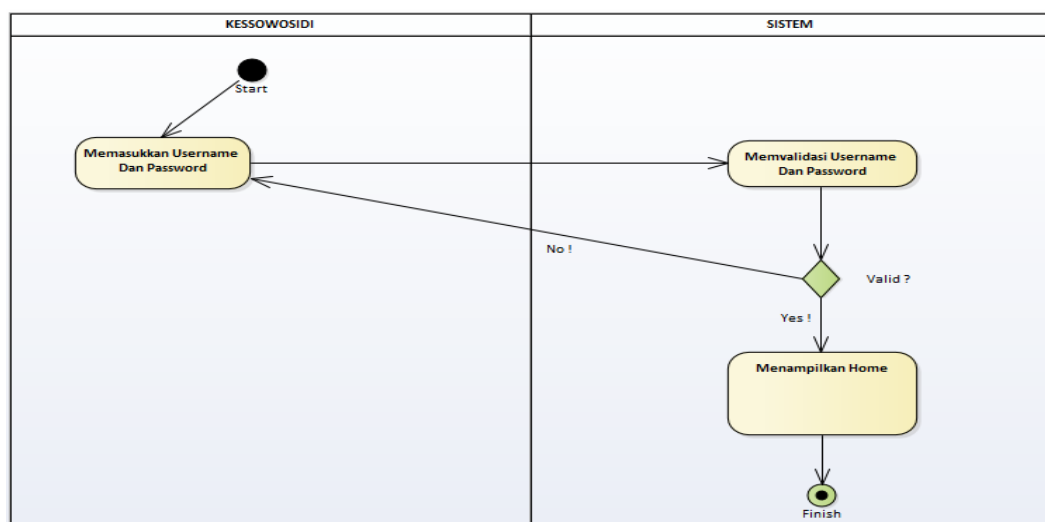
Pemilik merupakan orang yang memiliki perusahaan tersebut pemilik juga dapat melakukan proses login seperti petugas dan dapat melihat laporan data penjualan secara langsung.

### 2.2.2. Activity Diagram

Diagram ini mengandung aktivitas, pilihan tidak, perulangan dan hasil dari aktivitas tersebut.

Tabel 2.2. Activity Diagram

Komponen	Keterangan	Simbol
<i>Partition</i>	Menunjukkan siapa yang melakukan aktivitas dalam suatu diagram	
<i>Start state</i>	Menunjukkan dimana aliran kerja itu dimulai	
<i>End state</i>	Menunjukkan dimana aliran kerja itu berakhir	
<i>Action state</i>	Adalah langkah-langkah dalam sebuah <i>activity</i> . <i>Action</i> bisa terjadi saat memasuki <i>activity</i>	
<i>Decision</i>	Pilihan untuk pengambilan keputusan	








Gambar 2.3. Contoh Activity Diagram

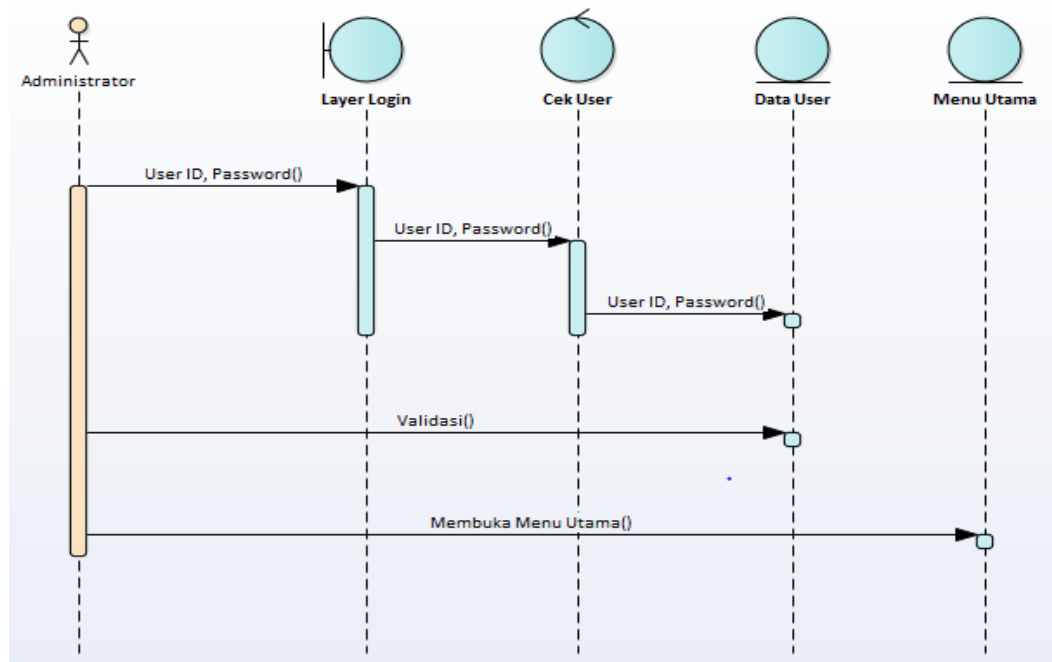
Diagram *activity* di atas menjelaskan *user* dan sistem saling berinteraksi dimulai dengan *user* memasukkan *username* dan *password* kedalam sistem. Kemudian sistem menerima dan memvalidasi inputan berupa *username* dan *password*, apabila inputan tersebut benar maka sistem akan menampilkan halaman menu utama. Apabila salah maka *user* dapat menginputkan *username* dan *password* lagi sampai benar.

### 2.2.3. *Sequence Diagram*

Merupakan suatu diagram yang memperlihatkan antara menampilkan interaksi – interaksi antar objek didalam sistem yang disusun pada sebuah urutan atau rangkaian waktu.

Tabel 2.3. Komponen *Sequence Diagram*

Komponen	Keterangan	Simbol
<i>Actor</i>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi	
<i>Boudary</i>	Diagram yang menjelaskan tampilan dan membatasi sistem dengan dunia luar	
<i>Control</i>	Diagram yang menjelaskan proses didalam program	
<i>Entity</i>	Diagram yang menjelaskan tabel pada <i>database</i> program	
<i>Message</i>	Digunakan untuk memanggil operasi atau metode yang dimiliki suatu objek	



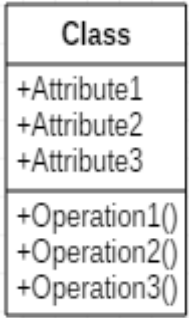

Gambar 2.4. Contoh *Sequence Diagram*

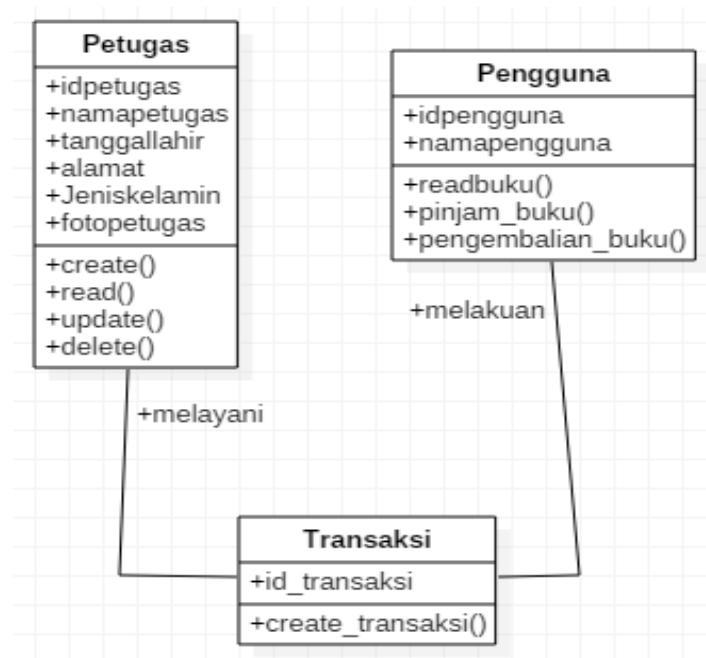
*Sequence diagram* di atas menjelaskan administrator dan 3 objek, yaitu : layar login, cek user, data user dan menu utama. Pertama administrator akan masuk ke layar login dengan menggunakan *user id* dan *password*. Dari layar login, admin akan melakukan cek *user* dengan memasukkan *user id* dan *password*. Setelah melakukan cek *user*, admin akan memasukkan *user id* dan *password* sekali lagi untuk melihat data *user*. *User id* dan *password* yang dimasukkan admin sebanyak 3 kali, digunakan melakukan validasi. Validasi ini bertujuan untuk membuka menu utama

#### 2.2.4. *Class Diagram*

*Class Diagram* digunakan untuk menampilkan kelas-kelas dan paket-paket di dalam sistem. *Class diagram* memberikan gambaran sistem secara statis dan relasi antar mereka. Biasanya, dibuat beberapa class diagram untuk sistem tunggal.

Tabel 2.4. komponen *Class Diagram*

Komponen	Keterangan	Simbol
<i>Class</i>	<i>Class</i> adalah blok-blok pembangunan pada pemrograman berorientasi obyek. Sebuah <i>class</i> digambarkan sebagai sebuah kotak yang terbagi atas 3 bagian. Bagian atas adalah bagian nama dari <i>class</i> . Bagian tengah mendefinisikan property/atribut <i>class</i> . Bagian akhir mendefinisikan <i>method-method</i> dari sebuah <i>class</i> .	
<i>Association</i>	Sebuah asosiasi merupakan sebuah relationship paling umum antara 2 <i>class</i> dan dilambangkan oleh sebuah garis yang menghubungkan antara 2 <i>class</i> . (contoh : <i>One-to-one</i> , <i>one-to-many</i> , <i>many-to-many</i> ).	

Gambar 2.5. Contoh *Class Diagram*



*Class diagram* di atas menjelaskan petugas mempunyai atribut `idpetugas`, `namapetugas`, `tanggallahir`, `alamat`, `jeniskelamin`, serta `fotopetugas`. Operasi apa yang bisa dilakukan petugas di sini yaitu menulis, membaca, memperbaharui, dan menghapus. Petugas diberikan asosiasi antara petugas dan transaksi sehingga petugas bisa melakukan transaksi untuk melayani pengguna. Kemudian diberikan asosiasi antara transaksi dan pengguna sehingga pengguna bisa melakukan transaksi.

## 2.3. Hama Dan Penyakit Pada Tanaman Melon

### 2.3.1. Hama

Hama merupakan organisme yang merusak tanaman dan secara ekonomi merugikan manusia. Cara kerja hama bermacam-macam mulai dari memakan bagian tanaman (akar, batang, daun, bunga dan buah) sampai menghisap cairan dari jaringan tanaman tersebut. Hama sering kali menyerang sejak penanaman benih, sampai buah siap panen. Serangan tersebut dapat mengganggu pertumbuhan dan perkembangan tanaman bahkan meyebabkan gagal panen sehingga merugikan petani.

#### 1. Thrips



Gambar 2.6. Thrips

a. Bagian yang diserang :

Daun muda, daun tua, bunga, dan buah.

b. Ciri dan gejala :

Daun muda dan tunas menjadi keriting, daun tua yang terserang menjadi kering, bunga rontok dan buah abnormal, tanaman menjadi kerdil.

c. Pengendalian :

Menjaga kebersihan lingkungan, memangkas bagian tanaman yang terserang, penyemprotan pestisida alami yang terbuat dari air rendaman biji sirsak kering dan dihaluskan, penyemprotan pestisida alami yang terbuat dari air rendaman biji sirsak kering dan dihaluskan, penyemprotan dilakukan 14 hari, menyemprotkan insektisida berbahan aktif *dimetoate* 400 g/l, *sipermetrin* 30, 36 g/l atau *tetasipermetrin* 30, 36 g/l.

2. Kutu daun atau lalat daun (*Aphids gossypii*)



Gambar 2.7. Kutu daun

a. Bagian yang diserang :

Pucuk dan daun melon.

b. Ciri dan gejala :

Daun menggulung, pucuk tanaman keriting karena cairan daunnya dihisap oleh aphids, pembetukan bunga terhambat.

c. Pengendalian :

Memangkas bagian tanaman yang terserang aphids kemudian membuangnya jauh dari area pertanaman, memasang perangkap seperti *yellow sticky trap* yang terbuat dari *impra board* yang kedua sisinya dilapisi oleh cairan oli, menyemprotkan insektisida berbahan aktif *tetasipermetrin-30* 36 g/l atau *dimethoate perfekthion* 400 EC (1-2ml/l).

3. Penggorok daun (*leaf miner*)



Gambar 2.8. Penggorok daun

a. Bagian yang diserang :

Daun (lalat betina meletakkan telurnya ke dalam jaringan daun).

b. Ciri dan gejala :

Terlihat alur-alur tak beraturan berwarna putih pada daun (akibat larva yang menggorok jaringan daun), Bila terjadi serangan hebat, daun akan tampak putih, karena yang tersisa hanya lapisan tipis bagian luarnya saja.

c. Pengendalian :

Memangkas daun yang terserang kemudian memusnahkannya jauh dari area penanaman, penyemprotan insektisida sistemik, seperti *Trigard* (*siramazin* 75%) untuk membunuh larva telur

yang berada di jaringan tanaman, minimal 14 hari sekali supaya siklusnya terputus, menyemprotkan *Curacron 500 EC* untuk mengendalikan *leaf miner* stadia dewasa, penyemprotan dilakukan pada seluruh bagian tanaman sesuai dengan dosis anjuran setiap 14 hari sekali.

#### 4. Lalat buah



Gambar 2.9. Lalat buah

Serangan hama ini dimulai dengan peletakan telur pada buah oleh lalat betina, beberapa hari kemudian telur akan menetas menjadi larva yang disebut belatung atau tempayak. Larvanya sangat merusak buah. Terdapat dua jenis lalat buah yang biasa menyerang melon, yaitu *Dacus* sp, dan *Bacrocera cucurbitae*.

##### a. Bagian yang diserang :

Buah yang menjelang matang.

##### b. Ciri dan gejala :

Penampilan buah tampak sehat dari luar namun daging buahnya busuk dan mengandung larva (belatung), buah busuk dan rontok (serangan berat), buah berwarna kehitaman dan keras (buah muda), timbul bercak bulat membusuk dan berlubang kecil, buah rusak dan rontok.

##### c. Pengendalian :

Memetik dan memusnahkan buah yang terserang, menanam selasih di sekeliling kebun sebagai tanaman perangkap, membungkus buah dengan kertas/kantong plastik, memasang perangkap yang dilengkapi antraktan, seperti minyak nabati berbahan aktif *metil eugenol*, menyemprotkan insektisida berbahan aktif *deltametrin* sesuai dengan dosis anjuran.

5. Ulat jengkal (*Palpita* sp) atau ulat grayak (*Spodoptera litura*)



Gambar 2.10. Ulat Grayak

a. Bagian yang diserang :

Daun.

b. Ciri dan gejala :

Daun menggulung dan berlubang-lubang, pada serangan parah yang tersisa hanya tulang daun, terdapat bekas gigitan pada kulit buah.

c. Pengendalian :

Menangkap dan memusnahkan ulat dan daun yang terserang, memasang perangkap yang dilumuri *methyl eugenol* untuk menangkap ulat grayak (seperti pengendalian lalat buah), menyemprotkan insektisida berbahan aktif *deltametrin* sesuai dengan dosis anjuran.

6. Kumbang daun atau oteng-oteng (*Aulachopora femoralis*)



Gambar 2.11. Oteng-oteng

a. Bagian yang diserang :

Daun dan akar.

b. Ciri dan gejala :

Luka bekas gigitan berbentuk lingkaran pada daun, larva menyerang jaringan perakaran hingga pangkal batang sehingga tanaman menjadi layu.

c. Pengendalian :

Pengolahan tanah yang sempurna sebelum tanam (mematikan telur/pupa yang ada di dalam tanah), tanaman yang layu segera dicabut dan dibakar, sterilisasi media dengan *Basamid* G dengan dosis 40 g/m<sup>2</sup>, menyemprotkan insektisida *Decis* 2,5 EC, *larvin* 75 WP, *Pegasus* 500 EC, atau *Supracid* 25 WP.

7. Tungau



Gambar 2.12. Tungau

a. Bagian yang diserang :

Daun.

b. Ciri dan gejala :

Titik-titik halus pada daun, berwarna kuning yang semakin lama berubah menjadi hitam, daun melengkung dan terpelintir berwarna abu-abu dengan jarring-jaring halus dan terdapat sekumpulan hama yang tampak seperti titik-titik berwarna kuning, *orange*, atau merah.

c. Pengendalian :

Membersihkan gulma di sekitar pertanaman, mencabut dan membakar tanaman yang terserang, menyemprotkan akarisida *meothrin* 55 EC atau *Mitac* 250 EC (*amitraz*) dengan konsentrasi 1,0 – 1,5 ml/l

2.3.2. Penyakit

Penyakit tanaman adalah proses gangguan fisiologis secara terus menerus pada tanaman yang disebabkan oleh suatu penyebab utama berupa mikroorganisme patogenik seperti bakteri, cendawan, dan virus. Serangan lebih berat terjadi pada musim hujan, karena kelembapan cukup tinggi.

1. Embun tepung (*Powder mildew*)



Gambar 2.13. Embun tepung

a. Bagian yang diserang :

Daun dan batang muda.

b. Ciri dan gejala :

Daun dan batang muda dilapisi semacam tepung (*powder*) berwarna putih, buah yang terserang berukuran kecil dan rasanya tidak manis.

c. Pengendalian :

Membuat sirkulasi udara lancar serta mengurangi kelembapan di sekitar tanaman, monitoring secara rutin terhadap tanaman sehingga dapat diketahui gejala awal serangan, penyemprotan fungisida *Calix 750 EC* atau *Afugan 300 EC (pyrazophos)* konsentrasi 1 ml/l, dilakukan 5 – 7 hari sekali pada musim hujan dan 10 – 14 hari sekali pada saat kemarau.

2. Embun bulu (*Downy mildew*)

a. Bagian yang diserang :

Daun.



Gambar 2.14. Embun bulu

b. Ciri dan gejala :

Terlihat bulu-bulu halus berwarna abu-abu di bagian bawah bercak (pada permukaan bawah daun), timbulnya bercak-bercak kuning pada daun, lalu berubah menjadi cokelat



kemerahan, buah yang terbentuk abnormal, berukuran kecil, rasa hambar, dan aromanya tidak ada.

c. Pengendalian :

Memotong daun-daun yang terserang cendawan ini dan memusnahkan/membakarnya jauh dari area penanaman, penyemprotan fungisida sistemik *Previcur N* dengan konsentrasi 2 – 3 ml/l, atau yang berbahan aktif *simoksanil* atau *mancozeb*, menghindari pengairan berlebih, melancarkan sirkulasi udara, dan melakukan pengendalian gulma di lahan.

3. Antraknosa



Gambar 2.15. Antraknosa

a. Bagian yang diserang :

Daun, batang muda, bunga, dan buah.

b. Ciri dan gejala :

Timbul berupa bercak-bercak berwarna coklat kelabu sampai kehitaman yang kemudian menyatu pada bagian tanaman seperti daun, batang muda, bunga, dan buah, cendawan dapat membentuk masa spora berwarna merah jambu pada bercak coklat yang terbentuk,.

c. Pengendalian :

Perendaman benih dengan fungisida berbahan aktif *azoksisitrobin* 250 g/l, *propineb* 70% atau *Derasol* 500 SC (1 ml/l) selama 4 jam, pemangkasan bagian tanaman yang terserang kemudian memusnahkannya jauh dari area penanaman, pengaturan jarak tanam yang tepat sehingga kondisi tanaman tidak lembap dan mendapatkan sinar matahari cukup, penyemprotan fungisida dilakukan bila serangan cukup berat, menggunakan fungisida sistemik, dilakukan 7 – 14 hari sekali, penyemprotan fungisida *Derasol* 60 WP dicampur *Dithane/ Vondozeb/ Mancozeb* (1 :5), konsentrasi 2,5 g/l.

#### 4. Layu *fusarium*



Gambar 2.16. Layu *fusarium*

##### a. Bagian yang diserang :

Akar atau batang.

##### b. Ciri dan gejala :

Sulur pada tanaman yang terserang menjadi kuning dan layu, kemudian seluruh tanaman layu dan kelamaan tanaman mati, pada batang terdapat goresan dan mempunyai massa spora cendawan berwarna merah jambu, batang busuk kecoklatan.

##### c. Pengendalian :

Melakukan rotasi tanaman dengan tanaman selain melon, pengapuran lahan untuk meningkatkan Ph tanah, menghindari pemupukan *urea* atau ZA berlebihan, Perendaman benih dengan fungisida *Derasol* 500 SC dengan konsentrasi 1 ml/l, penyemprotan fungisida *Derasol* 500 SC dengan konsentrasi 1,5 ml/l dan dosis 250 ml/tanaman, setiap 14 hari sekali sejak satu bulan setelah tanam atau sejak muncul bunga.

#### 5. Layu bakteri



Gambar 2.17. Layu bakteri

##### a. Bagian yang diserang :

Akar.

##### b. Ciri dan gejala :

Daun layu satu persatu, meskipun warnanya tetap hijau, akhirnya tanaman layu hingga keseluruhan, bila pangkal batang dipotong melintang, akan mengeluarkan lendir putih kental dan lengket.

##### c. Pengendalian :

Perendaman benih dengan bakterisida *Agrimicyn* atau *Agrept* konsentrasi 1,2 g/l, Pengaturan *drainase*, terutama pada musim hujan, jangan sampai ada air yang menggenang di

daerah perakaran, sterilisasi media dengan *Basamid* dosis 40 g/m.

#### 6. Puru akar



Gambar 2.18. Puru akar

##### a. Bagian yang diserang :

Akar.

##### b. Ciri dan gejala :

Daun menguning, pertumbuhan terhambat dan tanaman layu, akar tanaman benjol dan bengkak-bengkak (puru).

##### c. Pengendalian :

Sterilisasi lahan dengan Basamid G dosis 40 g/m, rotasi tanaman dan pembersihan gulma di sekitar pertanaman, pemberian *Furadan* (karbufuran 3%) atau *Rhocap* (*etoprofos* 10%).

#### 7. Busuk pangkal batang (*Gummy steam blight*)



Gambar 2.19. Busuk pangkal batang

##### a. Bagian yang diserang :

Pangkal batang.

b. Ciri dan gejala :

Awalnya pangkal batang seperti tercelup minyak, lalu keluar lendir berwarna merah-cokelat, tanaman layu dan mati, daun yang terserang mengering dan berbunyi kresek-kresek jika diterpa angin, dan seperti kerupuk jika diremas.

c. Pengendalian :

Mencegah kelembapan dan luka di perakaran/pangkal batang, sterilisasi lahan dengan *basamid* G dosis 40 g/m, Pangkal batang dioles dengan fungisida *Callixin* 750 EC (bahan aktif *tridemorph*) konsentrasi 5 ml/l, Daun yang terserang di rompes dan tanaman disemprot dengan fungisida *Derasol* 500 SC (*carbendazim*) konsentrasi 1 – 2 ml/l.

8. Busuk buah



Gambar 2.20. Busuk buah

a. Bagian yang diserang :

Batang, daun, dan buah

b. Ciri dan gejala :

Bercak cokelat kebasahan yang memanjang, daun seperti tersiram air panas kemudian meluas, bercak kebasahan pada buah yang menjadi cokelat kehitaman dan lunak, makin lama

bercak berkerut dan mengendap, bagian buah yang busuk diselimuti cendawan putih.

c. Pengendalian :

Mengurangi kelembapan di sekitar tanaman dengan memangkas daun atau cabang yang berlebihan, rotasi tanaman dengan yang bukan sefamili, mencabut dan membakar tanaman yang terserang, penyemprotan fungisida sistemik *Previcur N* konsentrasi 2 -3 ml/l, konsentrasi 2,5 g/l.

9. Bercak daun bersudut



Gambar 2.21. Bercak daun bersudut

a. Bagian yang diserang :

Daun.

b. Ciri dan gejala :

Terdapat bercak putih pada daun, kemudian berubah menjadi coklat kelabu, mengering, dan berlubang, seolah bersudut.

c. Pengendalian :

Merendam benih dengan bakterisida *Agrimycin* atau *Agrept* dengan konsentrasi 1,2 g/l, penyemprotan bakterisida yang sama diselingi fungisida tembaga konsentrasi 2 g/l, menghindari penyiraman daun dengan air irigasi, melakukan pergiliran tanaman dengan yang bukan sefamili.

## 10. Virus

### a. Bagian yang diserang :

Batang, daun, dan buah.

### b. Ciri dan gejala :

Pertumbuhan tanaman kerdil, daun keriting dan bergelombang dengan bercak kuning tidak teratur, tanaman umumnya gagal membentuk buah atau buahnya kerdil dan abnormal.

### d. Pengendalian :

Perendaman benih dengan fungisida berbahan aktif *azoxystrobin* 250 g/l, penyemprotan fungisida *Derasol* 60 WP dicampur *Dithane/ Vondozeb/ Mancozeb* (1:5), konsentrasi 2,5 g/l.

## 2.4. Metode Naïve Bayes

Merupakan pengklasifikasian dengan metode probabilitas dan statistik yang dikemukakan oleh ilmuwan Inggris Thomas Bayes, yaitu memprediksi peluang di masa depan berdasarkan pengalaman di masa sebelumnya sehingga dikenal sebagai *teorema bayes*. *Naïve bayes* untuk setiap kelas keputusan, menghitung probabilitas dengan syarat bahwa kelas keputusan adalah benar, mengingat vektor informasi obyek. Algoritma ini mengasumsikan bahwa atribut obyek adalah independen. Probabilitas yang terlibat dalam memproduksi perkiraan akhir dihitung sebagai jumlah frekuensi dari “master” table keputusan (Olson dan Delen, 2008:102).

Pengklasifikasi *Bayesian* memiliki tingkat kesalahan minimal dibandingkan dengan klasifikasi lainnya. Namun, dalam prakteknya hal ini

tidak selalu terjadi, karena ketidakakuratan asumsi yang dibuat untuk penggunaannya, seperti kondisi kelas independen, dan kurangnya data probabilitas yang tersedia. Pengklasifikasi *Bayesian* juga berguna dalam memberikan pembenaran teoritis untuk pengklasifikasi lain yang tidak secara eksplisit menggunakan teorema bayes.

Menurut Han dan Kamber (2011:351) Proses dari *The Naïve Bayesian classifier*, atau *Simple Bayesian classifier*, sebagai berikut:

1. *Variable D* menjadi pelatihan set tuple dan label yang terkait dengan kelas. Seperti biasa, setiap tuple diwakili oleh vector atribut n-dimensi,  $X = (x_1, x_2, \dots, x_n)$ , ini menggambarkan pengukuran n dibuat pada tuple dari atribut n, masing-masing,  $A_1, A_2, \dots, A_n$ .
2. Misalkan ada kelas  $m, C_1, C_2, \dots, C_m$ . Diberi sebuah tuple,  $X$ , *classifier* akan memprediksi  $X$  yang masuk kelompok memiliki probabilitas posterior tertinggi, kondisi-disebutkan pada  $X$ . artinya, *classifier naïve bayesian* memprediksi bahwa  $X$  tuple memiliki kelas  $C_i$  jika hanya jika :

$$P(C_i|X) > P(C_j|X) \text{ for } 1 \leq j \leq m, j \neq i. \quad (2.1)$$

Jadi memaksimalkan  $P(C_i | X)$ .  $C_i$  kelas yang  $P(C_i | X)$  dimaksimalkan disebut hipotesis posteriori maksimal. Dengan teoreman bayes :

$$P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)} \quad (2.2)$$

Keterangan :

$P(C_i|X)$  = probabilitas hipotesis  $C_i$  jika diberikan fakta atau record  $X$

$P(X|C_i)$  = Mencari nilai parameter yang memberi kemungkinan yang paling besar (*likelihood*)



$P(C_i)$  = Prior probability dari X (*Prior probability*)

$P(X)$  = Jumlah probability *tuple* yang muncul

3. Ketika  $P(X)$  adalah konstan untuk semua kelas, hanya  $P(X | C_i) P(C_i)$  butuh dimaksimalkan. Jika probabilitas kelas sebelumnya tidak diketahui, maka umumnya diasumsikan ke dalam kelas yang sama, yaitu,  $P(C_i) = P(C_i) = \dots = P(C_m)$ , maka dari itu akan memaksimalkan  $P(X | C_i)$ . Jika tidak, maka akan memaksimalkan  $P(X | C_i) P(C_i)$ . Perhatikan bahwa probabilitas sebelum kelas dapat diperkirakan oleh  $P(C_i) = |C_i \cap D| / |D|$ , dimana  $|C_i \cap D|$  adalah jumlah *tuple* pelatihan kelas  $C_i$  di D.
4. Mengingat *dataset* mempunyai banyak atribut, maka akan sangat sulit dalam mengkomputasi untuk menghitung  $P(X|C_i)$ . Agar dapat mengurangi perhitungan dalam mengevaluasi  $P(X|C_i)$ , asumsi naïve independensi kelas bersyarat dibuat. Dianggap bahwa nilai-nilai dari atribut adalah kondisional independen satu sama lain.

$$\begin{aligned}
 P(X|C_i) &= \prod_{k=1}^n P(x_k|C_i) \\
 &= P(x_1|C_i) \times P(x_2|C_i) \times \dots \times P(x_n|C_i). \quad (2.3)
 \end{aligned}$$

Maka dapat dengan mudah memperkirakan probabilitas  $P(x_1 | C_i)$ ,  $P(x_2|C_i)$ , ...,  $P(x_n|C_i)$  dari pelatihan *tuple*. Ingat bahwa di sini  $x_k$  mengacu pada nilai atribut  $A_k$  untuk *tuple* X. untuk setiap atribut, dilihat dari apakah atribut tersebut kategorikal atau *continuous-valued*. Misalnya, untuk menghitung  $P(X | C_i)$  mempertimbangkan hal – hal berikut :

- a) Jika  $A_k$  adalah kategorikal, maka  $P(X_k | C_i)$  adalah jumlah tuple kelas  $C_i$  di  $D$  memiliki nilai  $X_k$  untuk atribut  $A_k$ , dibagi dengan  $|C_i, D|$ , jumlah *tuple* kelas  $C_i$  di  $D$ .
- b) Jika  $A_k$  *continuous-valued*, maka perlu melakukan sedikit lebih banyak pekerjaan, tapi perhitungannya cukup sederhana. Sebuah atribut *continuous-valued* biasanya diasumsikan memiliki distribusi *Gaussian* dengan rata-rata  $\mu$  dan standar deviasi  $\sigma$ , didefinisikan oleh

$$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \quad (2.4)$$

sehingga:

$$P(x_k | C_i) = g(x_k, \mu_{C_i}, \sigma_{C_i}). \quad (2.5)$$

Setelah itu hitung  $\mu_{C_i}$  dan  $\sigma_{C_i}$ , yang merupakan deviasi *mean* (rata-rata) dan standar masing-masing nilai atribut  $A_k$  untuk *tuple* pelatihan kelas  $C_i$ . Setelah itu gunakan kedua kuantitas dalam persamaan, bersama-sama dengan  $x_k$ , untuk memperkirakan  $P(x_k | C_i)$ .

5. Untuk memprediksi label kelas  $x$ ,  $P(X|C_i)P(C_i)$  dievaluasi untuk setiap kelas  $C_i$ . Classifier memprediksi kelas label dari *tuple*  $x$  adalah kelas  $C_i$ , jika  $P(X|C_i)P(C_i) > P(X|C_j)P(C_j)$  for  $1 \leq j \leq m, j \neq i$ . (2.6)

Dengan kata lain, prediksi adalah  $C_i$  yang mana  $P(X|C_i)P(C_i)$

## 2.5. Certainty Factor

Faktor kepastian (*Certainty Factor*) diperkenalkan oleh Shortliffe Buchanan dalam pembuatan MYCIN untuk menunjukkan besarnya kepercayaan. Faktor kepastian (*Certainty Factor*) merupakan cara dari penggabungan kepercayaan (*belief*) dan ketidakpercayaan (*unbelief*) dalam

bilangan yang tunggal. Dalam certainty theory, data-data kualitatif direpresentasikan sebagai derajat keyakinan (*degree of belief*).

Dalam mengekspresikan derajat keyakinan digunakan suatu nilai yang disebut *certainty factor* (CF) untuk mengasumsikan derajat keyakinan seorang pakar terhadap suatu data. Formulasi certainty factor :

$$CF[H,E] = MB[H,E] - MD[H,E] \quad [2.1]$$

Keterangan :

CF = *Certainty Factor* (factor kepastian) dalam hipotesis H yang dipengaruhi oleh fakta E

MB = *Measure of Belief* (tingkat keyakinan), adalah ukuran kenaikan dari kepercayaan hipotesis H dipengaruhi oleh fakta E.

MD = *Measure of Disbelief* (tingkat tidak yakinan), adalah kenaikan dari ketidakpercayaan hipotesis H dipengaruhi fakta E.

E = *Evidence* (peristiwa atau fakta)

Menurut Kusrini (2008) pengertian mengenai *certainty factor* paralel dan contoh penerapannya adalah sebagai berikut. *Cerntainty factor* paralel merupakan CF yang diperoleh dari beberapa premis pada sebuah aturan. Besarnya CF paralel dipengaruhi oleh CF user untuk masing-masing dan operator dari premis.

Rumus untuk masing-masing operator adalah sebagai berikut :

$$CF(x \text{ Dan } y) = \text{Min}(CF(x), CF(y)) \quad [2.2]$$

$$CF(x \text{ Atau } y) = \text{Max}(CF(x), CF(y)) \quad [2.3]$$

$$CF(\text{Tidak } x) = -CF(x) \quad [2.4]$$

Bentuk dasar rumus *certainty factor* sebuah aturan JIKA E MAKA H ditunjukkan oleh Kusrini (2008) dalam rumus berikut :

$$CF(H,e) = CF(E,e)*CF(H,E) \quad [2.5]$$

Dimana :

$CF(E,e)$  : *Certainty factor evidence* E yang dipengaruhi oleh evidence e

$CF(H,E)$  : *Certainty factor* hipotesis dengan asumsi *evidence* diketahui dengan pasti, yaitu ketika  $CF(E,e) = 1$

$CF(H,e)$  : *Certainty factor* hipotesis yang dipengaruhi oleh *evidence* e

Jika semua *evidence* pada antecedent diketahui dengan harga pasti maka rumusnya adalah sebagai berikut :

$$CF(H,e) = CF(H,E) \quad [2.6]$$

CF sequensial diperoleh dari hasil perhitungan CF parallel dari semua premis dalam satu aturan dengan CF aturan yang diberikan oleh pakar. Rumusan untuk melakukan perhitungan CF sequensial adalah sebagai berikut :

$$CF(x,y) = CF(x)*CF(y) \quad [2.7]$$

Dengan

$CF(x,y)$  : CF sequential

$CF(x)$  : CF parallel dari semua premis

$CF(y)$  : CF pakar

*Certainty factor* gabungan merupakan *certainty factor* akhir dari sebuah calon kesimpulan. *Certainty factor* ini dipengaruhi oleh semua *certainty factor* parallel dari aturan yang menghasilkan konklusi tersebut. *Certainty factor* gabungan diperlukan jika suatu konklusi diperoleh dari beberapa aturan sekaligus. *Certainty factor* akhir dari suatu aturan dengan aturan yang

lain digabungkan untuk mendapatkan nilai *certainty factor* akhir bagi calon konklusi tersebut. Rumus untuk mendapatkan *certainty factor* gabungan adalah sebagai berikut (Kusrini, 2008) :

$$CF(x, y) = \begin{cases} CF(x) + CF(y) - (CF(x) * CF(y)), & CF(x) > 0 \text{ dan } CF(y) > 0 \\ \frac{CF(x) + CF(y)}{(1 - (Min(|CF(x)|, |CF(y)|)))}, & \text{salah_satu}(CF(x), CF(y)) < 0 \\ CF(x) + (CF(y) * (1 + CF(x))), & CF(x) < 0 \text{ dan } CF(y) > 0 \\ CF(y) + (CF(x) * (1 + CF(y))), & CF(x) > 0 \text{ dan } CF(y) < 0 \end{cases} \quad [2.8]$$