BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

A. Konsep Dasar Sistem Informasi

Menurut Pratama (2014:7) sistem didefinisikan sebagai sekumpulan prosedur yang saling berkaitan dan saling terhubung untuk melakukan suatu tugas bersama-sama. Informasi merupakan hasil pengolahan data dari satu atau berbagai sumber yang kemudian diolah, sehingga memberikan nilai, arti, dan manfaat.

Berdasarkan definisi mengenai sistem dan informasi yang telah dijelaskan diatas, maka dapat dinyatakan bahwa sistem informasi merupakan bagian dari empat bagian utama. Keempat bagian utama tersebut mencakup perangkat lunak (*software*), perangkat keras (*hardware*), infrastruktur dan Sumber Daya Manusia (SDM) yang terlatih. Keempat bagian utama ini saling berkaitan untuk menciptakan sebuah sistem yang dapat mengolah data menjadi informasi yang bermanfaat.

B. Konsep Dasar Pemrograman

Menurut Mustakini (2005:582) Pemrograman (*programming*) merupakan kegiatan menulis kode program yang akan dieksekusi oleh komputer. Kode program yang ditulis oleh pemrogram (*programmer*) harus berdasarkan dokumentasi ulang disediakan oleh analisis sistem hasil dari desain sistem secara rinci. Hasil program yang sesuai dengan desainnya akan menghasilkan program yang sesuai dengan yang dibutuhkan oleh pemakai sistem.

Pemrograman Terstruktur

Pemrograman terstruktur adalah suatu tindakan untuk mengorganisasikan dan membuat kode-kode program supaya program mudah untuk dimengerti dan dimodifikasi.

C. *Visual Basic 6.0*

Menurut Alam (2005:2) MS-Visual Basic 6.0 for Windows adalah bahasa pemrograman yang bekerja dalam lingkup MS-Windows. VB dapat memanfaatkan kemampuan MS. Windows secara optimal. VB versi 6 sendiri di Indonesia masih banyak digemari programmer, karena tidak membutuhkan persyaratan hardware yang berat, paketnya cukup ringkas, program mudah dipakai dan beberapa alasan lainnya.

D. *Konsep Dasar SQL (Structured Query Language)*

Menurut Paryudi (2006,2010:107) SQL adalah bahasa komputer standar ANSI (*American National Standard Institute*) untuk mengakses dan memanipulasi sistem basis data. Pernyataan *SQL* digunakan untuk mengambil dan meng-*update* data dalam basis data. *SQL* digunakan untuk mengambil dan mengupdate data dalam basis data.

E. *Unified Modelling Language (UML)*

Menurut Widodo dan Herlawati (2011:6) *UML* singkatan dari *Unified Modeling Language* yang berarti bahasa pemodelan standar. (Chonoles, 2003: bab 1) mengatakan sebagai bahasa, berarti *UML* memiliki sintaks dan semantik. Ketika kita membuat model menggunakan konsep *UML* ada aturan-aturan yang harus

diikuti. Bagaimana elemen pada model-model yang kita buat berhubungan satu dengan lainnya harus mengikuti standar yang ada. *UML* bukan hanya sekedar diagram, tetapi juga menceritakan konteksnya.

UML diaplikasikan untuk maksud tertentu, biasanya antara lain:

1. Merancang perangkat lunak.
2. Sarana komunikasi antara antara perangkat lunak dengan proses bisnis.
3. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
4. Mendokumentasi sistem yang ada, proses-proses dan organisasinya.

Blok pembangunan utama *UML* adalah diagram. Beberapa diagram ada yang rinci (jenis *timing diagram*) dan lainnya ada yang bersifat umum (misalnya diagram kelas). Para pengembang sistem berorientasi objek menggunakan bahasa model untuk menggambarkan, membangun dan mendokumentasikan sistem yang mereka rancang. *UML* memungkinkan para anggota team untuk bekerja sama dengan bahasa model yang sama dalam mengaplikasikan beragam sistem. Intinya *UML* merupakan alat komunikasi yang konsisten dalam mensupport para pengembang sistem saat ini.

Beberapa literatur menyebutkan bahwa UML menyediakan sembilan jenis diagram, yang lain menyebutkan delapan karena ada beberapa diagram yang digabung, misalnya diagram komunikasi, diagram urutan dan diagram pewaktuan digabung menjadi diagram interaksi. Namun demikian model-model itu dapat dikelompokkan berdasarkan sifatnya yaitu statis atau dinamis. Jenis diagram itu antara lain:

1. Digaram kelas. Bersifat statis. Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, serta relasi-relasi. Diagram ini umum dijumpai pada pemodelan sistem berorientasi objek. Meskipun bersifat statis, sering pula diagram kelas memuat kelas-kelas-kelas aktif.
2. Diagram Paket (*Package Diagram*). Bersifat statis. Diagram ini memperlihatkan kumpulan kelas-kelas, merupakan bagian dari diagram komponen.
3. Diagram *Use-Case*. Bersifat statis. Diagram ini memperlihatkan himpunan *use-case* dan aktor-aktor (suatu jenis khusus dari kelas). Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku suatu sistem yang di butuhkan serta di harapkan pengguna.
4. Diagram Interaksi dan *Sequence* (urutan). Bersifat dinamis. Diagram urutan adalah diagram interaksi yang menekankan pada pengiriman pesan dalam suatu waktu tertentu.
5. Diagram Komunikasi (*Communication Diagram*). Bersifat dinamis. Diagram sebagai pengganti diagram kolaborasi *UML* 1.4 yang menekankan organisasi struktural dari objek-objek yang menerima serta mengirim pesan.
6. Diagram *Statechart* (*Statechart Diagram*). Bersifat dinamis. Diagram status memperlihatkan keadaan-keadaan pada sistem, memuat status (*state*), transisi, kejadian serta aktifitas. Diagram ini terutama penting untuk memperlihatkan sifat dinamis dari antarmuka (*interface*), kelas,

kolaborasi dan terutama penting pada pemodelan sistem-sistem yang reaktif.

7. Diagram Aktivitas (*Activity Diagram*). Bersifat dinamis. Diagram aktifitas adalah tipe khusus dari diagram status yang memperlihatkan aliran dari suatu aktivitas ke aktivitas lainnya dalam suatu sistem. Diagram ini terutama penting dalam pemodelan fungsi-fungsi suatu sistem dan memberi tekanan pada aliran kendali antar objek.
8. Diagram Komponen (*Component Diagram*). Bersifat statis. Diagram komponen ini memperlihatkan organisasi serta kebergantungan sistem/perangkat lunak pada komponen-komponen yang telah ada sebelumnya. Diagram ini berhubungan dengan diagram kelas dimana komponen secara tipikal di petakan ke dalam satu atau lebih kelas-kelas, antarmuka-antarmuka serta kolaborasi-kolaborasi.
9. Diagram *Deployment* (*Deployment Diagram*). Bersifat statis. Diagram ini memperlihatkan konfigurasi saat aplikasi di jalankan (*run-time*). Memuat simpul-simpul beserta komponen-komponen yang ada di dalamnya. Diagram *deployment* berhubungan erat dengan diagram komponen dimana diagram ini memuat satu atau lebih komponen-komponen. Diagram ini sangat berguna saat aplikasi kita berlaku sebagai aplikasi yang di jalankan pada banyak mesin (*distributed computing*).

Kesembilan diagram ini tidak mutlak harus digunakan dalam pengembangan perangkat lunak, semuanya dibuat sesuai dengan kebutuhan. Pada *UML* dimungkinkan kita menggunakan diagram-diagram

lainnya (misalnya *Data Flow Diagram*, *Entity Relationship Diagram* dan sebagainya).

F. *Entity Relationship Diagram (ERD)*

Menurut Paryudi (2006,2010:67) *Entity Relationship Diagram* adalah alat pemodelan data utama dan akan membantu mengorganisasi data dalam suatu proyek kedalam entitas-entitas dan menentukan hubungan antar entitas.

G. *Logical Record Structure (LRS)*

Menurut Frieyadie (2007:13) LRS merupakan hasil dari pemodelan *Entity Relationship (ER)* beserta atributnya sehingga bisa terlihat hubungan-hubungan antar entitas.

Dalam pembuatan *LRS* terdapat tiga hal yang dapat mempengaruhi, yaitu:

1. Jika tingkat hubungan (*cardinality*) satu pada satu (*one-to-one*), maka digabungkan dengan entitas yang lebih kuat (*strong entity*), atau digabungkan dengan entitas yang memiliki atribut yang lebih sedikit.
2. Jika tingkat hubungan (*cardinality*) satu pada banyak (*one-to-many*), maka hubungan relasi atau digabungkan dengan entitas yang tingkat hubungannya banyak.
3. Jika tingkat hubungan (*cardinality*) banyak pada banyak (*many-to many*), maka hubungan relasi tidak akan digabungkan dengan entitas manapun, melainkan menjadi sebuah *LRS*.

2.2 Penelitian Terkait

Menurut Agustina (2015:7) menyimpulkan bahwa:

Di **Counter HP** yang beralamat Jl.Raya Pacitan-Lorok Km.16 Ketro Desa Ketro Kecamatan Kebonagung, merupakan bidang usaha bergerak di bidang jual beli HP, *acesories* dimana proses pengolahan data jual beli dan data persediaan barang masih menggunakan proses konvensional atau tanpa terkomputerisasi sehingga menimbulkan beberapa masalah, diantaranya ketidaktepatan pemeriksaan barang yang masih tersedia, sering terjadi persediaan barang yang tidak terkontrol, pencarian data barang dan data penjualan, pembuatan struk penjualan, pembuatan laporan yang masih relatif lama.

Menurut Indah (2013:124) menyimpulkan bahwa:

Suatu bidang usaha harus dapat mengenalkan diri dan produk-produk yang mereka berjualan sehingga konsumen yang belum tahu akan mengenalnya, yang sudah mengetahui akan semakin percaya sehingga akan mudah mendapatkan perhatian dari masyarakat. Bisnis merupakan suatu kegiatan usaha individu yang terorganisasi untuk menghasilkan dana menjual barang ataupun jasa agar mendapatkan keuntungan dalam pemenuhan kebutuhan masyarakat dan ada di dalam industri. (Allan Afuah, 2004: 4). Sedangkan penjualan sendiri merupakan proses dimana sang penjual memuaskan segala kebutuhan dan keinginan pembeli agar dicapai manfaat baik bagi sang penjual maupun sang pembeli yang berkelanjutan dan yang menguntungkan kedua belah pihak. (Winardi, 1991 :2).