

BAB II

KAJIAN PUSTAKA

2.1 Tinjauan Pustaka

Berdasarkan penelitian yang dilakukan oleh Harry Oktavianus Sofian (2010), software komputer memiliki peran yang heterogen khususnya dalam pengembangan bidang arsitektur. Hal ini memiliki persamaan karakteristik dengan penelitian yang dilakukan penulis khususnya pada objek penelitian (institusi pekerjaan). Pada prinsipnya software komputer dimanfaatkan di beberapa aspek kehidupan dalam memudahkan dan mengefektifkan peran kerja yang dilakukan oleh manusia.

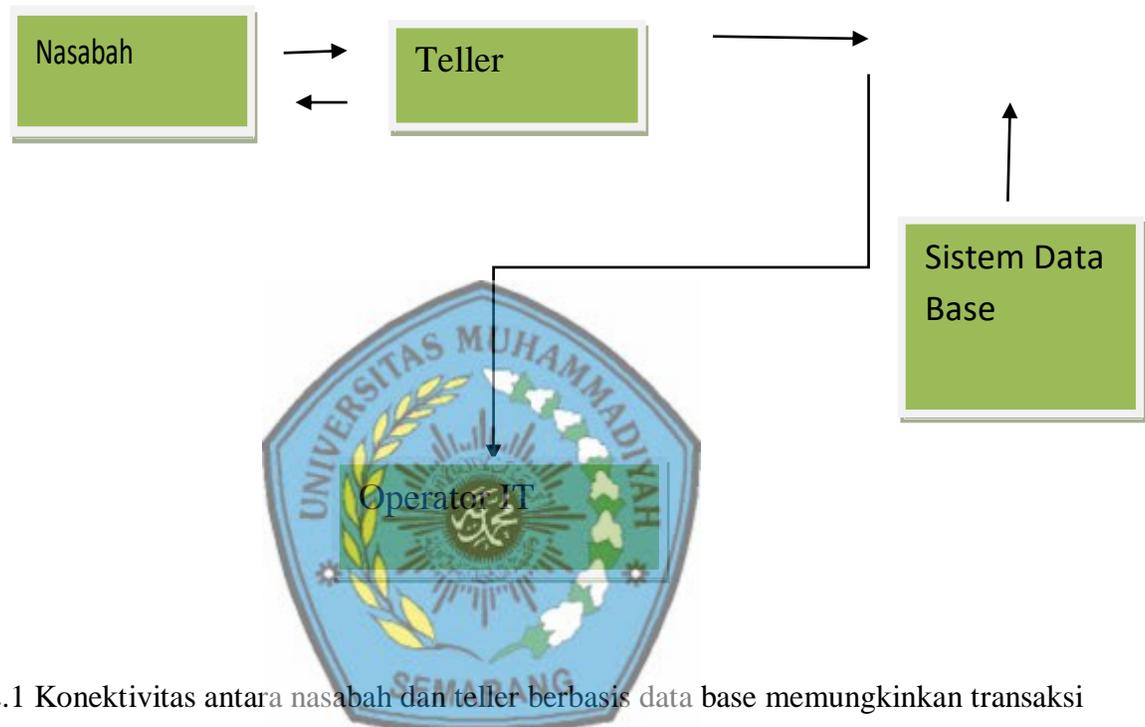
Menurut hasil penelitian dari Jones Garth N (2014) menunjukkan bahwa software komputer merupakan alat pendukung untuk meningkatkan pengetahuan terhadap beberapa sekolah di Afrika Selatan. Ini menunjukkan *tool / software* juga bisa diimplementasikan dalam aspek peningkatan kualitas pendidikan.

Mengutip dari beberapa kajian di atas, dalam penelitian ingin memberikan asumsi jika software komputer memiliki pengaruh positif apabila dimanfaatkan dengan baik sesuai dengan spesifikasi dan kegunaan dari software tersebut dalam membantu kinerja manusia. Oleh karena itu dalam penelitian akan meneliti lebih mendalam berkaitan dengan software yang sudah dikembangkan dalam melakukan evaluasi atau penilaian terhadap karyawan perbankan tempat penulis bekerja. Untuk mengkaji lebih mendalam dan menunjukkan terhadap pembaca, apabila *software authentic assessment* yang dilakukan dalam penilaian karyawan dapat memberikan pengaruh dalam peningkatan kapasitas dan kualitas kinerja sehingga dapat berpengaruh dalam peningkatan mutu atau kualitas layanan jasa.

2.1.1 Data Base System

Pada berbagai kepentingan, sistem database memegang peranan penting dalam menangani data. Sebagai contoh, sistem database dipakai pada pasar swalayan atau minimarket modern untuk menangani data persediaan barang dan juga mencatat data penjualan. Dengan adanya sistem ini manajemen pasar swalayan dapat mengetahui apakah persediaan barang masih cukup atau tidak. Bahkan informasi penjualan harian, mingguan, bulanan atau tahunan mudah sekali

ditemukan. Pada maskapai penerbangan sistem data base membuat calon penumpang dengan mudah bisa mendapatkan kursi. Hal itu terjadi karena ketersediaan sistem data base pada biro – biro perjalanan yang terhubung ke sistem data base maskapai penerbangan. Dalam dunia perbankan, semua transaksi keuangan terkoneksi dengan sistem online yang memudahkan pelaku ekonomi dalam melaksanakan transaksi keuangan di perbankan. Sistem ini cukup efektif dan efisien serta menghemat biaya, tenaga dan tentunya waktu (Kadir, 2010).



Gambar 2.1 Konektivitas antara nasabah dan teller berbasis data base memungkinkan transaksi keuangan berjalan efektif dan efisien

Sistem yang bersifat online tersebut memungkinkan proses transaksi keuangan berlangsung secara aman dan terstruktur. Untuk kepentingan pribadi sistem data base juga memberikan peran yang berarti. Sistem pada Gmail, misalnya memungkinkan pemakai menyimpan data kenalan. Hal itu bisa terlaksana karena keberadaan sistem data base. Begitu juga dengan blog atau website. Sistem data base lah yang mencatat isi website (blog) termasuk kapan suatu catatan dibuat dan data pemiliknya.

2.1.2 Komponen Data Base

Banyak istilah dasar yang berkaitan dengan data base. Di bawah ini beberapa istilah dijelaskan beberapa komponen dalam data base. Data Base menyatakan suatu wadah untuk

mengelola data. Sebuah data base melibatkan beberapa tabel. Contoh data base perbankan dalam format tabel.

Tabel 2.1 Contoh Tanggal Lahir Nasabah

Id.	Nama	Tgl_lahir
1.	Abdul Ghofur	19-01-1995
2.	Anggraeni Deqa	23-09-1988
3.	Joko Sulistyio	12-06-1978

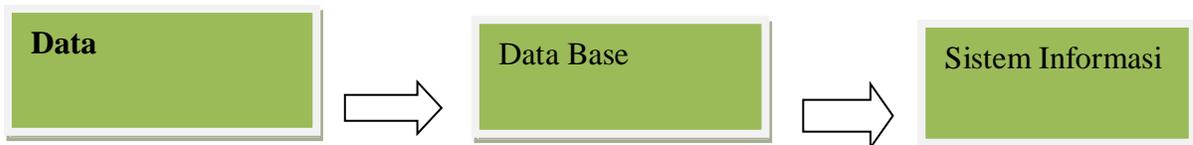
Tabel 2.2 Keterkaitan No.Id nasabah dan No. Rekening nasabah.

Id.	Nomor Rekening
1.	123-6666-0098
2.	344-990-008
3.	0999-099-890

Data Base Relasional adalah jenis data base yang mencatat hubungan antar tabel dalam bentuk data dalam tabel. Data Base Management System (DBMS) adalah perangkat lunak yang digunakan untuk mengelola dan mengontrol pengaksesan data base. Contoh DBMS terkenal adalah Microsoft Access dan Oracle.

Program Aplikasi adalah sebuah program komputer yang ditujukan untuk pemakai agar bisa berinteraksi dengan suatu data base. Program dapat dibuat menggunakan peranti pengembangan seperti Delphi, Visual Basic ataupun Visual C++. Sistem data Base adalah sistem yang digunakan untuk berinteraksi dengan data base, baik melalui DBMS ataupun program aplikasi. Data Base adalah bahan mentah yang disimpan dalam data base.

Informasi adalah hasil pemrosesan dalam data base sehingga menghasilkan suatu bentuk yang berguna bagi pemakai. Contohnya ditunjukkan pada gambar 2.2 berikut .



Gambar 2.2 Hasil Pemrosesan Data Base

2.1.3 Kelebihan Menggunakan DBMS

Sebagaimana telah dijelaskan di depan, DBMS adalah perangkat lunak yang digunakan untuk mengelola dan mengontrol pengaksesan data base. Namun definisi itu belum menggambarkan sepenuhnya kemampuan yang diberikan oleh DBMS. Oleh karena itu pembahasan mengenai kelebihan pada penggunaan DBMS perlu dikaji secara lebih mendalam. Empat kelebihan yang akan dibahas mencakup pengontrolan duplikasi data, penyediaan kemampuan berbagi data, peningkatan integritas data dan pengaturan keamanan data.(Ibnu Surya,2017).

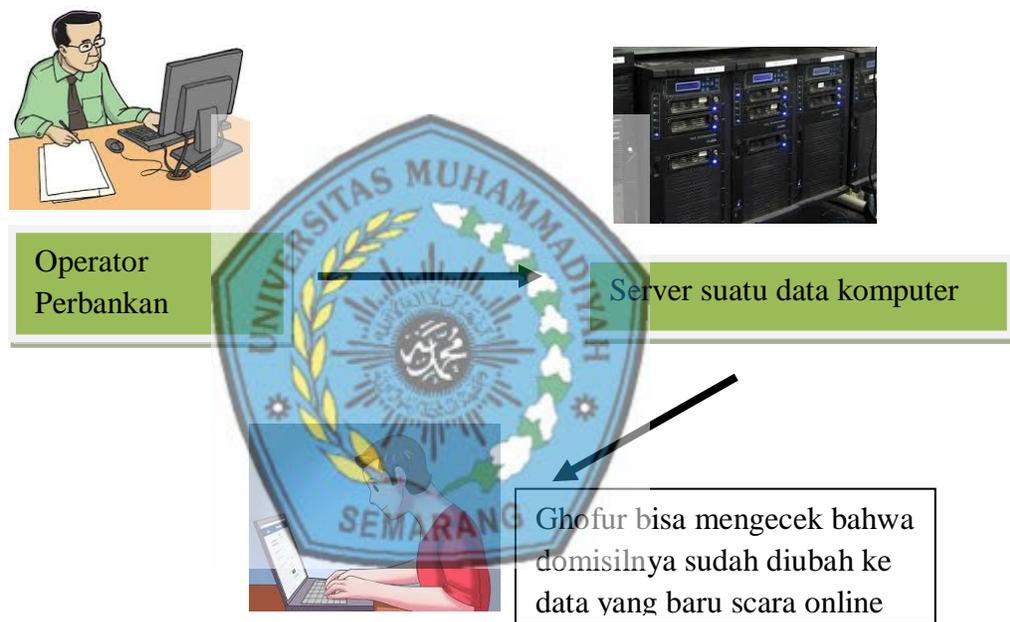
2.1.4 Mengontrol Duplikasi Data

DBMS memungkinkan pengontrolan terhadap duplikasi data. Sebagai contoh, dalam suatu sekolah setiap siswa memiliki nomor induk siswa. Bagi sekolah, satu nomor induk siswa hanya digunakan untuk satu siswa. Demikian halnya nomor rekening suatu nasabah. Pada Bank yang sama tentunya nomor rekening setiap nasabah akan berbeda – beda. Dengan menggunakan DBMS, data nomor induk siswa atau nomor rekening nasabah bisa diatur sedemikian hingga supaya tidak ada yang sama. Artinya satu nomor induk dipastikan tidak bisa digunakan untuk dua orang siswa demikian halnya satu nomor rekening juga dipastikan tidak bisa dimiliki oleh dua orang nasabah. Apa yang terjadi, apabila nomor induk siswa sudah digunakan untuk siswa bernama A dan kemudian digunakan lagi untuk siswa B nantinya DBMS dengan sendirinya akan menolaknya. Misalkan dalam sistem pencatatan nomor 237 untuk Esti Daniati ditolak secara otomatis karena nomor 237 sudah digunakan untuk yang lain. Semisal diketahui no.rekening 8997-9980-7897 pada suatu Bank dimiliki oleh nasabah Rudi Hertanto dan kemudian ada orang bernama Budi Santoso akan menggunakan nomor rekening tersebut tentunya akan ditolak oleh sistem. Inilah salah satu keunggulan dari DBMS kaitannya dalam duplikasi data.

2.1.5 Menyediakan Kemampuan Berbagi Data

Sebelum sistem database muncul, data dalam suatu organisasi disimpan dalam bentuk sistem berkas. Pada sistem seperti ini berkas umumnya menjadi milik seseorang atau departemen / unit tertentu. Akibatnya data yang tersimpan dalam berkas tersebut tidak dapat diakses oleh yang lain menggunakan komputer yang berbeda. Kemunculan DBMS membuat

data bisa digunakan oleh banyak orang bahkan dalam waktu yang bersamaan. Kemampuan yang membuat data bisa digunakan oleh banyak orang inilah yang disebut berbagi data (*data sharing*). Kemampuan ini sekaligus mengurangi duplikasi data. Bisa dibayangkan kalau setiap departemen / unit dalam suatu organisasi memiliki data tersendiri. Data seorang pegawai bisa tersimpan di beberapa berkas, yang berarti terjadi duplikasi. Dengan memanfaatkan kemampuan berbagi data, data seorang pegawai cukup disimpan di suatu tempat (dalam data base), tetapi data itu bisa digunakan oleh banyak orang. Bila ada perubahan terhadap data, misalnya pegawai pindah bagian, perubahan bisa dilihat oleh semua orang yang memang bisa mengakses data tersebut.



Gambar 2.3 Ilustrasi Gambar Berbagi data memungkinkan data bersifat transparan bagi pemakai

Gambar di atas menunjukkan kemampuan berbagi data yang memungkinkan data bisa diakses oleh banyak orang. Data bersifat transparan bagi semua orang. Begitu dapat diubah, perubahannya dapat dilihat oleh yang lain.

2.1.6 Meningkatkan Integritas Data

Pada aspek pengontrolan duplikasi data dapat dijelaskan bahwa DBMS memungkinkan pencegahan duplikasi data pada nomor siswa atau nomor rekening nasabah. Kemampuan ini sebenarnya hanya salah satu contoh untuk menjaga integritas data.

Pengertian integritas data sendiri berarti bahwa data diusahakan dalam keadaan valid. DBMS pada prinsipnya mempunyai berbagai fitur untuk menjaga integritas data.

2.1.7 Mengatur Keamanan Data

DBMS memberikan fasilitas yang berguna untuk mengatur akses terhadap data. Fasilitas ini memungkinkan data tertentu hanya bisa diakses oleh orang tertentu. Sebagai konsekuensinya, setiap pemakai perlu mengakses sistem dengan menggunakan identitas pemakai (biasa disebut nama pemakai atau username) yang dilengkapi dengan password. Melalui mekanisme ini, semisal ada karyawan X yang bekerja di bagian penggajian berhak untuk mengetahui gaji pegawai. Namun, karyawan Y yang bekerja di pelatihan hanya berhak mendapatkan informasi pegawai yang tentunya yang tidak ada kaitan dengan gaji. Ini merupakan aspek keunggulan DBMS berkaitan dengan keamanan dalam transparansi data yang sifatnya rahasia (Kadir, 2010)

2.2 Aplikasi Software

2.2.1. My SQL dalam Pembuatan Software *Authentic Assessment*

MySQL tergolong sebagai DBMS (*DataBase Management System*). Perangkat lunak ini bermanfaat untuk mengelola data dengan cara yang sangat fleksibel dan cepat. Berikut adalah sejumlah aktivitas yang terkait dengan data yang didukung oleh perangkat lunak tersebut.

1. Menyimpan data ke dalam tabel,
2. Menghapus data dalam tabel
3. Mengubah data dalam tabel,
4. Mengambil data yang tersimpan dalam tabel,
5. Memungkinkan untuk memilih data tertentu yang diambil,
6. Memungkinkan untuk melakukan pengaturan hak akses terhadap data.

My SQL banyak dipakai untuk kepentingan penanganan data base karena selain handal juga bersifat open source. Konsekuensi dari open source, perangkat lunak ini dapat dipakai oleh siapa saja tanpa membayar source code-nya bisa diunduh oleh siapa saja.

Software yang digunakan dalam penelitian ini adalah Wampserver. Perangkat lunak ini dapat diunduh melalui website.

2.2.2 Konsep Data Base dalam My SQL

Data Base menyatakan suatu wadah untuk mengelola data. Data Base mencakup sejumlah tabel dan berbagai objek yang terkait dengan pengolahan data. Data Base merupakan suatu wadah besar yang berisi sekumpulan data yang terstruktur dan sistematis.

Basis data (*Database*) adalah kumpulan informasi yang disimpan dalam computer secara sistematis sehingga dapat diperiksa menggunakan suatu program computer untuk memperoleh informasi dari basis data tersebut. Database digunakan untuk menyimpan informasi atau data yang terintegrasi dengan baik.

Untuk dapat mengolah database diperlukan suatu perangkat lunak yang disebut DBMS (Database Management System). DBMS merupakan suatu system perangkat lunak yang memungkinkan user pengguna computer untuk membuat, memelihara, mengontrol dan mengakses database secara praktis dan efisien. Dengan DBMS user dapat dengan mudah mengontrol dan memanipulasi data yang ada.

2.2.2 Authentic Assesment

Software Penilaian Autentik adalah perangkat yang diciptakan untuk menjelaskan berbagai metode penilaian alternatif yang memungkinkan seseorang dapat mendemonstrasikan kemampuannya dalam menyelesaikan tugas-tugas dan menyelesaikan masalah. Sekaligus, mengekspresikan pengetahuan dan keterampilannya dengan cara mensimulasikan situasi yang dapat ditemui di dalam dunia nyata di luar lingkungan kerja atau institusi. Dalam hal ini adalah simulasi yang dapat mengekspresikan prestasi (*performance*) seseorang yang ditemui di dalam praktik dunia nyata (Ma'Ruf dan Andy Lenny Rahim,2012).

Penilaian autentik dapat dibuat oleh penilai sendiri, penilai secara tim, atau penilai bekerja sama dengan object asesment. Dalam penilaian autentik, keterlibatan object assessment sangat penting. Asumsinya seseorang dapat melakukan aktivitas bekerja secara lebih baik jika mereka tahu bagaimana akan dinilai. Object assessment diminta untuk merefleksikan

dan mengevaluasi kinerja mereka sendiri dalam rangka meningkatkan pemahaman yang lebih dalam tentang tujuan pekerjaan serta mendorong kemampuan bekerja yang lebih tinggi. Pada penilaian autentik, penilai menerapkan kriteria yang berkaitan dengan konstruksi pengetahuan, kajian keilmuan, dan pengalaman yang diperoleh dari luar tempat bekerja.

Penilaian autentik mencoba menggabungkan kegiatan evaluator, kegiatan bekerja, motivasi dan keterlibatan object assessment, serta keterampilan belajar. Karena penilaian itu merupakan bagian dari proses pembelajaran, evaluator dan object assessment berbagi pemahaman tentang kriteria kinerja. Dalam beberapa kasus, object assessment bahkan berkontribusi untuk mendefinisikan harapan atas tugas-tugas yang harus mereka lakukan.

Penilaian autentik sering digambarkan sebagai penilaian atas perkembangan seseorang karena berfokus pada kemampuan mereka berkembang untuk belajar bagaimana belajar tentang subjek. Penilaian autentik harus menggambarkan sikap, keterampilan, dan pengetahuan apa yang sudah atau belum dimiliki oleh seseorang, bagaimana mereka menerapkan pengetahuannya, dalam hal apa mereka sudah atau belum mampu menerapkan perolehan belajar, dan sebagainya. Atas dasar itu, evaluator dapat mengidentifikasi kinerja apa yang sudah layak dilanjutkan.

2.3 Use Case

Use Case merupakan sebuah teknik yang digunakan dalam pengembangan sebuah software atau sistem informasi untuk menangkap kebutuhan fungsional dari sistem yang bersangkutan, Use Case menjelaskan interaksi yang terjadi antara 'aktor'—inisiator dari interaksi sistem itu sendiri dengan sistem yang ada, sebuah Use Case direpresentasikan dengan urutan langkah yang sederhana. Perilaku sistem adalah bagaimana sistem beraksi dan bereaksi. Perilaku ini merupakan aktifitas sistem yang bisa dilihat dari luar dan bisa diuji. Perilaku sistem ini dicapture di dalam USE CASE.

USE CASE sendiri mendeskripsikan sistem, lingkungan sistem, serta hubungan antara sistem dengan lingkungannya. Deskripsi dari sekumpulan aksi sekuensial yang ditampilkan sistem yang menghasilkan yang tampak dari nilai ke actor khusus. Use Case digunakan untuk menyusun behavioral things dalam sebuah model. Use case direalisasikan dengan sebuah collaboration.

Secara gambar, sebuah use case digambarkan dengan sebuah ellips dengan garis penuh, biasanya termasuk hanya namanya, seperti gambar berikut :

a. Manfaat Use Case

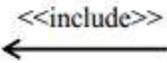
1. Digunakan untuk berkomunikasi dengan end user dan domain expert.
2. Memastikan pemahaman yang tepat tentang requirement / kebutuhan sistem.
3. Digunakan untuk mengidentifikasi siapa yang berinteraksi dengan sistem dan apa yang harus dilakukan sistem.
4. Interface yang harus dimiliki sistem.
5. Digunakan untuk verifikasi.

b. Karakteristik

1. Use cases adalah interaksi atau dialog antara sistem dan actor, termasuk pertukaran pesan dan tindakan yang dilakukan oleh sistem.
2. Use cases diprakarsai oleh actor dan mungkin melibatkan peran actor lain. Use cases harus menyediakan nilai minimal kepada satu actor.
3. Use cases bisa memiliki perluasan yang mendefinisikan tindakan khusus dalam interaksi atau use case lain mungkin disisipkan.
4. Use case class memiliki objek use case yang disebut skenario. Skenario menyatakan urutan pesan dan tindakan tunggal,



2. Komponen Use Case

Simbol	Keterangan
	Aktor : Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan <i>use case</i>
	<i>Use case</i> : Abstraksi dan interaksi antara sistem dan aktor
	<i>Association</i> : Abstraksi dari penghubung antara aktor dengan <i>use case</i>
	<i>Generalisasi</i> : Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i>
	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya
	Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi

- 1) Use case dibuat berdasar keperluan actor, merupakan “apa” yang dikerjakan system, bukan “bagaimana” system mengerjakannya.
- 2) Use case diberi nama yang menyatakan apa hal yang dicapai dari hasil interaksinya dengan actor.
- 3) Use case dinotasikan dengan gambar (horizontal ellipse).
- 4) Use case biasanya menggunakan kata kerja.
- 5) Nama use case boleh terdiri dari beberapa kata dan tidak boleh ada 2 use case yang memiliki nama yang sama.

c. Relasi Dalam Use Case

Ada beberapa relasi yang terdapat pada *use case diagram*:

1. **Association**, menghubungkan link antar element.
2. **Generalization**, sebuah elemen yang menjadi spesialisasi dari elemen yang lain.
3. **Dependency**, sebuah elemen yang bergantung beberapa cara kepada elemen lainnya.
4. **Aggregation**, bentuk association yang di mana sebuah elemen berisi elemen lainnya.

Tipe relasi/ *stereotype* yang mungkin terjadi pada *use case diagram*:

1. <<include>> , yaitu kelakuan yang harus terpenuhi agar sebuah *event* dapat terjadi, dimana pada kondisi ini sebuah *use case* adalah bagian dari *use case* lainnya.
2. <<extends>>, kelakuan yang hanya berjalan di bawah kondisi tertentu seperti menggerakkan alarm.
3. <<communicates>>, mungkin ditambahkan untuk asosiasi yang menunjukkan asosiasinya adalah *communicates association* . Ini merupakan pilihan selama asosiasi hanya tipe *relationship* yang dibolehkan antara *actor* dan *use case*.

Relasi Dalam Use Case

Ada beberapa relasi yang terdapat pada use case diagram: 1. Association, menghubungkan link antar element. 2. Generalization, disebut juga inheritance (pewarisan), sebuah elemen dapat merupakan spesialisasi dari elemen lainnya. 3. Dependency, sebuah element bergantung dalam beberapa cara ke element lainnya. 4. Aggregation, bentuk association dimana sebuah elemen berisi elemen lainnya.

Penjelasan

Pada use case diatas, actornya adalah Kepala Teller Bank Mandiri dan Nasabah Bank Mandiri. Tugas - tugas dari setiap actor berbeda-beda, dan dicantumkan pada use case yang ada. Tetapi, sebelum mereka bisa melakukan tugas tersebut ada include yang mengharuskan mereka untuk login kedalam system.

2.4 Interface dan Data Integrity

Dalam penggunaan umum, sebuah antarmuka, penghubung atau interface adalah sebuah titik, wilayah, atau permukaan di mana dua zat atau benda berbeda bertemu; dia juga digunakan secara metafora untuk perbatasan antara benda. Kata interface kadangkala (biasanya dalam bidang teknik) disingkat menjadi "i/f".

Bentuk kerja dari interface berarti menghubungkan dua atau lebih benda pada suatu titik atau batasan yang terbagi, atau untuk menyiapkan kedua benda untuk tujuan tersebut.

Kata interface juga memiliki arti khusus:

1. Antarmuka pengguna adalah fungsi dan atribut sensor dari suatu sistem (aplikasi, perangkat lunak, kendaraan, dll) yang berhubungan dengan pengoperasiannya oleh pengguna.
2. Dalam elektronik dan teknik komputer, sebuah antarmuka dapat berarti:
 - a) Batasan fisik dari dua subsistem atau alat.
 - b) Sebuah bagian atau sirkuit di beberapa subsistem yang mengirim atau menerima sinyal ke atau dari subsistem lainnya: antarmuka jaringan, antarmuka video, kartu network.
 - c) Sebuah standar yang menjelaskan sebuah himpunan karakteristik yang berfungsi, karakteristik interkoneksi fisik umum, dan karakteristik signal untuk pertukaran data atau signal; antarmuka USB, antarmuka SCSI.
3. Dalam telekomunikasi, sebuah titik interkoneksi antara pengguna peralatan terminal dan fasilitas komunikasi komersial.
4. Dalam rekayasa perangkat lunak, adalah sebuah spesifikasi dari properti sebuah komponen software yang komponen lainnya dapat bergantung kepadanya: lihat antarmuka (ilmu komputer).

Integritas Data

Pengertian integritas data secara luas mengacu pada kepercayaan dari sumber daya suatu sistem. Integritas data sangat penting karena dapat memastikan keakuratan, konsistensi, aksesibilitas, dan kualitas tinggi dari sebuah data, sehingga sangat penting untuk mengikuti aturan pengintegritasan suatu data. Data yang mempunyai integritas identik di pertahankan selama operasi apapun (seperti bisnis transfer, penyimpanan, atau pengambilan).

Secara sederhana dalam istilah bisnis, integritas data adalah jaminan bahwa data konsisten, bersertifikat dan dapat dirujukan.

Contoh dari mekanisme integritas data adalah hubungan orang tua dan anak dengan record terkait. Jika dalam catatan orangtua memiliki satu atau lebih dari catatan anak terkait semua proses integritas referensial akan ditangani oleh database itu sendiri, yang secara otomatis menjamin keakuratan dan integritas data sehingga tidak ada catatan anak bisa ada tanpa orang tua (atau disebut juga yatim piatu) dan bahwa tidak ada orangtua kehilangan catatan anak mereka. Ini juga menjamin bahwa tidak ada catatan induk dapat dihapus sementara orangtua memiliki record setiap anak. Semua ini ditangani pada tingkat database dan tidak memerlukan coding cek integritas ke dalam setiap aplikasi.

Integritas Data dapat di kelompokkan menjadi 4 bagian yaitu Integritas data yang berada dalam relasi, yaitu integritas entitas dan integritas domain. Integritas yang berada di luar relasi, yaitu integritas referensial dan selain itu ada juga integritas yang ditentukan sendiri di dalam suatu perusahaan, yaitu integritas perusahaan (*Enterprise integrity/ user Defined Integrity*).

Secara garis besar integritas data dalam model relasional meliputi :

1. Integritas Entitas
2. Integritas Domain
3. Integritas Referensial
4. Integritas Enterprise



2.4.1 Integritas Entitas

Integritas entitas mendefinisikan sebuah baris sebagai sebuah entitas yang unik untuk suatu tabel. Dalam integritas entitas, tidak ada baris yang di duplikat didalam suatu tabel.

Contoh :

```
create table Pembelian  
(ID Pembelian smallint,  
ID model smallint,  
DeskripsiModel varchar (40),  
Primary Key (IDPembelian));
```

2.4.2 Integritas Domain

Domain adalah nilai-nilai yang dimungkinkan diasosiasikan dengan setiap atribut, Integritas domain merupakan validasi masukan dari sebuah kolom. Dengan integritas domain, tidak ada data yang melanggar jangkauan nilai di tiap kolom data.

Jenis domain yang harus dimiliki oleh setiap atribut :

1. Karakter bebas
2. Alphanumerik
3. Alphabet
4. Numerik

Pemeliharaan integritas domain :

1. Pendefinisian skema
2. Pemanfaatan properti field
3. Penerapan proses validasi pada pemasukan data



Contoh :

Create table *biografi*

(*idpenulis* smallint unsigned not null,

tahunLahir year not null,

kotalahir varchar (40) not null default

'Kosong');

create domain *nilai numerik*(3,2)

constraint *value-test check* (value >=0.00)

2.4.3 Integritas Referensial

Integritas referensial adalah dasar relasi antar tabel yaitu antara foreign key dengan primary key. Integritas referensial memastikan bahwa seluruh nilai dari foreign key cocok dengan nilai primary key yang dihubungkanya.

Contoh :

create table *account*

(*account-number* char (10),

branch-name char (15),

balance integer,

primary key (*account-number*),

foreign key (*branch-name*) **reference** *branch*)

create table *depositor*

(*costumer-name* char (20),

account-number char (10),

primary key (*costumer-name*, *account-number*)

foreign key (*account-number*) **reference** *account*

foreign key (*costumer-name*) **reference** *costumer*)

on delete cascade on update cascade



2.5 Konsep Halaman Web

Konsep halaman Web merupakan diagram yang menggambarkan alur kinerja sistem authentic assessment dalam merepresentasikan penilaian kinerja seorang Teller Bank Mandiri. Bagan ini merupakan refleksi sistem kinerja software authentic assessment dalam memberikan evaluasi kepada Teller Bank Mandiri sehingga peningkatan kinerja bisa terlaksana dengan baik. Kaitan dengan bagan ini merupakan kombinasi dari sistem Interface dan beberapa aspek yang lain sehingga membentuk suatu elemen komponen penelitian. *Website* merupakan halaman situs sistem informasi yang dapat diakses secara cepat. *Website* ini didasari dari adanya perkembangan teknologi informasi dan komunikasi. Melalui perkembangan teknologi informasi, tercipta suatu jaringan antar komputer yang saling berkaitan. Jaringan yang dikenal dengan istilah internet secara terus-menerus menjadi pesan-pesan elektronik, termasuk *e-mail*, transmisi *file*, dan komunikasi dua arah antar individu atau komputer.

Sedangkan aplikasi web merupakan aplikasi yang diakses menggunakan web browser melalui jaringan internet atau intranet. Aplikasi web juga merupakan suatu perangkat lunak komputer yang dikodekan dalam bahasa pemrograman yang mendukung perangkat lunak berbasis web seperti HTML, JavaScript, CSS, Ruby, Python, Php, Java dan bahasa pemrograman lainnya.

Cara Kerja Web

Adapun cara kerja web adalah sebagai berikut:

1. Informasi web disimpan dalam dokumen dalam bentuk halaman-halaman web atau web page.
2. Halaman web tersebut disimpan dalam computer server web.
3. Sementara dipihak pemakai ada computer yang bertindak sebagai computer client dimana ditempatkan program untuk membaca halaman web yang ada di server web (browser).
4. Browser membaca halaman web yang ada di server web.

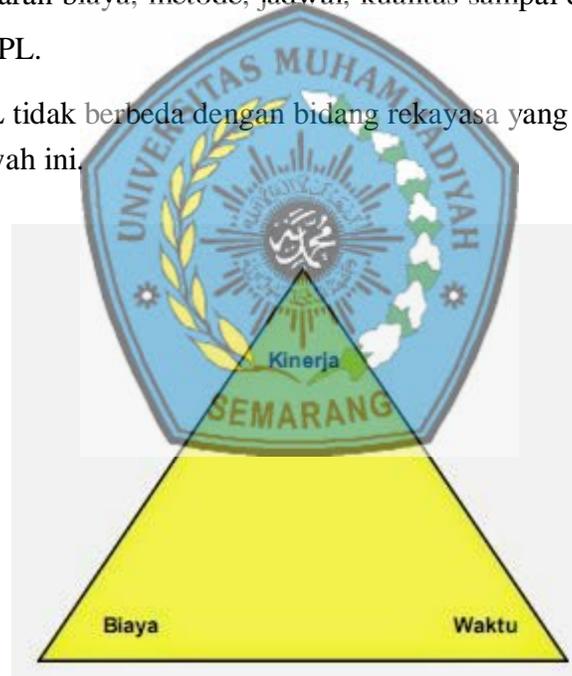
Ada 2 bagian pokok dalam aplikasi web, yang pertama adalah sisi client dan yang kedua adalah sisi server, sisi client dalam hal ini adalah PC atau bisa juga Perangkat mobile yang terhubung ke jaringan internet, Client dapat mengakses aplikasi web melalui web browser seperti internet explorer, mozilla fire fox, google chrome, opera dan lain-lain, sedangkan server adalah perangkat komputer dengan spesifikasi yang bagus digunakan untuk menyimpan aplikasi web beserta database server yang siap untuk diakses oleh client, Client bertugas meminta halaman web server melalui Web Browser, Web browser akan meneruskannya ke Server dimana Aplikasi

Web berada, Komputer Server akan mengolah permintaan dari client, ketika halaman web yang diminta.

2.6 Rekayasa Perangkat Lunak

RPL sendiri adalah suatu disiplin ilmu yang membahas semua aspek produksi perangkat lunak, mulai dari tahap awal yaitu analisa kebutuhan pengguna, menentukan spesifikasi dari kebutuhan pengguna, disain, pengkodean, pengujian sampai pemeliharaan sistem setelah digunakan. Dari pengertian ini jelaslah bahwa RPL tidak hanya berhubungan dengan cara pembuatan program komputer. Pernyataan "semua aspek produksi" pada pengertian di atas, mempunyai arti semua hal yang berhubungan dengan proses produksi seperti manajemen proyek, penentuan personil, anggaran biaya, metode, jadwal, kualitas sampai dengan pelatihan pengguna merupakan bagian dari RPL.

Secara umum tujuan RPL tidak berbeda dengan bidang rekayasa yang lain. Hal ini dapat kita lihat pada Gambar di bawah ini.



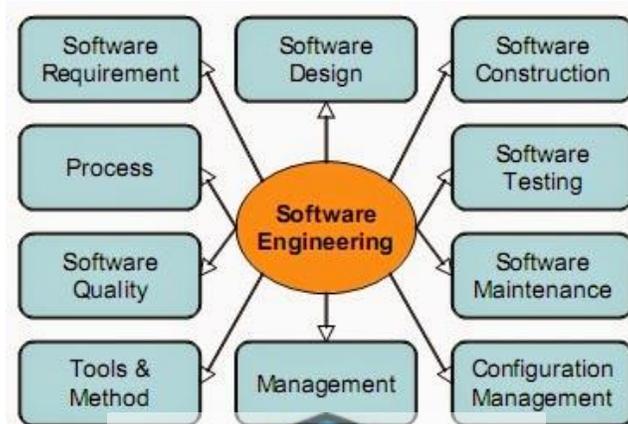
Gambar 2.4 Bidang Rekayasa

Dari Gambar di atas dapat diartikan bahwa bidang rekayasa akan selalu berusaha menghasilkan output yang kinerjanya tinggi, biaya rendah dan waktu penyelesaian yang tepat. Secara lebih khusus kita dapat menyatakan tujuan RPL adalah:

1. Memperoleh biaya produksi perangkat lunak yang rendah
2. Menghasilkan perangkat lunak yang kinerjanya tinggi, andal dan tepat waktu

3. Menghasilkan perangkat lunak yang dapat bekerja pada berbagai jenis platform
4. Menghasilkan perangkat lunak yang biaya perawatannya rendah

Sesuai dengan definisi yang telah disampaikan sebelumnya, maka ruang lingkup RPL dapat digambarkan sebagai berikut:



Gambar 2.5 Ruang Lingkup RPL

- a. Software Requirements berhubungan dengan spesifikasi kebutuhan dan persyaratan perangkat lunak
- b. Software Desain mencakup proses penampilan arsitektur, komponen, antar muka, dan karakteristik lain dari perangkat lunak
- c. Software Construction berhubungan dengan detail pengembangan perangkat lunak, termasuk algoritma, pengkodean, pengujian dan pencarian kesalahan
- d. Software Testing meliputi pengujian pada keseluruhan perilaku perangkat lunak
- e. Software Maintenance mencakup upaya-upaya perawatan ketika perangkat lunak telah dioperasikan
- f. Software Configuration Management berhubungan dengan usaha perubahan konfigurasi perangkat lunak untuk memenuhi kebutuhan tertentu
- g. Software Engineering management berkaitan dengan pengelolaan dan pengukuran RPL, termasuk perencanaan proyek perangkat lunak
- h. Software Engineering Tools And Methods mencakup kajian teoritis tentang alat bantu dan metode RPL
- i. Software Engineering Process berhubungan dengan definisi, implementasi pengukuran, pengelolaan, perubahan dan perbaikan proses RPL

2.7 Pengertian *Entity Relationship Diagram*

Entity Relationship Diagram (ERD) merupakan salah satu bentuk pemodelan basis data yang sering digunakan dalam pengembangan sistem informasi. Diagram hubungan entitas (ERD) menunjukkan hubungan dari entitas set disimpan dalam database. Entitas dalam konteks ini adalah komponen data. Dengan kata lain, diagram ER menggambarkan struktur logis dari database. Dalam rekayasa perangkat lunak, sebuah Entity-Relationship Model (ERM) merupakan abstrak dan konseptual representasi data. Entity-Relationship adalah salah satu metode pemodelan basis data yang digunakan untuk menghasilkan skema konseptual untuk jenis/model data semantik sistem. Dimana sistem seringkali memiliki basis data relasional, dan ketentuannya bersifat top-down. Diagram untuk menggambarkan model Entity-Relationship ini disebut Entity-Relationship diagram, ER diagram, atau ERD. Menurut salah satu para ahli, Brady dan Loonam (2010), Entity Relationship diagram (ERD) merupakan teknik yang digunakan untuk memodelkan kebutuhan data dari suatu organisasi, biasanya oleh System Analysts dalam tahap analisis persyaratan proyek pengembangan sistem. Sementara seolah-olah teknik diagram atau alat peraga memberikan dasar untuk desain database relasional yang mendasari sistem informasi yang dikembangkan. ERD bersama-sama dengan detail pendukung merupakan model data yang pada gilirannya digunakan sebagai spesifikasi untuk database.

Model data

a. *Model data konseptual*

Ini adalah model tingkat ER tertinggi di dalamnya berisi detail paling granular kan tetapi menetapkan lingkup keseluruhan apa yang harus disertakan dalam model set. Model ER konseptual biasanya mendefinisikan entitas data referensi utama yang biasa digunakan oleh organisasi. Mengembangkan model ER konseptual berguna untuk mendukung dan mendokumentasikan arsitektur data bagi suatu organisasi. Model ER konseptual dapat digunakan sebagai dasar untuk satu atau lebih logis model data. Tujuan dari model ER konseptual ialah untuk membangun struktur meta data kesamaan untuk data master entitas antara set model ER logis. Model data konseptual dapat digunakan untuk membentuk hubungan kesamaan antara model ER sebagai dasar untuk integrasi model data.

b. *Model data logis*

Sebuah model ER logis tidak memerlukan model ER konseptual, terutama jika lingkup model ER logis hanya mencakup pengembangan sistem informasi yang berbeda. Model ER logis mengandung lebih rinci dari model ER konseptual. Selain entitas data master, operasional dan transaksional entitas data sekarang didefinisikan. Rincian masing-masing entitas data yang dikembangkan dan hubungan antara entitas data ini didirikan. Model ER logis namun dikembangkan secara independen dari yang spesifik sistem manajemen database ke yang dapat diimplementasikan.

c. *Model data fisik*

Satu atau lebih model ER fisik mungkin dikembangkan dari masing-masing model ER logis. Model ER fisik biasanya dikembangkan untuk dipakai sebagai database. Oleh karena itu, masing-masing model ER fisik harus berisi cukup detail untuk menghasilkan database dan masing-masing model ER fisik tergantung teknologi karena setiap sistem. Model fisik biasanya dipakai dalam metadata struktural dari sistem manajemen database sebagai objek database relasional seperti tabel database, indeks database seperti kunci unik indeks, dan kendala database seperti batasan kunci asing atau kendala kesamaan. Model ER juga biasanya digunakan untuk merancang modifikasi objek database relasional dan untuk mempertahankan metadata struktural database.

Pada dasarnya ada tiga komponen yang digunakan, yaitu :

A. Entitas

Entiti merupakan objek yang mewakili sesuatu yang nyata dan dapat dibedakan dari sesuatu yang lain. Simbol dari entiti ini biasanya digambarkan dengan persegi panjang.

B. Atribut

Setiap entitas pasti mempunyai elemen yang disebut atribut yang berfungsi untuk mendeskripsikan karakteristik dari entitas tersebut. Isi dari atribut mempunyai sesuatu yang dapat mengidentifikasi isi elemen satu dengan yang lain.

Beberapa jenis Atribut antara lain : *Atribut key* dan *Attribute Composite*

Atribut Key adalah satu atau gabungan dari beberapa atribut yang dapat membedakan semua baris data (Row/Record) dalam tabel secara unik. Dikatakan unik jika pada atribut yang dijadikan key tidak boleh ada baris data dengan nilai yang sama

Contoh : Nomor pokok mahasiswa (NPM), NIM dan nomor pokok lainnya

Atribut composite adalah suatu atribut yang terdiri dari beberapa atribut yang lebih kecil yang mempunyai arti tertentu yang masih bisah dipecah lagi atau mempunyai sub attribute.

Contoh : dari entitas nama yaitu nama depan, nama tengah, dan nama belakang.

Atribut Derivatif. Atribut yang tidak harus disimpan dalam database

Contoh : Atribut yang dihasilkan dari atribut lain atau dari suatu relationship. Atribut ini dilambangkan dengan bentuk oval yang bergaris putus-putus

C. Hubungan / Relasi

Hubungan antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda.

Derajat relasi atau kardinalitas rasio menjelaskan jumlah maksimum hubungan antara satu entitas dengan entitas lainnya.

2.8 Pengertian *Data Flow Diagram*

Data Flow Diagram atau sering disingkat DFD adalah perangkat-perangkat analisis dan perancangan yang terstruktur sehingga memungkinkan peng-analis sistem memahami sistem dan subsistem secara visual sebagai suatu rangkaian aliran data yang saling berkaitan.



Gambar 2.6 Simbol-simbol dalam DFD

Entitas biasanya diberi nama dengan kata benda. Aliran data merupakan perpindahan data dari satu titik ke titik yang lain (penggambarannya dengan cara kepala tanda panah mengarah ke tujuan datanya). Proses biasanya selalu menunjukkan suatu perubahan data dan terjadinya proses

transformasi data. Penyimpanan Data (data store) diberi nama dengan kata benda, sesuai dengan data yang disimpan didalamnya.

Didalam DFD terdapat 3 level, yaitu :

1. Diagram Konteks : menggambarkan satu lingkaran besar yang dapat mewakili seluruh proses yang terdapat di dalam suatu sistem. Merupakan tingkatan tertinggi dalam DFD dan biasanya diberi nomor 0 (nol). Semua entitas eksternal yang ditunjukkan pada diagram konteks berikut aliran-aliran data utama menuju dan dari sistem. Diagram ini sama sekali tidak memuat penyimpanan data dan tampak sederhana untuk diciptakan.

2. Diagram Nol (diagram level-1) : merupakan satu lingkaran besar yang mewakili lingkaran-lingkaran kecil yang ada di dalamnya. Merupakan pemecahan dari diagram Konteks ke diagram Nol. di dalam diagram ini memuat penyimpanan data.

3. Diagram Rinci : merupakan diagram yang menguraikan proses apa yang ada dalam diagram Nol.

Fungsi dari Data Flow Diagram adalah :

- a) Data Flow Diagram (DFD) adalah alat pembuatan model yang memungkinkan profesional sistem untuk menggambarkan sistem sebagai suatu jaringan proses fungsional yang dihubungkan satu sama lain dengan alur data, baik secara manual maupun komputerisasi.
- b) DFD ini adalah salah satu alat pembuatan model yang sering digunakan, khususnya bila fungsi-fungsi sistem merupakan bagian yang lebih penting dan kompleks dari pada data yang dimanipulasi oleh sistem. Dengan kata lain, DFD adalah alat pembuatan model yang memberikan penekanan hanya pada fungsi sistem.
- c) DFD ini merupakan alat perancangan sistem yang berorientasi pada alur data dengan konsep dekomposisi dapat digunakan untuk penggambaran analisa maupun rancangan sistem yang mudah dikomunikasikan oleh profesional sistem kepada pemakai maupun pembuat program.