

BAB II

LANDASAN TEORI

2.1. Konsep Dasar Sistem

Secara garis besar sistem dibuat untuk mempermudah kegiatan ataupun pekerjaan manusia, sistem sendiri sebenarnya terdiri dari beberapa komponen yang saling berinteraksi dan bekerjasama dalam mencapai tujuan, contohnya seperti *money changer* yang mana para karyawannya bisa bekerjasama dengan baik dalam melayani konsumen. Sebagai gambaran jika didalam sebuah sistem terdapat elemen atau bagian yang tidak memberikan manfaat dalam mencapai tujuan bersama, maka elemen tersebut sudah dipastikan bukanlah bagian dari sistem.

2.1.1. Pengertian Sistem

Sistem digunakan organisasi dan perusahaan sebagai sekumpulan entitas yang memiliki hubungan dan menjamin kelancaran pengguna sistem tersebut. Sistem memudahkan aliran informasi, materi dan keluaran sistem untuk mencapai tujuan yang ingin dicapai. Menurut (Hutahean, 2014) “Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu”.

2.1.2. Karakteristik Sistem

Sistem yang baik memiliki karakteristik atau ciri-ciri yang mengidentifikasikan bahwa hal tersebut merupakan sistem. Karakteristik sistem menurut (Hutahean, 2014) dapat diuraikan sebagai berikut:

1. Komponen

Sistem terdiri dari sejumlah komponen-komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk satu kesatuan. Komponen sistem terdiri dari komponen yang berupa subsistem atau bagian-bagian dari sistem.

2. Batasan sistem (*Boundary*)

Batasan sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lain atau dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan. Batasan suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

3. Lingkungan Luar Sistem (*Environment*)

Lingkungan luar sistem (*environment*) adalah diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan dapat bersifat menguntungkan yang harus tetap dijaga dan yang merugikan yang harus dijaga dan dikendalikan, kalau tidak akan mengganggu kelangsungan hidup dari sistem.

4. Penghubung Sistem (*Interface*)

Penghubung sistem merupakan media penghubung antara satu subsistem dengan subsistem lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari subsistem ke subsistem lain. Keluaran (*output*) dari subsistem akan menjadi masukan (*input*) untuk subsistem lain melalui penghubung.

5. Masukan Sistem (*Input*)

Masukkan adalah energi yang dimasukkan kedalam sistem, yang dapat berupa perawatan (*maintenance input*), dan masukan sinyal (*signal input*). *Maintenance input* adalah energi yang dimasukkan agar sistem dapat beroperasi. *Signal input* adalah energi yang diproses untuk didapatkan keluaran. Contoh dalam sistem

komputer program adalah *maintenance input* sedangkan data adalah *signal input* untuk diolah menjadi informasi.

6. Keluaran Sistem (*Output*)

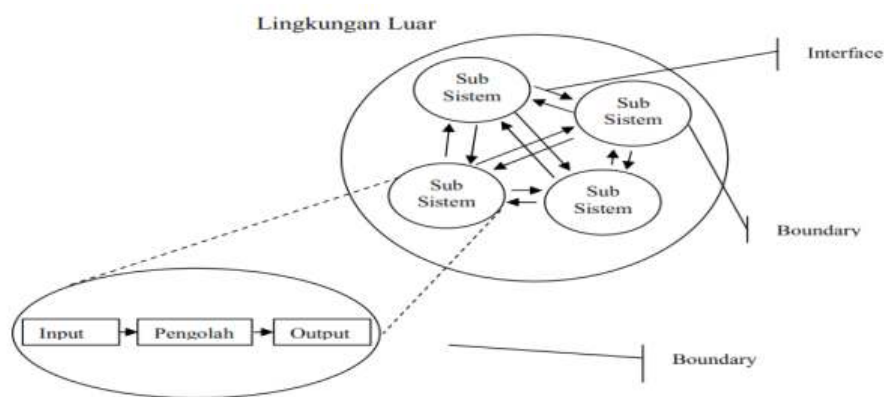
Keluaran sistem adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Contoh komputer menghasilkan panas yang merupakan sisa pembuangan, sedangkan informasi adalah keluaran yang dibutuhkan.

7. Pengolah Sistem (*Process*)

Suatu sistem menjadi bagian pengolah yang akan merubah masukan menjadi keluaran. Sistem produksi akan mengolah bahan baku menjadi bahan jadi, sistem akuntansi akan mengolah data menjadi laporan-laporan keuangan.

8. Sasaran Sistem (*Objectives*)

Suatu sistem pasti mempunyai tujuan (*goal*) atau sasaran (*objective*). Sasaran dari sistem sangat menentukan input yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem



Sumber: (Hutahean, 2014)

Gambar II. 1
Karakteristik Suatu Sistem

2.1.3. Klasifikasi Sistem

Sistem merupakan kumpulan komponen-komponen yang memiliki tujuan tertentu menurut (Hutahean, 2014). Sistem juga dapat diklasifikasikan dalam beberapa sudut pandang:

1. Sistem Abstrak (*abstract system*)

Sistem yang berupa pemikiran atau ide-ide yang tampak secara fisik. Misalnya sistem teologia, yaitu sistem pemikiran-pemikiran hubungan antara manusia dengan tuhan.

2. Sistem Fisik (*physical system*)

Sistem yang ada atau nampak secara fisik. Misalnya sistem komputer, sistem akuntansi sistem produksi dan lain sebagainya.

3. Sistem Alamiah (*natural system*)

Sistem yang terjadi melalui proses alam, tidak dibuat manusia. Misalnya sistem perputaran bumi.

4. Sistem Buatan Manusia (*human made system*)

Sistem yang dirancang dan dibuat oleh manusia. Sistem buatan manusia yang melibatkan manusia dengan mesin disebut dengan *human-machine* atau *man-machine system*.

5. Sistem tertentu (*deterministic system*)

Sistem yang beroperasi dengan tingkah laku yang sudah dapat diprediksi, sebagai keluaran sistem yang dapat diramalkan.

6. Sistem tak tentu (*probabilistic system*)

Sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur *probabilistik*.

7. Sistem tertutup (*closed system*)

Sistem yang tidak berhubungan dan terpengaruh dengan lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa adanya turut campur tangan dari pihak luarnya. Secara teoritis sistem tertutup ini ada, tetapi kenyataannya tidak ada sistem yang benar-benar tertutup yang ada hanyalah sistem relative tertutup (*relatively closed system*).

8. Sistem terbuka (*open system*)

Sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya. Sistem ini menerima input dan output dari lingkungan luar atau subsistem lainnya. Karena sistem terbuka terpengaruh lingkungan luar maka harus mempunyai pengendali yang baik.

2.1.4. Sistem Informasi

Informasi sangat penting didalam suatu organisasi tetapi sistem yang kurang mendapatkan informasi akan menjadi kurang efektif dan tidak tepat. Namun Sistem Informasi yang berada pada organisasi yang didalamnya terdapat sekelompok orang-orang, teknologi, media, fasilitas, prosedur-prosedur dan pengendalian data yang digunakan untuk tujuan mendapatkan jalur komunikasi, memproses data secara rutin, memberikan signal kepada manajemen mengenai kajadian-kejadian internal dan eksternal dan menyediakan informasi yang dapat digunakan sebagai pengambilan keputusan.

Seiring perkembangan zaman dan kemajuan teknologi organisasi atau perusahaan sekarang perlu pembaruan sistem untuk bersaing, penggabungan antara sistem dengan teknologi dinamakan dengan sistem informasi, menurut (Hutahean, 2014) “Sistem informasi adalah suatu sistem didalam suatu organisasi yang

mempertemukan kebutuhan pengelolaan transaksi harian, mendukung operasi, bersifat manajerial, dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang dibutuhkan”. sedangkan menurut (Adiguna, Saputra Chandra, & Pradana, 2018) “sistem informasi merupakan suatu sistem di dalam organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial”.

Kesimpulan sistem informasi dari beberapa ahli diatas adalah serangkaian kegiatan yang melibatkan serangkaian proses dan mempertemukan kebutuhan suatu organisasi demi mencapai sebuah tujuan.

2.1.5 Rancangan Dasar Program

Perangkat lunak (*software*) adalah program komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan, model desain, dan cara penggunaan (*user manual*). “Sebuah program komputer tanpa terasosiasi dengan dokumentasinya maka belum dapat disebut perangkat lunak (*software*). Sebuah perangkat lunak juga sering disebut dengan sistem perangkat lunak” (Rosa, 2018).

Program komputer dapat digolongkan menjadi tiga jenis yaitu:

1. Sistem operasi yaitu berisi perintah dasar untuk mengedalikan dan mengkoordinasi kegiatan sistem komputer yaitu mulai dari komputer dihidupkan sampai dimatikan.
2. Bahasa komputer yaitu Bahasa yang digunakan untuk membuat program komputer dimana Bahasa tersebut digunakan untuk mnerjemahkan intruksi-intrusi yang ditulis dalam bahasa pemrograman ke dalam bahasa mesin supaya dapat dimengerti oleh komputer.

3. Program aplikasi adalah program yang siap digunakan untuk mengolah data menjadi informasi. Program aplikasi ini sering disebut dengan program komputer yang sudah siap pakai.
4. Program *utility* adalah untuk keperluan khusus seperti diagnosa media penyimpanan, perawatan program, pengformatan, program anti virus, program analisa jaringan, dan program *tool* lainnya.

2.1.6 Basis Data (*Database*)

Basis data sering disebut juga pangkalan data yang merupakan tempat berkumpulnya data-data yang saling berhubungan dan terintegrasi mengenai operasional perusahaan yang disimpan dan dikelola dengan cara yang sistematis dengan tujuan agar dapat disimpan, dimodifikasi, dicari dan digunakan kembali dengan cepat. Menurut (Rosa, 2018) “Sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan. Pada intinya basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat”.

Ada beberapa perangkat lunak atau *software open source* yang disediakan untuk membuat suatu basis data. Perangkat lunak tersebut merupakan suatu pemrograman yang dikategorikan sebagai bahasa pemrograman tingkat tinggi (*high level language*) salah satunya adalah *mySQL*. Data base *MySQL* dapat dibuat menggunakan tampilan *phpmy admin* atau menggunakan sebuah *script* dalam *PHP*. Menurut (Gusrion, Kom, & Kom, 2018) “*MySQL* merupakan sebuah perangkat lunak (*software*) sistem manajemen basis data *SQL* atau *DBMS Multithread* dan *multi user*. *MySQL* sebenarnya merupakan turunan dari salah satu konsep utama

dalam *database* untuk pemilihan atau seleksi dan pemasukan data yang memungkinkan pengoperasian data dikerjakan secara mudah dan otomatis”. Adapun kelebihan MySQL dalam penggunaannya dalam *database* adalah:

1. Gratis sehingga MySQL dapat dengan mudah untuk mendapatkannya
2. MySQL stabil dalam pengoperasiannya
3. MySQL mempunyai sistem keamanan yang cukup baik
4. Sangat mendukung transaksi dan mempunyai banyak dukungan dari komunitas
5. Sangat *fleksibel* dengan berbagai macam program
6. Perkembangan dari MySQL sangat cepat

2.1.7 Model Pengembangan Perangkat Lunak

Proses pengembangan sistem seringkali menggunakan pemodelan prototipe (*prototyping*). Model ini sangat baik digunakan untuk menyelesaikan masalah kesalah pahaman antara *user* dan analis yang timbul akibat *user* tidak mampu mendefinisikan secara jelas kebutuhannya.

Prototyping adalah pengembangan yang cepat dan pengujian terhadap model kerja (*prototyping*) dari aplikasi baru melalui proses interaksi dan berulang-ulang yang bias digunakan ahli sistem informasi dan ahli bisnis menurut (Rosa, 2018). *Prototyping* disebut juga desain aplikasi cepat karena menyederhanakan dan mempercepat desain sistem.

Berikut adalah tahapan-tahapan dalam proses prototipe secara umum:

1. Pengumpulan Kebutuhan

Developer dan klain bertemu lebih dahulu dan menentukan tujuan umum, kebutuhan yang diinginkan dan bagian-bagian yang dibutuhkan nantinya.

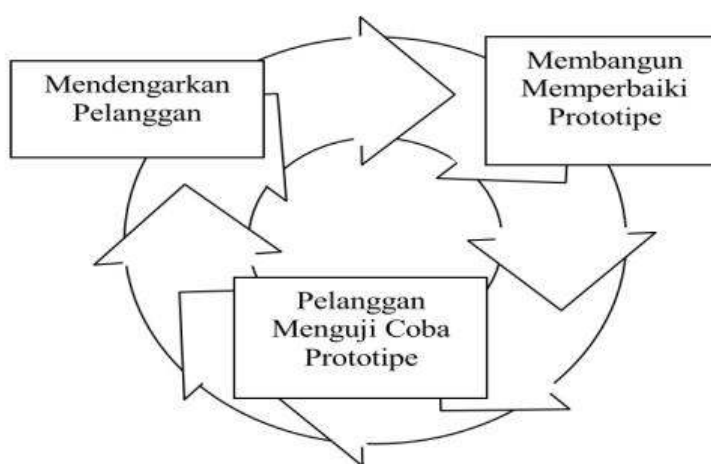
2. Perancangan

Perancangan dilakukan dengan cepat dan rancangan tersebut mewakili aspek *software* yang diketahui, dan rancangan ini menjadi dasar pembuatan prototipe.

3. Evaluasi prototipe

Klien akan mengevaluasi *prototype* yang dibuat dan menjelaskan kebutuhan *software*.

Berikut ini adalah gambar dari model prototipe:



Sumber: Rosa (2018)

Gambar II.2
Contoh *Model Prototyping*

2.2. Teori Pendukung (*Tools System*)

Dalam mendesain atau merancang model sistem informasi yang diusulkan dalam *logical model*, media ini merupakan alat yang tepat digunakan untuk menggambarkan model dari sistem dimana simbol-simbol, lambang-lambang dan diagram-diagram menunjukkan secara tepat arti fisiknya. Untuk memahami isi dari tugas akhir ini, maka diperlukan beberapa pendukung mengenai definisi serta uraian yang berkaitan dengan teori pendukung, sebagai berikut:

2.2.1. *Entity Relationship Diagram (ERD)*

Definisi diagram alir data menurut (Rosa, 2018) adalah “Pemodelan awal basis data yang paling banyak digunakan, dikembangkan berdasarkan teori himpunan dalam bidang matematika. ERD digunakan untuk pemodelan basis data relasional. Jika menggunakan OODBMS maka perancang ERD tidak perlu digunakan”.

Simbol-simbol yang sering digunakan dalam *Entity Relationship Diagram* sebagai berikut:

1. Entitas (*entity*)

Merupakan data inti yang akan disimpan, bakal tabel pada basis data, benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi komputer, penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel.

2. Atribut

Field atau kolom data yang butuh disimpan dalam suatu entitas.

3. Atribut kunci primer

Field atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses *record* yang diinginkan, biasanya berupa id, kunci *primer* dapat lebih dari satu kolom, asalkan kombinasi dari beberapa kolom tersebut dapat bersifat unik (berbeda tanpa ada yang sama).

4. Atribut multinilai / *multivalue*

Field atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu.

5. Relasi

Relasi yang menghubungkan antara entitas, biasanya diawali dengan kata kerja.

6. Asosiasi / *association*

Penghubung antara relasi dan entitas di mana di kedua ujungnya memiliki *multiplicity* kemungkinan jumlah pemakaian.

2.2.2. *Logical Record Structure (LRS)*

Menurut Tabrani (2014:33-41) “*Logical Record Structure (LRS)* dibentuk dengan nomor dari tipe record”. Beberapa tipe record digambarkan oleh kotak persegi panjang dan dengan nama yang unik. Perbedaan LRS dengan E-R diagram adalah nama tipe record berada diluar kotak *field* tipe record ditempatkan. LRS terdiri dari *link-link* diantara tipe record. Link ini menunjukkan arah dari satu tipe record *field-field* yang kelihatan pada kedua *link* tipe record. Penggambaran LRS mulai dengan menggunakan model yang dimengerti. Dua metode yang dapat digunakan, dimulai dengan hubungan kedua model yang dapat dikonversikan ke LRS, metode yang lain dimulai dengan ERDiagram dan langsung dikonversikan ke LRS. Berikut adalah kardinalitas yang terdapat pada *Logical Record Structure (LRS)*:

a. *One to One*

Satu entitas hanya berhubungan dengan satu entitas lain.

b. *One to Many*

Satu entitas dapat berhubungan dengan lebih dari satu entitas lain.

c. *Many to Many*

Beberapa entitas dapat berhubungan dengan beberapa entitas lain.

2.2.3. *Unified modeling language (UML)*

Sejarah UML ada beberapa usaha untuk menyatukan metode-metode berorientasi objek yang bermunculan. Contohnya *Fusion* oleh *Coleman* dan kawan-kawan (*Coleman-94*), yang memasukan konsep dari OMT (*Object Modeling Technique*) (*Rumbaugh-91*), *Booch* (*Booch-91*), dan CRC (*Wirfs-Brock-90*). Karena hal ini tidak melibatkan penulis asli, maka dianggap sebagai metode baru dan bukan menggantikan berbagai metode-metode yang sudah ada. Usaha penggabungan yang sukses pertamakali dan mengganti metode yang ada adalah ketika *Rumbaugh* bergabung dengan *Booch* pada perusahaan *Rational Software* tahun 1994. Mereka mulai mengkombinasikan konsep OMT dan metode *Booch*, yang menghasilkan proposal pertama tahun 1995. Pada waktu itu, *Jacobson* juga bergabung di *Rational* dan mulai bekerjasama dengan *Booch* dan *Rumbaugh*. Kerjasama mereka disebut *Unified Modeling Language (UML)*. Mereka merevisi metode masing-masing untuk menghasilkan satu metode lengkap yang harmonis. Pada tahun 1996 *Object Management Group* mengeluarkan permintaan untuk proposal-proposal untuk pendekatan standar pemodelan berorientasi objek. *Unified Modeling Language* diadopsi oleh anggota OMG sebagai standar pada November 1997. OMG bertanggung jawab untuk pengembangan lebih lanjut dari standar UML menurut (Akil, 2016).

Menurut (Rosa, 2018) “UML (*Unifeid modeling language*) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorintasi objek”.

Menurut (Hendini, 2016) “UML (*Unifeid modeling language*) adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan

dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem”. Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut:

1. *Use Case Diagram*

Use case diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut.

Simbol-simbol yang digunakan dalam *Use Case Diagram* yaitu:

2. Diagram Aktivitas (*Activity Diagram*)

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity Diagram* yaitu:

3. Diagram Urutan (*Sequence Diagram*)

Sequence Diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *Sequence Diagram* yaitu:

4. Diagram Kelas (*Class Diagram*)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. *Class Diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan.

Class Diagram secara khas meliputi: Kelas (*Class*), Relasi *Assosiations*, *Generalitaton* dan *Aggregation*, atribut (*Attributes*), operasi (*operation/method*) dan *visibility*, tingkat akses objek *eksternal* kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *Multiplicity* atau *Cardinality*.