BAB III

METODELOGI PENELITIAN

3.1. Objek Penelitian

Penelitian dilakukan di Hotel Bukit Dago Bandung yang berlokasi di Jl. Ir H Juanda No. 311 Dago Bandung.

3.1.1. Sejarah Singkat Hotel Bukit Dago Bandung

Sejarah Hotel Bukit Dago Bandung berdiri pada tahun 1989, sebelumnya masih bernama Bukit Utara Guest House, kemudian berubah menjadi Bukit Utara Permai dengan jumlah kamar 20 unit, kemudian berubah menjadi Hotel Bukit Dago tahun 1995 sampai sekarang.

3.1.2. Visi dan Misi Hotel Bukit Dago Bandung

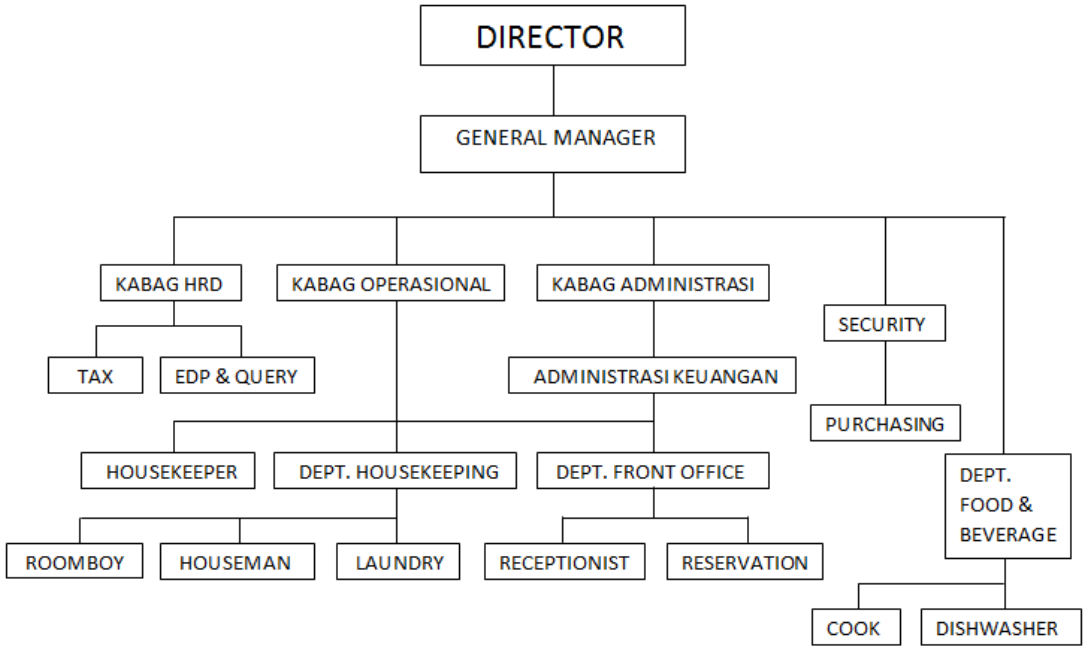
1. Visi Hotel Bukit Dago Bandung

   Mengembangkan bidang perhotelan itu sendiri dan mengembangkan usaha yang berhubungan dengan perkembangan industri pariwisata

2. Misi Hotel Bukit Dago Bandung

   1. Memberikan keuntungan pada pemegang saham.
   2. Menciptakan lapangan pekerjaan.
   3. Memberikan konstribusi asli bagi pendapatan asli daerah.
   4. Berpartisipasi dalam pengembangan industri pariwisata di Indonesia khususnya Jawa Barat.

3.1.3. Struktur Organisasi Hotel Bukit Dago Bandung



Gambar 3.1 Struktur Organisasi Hotel Bukit Dago Bandung

3.1.4 Job Descriptions

1. Direktur :

   Direktur merupakan orang yang mempunyai investasi terhadap hotel atau dalam hal pemilik modal

2. *General Manager* :

   Bertanggung jawab terhadap seluruh operasional hotel, serta bertanggung jawab terhadap total pendapatan hotel (*revenue*), total biaya yang dikeluarkan, dan pencapaian total penjualan. *General manager* hotel Bukit Dago Bandung memiliki wewenang mengkoordinir *department front office, housekeeping, laundry, food and beverafe* dan lain lain.

3. Kantor Bagian HRD :

   Bertanggung jawab terhadap proses *rekrutmen* karyawan baik dari dalam maupun dari luar hotel, seleksi, promosi, mutasi dan memenuhi kebutuhan karyawan bagi departemen-departemen.

4. Kantor Bagian Operasional :

   Mengorganisasikan, mendelegasikan dan memberikan tugas pekerjaan kepada bawahannya. Melakukan kontrol terhadap kualitas palayanan di semua bagian yang menjadi tanggung jawabnya. Mengkoordinasikan, menyelenggarakan pemeliharaan terhadap seluruh fasilitas pelayanan tamu hotel di semua bagian dan memastikan semua fasilitas dan kondisi baik dan siap pakai.

5. Kantor Bagian Administrasi :

   Bagian administrasi mempunyai tugas menyelenggarakan perencanaan, pelaksanaan, evaluasi, dan pelaporan di bidang administrasi akademik, perhotelan, kepegawaian, keuangnan, data dan informasi, dan umum.

6. *Tax* :

   Membuat laporan keuangan (GL, P&L, Jurnal) per bulan per perusahaan untuk kepentigan perpajakan *(tax).* Dan Membuat/mengisi SSP per bulan.

7. *EDP & Query* :

   Bertanggung jawab menangani kebutuhan selurug jaringan informasi di hotel Bukit Dago, dan membuat jaringan informasi yang menghubungkan satu departemen dengan departemen lainnnya.

8. Administrasi Keuangan :

   Bertanggung jawab atas segala aktifitas keuangan, tugas utama dari jabatan ini yatu melakukan pengaturan, transaksi, mebuat laporan keuangan perusahaan. Jabatan Administrasi keuangan harus melapor pada jabatan Manager Keuangan. Pada jabatan ini sangat dibutuhkan kedislipinan, kejujuran, ketelitian serta tanggung jawab yang tinggi karena jika terjadi kesalahan akan sangat fatal pada perusahaan karena menyangkut keuangan perusahaan

9. *Department Housekeeper* :

   Bertanggung jawab terhadap kebersihan seluruh area hotel, mulai dari kebersihan kamar sampai kebersihan segala fasilitas hotel, menyiapkan kamar yang siap pakai bagi tamu hotel, memberikan informasi kepada *front office*, dan berwenang mengatur *shift* kerja *housekeeper staff*.

10. *Housekeeping* :

    Menjalankan tugas yang telah ditetapkan dari *Department Housekeeping*, melayani tamu hotel, dan menyampaikan segala informasi kepada *Department Housekeeping*.

11. *Departement Front Office* :

    Petugas hotel yang bertanggung jawab atas pengelolaan kantor depan. Penerima tamu, memberikan informasi secara khusus dan umum. Menerima pesanan tamu baik langsung ataupu melalui telepon. Bertanggung jawab dalam proses tamu *check in* sampai dengan tamu *check out.*

12. *Receptionist* :

    Bertanggung jawab melayani tamu hotel pada saat proses *reservasi*, registrasi, memberikan informasi yang jelas kepada tamu hotel, dan bertugas sebagai operator telepon.

13. *Reservation* :

    Mengawasi, melatih, dan mengevaluasi jadwal Agen *Reservation*, memastikan bahwa semua prosedur dilakukan dengan standar hotel. Membantu di mana perlu untuk memastikan pelayanan yang optimal kepada para tamu, pengunjung dan lainnya departemen.

14. *Department Food & Beverage* :

    Bertugas untuk mengelola *Food and Beverage Department* untuk dapat menyajikan makanan dan minuman berkualitas yang disenangi tamu. Merencanakan menu, memastikan bahwa setiap bawahannya dapat menyajikan makanan dengan cepat dan ramah, dan mengendalikan biaya *Food and Beverage Department*.

15. *Cook* :

    Menjadi komando dalam menyajikan menu makanan.

16. *Dishwasher* :

    Mengantar pesanan makanan hotel dan mencuci perkakas dapur yang kotor.

17. *Security* :

    Petugas yang bertanggung jawab atas keamanan hotel secara keseluruhan. Ruang lingkup pengamanan hotel meliputi para karyawan, para tamu, dan aset hotel. Security perlu menciptakan kondisi yang aman sehingga tamu betah tinggal di hotel dan parakaryawan bekerja dengan nyaman.

18. *Purchasing* :

    Bertanggung jawab terhadap pengadaan atau pembelian seluruh perlengkapan hotel, seperti perlenkapan kamar, dan segala fasilitas hotel.

19. *Room boy* :

    Bertugas membersihkan ruangan kamar, mengantarkan tamu dari F.O menuju kamar, mencuci keperluan hotel seperti sprei, handuk dan lain lain, mengantar *breakfast* kekamar..

20. *House man* :

    Karyawan yang melakukan semua perawatan di belakang layar untuk memastikan kamar hotel da fasilitas yang bersih dan berfungsi dengan baik. *Houseman* bertugas membersihkan dan memelihara lobi, pintu

masuk dan lain lain. Membantu tmu dengan pelayanan umum dan pertanyaan,

21. *Laundry* :

    Bertugas untuk mencuci perlengkapan yang ada di kamar seperti sprei, bantal dan kasur handuk. Menyetrika sprei, selimut, sarung bantal. Menerima order mencuci dan menyetrika pakaian tamu.

3.2. Metode Penelitian

Metode penelitian adalah kegiatan yang dilakukan untuk memperoleh fakta-fakta dari suatu pengetahuan dengan cara mengumpulkan, mencatat, dan menganalisa data yang dikerjakan secara sistematis berdasarkan ilmu pengetahuan.

3.2.1. Desain Penelitian

Dalam melakukan penelitian perlu adanya metode penelitian, sehingga penelitian berjalan dengan baik. Oleh karena itu penulis memilih metode penelitian tindakan (*Action Research*).

Penelitian tindakan (*Action Research*) ialah penelitian yang bertujuan untuk mengembangkan keterampilan-keterampilan baru, cara pendekatan baru, atau produk pengetahuan yang baru dan untuk memecahkan masalah dengan penerapan langsung di dunia aktual atau lapangan (Narimawati, 2010).

3.2.2. Jenis dan Metode Pengumpulan Data

Data yang diperlukan dalam penelitian ini terdiri dari data primer dan data sekunder. Data primer yang diperlukan dalam penelitian ini dikumpulkan dengan melakukan penelitian lapangan yaitu datang langsung ke perusahaan yang bersangkutan. Data sekunder dikumpulkan melalui penelitian kepustakaan.

3.2.2.1. Sumber Data Primer

Sumber data primer yaitu sumber data yang berasal dari perorangan, kelompok, panel atau sumber terselubung. Dalam memperoleh data primer penulis melakukan wawancara, dan observasi

1. Pengamatan Langsung (Observasi)

   Yaitu dengan mengadakan tinjauan langsung ke Hotel Bukit Dago untuk melakukan pengamatan pada proses pemesanan kamar hotel.

2. Wawancara (Interview)

   Wawancara proses untuk mendapatkan data dengan melakukan dialog langsung kepada bagian *receptionist* yang dianggap bisa memberikan informasi yang dibutuhkan oleh penulis.

3.2.2.2. Sumber Data Sekunder

Sumber data sekunder yang terkumpul selama melakukan penelitian mengenai "Sistem Informasi Pemesanan Kamar Berbasis Android pada Hotel Bukit Dago Bandung" di dapat dari penyusunan melalui proses dokumentasi data tersebut meliputi:

1. Daftar kamar dan harga kamar Hotel Bukit Dago Bandung
2. Data-data Hotel Bukit Dago Bandung mengenai struktur organisasi hotel.
3. Data jumlah hunian kamar pertahun pada Hotel Bukit Dago Bandung.

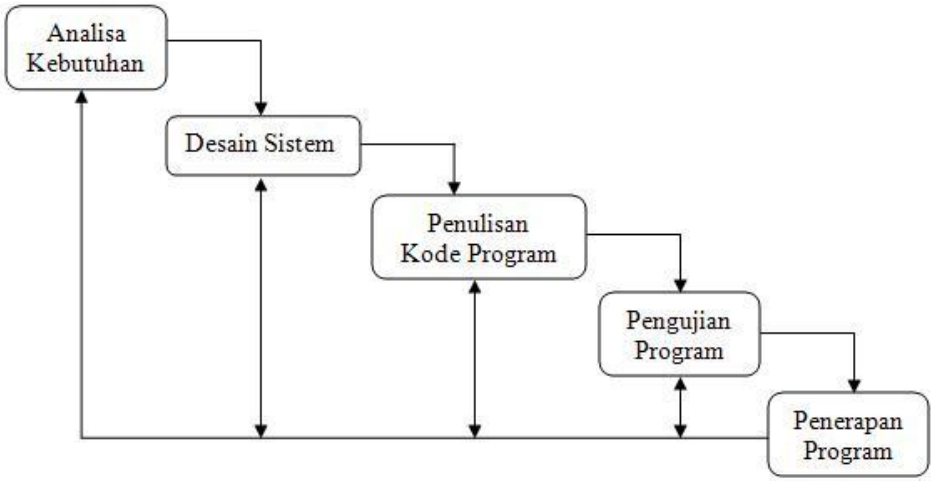
3.3 Metode Pendekatan Sistem

Metode pendekatan sistem yang di gunakan adalah pendekatan dengan *Object Oriented* yang menggunakan OOA (*Object Oriented Analysis*) di visualisasikan dengan UML dan di antara nya adalah sebagai berikut : *Use Case Diagram, Activity Diagram, Sequence Diagram, Class Diagram, Component Diagram* dan *Deployment Diagram*.

3.4 Metode Pengembangan Sistem

Menurut jogiyanto (2005 : 59) Metode pengembangan sistem adalah metode-metode, prosedur-prosedur, konsep-konsep pekerjaan yang akan digunakan untuk mengembangkan suatu sistem informasi.

Model *waterfall* mengusulkan sebuah pendekatan kepada perkembangan software yang sistematik dan sekuensial yang mulai pada tingkat dan kemajuan sistem pada seluruh analisis, desain, kode, pengujian, dan pemeliharaan. Model ini melingkupi aktivitas–aktivitas sebagai berikut : rekayasa dan pemodelan sistem/informasi, analisis kebutuhan, desain, *coding*, pemeliharaan dan pengujian.



Gambar 3.2 Metode Waterfall

Sumber: Roger S.Pressman (2002:37)

Tahapan penelitian pada model waterfall meliputi metodologi berupa :

1. Analisa

Proses pencarian kebutuhan diintensifkan dan difokuskan pada *software*. Untuk mengetahui sifat dari program yang akan dibuat, maka para *software* engineer harus mengerti tentang domain informasi dari *software*, misalnya fungsi yang dibutuhkan, *user interface*, dan sebagainya. Dari 2

aktivitas tersebut (pencarian kebutuhan sistem dan *software*) harus didokumentasikan dan ditunjukkan kepada pelanggan.

2. Desain

Proses ini digunakan untuk mengubah kebutuhan-kebutuhan diatas menjadi representasi ke dalam bentuk "*blueprint"* software sebelum coding dimulai. Desain harus dapat mengimplementasikan kebutuhan yang telah disebutkan pada tahap sebelumnya. Seperti 2 aktivitas sebelumnya, maka proses ini juga harus didokumentasikan sebagai konfigurasi dari *software*.

3. *Coding & Testing*

Coding 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 sistem. Dalam artian penggunaan komputer akan dimaksimalkan dalam tahapan ini.

Tahapan ini bisa dikatakan *final* dalam pembuatan sebuah sistem atau rancangan. Setelah melakukan analisa, desain dan pengkodean maka sistem atau rancangan yang sudah jadi akan digunakan oleh *user*.

4. Penerapan

Tahapan ini bisa dikatakan *final* dalam pembuatan sebuah sistem atau rancangan. Setelah melakukan analisa, design dan pengkodean maka sistem atau rancangan yang sudah jadi akan digunakan oleh *user*.

3.5 Alat Bantu dan Pengembangan Sistem

1. *Use case* Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah "apa" yang di perbuat sistem, dan bukan "bagaimana". Sebuah *Use case* mempresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu misalnya login sistem, meng-*create* sebuah daftar belanja dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.

Use case diagram dapat sangat membantu bila kita sedang menyusun requirement sebuah sistem, mengkomunikasikan rancangan klien, dan merancang test *case* untuk semua fitur yang ada pada sistem. Sebuah *use case* dapat meng-*include* fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang di *include* akan di panggil setiap kali *use case* yang mengincluded eksekusi secara normal.

Sebuah *Use case* dapat di include oleh lebih dari satu *Use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan menarik keluar fungsionalitas yang common. Sebuah *Use case* juga dapat meng *extend use case* lain dengan behaviornya sendiri. Sementara hubungan

generalisasi antar *Use case* menunjukkan bahwa *Use case* yang satu merupakan spesialisasi dari yang lain.

2. *Activity* Diagram

Activity Diagram atau yang disebut Diagram Aktivitas digunakan untuk mendokumentasikan alur kerja pada sebuah sistem, yang di mulai dari pandangan *business* level hingga ke operational level, adapun penjelasannya antara lain :

1. *Activity* diagram merupakan cara lain memodelkan aliran kejadian.
2. Kita gunakan *activity* diagram untuk menggambarkan *workflow*/aliran kerja proses bisnis.
3. *Activity* diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang di rancang bagaimana masing – masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir.
4. *Activity* diagram juga dapat menggambarkan proses parallel yang mungkin terjadi pada beberapa eksekusi.

3. *Class* Diagram

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek.

Class menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut

(metoda/fungsi). *Class* diagram menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan system untuk melakukan pekerjaan-pekerjaan tertentu.

Use case diagram dapat sangat membantu bula kita sedang menyusun requirement sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua fitur yang ada pada sistem. Sebuah *use case* dapat menginclude fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang di include akan di panggil setiap kali *use case* yang meng include dieksekusi secara normal. Sebuah *use case* dapat di *include* oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang common. Sebuah *use case* juga dapat meng *extend use case* lain dengan behavior sendiri. Sementara hubungan generalisasi antar *use case* menunjukan bahwa *use case* yang satu merupakan spesialisasi dari yang lain.

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan

tersebut (metode/fungsi). *Class* diagram menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan antar *class*.

4. *Sequence* Diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, display, dan sebagainya) berupa message yang digambarkan terhadap waktu. *Sequenc*e diagram terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah event untuk menghasilkan output tertentu. Diawali dari apa yang men- trigger aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan. Masing-masing objek, termasuk aktor, memiliki lifeline vertikal.

Message digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, message akan dipetakan menjadi operasi/metoda dari *class*. *Activation bar* menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah message.Untuk objek-objekyang memiliki sifat khusus, standar UML mendefinisikan icon khusus untuk objek boundary, controller dan persistent entity.

5. *Deployment* Diagram

Deployment/physical diagram menggambarkan detail bagaimana komponen di-deploy dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin,server atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server, dan hal-hal lain yang bersifat fisikal . Sebuah node adalah server, workstation, atau piranti keras lain yang digunakan untuk men-deploy komponen dalam lingkungan sebenarnya. Hubungan antar node (misalnya TCP/IP) dan requirement dapat juga didefinisikan dalam diagram ini.

3.6 Pengujian *Software*

Pengujian *Software* (perangkat Lunak) adalah elemen kritis dari jaminan kualitas perangkat lunak dan mempresentasikan kajian pokok dari spesifikasi, desain dan pengkodean.

Metode pengujian yang digunakan oleh penulis adalah *black-box testing*. *Black Box Testing* digunakan untuk menguji fungsi-fungsi dari perangkat lunak yang dirancang.Pengujian *black box* berfokus pada persyaratan fungsional perangkat lunak. Dengan demikian, pengujuan *black box* memungkinkan perekayasa perangkat lunak mendapat serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk semua program. Pengujian *black box* merupakan pendekatan komlementer yang kemungkinan besar mampu

mengungkap kelas kesalahan. Pengujian black box berusaha menemukan kesalahan dalam kategori sebagai berikut :

1. Fungsi yang tidak benar atau hilang.
2. Kesalahan antar muka.
3. Kesalahan dalam struktur data atau akses database eksternal.
4. Kesalahan kinerja dan Inisialisasi kesalahan terminasi.