BAB II
LANDASAN TEORI

2.1 Rancang

Menurut Pressman (2009) perancangan atau rancang merupakan serangkaian prosedur untuk menterjemahkan hasil analisa dan sebuah sistem ke dalam bahasa pemrograman untuk mendeskripsikan dengan detail bagaimana komponen-komponen sistem di implementasikan.

2.2 Bangun

Menurut Pressman (2009) pengertian pembangunan atau bangun sistem adalah kegiatan menciptakan sistem baru maupun mengganti atau memperbaiki sistem yang telah ada secara keseluruhan

2.3 Pelayanan

Menurut koetler (2003)pelayanan (Service) ialah sebagai suatu tindakan ataupun kinerja yang bisa diberikan pada orang lain. Pelayanan atau juga lebih dikenal dengan service bisa di klasifikasikan menjadi dua yaitu.

2.3.1 High contact service

sebuah klasifikasi dari sebuah pelayanan jasa dimana kontak diantara konsumen dan juga penyedia jasa yang sangatlah tinggi, konsumen selalu terlibat di dalam sebuah proses dari layanan jasa tersebut.

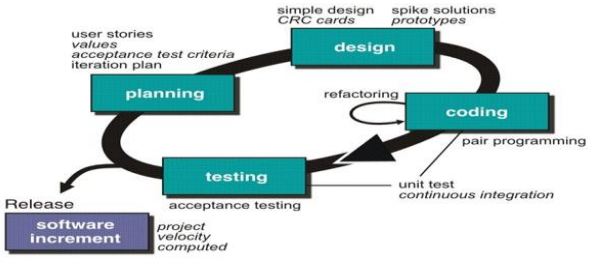
2.3.2 Low contact service

klasifikasi pelayanan jasa dimana kontak diantara konsumen dengan sebuah penyedia jasa tidaklah terlalu tinggi. *Physical contact* dengan konsumen hanyalah terjadi di front desk yang termasuk ke dalam klasifikasi low contact service. Misalkan ialah lembaga keuangan.

2.4 Pemesanan Pemesanan dapat diartikan sebagai proses, perbuatan, cara memesan atau memesankan. Pengertian lain dari pemesanan yaitu permintaan hendak membeli barang supaya dikirimkan, dibuatkan, dan sebagainya (Dendry Sugono, 2008)

2.5 *Extreme Programming*

Extreme Programming adalah metode pengembangan yang berasal dari turunan pengembangan *Agile SoftwareDevelopment*. *Agile SoftwareDevelopment* adalah teknik pengembangan yang dilakukan dengan cepat atau dalam arti memuaskan kebutuhan perangkat lunak atau sistem informasi yangmelibatkan pelanggan dengan tujuan meminimalkan kesalahan pengembang (Irawan, 2018),



Gambar 2.3 Tahapan Metode *Extreme Programming*(Irawan, 2018)

Pada tahap perancangan, *Extreme Programming* memiliki 4 tahap yang harus dikerjakan, *Extreme Programming* terdiri dari beberapa tahap yaitu sebagai berikut.

a. Perencanaan *(Planning)*

Tahap ini merupakan tahap awal dalam pembangunan sistem dimana dalam tahapan ini dilakukan beberapa kagiatan perencanaan yaitu, identifikasi permasalahan, menganalisa kebutuhan sampai dengan penetapan jadwal pelaksanaan pembangunan sistem.

b. Perancangan *(Design)*

Tahap berikutnya adalah perancangan, dimana pada tahapan ini dilakukan kegiatan pemodelan yang dimulai dari pemodelan sistem, pemodelan arsitektur sampai dengan basis data.

c. Pengkodean *(Coding)*

Tahapan ini merupakan kegiatan penerapan pemodelan yang sudah dibuat kedalam bentuk *user interface* dengan menggunakan bahasa pemograman

d. Pengujian *(Testing)*

Setelah tahapan pengkodean selesai, kemudian dilakukan tahapan pengujian sistem untuk mengetahui kesalahan apa saja yang timbul saat aplikasi sedang berjalan serta mengetahui apakah sistem yang dibangun sudah sesuai dengan kebutuhan pelanggan.

2.6 Pengertian Pemodelan Beriorentasi Objek

Object oriented programming adalah sebuah metode pemrograman dimana pengembang aplikasi tidak hanya mendefinisikan variabel yang berisi state dari sebuah struktur data, tetapi juga mendefinisikan fungsi untuk menunjukkan behavior yang diaplikasikan pada struktur data.

2.6.1 Ciri – Ciri Pemodelan Berorientasi Objek

Menurut Bambang Hariyanto dalam bukunya yang berjudul Rekayasa Perangkat Lunak Berorientasi Objek, bahwa *Object Oriented* memiliki ciri-ciri sebagai berikut:

1. Objek

   Bentuk baik yang nyata atau tidak, seperti manusia, hewan, benda, konsep, aliran, dan lain-lain. Objek merupakan inisiasi (turunan langsung) dari suatu kelas.

2. Kelas

   Kumpulan objek yang memiliki kemiripan perilaku (method), cirri atau karakteristik (*property*). Contoh objek orang dari kelas manusia, potongan sebagai berikut: Manusia orang1=*new* manusia(“Parto”); 24

3. Method

   Perilaku dari objek atau kelas tertentu. Merupakan perwujudan aksi atau tindakan dari dunia nyata di dalam pemrograman komputer.

4. Konstruktor

   Suatu fungsi yang dideklarasikan atau didefinisikan di dalam kelas, konstruktor harus mempunyai nama yang sama dengan fungsinya. Konstruktor dijalankan bersamaan dengan terciptanya kelas tersebut. Dalam suatu kelas bias terdapat lebih dari satu konstruktor. Konstruktor seperti method tetapi tidak mengembalikan nilai dan dapat didefinisikan tanpa parameter atau memakainya.

5. De-konstruktor

   Fungsi yang dideklarasikan dalam kelas, nama sama dengan nama fungsinya. Tetapi dijalankan bersamaan dengan dimusnahkannya kelas tersebut.

6. Karakteristik / properties

Ciri yang dimiliki oleh suatu objek, karakteristik ini juga sebagai pembeda objek satu dengan objek lainnya dalam kelas yang sama (konsep individu).

7. Variabel

   Tempat menampung data sementara, dalam pemrograman objek biasanya disebut data, sedangkan dalam pemrograman prosedural sering disebut dengan variabel.

8. Data

   Istilah lain dari variabel dalam OOP. Dalam pemrograman java bisa juga disebut *field*, data *member* atau *instance variable*.

9. Hak akses (*access attribute*)

   Hak akses digunakan untuk dapat menentukan data *member* mana yang dapat digunakan oleh kelas lain, dan mana yang tidak dapat digunakan. Hak akses ini sangat penting dalam membuat program turunan kelas.

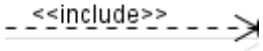
2.7 UML (*Unified Modeling Language*)

UML adalah:*Unified Modeling Language (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 pelangganan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek. (Rosa & Shalahuddin (2013:137)

2.8 Use Case Diagram

Merupakan pemodelan untuk melakukan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikansebuah 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. (Rosa A.S dan M. Shalahuddin, 2013).

Tabel 2.1 Simbol *Usecase Diagram* Sumber: (Rosa A.S dan M. Shalahuddin, 2013)

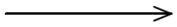
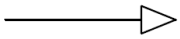
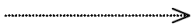
No	Simbol	Keterangan
1.	*UseCase* 	Fungsionalitas yang disediakan sistem sebagai unit-ubit yang saling bertukar pesan antara unit atau aktor.
2.	*Actor* 	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, walaupun simbol aktor adalah orang namun aktor belum tentu merupakan orang. Biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
3.	*Association*	Komunikasi antara aktor dan use case yang berpartisipasi pada usecase atau use case memiliki interaksi dengan aktor.
4.	Ekstensi/ *Extend* 	Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan dapat berdiri sendiri walaupun tanpa use case tambahan itu.
5.	*Generalization* 	Hubungan generalisasi dan spesialisasi (umum – khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
6.	*Include* 	Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsinya atau sebagai syarat dijalankan use case ini.

2.9 *Class Diagram*

Class Diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yangakan dibuat untuk membangun sistem. Kelas memilikiapa yang disebut atribut dan metode atau operasi. (Rosa A.S dan M. Shalahuddin, 2013).Sebuah kelas diagram terdiri dari sejumlah kelas yang dihubungkan dengan garis yang menunjukan hubungan antar kelas

.

Tabel 2.2 Simbol *Class Diagram* Sumber: (Rosa A.S dan M. Shalahuddin, 2013)

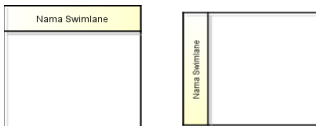
Simbol	Keterangan
Kelas/Class Nama_kelas +atribut +operasi()	Kelas pada struktur system
Antar muka/*interface* 	Sama dengan konsep *interface* dalam pemograman berorientasi objek
Asosiasi/*association* 	Relasi antar kelas dengan makna umum ,asosiasi biasanya juga disertai dengan *multiplicity*.
Asosiasi berarah/*directed* 	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan *multiplicity*.
Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
Kebergantungan atau *Dependency* 	Relasi antar kelas dengan makna kebergantungan antar kelas.
Agregasi/*aggregation* 	Relasi antar kelas dengan makna semua-bagian(whole-part)

2.10 *Activity Diagram*

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan kator,jadi aktivitas yang dapat dilakukan oleh sistem. (Rosa A.S dan M. Shalahuddin, 2013)

Berikut ini adalah simbol-simbol yang ada pada diagram aktivitas:

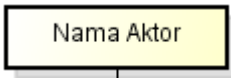
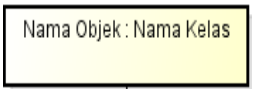
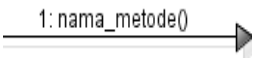
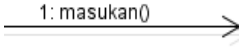
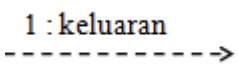
Tabel 2.3 Simbol *Activity Diagram* (Rosa A.S dan M. Shalahuddin, 2013

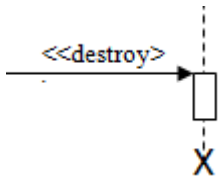
Simbol	Keterangan
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan/ *decision* 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
Penggabungan/ *join* 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
Status akhir 	Status akhir yang dilakukan sistem,sebuah diagram aktivitas memiliki sebuah status akhir
Swimlane 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

2.11 Sequence Diagram

Diagram sequence menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek atau message yang dikirimkan dan diterima antar objek.oleh karena itu untuk menggambarkan diagram *sequence* maka harus diketahui objek-objek yang terlibat dalam sebuah *usecase* berserta metode-metode yang dimiliki kelas.Berikut ini adalah simbol-simbol yang ada pada diagram aktivitas:

Tabel 2.4 Simbol *Sequence Diagram*. (Rosa A.S dan M. Shalahuddin, 2013)

Simbol	Keterangan
Aktor  atau  Tanpa waktu aktif	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;biasanya dinyatakan menggunakan kata benda diawal *frase* nama aktor.
Garis hidup 	Menyatakan kehidupan suatu objek
Objek  	Menyatukan objek yang berinteraksi pesan
Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya.
Pesan tipe *create*	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
Pesan tipe *call*  	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri.
Pesan tipe *send*  	Menyatakan bahwa suatu objek mengirim data /masukan/informasi ke objek lainnya , arah panah mengarah pada objek yang dikirim.
Pesan tipe *return*  	Menyatakan bahwa suatu objek yang telah menjalankan sesuatu operasi atau metode menghasilkan suatu

	kembalian ke objek tertentu.
san Tipe *Destroy*  	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada *create* maka ada *destroy*.

2.12 Penelitian Terdahulu

Penelitian terdahulu bertujuan untuk mendapatkan bahan perbandingan dan acuan Selain itu untuk menghindari anggapan kesamaan dengan penelitian lain,maka dalam Tinjaun pustaka ini peneliti mencantumkan hasil dari penelitian terdahulu sebagai Berikut:

1.SISTEM INFORMASI BOOKING SERVICE PADA CV.DWI JAYA OTOMATIF SAMPIT(2019)

2.RANCANG BANGUN APLIKASI WEB E-BOOKING DRIVER PADA BANK BUMN DISURABAYA(2020)

3.RANCANG BANGUN BOOKING SERVICE SYSTEM PADA BENGKEL RIA AUTO SMART BERBASIS WEBSITE(2017)

Tabel 2.5 Penelitian Terdahulu

Nama	Judul	Fitur	Metode analisis	Hasil
Muhammad kadapi	Rancang bangun booking service System pada Bengkel ria Auto smart Berbasis website	Membuat fitur layanan berbasis website dibengkel ria	Deskriptif dan kualitatif	Memberikan sarana informasi Alternatif pada costumer yang Akan melakukan service pada Bengkel ria
Dwi fatrianto suyatno	Rancang bangun aplikasi web E-Booking driver pada Bank BUMN disurabaya	Membuat fitur e-booking dibank bumn surabaya	Deskriptif dan kualitatif	Aplikasi ebooking driver Yang dirancang ini dapat membantu kebutuhan dalam proses ebooking
Adi surya Kurniawan	Sistem Informasi booking	Membuat fitur booking dicv.dwi jaya	Deskriptif dan kualitatif	Dengan adanya fitur booking

	service pada cv.dwi jaya otomatif sampit	otomotif		Service ini Pelanggan dapat melakukan pemesenan secara online