

BAB II

TINJAUAN PUSTAKA

II.1. Pengertian Hama

Hama adalah penyebab suatu kerusakan pada tanaman yang dapat dilihat dengan pancaindera (mata). Hama tersebut dapat berupa binatang, dan dapat merusak tanaman secara langsung maupun secara tidak langsung. Hama yang merusak tanaman secara langsung dapat dilihat bekasnya, misalnya greskan dan gigitan. Sedangkan hama yang merusak tanaman secara tidak langsung biasanya melalui suatu penyakit. Berdasarkan cara menyerangny dengan tipe alat mulut hama digolongkan sebagai berikut :

1. *Ordo Hemiptera*

Hama yang termasuk *ordo* ini tipe alat mulutnya pengisap.

Contohnya : kepik, walangsangit, dan wereng.

2. *Ordo Lepidoptera*

Hamanya ialah golongan kupu – kupu, tipe alat mulutnya pengisap yang berupa belalai. Golongan ini merusak karena mereka adalah, penggerek batang, penggerek buah, ulat dan sebagainya.

3. *Ordo Coleoptera*

Ordo ini merupakan tingkatan yang paling besar dari insekta lainnya. Hama ini sebangsa kumbang, tipe alat mulutnya penggigit.

4. *Ordo Ortoptera*

Yang termasuk *ordo* ini adalah sebangsa belalang, jengkerik, gangsir, kecoa dan lain – lain. *Ordo Ortoptera* mempunyai alat mulut penggigit pengunyah.

5. *Ordo hemiptera*

Sebangsa lebah dan tipe alat mulutnya penggigit pengunyah.

6. *Ordo Diptera*

Yang termasuk ordo ini ialah sebangsa lalat. Tipe mulutnya adalah penjilat pengisap dan penusuk pengisap.

7. *Ordo Tisanoptera*

Hama yang termasuk ordo ini sebangsa kutu, tipe alat mulutnya pengisap dan berujung tajam. (Hudi Matnawy;2011:11-12).

II.1.1. Pengendalian Hama

1. Hama Ulat

Gejala yang ditimbulkan oleh serangan ulat adalah rusaknya bagian tepi daun aglaonema. Daun yang diserang biasanya adalah daun muda dan setengah tua. Sedangkan daun tuanya tidak. Selain daun, kadang – kadang batang aglaonema juga menjadi sasaran serang ulat. Gejalanya muncul lubang – lubang pada batang. Pengendalian hama ulat bisa dilakukan dengan menyemprotkan insektisida decis 25 EC dengan dosis disesuaikan dengan petunjuk pakai yang ada dikemasannya.

2. Hama Kutu Perisai

Kutu perisai menyerang dan meyebabkan kerusakan daun aglaonema. Kutu yang bentuk punggungnya mirip perisai ini sering terlihat berkumpul membentuk barisan dibagian tulang daun. Pengendaliannya dilakukan dengan mengaplikasikan insektisida sistemik yang mengandung bahan aktif *acephate*.

3. Hama Kutu Putih

Hama yang merusak dengan cara menghisap cairan dalam tanaman ini sering terlihat dibagian ketiak daun, dibawah permukaan daun, dan dilipatan daun muda. Aglaonema yang terserang hama kutu putih menunjukkan gejala daun yang mengerut dan berjelaga. Pengendalian hama kutu putih dilakukan dengan menyemprotkan insektisida decis 25 EC dengan dosis disesuaikan dengan petunjuk pakai yang ada dikemasannya.

4. Hama Root Mealy Bugs

Hama ini memiliki bentuk yang sama seperti kutu putih. Namun, tempat hidupnya berada di akar. Aglaonema yang terserang hama root mealy bugs terlihat kurus, tumbuh kerdil, bentuk daunnya kecil, dan lunglai. Jika media tanamannya dibongkar, akan terlihat bintik – bintik putih dibagian akarnya. Pengendalian hama ini dilakukan dengan menyemprotkan insektisida Confidor 200SL dengan dosis disesuaikan dengan petunjuk pakai yang ada di kemasannya.

5. Hama Thrips

Hama yang berbentuk kutu yang berukuran sangat kecil ini biasa dijumpai diketiak daun, pelepah, batang, dan bunga aglaonema. Hama thrips merusak dengan cara menghisap cairan dalam tanaman, mengakibatkan daun aglaonema mengerut, berubah warna menjadi kekuningan, lalu mati. Pengendalian hama ini dilakukan dengan menyemprotkan insektisida confidor setiap dua minggu sekali dengan dosis disesuaikan dengan petunjuk pakai yang ada di kemasannya.

6. Hama Belalang

Sama seperti hama ulat, hama ini juga merusak daun aglaonema dengan cara memakannya. Dalam jumlah banyak, hama ini dapat menyebabkan kerusakan yang sangat besar. Peralnya, daya makan belalang tergolong tinggi. Pengendalian hama ini bisa dilakukan dengan cara menangkapnya satu persatu lalu dimusnahkan. Cara lainnya, semprotkan insektisida confidor 200 SL dengan dosis disesuaikan dengan petunjuk pakai yang ada dikemasannya.

7. Hama Kutu Sisik

Hama ini bentuknya mirip dengan lintah tetapi ukurannya lebih kecil. Kutu sisik menyerang bagian daun, pelepah batang, dan bunga. Akibat serangannya, daun akan mengerut, menguning, layu, dan akhirnya mati. Pengendalian kutu sisik dilakukan dengan membuangnya dengan cara dikerik. Selain itu, dapat juga disemprotkan dengan insektisida confidor 200 SL atau Agrimex 18 EC dengan dosis disesuaikan dengan petunjuk pakai di kemasannya. (Redaksi Agromedia; 2011:38).

II.2. Pengertian Penyakit

Menurut Hudi Matnawy (2011:12). Sakit adalah situasi dimana proses hidup suatu tanaman menyimpang dari keadaan normal dan menimbulkan kerusakan, sehingga tanaman itu tidak dapat tumbuh dan berkembang biak seperti biasa, bahkan dapat menyebabkan matinya tanaman tersebut. Penyakit tanaman adalah penyebab kerusakan pada tanaman selain yang disebabkan oleh hama. Ilmu yang mempelajari penyakit tanaman disebut *fitopatologi*. Penyakit disini dapat berupa :

1. Cendawan.
2. Bakteri.
3. Algae atau ganggang.
4. Virus.
5. Keadaan fisiologis yang merugikan. (Hudi Matnawy; 2011:12).

II.2.1. Pengendalian Penyakit

1. Penyakit Akibat Bakteri

Bakteri yang menyerang aglaonema adalah *Erwinia Coratovora*. Gejala kerusakannya, daun dan tangkai aglaonema rusak, berlendir, dan berubah warna menjadi kecokelatan, dan terkadang mengeluarkan bau busuk. Jika tidak ditangani secepatnya, bagian yang terserang akan hancur karena jaringannya telah rusak. Agar bakteri tidak menyerang sebaiknya jangan meletakkan pot odenium terlalu berdekatan, jangan menyiram berlebihan, dan pastikan drainase berlangsung dengan baik. Pengendalian penyakit ini dilakukan dengan menyemprotkan bakterisida seperti vinegar siltion atau agrept dengan dosis disesuaikan dengan petunjuk pakai yang ada di kemasannya. Selain itu, bagian yang terserang harus dipotong dan dimusnahkan agar tidak menular ke alaonema yang lain.

2. Penyakit Layu Fusarium

Tulang daun yang memucat dan berubah warna menjadi kecokelatan, lalu tangkainya membusuk merupakan gejala penyakit layu yang ditimbulkan oleh serangan jamur fusarium. Penyebab munculnya jamur fusarium adalah karena media tanam yang terlalu basah sehingga pH-nya menjadi asam. Untuk

mencegah munculnya jamur, jangan menyiram terlalu berlebihan dan pastikan drainase pot berjalan dengan baik. Pengendalian penyakit ini dilakukan dengan membuang bagian tanaman yang terserang dan media tanam harus diganti dengan media tanam baru, lalu siram dengan larutan 2-5cc/liter sebanyak 200ml/tanaman. Cara lainnya, semprotkan fungisida seperti folicur 25 WP, Folicur 250 EC, atau Delsone MX 200. Dosisnya disesuaikan dengan petunjuk pakai yang ada di kemasannya.

3. Penyakit Busuk Akar

Sesuai namanya, penyakit yang disebabkan oleh jamur *Phyitium* ini menyerang akar aglaonema. Serangan jamur ini menyebabkan pertumbuhan aglaonema menjadi tidak normal, daun mudanya tumbuh kerdil dan menguning. Munculnya jamur ini umumnya diakibatkan karena media tanam terlalu basah akibat terlalu sering disiram. Jika aglaonema sudah terserang dan jamur ini, segera ganti media tanamnya, lalu potong bagian akar yang terserang dan oleskan fungisida di bekas potongan tersebut. Setelah itu semprotkan *bycylcin*, yaitu larutan campuran fungisida dan bakterisida atau semprotkan fungisida previcur N dengan dosis disesuaikan dengan petunjuk pakai yang ada dikemasannya.

4. Penyakit Busuk Daun

Gejala penyakit yang disebabkan oleh jamur ini adalah munculnya bercak – bercak pada daun yang disusul dengan kebusukan. Daun yang menunjukkan gejala tersebut harus dibuang agar tidak menular ke daun yang lain. Pengendaliannya dilakukan dengan menyemprotkan fungisida folicur 25 WP,

Folicur 250 EC, atau score dengan dosis disesuaikan dengan petunjuk pakai yang ada dikemasannya.

5. Penyakit Akibat Virus

Gejala aglaonema yang terserang virus umumnya ditunjukkan dengan pertumbuhan daun yang kerdil dan mengeriting. Karena belum ada obat yang bisa digunakan untuk mengobati penyakit ini, satu – satunya cara yang dapat dilakukan adalah memusnahkan aglaonema dengan cara dibakar. Penyakit akibat virus biasanya ditimbulkan oleh pemakaian alat kerja yang digunakan untuk memotong aglaonema tidak steril. Oleh karena itu, sebelum melakukan pemotongan aglaonema pastikan terlebih dahulu obat yang digunakan telah steril. (Redaksi Agromedia; 2011:40).

II.3. Tanaman Hias Aglaonema

Aglaonema atau lebih dikenal dengan julukan ratu daun termasuk kelompok komoditas tanaman hias berdaun indah. Daunnya yang indah warnawarni menarik perhatian setiap mata memandang. Saat ini aglaonema masih menjadi salah satu tanaman yang populer, baik untuk menambah keindahan lingkungan halaman/pekarangan maupun untuk dekorasi ruangan. Aglaonema berasal dari daerah Asia beriklim tropis, dan tersebar dari Cina bagian Selatan, Myanmar, Thailand, Malaysia, Indonesia dan Filipina. Di habitat aslinya, tanaman ini hidup di hutan di bawah tegakan pohon yang terlindungi dari sinar matahari langsung. Jenisnya bermacam-macam dengan nama-nama yang menarik, seperti dona karmen, legacy, lipstick, tiara. Harga agalonema bervariasi, mulai dari

ribuan (dona karmen) hingga ratusan ribu (legacy). Dalam kurun waktu lima tahun terakhir, permintaan akan tanaman aglaonema meningkat pesat. Untuk memenuhi permintaan tersebut, perlu dilakukan perbanyak bibit yang seragam dan dalam jumlah yang besar. Peningkatan permintaan ini secara tidak langsung berdampak terhadap peningkatan kegiatan produksi di sentra produksi. Dengan demikian, kegiatan produksi tersebut perlu terus didorong agar memberi kontribusi lebih besar terhadap perekonomian Nasional.

Aglaonema dapat diperbanyak dengan menggunakan biji, anakan dan setek batang. Pada skala komersial, setek batang merupakan cara perbanyak yang umum dilakukan. Para petani biasanya memperbanyak tanaman dengan cara memotong batang sepanjang 5-7 cm (4-5 buku) yang hanya menghasilkan kurang lebih 2 tunas per tanaman baru. Hal ini yang berarti akan menambah kerumitan pengelolaan produksi dalam skala usaha tani. Tanaman aglaonema mempunyai batang berbuku dan tidak berkayu. Buku-buku batang merupakan indikator pertumbuhan daun, dimana buku merupakan tempat melekatnya tangkai daun. Pada setiap ruas buku terdapat satu mata tunas tidur yang sebenarnya mampu membentuk tanaman baru. (Redaksi Agromedia; 2007 : 30).

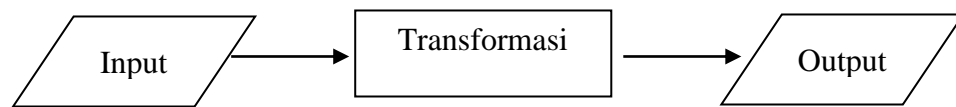
II.4. Pengertian Sistem

Menurut Prof. Dr. Ir. Marimin, M.Sc; 2011:1. Sistem adalah suatu kesatuan usaha yang terdiri dari bagian – bagian yang berkaitan satu sama lain yang berusaha mencapai suatu tujuan dalam suatu lingkungan kompleks. Pengertian

tersebut mencerminkan adanya beberapa bagian dan hubungan antar bagian, ini menunjukkan kompleksitas dari sistem yang meliputi kerja sama antara bagian yang interdependen satu sama lain. Selain itu, dapat dilihat bahwa sistem berusaha mencapai tujuan. Pencapaian tujuan ini menyebabkan timbulnya dinamika, perubahan yang terus menerus perlu dikembangkan dan dikendalikan. Definisi tersebut menunjukkan bahwa sistem sebagai gugus dari elemen – elemen yang saling berinteraksi secara teratur dalam rangka mencapai tujuan atau subtujuan.

Sifat – sifat dasar suatu sistem, antara lain :

1. Pencapaian tujuan, orientasi pencapaian tujuan akan memberikan sifat dinamis kepada sistem, memberi siri perubahan yang terus – menerus dalam usaha mencapai tujuan.
2. Kesatuan usaha, mencerminkan suatu sifat dasar dari sistem, dimana hasil keseluruhan melebihi dari jumlah bagian – bagiannya atau sering disebut konsep sinergi.
3. Keterbukaan terhadap lingkungan, lingkungan merupakan sumber kesempatan maupun hambatan pengembangan. Keterbukaan terhadap lingkungan membuat penilaian terhadap suatu sistem menjadi relatif atau yang dinamakan *equifinality* atau pencapaian tujuan suatu sistem tidak mutlak harus dilakukan dengan satu cara terbaik. Tetapi pencapaian tujuan suatu sistem dapat dilakukan melalui berbagai cara sesuai dengan tantangan lingkungan yang dihadapi.
4. Transformasi, merupakan proses perubahan input menjadi output yang dilakukan oleh sistem. Proses transformasi diilustrasikan pada gambar II.1.



Gambar II.1. Proses Transformasi Input Menjadi Output.

(Sumber : Prof. Dr. Ir. Marimin, M.Sc. ; 2011 : 2)

5. Hubungan antara bagian, kaitan antara subsistem inilah yang akan memberikan analisis sistem, suatu dasar pemahaman yang lebih luas.
6. Sistem ada berbagai macam, antara lain sistem terbuka, sistem tertutup, dan sistem dengan umpan balik.
7. Mekanisme pengendalian, mekanisme ini menyangkut sistem umpan balik yang merupakan suatu bagian yang memberi informasi kepada sistem mengenai efek dari perilaku sistem terhadap pencapaian tujuan atau pemecahan persoalan yang dihadapi. Skema proses transformasi sistem dengan mekanisme pengendalian disajikan pada gambar II.2.



Gambar II.2. Skema Proses Transformasi Sistem Dengan Mekanisme Pengendalian.

(Sumber : Prof. Dr. Ir. Marimin, M.Sc. ; 2011 : 3)

II.5. Pengertian Sistem Pakar

Sistem pakar adalah aplikasi berbasis komputer yang digunakan untuk menyelesaikan masalah sebagaimana yang dipikirkan oleh pakar. Pakar yang

dimaksud disini adalah orang yang mempunyai keahlian khusus yang dapat menyelesaikan masalah yang tidak dapat diselesaikan oleh orang awam. Sebagai contoh, dokter adalah seorang pakar yang mampu mendiagnosis penyakit yang diderita pasien serta dapat memberikan penatalaksanaan terhadap penyakit tersebut. Tidak semua orang dapat mengambil keputusan mengenai diagnosis dan memberikan penatalaksanaan suatu penyakit. Contoh yang lain, montir adalah seorang yang punya keahlian dan pengalaman dalam menyelesaikan kerusakan mesin motor/mobil, psikolog adalah orang yang ahli dalam memahami kepribadian seseorang, dan lain – lain. Sistem pakar, yang mencoba memecahkan masalah yang biasanya hanya bisa dipecahkan oleh seorang pakar, dipandang berhasil ketika mampu mengambil keputusan seperti yang dilakukan oleh pakar aslinya baik dari sisi proses pengambilan keputusannya maupun hasil yang diperoleh. Sebuah sistem pakar memiliki 2 komponen utama yaitu basis pengetahuan dan mesin inferensi. Basis pengetahuan merupakan tempat penyimpanan pengetahuan dalam memori komputer, dimana pengetahuan ini diambil dari pengetahuan pakar. (Kusrini; 2010:3).

Sistem pakar adalah sistem berbasis komputer yang menggunakan pengetahuan, fakta, dan teknik penalaran dalam memecahkan masalah yang biasanya hanya dapat dipecahkan oleh seorang pakar dalam bidang tersebut. Pada dasarnya sistem pakar diterapkan untuk mendukung aktivitas pemecahan masalah. Beberapa aktivitas pemecahan yang dimaksud antara lain : pembuatan keputusan (*decision making*), pemaduan pengetahuan (*knowledge fusing*), pembuatan desain (*designing*), perencanaan (*planning*), prakiraan (*forecasting*), pengaturan

(*regulating*), pengendalian (*controlling*), diagnosis (*diagnosing*), perumusan (*prescribing*), penjelasan (*explaining*), pemberian nasihat (*advising*) dan pelatihan (*tutoring*). Selain itu sistem pakar juga dapat berfungsi sebagai asisten yang pandai dari seorang pakar.

Sistem pakar dibuat pada wilayah pengetahuan tertentu untuk suatu kepakaran tertentu yang mendekati kemampuan manusia di salah satu bidang. Sistem pakar mencoba mencari solusi yang memuaskan sebagaimana yang dilakukan seorang pakar. Selain itu sistem pakar juga dapat memberikan penjelasan terhadap langkah yang diambil dan memberikan alasan atas saran atau kesimpulan yang ditemukannya. Biasanya sistem pakar hanya digunakan untuk memecahkan masalah yang memang sulit untuk dipecahkan dengan pemrograman biasa, mengingat biaya yang diperlukan untuk membuat sistem pakar jauh lebih besar dari pembuatan sistem biasa (Kusrini; 2010:11).

II.6. Metode *Certainty Factor*

Certainty Theory ini diusulkan oleh Shortliffe dan Buchanan pada tahun 1975 untuk mengakomodasi ketidakpastian pemikiran (*inexact reasoning*) seorang pakar. Teori ini berkembang bersamaan dengan pembuatan sistem pakar MYCIN. *Team* pengembang MYCIN mencatat bahwa tim ahli sering kali menganalisa informasi yang ada dengan ungkapan seperti misalnya : mungkin, kemungkinan besar, hampir pasti. Untuk mengakomodasi hal ini tim MYCIN menggunakan *certainty factor* (CF) guna menggambarkan tingkat keyakinan

pakar terhadap masalah yang sedang dihadapi. Secara umum, *rule* direpresentasikan dalam bentuk sebagai berikut :

$$\begin{aligned} & \text{IF } E_1 \text{ [AND/OR] } E_2 \text{ [AND/OR] } \dots E_n \\ & \text{THEN } H \text{ (CF = CF) } \dots \dots \dots (1). \end{aligned}$$

(Bain Khusnul Khotimah; 2010:13)

Cerainty Factor merupakan suatu metode yang digunakan untuk menyatakan kepercayaan dalam sebuah kejadian (fakta atau hipotesis) berdasarkan bukti atau penilaian pakar. Secara konsep, *certainty factor* (CF) merupakan salah satu teknik yang digunakan untuk mengatasi ketidakpastian dalam pengambilan keputusan. *Certainty Factor* (CF) dapat terjadi dengan berbagai kondisi. Diantara kondisi yang terjadi adalah terdapat beberapa *antesenden* (dalam *rule* yang berbeda) dengan satu konsekuen yang sama. Dalam kasus ini, kita harus mengagregasikan nilai CF keseluruhan dari setiap kondisi yang ada *belive* dan *disbelive*. *Belive* merupakan keyakinan, sedangkan *disbelive* merupakan ketidakyakinan. Adapun notasi atau rumusan dasar dari *certainty factor*, sebagai berikut.

$$CF [h,e] = MB [h,e] - MD [h,e]$$

Keterangan :

CF [h,e] = *Certainty Factor* dalam hipotesis h yang dipengaruhi oleh fakta e.

MB [h,e] = *Measure of Belive*, merupakan nilai kenaikan dari kepercayaan hipotesis h dipengaruhi oleh fakta e.

MD [h,e] = *Measure of Disbelive*, merupakan nilai kenaikan dari ketidakpercayaan hipotesis h dipengaruhi oleh fakta e.

h = Hipotesis

e = *Evidence*. (Putu Ary Darma Yasa; 2012:2).

II.7. Entity Relationship Diagram (ERD)

Pada dasarnya ERD (*Entity Relationship Diagram*) adalah sebuah diagram yang secara konseptual memetakan hubungan antar penyimpanan pada diagram DFD di atas. ERD ini digunakan untuk melakukan permodelan terhadap struktur data dan hubungannya. Penggunaan ERD ini dilakukan untuk mengurangi tingkat kerumitan penyusunan sebuah database yang baik.

Entity dapat berarti sebuah obyek yang dapat dibedakan dengan obyek lainnya. Obyek tersebut dapat memiliki komponen – komponen data (atribut atau field) yang membuatnya dapat dibedakan dari obyek yang lain. Dalam dunia database entity memiliki atribut yang menjelaskan karakteristik dari entity tersebut. Ada dua macam atribut yang dikenal dalam entity yaitu atribut yang berperan sebagai kunci primer dan atribut deskriptif. Hal ini berarti setiap entity memiliki himpunan yang diperlukan sebuah primary key untuk membedakan anggota – anggota dalam himpunan tersebut.


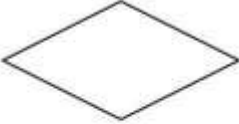


Atribut dapat memiliki sifat – sifat sebagai berikut :

- Atomic, atomik adalah sifat dari atribut yang menggambarkan bahwa atribut tersebut berisi nilai yang spesifik dan tidak dapat dipecah lagi. Contoh dari sifat atomik adalah field status dari tabel karyawan yang hanya berisi menikah atau single.

- multivalued, sifat ini menandakan atribut ini bisa memiliki lebih dari satu nilai untuk tiap entity tertentu. Misalnya adalah field hobi, hobi dari tiap karyawan mungkin dan hampir pasti lebih dari satu. Misalnya karyawan A memiliki hobi : membaca, nonton TV dan bersepeda.
- Composite, atribut yang bersifat komposit adalah atribut yang nilainya adalah gabungan dari beberapa atribut yang bersifat atomik. Contohnya adalah atribut alamat yang dapat dipecah menjadi atribut atomik berupa alamat, kode pos, no telepon, dan kota, (Wahana Komputer ; 2010 : 30).

Tabel II.1. Simbol ERD

Simbol – symbol ERD (Data Flow Diagram)

Simbol	Nama
	Entitas , adalah suatu objek yang dapat diidentifikasi dalam lingkungan pemakai.
	Relasi , menunjukkan adanya hubungan di antara sejumlah entitas yang berbeda.
	Atribut , berfungsi mendeskripsikan karakter entitas (atribut yg berfungsi sebagai key diberi garis bawah)
	Garis , sebagai penghubung antara relasi dengan entitas, relasi dan entitas dengan atribut.

II.8. Normalisasi

Normalisasi *file* adalah suatu proses pengelompokkan elemen data kedalam tabel yang menyatakan hubungan antara entitas sehingga terwujud suatu bentuk yang memudahkan adanya perubahan dengan dampak terkecil. Proses pada pengolahan data yang mengakibatkan efek samping yang tidak diharapkan sering disebut dengan istilah anomali yang dapat terjadi akibat *inserting*, *updating*, atau *deleting*. Oleh karena itulah diperlukan normalisasi *file*. Adapun teknik penyusunan normalisasi *file* adalah dengan menentukan hal – hal sebagai berikut :

- Kunci Tribut

Setiap *file* selalu mempunya kunci berupa satu *set field* yang dapat mewakili *record*. Sebagai contoh, pada tabel barang terdapat *field* kunci berupa kode barang yang mewakili nama barang, harga barang, jumlah barang dan sebagainya.

- Kunci Kandidat (*Candidate Key*)

Kunci kandidat adalah suatu atribut atau set minimal atribut yang mengidentifikasi secara unik suatu kejadian spesifik dari entiti. Jika kunci kandidat berisi lebih dari satu atribut, maka biasanya disebut sebagai kunci gabungan (*Composite Key*).

- Kunci Primer (*Primary Key*)

Kunci primer adalah satu atribut atau satu set minimal atribut yang tidak hanya mengidentifikasi secara unik suatu spesifik, tetapi juga dapat mewakili setiap kejadian dari suatu entiti. Setiap kunci kandidat mempunyai peluang untuk menjadi kunci primer, tetapi sebaiknya dipilih

satu saja yang dapat mewakili secara menyeluruh terhadap entiti yang ada. Contohnya adalah NIM, sifatnya unik dan tidak mungkin ganda dan mewakili secara menyeluruh terhadap entiti mahasiswa dan setiap mahasiswa selalu memiliki NIM. Selain itu kita harus melihat juga fungsi No KTP, ini dapat digunakan bila sampai suatu saat mahasiswa harus memiliki Kartu Tanda Mahasiswa tetapi NIM belum keluar, maka No KTP dapat digunakan.

- Kunci Alternatif (*Alternate Key*)

Kunci alternatif adalah kunci kandidat yang tidak dipakai sebagai kunci primer. Biasanya kunci ini digunakan sebagai kunci pengurutan data saja, misalnya dalam pembuatan laporan.

- Kunci Tamu (*Foreign Key*)

Kunci tamu adalah satu atribut (atau satu set atribut) yang melengkapi satu hubungan yang menunjukkan ke tabel induknya. Kunci tamu biasanya ditempatkan pada entiti anak dan sama dengan kunci primer induk yang direlasikan. Hubungan antara entiti induk dengan anak adalah (biasanya) hubungan satu ke banyak (*one to many*). (Uus Rusmawan; 2010:92-93).

II.8.1. Bentuk-bentuk Normalisasi

a. Bentuk tidak normal

Bentuk ini merupakan kumpulan data yang akan direkam, tidak ada keharusan mengikuti format tertentu, dapat saja tidak lengkap dan terduplikasi. Data dikumpulkan apa adanya sesuai keadaanya.

b. Bentuk normal tahap pertama (1st Normal Form)

Definisi :

Sebuah table disebut 1NF jika :

- Tidak ada baris yang duplikat dalam tabel tersebut.
- Masing-masing cell bernilai tunggal

Catatan: Permintaan yang menyatakan tidak ada baris yang duplikat dalam sebuah tabel berarti tabel tersebut memiliki sebuah kunci, meskipun kunci tersebut dibuat dari kombinasi lebih dari satu kolom atau bahkan kunci tersebut merupakan kombinasi dari semua kolom.

Berikut ini akan dicontohkan normalisasi dari tabel kuliah yang memiliki atribut : kode_kul, nama_kul, sks, semester, waktu, tempat, dan nama_dos.

Tabel kuliah tersebut tidak memenuhi normalisasi pertama, karena terdapat atribut waktu yang tergolong ke dalam atribut bernilai banyak. Agar tabel tersebut dapat memenuhi 1NF, maka solusinya adalah dengan mendekomposisi tabel kuliah menjadi :

- Tabel kuliah (kode_kul, nama_kul, sks, semester, nama_dos).
- Tabel jadwal (kode_kul, waktu, ruang).

c. Bentuk normal tahap kedua (2nd normal form)

Bentuk normal kedua (2NF) terpenuhi jika pada sebuah tabel semua atribut yang tidak termasuk dalam primary key memiliki ketergantungan fungsional pada primary key secara utuh.

Sebuah tabel dikatakan tidak memenuhi 2NF, jika ketergantungannya hanya bersifat parsial (hanya tergantung pada sebagian dari primary key). Bentuk normal kedua akan dicontohkan berikut.

Misal tabel nilai terdiri dari atribut kode_kul, nim dan nilai. Jika pada tabel nilai. Misalnya kita tambahkan sebuah atribut yang bersifat redundan, yaitu nama_mhs, maka tabel nilai ini dianggap melanggar 2NF.

Primary key pada tabel nilai adalah (kode_kul, nim).

Penambahan atribut baru (nama_mhs) akan menyebabkan adanya ketergantungan fungsional yang baru yaitu $\text{nim} > \text{nama_mhs}$. Karena atribut nama_mhs ini hanya memiliki ketergantungan parsial pada primary key secara utuh (hanya tergantung pada nim, padahal nim hanya bagian dari primary key). Bentuk normal kedua ini dianggap belum memadai karena meninjau sifat ketergantungan atribut terhadap atribut terhadap primary key saja.

d. Bentuk normal tahap ketiga (3rd normal form)

Sebuah tabel dikatakan memenuhi bentuk normal ketiga (3NF), jika untuk setiap ketergantungan fungsional dengan notasi $X \rightarrow A$, dimana A mewakili semua atribut tunggal di dalam tabel yang tidak ada di dalam X, maka :

- X haruslah superkey pada tabel tersebut.
- Atau A merupakan bagian dari primary key pada tabel tersebut.

Misalkan pada tabel mahasiswa, atribut alamat_mhs dipecah kedalam alamat_jalan, alamat_kota dan kode_pos. Bentuk ini tidak memenuhi 3NF, karena terdapat ketergantungan fungsional baru yang muncul pada tabel tersebut, yaitu :

alamat_jalan, nama_kota – kode_pos

Dalam hal ini (alamat_jalan, nama_kota) bukan superkey sementara kode_pos juga bukan bagian dari primary key pada tabel mahasiswa. Jika tabel mahasiswa didekomposisi menjadi tabel mahasiswa dan tabel alamat, maka telah memenuhi 3NF. Hal itu dapat dibuktikan dengan memeriksa dua ketergantungan fungsional pada tabel alamat tersebut, yaitu :

alamat_jalan, nama_kota – kode_pos

kode_pos – nama_kota

Ketergantungan fungsional yang pertama tidak melanggar 3NF, karena (alamat_jalan, nama_kota) merupakan superkey (sekali sebagai primary key) dari tabel alamat tersebut. Demikian juga dengan ketergantungan fungsional yang kedua meskipun (kode_pos) bukan merupakan superkey, tetapi nama_kota merupakan bagian dari primary key dari tabel alamat. Karena telah memenuhi 3NF, maka tabel tersebut tidak perlu di-dekomposisi lagi.

e. Bentuk Normal Tahap Keempat dan Kelima

Penerapan aturan normalisasi sampai bentuk normal ketiga sudah memadai untuk menghasilkan tabel berkualitas baik. Namun demikian, terdapat pula bentuk normal keempat (4NF) dan kelima (5NF). Bentuk Normal keempat berkaitan dengan sifat ketergantungan banyak nilai (*multivalued dependency*) pada suatu tabel yang merupakan pengembangan dari ketergantungan fungsional. Adapun bentuk normal tahap kelima merupakan nama lain dari *Project Join Normal Form* (PJNF).

f. Boyce Code Normal Form (BCNF)

- Memenuhi 1st NF
- Relasi harus bergantung fungsi pada atribut superkey (Kusrini, M.Kom, Grace Gata ; 2010 : 41-43).

II.9. Pengertian 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.

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.

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 : 2).

Hubungan data dan informasi dapat digambarkan sebagai berikut :



Gambar II.3. Data dan Informasi.

(Sumber : Yuhefizard ; 2010 : 2)

II.10. Mengenal Visual Basic

Visual Basic dibuat oleh microsoft, merupakan salah satu bahasa pemrograman berorientasi objek yang mudah dipelajari. Selain menawarkan kemudahan, Visual Basic juga cukup andal untuk digunakan dalam pembuatan berbagai aplikasi, terutama aplikasi database. Visual basic merupakan bahasa pemrograman event drive, dimana program akan menunggu sampai ada respons dari user/pemakai program aplikasi yang dapat berupa kejadian atau event, misalnya ketika user mengklik tombol atau menekan enter. Jika kita membuat aplikasi dengan visual basic maka kita akan mendapatkan file yang menyusun aplikasi tersebut, yaitu :

1. File Project (*.vbp)

File ini merupakan kumpulan dari aplikasi yang kita buat. File project bisa berupa file *.frm, *.dsr atau file lainnya.

2. File Form (*.frm)

File ini merupakan file yang berfungsi untuk menyimpan informasi tentang bentuk form maupun interface yang kita buat. (Edy Winarto ; 2010 : 1).

II.11. SQL Server

SQL *Server* 2008 adalah sebuah RDBMS (*Relational Database Management System*) yang di *develop* 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. (Cybertron Solution ; 2010 : 101).

II.12. Unified Modeling Language (UML)

Menurut Windu Gata (2013 : 4) Hasil pemodelan pada OOAD terdokumentasikan dalam bentuk *Unified Modeling Language (UML)*. UML adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak.

UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar


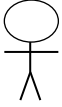

bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem.

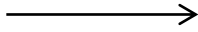
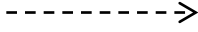

Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut :

1. *Use case* Diagram

Use case diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *use case* diagram, yaitu :

Tabel II.2. Simbol *Use Case*

Gambar	Keterangan
	<p><i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i>.</p>
	<p>Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i>, tetapi tidak memiliki control terhadap <i>use case</i>.</p>
	<p>Asosiasi antara aktor dan <i>use case</i>, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengidikasikan aliran data.</p>



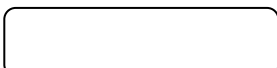
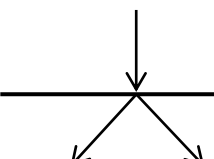
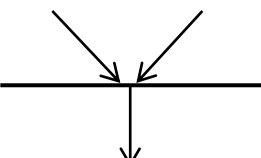
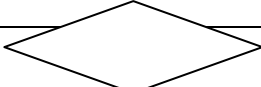
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.
	<i>Include</i> , merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

(Sumber : Windu Gata & Grace Gata; 2013 : 4)

2. Diagram Aktivitas (*Activity Diagram*)

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram*, yaitu :

Tabel II.3. Simbol *Activity Diagram*

Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan pararel menjadi satu.
	<i>Join</i> (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan pilihan untuk

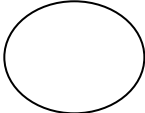
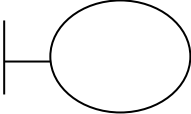
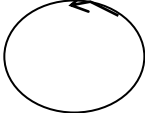

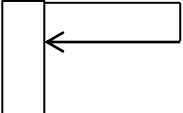
	pengambilan keputusan, <i>true, false</i> .
New Swimline	<i>Swimlane</i> , pembagian <i>activity</i> diagram untuk menunjukkan siapa melakukan apa.



(Sumber : Windu Gata & Grace Gata ; 2013 : 6)

3. Diagram Urutan (*Sequence Diagram*)

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram*, yaitu :

Tabel II.4. Simbol *Sequence Diagram*

Gambar	Keterangan
	<i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry dan <i>form</i> cetak.
	<i>Control class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	<i>Message</i> , simbol mengirim pesan antar <i>class</i> .
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.

	<i>Activation, activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

(Sumber : Windu Gata & Grace Gata; 2013 : 7)

4. *Class Diagram* (Diagram Kelas)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem.

Class diagram juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class diagram* secara khas meliputi: Kelas (*Class*), Relasi, *Associations*, *Generalization* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operations/Method*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti.

Tabel II.5. *Multiplicity Class Diagram*

Multiplicity	Penjelasan
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 : Windu Gata & Grace Gata ; 2013 : 8)