

BAB III

METODOLOGI PENELITIAN

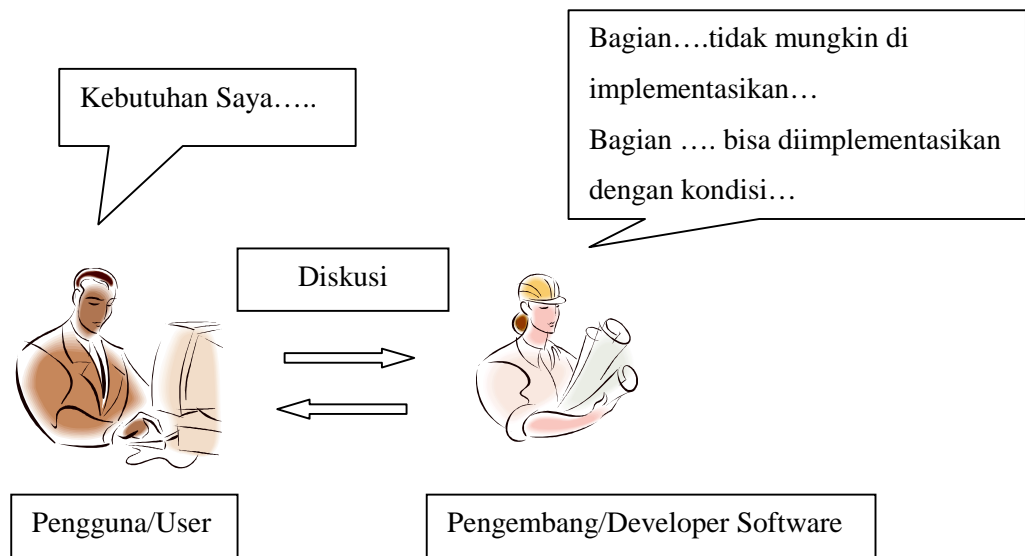
Metodologi penelitian adalah cara yang digunakan dalam memperoleh berbagai data untuk diproses menjadi informasi yang lebih akurat sesuai permasalahan yang akan diteliti. Metodologi penelitian digunakan sebagai pedoman dalam pelaksanaan penelitian agar hasil yang dicapai tidak menyimpang dari tujuan yang telah dilakukan sebelumnya. Adapun metode yang digunakan untuk analisa dan perancangan perangkat lunak adalah metode *Object Oriented Analysis Design* (OOAD).

3.1 Metode Object Oriented Analysis and Design (OOAD)

OOAD adalah metode pengembangan sistem yang lebih menekankan pada objek dibandingkan dengan data atau proses. Dalam Tahapannya OOAD terbagi menjadi dua yaitu OOA (*Object oriented analysis*) Dan OOD (*Object Oriented Design*).

3.1.1 OOA (*Object oriented analysis*)

Object oriented analysis (OOA) merupakan metode analisis yang memeriksa *requirements* (syarat/ keperluan yang harus dipenuhi oleh sistem) dari sudut pandang kelas – kelas dan objek – objek yang ditemui dalam ruang lingkup permasalahan. OOA mempelajari permasalahan dengan menspesifikasikannya atau mengobservasi permasalahan tersebut dengan menggunakan metode berorientasi objek. Biasanya analisa sistem dimulai dengan adanya dokumen permintaan yang diperoleh dari semua pihak yang berkepentingan. Analisa ini sebaiknya dilakukan oleh orang-orang yang benar-benar memahami implementasi sistem yang berbasis atau berorientasi objek, karena tanpa pemahaman itu maka sistem yang dihasilkan bisa jadi tidak realistis jika di implementasikan dengan berbasis objek.



Gambar 3.1 Ilustrasi OOA

Adapun langkah-langkah dalam OOA adalah sebagai berikut :

1. Menganalisis masalah

Data-data yang diperlukan untuk pembangunan suatu sistem dikumpulkan sebagai kebutuhan sistem. Setelah semua data-data terkumpul, penulis melakukan analisa untuk merumuskan permasalahan yang terjadi. Kemudian penulis menganalisa dan menggambarkan aliran sistem yang lama dari data yang telah didapat sebelumnya. Berdasarkan data yang didapat dan analisa dari aliran sistem lama ini, penulis kemudian menganalisa dan menggambarkan aliran sistem baru yang nantinya dapat memecahkan masalah yang sering terjadi ketika menggunakan sistem lama.

2. Menjelaskan proses yang terjadi dalam sistem

Fungsi dari sistem yang akan dibangun bersumber dari data-data yang diperlukan untuk pembangunan suatu sistem dikumpulkan digambarkan. Pada penelitian ini semua data-data yang diperlukan untuk membuat sistem digambarkan dalam bentuk rancangan analisis. Adapun rancangan analisis yang digunakan pada penelitian ini adalah *Usecase Diagram*, *Class Diagram*, dan *Sequence Diagram*.

3. Identifikasi Objek

Objek adalah benda yang secara fisik dan konseptual yang ada disekitar kita. Pada penelitian yang menjadi objeknya adalah semua penginapan di Pekanbaru, meliputi Hotel dan wisma.

4. Menentukan atribut

Atribut disebut juga dengan *class* yaitu definisi umum dari himpunan objek yang sejenis. Kelas menetapkan spesifikasi perilaku (*behaviour*) dan atribut-atribut dari objek tersebut. *Class* adalah abstraksi dari entitas dunia nyata. Pada penelitian ini yang menjadi atribut dari sebuah penginapan yaitu nama penginapan, alamat penginapan, koordinat penginapan, jenis kamar, harga kamar, dan fasilitas kamar

5. Mendefinisikan Operasi

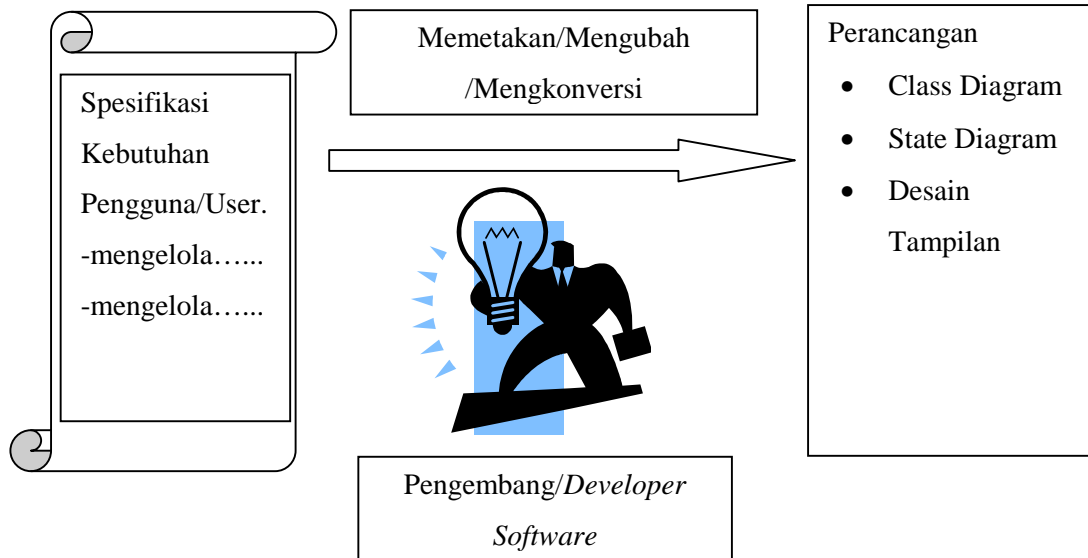
Maksudnya yaitu menjelaskan operasi yang memungkinkan bisa untuk di implementasikan dan yang tidak bisa di implementasikan. Pada penelitian ini yang dapat diimplementasikan meliputi pendaftaran penginapan, pengolahan penginapan, pengolahan kamar penginapan, pengolahan fasilitas kamar, dan pencarian penginapan berdasarkan metode MPE. Pada penelitian ini tidak dilengkapi dengan fitur jumlah kamar yang kosong dalam suatu penginapan dan fitur *booking* penginapan serta pembayaran secara *e-banking*.

3.1.2 *Object Oriented Design* (OOD)

Object Oriented Design (OOD) merupakan metode untuk mengarahkan arsitektur *software* yang didasarkan pada manipulasi objek – objek sistem atau subsistem. OOD adalah sebuah metode mendesain yang mencakup proses pendekomposisi objek dan digambarkan dalam notasi sehingga bisa menggambarkan *static* (*class diagram*) dan *dynamic* (*statechart diagram*) model sistem.

OOD memungkinkan *software engineer* untuk mengetahui object-object yang dihasilkan oleh tiap class dan hubungan antar *object*. Selain itu, OOD menggambarkan bagaimana hubungan antar *object* bisa dilakukan, bagaimana

behavior dari *object* diimplementasikan dan bagaimana komunikasi antar *object* diimplementasikan.



Gambar 3.2 Ilustrasi OOD

Pemodelan berorientasi objek biasanya dituangkan dalam dokumentasi perangkat lunak dengan menggunakan perangkat permodelan berorientasi objek, diantaranya adalah UML (*Unified Modeling Language*), Adapun tahanan dari *Object Oriented Design* (OOD) yaitu :

1. Desain Subsistem

Berisikan representasi masing-masing subsistem yang memungkinkan perangkat lunak mencapai persyaratan yang didefinisikan oleh pelanggannya dan untuk mengimplementasikan infrastruktur yang mendukung persyaratan pelanggan. Desain subsistem ini menggambarkan tabel-tabel yang digunakan dalam sistem. Adapun desain subsistem yang ada pada sistem ini meliputi tabel *member*, tabel admin, tabel penginapan, tabel galeri penginapan, tabel fasilitas dan tabel pasang fasilitas.

2. Desain Objek dan Kelas

Berisi hirarki kelas yang memungkinkan sistem diciptakan dengan menggunakan generalisasi dan spesialisasi yang ditarget secara perlahan. Lapisan ini juga berisi infrastruktur yang mendukung persyaratan pelanggan. Desain objek dan kelas ini meliputi gambaran relasi dari tiap-tiap kelas/objek yang ada pada sistem. Adapun desain objek dan kelas

pada penelitian ini meliputi tabel member yang berelasi dengan tabel penginapan (1:M), tabel penginapan berelasi dengan tabel fasilitas (M:M) dan tabel penginapan yang berelasi dengan tabel galeri (M:M).

3. Desain Pesan

Berisi *detail* yang memungkinkan masing-masing objek berkomunikasi dengan kolaboratornya. Lapisan ini membangun *interface internal* dan *eksternal* bagi sistem tersebut. Adapun desain pesan pada penelitian ini meliputi Rancangan Halaman *Home*, Rancangan Halaman Pencarian, Rancangan Halaman *Login*, Rancangan Halaman Tambah Penginapan, Rancangan Halaman Tambah Kamar, Rancangan Halaman Tambah Fasilitas Kamar, dan Rancangan Halaman Profil Penginapan.

3.2 Unified Modeling Language (UML)

Unified Modeling Language (UML) merupakan salah satu bentuk *language* atau bahasa, menurut pencetusnya UML di definisikan sebagai bahasa visual untuk menjelaskan, memberikan spesifikasi, merancang, membuat model, dan mendokumentasikan aspek-aspek dari sebuah sistem.

Definisi ini merupakan definisi yang sederhana. Pada kenyataannya, pendapat orang-orang tentang UML berbeda satu sama lain. Hal ini dikarenakan oleh sejarahnya sendiri dan oleh perbedaan persepsi tentang apa yang membuat sebuah proses rancang-bangun perangkat lunak efektif (Martin 2005:1).

Pada tahap analisis, meliputi usaha untuk mengetahui apa kemampuan sebuah sistem yang diinginkan pengguna dan pelanggan dari sebuah perangkat lunak. Beberapa teknik yang dapat membantu dalam tahapan analisis (Martin 2005:44) :

1. *Use case Diagram* adalah gambaran umum sistem dari sudut pandang pengguna sistem. Tujuan dari use case adalah untuk menggambarkan apa yang sistem dapat lakukan. Use case diagram dibentuk dari skenario tentang kegunaan sistem yang dinotasikan dengan sebuah use case. Setiap skenario menjelaskan suatu alur kegiatan. Setiap skenario dapat diinisialisasi oleh pengguna sistem atau yang disebut aktor.

2. *Class diagram* merupakan salah satu diagram struktur statis yang menggambarkan struktur dan hubungan antar kelas. *Class diagram* digunakan untuk mensimulasikan objek-objek dalam dunia nyata ke dalam sistem yang akan dibangun. Notasi UML pada *class diagram* adalah sebuah persegi yang dibagi menjadi 3 area, yaitu nama kelas, atribut, dan operasi (*method*). *Class diagram* dapat juga menggambarkan keanekaragaman (*multiplicity*), yaitu jumlah objek dari suatu kelas yang berhubungan dengan sebuah objek dari kelas yang berasosiasi.
3. *Sequence diagram* digunakan terutama untuk menunjukkan interaksi antar objek dalam urutan sekuensial. *Sequence diagram* sangat berguna untuk mengkomunikasikan bagaimana objek-objek berinteraksi dalam suatu proses bisnis. Analisis sistem umumnya menggunakan *sequence diagram* untuk memperjelas *use case*. *Sequence diagram* terdiri dari objek-objek yang digambarkan dengan sebuah persegi yang memiliki nama. Objek-objek tersebut diletakkan di atas dan diurutkan dari kiri ke kanan. Dari setiap objek, ada garis putus-putus memanjang ke bawah yang menggambarkan garis hidup (*Life line*) suatu objek. Di atas garis hidup tersebut, ada kotak kecil memanjang yang dinamakan aktivasi. Aktivasi merepresentasikan eksekusi dari operasi yang objek lakukan.