BAB II

LANDASAN TEORI

2.1 Konsep Dasar Sistem

2.1.1 Sistem

Secara sederhana sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen atau variabel-variabel yang terorganisir, saling berinteraksi, saling ketergantungan satu sama lain dan terpadu. Suatu sistem pada dasarnya adalah kelompok unsur yang erat hubungannya satu sama lainnya, yang berfungsi bersama-sama untuk mencapai tujuan tertentu.

Pengertian sistem menurut beberapa ahli yaitu, menurut (Ruhul Amin, 2017) mengemukakan bahwa "Sistem didefinisikan sebagai suatu kumpulan atau himpunan dari unsur komponen atau variabel yang terorganisir saling berinteraksi, saling tergantung satu sama lain, dan terpadu".

Sedangkan menurut (Herliana & Rasyid, 2016) sebuah sistem adalah "sekumpulan entitas (*hardware*, *brainware*, *software*) yang saling berinteraksi, bekerjasama dan berkolaborasi untuk mencapai tujuan tertentu".

Pengembangan industri tersebut dapat menimbulkan kapasitas produksi meningkat sehingga meningkatkan penyerapan tenaga kerja (Ariska, 2018).

2.1.2 Karakteristik Sistem

Suatu sistem memiliki karakteristik tertentu, yang merinci bahwa hal tersebut bisa dikatakan sebagai suatu sistem. Adapun yang termasuk ke dalam karakteristik sistem adalah sebagai berikut :

1. Komponen Sistem (*Component*)

   Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk satu kesatuan. Komponen sistem dapat berupa suatu sub sistem atau bagian-bagian dari sistem. Setiap sub sistem mempunyai sifat-sifat dari sistem untuk menjalankan suatu fungsi dan mempunyai proses secara keseluruhan.

2. Batasan Sistem (*Boundary*)

   Batasan Sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luar. Batasan sistem tersebut memungkinkan suatu sistem dipandang sebagai satu kesatuan. Dan batas suatu sistem juga menunjukan ruang lingkup (*scope*) dari sistem tersebut.

3. Lingkungan Luar Sistem (*Environment*)

   Lingkungan luar dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem.

4. Penghubung Sistem (*Interface*)

   Penghubung sistem merupakan media penghubung antara suatu sistem dengan sub sistem lainnya. Keluaran (*output*) dari sub sistem akan menjadi masukan (*input*) untuk sub sistem yang lainnya melalui penghubung.

5. Masukan Sistem (*Input*)

   Masukan adalah energi yang dimasukan ke dalam sistem. Masukan tersebut dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*). *Maintenance input* adalah energi yang digunakan agar sistem tersebut dapat beroperasi. Sedangkan *signal input* adalah energi yang diproses untuk mendapatkan keluaran.

6. Keluaran Sistem (*Output*)

   Keluaran adalah hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembangunan. Keluaran juga dapat dijadikan masukan untuk sub sistem yang lainnya.

7. Pengolah Sistem (*Process*)

   Pengolah sistem dapat mempunyai suatu bagian pengolah yang akan merubah masukan menjadi keluaran. Suatu sub sistem proses pengolahan data barang akan menjadi laporan-laporan berupa tagihan dan laporan yang dibutuhkan oleh manajemen.

8. Sasaran Sistem (*Objective*)

   Suatu sistem pasti mempunyai tujuan (*goal*) atau sasaran (*Objective*). *Goal* biasanya dihubungkan dengan ruang lingkup yang lebih sempit. Sasaran dari sistem sangat menentukan sekali masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem. Suatu sistem dikatakan berhasil jika mengenai sasaran atau tujuan yang diharapkan.

2.1.3 Klasifikasi Sistem

Sistem dapat diklasifikasikan dari beberapa sudut pandang yang diantaranya:

1. Sistem Abstrak Dan Sistem Fisik
   a. Sistem Abstark (*Abstract System*) adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik (sistem teoligia yang merupakan suatu sistem yang menggambarkam hubungan antara tuhan dengan manusia).
   b. Sistem Fisik (*Physical System*) adalah sistem yang tampak secara fisik sehingga setiap mahluk dapat melihatnya misalnya seperti sistem komputer, sistem akutansi, sistem produksi, dan lain-lain.
2. Sistem Alamiah Dan Sistem Buatan Manusia
   a. Sistem Alamiah (*Natural Sistem*) adalah sistem yang terjadi melalui proses alam, tidak dibuat oleh manusia misalnya seperti tata surya, sistem galaxy, sistem produksi dan lain-lain.
   b. Sistem Buatan Manusia (*Human Made System*) adalah sistem yang dirancang oleh manusia. Sistem buatan yang melibatkan interaksi manusia misalnya seperti akuntansi, sistem informasi dan lain-lain.
3. Sistem Deterministik dan Sistem Probabilistik
   a. Sistem Deterministik (*Deterministic System*) adalah sistem yang beroperasi dengan tingkah laku yang sudah diprediksi. Interaksi bagian-bagiannya dapat dideteksi dengan pasti sehingga keluaran dari sistem dapat diramalkan

misalnya sistem komputer adalah contoh sistem yang tingkah lakunya dapat dipastikan berdasarkan program-programnya komputere yang dijalankan.

b. Sistem Probabilistik (*Probabilistic System*) adalah sistem yang kondisi masa depan tidak dapat diprediksi karena mengundang unsur probabilitas misalnya sistem manusia.

4. Sistem Terbuka dan Sistem Tertutup

a. Sistem Terbuka (*Open System*) adalah sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya. Lebih spesifik dikenal dengan juga yang disebut dengan sistem terotomasi, yang merupakan bagian dari sistem yang digunakan dari masyarakat modern. Sistem ini menerima masukan dan menghasilkan keluaran untuk sub sistem lainnya misalnya sistem kebudayaan manusia.

b. Sistem Tertutup (*Close System*) adalah sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa adanya campur tangan dari pihak luar. Secara teroritis sistem tersebut ada, tetapi kenyataannya tidak ada sistem yang benar benar tertutup yang ada hanyalah *relatively closed system* (secara relative tertutup tidak benar benar tertutup).

2.1.4 Pengertian Permintaan Tenaga Kerja

Menurut (Ahmad, 2019) Tenaga kerja adalah aspek dari dunia bisnis yang menentukan bagi pelaku bisnis dalam mengeksekusi permintaan dari pangsa pasar.

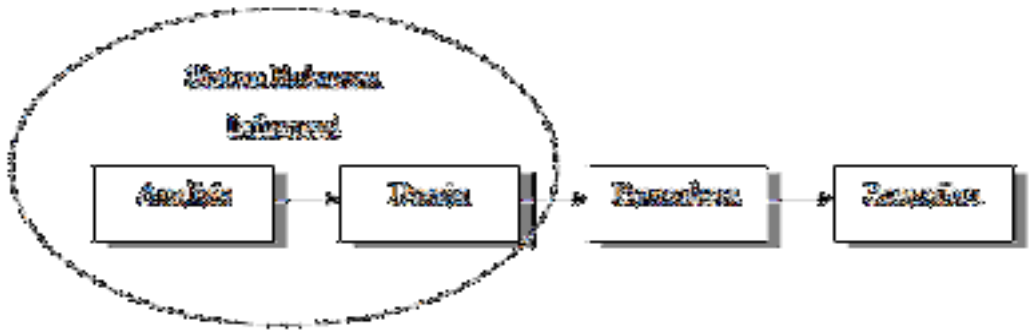
Hak bagi seluruh warga Indonesia dalam memiliki pekerjaan yang layak diatur dalam Undang-Undang Dasar Negara Republik Indonesia 1945.

Menurut (Ahmad, 2019) Sistem *Outsourcing* (Alih Daya) adalah solusi yang tepat dalam masalah yang dihadapi oleh pelaku usaha, penyediaan pekerjaan untuk perusahaan lain melalui perjanjian untuk menyediakan jasa pekerja/buruh dan perjanjian untuk Sebagian kontrak kerja. Praktik di lapangan yang menyeleweng dari UU Ketenagakerjaan menyebabkan banyak pihak tidak setuju dengan adanya sistem alih daya di Indonesia.

2.1.5 Metode Pengembangan Perangkat Lunak

Dalam rekayasa perangkat lunak, metodologi pengembangan perangkat lunak atau metodologi pengembangan sistem adalah suatu kerangka kerja yang digunakan untuk menstrukturkan, merencanakan, dan mengendalikan proses pengembangan suatu sistem informasi.

Menurut Rosa (2016:28) "Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengkodean, pengujian, dan tahapan pendukung (*support*)". Berikut adalah gambar model air terjun:



Sumber : Rosa dan Shalahuddin (2016:29)

Gambar II.1. *Waterfall* Model

Pada metode yang digunakan dalam pengembangan perangkat lunak ini menggunakan metode *waterfall* menurut Rosa dan Shalahuddin (2016:29) terdapat lima tahapan yaitu:

a. Analisis Kebutuhan Perangkat Lunak

   Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu didokumentasikan.

b. Desain

   Desain perangkat lunak adalah proses multi langkah yang focus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

c. Pembuatan kode program

Desain harus ditranslasikan kedalam program perangkat lunak. Hasil dari tahap ini adalah program komputer yang sesuai dengan desain yang telah dibuat pada tahap desain.

d. Pengujian

Pengujian fokus pada perangkat lunak secara dari segi logik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

e. Pendukung (S*upport*) dan Pemeliharaan (*Maintenance*)

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke *user*. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.

2.1.6 Basis Data

Basis data atau yang dikenal dengan *Database* digunakan sebagai suatu tempat untuk menampung informasi yang didalam komputer secara sistematik sehingga dapat diperiksa menggunakan program komputer agar dapat memperoleh informasi dari basis data tersebut.

Menurut Rosa (2016:43) “Sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan”.

Sedangkan menurut Herliana (2016:46) “*Database* merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan diperangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya”.

Berdasarkan penjelasan di atas penulis dapat menarik kesimpulan bahwa Basis Data adalah suatu tempat untuk memelihara dan menyimpan informasi beserta data agar lebih mudah digunakan dan ditampilkan kembali oleh pemiliknya.

2.2 Teori Pendukung

2.2.1 ERD (*Entity Relationship Diagram)*

ERD adalah suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar yang mempunyai hubungan antar relasi. ERD untuk memodelkan struktur data dan hubungan antar data, untuk menggambarkan digunakan beberapa notasi dan simbol.

Menurut Marlinda dalam Ruhul (2017:115) mengatakan bahwa “*Model Entity Relationship* merupakan suatu model untuk menjelaskan hubungan antara data dalam basis data berdasarkan suatu persepsi bahwa *real world* terdiri dari *object-object* dasar yang mempunyai hubungan atau relasi antar *object-object* tersebut”.

Sedangkan menurut Indrajani (2017:25) “*Entity Relational* (ER) *Modeling* adalah sebuah pendekatan *top-bottom* dalam perancangan basis data yang dimulai

dengan mengidentifikasikan data-data terpenting yang disebut dengan entitas dan hubungan antara entitas-entitas tersebut yang digambarkan dalam suatu model".

Berdasarkan penjelasan diatas dapat disimpulkan bahwa ERD adalah suatu pemodelan basis data yang menghubungkan atau merelasikan antar entitas.

Berikut ini beberapa komponen ERD yaitu:

1. Entitas

   Suatu objek dengan karakteristik sama yang dilengkapi oleh atribut.

2. Atribut

   Atribut merupakan *field* atau kolom data yang disimpan dalam suatu entitas

3. Relasi

   Relasi merupakan penghubung antar entitas dan biasanya diawali dengan kata kerja.

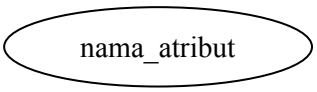
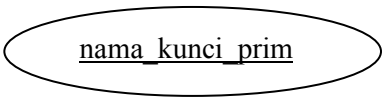
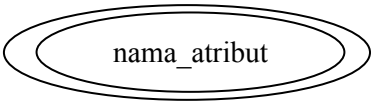
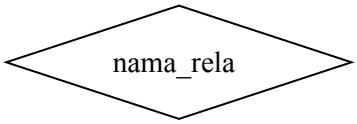
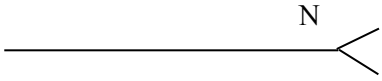
4. Garis Penghubung

   Sesuai dengan namanya garis tersebut sebagai penghubung antara himpunan relasi dengan himpunan entitas dan himpunan entitas dengan atributnya.

Tabel II.1.

Simbol-Simbol ERD (*Entity Relationship Diagram*)

Simbol	**Deskripsi**
Entitas / *entity* nama_entitas	Entitas 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

Atribut 	*Field* atau kolom data yang butuh disimpan dalam suatu entitas
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)
Atribut multinilai / *multivalue* 	*Field* atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu
Relasi 	Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja
Asosiasi / *association* 	Penghubung antara relasi dan entitas di mana di kedua ujungnya memiliki *multiplicity* kemungkinana jumlah pemakaian Kemungkinan jumlah maksismum keterhubungan antara entitas satu dengan entitas lain disebut kardinalitas. Misalkan ada kardinalitas 1 ke N atau sering disebut dengan *one to many* menghubungkan entitas A dan entitas B

Sumber: Rosa (2016:50)

2.2.2 LRS (*Logical Record Structure*)

LRS (*Logical Record Structure*) adalah representasi dari struktur *record-record* pada tabel-tabel yang terbentuk dari hasil antar himpunan entitas.

Menurut Frieyadie dalam Ruhul (2017:115) mengatakan bahwa "LRS merupakan hasil dari pemodelan *Entity Relationship* (ER) beserta atributnya sehingga bisa terlihat hubungan-hubungan antar entitas".

Terdapat tiga hal dapat mempengaruhi dalam pembuatan *Logical Record Structure* (LRS) menurut Taufik (2017:3), yaitu:

1. *One to One* (Satu ke Satu)

Jika tingkat hubungan (*cardinality*) satu pada satu (*one to one*), maka digabungkn dengan entitas yang lebih kuat (*strong entity*), atau digabungkan dengan entitas yang memiliki atribut yang lebih sedikit.

2. *One to Many* (Satu ke Banyak)

Jika tingkat hubungan (*cardinality)* satu pada banyak (*one to many*), maka hubungan relasi atau digabungkan dengan entitas yang tingkat hubungannya banyak.

3. *Many to Many* (Banyak ke Banyak)

Jika tingkat hubungan (*cardinality*) banyak pada banyak (*many to many*), maka hubungan relasi tidak akan digabungkan dengan entitas manapun, melainkan menjadi sebuah LRS.

Dari penjelasan diatas dapat disimpulkan bahwa LRS (*Logical Record Structure*) adalah representasi dari struktur *record-record* pada tabel-tabel yang terbentuk dari hasil antar himpunan entitas.

2.2.3 UML (*Unified Modeling Language*)

Unified Modeling Language adalah metode pemodelan (*tools/*model) secara visual sebagai sarana untuk merancang dan atau membuat software berorientasi objek dan memberikan standar penulisan sebuah sistem untuk pengembangan sebuah

software berorientasi objek dan memberikan standar penulisan sebuah sistem untuk pengembangan sebuah *software* yang dapat menyampaikan beberapa informasi untuk proses implementasi pengembangan *software*.

Menurut Rosa (2016:137) UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan mendokumentasikan dari sistem perangkat lunak. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan diagram teks-teks pendukung.

Untuk dapat memahami UML diperlukan pemahaman tentang konsep bahasa pemodelan dan tiga elemen utama UML yaitu antara lain:

1. Benda atau Objek (*Things*)

   Objek merupakan bagian paling statis dari sebuah model, yang menjelaskan elemen-elemen lainnya dari sebuah konsep. Bentuk dari objek tersebut adalah:

   a. *Classes*, sekelompok dari objek yang mempunyai *attribute*, operasi, dan hubungan yang semantic
   b. *Interfaces,* antar-muka yang menghubungkan dan melayani antarkelas dan atau elemen dan mendefinisikan sebuah kelompok dari spesifikasi pengoperasian.
   c. *Collaboration*, interaksi dari sebuah kumpulan kelas-kelas atau elemen-elemen yang bekerja secara bersamaan.
   d. *Use cases*, pembentukan tingkah laku objek dalam sebuah model serta di realisasikan oleh *collaboration.*

e. *Nodes,* bentuk fisik dari elemen-elemen yang ada pada saat dijalankan sebuah sistem.

2. Hubungan (*Relationship*)

Ada 4 (empat) macam hubungan dalam penggunaan UML, yaitu:

a. *Dependency*

Hubungan semantik antara dua objek yang mana sebuah objek berubah mengakibatkan objek satunya akan berubah pula.

b. *Association*

Hubungan antar benda secara struktural yang terhubung diantara objek dalam kesatuan objek.

c. *Generalizations*

Hubungan khusus dalam objek anak yang menggantikan objek induk dan memberikan pengaruhnya dalam hal struktur dan tingkah lakunya kepada objek induk.

d. *Realizations*,

Hubungan semantik antar pengelompokkan yang menjamin adanya ikatan diantaranya yang diwujudkan diantara *interface* dan kelas atau *elements,* serta antara *use cases* dan *collaborations*.

3. Bagan (*Diagrams*)

Diagrams adalah yang menggambarkan permasalahan maupun solusi dari permasalahan suatu model. Dalam hal ini berdasarkan sifatnya UML menyediakan 4 (empat) jenis diagram diantaranya yaitu :

a. *Use case diagram*

Suatu kumpulan urutan interaksi diantara *user* dengan sistem untuk mencapai suatu tujuan dimana *use case* ini menggambarkan kebutuhan fungsional suatu sistem tanpa menampilkan struktur internal sistem.

b. *Class diagram*

Diagram yang memberikan pandangan secara luas dari suatu sistem dengan menunjukan kelas-kelasnya dan hubungan mereka.

c. *Activity diagram*

Representasi secara grafis dari proses dan *control flow* dan berfungsi untuk memperlihatkan alur dari satu aktivitas ke aktvitas yang lainnya serta menggambarkan perilaku yang kompleks.

d. *Sequance diagram*

Diagram yang digunakan untuk menggambarkan event yang dilakukan *actor eksternal* pada sistem atau *inter system* dilihat dalam satu *use case*.