**BAB II**

**LANDASAN TEORI**

* 1. **Pengertian Sistem Informasi**

Sistem Informasi merupakan suatu sistem didalam 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 laporan-laporan yang diperlukan oleh pihak luar tertentu (Tata Sutabri, 2012).

Sistem Informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan (Jogiyanto, 2005).

* 1. **Pengertian Pelayanan**

Pelayanan menurut KBBI (Kamus Besar Bahasa Indonesia) adalah usaha melayani kebutuhan orang lain dengan memperoleh imbalan (uang) atau jasa, sedangkan melayani yaitu membantu menyiapkan (membantu apa yang diperlakukan seseorang).

Menurut R.A Supriyono pelayanan adalah kegiatan yang  
diselenggarakan organisasi menyangkut kebutuhan pihak konsumen dan akan  
menimbulkan kesan tersendiri, dengan adanya pelayanan yang baik maka  
konsumen akan merasa puas,dengan demikian pelayanan merupakan hal yang  
sangat penting dalam upaya menarik konsumen untuk menggunakan produk  
atau jasa yang ditawarkan.

* 1. **Pengertian Jasa**

Payne (2001) berpendapat dalam ilmu ekonomi, jasa atau layanan memiliki arti dengan aktivitas ekonomi yang melibatkan sejumlah interaksi dengan konsumen atau dengan barang-barang milik, tetapi tidak menghasilkan transfer kepemilikan.

* 2. **Alat Bantu Deain Sistem**

**2.4.1. Diagram Alir Data (Flow Of Document)**

Dalam analisa sistem salah satu alat bantu yang dipakai adalah diagram alir data (*Flow Of Document*), diagram ini menunjukkan alir (*flow*) didalam program atau prosedur sistem secara logika. Tujuan penggunaan diagram alir data adalah untuk menggambarkan suatu tahapan penyelesaian masalah secara sederhana, terurai, dan jelas menggunakan simbol – simbol yang standar. Adapun simbol – simbolnya sebagai berikut:

**Tabel 2. 1 Simbol Diagram Alir Data (Flow Of Document)**

|  |  |  |
| --- | --- | --- |
| SIMBOL | NAMA | FUNGSI |
|  | *Terminator* | Menunjukkan awal dan akhir proses |
|  | *Process* | Menunjukkan kegiatan proses dari operasi program computer |
|  | *Manual Operation*  (Operasi Manual) | MenunjukkanPengolahan yang tidak dilakukan dengan komputer |
|  | *Decision*  (Keputusan) | Menunjukkan pernyataan yang menghasilkan jawaban Ya/Tidak |
|  | *Document* | Menunjukkan dokumen *input* dan *output* |
|  | *Multi Document* | Menunjukkan dokumen yang lebih dari satu |
| N  A  C | *Off-line Storage* | Menunjukkan bahwa data dalam symbol ini akan disimpan ke suatu media tertentu.  Terdapat 3 simbol *Off-line Storage* :  C (*Chronological:* berdasarkan tanggal)  A (*Alphabetic:* berdasarkan huruf)  N (*Numerical:* berdasarkan angka) |

Bersambung

Lanjutan Tabel 2.1

|  |  |  |
| --- | --- | --- |
|  | *Input/Output*  (Masukan/Keluaran) | Menunjukkan data input/output |
|  | *Subproses*  (Proses Terdefinisi) | Menunjukkan suatu operasi yang rinciannya, ditunjukkan ditempat lain |
|  | *Arrow*  (Arah) | Untuk menyatakan jalannya arus suatu proses. |
|  | *Connector*  (Penghubung) | Menunjukan penghubung halaman yang sama atau halaman lain |

Sumber: Jogiyanto (2005)

* + 1. **UML (Unified Modelling Language)**

Menurut Rosa A. S dan M. Shalahuddin (2016), *Unified Modelling Language* (UML) adalah bahasa permodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek. UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun dan dokumentasi dari sistem perangkat lunak. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung.

**Diagram-diagram yang terdapat dalam UML**

Pada UML 2.3 terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori yaitu structure diagram, behavior diagram, intraction diagram. Beberapa penjelasan tentang macam – macam diagram pada UML yaitu:

* + - 1. ***Use Case Diagram***

*Use case diagram* 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 dibuat. Secara kasar, use case digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem

informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Adapun simbol-simbol yang ada pada *use case diagram* terlihat pada tabel 2.2.

**Tabel 2.2 Simbol Use Case Diagram**

|  |  |  |
| --- | --- | --- |
| Simbol | Nama | Keterangan |
| nama aktor | *Aktor/actor* | Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda diawal frase nama actor |
|  | *Use case* | Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama *use case* |
| <<include>>    <<uses>> | *include / uses* | Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsinya atau sebagai syarat dijalankan use case ini.  Include berarti use case yang ditambahkan akan selalu dipanggil saat use case tambahan dijalankan. |
|  | *Asosiasi* | Komunikasi antara aktor dan *use case* yang berpartisipasi pada *use case* atau *use case* memiliki interaksi dengan aktor. |

Sumber: Rosa A.S dan M. Shalahudin (2016)

1. ***Class Diagram***

Diagram kelas atau *Class Diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Adapun simbol-simbol yang ada pada *Class Diagram* terlihat pada table 2.3.

**Tabel 2.3 Notasi Class Diagram**

|  |  |  |
| --- | --- | --- |
| Simbol | Nama | Keterangan |
| + Attribute 1  + Attribute 1  Nama kelas  + Operation 1() | *Class* | Kelas pada struktur sistem. |
|  | *Interface* | Sama dengan konsep *interface* dalam pemrograman berorientasi objek. |
|  | *Association* | Relasi antar kelas dengan makna umum |
|  | *Directed Association* | Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain |

Sumber: Rosa A.S dan M. Shalahudin (2016)

1. ***Sequence Diagram***

*Diagram sequence* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirim dan diterima oleh objek. Oleh karena itu untuk menggambarkan diagram sekuen harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek. Adapun simbol-simbol yang ada pada *Squence Diagram* terlihat pada table 2.4.

**Tabel 2.4 Notasi Sequence Diagram**

|  |  |  |
| --- | --- | --- |
| Simbol | Nama | Keterangan |
| nama aktor  atau  nama\_aktor  tanpa waktu aktif | *Actor* | Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem infromasi yang akan dibuat itu sendiri, jadi walaupun simbol dari actor adalah gambar orang, tapi actor belum tentu merupakan orang. |
| nama objek : nama kelas | *Object* | Menyatakan objek yang berinteraksi pesan. |
| 1 : masukan | Pesan tipe *send* | Menyatakan bahwa suatu objek mengirimkan data |
| 1 : keluaran | Pesan tipe *return* | Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi |
|  | *Lifeline* | Menyatakan kehidupan suatu objek |
|  | Waktu aktif | Menyatakan objek dalam keadaan aktif dan berinteraksi pesan |
| <<create>> | Pesan tipe *create* | Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat. |
| 1 : nama\_metode() | Pesan tipe *call* | Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri. |

Sumber: Rosa A.S dan M. Shalahudin (2016)

1. ***Activity Diagram***

*Activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis yang ada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Adapun simbol-simbol yang ada pada *Activity Diagram* terlihat pada table 2.5.

**Tabel 2.5 Notasi Activity Diagram**

|  |  |  |
| --- | --- | --- |
| Simbol | Nama | Keterangan |
|  | Status awal | Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal |
| aktivitas | Aktivitas | Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja |
|  | Percabangan / *decision* | Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu |
|  | Penggabungan / *join* | Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu |
|  | Status akhir | Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir |
|  | *Swimlane* | Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi |

Sumber: Rosa A.S dan M. Shalahudin (2016)

1. ***Statechart Diagram***

*Statechart Diagram* digunakan untuk menggambarkan perubahan status atau transisi status dari sebuah mesin atau sistem. Perubahan tersebut digambarkan

dalam suatu graf berarah. Adapun simbol-simbol yang ada pada *Statechart Diagram* terlihat pada table 2.6.

**Tabel 2.6 Notasi Statechart Diagram**

|  |  |  |
| --- | --- | --- |
| Simbol | Nama | Keterangan |
|  | *Start (Initial State)* | *Start* atau *initial state* adalah *state* atau keadaan awal pada saat sistem mulai hidup |
|  | *End (Final State)* | *End* atau *final* *state* adalah *state* keadaan akhir dari daur hidup suatu sistem |
| *Event* | *Event* | *Event* adalah kegiatan yang menyebabkan berubahnya status mesin |
| *State* | *State* | *State* atau status adalah keadaan sistem pada waktu tertentu. *State* dapat berubah jika ada *event* tertentu yang memicu perubahan tersebut |

Sumber: Rosa A.S dan M. Shalahudin (2016)

* + 1. **ERD (Entity Relationship Diagram)**

*Entity Relationship Diagram* (ERD) merupakan suatu model data yang dikembangkan berdasarkan objek. Diagram hubungan entitas atau yang lebih dikenal dengan sebutan ERD adalah notasi grafik dari sebuah model data atau sebuah model jaringan yang menjelaskan tentang data yang tersimpan (*storage* data) dalam sistem secara abstrak. Adapun macam-macam simbolnya terlihat pada table 2.7.

**Tabel 2.7 Simbol-simbol ERD**

Bersambung

|  |  |
| --- | --- |
| Simbol | Keterangan |
| Entitas / entity  Nama\_entitas | Entitas merupakan data inti yang akan disimpan untuk tabel pada basis data |
| Atribut | *Field* atau kolom data yang butuh disimpan dalam suatu entitas. Jika f*ield* atau kolom data merupakan kunci akses *record* yang diinginkan, maka diberi garis bawah. |
| Relasi | Relasi yang menghubungkan antar entitas |
| Asosiasi / association | Penghubung antar relasi dan entitas dimana kedua ujungnya memiliki *multiplicity* kemungkinan jumlah kemungkinan. |

Sumber : Rosa A.S dan M. Shalahudin (2016)

Lanjutan Tabel 2.7