



BAB II

DASAR TEORI

BAB II

LANDASAN TEORI

2.1. Sistem Reservasi

Setiap pelanggan yang ingin mengunjungi sebuah salon, baik itu untuk keperluan perawatan atau layanan lainnya, biasanya akan melakukan perencanaan terlebih dahulu dengan melakukan reservasi. Reservasi berasal dari kata "*to reserve*", yang berarti menyiapkan atau menyediakan slot waktu sebelumnya untuk layanan yang diinginkan. Reservasi ini mengacu pada pemesanan slot waktu untuk layanan fasilitas salon. Jadi secara umum reservation yaitu pemesanan fasilitas yang diantaranya akomodasi, jasa, *meal*, *seat* pada pertunjukan, pesawat terbang, kereta api, bus, hiburan, *night club*, *discoutegue* dan sebagainya. [3]

Definisi dari reservasi adalah sebuah proses perjanjian reservasi baik itu berupa barang maupun jasa akan tetapi belum ditutup dengan sebuah transaksi jual beli. Biasanya proses ini ditandai dengan adanya proses tukar-menukar informasi antara produsen dan konsumen yang menghasilkan sebuah kesepakatan mengenai jasa yang ingin dipesan. [4]

Kesimpulan dari pengertian di atas adalah reservasi merupakan proses pemesanan atau penyiapan slot waktu untuk layanan atau fasilitas tertentu. Ini melibatkan interaksi antara produsen dan konsumen, di mana informasi ditukar untuk mencapai kesepakatan tentang barang atau jasa yang akan dipesan. Pentingnya reservasi adalah untuk memastikan ketersediaan dan pemenuhan kebutuhan pelanggan, meskipun proses tersebut belum menghasilkan transaksi jual beli yang final.

2.2. Pelayanan

Pelayanan adalah setiap tindakan atau kegiatan yang dapat ditawarkan oleh suatu pihak kepada pihak lain, yang pada dasarnya tidak berwujud dan tidak mengakibatkan kepemilikan apapun. Pada umumnya pelayanan yang diberikan oleh produsen atau perusahaan yang memiliki kualitas yang baik akan menghasilkan kepuasan yang tinggi serta pembelian ulang yang lebih sering. [5]

Istilah pelayanan berasal dari kata "layan" yang artinya menolong menyediakan segala apa yang diperlukan oleh orang lain untuk perbuatan melayani. Pada dasarnya setiap manusia membutuhkan pelayanan, bahkan secara ekstrim dapat dikatakan bahwa pelayanan tidak dapat dipisahkan dengan kehidupan manusia. [6]

Dari kedua kutipan di atas, dapat disimpulkan bahwa pelayanan merujuk pada segala tindakan atau kegiatan yang ditawarkan oleh satu pihak kepada pihak lain tanpa menghasilkan kepemilikan barang. Konsep ini sangat penting dalam bisnis karena pelayanan yang baik, terutama yang berkualitas tinggi, cenderung meningkatkan tingkat kepuasan pelanggan dan mendorong pembelian ulang. Secara umum, pelayanan dianggap sebagai bagian integral dari kehidupan manusia, menyoroti pentingnya interaksi sosial dan bantuan antara individu dalam masyarakat.

2.3. Salon

Salon adalah sebuah tempat yang menyediakan layanan perawatan tubuh dan kecantikan, seperti potong, perawatan kulit, perawatan kuku, dan makeup. Istilah "salon" berasal dari bahasa Prancis yang artinya ruangan atau tempat berkumpul. [7]

Menurut Kamus Besar Bahasa Indonesia, pengertian salon adalah ruang (kamar) yang diatur dan dihias dengan baik (untuk menerima tamu dsb). Secara umum, salon dapat dibagi menjadi dua bagian, yakni beauty salon (salon kecantikan, wajah atau tubuh) dan hair salon (salon rambut). Tetapi pemakaian kata salon saja sudah dapat mewakili keduanya. Salon merupakan tujuan masyarakat ketika hendak mempercantik diri. Oleh karena itu, ada pula istilah salon kecantikan, yaitu tempat (gedung dsb) orang merawat kecantikan (merias muka, menata rambut, dsb).

Dari pengertian di atas dapat disimpulkan bahwa salon adalah tempat yang menyediakan layanan perawatan tubuh dan kecantikan seperti potong rambut, perawatan kulit, perawatan kuku, dan makeup. Kata "salon" berasal dari bahasa Prancis yang berarti ruang berkumpul. Menurut Kamus Besar Bahasa Indonesia, salon adalah ruang yang dihias untuk menerima tamu. Salon terbagi menjadi beauty salon (kecantikan) dan hair salon (rambut), meskipun istilah "salon" dapat mencakup keduanya. Salon adalah tujuan masyarakat untuk mempercantik diri.

2.4. Website

Website atau situs *web* menurut Josi (2016), dapat diartikan sebagai kumpulan halaman yang menampilkan informasi data teks, data gambar, diam, atau gerak, data animasi, suara, video dan gabungan dari semuanya baik yang bersifat statis maupun dinamis yang membentuk suatu rangkaian bangunan saling terkait dimana masing-masing dihubungkan dengan jaringan halaman (*hyperlink*) yang dapat diakses melalui perangkat lunak yang disebut *browser* (perambah) adalah aplikasi yang mampu menjalankan dokumen-dokumen *web* dengan cara diterjemahkan. Prosesnya dilakukan oleh komponen yang terdapat didalam aplikasi browser yang biasa disebut web engine semua dokumen web ditampilkan dengan cara diterjemahkan. Beberapa contoh web browser yang populer saat ini adalah *Google Chrome*, *Mozilla Firefox*, dan *Opera*. [11]

Web merupakan representasi sebuah halaman di internet yang menyajikan berbagai macam informasi. Halaman – halaman tersebut merupakan dokumen yang tersebar di beberapa komputer di seluruh dunia yang terhubung satu sama lain. [12]

Website merupakan representasi halaman-halaman di seluruh dunia yang terhubung melalui internet untuk menyajikan informasi seperti teks, gambar, animasi, suara, dan video, yang dapat statis atau dinamis. Halaman-halaman ini saling terhubung melalui *hyperlink* dan diakses melalui *browser*, aplikasi yang menerjemahkan dokumen *web* menggunakan web engine.

2.5. Pemrograman Berorientasi Objek

Pemrograman Berorientasi Objek (PBO) adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya. Tujuan dari PBO diciptakan adalah untuk mempermudah pengembangan program dengan cara mengikuti model yang telah ada di kehidupan sehari-hari. Jadi setiap bagian dari suatu permasalahan adalah objek dan objek itu sendiri merupakan gabungan dari beberapa objek yang lebih kecil lagi, sebuah objek yang besar dibentuk dari beberapa objek yang lebih kecil, objek-objek itu saling berkomunikasi, dan saling berkiriman pesan kepada objek yang lain. [13]

Pemrograman Berorientasi Objek (PBO) adalah suatu cara baru dalam berpikir serta berlogika untuk menghadapi masalah-masalah yang akan dicoba atasi dengan komputer. PBO mencoba melihat permasalahan lewat pengamatan dunia nyata yang setiap objeknya adalah entitas tunggal yang memiliki kombinasi struktur data dan fungsi tertentu. [14]

a. Kelas (*class*)

Kelas adalah kumpulan objek-objek dengan karakteristik yang sama. Kelas merupakan definisi statik dan himpunan objek yang sama mungkin diciptakan oleh kelas tersebut. Sebuah kelas mempunyai sifat (atribut), kelakuan (*operasi/method*), hubungan (*relationship*) dan arti. Suatu kelas dapat diturunkan ke kelas yang lain, dimana atribut dan kelas semula dapat diwariskan ke kelas yang baru.

b. Objek (*object*)

Objek adalah abstraksi dan sesuatu yang mewakili dunia nyata seperti benda, manusia, satuan organisasi, tempat, kejadian, struktur, status, atau hal-hal lain yang bersifat abstrak. Objek merupakan suatu entitas yang mampu menyimpan informasi dan mempunyai operasi yang dapat diterapkan.

c. Metode (*method*)

Method adalah fungsi atau prosedur yang dibuat oleh seorang *programmer* didalam suatu *class*. Pada sebuah *method* didalam sebuah kelas juga memiliki izin akses seperti atribut, yaitu *private*, *public*, dan *protected*. Sebuah kelas boleh memiliki lebih dari satu *method* dengan nama yang sama asalkan memiliki parameter masukan yang berbeda sehingga *compiler* atau *interpreter* dapat mengenali *method* mana yang dipanggil.

d. Atribut (*attribute*)

Atribut dari sebuah kelas adalah variabel global yang dimiliki sebuah kelas. Atribut dapat berupa nilai atau elemen-elemen data yang dimiliki oleh objek dalam kelas objek. Atribut dipunyai secara individual oleh sebuah objek, misalnya berat, jenis, nama, dan sebagainya. Atribut sebaiknya bersifat privat untuk menjaga enkapsulasi.

e. Enkapsulasi (*encapsulation*)

Pembungkusan atribut data dan layanan (operasi/operasi) yang dipunyai objek untuk menyembunyikan implementasi dan objek sehingga objek lain tidak mengetahui cara kerjanya.

Kesimpulan dari penjelasan di atas yaitu Pemrograman Berorientasi Objek (PBO) mengorganisasi perangkat lunak sebagai kumpulan objek dengan data dan operasi yang diberlakukan terhadapnya. Tujuan utama PBO adalah mempermudah pengembangan program dengan model yang ada dalam kehidupan sehari-hari. Dalam PBO, setiap bagian permasalahan dianggap sebagai objek, yang dapat terdiri dari objek-objek lebih kecil.

2.6. Laravel

Laravel adalah *framework* PHP yang didistribusikan di bawah lisensi MIT dan didasarkan pada pola desain *model view controller* (MVC). Laravel adalah *framework* berbasis PHP yang dirancang untuk meningkatkan kualitas perangkat lunak dengan mengurangi biaya pengembangan dan pemeliharaan awal, dan untuk meningkatkan pengalaman bekerja dengan aplikasi dengan menyediakan sintaks yang ekspresif, tidak ambigu, dan menghemat waktu. [8]

Laravel adalah kerangka kerja yang dibangun dengan basis bahasa pemrograman PHP. Laravel memiliki komunitas dan pengguna yang terus berkembang hingga sekarang, Laravel sendiri tidak terlepas dari konsep MVC (*Model, View, Controller*) dimana merupakan sebuah konsep modern yang memisahkan bagian tampilan (*front-end*) dan juga bagian pengelolaan data atau biasa disebut *controller* (*back-end*). [10]

Laravel adalah sebuah *framework* PHP yang didistribusikan di bawah lisensi MIT dan didasarkan pada pola desain *Model View Controller* (MVC). *Framework* ini dirancang untuk meningkatkan kualitas perangkat lunak dengan cara mengurangi biaya pengembangan dan pemeliharaan awal, serta untuk meningkatkan pengalaman pengembang dengan menyediakan sintaks yang ekspresif, tidak ambigu, dan menghemat waktu.

2.7. Laragon

Laragon adalah perangkat lunak gratis yang mendukung banyak sistem operasi, berfungsi sebagai server diri sendiri atau localhost. Laragon menyediakan banyak *services*, *tools*, dan fitur mulai dari *Apache*, *MySQL*, *PHP Server*, *Memcached*, *Redis*, *Composer*, *Xdebug*, *PhpMyAdmin*, *Cmder* dan *Laravel*. [8]

Laragon adalah perangkat lunak yang bersifat *open source* (terbuka) yang dapat mendukung banyak sekali sistem operasi dimana laragon bertugas sebagai server virtual atau sering disebut sebagai *localhost*. [9]

Dari pengertian diatas dapat disimpulkan bahwa Laragon adalah perangkat lunak gratis dan *open-source* yang mendukung banyak sistem operasi. Fungsinya adalah sebagai *server* diri sendiri atau *localhost*. Laragon berperan sebagai server *virtual* atau *localhost*, memfasilitasi pengembangan dan pengujian aplikasi *web* secara lokal di berbagai sistem operasi.



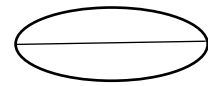
2.8. Database

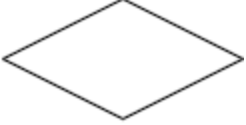

Definisi *database* adalah sekumpulan informasi atau data yang saling terkait satu dengan yang lainnya, yang dimana data itu tersimpan di luar komputer. Untuk memanipulasi data tersebut tentu dibutuhkan *software* tertentu atau *software* secara khusus. Secara konsep, *Database* atau Basis Data yaitu kumpulan dari data-data yang membentuk suatu berkas atau file yang saling berhubungan dengan tata cara yang tertentu untuk membentuk data baru atau informasi. Atau bisa diartikan *Database* atau Basis Data merupakan kumpulan dari data yang saling berhubungan antara satu dengan yang lainnya yang diorganisasikan berdasarkan skema atau struktur tertentu. [15] Database adalah suatu susunan atau kumpulan catatan data yang tersimpan di dalam komputer. Hubungan antar entri dalam database dapat digunakan sebagai sumber informasi bagi pengguna. [16]

A. Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) merupakan diagram yang digambarkan dalam bentuk grafis dalam pembuatan *database* yang menghubungkan hubungan antar data yang satu dengan data yang lain. Fungsi *ERD* adalah untuk membantu pembuatan *database* dan memberikan gambaran terhadap kerja *database* yang akan dibuat. [17] Simbol yang digunakan untuk membuat *ERD*, dapat dilihat pada **Tabel 2.1**

Tabel 2.1 Simbol Entity Relationship Diagram

No	Simbol	Nama	Keterangan
1		<i>Entity</i>	Simbol yang menyatakan himpunan entitas ini bisa berupa suatu elemen lingkungan, sumber daya, atau transaksi yang begitu pentingnya bagi perusahaan sehingga didokumentasikan dengan data
2		<i>Attribute</i>	Simbol terminal ini untuk menunjukkan nama-nama atribut yang ada pada suatu <i>entity</i>
3		<i>Primary Key, Attribute</i>	Simbol atribut yang digaris bawah, berfungsi sebagai key (kunci) diantara nama-nama atribut yang ada pada suatu <i>entity</i>

4		<i>Relationship</i>	Simbol ini menyatakan relasi ini digunakan untuk menunjukkan hubungan yang ada antara entiti yang satu dengan entiti yang lainnya
5		<i>Link</i>	Simbol berupa garis ini digunakan sebagai penghubung antara himpunan relasi dengan himpunan entitas dan himpunan entitas dengan atributnya

B. Query

1. Data Definition Language (DDL)

DDL (*Data Definition Language*) merupakan suatu perintah yang digunakan untuk menciptakan struktur data, atau untuk membangun *database*. DDL mempunyai tugas untuk membuat objek SQL dan menyimpan definisinya dalam tabel. Contoh objek tersebut yaitu tabel, *view*, dan *index*. DDL mempunyai fungsi untuk melakukan perubahan struktur tabel, mengubah nama tabel. Berikut perintah-perintah yang ada dalam golongan DDL: [18]

- a. *Create*, digunakan untuk membuat *database*, tabel, dan objek lain dalam *database*.
- b. *Alter*, digunakan untuk memodifikasi tabel, seperti contoh mengubah nama tabel, *field*, menambah nama *field*.
- c. *Drop*, digunakan untuk menghapus *database*, tabel, dan objek lain dalam *database*.

2. Data Manipulation Language (DML)

DML (*Data Manipulation Language*) merupakan *database* yang digunakan untuk melakukan modifikasi dan pengambilan data pada suatu *database*. Pengolahan atau modifikasi ini meliputi: [18]

- (a) *Insert*, digunakan untuk melakukan penambahan data baru ke dalam sebuah table.
- (b) *Select*, digunakan untuk pengambilan data, contohnya: mengambil semua data dari sebuah tabel.
- (c) *Update*, digunakan untuk melakukan perubahan data,
- (d) *Delete*, digunakan untuk melakukan penghapusan data.

2.9. MySQL

MySQL adalah suatu *Relational Database Management System (RDBMS)* yang mendukung *database* yang terdiri dari sekumpulan relasi atau tabel. Pada umumnya perintah yang paling sering digunakan dalam *MySQL* adalah *INSERT* (menambah), *SELECT* (mengambil), *UPDATE* (mengubah), dan *DELETE* (menghapus). Selain itu SQL juga menyediakan perintah untuk membuat *database*, *field* ataupun indeks untuk menambah ataupun menghapus data. [19]

MySQL merupakan *database* yang berbasis *server database*. *Database MySQL* mampu menangani data yang sangat besar hingga ukuran *Giga byte*, dengan kemampuan daya tampung data ini maka *MySQL* sangat cocok digunakan untuk mengcover data pada perusahaan baik yang kecil sampai perusahaan besar. [20]

2.10. Unified Modeling Language (UML)

Unified Modeling Language adalah sebuah bahasa yang berdasarkan grafik/gambar untuk memvisualisasi, menspesifikasikan, membangun, dan pendokumentasian dari sebuah sistem pengembangan *software* berbasis OO (*ObjectOriented*). UML tidak hanya merupakan sebuah bahasa pemrograman visual saja, namun juga dapat secara langsung dihubungkan ke berbagai bahasa pemrograman, seperti JAVA, C , *Visual Basic*, atau bahkan dihubungkan secara langsung ke dalam sebuah *object-oriented database*. [21]





Unified Modeling Language UML merupakan alat bantu yang memiliki kemampuan handal di dalam pengembangan system yang berorientasi objek. Kehandalan UML ini dibuktikan dengan disediakannya bahasa pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk baku, mudah dimengerti, dan dilengkapi dengan mekanisme yang efektif untuk berbagi serta mengkomunikasikan rancangan mereka dengan yang lain. [22]


Unified Modeling Language (UML) adalah Bahasa untuk menspesifikasi, memvisualisasikan, serta mengonstruksi bangunan dasar sistem perangkat lunak, termasuk melibatkan pemodelan aturan-aturan bisnis. UML merupakan perkakas utama untuk analisis dan perancangan sistem berorientasi objek. [14] Terdapat beberapa diagram UML yang sering digunakan pada untuk pengembangan sebuah sistem yaitu :

a. Use Case

Merupakan gambaran dari fungsionalitas yang diharapkan dari sebuah sistem, dan merepresentasikan sebuah interaksi antara aktor dan sistem. Didalam *use case* terdapat *actor* yang merupakan sebuah gambaran entitas dari manusia atau sebuah sistem yang melakukan pekerjaan di sistem. Simbol yang digunakan untuk membuat *use case* dapat dilihat pada **tabel 2.2**

Tabel 2. 2 Simbol *Use Case*

No	Simbol	Nama	Keterangan
1	 Actor	<i>Actor</i>	Merupakan pengguna dari sistem. Penamaan aktor menggunakan kata benda
2	 Use Case	<i>Use Case</i>	Merupakan pekerjaan yang dilakukan oleh aktor. Penamaan <i>use case</i> dengan kata kerja
3		<i>Association / Asosiasi</i>	Hubungan antara aktor dengan <i>use case</i>
4		<i>Include</i>	Hubungan antara <i>use case</i> dengan <i>use case</i> . <i>Include</i> menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsional dari <i>use case</i> lainnya.

5		<i>Extends</i>	Hubungan antara <i>use case</i> dengan <i>use case</i> . <i>Extends</i> menunjukkan bahwa suatu <i>use case</i> merupakan suatu tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi.
---	---	----------------	--


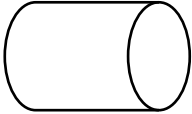

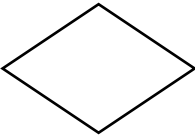
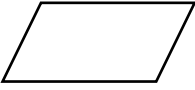

2.11. Flowchart


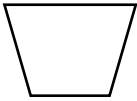

Flowchart merupakan penggambaran secara grafik dari langkah-langkah dan urutan prosedur suatu program, Biasanya mempengaruhi penyelesaian masalah yang khususnya perlu dipelajari dan dievaluasi lebih lanjut.[23]

Flowchart adalah langkah-langkah pemecahan masalah yang ditulis dalam simbol-simbol tertentu. *Flowchart* ini akan menampilkan alur dalam program secara logika. *Flowchart* ini tidak hanya digunakan sebagai alat komunikasi tetapi juga diperlukan sebagai panduan dan sebelum pemahaman yang lebih baik dari komponen-komponen *flowchart*. [24]

Flowchart dapat digunakan untuk menyajikan kegiatan manual, kegiatan pemrosesan ataupun keduanya. *Flowchart* merupakan rangkaian simbol-simbol yang digunakan untuk mengkontruksi. Simbol yang digunakan dapat dilihat pada **Tabel 2.3**

Tabel 2. 3 Simbol *Flowchart*

No	Simbol	Nama	Keterangan
1		<i>Process</i>	Simbol <i>flowchart</i> persegi panjang yang menunjukkan langkah aliran proses normal yang ada di dalam sebuah sistem.
2		<i>Direct Data</i>	Bentuk <i>flowchart</i> silinder, menunjukkan penyimpanan data yang memungkinkan akses langsung untuk operasi baca/tulis.
3		<i>Terminator</i>	Simbol <i>flowchart</i> oval yang menunjukkan awal atau akhir sebuah proses dan prosedur.
4		<i>Desicion</i>	Bentuk <i>flowchart</i> alir berlian yang menunjukkan percabangan dan situasi-situasi berbeda dalam sebuah aliran proses.
5		<i>Data</i>	Simbol <i>flowchart</i> jajar genjang yang menunjukkan <i>input</i> atau <i>output</i> data (I/O) dalam suatu sistem atau proses.
6		<i>Document</i>	Bentuk <i>flowchart</i> yang digunakan untuk menunjukkan sebuah dokumen atau laporan dari suatu proses, bisa dalam bentuk <i>print out</i> .

7		<i>Flow</i>	Simbol yang menentukan arah alur dalam suatu sistem dan prosedur.
8		<i>Manual Operation</i>	Simbol yang menyatakan suatu proses yang tidak dilakukan oleh komputer
9		<i>Manual Input</i>	Simbol yang menyatakan suatu proses memasukkan data yang tidak dilakukan oleh komputer