

BAB II

TINJAUAN PUSTAKA

II.1. Sistem

Sistem merupakan kumpulan dari unsur atau elemen-elemen yang saling berkaitan/berinteraksi dan saling mempengaruhi dalam melakukan kegiatan bersama untuk mencapai suatu tujuan tertentu. (Hendra, 2012 : 157)

Sistem adalah sekelompok elemen yang terintegrasi dengan maksud yang sama untuk mencapai tujuan. Menurut Krismiaji (2010 : 1) sistem adalah serangkaian komponen yang dikoordinasikan untuk mencapai serangkaian tujuan. (Palupi, 2012 : 3)

Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan untuk menyelesaikan suatu sasaran tertentu. (Fauziah, Widilestariningtyas, Yulianto ; 2010 : 5)

II.2. Informasi

Informasi merupakan data yang telah diproses menjadi bentuk yang memiliki arti bagi penerima dan dapat berupa fakta, suatu nilai yang bermanfaat. (Hendra, 2012 : 40)

Menurut George H. Bodnar (2006 : 6), informasi merupakan data yang berguna dan diolah sehingga dapat dijadikan dasar untuk pengambilan keputusan yang tepat. (Palupi, 2012 : 4)

Informasi adalah data yang telah diorganisasi dan telah memiliki kegunaan dan manfaat. (Fauziah, Widilestariningtyas, Yulianto ; 2010 : 5)

II.3. Sistem Informasi

Menurut Robert A. Leitch, sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial, dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan. (Hendra, 2012 : 169)

Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial, dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan. (Fauziah, Widilestariningtyas, Yulianto ; 2010 : 5)

II.4. Sistem Informasi Akuntansi

Akuntansi adalah sistem informasi yang menghasilkan informasi keuangan kepada pihak-pihak yang berkepentingan mengenai aktivitas ekonomi dan kondisi suatu perusahaan. (Rudianto, 2012 : 4)

Sistem informasi akuntansi adalah kumpulan sumber daya, seperti manusia dan peralatan, yang diatur untuk mengubah data menjadi informasi. Informasi ini dikomunikasikan kepada beragam pengambil keputusan. SIA mewujudkan perubahan ini secara manual atau terkomputerisasi.

SIA juga merupakan sistem yang paling penting di organisasi dan merubah cara menangkap, memproses, menyimpan dan mendistribusikan informasi.

SIA pada umumnya meliputi beberapa siklus pemrosesan transaksi :

1. Siklus pendapatan. Berkaitan dengan pendistribusian barang dan jasa ke entitas lain dan pengumpulan pembayaran-pembayaran yang berkaitan.
2. Siklus pengeluaran. Berkaitan dengan perolehan barang jasa dari entitas lain dan pelunasan kewajiban yang berkaitan.
3. Siklus produksi. Berkaitan dengan pengubahan sumberdaya menjadi barang dan jasa.
4. Siklus keuangan. Kejadian-kejadian yang berkaitan dengan perolehan dan manajemen dana-dana modal, termasuk kas. (Mujilan, 2012 : 3)

II.5. Persediaan

Persediaan adalah sejumlah barang jadi, bahan baku, dan baran dalam proses yang dimiliki perusahaan dengan tujuan untuk dijual atau diproses lebih lanjut. (Rudianto, 2012 : 222)

Menurut Ikatan Akuntansi Indonesia, persediaan adalah aktiva yang tersedia untuk dijual dalam kegiatan usaha normal dalam proses produksi dan atau dalam perjalanan atau dalam bentuk bahan atau perlengkapan (*supplies*) untuk digunakan dalam proses produksi atau pemberian jasa. (Palupi, 2012 : 4)

Menurut Kieso, dkk (2007), persediaan adalah pos-pos aktiva yang dimiliki untuk dijual dalam operasi bisnis normal atau barang yang akan

digunakan atau dikonsumsi dalam memproduksi barang yang akan dijual.
(Ratnasari, 2011 : 5)

II.6. Metode FIFO (*First In First Out*)

First In Frist Out Method (FIFO) adalah metode penilaian yang menganggap barang yang pertama kali masuk diasumsikan keluar pertama kali pula. (Palupi, 2012 : 4)

Dalam metode FIFO, barang yang masuk (dibeli atau diproduksi) terlebih dahulu akan dikeluarkan (dijual) pertama kali, sehingga yang tersisa pada akhir periode adalah barang yang berasal dari pembelian atau produksi terakhir. (Rudianto, 2012 : 223)

Menurut cara ini, harga barang yang masuk (dibeli) lebih awal dianggap dikeluarkan (dijual) lebih awal pula. Dengan demikian sisa persediaan barang pada akhir periode adalah barang-barang dengan harga yang masuknya (dibelinya) paling akhir.

Misalnya perhitungan pisik atas barang-barang dalam gudang tanggal 28 Februari 200X menunjukkan jumlah 300 kg. jumlah persediaan akhir ini terdiri dari :

Persediaan 24 Feb 100 kg @ Rp. 126,- = Rp. 12.600,-

Pembelian 15 Feb 200 kg @ Rp. 116,- = Rp. 23.200,-

300 kg = Rp. 35.800,-

II.7. Alat Bantu Pengembangan Sistem

Alat bantu pengembangan sistem yang digunakan dalam penulisan skripsi ini adalah sebagai berikut :

II.7.1. Sistem Basis Data

Database adalah sekumpulan tabel-tabel yang saling berelasi, relasi bisa tersebut bisa ditunjukkan dengan kunci dari tiap tabel yang ada. Satu database menunjukkan satu kumpulan data yang dipakai dalam satu lingkup perusahaan atau instansi.

Database mempunyai kegunaan dalam mengatasi penyusunan dan penyimpanan data, maka seringkali masalah yang dihadapi adalah :

1. Redudansi dan inkonsistensi data
2. Kesulitan dalam pengaksesan data
3. Multi user
4. Keamanan data
5. Integritas data
6. Kebebasan data (Asrianda, Fadlisyah ; 2008 : 1-2)

Pemanfaatan basis data dilakukan untuk memenuhi tujuan berikut ini :

1. Kecepatan dan kemudahan (*speed*)
2. Efisiensi ruang penyimpanan (*space*)
3. Keakuratan (*accuracy*)
4. Ketersediaan (*availability*)
5. Kelengkapan (*completeness*)
6. Keamanan (*security*)

7. Pemakaian bersama (*sharability*)

Basis data adalah satu komponen utama dalam sistem informasi dan tidak ada sistem informasi yang bisa dijalankan tanpa adanya basis data. Jadi pada intinya untuk meningkatkan kinerja perusahaan menggunakan aplikasi berbasis komputer maka basis data mutlak diperlukan. Beberapa pemanfaatan basis data antara lain :

1. Kepegawaian, untuk berbagai perusahaan yang memiliki banyak pegawai.
2. Pergudangan (*inventory*) dan penjualan, untuk perusahaan manufaktur (pabrik), supermarket, apotek.
3. Akuntansi, untuk bank dan perusahaan-perusahaan yang melibatkan uang.
4. Reservasi, untuk hotel, pesawat, kereta api.
5. Catatan medik, untuk rumah sakit.
6. Akademik, untuk perguruan tinggi atau sekolah.
7. Penggajian, untuk perusahaan yang memiliki pegawai yang cukup banyak.

Tujuan perancangan basis data adalah mendapatkan skema basis data yang meminimalisasi terjadinya redundansi dan duplikasi data serta menjaga integritas data. Kebanyakan metode perancangan berbasis pada model basis data relasional. Pada basis data relasional, data diatur melalui pembuatan tabel-tabel dan terdapat keterkaitan antara tabel yang satu dengan lainnya (relasi). (Hidayatullah, 2012 : 137-139)

Gabungan antara basis data dan perangkat lunak DBMS (*Database Management System*) termasuk di dalamnya program aplikasi yang dibuat dan bekerja dalam satu sistem disebut dengan sistem basis data. Sistem basis data

dapat dianggap sebagai tempat untuk sekumpulan berkas data yang terkomputerisasi dengan tujuan untuk memelihara informasi dan membuat informasi tersebut tersedia saat dibutuhkan. DBMS dapat diartikan sebagai program komputer yang digunakan untuk memasukkan, mengubah, menghapus, memodifikasi dan memperoleh data/informasi dengan praktis dan efisien.

Arsitektur DBMS dikenal dengan nama arsitektur tiga skema, dimana fungsi ini untuk memisahkan antara basis data fisik dengan program aplikasi user.

Skema-skema tersebut adalah sebagai berikut :

1. Level internal merupakan skema internal yang memuat deskripsi struktur penyimpanan basis data dan menggunakan model data fisikal serta mendefinisikan secara detail penyimpanan data dalam basis data, serta jalur pengaksesan data.
2. Level konseptual adalah skema yang memuat deskripsi struktur basis data secara keseluruhan untuk semua pemakai. Skema ini hanya memuat deskripsi tentang entitas, atribut, hubungan dan batasan, tanpa memuat deskripsi data secara detail.
3. Level eksternal merupakan skema eksternal (*user view*) yang mendefinisikan pandangan data terhadap sekelompok user (*local view*) dengan menyembunyikan data lain yang tidak diperlukan oleh kelompok user tersebut.

Model data dapat dikelompokkan berdasarkan konsep bagaimana deskripsi struktur basis data, yaitu :

1. Model data konseptual (*high level*) menyajikan konsep tentang bagaimana user memandang atau memperlakukan data. Dalam model ini dikenalkan tiga konsep penyajian data yaitu :
 - a. *Entity* (entitas) merupakan penyajian objek, kejadian atau konsep dunia nyata yang keberadaannya secara eksplisit didefinisikan dan disimpan dalam basis data, contohnya Mahasiswa, Matakuliah, Dosen, Nilai dan lain sebagainya.
 - b. *Attribute* (atribut) adalah keterangan-keterangan yang menjelaskan karakteristik dari suatu entitas seperti NIM, Nama, Fakultas, Jurusan untuk entitas Mahasiswa.
 - c. *Relationship* (hubungan) merupakan hubungan atau interaksi antara satu entitas dengan yang lainnya, misalnya entitas pelanggan berhubungan dengan entitas barang yang dibelinya.
2. Model data fiskal (*low level*) merupakan konsep bagaimana deskripsi detail data disimpan ke dalam komputer dengan menyajikan informasi rekaman, urutan rekaman, dan jalur pengaksesan data yang dapat membuat pencarian rekaman data lebih efisien.
3. Model data implementasi (*representational*) merupakan konsep deskripsi data disimpan dalam komputer dengan menyembunyikan sebagian detail deskripsi data sehingga para user mendapat gambaran global bagaimana data disimpan dalam komputer. Model ini merupakan konsep model data yang digunakan oleh model hirarki, jaringan dan relasional. (Dzacko, 2007 : 3)

II.7.2. *Entity Relationship Diagram*

Pengertian dari ERD (*Entity Relationship Diagram*) adalah suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi.

ERD untuk memodelkan struktur data dan hubungan antar data, untuk menggambarannya digunakan beberapa notasi dan simbol. Pada dasarnya ada tiga komponen yang digunakan, yaitu :

1. 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.

2. 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. Gambar atribut diwakili oleh simbol elips.

3. 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.

- a. *One to One* (1:1)

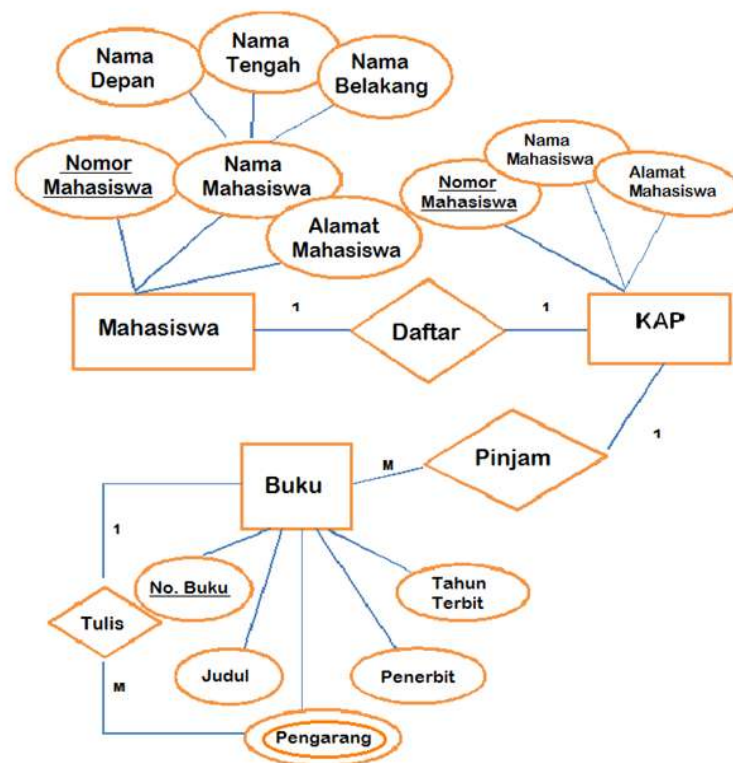
Setiap anggota entitas A hanya boleh berhubungan dengan satu anggota entitas B, begitu pula sebaliknya.

b. *One to many* (1:M / Many)

Setiap anggota entitas A dapat berhubungan dengan lebih dari satu anggota entitas B tetapi tidak sebaliknya.

c. *Many to Many* (M:M)

Setiap entitas A dapat berhubungan dengan banyak entitas himpunan entitas B dan demikian pula sebaliknya. (Kusrini, 2007 : 21 - 22)



Gambar II.1. Contoh Entity Relationship Diagram

(Sumber : Kusrini, 2007 : 23)

II.7.3. Kamus Data

Kamus data (*data dictionary*) adalah suatu penjelasan tertulis tentang suatu data yang berada di dalam *database* atau suatu daftar data elemen yang terorganisir dengan definisi yang tetap dan sesuai dengan sistem, sehingga *user* dan analis sistem mempunyai pengertian yang sama tentang *input*, *output*, dan komponen *data store*.

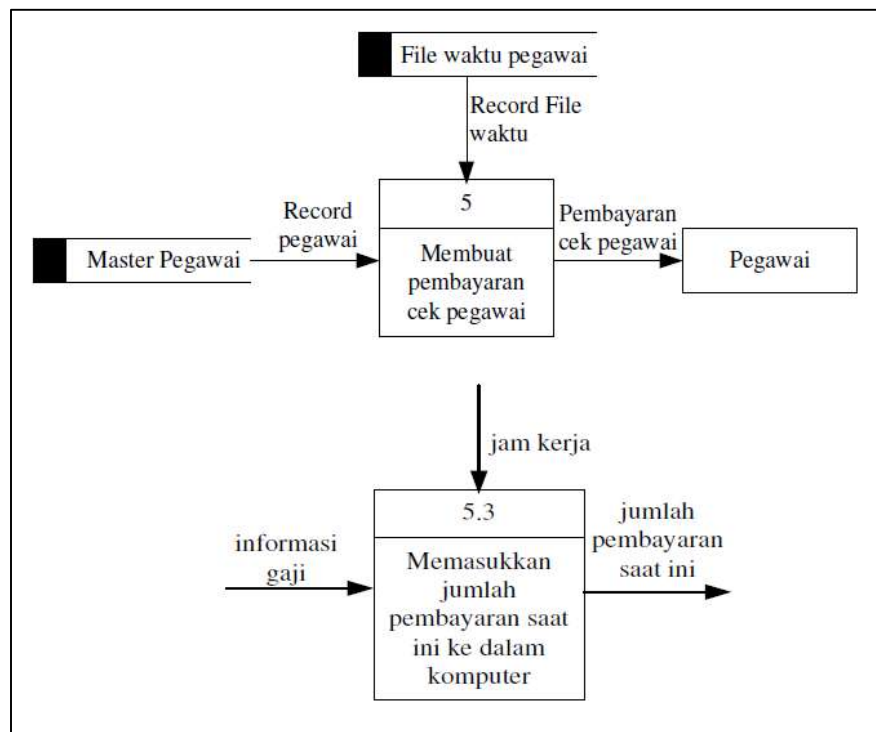
Kamus data pertama berbasis kamus dokumen tersimpan dalam suatu bentuk *hard copy* dengan mencatat semua penjelasan data dalam bentuk yang dicetak. Walau sejumlah kamus berbasis dokumen masih ada, praktik yang umum saat ini ialah mempergunakan kamus data yang berbasis komputer. Pada kamus data berbasis komputer, penjelasan data dimasukkan ke dalam komputer dengan memakai *Data Description Language* (DDL) dari *System Manajemen Database*, sistem kamus atau peralatan CASE. Kamus data tidak perlu dihubungkan dengan diagram arus data dan formulir-formulir kamus data dirancang untuk mendukung diagram arus data.

Kamus Data merupakan 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 ini sangat membantu analisis sistem dalam mendefinisikan data yang mengalir di dalam sistem, sehingga pendefinisian data itu dapat dilakukan dengan lengkap dan terstruktur. Pembentukan kamus data dilaksanakan dalam tahap analisis dan perancangan suatu sistem. (Kusrini, 2007 : 63)

Dua diagram alir data di bawah ini dan masukan-masukan kamus data yang berhubungan untuk memproduksi pembayaran cek pegawai.



Gambar II.2. Contoh Pengelolaan Kamus Data

(Sumber : Kusrini, 2007 : 67)

Contoh Struktur Data:

Record Pegawai = Nomor Pegawai +
 Informasi Pribadi +
 Informasi Gaji +
 Informasi Pembayaran Saat Ini +

Informasi Gaji Tahunan Sampai Hari Ini

Record File Waktu = Nomor Pegawai +

Nama Pegawai +

Jam Kerja

Pembayaran Cek Gaji = Nomor Pegawai +

Nama Pegawai +

Alamat +

Jumlah Pembayaran Saat Ini +

Jumlah Gaji Tahunan Sampai Saat Ini

Informasi Gaji = Perhitungan Pembayaran +

Jumlah Tanggungan

Jumlah Pembayaran Saat Ini = Gaji Kotor +

Potongan Pajak Pemerintah +

Potongan Pajak Negara Bagian +

Potongan Pajak Jaminan Sosial + Gaji Bersih

(Kusrini, 2007 : 67)

II.7.4. Normalisasi

Normalisasi merupakan sebuah teknik dalam *logical* desain sebuah basis data yang mengelompokkan atribut dari suatu relasi sehingga membentuk struktur relasi yang baik (tanpa redundansi).

Normalisasi adalah proses pembentukan struktur basis data sehingga sebagian besar *ambiguity* bisa dihilangkan.

Tujuan normalisasi data adalah untuk :

1. Untuk menghilangkan kerangkapan data.
2. Untuk mengurangi kompleksitas.
3. Untuk mempermudah pemodifikasian data.

Proses normalisasi adalah sebagai berikut :

1. Data diuraikan dalam bentuk tabel, selanjutnya dianalisis berdasarkan persyaratan tertentu ke beberapa tingkat.
2. Apabila tabel yang diuji belum memenuhi persyaratan tertentu, maka tabel tersebut perlu dipecah menjadi beberapa tabel yang lebih sederhana sampai memenuhi bentuk yang optimal. (Kusrini, 2007 : 39)

Tahapan dalam Normalisasi Data.

Tahap Normalisasi dimulai dari tahap paling ringan (1NF) hingga paling ketat (5NF). Biasanya hanya sampai pada tingkat 3NF atau BCNF karena sudah cukup memadai untuk menghasilkan tabel-tabel yang berkualitas baik. Urutan : 1NF, 2NF, 3NF, BCNF, 4NF, 5NF.

1. Bentuk 1NF (*First Normal Form*)

Aturan :

Tidak adanya atribut *multi-value*, atribut komposit atau kombinasinya.

Mendefinisikan atribut kunci.

Setiap atribut dalam tabel tersebut harus bernilai *atomic* (tidak dapat dibagi-bagi lagi).

2. Bentuk 2NF (*Second Normal Form*)

Aturan :

Sudah memenuhi dalam bentuk normal kesatu (1NF).

Semua atribut bukan kunci hanya boleh tergantung (*functional dependency*) pada atribut kunci.

Jika ada ketergantungan parsial maka atribut tersebut harus dipisah pada tabel yang lain.

Perlu ada tabel penghubung ataupun kehadiran *foreign key* bagi atribut-atribut yang telah dipisah tadi. (Kusrini, 2007 : 40)

3. Bentuk 3NF (*3th Normal Form*)

Aturan :

Sudah berada dalam bentuk normal kedua (2NF).

Tidak ada ketergantungan transitif (dimana atribut bukan kunci tergantung pada atribut bukan kunci lainnya).

4. Bentuk 4NF (*4th Normal Form*)

Bentuk normal 4NF terpenuhi dalam sebuah tabel jika telah memenuhi bentuk BCNF, dan tabel tersebut tidak boleh memiliki lebih dari sebuah *multivalued attribute*.

Untuk setiap *multivalued dependencies* (MVD) juga harus merupakan *functional dependencies*.

5. Bentuk 5NF (*5th Normal Form*)

Bentuk normal 5NF terpenuhi jika tidak dapat memiliki *sebuah lossless decomposition* menjadi tabel-tabel yg lebih kecil.

Jika 4 bentuk normal sebelumnya dibentuk berdasarkan *functional dependency*, 5NF dibentuk berdasarkan konsep *join dependence*. Yakni apabila sebuah tabel telah di-dekomposisi menjadi tabel-tabel lebih kecil,

harus bisa digabungkan lagi (*join*) untuk membentuk tabel semula (Kusrini, 2007 : 41)

Tabel II.1. Penerapan Normalisasi Data

NoProyek	NamaProyek	NoPegawai	NamaPegawai	Golongan	BesarGaji
NP001	BRR	Peg01	Anton	A	1.000.000
		Peg02	Paula	B	900.000
		Peg06	Koko	C	750.000
NP002	PEMDA	Peg01	Anton	A	1.000.000
		Peg12	Sita	B	900.000
		Peg14	Yusni	B	900.000

(Sumber: Kusrini, 2007 : 41)

Untuk mendapatkan hasil yang paling normal, maka proses normalisasi dimulai dari normal pertama. *Field-field* tabel di atas yang merupakan *group* berulang : NoPegawai, NamaPegawai, Golongan, BesarGaji.

Solusinya hilangkan duplikasi dengan mencari ketergantungan parsial. menjadikan *field-field* menjadi tergantung pada satu atau beberapa *field*. Karena yang dapat dijadikan kunci adalah NoProyek dan NoPegawai, maka langkah kemudian dicari *field-field* mana yang tergantung pada NoProyek dan mana yang tergantung pada NoPegawai. (Kusrini, 2007 : 41)

Field-field yang tergantung pada satu *field* haruslah dipisah dengan tepat, Misalnya NoProyek menjelaskan NamaProyek dan NoPegawai menjelaskan NamaPegawai, Golongan dan BesarGaji.

Tabel II.2. Pemisahan Field

NoProyek	NamaProyek
NP001	BRR
NP002	PEMDA

NoPegawai	NamaPegawai	Golongan	BesarGaji
Peg01	Anton	A	1.000.000
Peg02	Paula	B	900.000
Peg06	Koko	C	750.000
Peg12	Sita	B	800.000
Peg14	Yusni	B	900.000

(Sumber: Kusrini, 2007 : 42)

Pada tabel diatas masih terdapat masalah, bahwa BesarGaji tergantung kepada Golongan nya. Padahal disini Golongan bukan merupakan *field* kunci. Artinya kita harus memisahkan *field* non-kunci Golongan dan BesarGaji yang tadinya tergantung secara parsial kepada *field* kunci NoPegawai, untuk menghilangkan ketergantungan transitif. (Kusrini, 2007 : 42)

II.7.5. *MySQL*

MySQL dikembangkan oleh sebuah perusahaan Swedia bernama *MySQL AB* yang pada saat itu bernama *TcX DataKonsult AB*. Awalnya *TcX* membuat *MySQL* dengan tujuan mengembangkan aplikasi web untuk klien. *TcX* merupakan perusahaan pengembang *software* dan konsultan *database*.

MySQL adalah salah satu jenis *database server* yang sangat terkenal dan banyak digunakan untuk membangun aplikasi web yang menggunakan *database* sebagai sumber dan pengelolaan datanya. Kepopuleran *MySQL* antara lain karena

MySQL menggunakan *SQL* sebagai bahasa dasar untuk mengakses *database*-nya sehingga mudah untuk digunakan, kinerja *query* cepat, dan mencukupi untuk kebutuhan *database* perusahaan-perusahaan skala menengah-kecil.

MySQL merupakan *database* yang pertama kali didukung oleh bahasa pemrograman *script* untuk internet (*PHP* dan *Perl*). *MySQL* dan *PHP* dianggap sebagai pasangan *software* pengembangan aplikasi web yang ideal. *MySQL* lebih sering digunakan untuk membangun aplikasi berbasis web, umumnya pengembangan aplikasinya menggunakan bahasa pemrograman *script PHP*. (Arief, 2011 : 151)

II.7.6. *PHP (Hypertext Preprocessor)*

PHP adalah bahasa *server-side scripting* yang menyatu dengan *HTML* untuk membuat halaman web yang dinamis. Karena *PHP* merupakan *server-side scripting* maka sintaks dan perintah-perintah *PHP* akan di eksekusi di *server* kemudian hasilnya dikirimkan ke *browser* dalam format *HTML*. Dengan demikian kode program yang ditulis dalam *PHP* tidak akan terlihat oleh *user* sehingga keamanan halaman web lebih terjamin. *PHP* dirancang untuk membentuk halaman web yang dinamis, yaitu halaman web yang dapat membentuk suatu tampilan berdasarkan permintaan terkini, seperti menampilkan isi basis data ke halaman web. (Arief, 2011 : 43)

Script PHP termasuk dalam *HTML-embedded*, artinya kode *PHP* dapat disisipkan pada sebuah halaman *HTML*. Ada beberapa cara untuk menuliskan *script PHP*, yaitu :

1. `<?php`
 Script PHP
 `?>`
2. `<?`
 Script PHP
 `?>`
3. `<script language = "php">`
 Script PHP Anda
 `</script>`
4. `<%`
 Script PHP Anda
 `%>`

Cara pertama merupakan format yang dianjurkan tetapi mungkin cara kedua akan sering digunakan karena lebih ringkas. (Arief, 2011 : 45)

II.7.7. UML (*Unified Modelling Language*)

Unified Modelling Language (UML) adalah suatu alat untuk memvisualisasikan dan mendokumentasikan hasil analisa dan desain yang berisi sintak dalam memodelkan sistem secara visual. Juga merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem *software* yang terkait dengan objek. (Haviluddin, 2011 : 1)

Secara filosofi UML diilhami oleh konsep yang telah ada yaitu konsep permodelan *Object Oriented* karena konsep ini menganalogikan sistem seperti

kehidupan nyata yang didominasi oleh obyek dan digambarkan atau dinotasikan dalam simbol-simbol yang cukup spesifik. (Haviluddin, 2011 : 1)

II.7.7.1. Komponen – Komponen UML

Sejauh ini para pakar merasa lebih mudah dalam menganalisa dan mendesain atau memodelkan suatu sistem karena UML memiliki seperangkat aturan dan notasi dalam bentuk grafis yang cukup spesifik. (Haviluddin, 2011 : 3)

Komponen atau notasi UML diturunkan dari 3 (tiga) notasi yang telah ada sebelumnya yaitu Grady Booch, OOD (*Object-Oriented Design*), Jim Rumbaugh, OMT (*Object Modelling Technique*), dan Ivar Jacobson OOSE (*Object-Oriented Software Engineering*).

Pada UML versi 2 terdiri atas tiga kategori dan memiliki 13 jenis diagram yaitu :

1. Struktur Diagram.

Menggambarkan elemen dari spesifikasi dimulai dengan kelas, obyek, dan hubungan mereka, dan beralih ke dokumen arsitektur logis dari suatu sistem. Struktur diagram dalam UML terdiri atas :

a. *Class diagram*

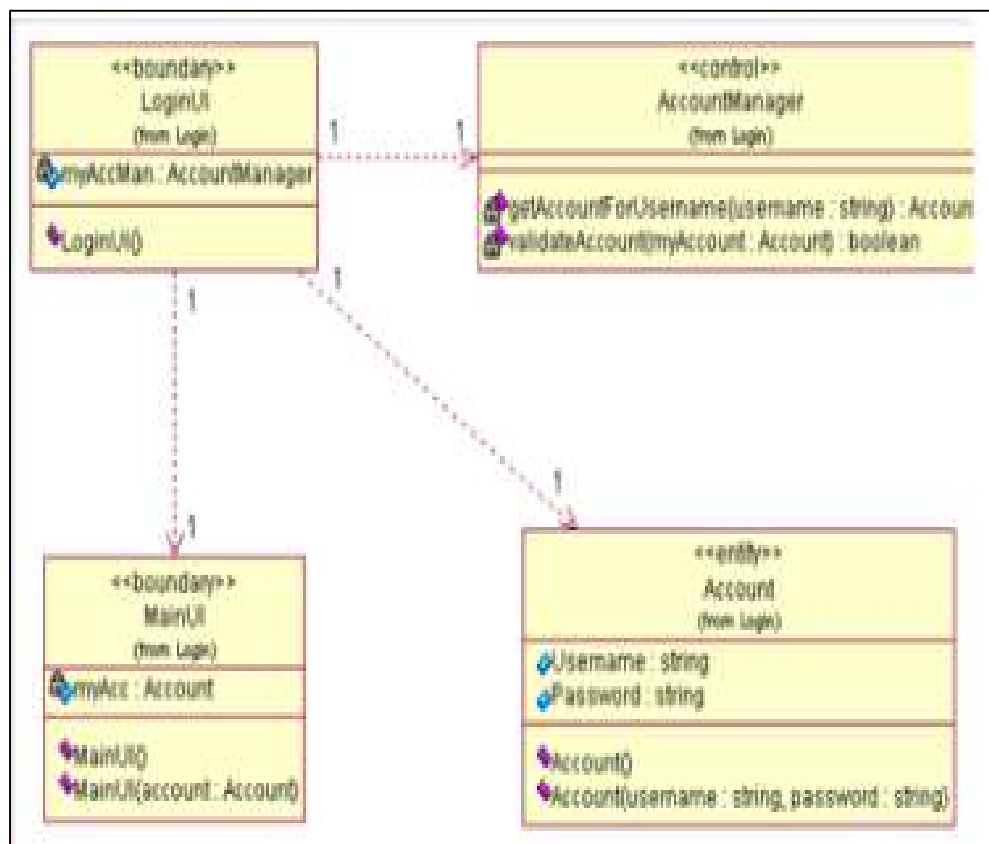
Class diagram menggambarkan struktur statis dari kelas dalam sistem anda dan menggambarkan atribut, operasi dan hubungan antara kelas.

Class diagram membantu dalam memvisualisasikan struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai. Selama tahap desain, *class diagram* berperan dalam

menangkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat.

Class memiliki tiga area pokok :

- 1) Nama (dan *stereotype*)
- 2) Atribut
- 3) Metoda

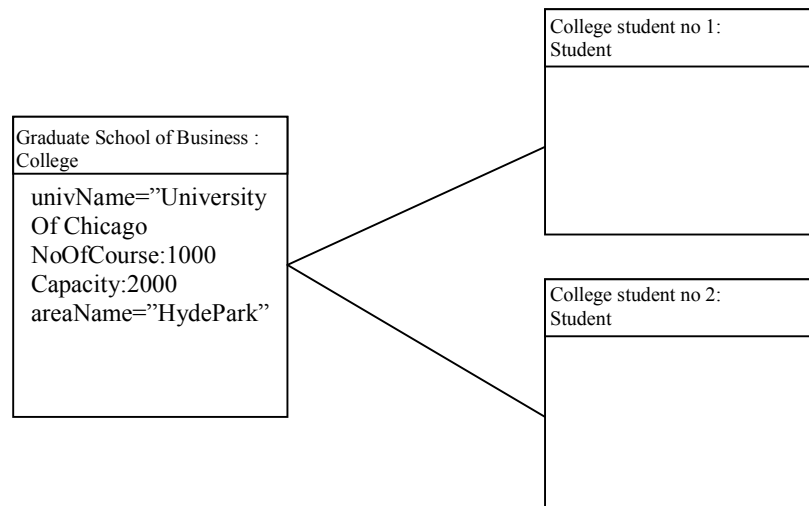


Gambar II.3. Notasi Class Diagram

(Sumber : Haviluddin, 2011 : 3)

b. *Object diagram*

Object diagram menggambarkan kejelasan kelas dan warisan dan kadang-kadang diambil ketika merencanakan kelas, atau untuk membantu pemangku kepentingan non-program yang mungkin menemukan diagram kelas terlalu abstrak.

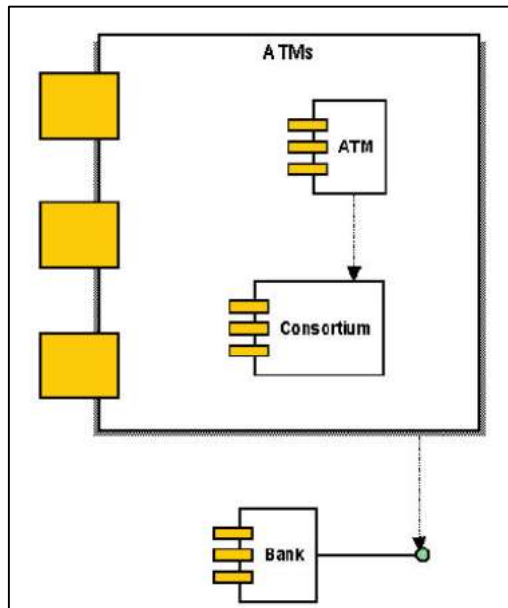


Gambar II.4. Object Diagram

(Sumber : Haviluddin, 2011 : 3)

c. *Component diagram*

Component diagram menggambarkan struktur fisik dari kode, pemetaan pandangan logis dari kelas proyek untuk kode aktual di mana logika ini dilaksanakan.

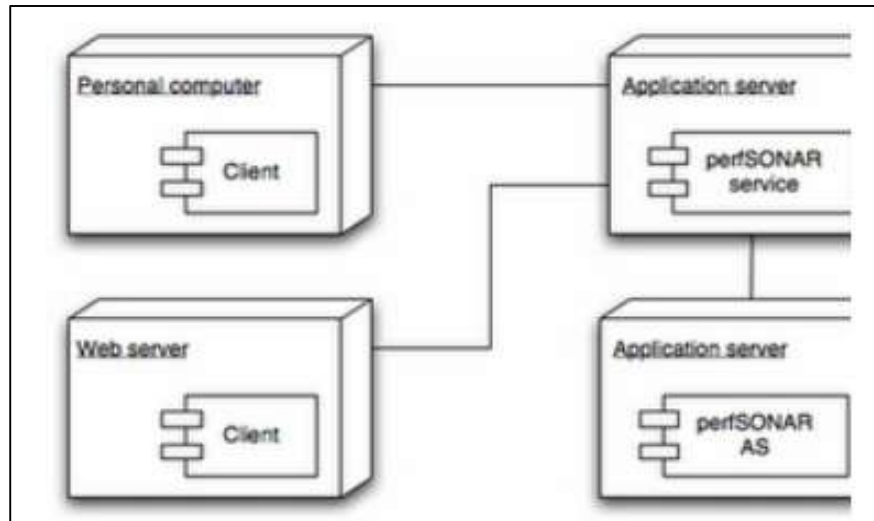


Gambar II.5. Component Diagram

(Sumber : Haviluddin, 2011 : 3)

d. *Deployment diagram (Collaboration diagram in version 1.x)*

Deployment diagram memberikan gambaran dari arsitektur fisik perangkat lunak, perangkat keras, dan artefak dari sistem. *Deployment diagram* dapat dianggap sebagai ujung spektrum dari kasus penggunaan, menggambarkan bentuk fisik dari sistem yang bertentangan dengan gambar konseptual dari pengguna dan perangkat berinteraksi dengan sistem.

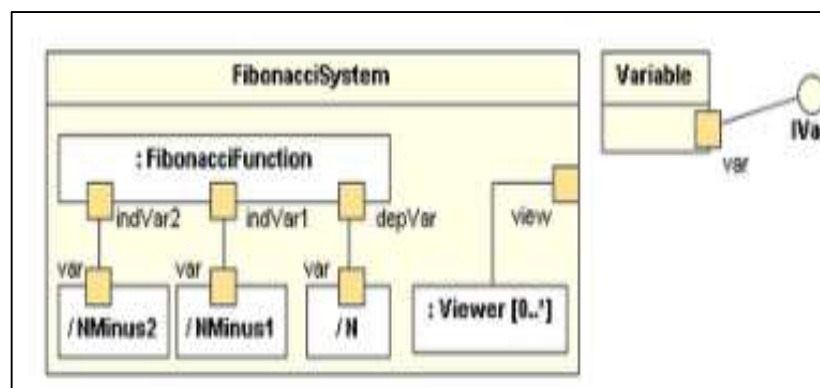


Gambar II.6. Notasi Deployment Diagram

(Sumber : Haviluddin, 2011 : 4)

e. *Composite structure diagram*

Sebuah diagram struktur komposit mirip dengan diagram kelas, tetapi menggambarkan bagian individu, bukan seluruh kelas. Kita dapat menambahkan konektor untuk menghubungkan dua atau lebih bagian dalam atau ketergantungan hubungan asosiasi.

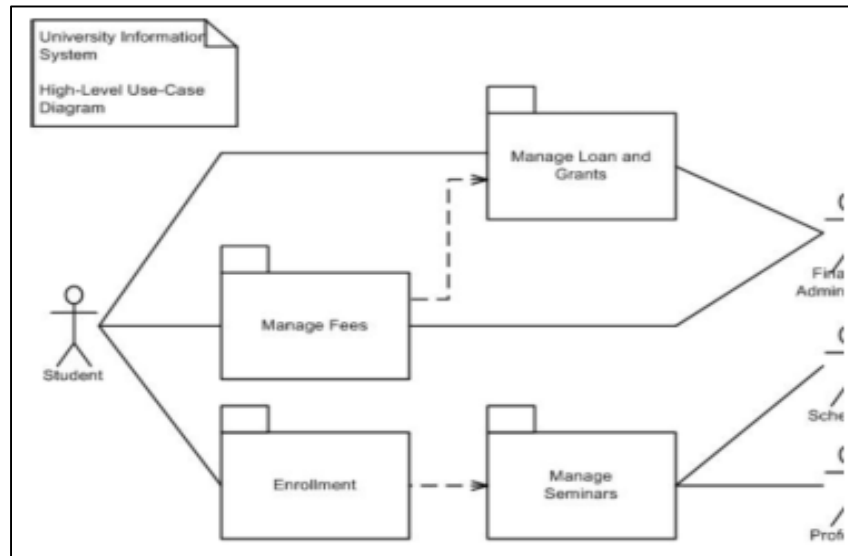


Gambar II.7. Notasi Composite Diagram

(Sumber : Haviluddin, 2011 : 4)

f. *Package diagram*

Paket diagram biasanya digunakan untuk menggambarkan tingkat organisasi yang tinggi dari suatu proyek *software*. Atau dengan kata lain untuk menghasilkan diagram ketergantungan paket untuk setiap paket dalam Pohon Model.



Gambar II.8. Notasi Package Diagram

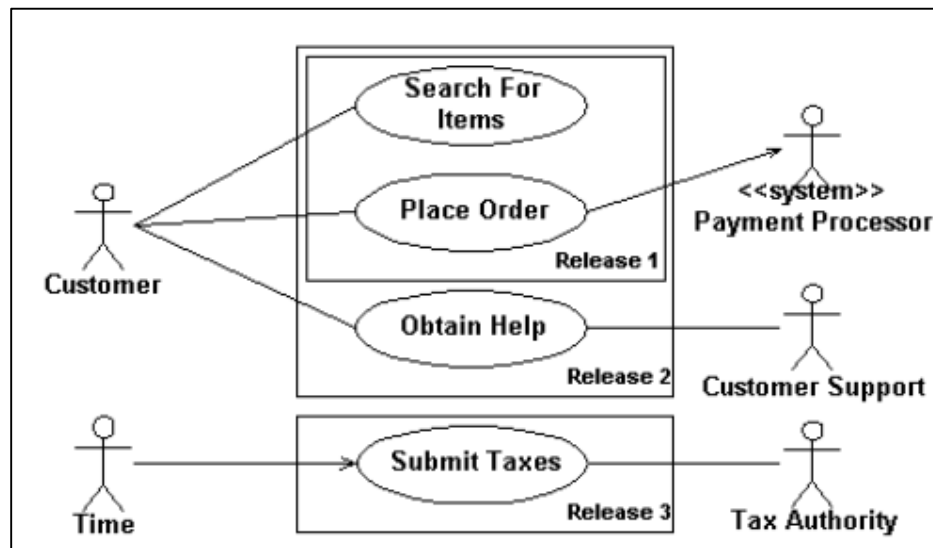
(Sumber : Haviluddin, 2011 : 4)

2. *Behavior Diagram*

Menggambarkan ciri-ciri *behavior*/metode/ fungsi dari sebuah sistem atau *business process*. *Behavior diagram* dalam UML terdiri atas :

a. *Use case diagram*

Diagram yang menggambarkan *actor*, *use case* dan relasinya sebagai suatu urutan tindakan yang memberikan nilai terukur untuk aktor. Sebuah *use case* digambarkan sebagai elips horizontal dalam suatu diagram UML *use case*.



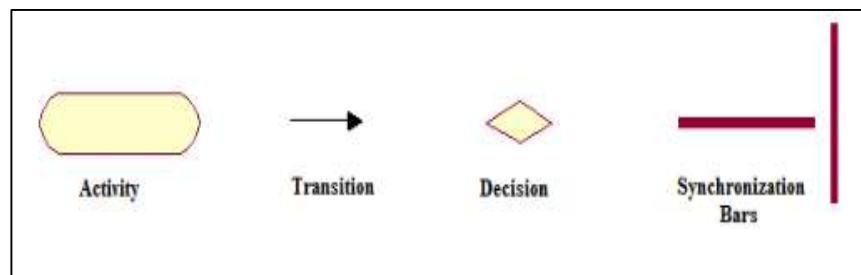
Gambar II.9. Use Case Diagram

(Sumber : Haviluddin, 2011 : 4)

b. *Activity diagram*

Menggambarkan aktifitas-aktifitas, objek, *state*, *transisi state* dan *event*.

Dengan kata lain kegiatan diagram alur kerja menggambarkan perilaku sistem untuk aktivitas.

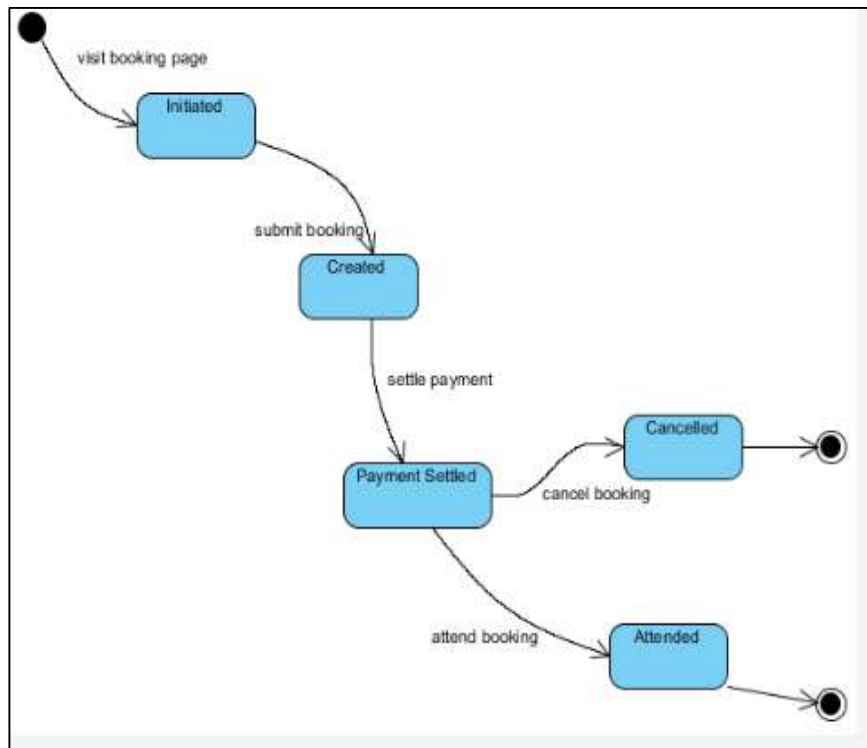


Gambar II.10. Notasi Activity Diagram

(Sumber : Haviluddin, 2011 : 4)

c. *State Machine diagram (State chart diagram in version 1.x)*

Menggambarkan *state*, *transisi state* dan *event*.



Gambar II.11. Notasi State Machine Diagram

(Sumber : Haviluddin, 2011 : 5)