

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

A. Konsep Dasar Model Pengembangan Sistem

Dalam konsep dasar pengembangan sistem, Penulis menggunakan model *Waterfall* yang sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*). Menurut Pressman (2010:39), salah satu model pengembangan sistem adalah dengan model *waterfall*. *Waterfall* model adalah model yang paling populer dan sering dianggap sebagai pendekatan klasik dalam daur hidup pengembangan sistem Model ini melakukan pendekatan secara sistematis dan urut mulai dari level kebutuhan system lalu menuju ketahap analisis, desain, coding, testing, dan support atau maintenance. Sesuai dengan namanya *Waterfall* (air terjun) maka tahapan dalam model ini disusun bertingkat, setiap tahap dalam model ini dilakukan berurutan,

Keterkaitan dan pengaruh antar tahap ini ada karena output sebuah tahapan dalam *waterfall* model merupakan input bagi tahap berikutnya, dengan demikian ketidak sempurnaan hasil pelaksanaan tahapan sebelumnya awal dari ketidak sempurnaan tahap berikutnya. Memperhatikan karakteristik ini, sangat penting bagitim pengembang perangkat lunak untuk secara bersama-sama melakukan analisa kebutuhan dan desain system sesempurna mungkin sebelum masuk ke

tahapan penulisan kode program atau *coding*. Secara garis besar metode waterfall mempunyai langkah-langkah sebagai berikut:

1. Analisa Kebutuhan (*Requirment Analysis*)

Menurut Pressman (2010:39), langkah ini merupakan analisa terhadap kebutuhan system. Pengumpulan data dalam tahap ini bisa melakukan sebuah penelitian, wawancara, atau *study literature*. Seorang system analis akan mengali informasi sebanyak-banyaknya dari user sehinggaa akan tercipta sebuah system computer yang bisa melakukan tugas-tugas yang diinginkan oleh *user* tersebut. Tahapan ini akan menghasilkan dokumen user requirement atau bisa dikatakan sebagai data yang berhubungan dengan keinginan user dalam pembuatan system. Dokumen ini lah yang akan menjadi acuan system analis untuk menterjemahkan ke dalam bahasa pemrograman.

2. Design

Menurut Pressman (2010:39), proses desain akan menerjemahkan syarat kebutuhan ke sebuah perancangan perangkat lunak yang dapat diperkirakan sebelum dibuat coding. Proses ini berfokus pada struktur data, arsitektur perangkat lunak, representasi, interface, dan detail (*algoritma*) procedural. Tahapan ini akan menghasilkan dokumen yang disebut *software requirement*. Dokumen inilah yang akan digunakan programmer untuk melakukan aktivitas pembuatan sistemnya.

3. Coding

Menurut Pressman (2010:39), merupakan penerjemahan design dalam bahasa yang bisa dikenali oleh komputer. Dilakukan oleh programmer yang akan

menterjemahkan transaksi yang diminta oleh *user*. Tahapan ini lah yang merupakan tahapan secara nyata dalam mengerjakan suatu system. Dalam artian penggunaan computer akan dimaksimalkan dalam tahapan ini.

4. Testing

Menurut Pressman (2010:39), setelah pengkodean selesai maka akan dilakukan testing terhadap sebuah system yang telah dibuat. Pengujian ini berfokus pada pengetesan perangkat lunak secara logic dan fungsional dan memastikan bahwa semua bagian telah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

5. Support atau Maintenance

Menurut Pressman (2010:39), pada tahapan ini tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah diberikan kepada user. Perubahan ini bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau maintenance dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada tanpa membuat perangkat lunak yang baru.

Dalam penggunaan metode waterfall ini ada beberapa keuntungan dan kekurangan seperti berikut :

1. Keuntungan metode waterfall

- a Kualitas dari system yang dihasilkan akan baik. Ini dikarenakan oleh pelaksanaanya dilakukan secara bertahap sehingga tidak terfokus pada tahapan tertentu.
- b Dokumen pengembangan system sangat terorganisir, karena setiap tahap fase harus terselesaikan dengan lengkap sebelum melangkah ke fase berikutnya. Jadi setiap fase atau tahapan akan mempunyai dokumen tertentu.
- c Metode ini masih lebih baik digunakan walaupun sudah tergolong kuno, dari pada menggunakan pendekatan lain. Selain itu, metode ini juga masih masuk akal jika kebutuhan sudah diketahui dengan baik.

2. Kelemahan Metode Waterfall

- a. Diperlukan manajemen yang baik, karena proses pengembangan tidak dapat dilakukan secara berulang sebelum terjadinya suatu produk.
- b. Kesalahan kecil akan menjadi masalah besar jika tidak diketahui dari awal pengembangan perangkat lunak yang berakibat pada tahapan selanjutnya tidak akan berjalan sesuai rencana.
- c. User sulit menyatakan kebutuhan eksplisit sehingga tidak dapat mengakomodasi ketidakpastian pada saat awal pengembangan perangkat lunak

- d. User harus sabar, karena pembuatan perangkat lunak akan dimulai ketika tahap desain sudah selesai. Sedangkan tahap sebelum desain bisa memakan waktu yang lama.

B Konsep Dasar Pemrograman

Dalam membuat sebuah konsep dasar pemrograman, penulis menggunakan konsep dasar pemrograman terstruktur. Menurut Yatini (2007:10) pemrograman terstruktur adalah suatu aktifitas pemrograman dengan memperhatikan urutan langkah-langkah perintah secara sistematis, logis , dan tersusun berdasarkan algoritma yang sederhana dan mudah dipahami. Untuk mengetahui lebih jauh lagi, penulis akan menjelaskan beberapa konsep dasar dalam metodologi pemrograman terstruktur yang harus dipahami :

- a. Modular

Merupakan suatu metode untuk mendesain perangkat lunak atau *software* dengan cara memecah komponen-komponen suatu program yang besar menjadi bagian-bagian yang lebih kecil. Bagian-bagian tersebut, atau disebut dengan “module”, dapat dikembangkan, diuji, dan diperhalus secara independen.

- b. *Top down analysis*

Top down analysis adalah suatu teknik dimana masalah yang kompleks dibagi-bagi kedalam beberapa tingkatan kelompok masalah

hingga subbagian yang paling kecil, kemudian disusun langkah-langkah untuk menyelesaikannya secara detail

c. *Bottom up*

Pemecahan masalah dilakukan dengan menggabungkan prosedur-prosedur yang ada menjadi satu kesatuan program guna menyelesaikan masalah tersebut.

C Peralatan Pendukung Sistem (*Tools System*)

Merupakan alat yang digunakan untuk menggambarkan bentuk logika model dari suatu system dengan menggunakan symbol-simbol, lambang-lambang, diagram-diagram yang menunjukkan secara tepat arti dan fungsinya. Dalam hal ini penulis menggunakan *tools system* menggunakan UML (*Unified Modeling Language*) dan ERD (*Entity Relationship Diagram*)

1 UML (*Unified Modeling Language*)

Menurut Fowler (2005:1) Unified Modeling Language (UML) adalah keluarga notasi grafis yang didukung oleh meta-model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek (OO).. Namun demikian UML dapat digunakan untuk memahami dan mendokumentasikan setiap system informasi.

a. Use Case Diagram

Menurut Fowler (2005:141) Use Case adalah teknik untuk merekam persyaratan fungsional sebuah sistem. Use case mendeskripsikan

interaksi ripikal antara para pengguna sistem dengan sistem itu sendiri, dengan memberi sebuah narsi tentang bagaimana sistem tersebut digunakan. Menggambarkan fungsionalitas yang diterapkan dari sebuah system. Use case mendeskripsikan sebuah interaksi antara satu atau lebih actor dengan system informasi yang akan dibuat. Secara kasar, use case digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah system informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Syarat penamaan pada use case adalah nama didefinisikan sesimple mungkin dan dapat dipahami. Ada dua hal utama pada use case yaitu pendefinisian apa yangdisebut actor dan use case.

- 1) Actor merupaka orang, proses, atau system lain yang berinteraksi dengan system informasiyang akan dibuat diluar system informasi yang akan dibuat itu sendiri, jadi walaupun symbol dari actor adalah gambar orang, tapi actor belum tentu merupakan orang.
- 2) Use case merupakan fungsionalitas yang disediakan system sebagai unit-unit yang saling bertukar pesan antar unit atau actor.

b. Activity Diagram

Menurut Fowler (2005:163) Activity diagram adalah teknik untuk menggambarkan logika prosedural, proses bisnis, dan jalur kerja. Dalam beberapa hal, diagram ini memainkan peran mirip sebuah diagram alir, tetapi perbedaan prinsip antara diagram ini dan

notasi diagram alir adalah diagram ini mendukung behavior paralel. Diagram aktivitas atau activity diagram menggambarkan *workflow* (aliran kerja atau aktivitas dari sebuah system atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa apa yang dilakukan actor, jadi aktivitas yang dapat dilakukan oleh system. Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut :

- 1) Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis system yang didefinisikan
- 2) Urutan atau pengelompokan tampilan dari sistem atau *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.

c. *Deployment Diagram*

Menurut Fowler (2005:183) Diagram deployment atau deployment diagram menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi. Diagram deployment juga dapat digunakan untuk memodelkan hal-hal berikut :

- 1) Sistem tambahan (*embedded system*) yang menggambarkan rancangan device, node, hardware

- 2) Sistem client/server
- 3) Sistem terdistribusi murni
- 4) Rekayasa ulang aplikasi

d. *Component Diagram*

Menurut Fowler (2005:193) Diagram komponen atau component diagram dibuat untuk menunjukkan organisasi dan ketergantungan di antara kumpulan komponen dalam sebuah system. Diagram komponen fokus pada komponen system yang dibutuhkan dan ada di dalam system. Diagram komponen juga dapat digunakan untuk memodelkan hal-hal berikut :

- 1) *Source code* program perangkat lunak
- 2) Komponen *executable* yang dilepas ke user
- 3) Basis data secara fisik

2 ERD (*Entity Relationship Diagram*)

Basis data didefinisikan sebagai sekumpulan data yang saling terhubung secara logikal, yang dirancang dalam rangka memenuhi kebutuhan informasi dalam suatu organisasi (Connolly dan Begg, 2005, 14). ERD adalah suatu pemodelan dari basis data relasional yang didasarkan atas persepsi didalam dunia nyata, dunia ini terdiri dari sekumpulan objek yang saling berhubungan antara satu dengan yang lainnya. Suatu objek disebut *entity* dan hubungan yang dimilikinya disebut

relationship. Suatu *entity* bersifat unik dan memiliki attribute sebagai pembeda *entity* lainnya.

ERD merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. ERD untuk memodelkan struktur data dan hubungan antar data, untuk menggambarkan digunakan beberapa notasi dan simbol. Pada dasarnya ada tiga symbol yang digunakan yaitu :

a. Entity

Entity merupakan objek yang mewakili sesuatu yang nyata dan dapat dibedakan dari sesuatu yang lain. Symbol dari entity ini biasanya digambarkan dengan persegi panjang.

b. Atribut

Setiap entitas pasti mempunyai elemen yang disebut atribut yang berfungsi untuk mendeskripsikan karakteristik dari entitas tersebut. Isi dari atribut mempunyai sesuatu yang dapat mengidentifikasi isi elemensatu dengan yang lain. Gambar atribut diwakili oleh simbol elips.

Macam – macam atribut sebagai berikut :

- 1) *Simple attribute* yaitu atribut yang terdiri atas satu komponen tunggal dengan keberadaanya yang independen dan tidak dapat dibagi lagi ke komponen yang lebih kecil
- 2) *Composite attribute* yaitu atribut yang memiliki berbagai komponen dimana semua komponennya memiliki keberadaan yang independen

- 3) *Single value attribute* yaitu sebuah atribut yang mempunyai nilai tunggal untuk setiap kejadian
- 4) *Multi value attribute* yaitu sebuah atribut yang mempunyai beberapa nilai untuk setiap kejadian pada sebuah entitas
- 5) *Derived attribute* yaitu atribut yang memiliki nilai yang dihasilkan dari satu atau beberapa atribut lainnya, dan tidak harus dari satu entitas.

c. Hubungan / Relasi

Relasi didefinisikan sebagai hubungan yang terjadi antar *entity*. Representasi diagram relasi adalah sebuah garis lurus yang menghubungkan dua buah *entity*. Jenis-jenis atau hubungan yang biasa terjadi antar satu *entity* dengan *entity* yang lain dalam sebuah basis data, meliputi :

1) Derajat Relationship

Menjelaskan jumlah *entity* yang terlibat dalam suatu relationship, derajat relationship di bagi atas 3 yaitu:

a) *Unary* (Derajat Satu)

satu buah relationship menghubungkan satu buah *entity*.

b) *Binary* (Derajat Dua)

satu buah relationship yang menghubungkan dua buah *entity*.

c) *Ternary* (Derajat Tiga)

satu buah relationship menghubungkan tiga buah *entity*.

2) *cardinality ratio*

menjelaskan batasan pada jumlah entity yang berhubungan melalui suatu relationship. *cardinality ratio* dibagi atas 3 bagian yaitu:

a) One to One / Satu ke Satu (1:1)

Hubungan relasi satu ke satu yaitu setiap entitas pada himpunan entitas A berhubungan paling banyak dengan satu entitas pada entitas B.

b) One to Many / Satu ke Banyak (1:*).

Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, tetapi setiap entitas pada entitas B dapat berhubungan dengan satu entitas pada himpunan entitas A.

c) Many to Many / Banyak ke Banyak (*:*)

Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B.

3) *participation constrain*.

Menjelaskan apakah keberadaan suatu entity tergantung pada hubungannya dengan entity lain. *participation constrain* dibagi atas 2 bagian yaitu:

a) Total participation

keberadaan suatu entity tergantung pada hubungannya dengan entity lain. Di dalam diagram ER digambarkan dengan dua garis penghubung antara entity dengan relationship.

b) Partial participations

yaitu keberadaan suatu entity tidak tergantung pada hubungan dengan entity lain. Di dalam diagram ER digambarkan dengan satu garis penghubung antara entity dengan relationship.

2.2 Penelitian Terkait

Menurut Afandi,dkk (2013:1) Prima Game adalah sebuah toko yang melayani berbagai macam kebutuhan game dan terdapat banyak layanan-layanan untuk game. Dengan sistem yang masih bersifat konvensional, sangat kurang efektif dan kurang efisien dalam proses pengolahan data penjualan, rental dan servis game. Maka perlu adanya sistem informasi untuk lebih mengefektifkan proses pengolahan data di Prima Game Metode yang digunakan adalah : Pustaka, Observasi, Analisis, perancangan, Pengujian, Implementasi. Sistem yang dibangun diharapkan dapat lebih mengefektifkan proses pengolahan data penjualan, servis game dan mempermudah pengolahan data di Toko Prima Game yang sebelumnya masih menggunakan sistem konvensional menjadi terkomputerisasi.

Menurut Nugroho dan Siang (2015,50) Aturan pelayanan servis sepeda motor diawali dengan penyerahan sepeda motor konsumen pada petugas, kemudian konsumen menunggu hingga proses servis selesai. Permasalahan yang terjadi di bengkel Naga Mulya Motor adalah ruang tunggu yang berbeda dengan ruang servis. Akibatnya, pelanggan tidak mengetahui kegiatan yang dilakukan bengkel, dan pengelola bengkel kadang lupa memberi tahu pelanggan bahwa sepeda motornya dapat diambil karena telah selesai diservis. Pada penelitian ini dibuat sebuah sistem informasi untuk memvisualisasikan status sepeda motor yang diservis. Status sepeda motor berupa status antri, servis dan selesai. Program dibuat menggunakan jaringan wifi yang terdapat pada server laptop sehingga dapat menampilkan 2 keadaan pada layar yang berbeda. Layar utama digunakan sebagai server yang diakses oleh admin untuk melakukan proses administrasi servis. Layar kedua diletakkan di ruang tunggu yang digunakan untuk menampilkan status sepeda motor kepada pelanggan. Hasil penelitian berupa sebuah sistem yang dapat memvisualisasikan status sepeda motor yang sedang diservis sehingga pelanggan yang berada di ruang tunggu mengetahui apakah sepeda motor yang di servis telah selesai.