

BAB II

LANDASAN TEORI

2.1. Konsep Dasar Sistem

Pendekatan sistem yang merupakan kumpulan-kumpulan dari elemen-elemen atau komponen-komponen atau subsistem-subsistem merupakan definisi yang lebih luas. Definisi ini lebih banyak diterima, karna kenyataannya suatu sistem dapat terdiri dari beberapa subsistem atau sistem-sistem bagian. Terdapat dua kelompok pendekatan didalam mendefinisikan sistem, yaitu yang menekankan pada prosedurnya dan yang menekankan pada komponen atau elemennya. Komponen-komponen atau subsistem-subsistem saling berinteraksi dan saling berhubungan membentuk satu kesatuan sehingga tujuan atau sasaran sistem tersebut dapat tercapai. Pendekatan sistem yang menekankan pada komponen akan lebih mudah didalam mempelajari suatu sistem untuk tujuan analisis dan perancangan suatu sistem.

A. Sistem

1) Pengertian Sistem

Menurut (Mulyani 2016:2) mengatakan bahwa “Sistem dapat diartikan sebagai sekumpulan subsistem, komponen, ataupun elemen yang saling bekerjasama dengan tujuan yang sama untuk menghasilkan output yang sudah ditentukan sebelumnya”.

Menurut (Anggrarni & Irviani, 2017:1) Mengatakan bahwa “Sistem adalah kumpulan orang yang saling bekerjasama dengan ketentuan-ketentuan aturan yang

sistematis dan terstruktur untuk membentuk satu kesatuan yang melaksanakan suatu fungsi untuk mencapai tujuan”.

Berdasarkan pendapat dari para ahli diatas, dapat disimpulkan bahwa sistem merupakan komponen atau prosedur yang saling bekerja sama dengan satu tujuan untuk memperoleh hasil yang telah ditentukan.

2) Sistem Informasi

Menurut (Anggrani dan Irviani 2017:2) “Sistem Informasi merupakan suatu kombinasi teratur dari orang-orang *hardware*, *software*, jaringan komunikasi dan sumber daya data yang mengumpulkan, mengubah, dan menyebarkan informasi dalam sebuah organisasi”.

3) Sistem Berorientasi Objek (OOP)

Menurut(Sukanto & Shalahuddin, 2015:100) menyimpulkan bahwa : Metodologi berorientasi objek adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya. Metodologi berorientasi objek merupakan suatu cara bagaimana sistem perangkat lunak di bangun melalui pendekatan objek secara sistematis. Metode berorientasi objek meliputi rangkaian aktivitas analisis berorientasi objek, perancangan berorientasi objek, pemrograman berorientasi objek. Dan pengujian berorientasi objek.

B. Program

1) Visual Basic. NET

Menurut (Kurniawan, 2013:10) mengatakan bahwa : Visual Basic. NET adalah salah satu bahasa pemrograman tingkat tinggi yang mendekati bahasa manusia. Kemunculan bahasa VB.NET ini sebagai jawaban untuk menyederhanakan bahasa pemrograman pada *platform* NET yang diluncurkan tahun 2002 dan untuk menjembatani

programmer Visual Basic. Bahasa VB.NET secara teknis mengadopsi sintak bahasa Visual Basic. Konsistensi API membuat bahasa VB.NET menjadi pilihan dalam membuat kode program diatas platform Windows. Fitur baru bahasa VB.NET dibandingkan Visual Basic bahwa bahasa VB.NET mendukung object-oriented dan juga Dynamics programming. Ini menambah daftar kemudahan untuk belajar bahasa VB.NET.

2) *Crystal Report*

Menurut (Anggrawan dan Syahrir 2017:119) “Crystal Report merupakan hasil cetakan, membuat laporan dengan crystal report lebih mudah, karna pada crystal report banyak tersedia objek-objek maupun komponen yang mudah digunakan”.

C. **Basis Data**

Menurut (Yanto, 2016:10) menyatakan bahwa: Basis data terdiri dari 2 kata, yaitu basis dan data. Basis dapat diartikan sebagai markas, gudang, tempat berkumpul. Sedangkan data adalah fakta yang mewakili suatu objek seperti manusia, barang, hewan, peristiwa, keadaan dan sebagainya, yang direkam dalam bentuk angka, huruf simbol, teks gambar, bunyi atau kombinasinya

Menurut (Tufik, 2016:1) “Microsoft Acces adalah program database bagian dari microsoft office profesional yang diluncurkan oleh microfostr corporation”.

Menurut (Tim EMS, 2014:1) “Microsoft Acces adalah program aplikasi basis data komputer relasional yang ditujukan untuk kalangan rumahan dan perusahaan kecil hingga menengah.”

D. **Model Pengembangan Perangkat Lunak**

Model prototipe cocok digunakan untuk menjabarkan kebutuhan pelanggan secara lebih detail karena pelanggan sering kali kesulitan

menyampaikan kebutuhannya secara detail tanpa melihat gambaran yang jelas. Untuk mengantisipasi agar proyek dapat berjalan sesuai dengan target waktu dan biaya di awal, maka sebaiknya spesifikasi kebutuhan sistem harus sudah disepakati oleh pengembang dengan pelanggan secara tertulis. Dokumen tersebut akan menjadi patokan agar spesifikasi kebutuhan sistem masih dalam ruang lingkup proyek.

Menurut (Sukanto & Shalahuddin, 2015:31) Model prototipe (*prototyping model*) dimulai dari mengumpulkan kebutuhan pelanggan terhadap perangkat lunak yang akan dibuat. Lalu dibuatlah program prototipe agar pelanggan lebih terbayang dengan apa yang sebenarnya diinginkan. Program prototipe biasanya merupakan program yang belum jadi. Program ini biasanya menyediakan tampilan dan simulasi alur perangkat lunak sehingga tampak seperti perangkat lunak yang sudah jadi. Program prototipe ini dievaluasi oleh pelanggan atau *user* sampai ditemukan spesifikasi yang sesuai dengan keinginan pelanggan atau *user*.

2.2. Teori Pendukung

A. *Entity Relationship Diagram (ERD)*

Menurut (Mulyani 2016:100) *Entity Relationship Diagram (ERD)* digunakan untuk menggambarkan hubungan antara penyimpanan atau data *storage* yang dapat pada DFD, ERD menggunakan sejumlah notasi untuk menggambarkan struktur dan hubungan antar data.

Menurut (Sukanto & Shalahuddin, 2015:50) “Pemodelan awal basis data yang paling banyak digunakan adalah menggunakan *Entity Relationship Diagram (ERD)*”. ERD digunakan untuk pemodelan basis data relational. Sehingga jika penyimpanan basis data menggunakan OODBMS maka perancangan basis data tidak perlu menggunakan ERD

B. *Logical Relational Structure (LRS)*

Menurut (Tabrani, 2014:35) “*Logical Record Structure* terdiri dari link-link diantara tipe record, Link ini menunjukkan arah dari satu *tipe record* lainnya. Banyak link dari LRS yang diberi tanda *field-field* yang kelihatan pada kedua link *tipe record*.”

C. *Unified Modelling Language (UML)*

Menurut (Sukanto & Shalahuddin, 2015:133) menyimpulkan bahwa : *Unified Modeling Language (UML)* adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan *teks-teks* pendukung.

Menurut (Anggoro et al. 2015) menyimpulkan bahwa : *Unified Modelling Language (UML)* adalah sebuah bahasa berdasarkan grafik/gambar untuk memvisualisasi, menspesifikasikan, membangun dan pendokumentasian dari sebuah sistem pengembangan software berbasis OO (*Object Oriented*) UML tidak hanya merupakan sebuah bahasa pemrograman visual saja, namun juga dapat secara langsung dihubungkan ke berbagai bahasa pemrograman, seperti JAVA, C++, Visual Basic, atau bahkan dihubungkan secara langsung ke dalam sebuah *objectoriented database*.

1) *Activity Diagram*

Menurut (Sukanto & Shalahuddin, 2015:5161), mengatakan bahwa : diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu di perhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh system”.

2) *Use Case Diagram*

Menurut (Sukanto & Shalahuddin, 2015:155) mengatakan bahwa : *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 di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

3) *Class Diagram*

Menurut (Sukamto & Shalahuddin, 2015:141), “diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem”.

Kelas memiliki apa yang disebut atribut dan method atau operasi. Berikut penjelasan atribut dan method :

- Atribut merupakan *variable*-variabel yang dimiliki oleh suatu kelas.
- Operasi atau *method* adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

4) *Sequence Diagram*

Menurut (Sukamto & Shalahuddin, 2015:165), Menyimpulkan bahwa : diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dengan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.

Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case*. Banyaknya diagram sekuen yang harus digambar adalah minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup dalam diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak.