

BAB II

LANDASAN TEORI

2.1. Konsep Dasar

2.1.1. Sistem

Menurut (Tohari, 2017) mengemukakan bahwa “Sistem adalah kumpulan atau himpunan dari unsur atau variabel-variabel yang saling terkait, saling berinteraksi, dan saling tergantung satu sama lain”.

Karakteristik suatu sistem menurut (Tohari, 2014) adalah:

1. Komponen atau elemen (*components*)
2. Batas Sistem (Boundary)
3. Lingkungan luar subsistem (*Environment*)
4. Penghubung sistem (*interface*)
5. Masukan (*input*)
6. Luaran (*output*)
7. Pengolah (*process*)
8. Sasaran (*objekive*)

Sedangkan (Abdussomad et al., 2016) mengemukakan pendapatnya bahwa “Sistem adalah jaringan kerja yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran.”

2.1.2. Informasi

Menurut (Tohari, 2017) mengemukakan bahwa “Informasi merupakan data yang telah diolah untuk digunakan dalam proses pengambilan keputusan”.

Sedangkan kualitas dari satu informasi tergantung dari tiga hal menurut (Tohari, 2017) adalah:

1. Akurat artinya informasi harus bebas dari kesalahan-kesalahan.
2. Tepat artinya informasi yang datang pada penerima tidak boleh terlambat.
3. *Relevan* artinya informasi harus mempunyai manfaat bagi pemakainnya.

2.1.3. Sistem Informasi

Menurut (Lubis, 2016) Mengemukakan bahwa “Sistem informasi adalah sistem yang ada di dalam sebuah organisasi yang mengolah transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari organisasi tersebut untuk dapat menyediakan laporan-laporan yang diperlukan kepada pihak luar”.

2.1.4. Akuntansi

Menurut (Hery, 2015) mengemukakan bahwa “Akuntansi merupakan sebuah sistem informasi yang memberikan laporan hasil kinerja dan kondisi keuangan perusahaan yang akan berikan kepada kepada pihak-pihak yang memiliki kepentingan”.

2.1.5. Sistem Informasi Akuntansi

Menurut (Tresnawati et al., 2017) mengemukakan “Sistem Informasi Akuntansi adalah komponen organisasi yang dirancang untuk mengolah data keuangan menjadi informasi atau laporan keuangan, yang di tujukan kepada pihak internal maupun eksternal perusahaan”.

2.1.6. Pembelian dan Penjualan

Menurut (Sholikhah et al., 2017) mengemukakan bahwa “Aktivitas pembelian dapat disebut dengan prokuremen. Sedangkan penjualan merupakan aktivitas jual dan beli barang dan jasa kepada konsumen”.

Penjualan tunai yaitu penjualan yang dilakukan dengan tidak ada sistem hutang. Sedangkan penjualan yang dilakukan dengan sistem hutang disebut dengan penjualan kredit. Berikut ini merupakan jurnal pembelian dan penjualan:

1. Jurnal Pembelian Secara Tunai

Persediaan	XX	
Kas		XX

2. Jurnal Penjualan Tunai

Kas	XX	
Penjualan		XX

2.1.7. Bahasa Pemrograman Java

Menurut (Nofriadi, 2018) mengemukakan bahwa: “Bahasa pemrograman yang dapat dijalankan diberbagai operasi salah satunya telepon genggam merupakan Bahasa Pemrograman Java. Bahasa pemrograman ini merupakan pengembangan dari bahasa pemrograman C++ karena banyak mengadopsi sintak C dan C++”.

2.1.8. Netbeans IDE 8.2

Menurut (Rusmayanti, 2015) mengemukakan bahwa “Netbeans merupakan salah satu IDE yang dikembangkan dengan bahasa pemrograman java. ”.

2.1.9. Basis Data (*Database*)

Menurut (Nurrahman et al., 2018) menyimpulkan 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”

2.2.10. Entity Relationship Diagram (ERD)

Menurut (Abdussomad, 2016) pada dasarnya ada tiga simbol yang digunakan sebagai berikut:

1. Entitas digambarkan dengan kotak persegi panjang.
2. Hubungan atau Relasi digambarkan dengan kotak berbentuk diamond..
3. Atribut digambarkan dengan bentuk elips.
4. Tingkat Hubungan atau Cardinality,

Entity Relationship Diagram (ERD) juga menunjukkan tingkat hubungan yang terjadi

2.2.11. Logical Record Structure (LRS)

Menurut (Junaidi, 2016) mengemukakan bahwa “*Logical Record Structure* (LRS) merupakan hasil dari pemodelan *entity relationship* (ER) beserta atributnya sehingga bisa terlihat hubungan-hubungan antarentitas”.

2.2. Peralatan Pendukung (Tools System)

Peralatan pendukung merupakan alat yang digunakan untuk menggambarkan bentuk logika model dari sistem yang akan dibuat. Adapun peralatan yang dipakai penulis sebagai berikut:

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

Menurut (Puspitasari, 2017) mengemukakan bahwa “ UML adalah sebuah bahasa untuk visualisasi, merancang dan mendokumentasikan sistem perangkat lunak”.

Menurut (Abdussomad, 2016) “*UML (Unified Modeling Language)* adalah bahasa pemodelan dalam sistem atau perangkat lunak yang berparadigma berorientasi objek”

2.2.2. Use Case Diagram

Menurut (Sukamto dan Shalahuddin, 2018) mengatakan bahwa “*Use Case Diagram* merupakan pembuatan model kelakuan dari sistem informasi. *Use case* sendiri merupakan interaksi dari actor dengan pembuatan sistem informasi”.

Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case* menurut (Sukamto dan Shalahuddin, 2018), adalah:

1. Aktor merupakan 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.
2. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

2.2.3. Activity Diagram

Menurut (Sukamto dan Shalahuddin, 2016) mengatakan bahwa “*Activity diagram* menggambarkan proses bisnis dari sebuah sistem atau menu yang tersedia di dalam perangkat lunak”.

Berikut menurut (Sukamto dan Shalahuddin, 2016) adalah:

1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
2. Urutan atau pengelompokan tampilan dari sistem/*user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
3. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.
4. Rancangan menu yang ditampilkan pada perangkat lunak.

2.2.4. *Sequence Diagram*

Menurut (Tohari, 2017) mengemukakan bahwa “*Sequence diagram* menggambarkan interaksi antara sejumlah objek dalam urutan waktu”.

Sekuen yang di gambar harus sebanyak pendefinisian *use case*. Maka banyak *use case* yang dibuatkan sama banyak dengan banyak diagram sekuen.

2.2.5. *Deployment Diagram*

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

Untuk menggambar diagram *deployment*, ada beberapa hal yang harus diidentifikasi terlebih dahulu menurut (Tohari, 2017) yaitu:

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

Penentuan *node*, diantaranya dapat dilakukan dengan beberapa pedoman menurut (Tohari, 2017) sebagai berikut:

1. *Node* merupakan elemen fisik yang sudah tersedia dalam sistem atau aplikasi.
Amatilah dan jadikan elemen fisik sebagai *node* di diagram *deployment*.
2. *Node* mewakili sumber-sumber komputasi.
3. *Node* dilambangkan dengan kubus.

2.2.6. *User Interface*

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

Misalnya, *keyboard* pada komputer sebenarnya merupakan *interface* yang bisa dipakai ulang karena tombol-tombol *keyboard* sebenarnya berasal dari mesin ketik, hanya saja mungkin ada beberapa operasionalisasi tombol-tombol yang berbeda yang sudah ditransfer ke sistem yang lain. Misalnya tombol *control*, *page up*, *page down* dan lain-lain.

2.2.7. Konsep Dasar *Black Box Testing*

Menurut (Sukamto dan Shalahuddin, 2016) menjelaskan bahwa, “Pengujian *Blackbox testing* yaitu pengujian spesifikasi fungsional pada perangkat lunak tanpa mengujikan desain dan kode program”.

