BAB II

LANDASAN TEORI

2.1. Konsep Dasar Sistem

2.1.1. Pengertian Sistem

Secara umum, pengertian sistem informasi adalah suatu kesatuan yang terdiri dari komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi atau energi untuk mencapai suatu tujuan. Kata Sistem berasal dari bahasa Latin (*systema*) dan bahasa Yunani (*sustema*). Menurut (Kabanga, & Robert 2016) “Sistem merupakan kesatuan bagian-bagian yang saling berhubungan yang berada dalam suatu wilayah serta memiliki item-item penggerak.”

2.1.2. Pengertian Informasi

Menurut (Jayanti, Dwi & Iriani 2014) “Defenisi kata informasi sendiri secara internasional telah disepakati sebagai hasil dari pengolahan data yang secara prinsip memiliki nilai atau *value* yang lebih dibandingkan dengan data mentah

2.1.3. Pengertian Sistem Informasi

Menurut (Lestari 2014) ”Sistem informasi didefinisikan sebagai suatu alat yang menyajikan informasi dengan cara sedemikian rupa sehingga bermanfaat bagi penerimanya.”

2.1.4. Pengertian Sistem Informasi Penggajian

Menurut (Wignyowiyoto, Imron & Rofiah 2017) “Sistem Informasi Penggajian merupakan bagian dari Sistem informasi Sumber Daya manusia yang merupakan sub sistem dari Sistem Informasi Manajemen (SIM)”.

2.1.5. Karakteristik Sistem

Menurut (Reni, Haerani & Reza 2018) sistem tidak selalu memiliki kombinasi elemen-elemen yang sama, tetapi susunan dasar suatu sistem itu sama seperti adanya input/masukan yang akan diubah menjadi output/keluaran melalui suatu proses/transformati. Suatu mekanisme pengendalian akan memantau proses transformasi untuk meyakinkan bahwa sistem telah memenuhi tujuannya.

Adapun karakteristik yang dimaksud sebagai berikut :

1. Komponen Sistem (*Component*)

Komponen-komponen sistem tersebut dapat berupa suatu bentuk subsistem. Setiap subsistem memiliki sifat dari sistem yang menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan.

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 satu kesatuan yang tidak dapat dipisahkan.

3. Lingkungan Luar Sistem (*Environment*)

Bentuk apapun yang ada diluar ruang lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut disebut lingkungan luar sistem. Lingkungan luar sistem ini dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut.

4. Penghubung (*Interface*)

Media yang menghubungkan sistem dengan subsistem yang lain disebut penghubung sistem atau *interface*. Penghubung ini memungkinkan sumberdaya mengalir dari satu subsistem ke subsistem lain. Bentuk keluaran dari satu subsistem akan menjadi masukan untuk subsistem lain melalui penghubung tersebut. Dengan demikian, dapat terjadi suatu integrasi sistem yang membentuk satu kesatuan.

5. Masukan Sistem (*Input*)

Energi yang dimasukkan ke dalam sistem disebut masukan sistem, yang dapat berupa pemeliharaan (*maintenance input*) dan sinyal (*signal input*).

6. Keluaran Sistem (*Output*)

Hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran ini dapat menjadi masukan bagi subsistem yang lain seperti sistem informasi. Keluaran yang dihasilkan adalah informasi. Informasi ini dapat digunakan sebagai masukan untuk pengambilan keputusan atau hal-hal lain yang menjadi input bagi subsistem lain.

7. Pengolah Sistem (*Process*)

Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran.

8. Sasaran (*Objectives*)

Suatu sistem mempunyai tujuan dan sasaran yang pasti dan bersifat deterministik. Kalau suatu sistem tidak memiliki sasaran maka operasi sistem tidak ada gunanya. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuan yang telah direncanakan.

9. Tujuan (*Goal*)

Suatu sistem dikatakan berhasil apabila mengenai sasaran atau tujuannya.

Jika suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya

2.1.6. Klasifikasi Sistem

1. Sistem Abstrak dan Sistem Fisik

a. Sistem abstrak (*abstract system*) merupakan sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik (sistem teologia yang merupakan suatu sistem yang menggambarkan hubungan antara Tuhan dengan manusia).

b. Sistem Fisik (*Physical System*) diartikan sebagai sistem yang nampak secara fisik sehingga setiap makhluk dapat melihatnya, misalnya sistem komputer, sistem akuntansi, sistem produksi, dan lain-lain.

2. Sistem Alamiah dan Sistem Buatan Manusia

a. Sistem Alamiah (*Natural System*) merupakan sistem yang terjadi melalui proses alam, tidak dibuat oleh manusia, misalnya sistem tata surya, sistem galaksi, sistem reproduksi dan lain-lain.

b. Sistem Buatan Manusia (*Human Made System*) merupakan sistem yang dirancang oleh manusia. Sistem buatan yang melibatkan interaksi manusia, misalnya sistem akuntansi, sistem informasi, dan lain-lain.

3. Sistem Deterministik dan Sistem Probabilistik

- a. Sistem Deterministik (*Deterministic System*) merupakan sistem yang beroperasi dengan tingkah laku yang sudah dapat 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-program komputer yang dijalankan.
- b. Sistem Probabilistik (*Probabilistic System*) merupakan sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas, misalnya sistem manusia.

4. Sistem Terbuka dan Sistem Tertutup

- a. Sistem Terbuka (*Open System*) merupakan sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya. Lebih spesifik dikenal juga yang disebut dengan sistem terotomasi, yang merupakan bagian dari sistem buatan manusia dan berinteraksi dengan kontrol oleh satu atau lebih komputer sebagai bagian dari sistem yang digunakan dalam masyarakat modern. Sistem ini menerima masukan dan menghasilkan keluaran untuk subsistem lainnya, misalnya sistem kebudayaan manusia.
- b. Sistem Tertutup (*Closed System*) merupakan sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa campur tangan dari pihak luar. Secara teoritis sistem tersebut ada, tetapi kenyataannya tidak ada sistem yang benar-benar tertutup, yang ada hanyalah *relatively closed system* (secara relatif tertutup, tidak benar-benar tertutup).

2.1.7. Basis Data (*Database*)

Menurut (Reni, Haerani & Reza 2018) “mendefinisikan Basis Data sebagai berikut “Basis Data adalah kumpulan data yang dapat digambarkan sebagai aktivitas dari satu atau lebih organisasi yang berelasi.”

Secara umum, Basis Data adalah kumpulan informasi yang disimpan didalam komputer secara sistematis, sehingga dapat di periksa menggunakan suatu program komputer untuk memperoleh informasi dari Basis Data tersebut. Penerapan Basis Data dalam sistem informasi disebut dengan *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 didalam suatu organisasi.

1. Jenjang Data

Jenjang data terdiri dari :

a. *Character*

Merupakan data yang terkecil, dapat berupa karakter-karakter khusus (*Special Character*) yang membentuk suatu sistem.

b. *Field*

Suatu *Field* menggambarkan suatu *atribut* dari *record* yang menunjukkan suatu item dari data.

c. *Record*

Terdiri dari beberapa kumpulan elemen-elemen yang saling berkaitan menginformasikan tentang suatu *Entity* secara lengkap.

d. *File*

Kumpulan *Record-record* yang sejenis yang mempunyai panjang elemen yang sama, namun berbeda-beda dengan data *Value* nya.

e. *Database*

Merupakan kumpulan file-file yang mempunyai ikatan antara satu file dengan file lainnya sehingga membentuk satu bangunan dan data untuk menginformasikan satu perusahaan, instalasi dalam batasan tertentu.

2.1.8. Metode Pengembangan Perangkat Lunak

Metode rekayasa piranti lunak yang digunakan peneliti adalah metode waterfall.

Model *waterfall* atau seringkali disebut sebagai *classic life cycle* adalah model pengembangan perangkat lunak yang menekankan fase-fase yang berurutan dan sistematis, dimulai dari spesifikasi kebutuhan konsumen dan berkembang melalui proses perencanaan (*planning*), pemodelan (*modelling*), pembangunan (*construction*), dan penyebaran (*deployment*), yang berujung pada dukungan terus menerus untuk sebuah perangkat lunak yang utuh. Model ini dapat digunakan pada saat kebutuhan untuk sebuah masalah telah dipahami dengan baik, dan pekerjaan dapat mengalir secara linear dari proses komunikasi hingga penyebaran (*deployment*). Situasi ini ditemui saat adaptasi atau perpanjangan dari sistem yang ada sudah terdefinisi dengan baik. Adapun model ini juga dapat digunakan pada situasi di mana dibutuhkan usaha yang terbatas untuk pengembangan perangkat lunak, namun kebutuhan perangkat lunak sudah terdefinisi dengan baik dan cenderung stabil. Namun, dalam pengembangan perangkat lunak, model ini cenderung menjadi salah satu pendekatan yang kurang iteratif dan fleksibel, karena proses mengalir satu arah ("ke bawah" seperti air terjun)

2.2. Teori Pendukung

2.2.1. *Entity Relationship Diagram (ERD)*

Menurut (Reni, Haerani & Reza 2018) “ERD (*Entity Relationship Diagram*) adalah model teknik pendekatan yang menyatakan atau menggambarkan hubungan suatu model”. Didalam hubungan tersebut dinyatakan yang utama dari ERD adalah menunjukkan objek data (*Entity*) dan hubungan (*Relationship*) yang ada pada Entity berikutnya.

Simbol-simbol dalam ERD (*Entity Relationship Diagram*) adalah sebagai berikut :

1. *Entity*

Entity adalah sesuatu apa saja yang ada dalam sistem, nyata maupun abstrak dimana data tersimpan atau dimana terdapat data. Entitas diberi nama dengan kata benda atau dapat di kelompokkan dalam empat jenis nama yaitu orang, benda, lokasi, dan kejadian.

2. *Relationship*

Relationship adalah hubungan alamiah yang terjadi antara entitas.

3. *Relationship Degree*

Relationship Degree atau derajat *relationship* adalah jumlah entitas yang berpartisipasi dalam suatu *relationship*

4. *Atribut*

Atribut adalah sifat atau karakteristik dari tiap entitas maupun tiap *relationship*.

5. Kardinalitas (*Cardinalitas*)

Kardinalitas relasi menunjukkan jumlah maksimum tupel yang dapat berelasi dengan entitas pada entitas lain. Terdapat tiga macam kardinalitas relasi yaitu :

a. One To One

Tingkat hubungan satu ke satu, dinyatakan dengan satu kejadian pada entitas pertama, hanya mempunyai hubungan dengan satu kejadian pada entitas yang kedua atau sebaliknya.

b. One To Many atau Many To One

Tingkat hubungan satu ke banyak adalah sama dengan banyak ke satu. Tergantung dari arah mana hubungan tersebut dilihat.

c. Many To Many

Tingkat hubungan banyak ke banyak terjadi jika tiap kejadian pada sebuah entitas akan mempunyai banyak hubungan dengan kejadian pada entitas lainnya. Baik dilihat dari sisi entitas yang pertama, maupun dilihat dari sisi entitas yang kedua.

2.2.2. *Logical Record Structure (LRS)*

Menurut (Kushardono 2014) "*Logical Record Structure (LRS)*" adalah representasi dari struktur record pada tabel-tabel yang terbentuk

dari hasil relasi antar himpunan entitas. Menentukan kardinalitas, jumlah tabel, dan *Foreign Key* (KY)”.

Berikut adalah cara membentuk skema database atau LRS (*Logical Record Structured*) berdasarkan Entity Relationship Diagram :

- a. Jika relasinya satu ke satu, maka foreign key diletakkan pada salah satu dari dua entitas yang ada guna menyatukan kedua entitas tersebut.
- b. Jika relasinya satu ke banyak, maka foreign key diletakkan pada entitas Many.
- c. Jika relasinya banyak ke banyak, maka dibuat “file konektor” yang berisi dua foreign key yang berasal dari kedua entitas.

2.2.3. *Unfied Modeling Languange* (UML)

Menurut (Syahputera 2014) “*Unfied Modeling Languange* (UML) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requitmen, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemograman berorientasi objek”.

UML muncul karena adanya kebutuhan pemodel visual untuk menspesifikasikan, menggambar, membangun dan dokumentasi dari sistem perangkat lunak.

UML mempunyai beberapa atau sejumlah elemen grafis yang bisa dikombinasikan menjadi diagram. Diagram tersebut akan menggambarkan

atau mendokumentasikan beberapa aspek dari sebuah sistem. Abstraksi konsep dasar UML terdiri dari *structure classification*, *dynamic behavior*, dan *model management*. Berikut adalah diagram yang ada pada UML :

1. *Activity Diagram*

Activity Diagram atau diagram aktivitas menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu, yang ada pada perangkat lunak. Diagram aktivitas menggambarkan aktivitas sistem, bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

2. *Use Case Diagram*

Use case diagram atau *diagram use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan di buat.

Berikut contoh *Use Case Diagram* Penggajian

3. *Class Diagram*

Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut dengan atribut dan metode atau operasi.

4. *Sequence Diagram*

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirim dan diterima antar objek.