BAB II
LANDASAN TEORI

2.1. Sistem Pendukung Keputusan

2.1.1. Pengertian Sistem Pendukung Keputusan / *Decision Support System* (DSS)

Menurut Alter (2002) dalam Kusrini (2007), DSS (*Decision Support System*) atau dalam bahasa Indonesia disebut SPK (Sistem Pendukung Keputusan) merupakan sistem informasi interaktif yang menyediakan informasi, pemodelan dan manipulasi data. Sistem ini digunakan untuk membantu pengambilan keputusan dalam situasi yang semiterstruktur dan situasi yang tidak terstruktur, dimana tak seorangpun tahu secara pasti bagaimana keputusan seharusnya dibuat.

2.1.2. Konsep Sistem Pendukung Keputusan/ *Decision Support System* (DSS)

Menurut Kusrini (2007), DSS biasanya dibangun untuk mendukung solusi atas suatu masalah atau untuk mengevaluasi suatu peluang. DSS yang seperti ini disebut aplikasi DSS. Aplikasi DSS digunakan dalam pengambilan keputusan. Aplikasi DSS menggunakan CBIS (*Computer Based Information System*) yang fleksibel, interaktif dan dapat diadaptasi , yang dikembangkan untuk mendukung solusi atas masalah manajemen spesifik yang tidak terstruktur.

Aplikasi DSS menggunakan data, memberikan antarmuka pengguna yang mudah dan dapat menggabungkan pemikiran pengambil keputusan. DSS lebih ditujukan untuk mendukung manajemen dalam melakukan pekerjaan yang bersifat analitis dalam situasi yang kurang terstruktur dengan kriteria yang kurang jelas.

DSS tidak dimaksudkan untuk mengotomatisasikan pengambilan keputusan, tetapi memberikan perangkat interaktif yang memungkinkan pengambil keputusan untuk melakukan berbagai analisis menggunakan model-model yang tersedia.

2.1.3. Tujuan Sistem Pendukung Keputusan / *Decision Support System* (DSS)

Menurut Turban (2005) dalam Kusrini (2007), tujuan dari DSS adalah:

a. Membantu manajer dalam pengambilan keputusan atas masalah semi terstruktur.
b. Memberikan dukungan atas pertimbangan manajer dan bukannya dimaksudkan untuk menggantikan fungsi manajer.
c. Meningkatkan efektivitas keputusan yang diambil manajer lebih daripada perbaikan efisiensinya.
d. Kecepatan komputasi. Komputer memungkinkan para pengambil keputusan untuk melakukan banyak komputasi secara cepat dengan biaya rendah.
e. Peningkatan produktivitas. Pendukung terkomputerisasi bisa mengurangi ukuran kelompok dan memungkinkan para anggotanya untuk berada di berbagai lokasi yang berbeda-beda(menghemat biaya perjalanan).
f. Dukungan kualitas. Komputer bisa meningkatkan kualitas keputusan yang dibuat.
g. Berdaya saing. Persaingan didasarkan tidak hanya pada harga, tetapi juga pada kualitas, kecepatan, kustomasi produk dan dukungan pelanggan.
h. Mengatasi keterbatasan kognitif dalam pemrosesan dan penyimpanan.

2.1.4. Tingkat dukungan Sistem Pendukung Keputusan / *Decision Support System* (DSS) dan keterstrukturan keputusan

Menurut Kusrini (2007), berdasarkan tingkat dukungan DSS dibagi menjadi 6, yaitu:

a. *Retrieve Information Elements*

   Dukungan terendah yang bisa diberikan oleh DSS, yakni berupa akses selektif terhadap informasi.

b. *Analyze Entire File*

   Tahap ini, para manajer diberi akses untuk melihat dan menganalisis *file* secara lengkap.

c. *Prepare Reports from Multiple Files*

   Dukungan ini cenderung dibutuhkan mengingat para manajer berhubungan dengan banyak aktivitas dalam satu momen tertentu.

d. *Estimate Decision Consequences*

Manajer dimungkinkan untuk melihat dampak dari setiap keputusan yang mungkin diambil.

e. *Propose Decision*

Suatu alternatif keputusan bisa disodorkan ke hadapan manajer untuk dipertimbangkan.

f. *Make Decision*

Memberikan sebuah keputusan yang tinggal menunggu legitimasi dari manajer untuk dijalankan.

Menurut Kusrini (2007), keputusan yang diambil untuk menyelesaikan suatu masalah dilihat dari keterstrukturannya bisa dibagi menjadi:

a. Keputusan Terstruktur (*Structured Decision*)

Keputusan terstruktur adalah keputusan yang dilakukan secara berulang-ulang dan bersifat rutin. Keputusan tersebut terutama dilakukan manajemen tingkat bawah.

b. Keputusan Semi terstruktur (*Semi structuted Decision*)

Keputusan semi terstruktur adalah keputusan yang memiliki dua sifat. Sebagian keputusan bisa ditangani oleh komputer, keputusan yang lain tetap harus dilakukan oleh pengambil keputusan. Biasanya keputusan semacam ini diambil oleh manajer level menengah dalam suatu organisasi.

c. Keputusan tak terstruktur (*Unstructured Decision*)

Keputusan tak terstruktur adalah keputusan yang penanganannya rumit karena tidak terjadi berulang-ulang atau tidak selalu terjadi. Keputusan tersebut umumnya terjadi pada manajemen tingkat atas.

2.1.5. Karakteristik Sistem Pendukung Keputusan/ *Decision Support System* (DSS)

Menurut Turban, EE., (2005) dalam Kusrini (2007), karakteristik yang diharapkan ada di DSS antara lain :

a. Dukungan kepada pengambil keputusan, terutama pada situasi semiterstruktur dan tak terstruktur, dengan menyertakan penilaian manusia dan informasi terkomputerisasi.
b. Dukungan untuk semua level manajerial, dari eksekutif puncak sampai manajer lini.
c. Dukungan untuk individu dan kelompok.
d. Dukungan untuk keputusan independen dan/atau sekuensial.
e. Dukungan di semua fase proses pengambilan keputusan: intelegensi, desain, pilihan dan implementasi.
f. Dukungan di berbagai proses dan gaya pengambilan keputusan.
g. Adaptivitas sepanjang waktu.
h. Pengguna merasa seperti di rumah.
i. Peningkatan efektivitas pengambilan keputusan (akurasi, *timelines*, kualitas) ketimbang pada efisiensinya (biaya pengambilan keputusan).
j. Kontrol penuh oleh pengambil keputusan terhadap semua langkah proses pengambilan keputusan dalam memecahkan suatu masalah.
k. Pengguna akhir bisa mengembangkan dan memodifikasi sendiri sistem sederhana.
l. Biasanya, model-model digunakan untuk menganalisis situasi pengambilan keputusan.
m. Akses disediakan untuk berbagai sumber data, format dan tipe, mulai dari sistem informasi geografis (GIS) sampai sistem berorientasi-objek.
n. Dapat digunakan sebagai alat *standalone* oleh seorang pengambil keputusan pada satu lokasi atau didistribusikan di suatu organisasi secara keseluruhan dan di beberapa organisasi sepanjang rantai persediaan.

Karakteristik dari DSS tersebut memungkinkan para pengambil keputusan untuk membuat keputusan yang lebih baik dan lebih konsisten dalam satu cara yang dibatasi oleh waktu.

2.2. Metode Sistem Pendukung Keputusan yang digunakan

Proses penyeleksian bibit buah naga pada Taman Budidaya Buah Naga

Burikan Kudus menggunakan metode *Technique For Others Reference by Similarity to Ideal Solution* (Topsis).

2.2.1. *Technique for Others Reference by Similarity to Ideal Solution (Topsis)*

TOPSIS (*Technique For Others Reference by Similarity to Ideal Solution*) adalah salah satu metode pengambilan keputusan multikriteria yang pertama kali diperkenalkan oleh Yoon dan Hwang (1981). TOPSIS menggunakan prinsip bahwa alternatif yang terpilih harus mempunyai jarak terdekat dari solusi ideal positif dan terjauh dari solusi ideal negatif dari sudut pandang geometris dengan menggunakan jarak Euclidean untuk menentukan kedekatan relatif dari suatu alternatif dengan solusi optimal.

Solusi ideal positif didefinisikan sebagai jumlah dari seluruh nilai terbaik yang dapat dicapai untuk setiap atribut, sedangkan solusi negatif-ideal terdiri dari seluruh nilai terburuk yang dicapai untuk setiap atribut. TOPSIS mempertimbangkan keduanya, jarak terhadap solusi ideal positif dan jarak terhadap solusi ideal negatif dengan mengambil kedekatan relatif terhadap solusi ideal positif. Berdasarkan perbandingan terhadap jarak relatifnya, susunan prioritas alternatif bisa dicapai.

Metode ini banyak digunakan untuk menyelesaikan pengambilan keputusan secara praktis. Hal ini disebabkan konsepnya sederhana dan mudah dipahami, komputasinya efisien,dan memiliki kemampuan mengukur kinerja relatif dari alternatif-alternatif keputusan.

Prosedur TOPSIS mengikuti langkah-langkah sebagai berikut:

a. Membuat matriks keputusan yang ternormalisasi
b. Membuat matriks keputusan yang ternormalisasi terbobot
c. Menentukna matriks solusi ideal positif dan matriks solusi ideal negatif
d. Menentukan jarak antara nilai setiap alternatif dengan matriks solusi ideal positif dan matriks solusi ideal negatif
e. Menentukan nilai preferensi untuk setiap alternative

Bobot Kriteria :

Alternatif / Kriteria	Bobot Kriteria (W)
Luas rumah	25
MCK	20
Lantai rumah	20
Bahan bangunan	15
Lingkungan	10
Penerima bantuan	10

TOPSIS membutuhkan rating kinerja setiap alternatif A_i pada setiap kriteria C_i yang ternormalisasi, yaitu :

$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^{m} x_{ij}^2}}$; dengan i=1,2,…,m; dan j=1,2,…,n. ....……(2.1)

Solusi ideal positif A^+ dan solusi ideal negatif A^- dapat ditentukan berdasarkan rating bobot ternormalisasi (y_{ij}) sebagai:

$y_{ij}=w_i r_{ij}$; dengan i=1,2,…,m; dan j=1,2,…,n ……...(2.2)

$A^+ = (y_1^+, y_2^+, \dots, y_n^+)$; …….....……(2.3)

$A^- = (y_1^-, y_2^-, \dots, y_n^-)$; …....…….....……(2.4)

dengan

$$y^+_j = \begin{cases} \max_i y_{ij} \text{ ; jika j adalah atribut keuntungan} \\ \min_i y_{ij} \text{ ; jika j adalah atribut biaya} \end{cases}$$ ……...…..(2.5)

$$y^-_j = \begin{cases} \min_i y_{ij} \text{ ; jika j adalah atribut keuntungan} \\ \max_i y_{ij} \text{ ; jika j adalah atribut biaya} \end{cases}$$ ……(2.6)

j=1,2,…,n.

Jarak antara alternatif A_i dengan solusi ideal positif dirumuskan sebagai :

$$D_i^+ = \sqrt{\sum_{j=1}^{n}(y_{ij} - y_i^+)^2} \text{ ; i=1,2,…,m.} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(2.7)$$

Jarak antara alternatif A_i dengan solusi ideal negatif dirumuskan sebagai :

$$D_i^- = \sqrt{\sum_{j=1}^{n}(y_{ij} - y_i^-)^2} \text{ ; i=1,2,…,m.} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(2.8)$$

Nilai preferensi untuk setiap alternatif (V_i) diberikan sebagai :

$$V_i = \frac{D_i^-}{D_i^- + D_i^+} \text{ ; i=1,2,…,m.} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(2.9)$$

Nilai V_i yang lebih besar menunjukkan bahwa alternatif A_i lebih dipilih.

Perhitungan Metode Topsis secara manual sebagai berikut :

Cost Benefit	Benefit	Benefit	Benefit	Benefit	Benefit	Benefit
bobot kriteria (W)	25	20	20	15	10	10
alternatif / kriteria	Luas rumah	MCK	Lantai rumah	Bahan bangunan	Lingkungan	Penerima bantuan
Sufi	100	50	50	50	100	75
Maman	75	75	75	75	50	100
Mawar	75	100	50	75	100	75
	Kolom 1	Kolom 2	Kolom 3	Kolom 4	Kolom 5	Kolom 6

Pembagi Kolom 1

Rumus : $\sqrt{\sum_{i=1}^{m} x_{ij}^2}$

=SQRT((C7^2)+(C8^2)+(C9^2))

=SQRT((100^2)+(75^2)+(75^2))

= 145,773797 . . . untuk Kolom 2. . Kolom 6 perhitungannya sama

145,773797	134,62912	103,0776406	117,260394	150	145,7737974
Kolom 1	Kolom 2	Kolom 3	Kolom 4	Kolom 5	Kolom 6

Ternormalisasi (R)

Dengan rumus sebagai berikut :

$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^{m} x_{ij}^2}}$

= C7/C$12

= 100/145,773797

= 0,68599434

Untuk menghitung R yang lainnya perhitungan sama Kolom 1 dengan Kolom 1, dan seterusnya

0,68599434	0,37139068	0,48507125	0,426401433	0,66666667	0,514495755
0,51449576	0,55708601	0,727606875	0,639602149	0,33333333	0,685994341
0,51449576	0,74278135	0,48507125	0,639602149	0,66666667	0,514495755
Kolom 1	Kolom 2	Kolom 3	Kolom 4	Kolom 5	Kolom 6

Terbobot (Y)

Rumus :

$y_{ij}=r_{ij}w_i$

=C14*C$5

= 0,68599434 * (25*1000)

= 17,1498585

Untuk yang lainnya sama : Kolom 1 dikalikan W (bobot kriteria) pada Kolom 1

17,1498585	7,42781353	9,701425001	6,396021491	6,66666667	5,144957554
12,8623939	11,1417203	14,5521375	9,594032236	3,33333333	6,859943406
12,8623939	14,8556271	9,701425001	9,594032236	6,66666667	5,144957554
Kolom 1	Kolom 2	Kolom 3	Kolom 4	Kolom 5	Kolom 6

A^+ (Solusi ideal positif)

A^+ pada Kolom 1 mencari nilai terbesar dari Kolom 1 pada Y (Terbobot), sesuai warna berikut :

17,1498585	14,8556271	14,5521375	9,594032236	6,66666667	6,859943406
Kolom 1	Kolom 2	Kolom 3	Kolom 4	Kolom 5	Kolom 6

A^- (Solusi ideal negatif)

A^- pada Kolom 1 mencari nilai terkecil dari Kolom 1 pada Y (Terbobot), sesuai warna berikut :

12,8623939	7,42781353	9,701425001	6,396021491	3,33333333	5,144957554
Kolom 1	Kolom 2	Kolom 3	Kolom 4	Kolom 5	Kolom 6

Menghitung D^+ , D^-

D^+ (Jarak antara alternatif A_i dengan solusi ideal positif)

Rumus : $D_i^+ = \sqrt{\sum_{j=1}^{n}(y_{ij} - y_i^+)^2}$

=SQRT((C19-C$24)^2+(D19-D$24)^2+(E19-E$24)^2+(F19-F$24)^2+(G19-G$24)^2+(H19-H$24)^2)

=SQRT((17,1498585-(17,1498585*1000)^2+(7,42781353-(14,8556271 *1000)^2+(9,701425001-(14,5521375*1000)^2+(6,396021491 -(9,594032236*1000)^2+(6,66666667-(6,66666667*1000)^2+(5,144957554 -(6,859943406*1000)^2)

= 9,584898

Untuk baris 2 dan 3 perhitungannya sama

D^- (Jarak antara alternatif A_i dengan solusi ideal negatif)

Rumus : $D_i^- = \sqrt{\sum_{j=1}^{n}(y_{ij} - y_i^-)}$

=SQRT((C19-C$25)^2+(D19-D$25)^2+(E19-E$25)^2+(F19-F$25)^2+(G19-G$25)^2+(H19-H$24)^2)

=SQRT((17,1498585-(12,8623939*1000)^2+(7,42781353-(7,42781353 *1000)^2+(9,701425001-(9,701425001*1000)^2+(6,396021491 -(6,396021491*1000)^2+(6,66666667-(3,33333333*1000)^2+(5,144957554 -(5,144957554*1000)^2)

= 5,695142

Untuk baris 2 dan 3 perhitungannya sama

	D+	D-
Baris 1	9,584898	5,695142
Baris 2	6,579253	6,895635
Baris 3	6,697234	8,913584

Langkah terakhir adalah mencari (V) nilai preferensi untuk setiap alternatif

Rumus : $V_i = \frac{D_i^-}{D_i^- + D_i^+}$

V1 =K19/(K19+J19)

= 5,695142/(5,695142+ 9,584898)

= 0,372718 (Sufi)

V2 =K20/(K20+J20)

= 6,895635/(6,895635+ 6,579253)

= 0,51174 (Maman)

V3 =K21/(K21+J21)

= 8,913584/(8,913584+ 6,697234)

= 0,570988 (Mawar)

	V	Hasil
V1	0,372718	Sufi
V2	0,51174	Maman
V3	0,570988	Mawar

Dari perhitungan di atas maka yang akan dipilih adalah Mawar karena memiliki nilai V terbesar.

2.3. Landasan Teori Yang Berkaitan dengan Tema

2.3.1. Pengertian Taman Budidaya

Taman Budidaya adalah suatu taman atau tempat yang digunakan untuk kegiatan terencana pemeliharaan sumber daya hayati yang dilakukan secara real lahan untuk diambil manfaat/hasil panennya. Kegiatan budidaya dapat dianggap

sebagai inti dari usaha tani. Menurut Kamus Besar Bahasa Indonesia, budi daya adalah usaha yang bermanfaat dan memberikan hasil.
(Sumber: http://id.wikipedia.org/wiki/Budi_daya)

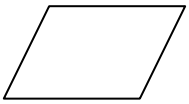
2.3.2. Pengertian buah naga

Buah Naga (inggris:pitaya) adalahbuah dari beberapa jenis kaktus dari marga *Hyloceravus* dan *Selenicereus.* Buah ini berasal dari Meksiko, Amerika Tengah dan Amerika Selatan namun sekarang juga dibudidayakan di Negara-negara Asia seperti Taiwan, Vietnam, Filipina, Indonesia dan Malaysia.
(Sumber: http://id.wikipedia.org/wiki/Buah_naga)

2.4. Bagan Alir Dokumen

Menurut Jogiyanto HM (2005), bagan alir dokumen (*document flowchart*) atau disebut juga bagan alir formulir (*form flowchart*) atau *paperwork flowchart* merupakan bagan alir yang menunjukkan arus dari laporan dan formulir termasuk tembusan-tembusannya. Bagan alir dokumen ini menggunakan simbol-simbol yang sama dengan yang digunakan di dalam bagan alir sistem. Adapun penjelasan dari simbol-simbol yang digunakan dapat dilihat pada table 2.1.

Tabel 2.1: Simbol bagan alir dokumen

No	Nama	Gambar	Fungsi
1.	Simbol titik terminal		Simbol ini digunakan untuk menunjukkan awal dan akhir suatu proses.
2.	Simbol dokumen		Menunjukkan dokumen input dan output baik untuk proses manual, mekanik atau komputer.
3.	Simbol kegiatan manual		Menunjukkan pekerjaan manual.
4.	Simbol *input/output*		Digunakan untuk mewakili data *input/output.*

5.	Simbol *keyboard*		Menunjukkan input yang menggunakan *on-line keyboard*.
6.	Simbol proses		Digunakan untuk mewakili suatu proses.

Bersambung

Tabel Lanjutan

7.	Simbol simpan *offline*	N A C	File non komputer yang diarsip urut angka (*numerical*). File non komputer yang diarsip urut huruf (*alphabetic*). File non computer yang diarsip urut tanggal (*cronological*).
8.	Simbol keputusan		Simbol keputusan (*decision symbol*) digunakan untuk suatu penyeleksian kondisi di dalam program.
9.	Simbol garis alir		Menunjukkan arus dari suatu proses.
10.	Simbol *connector*		Simbol penghubung (*connector symbol*) digunakan untuk menunjukkan sambungan dari bagan alir yang terputus di halaman yang masih sama atau di halaman lainnya.

Sumber: Jogiyanto HM (2005)

2.5. UML (*Unified Modeling Language*)

Menurut Munawar (2005), UML (*Unified Modelling Language*) adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi pada obyek. Hal ini disebabkan karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk baku, mudah dimengerti serta dilengkapi

dengan mekanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan mereka dengan yang lain.

UML adalah sistem notasi yang sudah dibakukan di dunia pengembangan sistem, hasil kerja bersama dari Grady Booch, James Rumbaugh dan Ivar Jacobson. UML yang terdiri dari serangkaian diagram memungkinkan bagi sistem analis untuk membuat cetak biru sistem yang komprehensif kepada klien, programmer dan tiap orang yang terlibat dalam proses pengembangan tersebut. Cetak biru ini akan bisa diketahui informasi detil tentang *coding* program (*forward engineering*) atau bahkan membaca programdan menginterpretasikannya kembali ke dalam diagram (*reverse engineering*). *Reverse engineering* sangat berguna pada situasi dimana *code* program yang tidak terdokumentasi akan dimodifikasi/dipelihara. Hal ini bisa terjadi ketika dokumentasi asli hilang atau bahkan belum dibuat sama sekali.

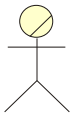
Sangat penting untuk bisa mengeluarkan semua diagram-diagram UML, karena setiap diagram bisa mewakili *stakeholder* yang berbeda di sistem tersebut. UML bisa menceritakan 'apa' yang seharusnya dilakukan oleh sebuah sistem, bukan 'bagaimana' yang seharusnya dilakukan oleh sebuah sistem.

2.5.1. Notasi-notasi dalam diagram UML

2.5.1.1. Notasi-notasi *use case diagram*

Berdasarkan hasil rangkuman dari buku yang berjudul Pemodelan Sistem Informasi Berorientasi Objek dengan UML, penulis Sholiq (2006), notasi-notasi *use case diagram* dapat dilihat pada tabel 2.2.

Tabel 2.2: Notasi-notasi *Use Case Diagram*

Notasi	Nama Notasi	Keterangan
	**Aktor Bisnis**	Aktor bisnis atau *business actor* adalah seseorang atau sesuatu yang ada di luar organisasi dan berinteraksi dengan organisasi yang terlibat dalam kegiatan bisnis organisasi.

	Pekerja Bisnis	Pekerja bisnis atau *business worker* adalah suatu peranan di dalam organisasi, bukan posisi. Seseorang boleh memainkan banyak peran tetapi hanya memegang hanya satu posisi.

Bersambung

Tabel Lanjutan

	***Use case* Bisnis**	*Use case* bisnis atau *business use case* adalah model yang digunakan untuk menggambarkan proses bisnis organisasi.
	Aktor	Aktor adalah seseorang atau apa saja yang berhubungan dengan sistem yang sedang dibangun.
	***Use case* Sitem**	*Use case* sistem adalah bagian tingkat tinggi dari fungsionalitas yang disediakan oleh sistem.
<<include>>	**Relasi *Include***	Relasi *include* menyatakan bahwa satu *use case* selalu menggunakan fungsionalitas yang disediakan oleh *use case* lainnya.
<<extend>>	**Relasi *Extend***	Relasi *extend* memungkinkan satu *use case* secara opsional menggunakan fungsionalitas yang disediakan oleh *use case* lainnya.
	Unidirectional Association	*Unidirectional association* atau relasi assosiasi searah adalah relasi antara aktor dan *use case*. Ia mengindikasikan bahwa aktor tertentu berkomunikasi terhadap fungsionalitas yang disediakan dalam *use case*. Satu anak panah

		menunjukkan navigasi satu arah.
	Bidirectional Association	*Bidirectional association* atau relasi assosiasi dua arah adalah relasi antara aktor dan *use case*. Ia mengindikasikan bahwa aktor tertentu berkomunikasi terhadap fungsionalitas yang disediakan dalam *use case*. Tanpa anak panah menspesifikasikan navigasi dua arah yang menghasilkan *coupling* lebih kuat.

Bersambung

Tabel Lanjutan

	Dependensi	Perubahan satu elemen dapat mempengaruhi elemen yang bergantung padanya.
	Relasi Generalisasi	Relasi generalisasi digunakan ketika ada dua atau lebih aktor, atau *use case* yang sangat serupa. Pada relasi generalisasi, arah panah menunjukkan dari aktor spesifik ke aktor umum.

Sumber : Sholiq (2006)

2.5.1.2. Notasi-notasi *class diagram*

Berdasarkan hasil rangkuman dari buku yang berjudul Pemodelan Sistem Informasi Berorientasi Objek dengan UML, penulis Sholiq (2006), notasi-notasi *class diagram* dapat dilihat pada tabel 2.3.

Tabel 2.3: Notasi-notasi *Class Diagram*

Notasi	Nama Notasi	Keterangan
NamaKelas DaftarAtribut DaftarOperasi()	**Kelas (*Class*)**	Kelas adalah sebuah kategori yang membungkus informasi dan perilaku. Bagian paling atas pada notasi kelas digunakan sebagai nama kelas dan secara

		opsional juga dapat disertakan *stereotype*-nya. Bagian tengah digunakan untuk mendeklarasikan atribut dan bagian paling bawah digunakan mendeklarasikan operasi.
	Bidirectional Association	Assosiasi adalah koneksi yang saling terkait antar kelas yang ada. Tanpa anak panah menspesifikasikan navigasi dua arah yang menghasilkan *coupling* lebih kuat.

Bersambung

Tabel Lanjutan

	Relasi Dependensi	Dependensi juga menghubungkan dua buah kelas, tetapi dengan cara yang sedikit berbeda dibanding assosiasi, dimana relasi dependensi selalu searah (*undirectional*) dan menunjukkan bahwa suatu kelas tidak instanisasi (sebagai variabel instan) oleh kelas lain. Relasi dependensi diperlukan ketika suatu kelas diperlukan sebagai parameter atau nilai balik dalam operasi suatu kelas tertentu.
	Relasi Aggregasi	Aggregasi merupakan bentuk yang lebih kuat dari assosiasi. Sebuah aggregasi adalah relasi antara “keseluruhan” dengan “bagian”.
	Relasi Realisasi	Relasi realisasi digunakan untuk relasi antara kelas dan *interface*-nya, paket dan *interface*-nya, suatu komponen dan *interface*-nya, atau antar sebuah *use case* dan *use case realization*. Relasi ini menunjukkan sebuah *interface* dan implementasinya.

	Relasi Generalisasi	Generalisasi digunakan memperlihatkan relasi pewarisan sifat antara dua model elemen (aktor, *use case*, kelas, atau paket). Pewarisan memperbolehkan suatu kelas untuk mewarisi semua atau sebagian atribut, operasi, relasi dan elemen model lain yang terkait.
	Unidirectional Association	Relasi searah (*unidirectional*) adalah koneksi yang saling terkait antar kelas yang ada.

Sumber : Sholiq (2006)

Dalam *class diagram* terdapat m*ultiplicity* yang mengindikasikan berapa banyak obyek satu kelas terelasi ke obyek tunggal di kelas lainnya pada satu waktu.

Tabel 2.4: Notasi *Multiplicity* dalam UML

Multiplicity	**Arti**
*	Banyak
0	Nol
1	Satu
0..*	Antara nol sampai banyak
1..*	Antara satu sampai banyak
0..1	Nol atau satu
1..1	Tepat satu

Sumber : Sholiq (2006)

2.5.1.3. Notasi-notasi *sequence diagram*

Berdasarkan hasil rangkuman dari buku yang berjudul Pemrograman Visual, penulis Munawar (2005), notasi-notasi *sequence diagram* dapat dilihat pada tabel 2.5.

Tabel 2.5: Notasi-notasi *Sequence Diagram*

Notasi	**Nama Notasi**	**Keterangan**

	Obyek/*Participant*	Obyek yang diatur dalam urutan dari kiri ke kanan. Mereka diatur guna menyederhanakan diagram.
	Lifeline	Garis titik-titik yang terhubung pada setiap *participant.*
	Activation	*Activation* mewakili sebuah eksekusi operasi dari *participant*. Panjang kotak ini berbanding lurus dengan durasi *activation*.
	Message	Sebuah *message* bergerak dari satu *participant* ke *participant* yang lain dan dari satu l*ifeline* ke *lifeline* yang lain.

Bersambung

Tabel Lanjutan

	Recursive	Sebuah obyek yang mempunyai sebuah *operation* kepada dirinya sendiri.
	***Control* (Kontrol)**	Kontrol digunakan untuk mengkoordinasikan kegiatan-kegiatan terhadap kelas lainnya.
	Interface	Satu *set operation* yang memberikan Spesifikasi beberapa aspek dari perilaku dan operation di suatu class ke class yang lain.

Sumber : Munawar (2005)

2.5.1.4. Notasi-notasi *Activity Diagram*

Berdasarkan hasil rangkuman dari buku yang berjudul Pemrograman Visual, penulis Munawar (2005), notasi-notasi *activity diagram* dapat dilihat pada tabel 2.6.

Tabel 2.6: Notasi-notasi *Activity Diagram*

Simbol	Keterangan
	Titik Awal

	Titik Akhir
	Activity
	Pilihan Untuk mengambil Keputusan
	Fork; Digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	Rake; Menunjukkan adanya dekomposisi
	Tanda Waktu
	Tanda pengiriman
	Tanda penerimaan
	Aliran akhir (*Flow Final*)

Sumber : Munawar (2005)

2.5.1.5. Notasi-notasi *Statechart Diagram*

Berdasarkan hasil rangkuman dari buku yang berjudul Pemodelan Sistem Informasi Berorientasi Objek dengan UML, penulis Sholiq (2006), notasi-notasi *sequence diagram* dapat dilihat pada tabel 2.7.

Tabel 2.7: Notasi-notasi *Statechart Diagram*

Notasi	Nama Notasi	Keterangan
NewState	**Keadaan (*State*)**	Keadaan / *state* adalah kondisi yang mungkin dialami oleh suatu obyek.
	Transisi	Transisi adalah sebuah pergerakan dari satu keadaan / *state* ke keadaan / *state* lainnya.
	Transisi Refleksif	Transisi model ini tidak merubah keadaan / *state* (bergerak balik ke keadaan / *state* itu sendiri)
	Start State	Keadaan ini digunakan untuk memulai

		diagram *statechart*.
	End State	Keadaan ini digunakan untuk mengakhiri diagram. *End state* dibuat lebih dari satu untuk satu diagram *statechart*, dimana dibuat sesuai kebutuhan.

Sumber : Sholiq (2006)

2.5.2. Tujuan UML

Menurut Prabowo Pudjo Widodo dan Herlawati (2011), tujuan UML ialah:

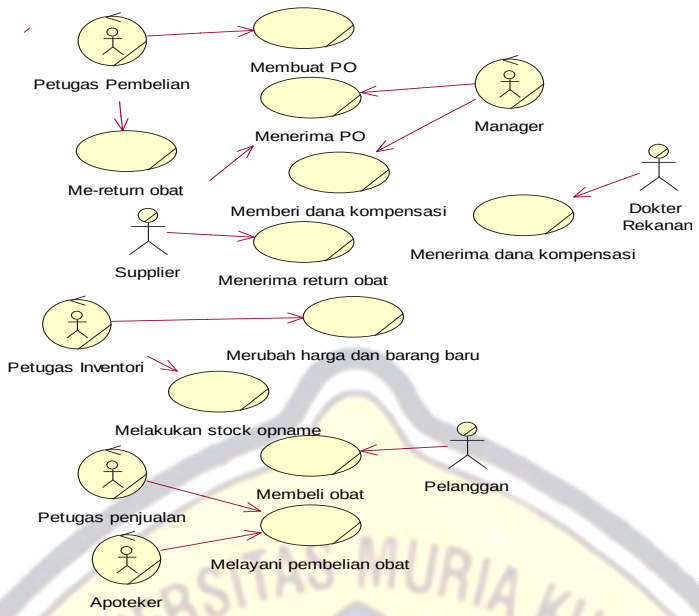
a. Merancang perangkat lunak.
b. Sarana komunikasi antara perangkat lunak dengan proses bisnis.
c. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
d. Mendokumentasikan sistem yang ada, proses-proses dan organisasinya.

2.5.3. Bangunan dasar UML

a. *Business Use Case Diagram* (Diagram *Use Case* Bisnis)

Menurut Sholiq (2006), pemodelan bisnis (*business use case diagram*) mencoba memahami apa yang ada di dalam dan di luar bisnis dan bagaimana yang ada di dalam dan di luar organisasi berkomunikasi satu sama lain. Informasi-informasi ini akan didokumentasikan ke suatu model bisnis. *Business Use Case Diagram* disajikan pada gambar 2.1.

Contoh:

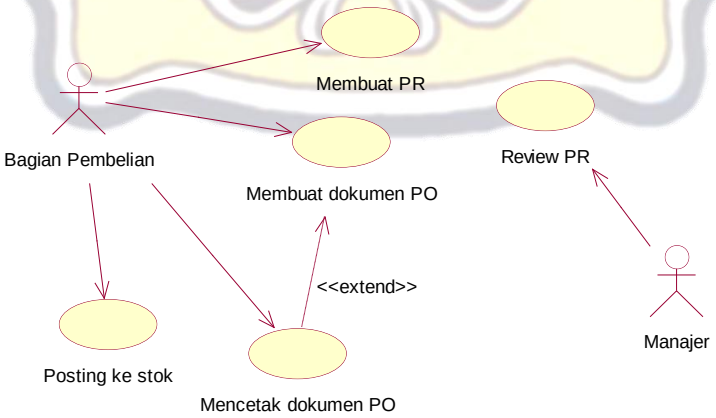


Gambar 2.1: *Business Use Case Diagram* (Sumber: Sholiq, 2006)

b. *System Use Case Diagram* (Diagram Use Case Sistem)

Menurut Sholiq (2006), *system use case diagram* menunjukkan beberapa *use case* dalam sistem, beberapa aktor dalam sistem dan relasi antar mereka. *Use case* adalah potongan fungsionalitas tingkat tinggi yang akan disediakan oleh sistem. Aktor adalah seseorang atau sesuatu yang berinteraksi terhadap sistem yang akan dibangun. Diagram *use case* mempunyai tujuan khusus: untuk mendokumentasikan beberapa aktor (sesuatu yang ada di luar sistem), beberapa *use case* (sesuatu di dalam sistem) dan hubungan antar mereka. *System use case diagram* disajikan pada gambar 2.2.

Contoh :

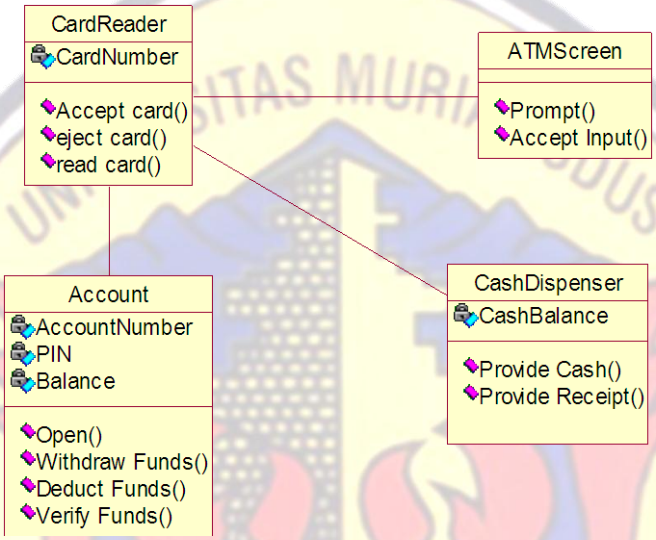


Gambar 2.2: *System Use Case Diagram* (Sumber: Sholiq, 2006)

c. *Class Diagram* (Diagram Kelas)

Menurut Sholiq (2006), diagram kelas digunakan untuk menampilkan kelas-kelas atau paket-paket di dalam sistem dan relasi antar mereka. Ia memberikan gambaran sistem secara statis. Diagram kelas adalah alat perancangan terbaik untuk tim pengembang perangkat lunak. Diagram tersebut membantu pengembang mendapatkan struktur sistem sebelum menuliskan kode program, membantu untuk memastikan bahwa sistem adalah rancangan terbaik. *Class Diagram* disajikan pada gambar 2.3.

Contoh :



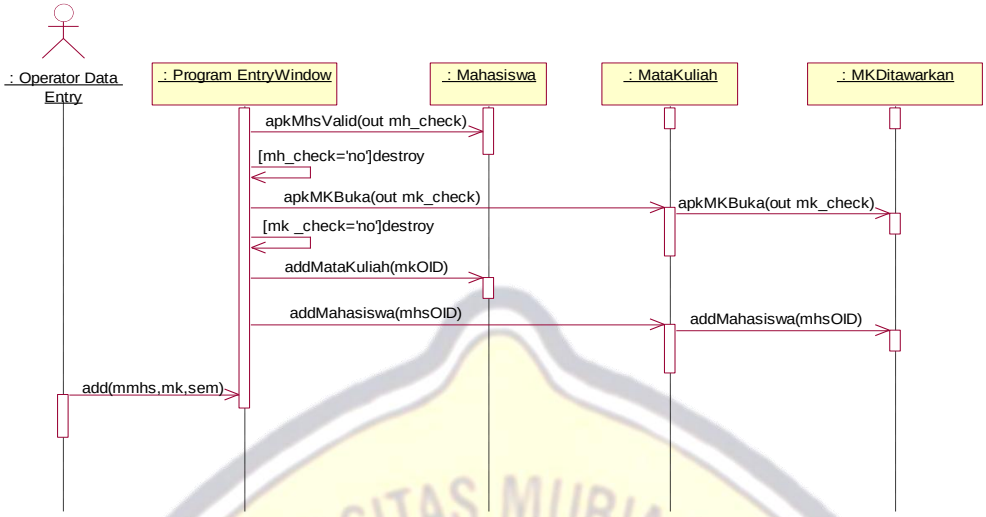
Gambar 2.3: *Class Diagram* (Sumber: Sholiq, 2006)

d. *Sequence Diagram* (Diagram Sekuensial)

Menurut Munawar (2005), *sequence diagram* digunakan untuk menggambarkan perilaku pada sebuah *scenario*. Diagram ini menunjukkan sejumlah contoh obyek dan *message* (pesan) yang diletakkan diantara obyek-obyek ini di dalam *use case*.

Komponen utama *sequence diagram* terdiri atas obyek yang dituliskan dengan kotak segiempat bernama. *Message* diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan progress vertikal. *Sequence Diagram* disajikan pada gambar 2.4.

Contoh :

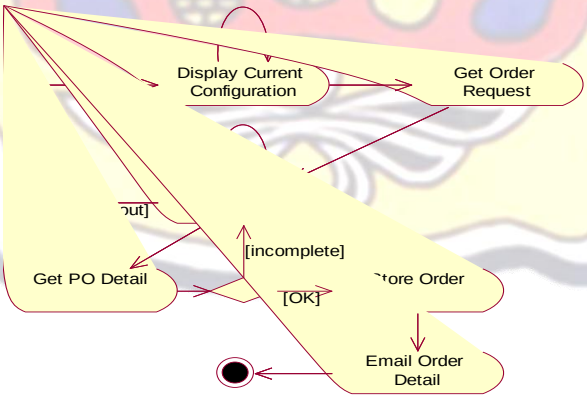


Gambar 2.4: ***Sequence Diagram*** (Sumber: Munawar (2005))

e. ***Activity Diagram***

Menurut Munawar (2005), *activity diagram* adalah teknik untuk mendiskripsikan logika prosedural, proses bisnis dan aliran kerja dalam banyak kasus. *Activity diagram* mempunyai peran seperti halnya *flowchart*, akan tetapi perbedaannya dengan *flowchart* adalah *activity diagram* bisa mendukung perilaku paralel sedangkan *flowchart* tidak bisa. *Activity Diagram* disajikan pada gambar 2.5.

Contoh :



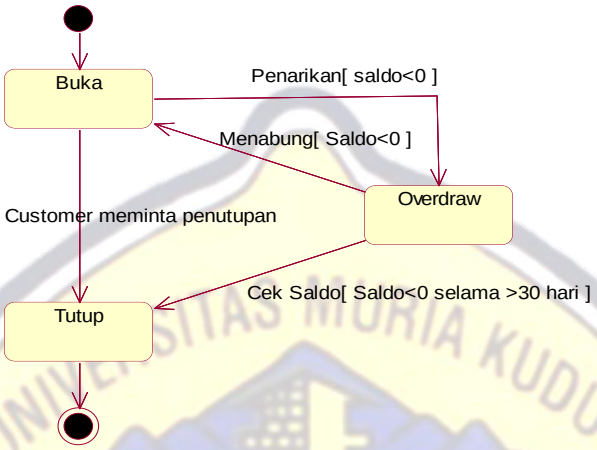
Gambar 2.5: ***Activity Diagram*** (Sumber: Munawar (2005)

f. ***Statechart Diagram***

Menurut Sholiq (2006), diagram *statechart* atau *statechart diagram* menyediakan sebuah cara untuk memodelkan bermacam-macam keadaan yang

mungkin dialami oleh sebuah obyek. Jika dalam diagram kelas menunjukkan gambaran statis kelas-kelas dan relasinya, diagram *statechart* digunakan untuk memodelkan tingkah laku dinamik sistem. *Statechart Diagram* disajikan pada gambar 2.6.

Contoh :



Gambar 2.6: *Statechart Diagram* (Sumber: Sholiq (2006)

2.6. *Entity Relationship Diagram* (ERD)

Menurut Nugroho.A. (2011), model E-R pertama kali diperkenalkan oleh Chen (1976) pada artikelnya yang mendiskusikan konstruksi utama dari model E-R — entitas, hubungan antarentitas (*relationship*), serta atribut-atribut yang bersesuaian dengan tiap entitas. Model yang diperkenalkan oleh Chen itu kemudian diperluas dan dikembangkan oleh Teorey, Yang, Fry (1986), serta Storey (1991). Model E-R digunakan untuk mengonstruksi model data konseptual, yang mencerminkan struktur dan batasan dari basis data, yang mandiri dari perangkat lunak pengelola basis data (DBMS) dan berhubungan erat dengan model data yang langsung bisa digunakan untuk mengimplementasikan basis data secara logika maupun secara fisik dengan DBMS yang dipilih pada tahapan implementasi.

Model E-R (ERD) terdiri atas tiga konsep dasar, yaitu entitas, hubungan antarentitas/relasi, serta atribut.

Elemen ERD adalah sebagai berikut:

a. Entitas

Entitas adalah sesuatu atau objek di dunia nyata (*real world*) yang dapat dibedakan dengan sesuatu atau objek lainnya. Entitas bisa bersifat konseptual/abstrak atau nyata hadir di dunia nyata. Entitas adalah orang, tempat, objek, kejadian (*event*), atau konsep dalam lingkup pengguna yang oleh organisasi/perusahaan perlu dipelihara datanya.

Kebanyakan entitas dalam suatu organisasi dapat digolongkan sebagai entitas kuat (*strong entity*), yaitu entitas mandiri yang keberadaannya tidak bergantung pada keberadaan entitas yang lainnya. Sebaliknya, entitas lemah (*weak entity*) adalah entitas yang keberadaannya sangat bergantung pada keberadaan entitas yang lainnya. Entitas lemah ini tidak memiliki arti apa-apa dan tidak dikehendaki kehadirannya dalam diagram E-R tanpa kehadiran entitas tempat mereka bergantung (yaitu entitas kuat).

b. Relasi

Relasi adalah hubungan antara suatu himpunan entitas dengan himpunan entitas yang lainnya. Pada penggambaran model E-R, relasi adalah perekat yang menghubungkan suatu entitas dengan entitas lainnya. Relasi digambarkan dengan jajaran genjang yang berisi kata kerja.

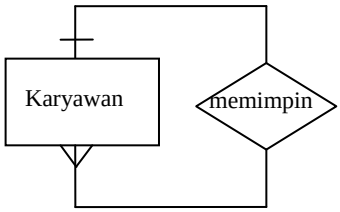
c. Derajat Relasi

Derajat relasi adalah jumlah entitas yang berpartisipasi dalam suatu relasi. Derajat-derajat relasi yang umum dijumpai pada penggambaran diagram E-R adalah satu (*unary*), dua (*binary*), serta tiga (*ternary*) :

1. *Unary Relationship*

Relasi berderajat satu (*unary relationship*) adalah relasi dimana entitas yang terlibat hanya satu (relasi berderajat satu sering disebut juga relasi rekursif/*recursive relationship*). *Unary Relationship* disajikan pada gambar 2.7.

Contoh:



Sumber: Nugroho.A (2011)

Gambar 2.7: Relasi berderajat satu (*unary relationship*)

2. *Binary Relationship*

Relasi berderajat dua, sering juga disebut relasi biner (*binary relationship*), adalah relasi yang melibatkan dua entitas. *Binary Relationship* disajikan pada gambar 2.8.

Contoh:

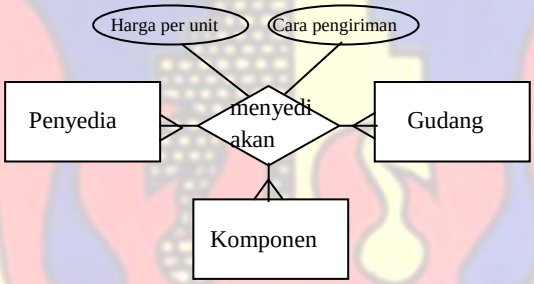
Sumber: Nugroho.A (2011)

Gambar 2.8: Relasi berderajat dua (*binary relationship*)

3. *Ternary Relationship*

Ternary relationship adalah relasi berderajat tiga, yaitu relasi tunggal yang menghubungkan 3 entitas yang berbeda. Relasi berderajat tiga (*ternary relationship*) bisa dikonversikan menjadi entitas assosiatif. *Ternary Relationship* disajikan pada gambar 2.9.

Contoh:



Sumber: Nugroho.A (2011)

Gambar 2.9: Relasi berderajat tiga (*ternary relationship*)

d. Atribut

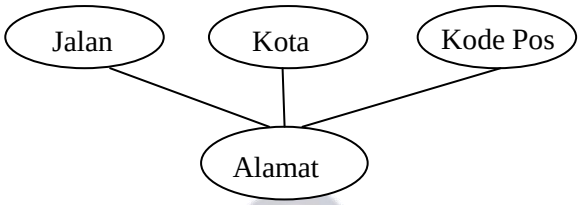
Suatu entitas direpresentasikan dengan sejumlah atribut. Atribut adalah properti deskriptif yang dimiliki oleh setiap anggota dari himpunan entitas. Dalam penggambaran diagram E-R, atribut digambarkan sebagai elips dengan label nama atribut yang bersangkutan.

1. Atribut Komposit

Beberapa atribut dapat dipecah (didekomposisi) menjadi beberapa komponen. Suatu contoh yang paling umum adalah atribut alamat. Alamt ini dapat dipecah menjadi atribut-atribut lain, yaitu jalan, kota, serta kode pos. atribut

yang dapat dipecah menjadi atribut-atribut lainnya disebut atribut komposit. Atribut Komposit disajikan pada gambar 2.10.

Contoh:

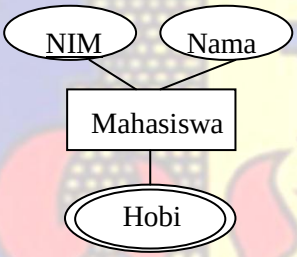


Sumber: Nugroho.A (2011)

Gambar 2.10: Atribut Komposit

2. Atribut Bernilai Banyak
3. Atribut bernilai banyak (*multivalue attribute*) adalah suatu atribut yang memiliki nilai lebih dari satu untuk suatu entitas tertentu. Atribut bernilai banyak digambarkan dengan elips berbatas ganda. Atribut Bernilai Banyak disajikan pada gambar 2.11.

Contoh:



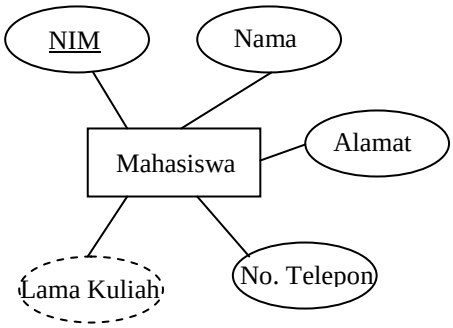
Sumber: Nugroho.A (2011)

Gambar 2.11: Atribut Bernilai Banyak

4. Atribut Turunan

Atribut turunan adalah atribut yang nilainya bisa didapatkan dari nilai atribut yang lainnya. Atribut turunan ini bisa digambarkan dengan elips berbatas putus-putus. Atribut Turunan disajikan pada gambar 2.12.

Contoh:



Sumber: Nugroho.A (2011)

Gambar 2.12: Atribut Turunan

5. Atribut Kunci (*Key*)

Kunci merupakan suatu atribut yang unik yang dapat digunakan untuk membedakan suatu entitas dengan entitas lainnya dalam suatu himpunan entitas.

Superkey adalah satu atau lebih atribut yang dimiliki suatu entitas, yang dapat dibedakan untuk membedakakn dengan entitas lainnya. Contohnya NIM.

Atribut-atribut yang mungkin dapat digunakan sebagai kunci dengan batasan dinamakan kunci calon (*candidate key*). Contohnya gabungan atribut nama, alamat, serta nomor telepon dari entitas mahasiswa. Atribut NIM, kemudian gabungan atribut NIM dan nama, dan gabungan atribut-atribut nama, alamat, nomor telepon, merupakan kunci calon.

Atribut minimal yang dapat digunakan untuk basis pembedaan suatu entitas dengan entitas lainnya disebut kunci primer (*primary key*). Contohnya NIM. NIM memadai untuk digunakan sebagai kunci primer karena unik dan cukup ringkas untuk digunakan sebagai pembeda satu entitas dengan entitas lainnya dalam himpunan entitas mahasiswa.

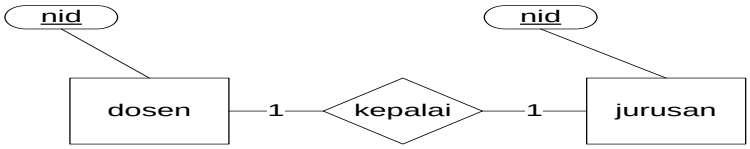
e. Kardinalitas Relasi

Definisi kardinalitas relasi menurut Al Bahra Bin Ladjamudin (2005:147) menyatakan bahwa kardinalitas relasi menunjukkan jumlah maksimum tupel yang dapat berelasi dengan entitas pada entitas yang lain. Terdapat 3 macam kardinalitas relasi yaitu sebagai berikut:

1. *One to One*

Tingkat hubungan ini menunjukkan hubungan satu ke satu, dinyatakan dengan satu kejadian pada entitas pertama, dan hanya mempunyai satu hubungan dengan satu kejadian pada entitas yang kedua dan sebaliknya. Diagram Kardinalitas *One to One* disajikan pada gambar 2.13.

Contoh:



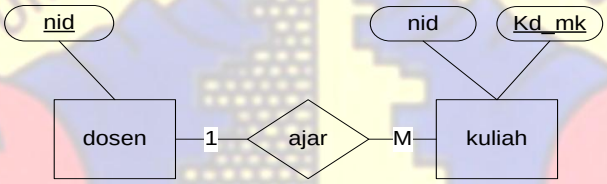
Sumber: Ladjamudin.AB (2005)

Gambar 2.13: Diagram Kardinalitas *One to One*

2. *One to Many* atau *Many to One*

Tingkat hubungan satu ke banyak adalah sama dengan banyak ke satu, tergantung dari arah mana hubungan tersebut dilihat. Untuk satu kejadian pada entitas yang pertama dapat mempunyai banyak hubungan dengan kejadian pada entitas yang kedua. Sebaliknya, satu kejadian pada entitas yang kedua hanya dapat mempunyai satu hubungan dengan satu kejadian pada entitas yang pertama. Diagram Kardinalitas *One to Many* disajikan pada gambar 2.14.

Contoh:



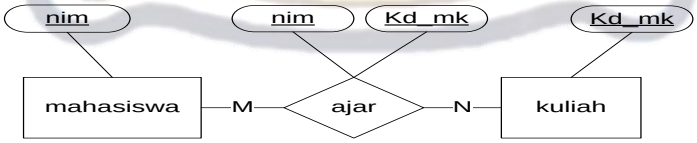
Sumber: Ladjamudin.AB (2005)

Gambar 2.14: Diagram Kardinalitas *One to Many*

3. *Many to Many*

Tingkat hubungan banyak ke banyak terjadi jika tiap kejadian pada sebuah entitas akan mempunyai banyak hubungan dengan kejadian pada entitas lainnya, dilihat dari sisi entitas yang pertama maupun dilihat dari sisi yang kedua. Diagram Kardinalitas *Many to Many* disajikan pada gambar 2.15.

Contoh:



Sumber: Ladjamudin.AB (2005)

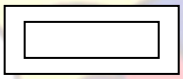
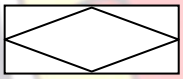
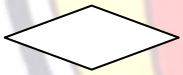
Gambar 2.15: Diagram Kardinalitas *Many to Many*

Menurut Al-Bahra Bin Ladjamudin (2005), langkah-langkah teknis yang dapat dilakukan untuk menghasilkan Diagram E-R adalah sebagai berikut:

a. Mengidentifikasi dan menetapkan seluruh entitas yang akan terlibat.
b. Menentukan atribut-atribut key (primary key) dari masing-masing entitas.
c. Mengidentifikasi dan menetapkan derajat/kardinalitas seluruh relasi diantara entitas-entitas yang ada beserta foreign key-nya.
d. Melengkapi entitas dan relasi dengan atribut-atribut deskriptif.

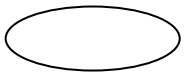
Berdasarkan hasil rangkuman dari buku yang berjudul Perancangan dan Implementasi Sistem Basis Data, penulis Adi Nugroho (2011), simbol-simbol dalam Diagram E-R dapat dilihat pada tabel 2.15.

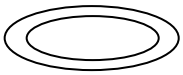
Tabel 2.8: Simbol-simbol ERD

No	Simbol	Nama Simbol	Keterangan
1		Entitas Kuat	Entitas kuat yaitu entitas mandiri yang keberadaannya tidak bergantung pada keberadaan entitas yang lainnya.
2		Entitas Lemah	Entitas lemah yaitu entitas yang keberadaannya sangat bergantung pada keberadaan entitas lainnya.
3		Entitas Assosiatif	Entitas assosiatif yaitu entitas yang terbentuk dari suatu relasi.
4		Relasi	Relasi adalah hubungan antara suatu himpunan entitas dengan himpunan entitas yang lainnya.

Bersambung

Tabel Lanjutan

5		Relasi Pengidentifikasi	Relasi pengidentifikasi adalah relasi antara entitas lemah dan entitas kuat.
6		Atribut	Atribut adalah properti deskriptif yang dimiliki oleh setiap anggota dari himpunan entitas.

7		Atribut Bernilai Banyak	Atribut bernilai banyak (*multivalue attribute*) adalah suatu atribut yang memiliki nilai lebih dari satu untuk suatu entitas tertentu.
8		Atribut Turunan	Atribut turunan adalah atribut yang nilainya bisa didapatkan dari nilai atribut yang lainnya.

Sumber: Nugroho.A (2011)

UNIVERSITAS MURIA KUDUS