

BAB II

TINJAUAN PUSTAKA

II.1. Sistem

Menurut Jogiyanto, Sistem merupakan sekumpulan elemen-elemen yang saling terintegrasi serta melaksanakan fungsinya masing-masing untuk mencapai tujuan tertentu (2005).

Sistem adalah kumpulan atau himpunan dari unsur atau variable-variable yang saling terkait, saling berinteraksi, dan saling tergantung satu sama lain untuk mencapai tujuan. Selain itu, sistem juga dapat didefinisikan sebagai sekumpulan objek-objek yang saling berelasi dan berinteraksi, serta hubungan antar objek bisa dilihat sebagai satu kesatuan yang dirancang untuk mencapai satu tujuan yang telah ditetapkan. (Hamim Tohari ; 2014 : 2)

II.2. Informasi

Ditinjau dari pengertiannya, sistem informasi dapat dianalogikan sebagai sebuah permintaan (demand) dari masyarakat industri, ketika kebutuhan akan sarana pengolahan data dan komunikasi yang cepat dan murah (menembus ruang dan waktu) (2000). Kualitas dari suatu informasi tergantung pada tiga hal yaitu :

1. Akurat, berarti informasi harus bebas dari kesalahan-kesalahan dan tidak bisa atau menyesatkan. Akurat juga berarti informasi harus jelas mencerminkan maksudnya. Informasi harus akurat karena dari sumber informasi sampai ke

penerima informasi kemungkinan banyak terjadi gangguan (*noise*) yang dapat merubah atau merusak informasi tersebut.

2. Tepat pada waktunya, berarti informasi yang pada penerima tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai lagi. Karena informasi merupakan landasan di dalam pengambilan keputusan. Bila pengambilan keputusan tersebut, maka dapat berakibat fatal untuk organisasi. Relevan, berarti informasi tersebut mempunyai manfaat untuk pemakainya. relevansi informasi untuk tiap-tiap orang satu dengan yang lainnya berbeda (Hamim Tohari ; 2014 : 7).

II.3. Sistem Pendukung Keputusan

Sistem Pendukung Keputusan adalah sistem komputer yang mengolah data menjadi informasi untuk mengambil keputusan dari masalah semi- terstruktur yang spesifik. Dengan demikian dapat ditarik satu definisi tentang SPK yaitu sebuah sistem berbasis komputer yang adaptif , fleksibel, dan interaktif yang digunakan untuk memecahkan masalah-masalah tidak terstruktur sehingga dapat meningkatkan nilai keputusan yang diambil Terstruktur, yaitu berhubungan dengan persoalan yang telah diketahui sebelumnya dengan penyelesaian standar aturan yang telah ditentukan. (Beni Irawan;2013:2)

Semi terstruktur, yaitu berhubungan dengan persoalan yang belum diketahui sebelumnya, dengan parameter yang sudah ada. Tidak terstruktur, yaitu berhubungan dengan persoalan baru yang cukup pelik, karena banyaknya data yang belum diketahui. (Beni Irawan;2013:2)

II.3.1. Tujuan sistem pendukung keputusan

1. Membantu manager membuat keputusan untuk memecahkan masalah semi terstruktur.
2. Mendukung penilaian manajer bukan mencoba menggantikannya.
3. Meningkatkan efektivitas pengambilan keputusan manager daripada efisiensinya.

Tujuan dari Sistem Pendukung Keputusan ini untuk mengambil keputusan seefisien mungkin, juga terdapat manfaat utamanya adalah keputusan yang lebih baik Meningkatkan efektivitas pengambilan keputusan manajer terhadap suatu pemecahan masalah. Tujuan dari SPK selain untuk membuat proses pengambilan keputusan seefisien mungkin, juga terdapat manfaat utamanya adalah keputusan yang lebih baik. (Beni Irawan;2013:2)

II.3.2. Tahapan Proses Pengambilan Keputusan

Merupakan suatu proses yang dilaksanakan pengambil keputusan berdasarkan pengetahuan dan informasi yang ada padanya pada saat tersebut dengan harapan bahwa sesuatu akan terjadi. (Beni Irawan;2013:2)

Adapun tahapannya adalah sebagai berikut ini :

1. Identifikasi masalah
2. Pemilihan metode pemecahan masalah.
3. Pengumpulan data yang dibutuhkan untuk melaksanakan model tersebut.
4. Mengimplementasikan model tersebut.
5. Mengevaluasi sisi positif dari setiap alternative yang ada.

6. Melaksanakan solusi terpilih.

II.4. Kualitas Bibit Kelapa Sawit

Tanaman kelapa sawit (*Elaeis guineensis* Jacq) saat ini merupakan salah satu jenis tanaman perkebunan yang menduduki posisi penting disektor pertanian umumnya, dan sektor perkebunan khususnya, hal ini disebabkan karena dari sekian banyak tanaman yang menghasilkan minyak atau lemak, kelapa sawit yang menghasilkan nilai ekonomi terbesar per hektarnya didunia. Melihat pentingnya tanaman kelapa sawit di masa ini dan masa yang akan datang, seiring dengan meningkatnya kebutuhan. (Beni Irawan;2013:1-2)

Bahan tanaman kelapa sawit unggul bisa berasal dari persilangan dari berbagai sumber (inter and intra specific crossing) disamping itu bahan tanaman kelapa sawit unggul juga bisa dihasilkan dari pemulihan pada tingkat molikuler yang diperbanyak secara vegetative dengan teknik kultur jaringan, bahan tanaman kelapa sawit yang umum ditanam diperkebunan komersial yaitu persilangan dura x pisifera (D x P) yang disebut tenera. Pertumbuhan awal bibit merupakan periode kritis yang sangat menentukan keberhasilan tanaman dalam mencapai pertumbuhan yang baik, dipembibitan pertumbuhan dan figur bibit tersebut sangat ditentukan oleh kecambah yang ditanam, Berdasarkan latar belakang permasalahan tersebut maka dalam penelitian ini akan dibuat sebuah Sistem Pendukung Keputusan yang dapat membantu pengelola perkebunan kelapa sawit dalam proses pemilihan bibit – bibit sawit yang berkualitas baik. Dengan dibuatnya sistem ini diharapkan dapat mempercepat dan mempermudah asisten

dilapangan dalam mencari bibit kelapa sawit yang berkualitas baik. (Beni Irawan;2013:1-2)

II.5. Metode Backpropagation

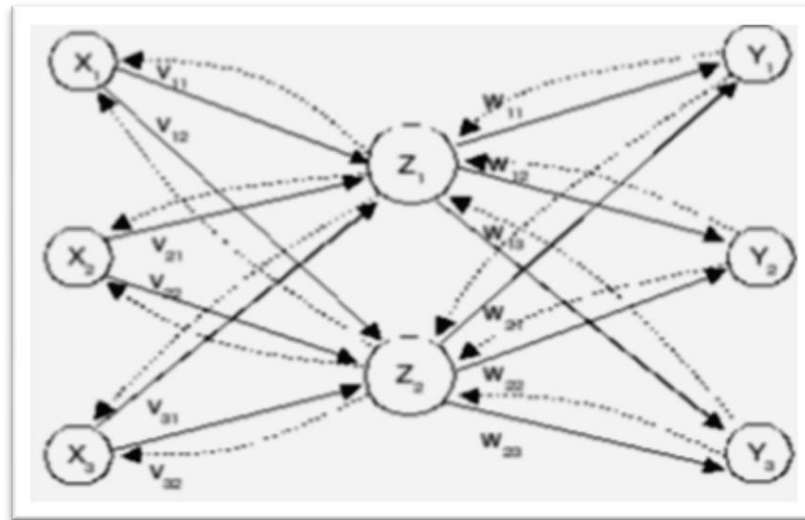
Backpropagation merupakan algoritma pembelajaran yang terawasi dan biasanya digunakan oleh perceptron dengan banyak lapisan untuk mengubah bobot-bobot yang terhubung dengan neuron-neuron yang ada pada lapisan tersembunyinya. Backpropagation merupakan algoritma yang sangat baik dalam menangani masalah pengenalan pola-pola kompleks, dan juga sangat populer. Istilah *backpropagation* atau “propagasi balik” atau “penyiaran kembali” diambil dari cara kerja jaringan ini, yaitu bahwa *gradien error* unit-unit tersembunyi diturunkan dari penyiaran kembali *error-error* yang diasosiasikan dengan unit-unit output. Nama lain dari propagasi balik adalah aturan delta yang digeneralisasi (*generalized delta rule*) Algoritma *backpropagation* menggunakan error output untuk mengubah nilai bobot-bobotnya dalam arah mundur (*backward*). Untuk mendapatkan error ini, tahap perambatan maju (*forward propagation*) harus dikerjakan terlebih dahulu. Pada saat perambatan maju, neuron-neuron diaktifkan dengan menggunakan fungsi aktivasi sigmoid, yaitu:

Algoritma backpropagation terdiri dari :

1. Terdiri dari 3 lapis (Layer) yaitu *Input Layer*, *Hidden Layer*, *Output Layer*.
2. Masing-masing layer mempunyai neuron (*Node*). Dimana setiap neuron pada input dan output layer bergantung pada data yang digunakan. Sedangkan neuron pada *hidden layer* bervariasi.

3. Antara satu Neuron dengan Neuron lainnya saling berhubungan dan mempunyai nilai yang disebut “bobot”.

Arsitektur jaringan ini terdiri dari *input layer*, *hidden layer*, dan *output layer* seperti gambar II.1. berikut:



Gambar II.1. Arsitektur Jaringan Backpropagation
Sumber : (Yani Maulita; 2014 :72)

Algoritma backpropagation:

- Inisialisasi bobot (ambil bobot awal dengan nilai random yang cukup kecil).
- a. Kerjakan langkah-langkah berikut selama kondisi berhenti bernilai FALSE:

Untuk tiap-tiap pasangan elemen yang akan dilakukan pembelajaran, kerjakan:

Feedforward:

- a. Tiap-tiap unit input (X_i , $i=1,2,3,\dots,n$) menerima sinyal x_i dan meneruskan sinyal tersebut ke semua unit pada lapisan yang ada di atasnya (lapisan tersembunyi).
- b. Tiap-tiap unit tersembunyi (Z_j , $j=1,2,3,\dots,p$) menjumlahkan sinyal-sinyal input terbobot:

$$z_{in_j} = v_{oj} + \sum_{i=1}^n x_i v_{ij} \quad \dots\dots\dots (1)$$

gunakan fungsi aktivasi untuk menghitung sinyal outputnya:

$$z_j = f(z_{in_j}) \quad \dots\dots\dots (2)$$

dan kirimkan sinyal tersebut ke semua unit di lapisan atasnya (unit-unit output).

- c. Tiap-tiap unit output $Y_k, k = 1, 2, 3, \dots, m$ menjumlahkan sinyal-sinyal input terbobot.

$$y_{in_k} = w_{ok} + \sum_{i=1}^n z_i w_{jk} \quad \dots\dots\dots (3)$$

gunakan fungsi aktivasi untuk menghitung sinyal outputnya:

$$y_k = f(y_{in_k}) \quad \dots\dots\dots (4)$$

dan kirimkan sinyal tersebut ke semua unit di lapisan atasnya (unit-unit output).

- d. Tiap-tiap unit output ($Y_k, k=1,2,3,\dots,m$) menerima target pola yang berhubungan dengan pola input pembelajaran, hitung informasi errornya:

$$\delta_k = (t_k - y_k) f'(y_{in_k}) \quad \dots\dots\dots(5)$$

kemudian hitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki nilai w_{jk}):

$$\Delta w_{jk} = \alpha \delta_k z_j \quad \dots\dots\dots(6)$$

hitung juga koreksi bias (yang nantinya akan digunakan untuk memperbaiki nilai w_{0k}):

$$F\Delta w_{0k} = \alpha \delta_k \quad \dots\dots\dots (7)$$

kirimkan δ_k ini ke unit-unit yang ada di lapisan bawahnya.

- e. Tiap-tiap unit tersembunyi (Z_j , $j=1,2,3,\dots,p$) menjumlahkan delta inputnya (dari unit-unit yang berada pada lapisan di atasnya):

$$z_{jn_j} = x_{ol} + \sum_{i=1}^n Z_i x_{jk} \quad \dots\dots\dots (8)$$

kalikan nilai ini dengan turunan dari fungsi aktivasinya untuk menghitung informasi error:

$$\delta_j = \delta_{in_j} f'(z_{in_j}) \quad \dots\dots\dots (9)$$

kemudian hitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki nilai v_{ij}):

$$\Delta v_{jk} = \alpha \delta_j x_i \quad \dots\dots\dots (10)$$

hitung juga koreksi bias (yang nantinya akan digunakan untuk memperbaiki nilai v_{0j}):

$$\Delta v_{0j} = \alpha \delta_j \quad \dots\dots\dots (11)$$

- f. Tiap-tiap unit output (Y_k , $k=1,2,3,\dots,m$) memperbaiki bias dan bobotnya ($j=0,1,2,\dots,p$):

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk} \quad \dots\dots\dots (12)$$

Tiap-tiap unit tersembunyi (Z_j , $j=1,2,3,\dots,p$) memperbaiki bias dan bobotnya ($i=0,1,2,\dots,n$):

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij} \quad \dots\dots\dots (13) \text{ (Yani Maulita; 2014 :72)}$$

II.6. Kamus Data

Kamus Data adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi. Kamus data selain digunakan untuk dokumentasi dan mengurangi redundansi, juga dapat digunakan untuk :

1. Memvalidasi diagram aliran data dalam hal kelengkapan dan keakuratan.
2. Menyediakan suatu titik awal untuk mengembangkan layar dan laporan-laporan.
3. Menentukan muatan data yang disimpan dalam file-file.
4. Mengembangkan logika untuk proses-proses diagram aliran data.

Kamus data dibuat pada tahap analisis sistem dan digunakan baik pada tahap analisis maupun pada tahap perancangan sistem. Pada tahap analisis sistem, Kamus Data dapat digunakan sebagai alat komunikasi antara analisis sistem dengan pemakai sistem tentang data yang mengalir di sistem, yaitu tentang data yang masuk ke sistem dan tentang informasi yang dibutuhkan oleh pemakai sistem. Pada tahap perancangan sistem, Kamus Data digunakan untuk merancang input, merancang laporan-laporan dan database.

Kamus Data mendefinisikan elemen data dengan fungsi sebagai berikut:

1. Menjelaskan arti aliran data dan penyimpanan data dalam DFD.
2. Mendeskripsikan komposisi paket data yang bergerak melalui aliran (misalnya alamat diuraikan menjadi kota, negara dan kode pos).
3. Mendeskripsikan komposisi penyimpanan data.
4. Menspesifikasikan nilai dan satuan yang relevan bagi penyimpanan dan aliran.

5. Mendeskripsikan hubungan detail antar penyimpanan (yang akan menjadi titik perhatian dalam *entity-relationship diagram*). (Muhammad Arhami ; 2010 : 7)

Tabel II.1. Simbol-simbol dalam kamus data

No	Simbol	Uraian
1.	=	Terdiri atas, mendefinisikan, diuraikan menjadi, artinya
2.	+	Dan
3.	()	Menunjukkan suatu elemen yang bersifat pilihan (opsional). Elemen-elemen yang bersifat pilihan ini bisa dikosongkan pada layar masukan atau bisa juga dengan memuat spasi atau nol untuk field-field numeric pada struktur file.
4.	{ }	Menunjukkan elemen-elemen repetitive, juga disebut kelompok berulang atau tabel-tabel. Kemungkinan bisa ada satu atau beberapa elemen berulang didalam kelompok tersebut. Kelompok berulang bisa mengandung keadaan-keadaan tertentu, seperti misalnya, jumlah pengulangan yang pasti atau batas tertinggi dan batas terendah untuk jumlah pengulangan.
5	[]	Menunjukkan salah satu dari dua situasi tertentu. Satu elemen bisa ada sedangkan elemen lainnya juga ada, tetapi tidak bisa kedua-duanya ada secara bersamaan. Elemen-elemen yang ada di dalam tanda kurang ini saling terpisah satu sama lain. (dengan kata lain, memilih salah satu dari sejumlah alternatif, seleksi)
6		Pemisah sejumlah alternatif pilihan antara symbol []
7	@	Identifikasi atribut kunci
8	**	Komentar

Sumber : (Muhammad Arhami ; 2010 : 12)

II.7. Teknik Normalisasi

Sebuah rancangan database dapat dikategorikan kurang baik, apabila :

1. Data disimpan di beberapa tempat atau *record*
2. Adanya redudansi atau pengulangan data sehingga memboroskan ruang penyimpanan serta menyulitkan ketika melakukan update atau perubahan data.
3. Timbul nilai null (*null value*)

4. Menimbulkan informasi yang tidak bermanfaat.

oleh karena itu, normalisasi perlu dilakukan. Normalisasi database membuat struktur database yang dihasilkan semakin optimal. Intinya, proses ini menghindarkan pengguna dari redudansi data. (Eko Priyo Utomo ; 2014 :72-73)

II.7.1. Bentuk-bentuk Normalisasi

1. Bentuk normal tahap pertama (1NF)

Aturan yang ada dalam 1NF adalah sebuah baris data tidak dapat terdiri dari kumpulan data yang sama, masing-masing kolom harus mempunyai nilai yang unik, dan masing-masing baris mempunyai *identifier* unik yaitu *primary key*.

Tabel II.2. Normalisasi Bentuk 1 (1NF)

Mhs_ID	Mhs_Nama
201	Amir
202	Agus
203	Bobby
204	Cahya
205	Joko

Matkul_ID	Matkul_Mhs_ID	Matkul_Nama
50	201	Rekayasa Perangkat Lunak
51	201	Komunikasi Data
52	202	Manajemen Proyek
53	203	Rekayasa Perangkat Lunak
54	204	Penambangan Data
55	205	Manajemen Strategis

(Sumber : Eko Priyo Utomo ; 2014 :74)

Penggabungan antara kolom Matkul_Id dan Matkul_Mhs_Id merupakan *primary key*.

2. Bentuk normal tahap kedua (2NF)

Definisi bentuk normal kedua menyatakan bahwa tabel dengan kunci utama gabungan hanya dapat berada pada 1NF, tetapi tidak pada 2NF. Sebuah tabel relasional berada pada bentuk normal kedua jika dia berada pada bentuk normal kedua jika dia berada pada 1NF dan setiap kolom bukan kunci yang sepenuhnya tergantung pada seluruh kolom yang membentuk kunci utama.

Normalisasi bentuk kedua (2NF) harus memenuhi syarat 1NF dan tidak ada ketergantungan *primary key*.

Tabel II.3. Normalisasi Bentuk 2 (2NF)

Pelanggan_ID	Pelanggan_Nama	Order_ID	Order_Nama	Detail
201	Amir	60	Jeruk	Jeruk bali
202	Amir	61	Apel	Apel Malang

(Sumber : Eko Priyo Utomo ; 2014 : 75)

Pada tabel pelanggan diatas, penggabungan antara Pelanggan_Id dan Ordeer_Id merupakan *primary key*. Tabel pelanggan diatas memenuhi persyaratan normalisasi pertama (1NF) namun belum 2NF karena ada ketergantungan parsial kolom pada *primary key*. Kolom Pelanggan_Nama hanya bergantung pada Pelanggan_Id, Order_Nama bergantung pada Order_Id dan tidak ada link/hubungan antara Detail dan Pelanggan_Nama.

Pelanggan_ID	Pelanggan_Nama
201	Amir
202	Amir

Order_ID	Order_Nama
60	Jeruk
61	Apel

Pelanggan_ID	Order_ID	Barang_Detail
201	60	Jeruk bali
202	61	Apel Malang

(Sumber : Eko Priyo Utomo ; 2014 : 76)

3. Bentuk normal tahap ketiga (3NF)

Bentuk normal ketiga mengharuskan semua kolom pada tabel relasional tergantung hanya pada kunci utama. Secara definisi, sebuah tabel berada pada bentuk normal ketiga (3NF) jika tabel sudah berada pada 2NF dan setiap kolom yang bukan kunci tidak tergantung secara transitif pada kunci utamanya.(Eko Priyo Utomo, 2014).

Tabel II.4. Normalisasi Bentuk 3 (3NF)

Mahasiswa_ID	Mahasiswa_Nama	Mahasiswa_KodePos
10	Melani	44123
11	Nania	64111
12	Rosnia	22780

Mahasiswa_KodePos	Mahasiswa_Jalan	Mahasiswa_Kota	Mahasiswa_Kabupaten
44123	Jl. Kucing 12	Condong Catur	Sleman
64111	Jl. Harimau 21	Keras	Kediri
22780	Jl. Pelanduk 33	Nagrek	Sumedan

(Sumber : Eko Priyo Utomo ; 2014 : 77)

II.8. PHP

PHP merupakan salah satu bahasa pemrograman web *server-side* yang populer dan banyak digunakan sampai saat ini. PHP dapat diperoleh secara *free* dengan men-download di situs resmi <http://www.php.net>, atau menggunakan paket instan yang telah terintegrasi dengan berbagai aplikasi untuk kebutuhan server dan database.

PHP dirancang oleh Rasmus Ledrof pada tahun 1994. Awalnya, PHP digunakan untuk mendeteksi *user* yang berkunjung pada situs. Selain sebagai bahasa pemrograman web yang dapat dieksekusi sendiri, PHP juga dikenal sebagai *embedded language* artinya, anda dapat memasukkan kode *PHP* ke dalam bahasa HTML yang dikenal sebagai bahasa pembentuk halaman website.

Sebagai *embedded language*, PHP juga hampir sama dengan bahasa pemrograman lain yang dapat dimasukkan kedalam kode HTML seperti Java Script dan VBScript. Perbedaannya terletak pada tempat eksekusi. Jika Java Script dieksekusi di sisi klien atau browser, PHP dieksekusi di sisi server. Karena eksekusi dilakukan pada sisi server, maka PHP dapat mengambil data, mengolah, dan mengirimkan data dari database server ke halaman browser. (Eko Priyo Utomo ;2015:1-2).

II.9. MySQL

MySQL merupakan salah satu database yang banyak digunakan oleh para pengguna komputer di seluruh dunia. Database MySQL versi terbaru adalah versi 5. Ada banyak fitur serta perbaikan yang ditawarkan antara lain :

1. adanya fitur view atau tabel virtual penyimpanan data yang dapat digunakan untuk mempercepat menampilkan data.
2. Penggunaan function
3. Penggunaan trigger
4. Penggunaan stored procedure

SQL (*structured query language*) merupakan bahasa yang digunakan untuk berinteraksi dengan database. Setiap database menggunakan bahasa SQL dalam pengoperasiannya, tidak terkecuali MySQL. Pada sub bab berikut ini anda akan belajar menggunakan bahasa SQL yang sering digunakan. Pada bahasa SQL, perintah yang digunakan bersifat *case-insensitive*. Sebaiknya anda membiasakan menggunakan huruf kapital untuk perintah SQL. Kebiasaan ini untuk membedakan nama database, nama tabel, atau nama kolom. (Eko Priyo Utomo ;2015:58-59).

II.9.1. Tipe Data di MySQL

pada sebuah tabel terdapat field yang berisi nilai dari data. Nilai data dalam field mempunyai tipe sendiri. Ada beberapa tipe data yang dapat digunakan pada MySQL, antara lain :

1. Tipe Data Numerik

Tipe data numerik dibedakan menjadi dua kelompok, yaitu *Integer* dan *Floating Point*. *Integer* digunakan untuk data bilangan bulat, sedangkan *floating* digunakan untuk bilangan desimal. Untuk tipe data numerik, selengkapnya dapat dilihat pada tabel berikut ini :

Tabel II.5. Tipe Data Numerik

Tipe Data	Rentang Nilai
Tinyint	-128 s/d 127
Smallint	-32768 s/d 332767
Mediumint	-3888608 s/d 8388607
Integer	-2147683648 s/d 21447683647
Bigint	-922337203685775808 s/d 9223372036854775807
Float	-3,4E+38 s/d -1,17E-38
Double	-1,79E+308 s/d -2,225E-308

Sumber : (Eko Priyo Utomo ;2015:61).

2. Tipe data String

Merupakan rangkaian karakter. Tipe data yang biasa digunakan adalah sebagai berikut :

Tabel II.6. Tipe Data String

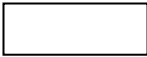
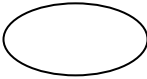
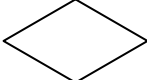

Tipe Data	Rentang Nilai
Char	1-255 karakter
Varchar	1-255 karakter

Sumber : (Eko Priyo Utomo ;2015:61).

II.10. Entity Relationship Diagram (ERD)

Entity Relationship Diagram atau ERD adalah data model yang didasarkan pada persepsi terhadap dunia nyata yang tersusun atas kumpulan objek-objek dasar yang disebut entitias dan hubungan antar objek(simarmata & Paryudi, 2006:59). Berikut Penjelasan tentang komponen-komponen ERD ditunjukkan pada Tabel II.8 :

Tabel II.7. Simbol ERD

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

(Sumber : D. Tri Octafian ; 2011 : 151)

II.10.1. Pemetaan Kardinalitas

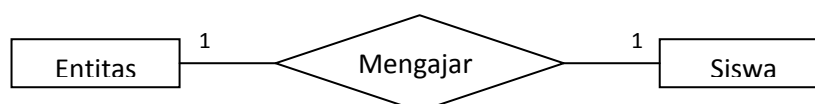
Pemetaan kardinalitas menyatakan jumlah entitas di mana entitas lain dapat dihubungkan ke entitas tersebut melalui sebuah himpunan relasi.

1. *One to One*

Sebuah entitas pada A berhubungan dengan paling banyak satu entitas pada B dan sebuah entitas pada B berhubungan dengan paling banyak satu entitas pada A.

Contoh :

Pada pengajaran privat, satu guru satu siswa. Seorang guru mengajar seorang siswa, seorang siswa diajar oleh seorang guru.



Gambar II.2 Contoh Pemetaan kardinalitas One to One
(Sumber : D. Tri Octafian ; 2011 : 151)

2. *One to Many/ Many to One*

Sebuah entitas pada A berhubungan dengan lebih dari satu entitas pada B dan sebuah entitas pada B berhubungan dengan paling banyak satu entitas pada A, atau sebaliknya (*Many to One*).

Contoh :

Dalam satu perusahaan, satu bagian mempekerjakan banyak pegawai. Satu bagian mempekerjakan banyak pegawai, satu pegawai kerja dalam satu bagian.



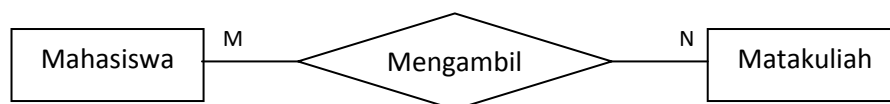
Gambar II.3 Contoh Pemetaan kardinalitas One to Many/ Many to One
(Sumber : D. Tri Octafian ; 2011 : 151-152)

3. *Many to Many*

Sebuah entitas pada A berhubungan dengan lebih dari satu entitas pada B dan sebuah entitas pada B berhubungan dengan lebih dari satu entitas pada A.

Contoh :

Dalam universitas, seorang Mahasiswa dapat mengambil banyak mata kuliah. Satu mahasiswa mengambil banyak mata kuliah dan satu mata kuliah diambil banyak mahasiswa.



Gambar II.4 Contoh Pemetaan kardinalitas Many to Many
(Sumber : D. Tri Octafian ; 2011 : 152)

II.11. *Unified Modeling Language (UML)*

UML (*Unified Modelling Language*) adalah salah satu alat bantu yang sangat handal dalam mengembangkan sistem berorientasi objek. Hal ini dikarenakan UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembangan sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*Sharing*) dan mengkomunikasikan rancangan dengan baik.



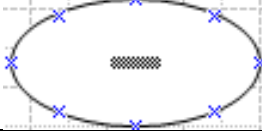

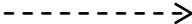
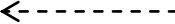
UML Merupakan satu kesatuan bahasa pemodelan yang dikembangkan oleh Booch, *Object Modeling Technique (OMT)* dan *Object Oriented Engineering (OOSE)*. Metode Booch dari Grady Booch sangat terkenal dengan nama Metode *Design Object Oriented* dimana metode ini menjadikan proses analisis dan design ke dalam empat tahapan iteratif, yaitu : Identifikasi kelas-kelas dan objek-objek, Identifikasi semantik dari hubungan Objek dan Kelas tersebut, perincian Interface dan Implementasi.(Haviluddin ; 2011 : 1)

II.11.1. Use Case Diagram

Use case adalah rangkian atau uraian sekelompok yang saling terkait dan membentuk sistem secara teratur yang dilakukan atau diawasi oleh sebuah aktor. *Use case* digunakan untuk membentuk tingkah laku benda dalam sebuah model serta direalisasikan oleh sebuah kolaborasi. Diagram *Use case* menggambarkan fungsionalitas yang diharapkan sebuah sistem. Hal yang ditekankan pada diagram ini adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case*

menyatakan sebuah aktivitas atas pekerjaan tertentu, misalnya *login* ke sistem, meng-*create* sebuah daftar belanja, dll. Untuk lebih jelas tentang *Use case* dapat dilihat dibawah ini :

Tabel II.8. Elemen-Elemen Digaram *Use Case*

Symbol	Nama	Keterangan
	<i>Sistem</i>	Sistem menyatakan batasan sistem dalam relasi dengan <i>actor-actor</i> yang menggunakannya (diluar sistem) dan fitur-fitur yang disediakan.
	<i>Actor</i>	<i>Actor</i> dapat berupa manusia, sistem, atau <i>device</i> yang memiliki peran dalam keberhasilan operasi dari sistem.
	<i>Use Case</i>	<i>Use case</i> mengidentifikasi fitur kunci dari sistem. Tanpa fitur ini, sistem tidak akan memenuhi permintaan <i>user/actor</i>
	<i>Association</i>	Digunakan untuk mengidentifikasi interaksi antara setiap <i>actor</i> tertentu dengan setiap <i>use case</i> tertentu
-	<i>Stereotape</i>	Berperan sebagai kualifier pada suatu elemen model.
	<i>Include Line</i>	<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 Line</i>	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

(Sumber : Hamim Tohari; 2014 : 52-54)

II.11.2. Class Diagram

Class Diagram adalah hubungan antar kelas yang menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class diagram* menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan.

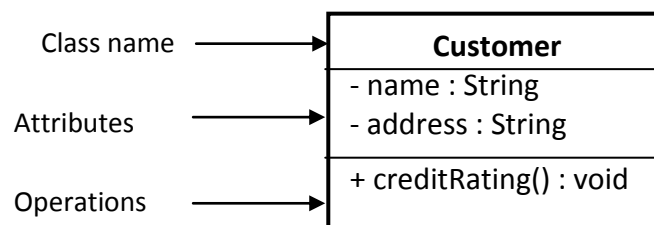
Class diagram menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *Containment*, *Pewarisan*, *asosiasi*, dll. *Class* secara khas meliputi:

1. Kelas (*Class*),
2. Atribut (*Attributes*),
3. Operasi (*Operations/Method*),

Atribut dan metode dapat memiliki salah satu sifat berikut :

1. *Private*, tidak dapat dipanggil dari luar *class* yang bersangkutan.
2. *Protected*, hanya dapat dipanggil oleh *class* yang bersangkutan.
3. *Public*, dapat dipanggil oleh siapa saja.

Class dapat merupakan implementasi dari sebuah *interface*, yaitu *class* abstrak yang hanya memiliki metode. *Interface* tidak dapat langsung diinstansiasikan, tetapi harus diimplementasikan dahulu menjadi sebuah *class*. Contoh diagram *class* dapat dilihat pada gambar II.5. dibawah ini:



Gambar II.5. Class Diagram
(Sumber : Hamim Tohari ; 2010 : 84)

Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti , Tabel II.9. Berikut Keterangannya :

Tabel II.9. Multiplicity Class Diagram



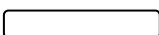
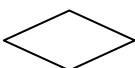
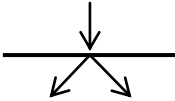
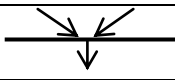

Indikator	Arti
0..1	Nol atau satu
1	Hanya satu
0..*	Nol atau lebih
1..*	Satu atau lebih
N	Hanya n (dengan $n > 1$)
0..n	Nol sampai n (dengan $n > 1$)
1..n	Satu sampai n (dengan $n > 1$)

(Sumber : Hamim Tohari; 2014 : 86)

II.11.3. Activity Diagram

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. Adapun elemen-elemen *activity diagram* dapat dilihat pada table II.10. :

Tabel II.10. Simbol Activity Diagram


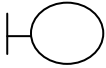


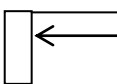


Notasi	Keterangan
	<i>Start point</i> , digunakan sebagai elemen mulai
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> .
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	<i>Join</i> (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	<i>Swimlane</i> , pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa.

(Sumber : Hamim Tohari; 2014 : 114-115)

II.11.4. Sequence Diagram

Sequence diagram menggambarkan interaksi antara sejumlah objek dalam urutan waktu. Kegunaannya untuk menunjukkan rangkaian pesan yang dikiri antara objek juga interaksi antar objek yang terjadi pada titik tertentu dalam eksekusi sistem. Dalam UML, objek, pada diagram sequence digambarkan dengan segi empat, yang berisi nama dari objek yang digaris bawah.

Tabel II.11. Simbol *Sequence Diagram*

Symbol	Nama	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.
		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</i>	<i>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 : Hamim Tohari; 2014 : 101)