

BAB II

LANDASAN TEORI

2.1. Tinjauan Pustaka

Dalam tinjauan pustaka ini penulis mengambil acuan untuk mengembangkan program “Sistem Informasi Reservasi *Villa* Berbasis *Web* pada *Villa Ratu Bogor*”

A. Konsep Dasar Sistem Informasi

Menurut Alter dalam Mulyanto (2009:28) “Sistem Informasi sebagai kombinasi antar prosedur kerja, informasi, orang dan teknologi informasi yang diorganisasikan untuk mencapai tujuan dalam sebuah organisasi”.

B. E-Commerce

Definisi – definisi tentang *e-commerce* dapat diartikan dalam pengertian menurut Nugroho (2006:5), adalah:

1. Berhubungan dengan pembelian dan penjualan barang atau jasa melalui internet, khususnya *World Wide Web*.
2. Suatu tindakan melakukan transaksi bisnis secara elektronik dengan menggunakan *internet* sebagai media komunikasi yang paling utama.
3. Menjual barang dagangan atau jasa melalui *internet*.

Sedangkan menurut Marpaung dalam Sunarto (2009:27) menyebutkan “*e-commerce* atau yang lebih dikenal dengan *e-com* dapat diartikan sebagai cara berbelanja atau berdagang secara *online* atau *direct selling* yang memanfaatkan fasilitas Internet di mana terdapat *website* yang dapat menyediakan layanan *get and deliver*”.

C. Dreamweaver

Menurut Sadeli (2011:2) “*Dreamweaver* merupakan suatu perangkat lunak *web editor* keluaran *Adobe System* yang digunakan untuk membangun dan mendesain suatu website dengan fitur – fitur yang menarik dan kemudahan dalam penggunaannya”.

D. PHP

Menurut Arief (2011:43) “*PHP (Hypertext Preprocessor)* adalah *server-side scripting* yang menyatu dengan *HTML* untuk membuat halaman *web* yang dinamis. Karena *PHP* merupakan *server-side scripting* maka sintaks dan perintah perintah *PHP* akan dieksekusi di *server* kemudian hasilnya dikirim ke *browser* dalam format *HTML*”.

E. MySQL

Menurut Arief (2011:151) “*MySQL* adalah salah satu jenis *database server* yang sangat terkenal dan banyak digunakan untuk membangun aplikasi *web* yang menggunakan *database* sebagai sumber dan pengolahan datanya”.

F. HTML

Menurut Arief (2011:23) “*HTML* atau *Hyper Text Markup Language* merupakan salah satu format yang digunakan dalam pembuatan dokumen dan aplikasi yang berjalan di halaman *web*. Dokumen ini dikenal sebagai *page*. Dokumen *HTML* merupakan dokumen yang disajikan pada *webbrowser*”.

G. Unified Modelling Language (UML)

Menurut Nugroho (2010:6) “*Unified Modelling Language (UML)* adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berparadigma ‘berorientasi objek’ ”.

UML (*Unified Modelling Language*) menawarkan sebuah standar untuk merancang model sebuah sistem. UML (*Unified Modelling Language*) mendefinisikan diagram-diagram berikut:

1. *Use Case Diagram*

Use Case Diagram menurut Nugroho (2010:34), “*Use Case* digunakan untuk memodelkan fungsionalitas-fungsionalitas sistem/perangkat lunak dilihat dari pengguna yang ada diluar sistem (yang sering digunakan sebagai *actor*)”. *Use Case* pada dasarnya merupakan unit fungsionalitas koheren yang di ekspresikan sebagai transaksi-transaksi yang terjadi antara *actor* dan sistem.

2. *Activity Diagram*

Activity Diagram menurut Nugroho (2010: 62), “Merupakan bentuk khusus dari *state machine* yang bertujuan memodelkan komputasi-komputasi dan aliran-aliran kerja yang terjadi dalam sistem atau perangkat lunak yang sedang dikembangkan”. Suatu diagram aktivitas memuat didalamnya *activity state* dimana suatu *activity state* merepresentasikan pernyataan dalam suatu prosedur atau kinerja suatu aktivitas dalam suatu aliran kerja.

3. *Class Diagram*

Class Diagram menurut Nugroho (2010:13), “*Class* digambarkan sebagai bentuk persegi panjang, dengan nama kelas di kompartemen paling atas, daftar kompartemen atribut berada dibagian tengah, serta daftar operasi-operasi berada di kompartemen paling bawah”. Relasi antara kelas digambarkan menggunakan bentuk garis lurus yang menghubungkan suatu kelas dengan kelas yang lainnya.

4. *Sequence Diagram*

Sequence Diagram menurut Nugroho (2010:42), “*Sequence Diagram* memperlihatkan interaksi sebagai diagram dua dimensi. Dimensi vertikal adalah sumbu waktu: waktu berubah dari atas kebawah. Dimensi horisontal memperlihatkan peran pengklasifikasi yang merepresentasikan objek-objek mandiri yang terlibat dalam kolaborasi”. Masing-masing peran pengklasifikasi direpresentasikan sebagai kolom-kolom vertikal dalam *sequence diagram* sering disebut sebagai garis waktu (*lifeline*).

5. *Component Diagram*

Component diagram menurut Nugroho (2010:66), “komponen sesungguhnya merupakan unit implementasi fisik yang memiliki antarmuka yang terdefinisi dengan baik dan digunakan sebagai bagian yang dapat digantikan dalam suatu sistem atau perangkat lunak yang kita kembangkan. Masing-masing komponen pada umumnya membungkus beberapa kelas perancangan dari tahap perancangan sistem”.

6. *Deployment Diagram*

Deployment diagram menurut Nugroho (2010:227), “*Deployment Diagram* merupakan suatu model objek yang mendeskripsikan sebaran fisik sistem atau perangkat lunak pada terminologi bagaimana fungsionalitas ditebarkan diantara simpul-simpul komputasional”. *Deployment* umumnya digunakan sebagai asupan esensial untuk aktifitas-aktifitas dalam tahapan perencanaan dan implementasi karena sebaran sistem memiliki imbas yang cukup berarti pada perancangan.

H. ERD (*Entity Relationship Diagram*)

ERD merupakan salah satu pemodelan data konseptual yang paling sering digunakan adalah proses pengembangan basis data bertipe relasional.

Menurut Kusriani dan Koniyo (2007:99), “ERD merupakan notasi grafis dalam permodelan data konseptual yang mendeskripsikan hubungan antar penyimpanan”.

ERD digunakan untuk memodelkan struktur data dan hubungan antar data, karena hal ini relatif kompleks. Dengan ERD kita dapat menguji model dengan mengabaikan proses yang harus dilakukan. ERD menggunakan sejumlah notasi dan simbol untuk menggambarkan struktur dan hubungan antar data. Setiap ERD bisa terdapat lebih dari satu atribut. Isi atribut mempunyai sesuatu yang dapat mengidentifikasi isi *entity* satu dengan yang lain. Atribut diawali oleh simbol *ellips* sebagai salah satu cara menggambarkan atribut. Komponen-komponen ERD yaitu:

1. Jenis Entitas

Jenis entitas (*Entitas Type*) dapat berupa suatu elemen lingkungan, sumber daya, atau transaksi yang begitu pentingnya bagi perusahaan sehingga di dokumentasikan dengan data jenis entitas di dokumentasikan dengan simbol persegi panjang.

2. Hubungan (*Realionship*)

Hubungan adalah suatu asosiasi yang ada antara dua jenis entitas. Hubungan digambarkan dengan bentuk belah ketupat. Tiap belah ketupat diberi label kata kerja.

3. Identifikasi dan deskripsi entitas

Tiap kejadian dari tiap entitas perlu diidentifikasi dan dideskripsikan dan ini dicapai dengan menggunakan atribut. Atribut adalah karakteristik dari suatu

entitas. Atribut-atribut tersebut sebenarnya adalah elemen-elemen data dan masing-masing diberikan satu nilai tunggal, yang disebut nilai atribut digambarkan dalam bentuk *ellips*.

4. Tingkatan dari *relationship*

Tingkatan dari *relationship* menyatakan jumlah *entity* yang berpartisipasi dalam suatu *relationship* terbagi menjadi:

- a) *Unary degree* (tingkat satu)
- b) *Binary degree* (tinggal dua)
- c) *Ternary degree* (tingkat tiga)

5. *Cardinality Ration*

Cardinality Ration adalah batasan yang menjelaskan jumlah keterhubungan satu *entity* dengan *entity* lainnya. Jenis *Cardinality Ration* antara lain:

a) *One to One Relationship* (1:1)

Hubungan antara *file* pertama dengan *file* kedua adalah satu berbanding satu, hubungan tersebut dapat digambarkan dengan tanda lingkaran untuk menunjukkan tabel atau relasi antara keduanya yang diwakilkan dengan tanda panah tunggal.

b) *One to Many Relationship* (1:M/M:1)

Hubungan antara *file* pertama dengan *file* kedua adalah berbanding banyak atau dapat pula dibalik banyak lawan satu. Hubungan tersebut dapat digambarkan dengan tanda lingkaran untuk menunjukkan tabel dan relasi antara keduanya diwakilkan dengan tanda panah ganda untuk menunjukkan hubungan banyak tersebut.

c) *Many to Many Relationship* (M:N/N:M)

Hubungan *file* pertama dengan *file* kedua adalah banyak berbanding banyak. Hubungan tersebut digambarkan dengan tanda lingkaran untuk menunjukkan tabel dan relasi antara keduanya diwakilkan dengan tanda panah ganda untuk menunjukkan hubungan banyak.

I. Metode *Waterfall*

Menurut Pressman (2010:36), “Sekuensial linier atau lebih sering disebut metode *waterfall* mengusulkan sebuah pendekatan kepada perkembangan perangkat lunak yang sistematis dan sekuensial yang mulai pada tingkat dan kemajuan sistem pada seluruh analisis, desain, kode, pengujian, dan pemeliharaan”. Di modelkan setelah siklus rekayasa konvensional, model sekuensial linier melingkupi aktivitas-aktivitas sebagai berikut:

1. Rekayasa dan pemodelan sistem atau informasi

Karena perangkat lunak selalu merupakan bagian dari sistem (bisnis) yang lebih besar, kerja dimulai dengan membangun syarat dari semua elemen sistem dan mengalokasikan beberapa subnet dari kebutuhan ke perangkat lunak tersebut. Pandangan sistem ini penting ketika perangkat lunak harus berhubungan dengan elemen-elemen yang lain seperti perangkat lunak, manusia dan *database*. Rekayasa dan analisis sistem menyangkut pengumpulan kebutuhan pada tingkat sistem dengan sejumlah kecil analisis serta desain tingkat puncak. Rekayasa informasi mencakup juga pengumpulan kebutuhan pada tingkat bisnis strategis dan tingkat area bisnis.

2. Analisis kebutuhan perangkat lunak

Proses pengumpulan kebutuhan diintensifkan dan difokuskan, khususnya pada perangkat lunak. Untuk memahami sifat program yang dibangun, perekayasa perangkat lunak (analisis) harus memahami domain informasi, tingkah laku, unjuk kerja dan antarmuka (*interface*) yang diperlukan. Kebutuhan baik untuk sistem maupun perangkat lunak di dokumentasikan dan dilihat lagi dengan pelanggan.

3. Desain

Desain perangkat lunak sebenarnya adalah proses multi langkah yang berfokus pada empat atribut sebuah program yang berbeda, struktur data, arsitektur perangkat lunak, representasi *interface*, dan detail (algoritma) prosedural. Proses desain menerjemahkan syarat atau kebutuhan ke dalam sebuah representasi perangkat lunak yang dapat diperkirakan demi kualitas sebelum dimulai pemunculan kode. Sebagaimana persyaratan, desain di dokumentasikan dan menjadi bagian dari konfigurasi perangkat lunak.

4. Generasi kode

Desain harus diterjemahkan ke dalam bentuk mesin yang bisa dibaca. Langkah pembuatan kode melakukan tugas ini. Jika desain dilakukan dengan cara yang lengkap, pembuatan kode dapat diselesaikan secara mekanis.

5. Pengujian

Sekali kode dibuat, pengujian program dimulai. Proses pengujian berfokus pada logika internal perangkat lunak, memastikan bahwa semua pernyataan sudah diuji, dan pada eksternal fungsional yaitu mengarahkan pengujian untuk menemukan kesalahan-kesalahan dan memastikan bahwa input yang dibatasi akan memberikan hasil aktual yang sesuai dengan hasil yang dibutuhkan.

6. Pemeliharaan

Perangkat lunak akan mengalami perubahan setelah disampaikan kepada pelanggan (perkecualian yang mungkin adalah perangkat lunak yang diletakan). Perubahan akan terjadi karena kesalahan-kesalahan ditentukan, karena perangkat lunak harus disesuaikan untuk mengakomodasi perubahan-perubahan didalam lingkungan eksternalnya, atau karena pelanggan membutuhkan perkembangan fungsional atau untuk kerja. Pemeliharaan perangkat lunak mengaplikasikan lagi setiap *fase* program sebelumnya dan tidak membuat yang baru lagi.

J. Black Box Testing

Menurut Pressman (2010:459), “Pengujian *Black Box* adalah pengujian aspek fundamental sistem tanpa memperhatikan struktur logika internal perangkat lunak”. Metode ini digunakan untuk mengetahui apakah perangkat lunak berfungsi dengan benar. Pengujian *black box* merupakan metode perancangan data uji yang didasarkan pada spesifikasi perangkat lunak. Data uji dieksekusi pada perangkat lunak dan kemudian keluar dari perangkat lunak di cek apakah telah sesuai yang diharapkan.

Menurut Pressman (2010:459), Pengujian *black box* berusaha menentukan kesalahan dalam kategori, di antaranya:

1. Fungsi-fungsi yang tidak benar atau hilang
2. Kesalahan interface
3. Kesalahan dalam struktur data atau akses *database* eksternal
4. Kesalahan kinerja
5. Inisialisasi dan kesalahan terminasi

Keunggulan *black box* yaitu bisa memilih *subset test* secara efektif dan efisien, dapat menemukan cacat, dan bisa memaksimalkan tentang *testing*

investmen. Sedangkan kelemahan dari *black box* yaitu *tester* tidak pernah yakin apakah perangkat lunak tersebut benar-benar lolos uji.

Menurut Pressman (2010:459), Langkah-langkah dalam pengujian *black box*, yaitu:

1. Analisa kebutuhan dan spesifikasi
2. Pemilihan *input*
3. Pemilihan *output*
4. Seleksi *input*
5. Pengujian
6. Revisi hasil
7. Evaluasi

2.2. Penelitian Terkait

Berikut jurnal penelitian yang terkait dan mendukung dari permasalahan yang diangkat:

Hotel Remaja Pacitan sebagai salah satu hotel modern di kawasan kota Pacitan, merupakan hotel yang sedang berkembang dan terus berupaya untuk meningkatkan layanan serta standar kualitas hotel dari berbagai sisi antara lain sisi pemanfaatan teknologi informasi. Selama ini, sistem sewa kamar atau check in yang ada di Hotel Remaja pacitan masih kurang efektif, di lakukan dengan cara melakukan pencatatan pada buku, terlambatnya untuk mencari kamar yang kosong dan belum dipesan, pembuatan laporan yang sering terlambat, dan perhitungan biaya check out yang lama, sehingga sistem tersebut perlu dibenahi. Beberapa sistem yang dinilai perlu dibenahi adalah sistem pemesanan sewa kamar, sistem check-in, sistem check-out serta pembuatan laporan. Sistem tersebut perlu dibenahi dengan sistem baru yang terkomputerisasi. Sehingga praktikan tertarik mengambil judul Perancangan sistem informasi pemesanan dan pembayaran kamar pada hotel Remaja Pacitan (Damayanti & Wardati, 2014:10-11).

Sebenarnya banyak kendala yang dihadapi oleh pihak hotel bila pemesanan masih dilakukan secara manual. Pertama, tenaga kerja *front desk* harus lebih banyak dan terus menerus berada di tempat untuk menangani reservasi dan *check-in*. Kedua, walaupun pelanggan berhasil menelepon, ketersediaan kamar belum tentu ada. Informasi ketersediaan kamar hotel tidak tersalur dengan cepat ke pelanggan. Ketiga, ketika pelanggan menelepon untuk pemesanan, staf *front*

desk juga perlu memeriksa buku pemesanan secara manual dan hal tersebut membutuhkan waktu yang tidak sebentar jika kamar yang dimiliki cukup banyak dengan tanggal pemesanan yang berbeda - beda. Kesalahan (*human error*) juga dapat terjadi dalam pemeriksaan dan pencatatan tersebut, apalagi jika ada pembatalan pemesanan, sehingga ketika tamu hotel yang sudah melakukan pemesanan datang, kamar tidak tersedia dan sebaliknya. Keempat, pihak manajemen hotel juga sulit untuk menghitung pendapatan, melihat daftar pelanggan, daftar kamar, dan lain-lain. Semuanya harus dilakukan dan dikelola secara manual, dimana untuk jangka panjang hal tersebut akan mempengaruhi perkembangan hotel ke depannya. Dari permasalahan yang diuraikan di atas, maka penulis merumuskan masalah dalam penelitian ini adalah Proses pemesanan dan pembayaran pada umumnya masih menggunakan cara konvensional. Untuk itu para pelanggan harus mendatangi langsung ke tempat pemasaran kamar hotel tersebut. Adapun tujuan penelitian ini adalah untuk mengembangkan aplikasi reservasi kamar hotel berbasis *web* serta dapat memberikan informasi tentang layanan hotel yang tersedia yang bermanfaat bagi konsumen. Perusahaan dapat meningkatkan mutu pelayanan terhadap konsumen terutama dalam hal pelayanan penjualan tiket kamar hotel (Damanik, 2011: 123-124).

Berdasarkan dua kutipan jurnal diatas, penulis menyadari bahwa adanya permasalahan yang sama pada *Villa Ratu*.