

BAB II

LANDASAN TEORI

2.1. Tinjauan Pustaka

A. Konsep Dasar Sistem Informasi

Model pengembangan sistem yang penulis gunakan dalam skripsi ini adalah *waterfall*. Menurut Pressman (2010:39) “Waterfall adalah model klasik yang bersifat sistematis, berurutan dalam membangun software”.

Kelebihan dari model ini adalah selain karena mengaplikasikan menggunakan model ini mudah, kelebihan dari model adalah ketika semua kebutuhan sistem dapat didefinisikan secara utuh, eksplisit dan benar diawal proyek maka *Software Engineering* (SE) dapat berjalan dengan baik dan tanpa masalah meskipun sering kali kebutuhan sistem dapat didefinisikan seeksplisit yang diinginkan tetapi paling tidak, problem pada kebutuhan sistem diawal proyek lebih ekonomis dalam haluang atau lebih murah, usaha dan waktu yang terbuang sedikit jika dibandingkan problem yang muncul pada tahap-tahap selanjutnya.

Kekurangan yang utama dari model ini adalah kesulitan dalam mengakomodasi perubahan setelah proses dijalani. *Fase* sebelumnya harus lengkap dan selesai sebelum mengerjakan *fase* berikutnya.

Masalah dengan *waterfall* :

1. Perubahan sulit dilakukan karena sifatnya yang kaku.
2. Karena sifat kakunya model ini cocok ketika kebutuhan dikumpulkan secara lengkap sehingga perubahan bisa ditekan sekecil mungkin. Tapi

pada kenyataannya jarang sekali konsumen/pengguna yang bisa memberikan kebutuhan secara lengkap, perubahan kebutuhan adalah sesuatu yang wajar terjadi.

3. *Waterfall* pada umumnya digunakan untuk rekayasa sistem yang besar yaitu dengan proyek yang dikerjakan beberapa tempat berbeda dan dibagi menjadi beberapa sub proyek.

B. Konsep Dasar Pemrograman

Menurut Rosa dan Shalahuddin (2013:67) “Konsep atau paradigma atau sudut pandang pemrograman yang membagi-bagi program berdasarkan fungsi-fungsi atau prosedur-prosedur yang dibutuhkan program komputer”. Modul-modul (pembagian program) biasanya dibuat dengan mengelompokkan fungsi-fungsi dan prosedur-prosedur yang diperlukan sebuah proses tertentu. Fungsi-fungsi dan prosedur-prosedur ditulis secara sekuensial atau terurut dari atas ke bawah sesuai dengan kebergantungan antar fungsi atau prosedur (fungsi atau prosedur yang dapat dipakai oleh fungsi prosedur dibawahnya harus yang sudah ditulis atau dideklarasikan di atasnya).

Tujuan teknik pemrograman dalam pemrograman terstruktur adalah :

- a. Meningkatkan kehandalan program
- b. Agar program mudah dibaca dan ditelusuri
- c. Menyederhanakan kerumitan dan pemeliharaan program
- d. Meningkatkan produktivitas pemogram Standar teknik pemecahan

masalah dalam pemrograman terstruktur adalah sebagai berikut:

1. **Modular**

Dalam pemrograman secara modular, suatu program akan dipilah ke dalam sejumlah modul, dimana setiap modul menjalankan fungsinya sendiri. Tentunya fungsi yang dijalankan oleh setiap modul sangat terbatas sesuai dengan ruang lingkup yang akan dikerjakan. Dengan adanya sejumlah modul program ini tentu saja kesalahan yang timbul dapat dikurangi. Setiap program tentu akan memiliki program utamanya, yang kemudian akan mengambil sejumlah modul-modul yang ada.

2. ***Top Down Analisis***

Menurut Rosa dan Shalahuddin (2013:279) “Pengujian *top-down integration* atau intregasi dari atas kebawah merupakan pendekatan bertahap (*incremental*) untuk mengonstruksi struktur program. Modul diintegrasikan berdasarkan hirarki, dimulai dari modul yang lebih besar lalu di dekomposisi ke modul yang lebih kecil”.

3. ***Buttom-Up***

Menurut Rosa dan Shalahuddin (2013:281) “Pengujian integrasi dari bawah keatas (*buttom-up integration*) memulai pengujian dari modul yang paling kecil ke modul yang lebih besar. Pengujian dari bawah ke atas sering dilakukan untuk pengembangan perangkat lunak yang tidak menggunakan alur prototype sehingga perangkat lunak dibangun dari modul-modul yang besar sesuai dengan hirarki pemakaiannya”.

C. UML (*Unified Modeling Language*)

Menurut Rosa dan Shalahuddin (2013:133) “UML (*Unified Modeling Language*) adalah standar bahasa yang banyak digunakan di dunia instruksi untuk mendefinisikan *requirement*, membuat analisis & desain, serta menggambarkan arsitektur dalam 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. UML hanya berfungsi untuk melakukan pemodelan. Jadi pengguna UML tidak terbatas pada metodologi berorientasi objek.

UML (*Unified Modeling Language*) untuk pemrograman terstruktur mendefinisikan diagram-diagram sebagai berikut :

1. *Use Case Diagram*

Menurut Rosa dan Shalahuddin (2013 : 155)” *Use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) 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.”

2. *Activity Diagram*

Menurut Rosa dan Shalahuddin (2013 : 161) “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 diperhatikan disini adalah bahwa diagram aktivitas

menggambarkan aktivitas sistem bukan apa yang dilakukan actor, jadi aktivitas yang dapat dilakukan oleh sistem.”

3. *Component Diagram*

Menurut Rosa dan Shalahuddin (2013 :148) “Diagram komponen atau *component* diagram dibuat untuk menunjukkan organisasi dan ketergantungan diantara kumpulan komponen dalam sebuah sistem. Diagram komponen focus pada komponen sistem yang dibutuhkan dan ada di dalam sistem.”

4. *Diagram Deployment (Deployment Diagram)*

Menurut Rosa dan Shalahuddin (2013 :154) “Diagram *deployment* atau *deployment* diagram menunjukkan konfigurasi komponen dan proses eksekusi aplikasi.”

D. *ERD (Entity Relationship Diagram)*

Pada dasarnya “ERD (*Entity Relationship Diagram*) adalah sebuah diagram yang secara konseptual memetakan hubungan antar penyimpanan pada diagram DFD (*Data Flow Diagram*)” Whitten (2010:29). Dimana DFD digunakan untuk menggambarkan aliran data dalam sebuah sistem informasi yang dikembangkan menggunakan komputer. ERD ini digunakan untuk melakukan permodelan terhadap struktur data dan hubungannya. Penggunaan ERD ini dilakukan untuk mengurangi tingkat kerumitan penyusunan sebuah *database* yang baik.

Entity dapat berarti sebuah objek yang dapat dibedakan dengan objek lainnya. Dalam dunia database *entity* memiliki atribut yang menjelaskan karakteristik dari *entity* tersebut. Ada dua macam atribut yang dikenal dalam *entity* yaitu atribut yang berperan sebagai kunci primer dan atribut deskriptif.

Hal ini berarti setiap *entity* memiliki himpunan yang diperlukan sebuah *primarykey* untuk membedakan anggota-anggota dalam himpunan tersebut.

Atribut dalam ERD dapat memiliki sifat-sifat menurut Whitten (2010 :30) sebagai berikut :

1. Atomic, menggambarkan atribut berisi nilai yang spesifik dan tidak dapat dipecah lagi.
2. Multivalue, menandakan atribut bisa memiliki lebih dari satu nilai untuk tiap *entity* tertentu.
3. Composite, gabungan dari beberapa atribut yang bersifat atomic.

Ada beberapa derajat relasi yang dapat terjadi pada ERD menurut Whitten (2010 :31), yaitu :

1. *One to one (1-1)*, menggambarkan bahwa antara satu anggota *entity* A hanya dapat berhubungan dengan satu anggota *entity* B.
2. *One to many (1-N)*, menggambarkan bahwa satu anggota *entity* A dapat memiliki hubungan dengan lebih dari satu anggota *entity* B.
3. *Many to many (N-N)*, menggambarkan bahwa lebih dari satu anggota *entity* A dapat memiliki hubungan dengan lebih dari satu anggota *entity* B.

E. LRS (Logical Record Structure)

LRS (*Logical Record Structure*) Adalah representasi dari struktur record-record pada table-tabel yang terbentuk dari hasil antar himpunan entitas. Menentukan kardinalitas, jumlah *table* dan *Foreign Key* (FK).

Berikut contoh serta cara kerja dari LRS berdasarkan kardinalitas antar table :

1. One to one

Menunjukkan relasi dengan kardinalitas one to one karena 1 nasabah hanya bisa melakukan 1 pengajuan pembiayaan.

2. One to many

Menunjukkan relasi dengan kardinalitas 1 jabatan dapat dimiliki oleh banyak karyawan, akan tetapi karyawan hanya bisa mengakses atau memiliki satu jabatan saja.

3. Many to many

Menunjukkan relasi dengan kardinalitas 1 nasabah dapat melihat banyak produk dan 1 produk juga dapat diakses oleh banyak nasabah.

F. Black-Box Testing

Menurut Pressman (2010 : p495) *Black-Box testing* berfokus pada persyaratan fungsional perangkat lunak yang memungkinkan *engineers* untuk memperoleh set kondisi *input* yang sepenuhnya akan melaksanakan persyaratan fungsional untuk sebuah program. *Black-Box testing* berusaha untuk menemukan kesalahan dalam kategori berikut :

1. Fungsi yang tidak benar atau fungsi yang hilang
2. Kesalahan antarmuka
3. Kesalahan dalam struktur data atau akses *database* eksternal

4. Kesalahan perilaku (*behavior*) atau kesalahan kinerja
5. Inisialisasi dan pemutusan kesalahan

2.2 Penelitian Terkait

Adapun tinjauan jurnal untuk penelitian terkait yang penulis lakukan dalam pembuatan skripsi ini yaitu :

SMK Ciledug Al_Musaddadiyah Garut merupakan salah satu lembaga pendidikan yang bekerja untuk menciptakan murid-murid yang berahlakul karimah dan berprestasi berdasarkan iman da takwa. Untuk saat ini fasilitas yang disediakan oleh pihak sekolah sangat membantu siswa dalam proses belajar mereka. Namun disamping fasilitas yang lengkap, SMK Ciledug Almusaddadiyah Garut masih memiliki kelemahan yang cukup kompleks dalam pengelolaan data, khususnya pengelolaan informasi nilai siswa. Metodologi yang digunakan dalam perancangan sistem informasi ini menggunakan Unified Approach (UA), metodologi ini dapat digunakan sebagai metodologi untuk menganalisis dan merancang Sistem Informasi Akademik Nilai Siswa Berbasis Web di SMK Ciledug Al-Musaddadiyah. Hasil penelitian ini menunjukkan bahwa penggunaan sistem informasi akademik nilai siswa berbasis web telah mengakomodasi guru dan staf yang bertugas untuk dapat mengelola data-data nilai harian siswa, dengan mudah dapat diakses lewat internet. (Kurnia, dkk, 2012:1).

Sistem Informasi Akademik pada Sekolah Menengah Atas Negeri 1 Ngadirojo masih dilakukan secara manual. Dalam pengolahan data akademik masih konvensional dengan sistem pendataan yang sekarang dirasakan masih banyak kekurangan yang terjadi, dikarenakan sistem yang ada masih menggunakan lembaran kertas dan arsip sehingga dapat menyebabkan data-data yang ada mudah hilang ataupun rusak. Pengembangan sistem berupa pembuatan sistem informasi akademik berbasis web pada Sekolah Menengah Atas Negeri 1 Ngadirojo merupakan sistem yang memberikan informasi laporan siswa secara online yang berupa laporan siswa yang bersangkutan dengan berbasiskan web, sehingga membantu kecepatan dan kualitas dalam penyampaian informasi. Selain itu dengan berbasiskan web maka informasi data dapat diakses dengan waktu dan tempat yang tidak ditentukan. Pada sistem ini menu hanya dapat diakses oleh user dengan waktu dan tempat yang tidak ditentukan. Pada sistem ini menu hanya dapat diakses oleh user tertentu yaitu siswa, guru dan administrator. Dalam penelitian ini selain memaparkan kajian teori yang digunakan sebagai dasar penyusunan, juga akan dibahas mengenai perancangan sistem dan pembuatan sistem informasi. Sehingga dapat diharapkan diimplementasi dalam sebuah

produk yang akan memperbaiki segala kekurangan yang ada pada sistem lama. (Setiyawan, dkk, 2013:1).