

BAB II

LANDASAN TEORI

2.1. Konsep Dasar

2.1.1. Sistem

Menurut (Tohari, 2017) mengemukakan bahwa “Sistem adalah himpunan dari *variable* saling berinteraksi, dan satu sama lain saling bergantung”.

Karakteristik suatu sistem menurut (Tohari, 2017) adalah:

1. **Komponen atau elemen (*Components*)**
satu kesatuan sistem yang terdiri dari interaksi komponen-komponen yang bekerja sama membentuk satu kesatuan.
2. **Batas Sistem (*Boundary*)**
Daerah yang membatasi antara sistem yang satu dengan sistem yang berada di luar.
3. **Lingkungan luar subsistem (*Environment*)**
Segala sesuatu yang mempengaruhi operasi suatu sistem diluar batas sistem itu sendiri.
4. **Penghubung sistem (*Interface*)**
Satu kesatuan yang dapat menghubungkan subsistem dengan subsistem lainnya.
5. **Masukan (*Input*)**
Sesuatu yang dapat mengasilkan suatu luaran yaitu berupa masukan *energy* ke dalam suatu sistem supaya sistem dapat beroperasi .

6. Luaran (*Output*)

Merupakan hasil dari energi yang diolah dan diklasifikasikan menjadi luaran yang berguna, juga merupakan luaran atau tujuan akhir dari sistem.

7. Pengolah (*Process*)

Suatu sistem mempunyai bagian pengolah yang akan mengubah *input* menjadi *output*.

8. Sasaran (*Objekive*)

Dari sistem sangat menentukan sekali masukkan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem.

2.1.2. Informasi

Menurut (Syukron, 2015) mengemukakan bahwa “Informasi adalah data yang diolah menjadi bentuk yang lebih berarti bagi penerima dan bermanfaat dalam pengambilan keputusan saat ini atau mendatang dan lebih berarti bagi yang menerimanya”.

2.1.3. Sistem Informasi

Menurut (Kristanto, 2018) mengemukakan bahwa “Sistem Informasi merupakan kumpulan dari perangkat keras, perangkat lunak komputer serta perangkat manusia yang akan mengolah data menggunakan perangkat keras dan perangkat lunak tersebut”.

2.1.4. Akuntansi

Menurut Hery, (2015:8) mengemukakan bahwa “Akuntansi merupakan sebuah sistem informasi yang memberikan laporan kepada para pengguna informasi akuntansi atau kepada pihak-pihak yang memiliki kepentingan terhadap hasil kinerja dan kondisi keuangan perusahaan”.

2.1.5. Penjualan

Menurut (Linggariama, 2020) Mengemukakan bahwa “Aktivitas penjualan merupakan sumber pendapatan perusahaan yang dapat berfungsi untuk membiayai kegiatan operasional perusahaan sehingga perlu adanya pengelolaan yang baik agar sasaran penjualan tercapai”.

2.1.6. Jurnal Umum

Menurut (Sudrajat et al., 2016) “Jurnal umum adalah sebuah daftar khusus untuk mencatat setiap aktivitas transaksi yang dilakukan suatu usaha secara kronologis sesuai urutan tanggal ke dalam jumlah yang harus didebet dan dikredit. Jurnal umum merupakan tempat pertama kali untuk mencatat transaksi”.

Jurnal penjualan diantaranya:

1. Jurnal untuk penjualan tunai

Kas	XXX
Penjualan	XXX

2. Jurnal untuk penjualan kredit

Piutang Usaha	XXX
Penjualan	XXX

2.1.7. Program

Menurut (Sukamto A & Shalahuddin, 2016) mengemukakan bahwa “Pemrograman terstruktur adalah konsep paradigma atau sudut pandang pemrograman yang membagi-bagi program berdasarkan fungsi-fungsi atau prosedur-prosedur yang dibutuhkan program komputer”.

Berikut contoh bahasa pemrograman yang mendukung pemrograman berorientasi objek menurut Sukamto dan Shalahuddin (Sukamto A & Shalahuddin, 2016) adalah:

1. Bahasa Pemrograman *Smalltalk*
2. Bahasa Pemrograman *Eiffel*
3. Bahasa Pemrograman *C++*
4. Bahasa Pemrograman (*web*) *PHP*
5. Bahasa Pemrograman *Java*

2.1.8. Bahasa Pemrograman

Menurut (Puspitasari, 2016) mengemukakan bahwa “*PHP* atau yang memiliki kepanjangan *PHP Hypertext Preprocessor*, merupakan suatu bahasa pemrograman yang difungsikan untuk membangun suatu *website* dinamis. *PHP* menyatu dengan kode *HTML*, *HTML* digunakan sebagai pembangun atau pondasi dari kerangka *layout web*, sedangkan *PHP* difungsikan sebagai prosesnya, sehingga dengan adanya *PHP* tersebut, sebuah web akan sangat mudah di *maintenance*”.

“*HyperText Markup Language* merupakan suatu metode untuk mengimplementasikan konsep *hypertext* dalam suatu naskah atau dokumen. *HTML* sendiri bukan tergolong pada suatu bahasa pemrograman karena sifatnya yang hanya memberikan tanda (*marking up*) pada suatu naskah teks dan bukan sebagai program”.

“*Cascading Style Sheet (CSS)* adalah suatu bahasa *style sheet* yang digunakan untuk mengatur tampilan suatu dokumen yang ditulis dalam bahasa *markup*. Penggunaan yang paling umum dari *CSS* adalah untuk memformat halaman *web* yang ditulis dengan *HTML* dan *XHTML*. Walaupun demikian, bahasanya sendiri dapat dipergunakan untuk semua jenis dokumen *XML* termasuk *SVG* dan *XUL*. Spesifikasi *CSS* diatur oleh *World Wide Web Consortium (W3C)*”.

2.1.9. Basis Data

Menurut (Nurrahman et al., 2018) mengemukakan bahwa :

“Basis data dapat dipahami sebagai suatu kumpulan data terhubung (*interrelated data*) yang disimpan bersama-sama dalam suatu media, tanpa mengatap satu sama lain atau tidak perlu satu kerangkapan data”. Data disimpan dengan cara-cara tertentu sehingga mudah digunakan atau ditampilkan kembali, data dapat digunakan oleh satu atau lebih program-program aplikasi secara optimal”.

2.2. Peralatan Pendukung (*Tools System*)

2.2.1. Konsep Dasar *Unitified Modelling Language*(UML)

Menurut (Puspitasari, 2016) mengemukakan bahwa “ *UML* merupakan Bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung”.

2.2.2. *Use Case Diagram*

Menurut (Sukamto A & Shalahuddin, 2016) “*Use Case* 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 dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu”.

2.2.3. *Activity Diagram*

Menurut (Abdussomad et al., 2016) “*Activity Diagram* digunakan untuk model perilaku dalam independen proses bisnis Suatu objek. Dalam banyak hal, *Activity Diagram* dapat dipandang sebagai diagram aliran data yang canggih yang dapat digunakan dalam hubungannya dengan analisis terstruktur”.

2.2.4. *Entity Relationship Diagram* (ERD)

Menurut (Abdussomad et al., 2016) “*Entity Relation Diagram* adalah alat pemodelan data utama dan akan membantu mengorganisasi data dalam suatu proyek ke dalam entitas-entitas dan menentukan hubungan antar entitas”

2.2.5. *Logical Record Structure* (LRS)

Menurut (Puspitasari, 2016) mengemukakan bahwa :
“Sebuah model sistem yang digambarkan dengan sebuah diagram-ER akan mengikuti pola/ aturan pemodelan tertentu dalam kaitannya dengan konversi ke LRS, maka perubahan yang terjadi adalah mengikuti aturan - aturan berikut ini :
Setiap *entitas* akan diubah kebentuk kotak, Sebuah *atribut* relasi disatukan dalam sebuah kotak bersama entitas jika hubungan yang terjadi pada diagram-ER 1:M (relasi bersatu dengan *cardinality* M) atau tingkat hubungan 1:1 (relasi bersatu

dengan *cardinality* yang paling membutuhkan referensi), sebuah relasi dipisah dalam sebuah kotak tersendiri (menjadi entitas baru) jika tingkat hubungannya M:M (*many to many*) dan memiliki *foreign key* sebagai *primary key* yang diambil dari kedua *entitas* yang sebelumnya saling berhubungan”.

2.2.6. Sequence Diagram

Menurut (Sukamto A & Shalahuddin, 2016) menjelaskan bahwa, “Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek”.

2.2.7. Deployment Diagram

Menurut (Tohari, 2017) menjelaskan bahwa, “Diagram *deployment* menunjukkan tata letak sebuah sistem secara fisik. Diagram ini akan menampakan bagian-bagian *software* yang berjalan pada bagian-bagian *hardware* yang digunakan untuk mengimplementasikan sebuah sistem dan kethubungan antara komponen-komponen *hardware* tersebut”.

Ada beberapa hal yang harus diidentifikasi terlebih dahulu menurut Tohari (Tohari, 2017) yaitu:

1. Menentukan *node*.
2. Hubungan antar *node*.

2.2.8. User Interface

Menurut (Tohari, 2017)t menjelaskan bahwa, “*Interface* adalah satu *set operation* yang memberikan spesifikasi beberapa aspek dari perilaku dan operasi disuatu *class* ke *class* yang lain”.

2.2.9. Konsep Dasar *Black Box Testing*

Menurut (Sukamto A & Shalahuddin, 2016) menjelaskan bahwa, “*Black-Box Testing* (pengujian kotak hitam) yaitu menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan”.

Kasus uji yang dibuat untuk melakukan pengujian kotak hitam harus dibuat dengan kasus benar dan kasus salah, misalkan untuk kasus proses *login* maka kasus uji yang dibuat menurut (Sukamto A & Shalahuddin, 2016) adalah:

1. Jika *user* memasukkan nama pemakai (*username*) dan kata sandi (*password*) yang benar.
2. Jika *user* memasukkan nama pemakai (*username*) dan kata sandi (*password*) yang salah, misalnya nama pemakai benar tapi kata sandi salah, atau sebaliknya, atau keduanya salah.