BAB II

TINJAUAN PUSTAKA

II.1. Sistem

Sistem merupakan kumpulan aatau himpunan dari unsur, komponen, atau variable yang terorganisir, yang saling berkaitan yang memproses masukan (*input*) sehingga menghasilkan keluaran (*output*). (Tata Sutabri ; 2012 : 8)

II.1.1. Karakteristik Sistem

Sistem mempuyai beberapa karakteristik atau sifat-sifat tertentu yang mencirikan bahawa hal tersebut bisa dikatakan sebagai suatu sistem adapun karakteristik yang dimaksud adalah :

1. Komponen Sistem (*Components*)

Sistem terdiri dari sejumlah komponen yang saling berinteraksi, artinya saling bekerjasama membentuk satu kesatuan. Komponen sistem tersebut dapat berupa suatu bentuk subsistem. Setipa subsistem memiliki sifat dari sistem yang menjalankan susatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan. Suatu sistem dapat mempunyai sistem yang lebih besar atau serig disebut supra sistem.

2. Batasan Sistem (*Boundary*)

Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem yang lain atau sistem dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan yang tidak dapat di pisahkan.

3. Lingkungan Luar Sistem (*Environtment*)

Bentuk apapun yang ada di luar lingkungan atau batasan sistem yang mempengaruhi operasi sistem tersebut disebut lingkungan luar sistem. Lingkungan luar sistem ini dapat bersifat menguntungka dan dapat juga bersifat merugikan sistem tersebut. Dengan demikian, lingkungan luar tersebut harus tetap di jaga dan di pelihara.

4. Penghubung Sistem (*Interface*)

Media yang menghubungkan sistem dengan subsistem lain disebut penghubung sistem atau *interface.* Penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem lain. Bentuk keluaran dari satu subsistem akan menjadi masukan untuk subsistem lain melalui penghubung tersebut.

5. Masukan Sistem (*Input*)

Energi yang dimasukan ke dalam sistem disebut memasukan sistem, yang dapat berupa pemeliharaan (*maintenance input*) dan sinyal *(signal input*). Contoh, di dalam suatu unit sistem computer, program adalah *maintenance input* yang digunakan untuk mengoperasi komputernya dan data adalah *signyal input* untuk diolah menjadi informasi.

6. Keluaran Sistem (*Output*)

Hasil energi yang diolah dan di klasifikasikan menjadi keluaran yang berguna. Keluaran ini merupakan masukan bagi subsistem yang lain seperti sistem informasi. Keluaran yang dihasilkan adalah informasi. Informasi inin dapat

digunakan sebagai masukan untuk pengambilan keputusan atau hal-hal yang menjadi input bagi subsistem lain.

7. Pengolah Sistem (*Proses*)

Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran, contohnya adalah sistem akuntansi. Sistem ini akan mengolah data transaksi menjadi laporan-laporan yang di butuhkan oleh pihak manajemen.

8. Sasaran Sistem (*Objective*)

Suatu sistem memliliki tujuan dan sasaran yang pasti dan bersifat *deterministic.* Kalau suatu sistem tidak memiliki sasaran maka operasi sistem tidak ada gunanya. Suatu Sistem dikatakan berhasil bila mengenai sasaran yang telah di rencanakan. (Tata Sutabri ; 2012 : 20)

II.1.2. Klasifikasi Sistem

Sistem merupakan suatu bentuk integrasi satu komponen dengan komponen lain karena sistem memiliki sasaran yang berbeda untuk setiap kasus yang terjadi yang ada di dalahm sistem tersebut. Oleh karena itu, sistem dapat di klasifikasikan dari bebeapa sudut pandang di antaranya :

1. Sistem Abstarak dan Sistem fisik

Sistem Abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik, misalnya sistem teologia, yaitu sistem yang berupa pemikiran hubungan antara manusia dengan tuhan, sedangkan sistem fisik merupakan sistem yang ada secara fisik, misalnya sistem computer, sistem produksi, sistem penjualan dan lain sebagainya.

2. Sistem alamiah dan sistem buatan manusia

Sistem alamian adalah sistem yang berupa proses alam, tidak dibuat oleh manusia. Sedangkan sistem buatan manusia merupakan sistem yang melibatkan interaksi manusia dengan mesin yang disebut *human machine sistem*, contohnya sistem informasi berbasis computer.

3. Sistem determinasi dan sistem probabilistic

Sistem yang beroperasi dengan tingkah laku yang dapat di prediksi disebut sistem *deterministic*. Sistem computer adalah contoh dari sistem yang tingkah lakunya dapat dipastikan berdasarkan program-program computer yang dijalankan, sedangkan sistem yang bersifat probabilistic adalah sistem yang kondisinya tidak dapat diprediksi karena mengandung unsur *probabilistic*.

4. Sistem terbuka dan sistem tertutup

Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak tepengaruh oleh lingkungan luarnya, Sistem ini bekerja secara otomatis tanpa campur tangan pihak luar, seangan sistem erbuka adalah sistem yang berhubungan dan di pengaruhi oleh lingkungan luarnya. Sistem ini menerima masukan dan menghasilkan keluaran untuk subsistem lainnya. (Tata Sutabri ; 2012 : 22)

II.2. Informasi

Informasi adalah data yang telah diproses menjadi bentuk yang memiliki arti bagi penerima dan dapat berupa fakta, suatu nilai yang bermanfaat.Jadi ada suatu proses transformasi data menjadi suatu informasi input - proses – output. (Tata Sutabri ; 2012 : 1)

II.3. Sistem Informasi

Sistem informasi adalah suatu sistem dimana di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kapada pihak luar tertentu dengan laporan – laporan yang diperlukan. (Tata Sutabri ; 2012: 9)

II.3.1. Komponen dan jenis sistem informasi

Sistem informasi terdiri dari komponen-komponen yang disebut blok bangunan *(building block)* yang masing-masing saling beriteraksi satu dengan yang lain membentuk suatu kesatuan untuk memcapai sasaran dan komponen-komponen itu adalah :

1. **Block masukan (*input block)***

   Input mewakili data yang masuk kedalam sistem informasi. *Input* yang dimaksud adalah metode dan media untuk menangkap data yang akan dimasukan, yang dapat berupa dokumen-dokumen dasar.

2. **Block model (*model bock)***

   Block 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. Block keluaran (*output block)*

Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen sert semua pemakai sistem.

4. Block teknologi (*technology block)*

Teknologi merupakan *tool box* dalam sistem informasi. Teknologi digunakan untuk menerima *input*, menjalankan model, menyimpan dang mengakses data, menghasilkan dan mengirim keluaran, dan membantu pengendalian dari sistem secara keseluruhan.

5. Block basis data (*database block)*

Basis data merupakan kumpulan data yang saling berkaitan dan berhubungan satu sama lain, tersimpan di perangkat keras computer dan menggunakan perangkat lunak untuk memanipulasinya. Data perlu disimpan dalam basis data untuk keperluan penyediaan informasi lebih lanjud. Data si dalam basis data data perlu diorganisasikan sedemikian rupa supaya informasi yang di hasilkan berkualitas.

6. Block kendali (*control block)*

Beberapa pengendalian perlu dirancang dan diterapkan untuk meyainkan bahwa hal-hal yang dapat merusak sistem dapat dicegah ataupun bila terlanjur terjadi kesalahan-kesalahan dapat langsung cepat diatasi.(Tata Sutabri ; 2012 : 47)

II.4. Data

Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Kesatuan yang nyata adalah brupa objek yang nyata seperti tempat, benda, dan orang yang betul-betul ada dan terjadi. Dapat disimpulkan bahwa data adalah bahan mentah yang di proses untuk menyajikan informasi. (Tata Sutabri ; 2012 : 2)

II.4.1. Klasifikasi Data

Informasi sangat erat hubungannya dengan data. Informasi berasal dari data. Sedangkan data itu sendiri dapat diklasifikasikan menurut jenis, sifat dan sumber seperti di jelaskan di bawah ini :

1. **Klasifikasi data menurut jenis data**
   a. **Data hitung (*enumeration/counting data)***

      Data hitung adalah hasil penghitungan atau jumlah tertentu. Yang termasuk data hitung adalah persentase dari jumlah tertentu.

   b. **Data ukur(*measurement data)***

      Data ukur adalah data yang menunjukan ukuran mengenai nilai sesuatu. Angka tertentu atau huruf tertentu yang diberikan oleh seorang dosen kepada seorang mahasiswa setelah memeriksa hasil tentemennya merupakan data ukur.

2. Klasifikasi menurut sifat data

a. Data kuantitatif (*quantitative data)*

Data kuantitatif adalah data mengenai penggolongan dalam hubungannya dengan penjumlahan.

b. Data kualitatif (*qualitative data)*

Data kualitatif adalah data mengenai penggolongan dalam hubungannya dengan kualitas atau sifat sesuatu.

3. Klsifikasi data menurut sumber data

a. Data internl (*Internal data)*

Data internal adalah data asli artinya data sebagai hasil observasi yang di lakukan sendiri, bukan hasil karya orang lain**.**

b. Data eksternal (*external data)*

Data eksternal adalah hasil observsi orang lain. Seseorang boleh saja menggunakan data untuk suatu keperluan , meskipun data tersebut hasil kerja orang lain**.** (Tata Sutabri ; 2012 : 3)

II.5. Sistem Informasi Akuntansi

Sistem informasi akuntansi adalah sistem yang bertujuan untuk mengumpulkan dan memproses data serta melaporkan informasi yang berkaitan dengan transaksi keuangan. (Anastasia Diana ; 2011: 4)

II.6. Arus Kas (*Cash Flow*)

Cash Flow adalah laporan yang menghitung arus kas masuk dan arus kas keluar dari sebuah perusahaan selama periode tertentu. Laporan arus kas menyediakan informasi yang berguna mengenai kemampuan perusahaan untuk menghasilkan kas dari kegiatan operasi, memertahankan dan meningkatkan kapasitas operasi, sering digunakan untuk mengevaluasi kegiatan operasi yang telah lalu dan dalam membuat perencanaan investasi dan kegiatan pendanaan di masa depan. (Salemba Empat ; 2010: 262)

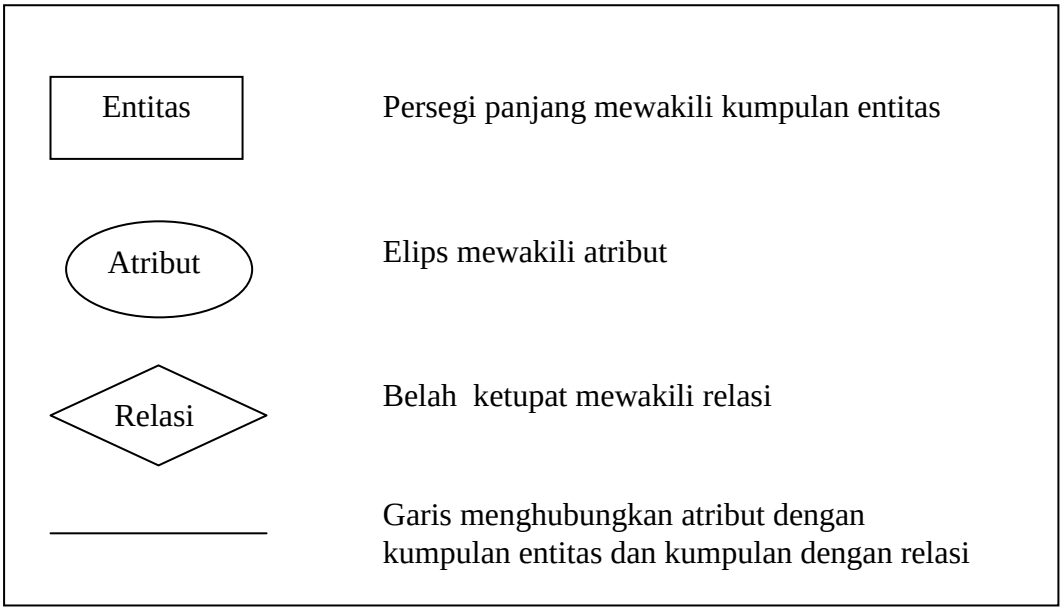
II.8. Basis data (database)

Basis data merupakan kumpulan data yang saling berkaitan dan berhubungan satu sama lain, tersimpan di perangkat keras computer dan menggunakan perangkat lunak untuk memanipulasinya. Data perlu disimpan dalam basis data untuk keperluan penyediaan informasi lebih lanjut. Data si dalam basis data data perlu diorganisasikan sedemikian rupa supaya informasi yang di hasilkan berkualitas. (Tata sutabri ;2012: 47).

II.8.1. Entity Relationship (ER)

Entitiy Relationship data model didasarkan pada persepsi terhadap dunia nyata yang tersusun atas kumpulan objek-objek dasar yang disebut entitas dan hubungan antar objek. Entitas adalah sesuatu ata objek dalam dunia nyata yang dapat dibedakan dari objek lain. Sebagai contoh, masing-masing mahasiswa adalah entitas dan mata kuliah dapat pula dianggap sebagai entitas. Relasi adalah hubungan antara beberapa entitas. Sebagai contoh, relasi menghubungkan

mahasiswa dengan mata kuliah yang diambilnya. Kumpulan semua entitas bertipe sama disebut kumpulan entitas *(entity set*), sedangkan kumpulan semua relasi bertipe sama disebut kumpulan relasi (*relationship set).* Komponen-komponen diagram dijelaskan pada Gambar dibawah ini :



Gambar II.1 Komponen ER
(*Sumber : Janner Simarmata; 2010)*

II.8.2. Entity Relationship Diagram (ERD)

Entity Relationship Diagram adalah alat pemodelan data utama dan akan membantu mengorganisasi data dalam suatu proyek ke dalam entitas-entitas dan menentukan hubungan antar entitas. Proses memungkinkan analis menghasilakn struktur basisdata yang baik sehingga data dapat disimpan dan diambil secara efisien. (Janner Simarmata ; 2010: 67).

1. Entitas *(Entity)*

Entitas adalah sesuatu yang nyata atau abstrak dimana kita akan menyimpan data. Ada 4 kelas entitas, yaitumisalnya pegawai, pembayaran, kampus, dan buku. Contoh suatu entitas disebut instansi, misalnya pegawai andi, pembayaran joko, dan lain sebagainya.

2. Relasi (*Relationship)*

Relasi adalah hubungan alamiah yang terjadi antara satu atau lebih entitas, misalnya proses pembayaran pegawai, kardinalitas menentukan kejadian suatu entitas untuk satu kejadian pada entitas yang berhubungan. Misalnya mahasiswa bisa mengambil banyak mata kuliah.

3. Atribut (*Attribute)*

Atribut adalah ciei umum semua atau sebagian besar instansi pada entitas tertentu. Sebutan lain atribut adalah property, elemen data, dan fiels. Misalnya nama, alamat, nomor pegawai, dan gaji adalah atribut entitas pegawai. Sebuah atribut atau kombinasi atribut yang mengidentifikasikan satu dan hanya satu instansi suatu entitas disebut kunci utama atau pengenal. Misalnya nomor pegawai adalah kunci utama untuk pegawai. (Janner Simarmata ; 2010: 67).

II.8.2. Normalisasi

Normalisasi adalah teknik perancangan yang banyak digunakan sebagai pemandu dalam merancang basisdata relasional. Pada dasarnya, normalisasi adalah proses dua langkah yang meletakkan data dalam bentuk tabulasi dengan menghilangkan kelompok berulang lalu menghilangkan serta yang terduplikasi dari table relasional. (Janner Simarmata ; 2010: 76).

Adapun tujuan yang yang harus di capai yaitu :

1. Mengatur data dalam kelompok-kelompok sehinga masing-masing kelompok menangani bagian kecil sistem.
2. Meminimalkan jumlah data berulan dalam basisdata.
3. Membuat basisdata yang datanya diakses dan dimanipulasi secara ceatdan efisien tanpa melupakan integritas data.
4. Mengatur data sedemikian rupa sehingga ketika memodifikasi data, anda hanya mengubah pada satu tempat.

Tujuan normalisasi adalah membuat kumpulan table relasional yang bebeas dari data berulang dan dapat dimidifikasi secara benar dan konsisten. Ini berarti bahwa semua table pada basisdata relasional harus berada pada bentuk nomal ketiga (3NF).

Normalisasi sendiri dilakukan melalui sejumlah langkah. Setiap langkah berhubungan dengan bentuk normal tertentu di antaranya :

1. Bentuk Normal pertama (1NF)

Contoh sebuah perusahaan yang mendapatkan barang dari sejumlah pemsok. Msing-masing pemasok berada ada satu kota. Sebuah kota dapat mempunyai lebih dari satu pemasok dan masingmasing kota mempunyai kode status tersendiri.

2. Bentuk Normal kedua (2NF)

Menyatakan bahwa table dengan kunci utama gabungan hanya dapat berada pada 1NF, tetapi tidak pada 2NF. Sebuah table relasional berada pada bentuk normal kedua jika dia berada pada 1NF dan setiap klomnya bukan kunci yang

sepenuhnya tergantung pada kunci utama.Ini berarti bahwa setiap kolom bukan kunci harus tergantung pada seluruh kolom yang membentuk kunci utama.

3. Bentuk Normal ketiga (3NF)

Betuk normal yang ketiga mengharuskan semua kolom pada table relasional tergantung hanya pada kunci utama. Secara definisi, sebuah table berada pada bentuk normal ketiga (3NF) jika table sudah berada pada 2NF dan setiap kolom yang bukan kunci tidak tergantung secara transitif pada kunci utamanya. Dengan kata lain, semua atribut bukan kunci tergantung secara fungsional hanya pada kunci utama.

4. Bentuk Nomal Boyce-Conde (BCNF)

Bentuk normal boyce-code adalah versi 3NF yang lebih teliti dan berhubungan dengan table relasional yang mempunyai banyak kunci kendidat, kunci kandidat gabungan, dan kunci kandidat yang saling tumpang tindih. CNF didasarkan pada konsep penentu. Sebuah kolom penentu adalah kolom dimana kolom-kolom lain sepenuhya secara fungsional. Sebuah table relasional brada pada BCNF jika hanya setiap penentu adalah kunci kandidat.

5. Bentuk Noraml keempat (4NF)

Sebuah table relasional berada pada bentuk normal keempat (4NF jika dia dalan BCNF dan semua ketergantungan multivalue merupakan ketergantungan fungsional. Bentuk normal keempat (4NF) didasarkan pada konsep ketergantungan multivalue (MVN). Sebuah ketergantungan multivalue terjadi ketikadalam sebuah table relasional yang mengandung setidaknya tiga kolom, satu kolom mempunyai banyak baris bernilai sama, tetapi kolom lain bernilai berbeda.

6. Bentuk Normal kelima (5FN)

Empat bentuk normal pertama berdasarkan pada konsep ketergantungan fungsional, Sedangkan bentuk normal kelima berdasarkan pada konsep ketergantungan gabungan (*join dependence).* Ketergantungan gabungan berarti bahwa sebuah table, setelah didekomposisi menjadi tiga atau lebih table yang lebih kecil, harus dapat digabungkan kembali untuk membentuk table asal. Dengan kata lain, 5NF menunjukan ketik sebuah table tidak dapat di dekomposisi lagi. (Janner Simarmata ; 2010: 86).

II.9. *Unified Modeling Language* (UML)

UML singkatan dari Unified Modeling Language yang berarti bahasa peodelan standar. Unified Modeling Language (UML) adalah sebuah bahasa yang berdasarkan grafik atau gambar untuk menvisualisasikan, menspesifikasikan, membangun dan pendokumentasian dari sebuah sistem pengembangan perangkat lunak berbasis Objek OOP *(*Object Oriented programming). (Herlawati ; 2010: 7)

UML diaplikasikan untuk maksud tertentu, biasanya antara lain untuk :

1. Merancang perangkat lunak.
2. Sarana komunikasi antara perangkat lunak dengan proses bisnis.
3. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
4. Mendokumentasi sistem yang ada, proses-proses dan organisasinya.

II.9.1. Diagram UML

Beberapa literature menyebutkan bahwa UML menyediakan Sembilan jenis diagram, yang lain menyebutkan delapan kerena ada beberapa diagram yang digabung, namun demikian diagram itu bisa dikelompokan berdasarkan sifatnya yaitu stasis atau dinamis**.** Adapun jenis-jenis diagram antara lain :

1. **Diagram kelas**

Bersifat statis. Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, serta relasi-relasi. Diagram ini umum dijumpai pada pemodelan sistem berorientasi objek. Meskipun bersifat statis, sering pula diagram kelas membuat kelas-kelas aktif.

2. **Diagram paket (package diagram)**

Bersifat statis diagram ini memperlihatkan kumpulan kelas-kelas merupakan bagian dari diagram komponen.

3. **Diagram use case**

Bersifat statis siagram ini memperlihatkan himpunan use case dan actor-aktor. Diagram ini terutama sangat oenting untuk mengorganisasi dan memodelkan perilaku suatu sistem yang dibutuhkan serta di harapkan pengguna.

4. **Diagram interaksi dan *sequence*(urutan)**

Bersifat dinamis diagram ini urutan adalah diagram interaksi yang menekankan oada pengiriman pesan dala suati waktu tertentu.

5. Diagram komunikasi (*communication diagram*)

Bersifat dinamis diagram sebagai pengganti diagram kolaborasi UML 1.4 yang menekankan organisasi struktual dari objek-objek yang menerima serta mengirim pesan.

6. Diagram Statechart (*Statechart diagram*)

Bersifat dinamis diagram status memperlihatkan keadaan-keadaan pada sistem, membuat status (state), transisi,kejadian serta aktifitas. Diagram ini terutama penting untuk memperlihatkan sifat dinamis dari antarmuka (interface), kelas, lolaborasi, dan terutama penting pada pemodelan sistem-sistem yang reaktif.

7. Diagram Aktifitas (*Activity diagram*)

Bersifat dinamis diagram ini adalah tipe khusus dari diagram status yang memperlihatkan aliran dari suatu aktifitas ke aktifitas lainnya dalam suatu sistem. Diagram ini trutama penting dalam pemodelan fungsi-fungsi suatu sistem dan memberi tekanan pada aliran kendali antar objek.

8. Diagram Komponen (*Component Diagram*)

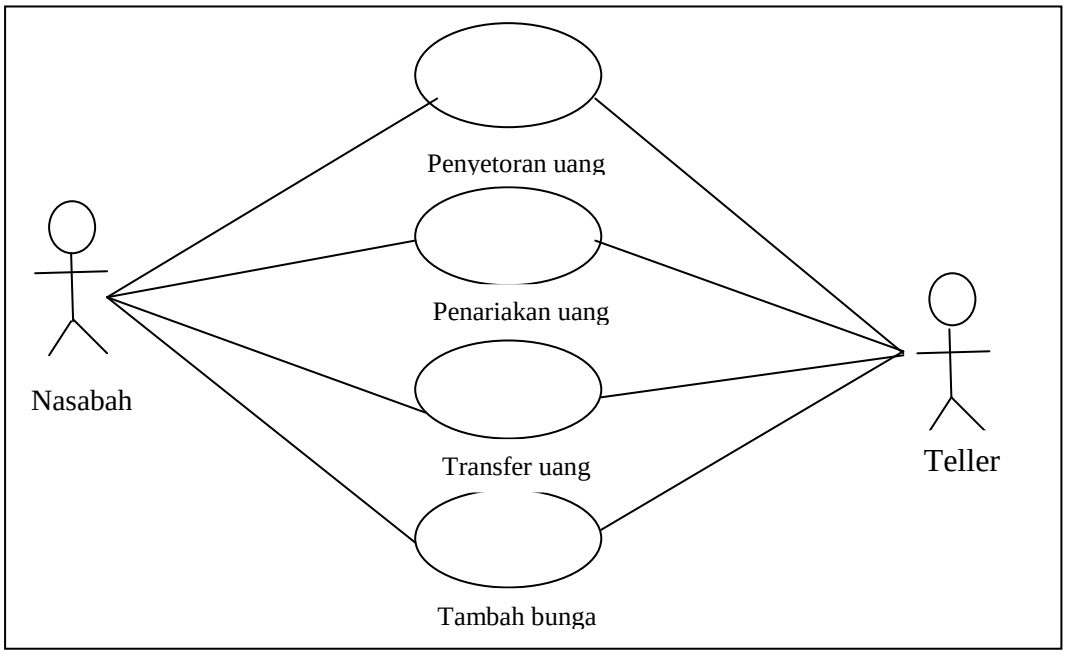
Bersifat statis diagram komponen ini meperlihatkanorganisasi serta ketergantungan sistem atau perangkat lunak pada komponen-komponen yang telah ada sebelumnya. Diagram ini berhubungan dengan diagram kelas dimana komponen secara tipikal di petakan ke dalam suatu atau lebih kelas-kelas, antarmuka-antarmuka serta kolaborasi-kolaborasi.

9. Diagram Deployment (*Deployment Diagram)*

Bersifat statis diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan (*run-time).* Memuat simpul-simpul berserta komponen-komponen yang ada di dalamnya. Diagram deployment berhubungan erat dengan diagram komponen dimana diagram ini memuat satu atau lebih komponen-komponen Diagram ini sangat berguna saat aplikasi dijalankan pada banyak mesin (*distributed computing).* (Herlawati ; 2010: 7).

II.9.2. Use Case

Use case menggambarkan external view dari sistem yang akan kita buat modelnya. Use case dapat di jabarkan dalam diagram use case, tetapi yang perlu diingat, diagram tidak identic dengan model karena model lebih luas dari diagram. Berikut contoh Gambar diagram use case :



Gambar II.2. Diagram Use case
(Sumber : Herlawati)

1. Aktor

Memeperlihatkan diagram use case dengan dua actor nasabaj dah teller dan empat use case. Simbol aktor adalah gambar orang. Sebelum membuat use case dan menentukan aktornya, agar mengidentifikasi siapa saja pihak yang terlibat dalam sistem kita. Pihak yang terlibat biasanya dinamakan *stakeholder*.

2. Use Case

Use case menggambarkan fungsi tertentu dalam suatu sistem berupa komponen, kejadian atau kelas, use case sangat menetukan karakteristik sistem yang kita buat, oleh karena itu disini dijelaskan cara untuk menghasilkan usecase yang baik yaitu :

a. Pilihlah nama yang baik

Use cse adalah sebuah behviour (perilaku), jadi seharusnya dalam frase kata kerja. Untuk membuat namanya ebih detil, tambahkan kata benda yang mengindikasi dampak aksinya terhadap suatu kelas objek. Oleh karena itu siagram use case seharusnya berhubungan dengan diagram kelas.

b. Ilustrasikan perilaku dengan lengkap

Use case dimulai dari inisiasi oleh aktor primer dan berakhir pada actor primer dan berakhir pada aktor dan menghasilkan tujuan. Jangan membuat use case kecuali anda mengetahui tujuan.

c. Identifikasi perilaku dengan lengkap

Untuk mencapai tujuan dan menghasilkan nilai tertentu dari aktor, use case harus lengkap. Ketika memberi naa pada use case, pilihlah frasa kata kerja yang implikasinya hingga selesai.

d. Menyediakan use case lawan (inverse)

Kita biasanya membutuhkan use case yang membatalkan tujuan, misalnya pada use case pemesanan kamar, dibutuhkan pula use case pembatalan pemesanan kamar.

e. Batasi use case hingga satu prilaku saja

Kadang kita cenderung membuat use case yang menghasilkan lebih dari satu tujuan aktifitas. Guna menghindari keracunan, jagalah use case kita hanya focus pada satu hal. (Herlawati ; 2010: 21).

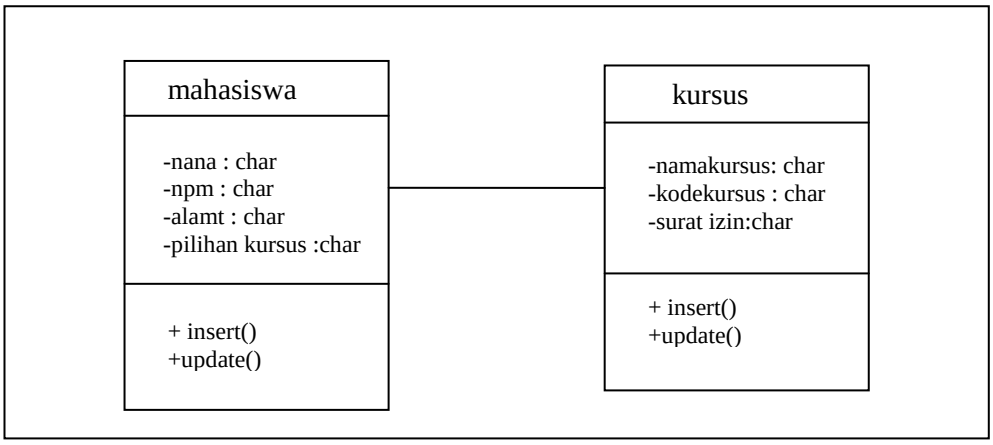
II.9.3. Diagram Kelas (*Class Diagram)*

Diagram adalah inti proses pemodelan objek. Baik *forward engineering* maupun *reverse engineering* memanfaatkan diagram ini. *Forward engineering* adalah proses perubahan model menjadi kode program sedangkan *reverse engineering* sebalinya kode program menjadi model. Tabel berikut ini penjelasan symbol relationships antar class yang digunakan pada diagram class

Tabel II.1 Symbol Class Diagram

Asosisi / association	Relasi antas kelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity
Asosiasi berarah / directed association	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan multiplicty
generalisasi	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
Kebergantungan/ dependency	Relasi antar kelas dengan makna kebergantungan ntar kelas
Agregasi/ aggregation	Relasi antar kelas dengan makna semua bagian (whole-part)
class	Himpunan daro objek-objek yang berbagi atribut serta operasi yang sama

(*Sumber : Herlawati ; 2010: 7)*



Gambar II.4 Clas Diagram

(*Sumber : Herlawati ; 2010: 7)*

II.9.4. Diagram Aktifitas (*Activity Diagram)*

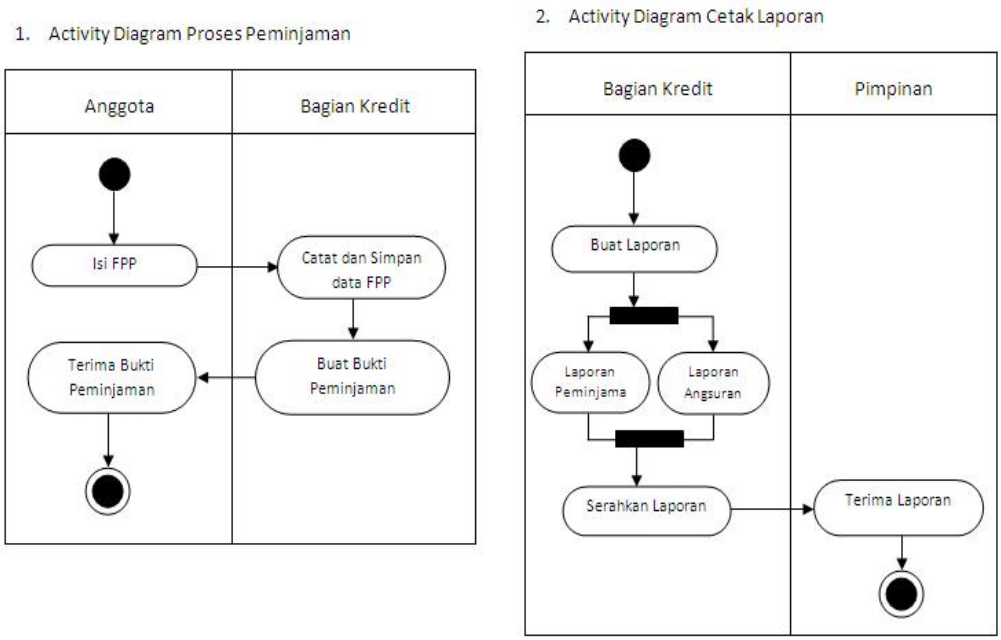
Pada aktifitas ini memfokuskan diri pada eksekusi dan alur sistem dari pada bagaimana sistem ini dirakit. Diagram ini memperlihatkan aliaran dari suatu aktifitas ke aktifitas lainnya dalam suatu sistem. Diagram ini terutama penting dalam pemodelan fungsi-fungsi dalam suatu sistem dan memberi tekanan pada aliran kendali antar objek.

Table II.2 Aktivity Diagram

No.	Gambar	Nama	Keterangan
1		*Actifity*	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
2		*Action*	State dari sistem yang mencerminkan eksekusi dari suatu aksi.
3		*Initial Node*	Bagaimana objek dibentuk atau diawali
4		*Actifity Final Node*	Bagaimana objek dibentuk dan dihancurkan.
5		*Fork Node*	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

(Sumber : Rosa A.S dan M.Shalahuddin 2011 : 13)

Berikut ini adalah contoh Activity Diagram proses peminjaman dan laporan :



Gambar II.5 Activity Diagram

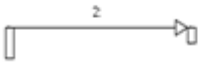
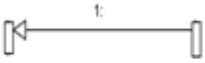
(Sumber : Rosa A.S dan M.Shalahuddin 2011 : 13)

II.9.5 Sequence Diagram

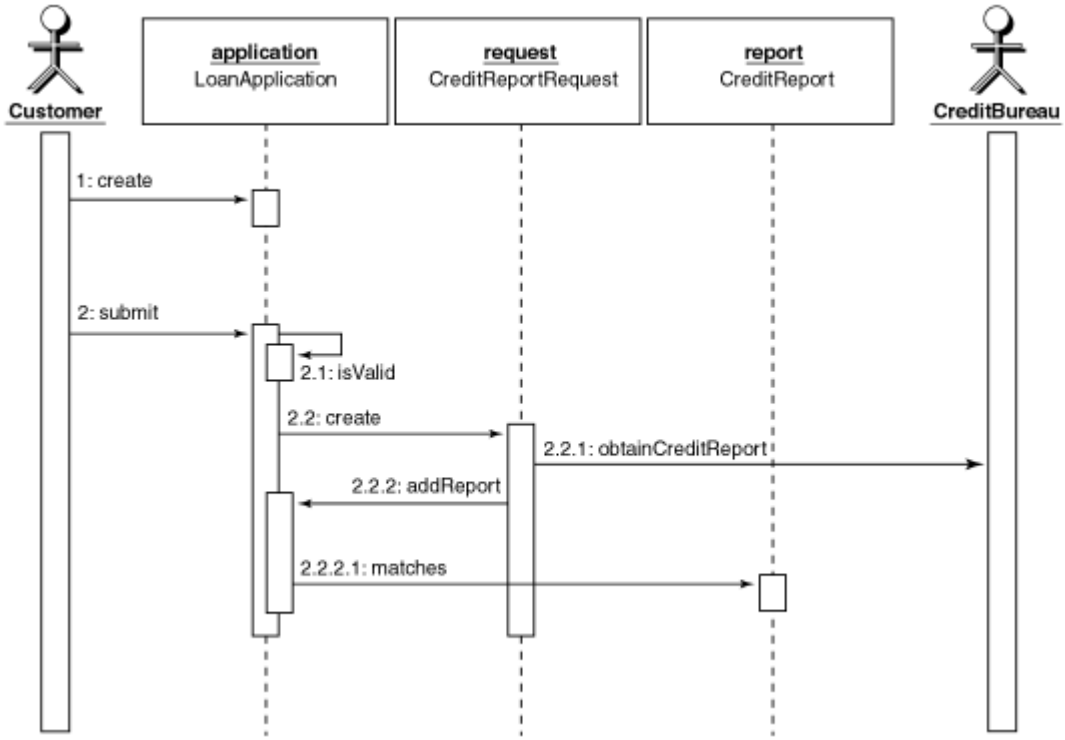
Sequence diagram adalah suatu diagram yang menggambarkan interaksi antar obyek dan mengindikasikan komunikasi diantara obyek-obyek tersebut. Diagram ini juga menunjukkan serangkaian pesan yang dipertukarkan oleh obyek-obyek yang melakukan suatu tugas atau aksi tertentu. Obyek-obyek tersebut kemudian diurutkan dari kiri ke kanan, aktor yang menginisiasi interaksi biasanya ditaruh di paling kiri dari diagram. Pada diagram ini, dimensi vertikal merepresentasikan waktu. Bagian paling atas dari diagram menjadi titik awal dan waktu berjalan ke bawah sampai dengan bagian dasar dari diagram. Garis Vertical, disebutlifeline, dilekatkan pada setiap obyek atau aktor. Kemudian

lifeline tersebut digambarkan menjadi kotak ketika obyek melakukan suatu operasi, kotak tersebut disebut activation. Obyek dikatakan mempunyai live activation pada saat tersebut. Pesan yang dipertukarkan antar obyek digambarkan sebagai sebuah anak panah antara activation box pengirim dan penerima. Kemudian diatasnya diberikan label pesan, berikut adalah symbol-simbol dari sequence diagram :

Tabel II.3 Sequence Diagram

No.	Gambar	Nama	Keterangan
1		*LifeLine*	objek *entity*, antarmuka yang saling berinteraksi.
2		*Message*	spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
3		*Message*	spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi

(Sumber : Rosa A.S dan M.Shalahuddin 2011 : 138)



Gambar II.6 Sequence Digram

(Sumber : Rosa A.S dan M.Shalahuddin 2011 : 138)

II.10. *Microsoft Visual Basic 2012*

Microsoft Visual Studio adalah sebuah *integrated development environment* buatan *Microsoft Coroporation*. Microsoft Visual Studio dapat digunakan untuk mengembangkan aplikasi dalam native code ataupun managed code. Selain itu, Visual studio juga dapat digunakan untuk mengembangkan aplikasi Silverlight, aplikasi Windows Mobile. Visual studio mencakup sebuah kode edito yang di dukung oleh fitur intellisense atau yang di sebut dengan code refactoring. Debugger telah terintegrasi bekerja pada level source level debugger mesin. Tool built in mencakup form desainer untuk membangun sebuah aplikasi GUI, web desainer, class desainer dan database schema desainaer. Microsoft

Visual studio mendukung bahasa pemograman yang berbeda adapun bahasa pemograman yang di dukung oleh Visual Studio adalah visual C++, Visual Basic, Visual C#, Visual studio juga dapat mendukung bahasa pemograman lain seperti .M, phyton dan ruby yang semuanya itu terdapat pada pack extra yang terpisah dari visual studio. (Wahana Komputer: 2013: 2)

Visual basic adalah bahasa pemograman yang menawarkan integrated Development Environment (IDE) visual untuk membuat program perangkat lunakberbasis sistem operasi Microsoft windows dengan menggunakan model pemograman (COM). Visual Basic merupakan peningkatan bahasa pemograman BASIC dan menawarkan pengembangan erangkat lunak compute berbasis grafik dengan cepat. Beberapa bahsa skrip seperti Visual Basic for Application(VBA) dan Visual Basic Scripting Edition (VBScript), mirip seperti halnya Visual Basic, tetapi cara kerjanya yang berbeda. Pemograman Visual basic pada Microsoft Visual Stodio 2012 juga menyediakan tingkat level fitur yang sama dengan bahasa pemograman lainny, hanya saja jenis bahasa pemogramannya saja yang berbeda. (Wahana Komputer; 2013: 4).

II.11 .SQL Server

SQL is now entering middle age (as is this author, alas), and it has undergone a great deal of change along the way. In the mid-1980s, the American National Standards Institute (ANSI) began working on the first standard for the SQL language, which was published in 1986. Subsequent refinements led to new releases of the SQL standard in 1989, 1992, 1999, 2003, and 2006. Along with

refinements to the core language, new features have been added to the SQL language to incorporate object-oriented functionality, among other things. The latest standard, SQL: 2006, focuses on the integration of SQL and XML and defines a language called XQuery which is used to query data in XML documents.

“ SQL sekarang memasuki usia paruh baya (seperti penulis ini, sayangnya), dan telah mengalami banyak mengubah sepanjang jalan. Pada pertengahan 1980-an, American National Standards Institute (ANSI) mulai bekerja pada standar pertama untuk bahasa SQL, yang diterbitkan pada tahun 1986. Perbaikan selanjutnya menyebabkan rilis baru dari standar SQL pada tahun 1989, 1992, 1999, 2003, dan 2006. Seiring dengan penyempurnaan bahasa inti, fitur baru telah ditambahkan ke SQL bahasa untuk menggabungkan fungsionalitas berorientasi objek, antara lain. Terbaru standar, SQL: 2006, berfokus pada integrasi SQL dan XML dan mendefinisikan bahasa yang disebut XQuery yang digunakan untuk query data dalam dokumen XML”. (Alan Beaulieu ; 2009: 18)

Microsoft SQL Server 2008 R2 is the most advanced, trusted, and scalable data platform released to date .Building on the success of the original SQL Server 2008 release, SQL Server 2008 R2 has made an impact on organizations worldwide with its groundbreaking capabilities, empowering end users through self-service business intelligence (BI), bolstering efficiency and collaboration between database administrators (DBAs) and application developers, and scaling to accommodate the most demanding data workloads .

“Microsoft SQL Server 2008 R2 adalah yang paling maju, terpercaya, dan scalable Data platform yang dirilis hingga saat ini. Membangun kesuksesan asli

SQL Server 2008 rilis, SQL Server 2008 R2 telah membuat dampak pada organisasi di seluruh dunia itu terobosan kemampuan, pemberdayaan pengguna akhir melalui pelayanan sendiri intelijen bisnis (BI), memperkuat efisiensi dan kolaborasi antara database administrator (DBA) dan pengembang aplikasi, dan skala untuk mengakomodasi beban kerja data yang paling menuntut “. (Microsoft Corporation ;2010: 3).