

BAB II

TINJAUAN PUSTAKA

II.1. Sistem Informasi

II.1.1. Pengertian Sistem

Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu (JSIKA Vol 2 No 2 Kurnia, et al., 2013:8). Secara sederhana, suatu sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari user, komponen, atau variabel yang terorganisir, saling berinteraksi, saling tergantung satu sama lain, dan terpadu. Teori sistem secara umum yang pertama kali diuraikan oleh Kenneth Boulding, terutama menekankan pentingnya perhatian terhadap setiap bagian yang membentuk sebuah sistem (Sutabri, 2012:10).

Teori sistem mengatakan bahwa setiap unsur membentuk organisasi adalah penting dan harus mendapat perhatian yang utuh supaya manager dapat bertindak lebih efektif. Yang dimaksud unsur atau komponen membentuk organisasi di sini bukan hanya bagian-bagian yang tampak secara fisik, tetapi juga hal-hal yang mungkin bersifat abstrak atau konseptual seperti misi, pekerjaan, kegiatan, kelompok informasi, dan lain-lain sebagainya (Sutabri, 2012:10).

Selain itu, suatu sistem tidak bisa lepas dari lingkungan sekitarnya maka umpan balik (*feedback*) dapat berasal dari lingkungan sistem yang dimaksud. Organisasi dipandang sebagai suatu sistem yang tentunya akan memiliki semua unsur ini (Sutabri, 2012:10).

II.1.2. Pengertian Informasi

Informasi adalah data yang telah diproses menjadi bentuk yang memiliki arti bagi penerima dan dapat berupa fakta (JSIKA Vol 2 No 2 Kurnia, et al., 2013:8). Informasi adalah data yang telah diklasifikasi atau diinterpretasi untuk digunakan dalam proses pengambilan keputusan. Sistem pengolahan informasi mengolah data menjadi informasi atau tepatnya mengolah data dari bentuk tak berguna menjadi berguna bagi penerimanya (Sutabri, 2012:29).

II.1.3. Pengertian Sistem Informasi

Sistem informasi dapat didefinisikan sebagai prosedur-prosedur yang dapat digunakan untuk mengolah data sehingga dapat digunakan oleh penggunanya (JSIKA Vol 2 No 2 Kurnia, et al., 2013:8). Sistem informasi adalah suatu sistem di dalam organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan (Sutabri, 2012:46)..Dalam suatu sistem informasi terdapat komponen-komponen yang disebut blok adalah sebagai berikut:

1. Blok masukan (*input blok*), input mewakili data yang masuk kedalam sistem informasi.
2. Blok model (*model blok*), blok ini terdiri dari kombinasi prosedur, logika, dan model matematik yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara yang sudah tertentu untuk menghasilkan keluaran yang diinginkan.

3. Blok keluaran (*output blok*), produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.
4. Blok teknologi (*technology blok*), teknologi merupakan “tool box” dalam sistem informasi.
5. Blok basis data (*database blok*), merupakan kumpulan data yang saling berkaitan dan berhubungan satu sama lain (Sutabri, 2012:47).

II.2. Sistem Informasi Akuntansi

II.2.1. Akuntansi

Akuntansi mempunyai pengertian yang beraneka ragam menurut sudut pandang masing-masing ahli yang memberikan defenisi atas akuntansi. Secara umum akuntansi mempunyai suatu aktivitas jasa yang berfungsi menyediakan informasi kuantitatif tentang kondisi keuangan dan hasil operasi perusahaan yang diharapkan bermanfaat dalam mengambil keputusan ekonomis. Pengertian ini menekankan pada peran akuntansi, yaitu untuk memberikan informasi bagi kepentingan para pemakai daftar keuangan sebagai bahan pertimbangan dalam mengambil keputusan. Pengertian akuntansi tersebut merupakan akuntansi oleh *Financial Accounting Standart Board (FASB)*. *American Insetitite of Certified Public Accountants (AICPA)* melalui *Committee on Terminology* (1941) yang diterjemahkan oleh Hadibroto mendefinisikan akuntansi adalah suatu keahlian untuk mencatat, mengklasifikasikan dan mengikhtisarkan dengan cara yang tepat dan dinyatakan dengan uang, transaksi dan kejadian yang sebagian sekurang-

kurangnya bersifat keuangan dan menginterpretasikan hasil yang diperoleh (Jurnal Ilmiah Mahasiswa Akuntansi Vol. 1, No. 2 Renaldo, 2012:57-58).

II.2.2. Pengertian Sistem Informasi Akuntansi

Pada bagian ini akan dibahas definisi sistem informasi akuntansi dan istilah yang berkaitan dengan hal tersebut untuk membantu anda memahami seluruh materi dalam buku ini (anastasia, lilis, 2011:2).

Sistem informasi akuntansi adalah sistem yang bertujuan untuk mengumpulkan dan memproses data serta melaporkan informasi yang berkaitan dengan transaksi keuangan. Misalnya, salah satu input dari sistem informasi akuntansi pada sebuah toko baju, seperti pada contoh sebelumnya adalah transaksi penjualan. Kita memproses transaksi dengan mencatat penjualan tersebut kedalam jurnal penjualan, mengklasifikasikan transaksi dengan menggunakan kode rekening, dan memposting transaksi ke dalam jurnal. Kemudian, secara periodik sistem informasi akuntansi akan menghasilkan output berupa laporan keuangan yang terdiri dari neraca dan laporan laba rugi (anastasia, lilis, 2011:4-5).

II.2.3. Laporan Keuangan

Laporan keuangan pada dasarnya adalah merupakan hasil refleksi dari sekian banyak transaksi yang terjadi dalam suatu perusahaan (Jurnal Nominal Vol 1 No 1 Rosita, 2012:4). Transaksi-transaksi dan peristiwa-peristiwa yang bersifat finansial dicatat, digolongkan, dan diringkaskan dalam satuan uang, dan kemudian diadakan penafsiran untuk berbagai tujuan (Jurnal Nominal Vol 1 No 1 Rosita, 2012:4). Laporan keuangan merupakan sebagai hasil akhir dari proses akuntansi

atau hasil tindakan pembuatan ringkasan data keuangan perusahaan (Jurnal Nominal Vol 1 No 1 Rosita, 2012:4). Laporan keuangan ini disusun dan ditafsirkan untuk kepentingan manajemen dan pihak-pihak yang menaruh perhatian atau mempunyai kepentingan dengan data keuangan perusahaan tersebut (Jurnal Nominal Vol 1 No 1 Rosita, 2012:4).

Dalam integritas laporan keuangan terkandung kewajiban perusahaan untuk menyajikan dan melaporkan informasi keuangan dengan jujur dan benar. Hal ini harus menggambarkan dengan jujur dan transaksi serta peristiwa lainnya yang seharusnya disajikan atau secara wajar dapat diharapkan untuk disajikan. Integritas laporan keuangan hal yang paling penting yang harus diwujudkan dan dicapai perusahaan terutama dalam hal keuangan yaitu laporan keuangan sebagai media pertanggungjawaban oleh pihak manajemen kepada pemilik perusahaan.

Unsur-unsur laporan keuangan yang berkaitan langsung dengan pengukuran posisi suatu keuangan (Jurnal Ilmiah Mahasiswa Akuntansi Vol. 1 No. 1 Renaldo, 2012:58), yaitu :

1. Aset, merupakan sumber daya yang dikuasai oleh entitas sebagai akibat dari peristiwa masa lalu dan dari mana manfaat ekonomi dimasa depan diharapkan akan diperoleh entitas.
2. Kewajiban merupakan utang entitas masa kini yang timbul dari peristiwa di masa lalu, yang penyelesaiannya diharapkan mengakibatkan arus keluar dari sumber daya entitas yang mengandung manfaat ekonomi.

3. Ekuitas adalah hak residual atas aset entitas setelah dikurangi semua kewajiban (Jurnal Ilmiah Mahasiswa Akuntansi Vol.1 No.1 Renaldo, 2012:58).

Pada hakikatnya laporan keuangan bertujuan untuk memberikan informasi keuangan suatu perusahaan dalam periode tertentu. Laporan keuangan dapat disusun secara mendadak, artinya ketika ada permintaan untuk menampilkan laporan keuangan pada saat tersebut, maka pihak manajemen dapat langsung menyusun laporan keuangan tersebut (Jurnal Nominal Vol.1 No.1 Rosita, 2012:5).

II.2.3.1. Laporan Neraca

Laporan neraca digunakan untuk menggambarkan kondisi keuangan perusahaan. Neraca bisa digunakan sebagai gambaran potret kondisi keuangan suatu perusahaan pada suatu waktu tertentu, yang meliputi aset sumber daya perusahaan dan klaim atas aset tersebut meliputi utang dan saham pribadi (Jurnal Nominal Vol.1 No.1 Rosita, 2012:4).

II.2.3.2. Laporan Arus Kas

Laporan arus kas menyajikan informasi mengenai aliran yang masuk maupun keluar bersih pada suatu periode waktu yang merupakan hasil dari kegiatan pokok perusahaan, yaitu operasional, investasi dan pendanaan (Jurnal Nominal Vol.1 No.1 Rosita, 2012:5).

II.2.3.3. Laporan Laba Rugi

Laporan laba rugi merupakan bagian dari laporan keuangan suatu perusahaan yang dihasilkan pada suatu periode akuntansi yang menyajikan unsur-unsur pendapatan dan biaya perusahaan sehingga menghasilkan laba atau rugi bersih. Menurut (Juniady, 2001:159), laporan laba rugi merupakan laporan utama untuk melaporkan kinerja dari suatu perusahaan selama periode tertentu. Pengertian lain mengenai laporan laba rugi adalah laporan prestasi perusahaan selama jangka waktu tertentu (Jurnal Nominal Vol.1 No.1 Rosita, 2012:5).

New Hope Manufacturing Company Laporan Laba Rugi Untuk Bulan yang Berakhir pada Tanggal 31 Januari 20..		
Penjualan		\$ 384.000
Dikurangi harga pokok penjualan (Skedul 1)		<u>288.000</u>
Laba kotor		\$96.000
Dikurangi beban komersial :		
Beban pemasaran	\$ 20.800	
Beban administrative	<u>11.200</u>	32.000
Laba operasi		\$64.000
Dikurangi provisi pajak penghasilan		<u>26.000</u>
Laba bersih		<u>\$38.000</u>

Gambar II.1. Contoh Laporan Laba Rugi

(Sumber : William, 2009:119)

II.3. Basis Data

Basis data (*database*) merupakan kumpulan data yang saling berkaitan dan berhubungan satu dengan lainnya, tersimpan di perangkat keras komputer dan menggunakan perangkat lunak untuk memanipulasinya. Data perlu disimpan di

basis data untuk keperluan penyediaan informasi lebih lanjut. Data di dalam basis data perlu diorganisasikan sedemikian rupa supaya informasi yang dihasilkan berkualitas. Organisasi basis data yang baik juga berguna untuk efisiensi kapasitas penyimpanannya. Basis data diakses atau dimanipulasi menggunakan perangkat lunak paket yang disebut DBMS (*database management system*).

Dalam perkembangannya, basis data memiliki banyak pengertian. Hal ini didasarkan kepada fungsi dan tujuan dari basis data tersebut. Ada beberapa definisi dari basis data, yaitu :

1. Sistem basis data adalah suatu sistem menyusun dan mengelola *record-record* menggunakan komputer untuk menyimpan dan merekam serta memelihara data operasional lengkap sebuah organisasi/perusahaan sehingga pemakai mampu menyediakan informasi yang optimal yang diperlukan pemakai untuk proses mengambil keputusan (Marlinda, 2004:1).
2. Basis data adalah bahwa basis data merupakan kumpulan dari seluruh data berbasis komputer sebuah organisasi/perusahaan (Raymond, George, 2008:158).
3. Basis data adalah bagian dari sistem informasi, di dalamnya termasuk sumber daya yang dilibatkan dalam koleksi, manajemen, penggunaan dan disseminasi sumber daya informasi dari organisasi (Jurnal Ilmiah Foristek Vol.2, No.21 Yuri, Protus, 2012:191).

Pengertian-pengertian basis data tersebut diatas memberikan penekanan pada fungsi utama basis data, yakni pengorganisasian data. Hal ini dapat disimpulkan bahwa basis data merupakan kumpulan dari beberapa data yang

dikelola menjadi satu kesatuan yang saling berhubungan, disimpan dalam suatu tempat dan dapat dipergunakan di kemudian masa akan datang. *Database* merupakan salah satu komponen yang penting dalam sistem informasi, karena merupakan basis dalam menyediakan informasi bagi para pemakai. Penerapan *database* dalam sistem informasi disebut dengan *database system*. Sistem basis data (*database system*) adalah suatu sistem informasi yang mengintegrasikan kumpulan dari data yang saling berhubungan satu dengan yang lainnya dan membuatnya tersedia untuk beberapa aplikasi yang bermacam – macam di dalam suatu organisasi.

Basis data merupakan kumpulan dari beberapa elemen yang saling berhubungan. Pernyataan ini memberi penekanan bahwa basis data tidak akan tercipta jika tidak ada suatu elemen yang saling terkait. Ada beberapa komponen dasar dalam sistem basis data (Marlinda, 2004:2), yaitu :

1. Data.

Data di dalam sebuah basis data dapat disimpan secara terintegrasi dan dapat pula dipakai secara bersama-sama. Terdapat tiga jenis data, yaitu :

- a. Data Operasional
- b. Data Masukan
- c. Data Keluaran

2. *Hardware* (Perangkat Keras)

Terdiri dari semua peralatan komputer yang digunakan untuk pengelolaan sistem basis data yang berupa: peralatan untuk penyimpanan, peralatan masukan dan keluaran, dan peralatan komunikasi data.

3. *Software*(Perangkat Lunak)

Berfungsi sebagai perantara antara pemakai dengan data fisik pada basis data. Perangkat lunak dalam basis data berupa *Database Management System* (DBMS) atau program aplikasi prosedur.

4. *User* atau Pemakai

User atau pemakai dibagi atas tiga klasifikasi, yaitu :

a. *Database Administrator* (DBA)

Database Administrator (DBA) adalah orang atau tim yang bertugas untuk mengelola sistem basis data secara keseluruhan. Adapun tugas dari DBA adalah sebagai berikut :

- 1) Mengontrol DBMS dan *software-software*.
- 2) Memonitor siapa saja yang mengakses basis data.
- 3) Mengatur pemakaian basis data.
- 4) Memeriksa keamanan, *integrity*, *recovery* dan *concurrency*.

b. *Programmer*

Programmer adalah orang atau tim yang bertugas membuat program aplikasi, misalnya untuk perbankan, administrasi dan lain-lain.

c. *End User*

End User adalah orang yang mengakses basis data melalui terminal dengan menggunakan *query language* atau program aplikasi yang telah dibuat oleh *programmer*.

II.3.1. *Entity Relationship Diagram*

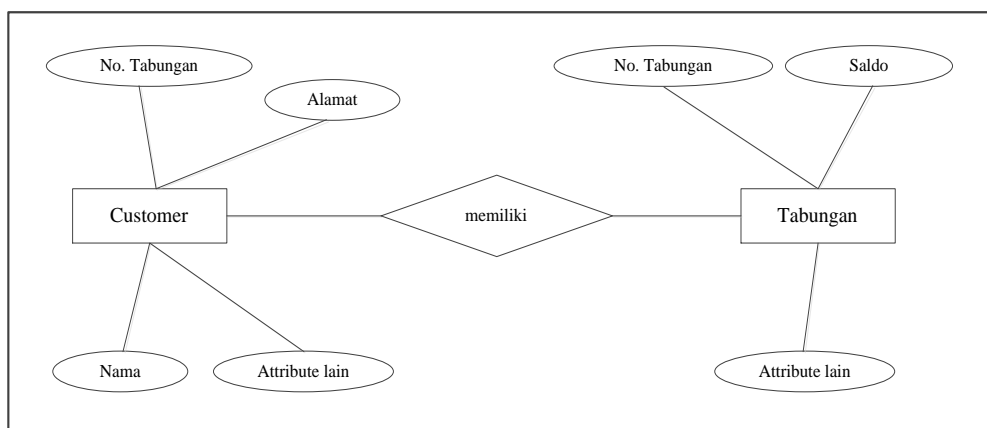
Entity relationship diagram merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data. Menurut Marlinda (2004:17), *model entity relationship* atau *entity relationship diagram* adalah suatu penyajian data dengan menggunakan *entity* dan *relationship*. Diperkenalkan pada tahun 1976 oleh P.P. Chen. Ada beberapa alasan diperlukannya *entity relationship diagram*, yaitu :

1. Dapat menggambarkan hubungan antar entitas dengan jelas.
2. Dapat menggambarkan batasan jumlah entitas dan partisipasi antar entitas.
3. Mudah dimengerti oleh pemakai.
4. Mudah disajikan oleh perancang basis data.

Entity relationship diagram terbentuk karena didukung oleh beberapa komponen yang saling berhubungan satu sama lain. Komponen-komponen yang terdapat pada *entity relationship diagram* adalah sebagai berikut :

1. *Entity*, suatu yang dapat dibedakan dalam dunia nyata dimana informasi yang berkaitan dengannya dikumpulkan. Simbol *entity* adalah persegi panjang.
2. *Relationship*, merupakan hubungan yang terjadi antara satu atau lebih *entity*.
3. *Attribute*, merupakan karakteristik dari *entity* atau *relationship* yang menyediakan penjelasan detail tentang hal tersebut. Nilai *attribute* adalah suatu data yang aktual.
4. Indikator Tipe, ada dua, yakni : indikator tipe *associative object* dan indikator tipe supertipe.

5. *Cardinality Rasio*, menjelaskan hubungan batasan jumlah keterhubungan satu *entity* dengan *entity* lainnya atau banyaknya *entity* yang bersesuaian dengan *entity* yang lain melalui *relationship*.
6. Derajat *Relationship*, menyatakan jumlah *entity* yang berpartisipasi di dalam suatu *relationship*.
7. *Participantion Constraint*, menjelaskan apakah keberadaan suatu *entity* tergantung pada hubungannya dengan *entity* yang lain.
8. Representasi dari *Entity Set*, dipresentasikan dalam bentuk tabel dan nama yang *unique*. Setiap tabel terdiri dari sejumlah kolom dan diberi nama yang *unique*.



Gambar II.2. Contoh Entity Relationship Diagram

(Sumber : Marlinda, 2004:25)

II.3.2. Normalisasi

Normalisasi merupakan proses pengelompokan elemen data menjadi tabel-tabel yang menunjukkan entitas dan relasinya. Teori normalisasi secara umum merupakan satu set peraturan yang membenarkan perkara pangkalan data mengenai kesalahan-kesalahan perkumpulan data yang tidak memuaskan dan

menentukan hubungan yang boleh ditukar menjadi bentuk yang lebih baik. Untuk menggunakan normalisasi yang baik, pangkalan data harus mengetahui maksud data-data. Ada beberapa bentuk normalisasi (Marlinda, 2004:122-123), yaitu :

1. Bentuk Tidak Normal (*Unnormalized Form*)

Bentuk ini merupakan kumpulan data yang akan direkam, tidak ada keharusan untuk mengikuti suatu format tertentu. Dapat saja data tidak lengkap atau terduplikasi. Data tersebut dikumpulkan apa adanya.

No_pasien
No_pen
Tanggal
Jalan
Kelurahan
Wilayah
Kecamatan
Kode_pos
Tilpun
Tanggal_periksa
Kode_dok
Nama_dok
Kode_sakit
Diskripsi_sakit
Kode_obat
Nama_obat

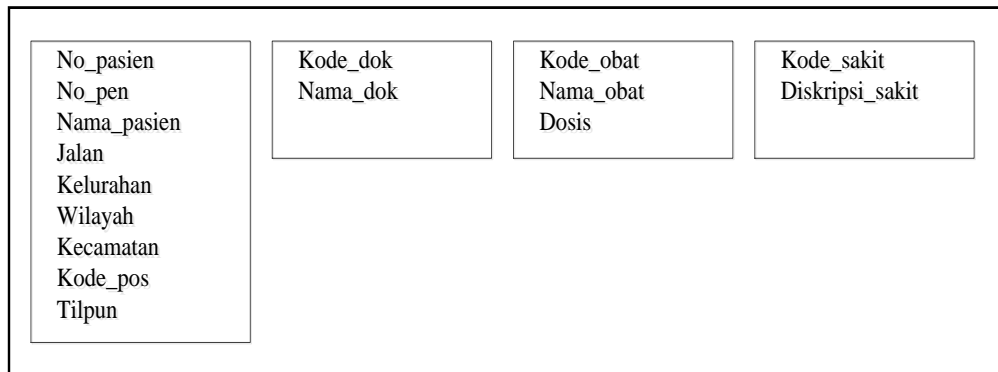
Gambar II.3. Bentuk Tidak Normal

(Sumber : Marlinda, 2004:130)

2. Bentuk Normal Pertama (1NF)

Suatu relasi 1NF jika dan hanya jika sifat dan setiap relasi atributnya bersifat atomik. Atomik bermaksud tidak berkepunyaan untuk berada dalam keadaan satu bagian. Ciri-ciri bentuk normal pertama, yaitu :

- a. Setiap data dibentuk dalam *flat file*.
- b. Tidak ada *set* atribut yang berulang atau bernilai ganda.
- c. Tiap *field* hanya satu pengertian.

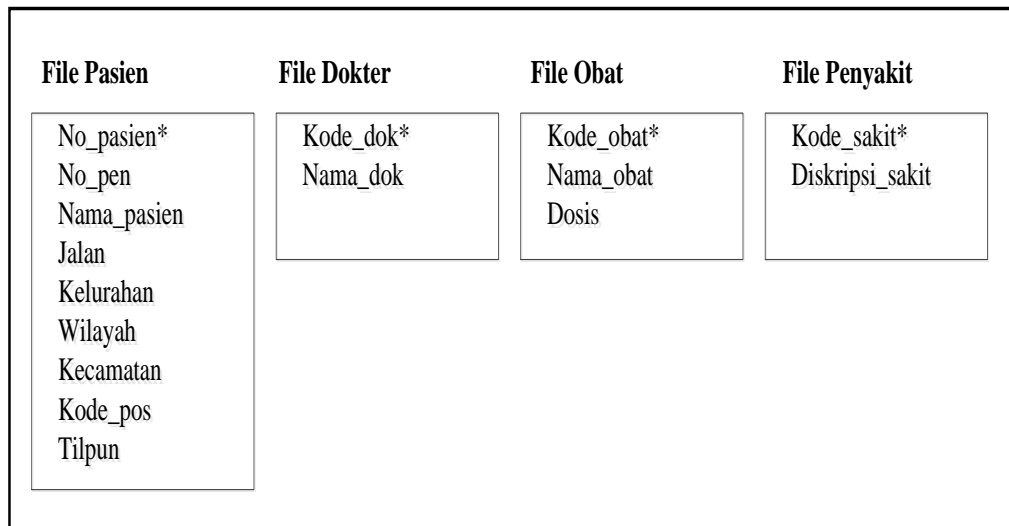


Gambar II.4. Bentuk Normal Pertama

(Sumber : Marlinda, 2004:130)

3. Bentuk Normal Kedua (2NF)

Bentuk normal kedua mempunyai syarat yaitu bentuk data telah memenuhi kriteria bentuk normal pertama. Atribut bukan kunci haruslah bergantung secara fungsi pada *primary key*. Peraturan ini menentukan kebergantungan sepenuhnya. Beberapa sumber teks menjelaskan sebagai kebergantungan secara fungsi dan transitif.

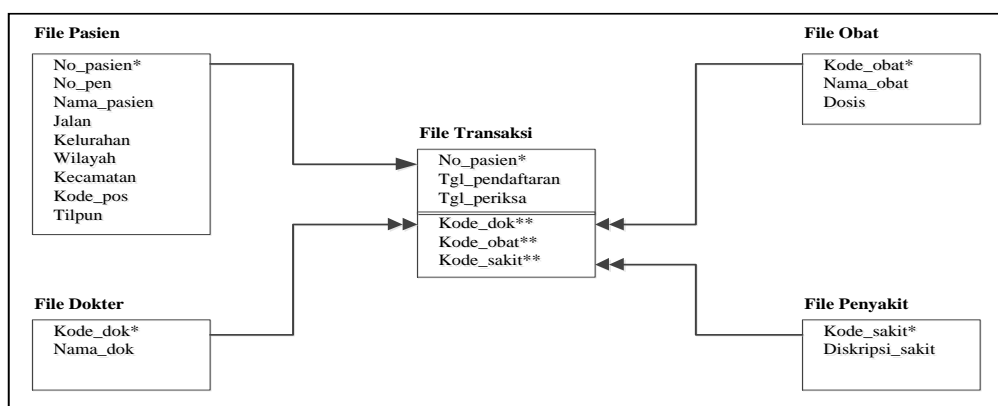


Gambar II.5. Bentuk Normal Kedua

(Sumber : Marlinda, 2004:131)

4. Bentuk Normal Ketiga (3NF)

Satu hubungan dikatakan dalam bentuk 3NF jika dan hanya jika ia dalam bentuk 2NF dan setiap atribut tanpa kunci pula bergantung secara tidak transitif dengan kunci primer.



Gambar II.6. Bentuk Normal Ketiga

(Sumber : Marlinda, 2004:131)

II.4. *Microsoft SQL Server*

Microsoft SQL Server merupakan salah satu *Relational Database Management System* (RDBMS) yang diproduksi oleh *Microsoft*. *SQL Server* adalah hasil kerja sama antara *Microsoft* dengan *Sybase* untuk memproduksi sebuah *software* penyimpan data (*database*). Beberapa komponen yang terdapat pada *SQL Server* adalah sebagai berikut (Santoso, 2005:18) :

1. *Relational Database Engine*, merupakan inti dari *SQL Server* yang menyediakan layanan prima, baik dalam bentuk penyimpanan data, permintaan data maupun pengamanan data itu sendiri.
2. *Analysis Services*, memfasilitasi analisa terhadap data tanpa mempengaruhi data yang ada.
3. *Data Transformation Services*, merupakan fasilitas transfer data yang dimiliki oleh *SQL Server* yang penggunaannya dapat melibatkan RDBMS dari produk yang berbeda.
4. *Notification Services*, mengirimkan berita seandainya suatu kejadian tertentu terjadi (teraktifkan) ke banyak jenis tipe *device*.
5. *Reporting Services*, merupakan fasilitas untuk memberikan laporan dalam bentuk HTML yang dapat diakses melalui *protocol* TCP/IP.
6. *Service Broker*, merupakan fasilitas untuk memberikan antrian layanan dari suatu piranti *software* ke piranti *software* lainnya.
7. *.NET Common Language Runtime* (CLR), merupakan salah satu fitur yang terdapat pada *SQL Server* yang memungkinkan untuk menjalankan program berbasis *.NET*.

8. *Native HTTP Support*, merupakan fasilitas yang memberikan layanan mirip *Internet Information Services (IIS)*.
9. *SQL Server Agent*, memberikan layanan yang sudah direncanakan sebelumnya.
10. *Replication*, memungkinkan untuk mengkopi dan mendistribusikan data dari suatu *database* ke *database* lainnya.








II.5. *Unified Modeling Language (UML)*

UML merupakan salah satu alat yang sangat handal di dunia pengembangan system yang berorientasi objek. Hal ini disebabkan karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembangan sistem untuk membuat cetak biru atas mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan mereka dengan yang lain.

Unified Modeling Language (UML) adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berparadigma berorientasi objek (Adi Nugroho, 2010:6). Hal ini disebabkan karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembangan sistem untuk membuat cetak biru atas mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan mereka dengan yang lain.

Ada beberapa pengklasifikasi (*Classifier*) dan notasi dalam UML (Adi Nugroho, 2010) yang ditunjukkan pada Tabel II.1.dibawah ini.

Tabel II.1. Pangklasifikasi dan Notasi UML

Pengklasifikasi	Kegunaan	Notasi
<i>Actor</i>	Menggambarkan semua objek diluar sistem yang berinteraksi dengan sistem yang dikembangkan.	
<i>Use Case</i>	Menggambarkan fungsionalistis yang dimiliki sistem.	
Kelas (<i>Class</i>)	Menggambarkan konsep dasar pemodelan sistem.	
Subsistem (<i>Subsystem</i>)	Menggambarkan paket spesifikasi serta implementasi.	
Komponen (<i>Component</i>)	Menggambarkan bagian-bagian fisik sistem/perangkat lunak yang dikembangkan.	
Antarmuka (<i>Interface</i>)	Menggambarkan antarmuka pengiriman pesan (<i>message</i>) antar pengklasifikasi.	
Simpul (<i>Node</i>)	Menggambarkan sumber daya komputasional yang digunakan oleh sistem.	

(Sumber : Adi Nugroho, 2010:16)

Relasi-relasi antar pengkalsifikasi yang dikenali UML adalah asosiasi, generalisasi, aliran, dan berbagai jenis kebergantungan termasuk didalamnya realisasi dan penggunaan. Untuk lebih jelasnya dapat dilihat pada Tabel II.2. berikut ini :

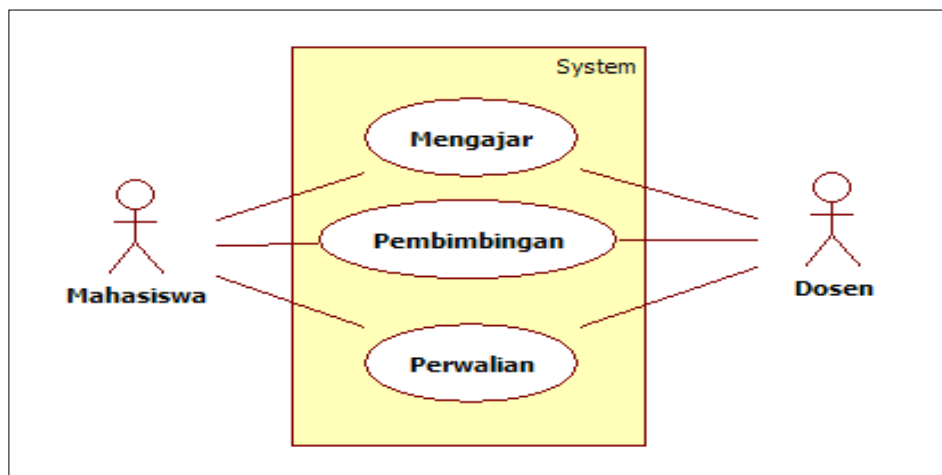
Tabel II.2. Relasi-relasi dalam UML

Relasi	Fungsi	Notasi
Asosiasi (<i>Association</i>)	Mendeskripsikan hubungan antar instance suatu kelas.	—————
Kerbergantungan (<i>Dependency</i>)	Relasi antar dua elemen model.	----->
Aliran (<i>Flow</i>)	Relasi antar dua versi suatu model.	----->
Generalisasi (<i>Generalization</i>)	Relasi antar pengklasifikasi yang memiliki deskripsi yang bersifat lebih umum dengan berbagai pengklasifikasi yang lebih spesifik, digunakan dalam struktur pewarisan.	—————>
Realisasi (<i>Realization</i>)	Relasi antara spesifikasi dan implementasinya.	----->
Penggunaan (<i>Usage</i>)	Situasi di mana salah satu elemen membutuhkan elemen yang lainnya agar dapat berfungsi dengan baik.	----->

(Sumber : Adi Nugroho, 2010:23)

II.5.1. Use Case Diagram

Diagram ini bersifat statis. Diagram ini memperlihatkan himpunan *use case* dan aktor-aktor (suatu jenis khusus dari kelas). Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku dari suatu sistem yang dibutuhkan serta diharapkan pengguna. *Use Case Diagram* adalah fungsionalitas atau persyaratan – persyaratan sistem yang harus dipenuhi oleh sistem yang akan dikembangkan tersebut menurut pandangan pemakai sistem (Sholih, 2010:21). *Use case* diagram juga dapat diartikan sebagai urutan langkah-langkah yang secara tindakan saling terkait (skenario), baik terotomatisasi maupun secara manual, untuk tujuan melengkapi satu tugas bisnis tunggal. *Use case* digambarkan dalam bentuk *ellips/oval*.

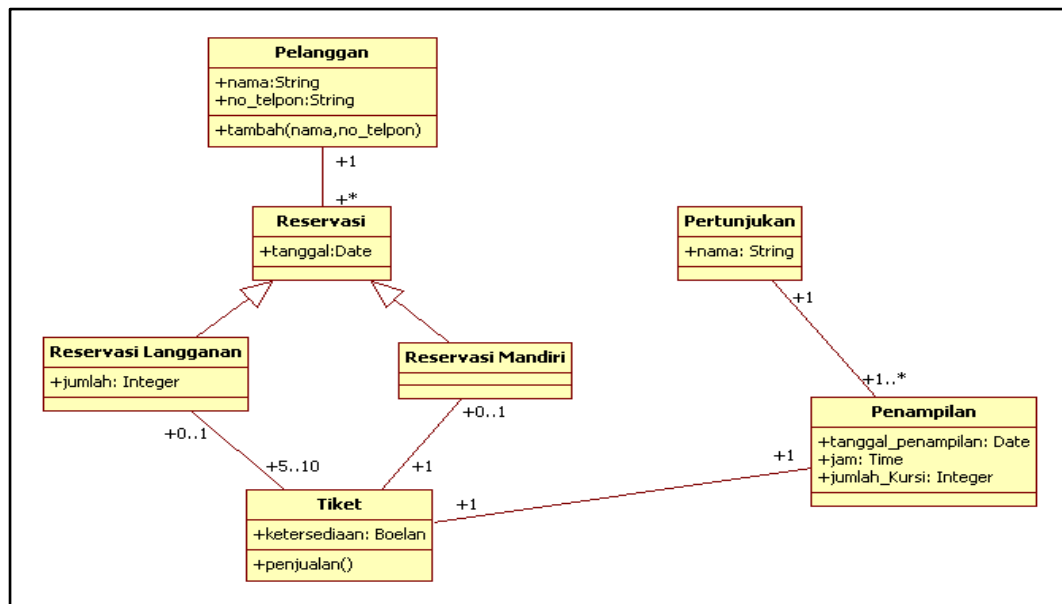


Gambar II.7. Contoh Use Case Diagram

(Sumber : Adi Nugroho, 2010:34)

II.5.2. Class Diagram

Class Diagram menunjukkan interaksi antar kelas – kelas dalam sistem. Sebuah kelas mengandung informasi dan tingkah laku (*behavior*) yang berkaitan dengan informasi tersebut. *Class diagram* sesungguhnya merupakan deskripsi dari konsep yang datang dari ranah aplikasi atau solusi aplikasi (Adi Nugroho, 2010 :11). Oleh karena itu pengertian kelas sangat penting sebelum merancang diagram kelas. Kelas diagram sebagai satu *set* objek yang memiliki atribut dan perilaku yang sama. *Class diagram* membantu dalam visualisasi struktur kelas – kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak. *Class diagram* memperlihatkan hubungan antar kelas dan penjelasan detail tiap – tiap kelas di dalam model desain (dalam *logical view*) dari suatu sistem. Selama proses analisis, *class diagram* memperlihatkan aturan – aturan dan tanggung jawab entitas yang menentukan perilaku sistem.

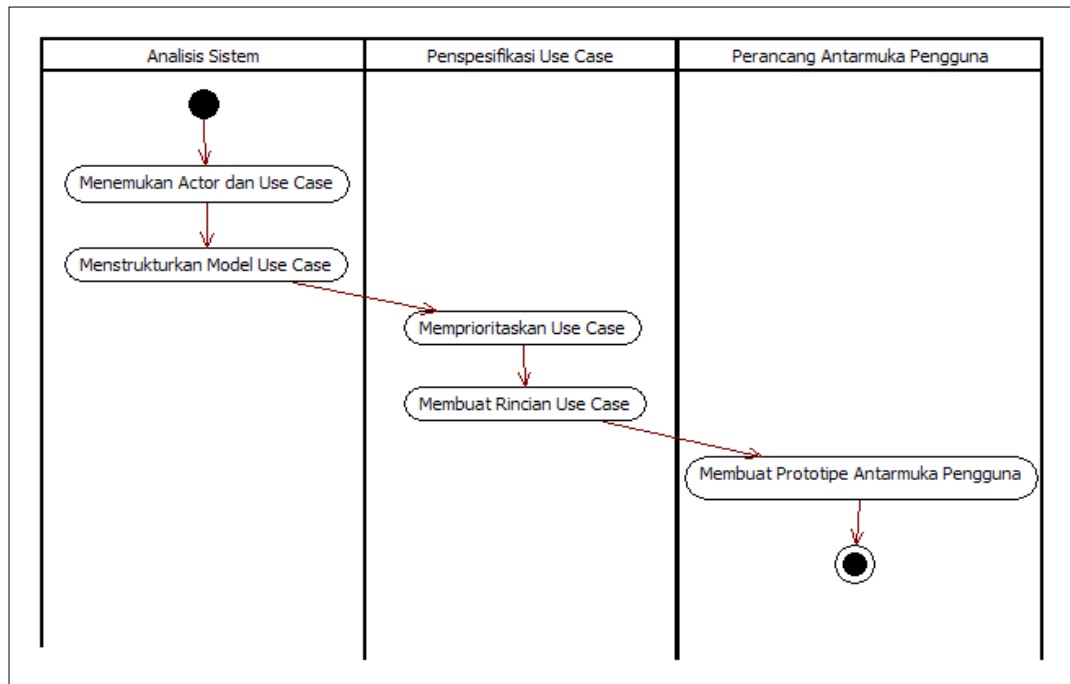


Gambar II.8. Contoh Class Diagram

(Sumber : Adi Nugroho, 2010:35)

II.5.3. Activity Diagram

Activity diagram memodelkan alur kerja (*workflow*) sebuah proses bisnis dan urutan aktivitas dalam suatu proses (Jurnal Teknologi Informasi Vol XIV, No.1, Sulistyorini, 2009:27). Diagram ini sangat mirip dengan sebuah *flowchart* karena dapat dimodelkan sebuah alur kerja dari satu aktivitas ke aktivitas lainnya atau dari satu aktivitas ke dalam keadaan sesaat (*state*). Seringkali bermanfaat bila dibuat sebuah *activity* terlebih dahulu dalam memodelkan sebuah proses untuk membantu memahami proses secara keseluruhan. *Activity diagram* juga sangat berguna ketika ingin menggambarkan perilaku paralel atau menjelaskan bagaimana perilaku dalam berbagai *use case* berinteraksi.



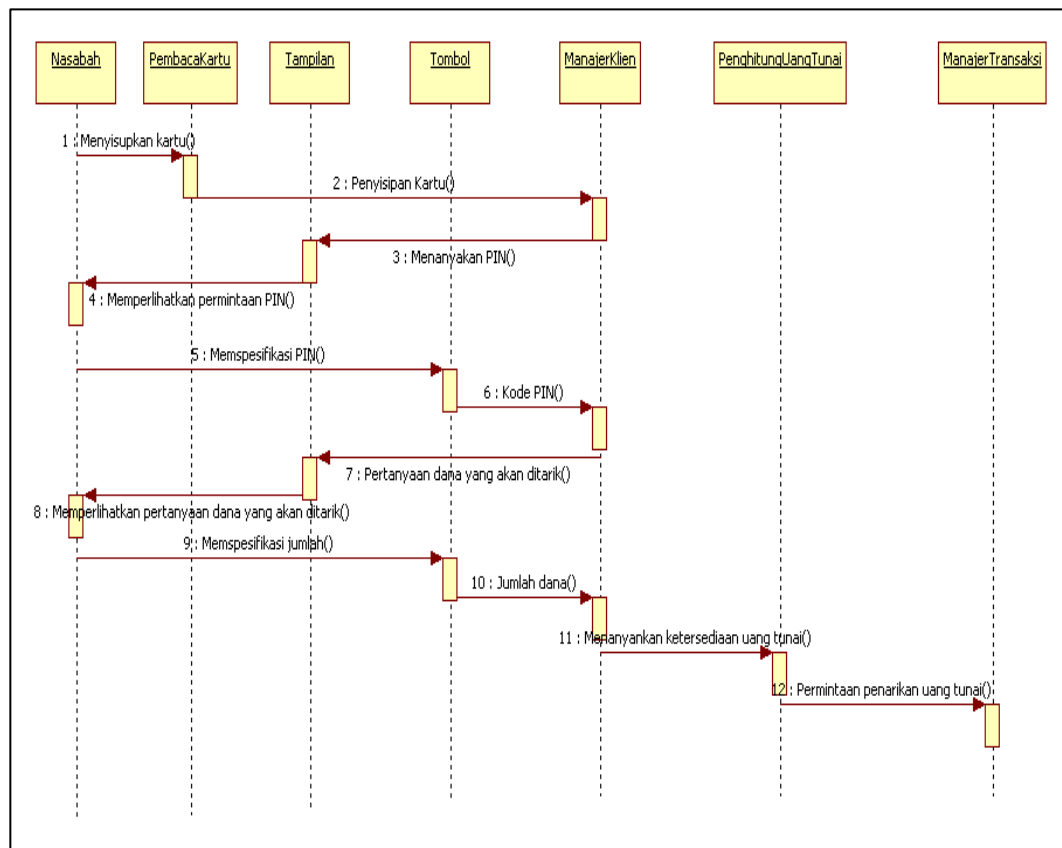
Gambar II.9. Contoh Activity Diagram

(Sumber : Adi Nugroho, 2010:63)

Dapat digunakan *statechart* diagram untuk memodelkan perilaku dinamis satu kelas atau objek. *Statechart* diagram memperlihatkan urutan keadaan sesaat (*state*) yang dilalui sebuah objek, kejadian yang menyebabkan sebuah transisi dari satu *state* atau aktivitas ke *state* atau aktivitas lainnya, dan aksi yang menyebabkan perubahan satu *state* lainnya, dan aksi yang menyebabkan perubahan satu *state* atau aktivitas. Diagram aktivitas paling cocok digunakan untuk memodelkan aktivitas dalam suatu proses.

II.5.4. *Sequence Diagram*

Sequence Diagram memperlihatkan interaksi sebagai diagram dua mantra (dimensi). Mantra vertical adalah sumbu waktu; waktu bertambah dari atas ke bawah sedangkan mantra horizontal memperlihatkan pengklasifikasi yang mempresentasikan objek-objek mandiri yang terlibat dalam kolaborasi. *Diagramsequence* merupakan diagram interaksi yang menekankan pada pengiriman pesan (*message*) dalam suatu waktu tertentu (Jurnal Teknologi Informasi Vol XIV, No.1, Sulistyorini, 2009:24). *Sequence Diagram* digunakan untuk menggambarkan perilaku pada sebuah skenario. Diagram ini menunjukkan sejumlah contoh objek dan *messege* (pesan) yang diletakkan diantara objek-objek ini dalam *use case*.



Gambar II. 10. Contoh Sequence Diagram

(Sumber : Adi Nugroho, 2010:42)

Ada beberapa komponen yang terdapat pada *sequence diagram*, yaitu :

1. Objek/*Participant*

Objek diletakkan di dekat bagian atas diagram dengan urutan dari kiri ke kanan. Objek ini diatur dalam urutan guna menyederhanakan diagram. Setiap *participant* terhubung dengan garis titik-titik yang disebut dengan *lifeline*. Sepanjang *lifeline* ada kotak yang disebut *activation*. *Activation* mewakili sebuah eksekusi operasi dari *participant*. Panjang kotak ini berbanding lurus dengan durasi *activation*.

2. *Messege*

Sebuah *messege* bergerak dari satu *participant* ke *participant* yang lain dan dari satu *lifeline* ke *lifeline*. Sebuah *participant* bisa mengirim sebuah *messege* kepada dirinya sendiri. Jika sebuah *participant* mengirimkan sebuah *messege synchronous*, maka jawaban atas *messege* tersebut akan ditunggu sebelum diproses dengan urusannya. Namun jika *messege synchronous* yang dikirimkan, maka jawaban atas *messege* tersebut tidak perlu ditunggu.

3. *Time*

Time adalah diagram yang mewakili waktu pada arah *vertical*. Waktu dimulai dari atas ke bawah. *Messege* yang lebih dekat dari atas akan dijalankan terlebih dahulu dibanding *messege* yang lebih dekat ke bawah.

II.6. Microsoft Visual Basic .Net

Visual Basic merupakan salah satu bahasa pemrograman yang andal dan banyak digunakan oleh pengembang untuk membangun berbagai macam aplikasi *windows* (Wahana Komputer, 2010:2). Net mempertahankan kemudahan dan kesederhanaan dari VB versi sebelumnya ditambah dengan kemampuan berorientasi obyek yang mengikuti keandalan C++. Kemampuan berorientasi obyek diantaranya mendukung abstraksi, *enkapsulasi*, *inheritance*, *constructors*, *polymorphism*, dan *overloading*.

Bahasa pemrograman *Visual Basic* tidaklah hanya identik dengan *Visual Basic* saja. Sistem Pemrograman *Visual Basic* dalam bentuk Edisi Aplikasi, telah dimasukkan ke dalam *Microsoft Excel*, *Microsoft Access*, dan banyak aplikasi

Windows lainnya juga menggunakan bahasa yang sama. *Visual Basic Scripting Edition (VBScript)* adalah sebuah bahasa skrip yang digunakan secara lebih umum dan merupakan bagian dari bahasa *Visual Basic*. Dengan mempelajari *Visual Basic*, maka Anda akan dibawa ke area-area yang telah disebutkan tadi.

Service-service atau pelayanan yang terdapat pada *Microsoft Visual Basic .NET*, antara lain:

1. Sebuah model pemrograman yang memungkinkan *developer* membangun XML *Web Service* dan aplikasinya.
2. Sekumpulan XML *Web Service* seperti *Microsoft .NET My Service* yang membantu pengembang menghasilkan aplikasi yang mudah dan terpadu.
3. Sekumpulan *server*, termasuk *Windows 2000* dan *2003*, *SQL Server 2005 64 Bit*, yang memadukan, menjalankan, dan mengoperasikan, serta menangani XML *Web Services* dan aplikasinya.
4. *Tool* seperti *Visual Basic .NET* untuk membangun XML *Web Service* dan aplikasi untuk *window* dan *web*.
5. Peranti lunak klien seperti *Windows XP* dan *Windows Vista*