

PERANCANGAN APLIKASI *CUSTOMER RELATIONSHIP MANAGEMENT* BERBASIS WAP PADA *THE ALIGA HOTEL*

Putri Anggraini¹⁾, Dio Prima Mulya²⁾

¹⁾Prodi Sistem Informasi, Universitas Dharma Andalas, Jl. Sawahan No.103A, Simpang Haru, Kota Padang
email: bontetga@unidha.ac.id

²⁾ Prodi Sistem Informasi, Universitas Dharma Andalas, Jl. Sawahan No.103A, Simpang Haru, Kota Padang
email: dio@unidha.ac.id

ABSTRAK

Kemajuan teknologi informasi dan komunikasi yang sangat cepat membuat semua aspek kehidupan harus mengikutinya. Salah satunya adalah dunia usaha, dengan memanfaatkan teknologi informasi dan komunikasi maka akan meningkatkan efisiensi di segala bidang. Berdasarkan fakta diatas, pada penulisan tugas akhir ini penulis membahas tentang bagaimana cara merancang sebuah sistem aplikasi pemesanan fasilitas dengan menggunakan WAP, PHP dan MySQL pada The Aliga Hotel.

Disini penulis menganalisa dan merancang infrastruktur sistem aplikasi yang akan dan harus dibangun, struktur navigasi, basis data yang digunakan, bahasa pemrograman yang digunakan serta integrasi dari keduanya. Untuk itu dalam pembuatan aplikasi pemesanan fasilitas hotel ini penulis menggunakan bahasa pemrograman WAP dan PHP sebagai komponen dasar dari perancangan sistem, basis data menggunakan MySQL, untuk pemodelan dengan bahasa UML menggunakan Rational Rose Enterprise 2002 dan untuk proses design dan coding menggunakan Adobe Photoshop CS5, Corel Draw X4, serta Adobe Dreamweaver CS5.

Dari hasil penelitian tugas akhir ini, dapat disimpulkan bahwa proses pemesanan fasilitas berlangsung secara online dan terkomputerisasikan dengan baik.

Kata kunci : CRM, Pemesanan Fasilitas, WAP, PHP, MySQL, Web.

Abstract

The rapid advancement of information and communication technology makes all aspects of life must follow. One is the business world, by utilizing information and communication technology it will increase efficiency in all fields. Based on the above facts, the writing of this final task the author discusses about how to design an application facility booking system using WAP, PHP and MySQL on The Aliga Hotel.

Here the authors analyze and design the application system infrastructure that will and should be built, the navigation structure, the database used, the programming language used and the integration of both. For that in the making of the booking application of this hotel facility the author uses the WAP programming language and PHP as a basic component of system design, the database using MySQL, for modeling with UML language using Rational Rose Enterprise 2002 and for the process of design and coding using Adobe Photoshop CS5, Corel Draw X4, and Adobe Dreamweaver CS5.

From the results of this final project, it can be concluded that the process of ordering the facility takes place online and well computerized.

Keywords: CRM, Booking Facility, WAP, PHP, MySQL, Web.

1. Pendahuluan

Dalam dunia internet sekarang ini, telah banyak dibangun aplikasi, seperti aplikasi pada proses penjualan atau penawaran online yang dapat digunakan untuk memudahkan mendapatkan dan menyebarkan informasi secara online baik untuk kalangan sendiri (*admin*) maupun kalangan luas terutama pelanggan dan pengunjung *website* lainnya. Seorang *admin* dapat melihat dan mengidentifikasi layanan-layanan yang tersedia pada sistem tersebut, menambah informasi berita atau kritikan, serta memberi hak akses kepada user (*user account*) yang berpartisipasi dalam menambahkan sejumlah informasi ke dalam sistem tersebut, dimana

hak akses yang diberikan kepada user adalah terbatas.

Perkembangan aplikasi penjualan online disini salah satu contohnya adalah perkembangan aplikasi CRM (*Customer Relationship Management*). CRM adalah kombinasi dari strategi perusahaan, proses bisnis dan teknologi informasi yang digunakan untuk mempelajari lebih lanjut tentang kebutuhan pelanggan dan perilaku dalam rangka untuk mengembangkan hubungan yang kuat dengan mereka. Dalam kata lain CRM adalah dimaksudkan untuk memberikan pengetahuan untuk mengembangkan dan menerapkan strategi cerdas dan sebagian besar pelanggan untuk memaksimalkan profitabilitas pelanggan.

Inisiatif CRM dirancang dengan tujuan memenuhi harapan dan kebutuhan pelanggan untuk mencapai nilai pelanggan seumur hidup maksimum dan kembali ke perusahaan. Penggunaan produk CRM, perangkat lunak CRM dan solusi CRM akan meningkatkan pelaksanaan yang efektif dari CRM dalam organisasi.

Trend kompetisi perusahaan tidak hanya atas dasar produk, tetapi juga pada aspek hubungan pelanggan. Kedua produk dan hubungan pelanggan yang dibutuhkan untuk mempertimbangkan kembali dan diperbaiki untuk meningkatkan keuntungan perusahaan untuk memenangkan persaingan.

CRM berbasis WAP ini dibangun dengan mengacu pada prinsip *user friendly*, mudah untuk diperbaharui, sehingga ketika suatu saat terjadi penggantian administrator akan mudah dalam menggunakannya. Perawatan dan penggunaan aplikasi CRM berbasis WAP ini tidak rumit karena sudah disesuaikan dengan kebutuhan pelanggan atau pengguna *website*, serta penggunaan dan pengembangan di masa yang akan datang pun akan lebih mudah dilakukan.

Sebagaimana yang kita ketahui seperti aplikasi CRM pada hotel. Hotel memberikan pelayanan yang sudah ditentukan sebelumnya seperti pelayanan kamar dengan berbagai jenis/model, pelayanan Resepsionis, pelayanan office boy dan lain-lainnya yang dilakukan secara manual. Maka dalam proses pengembangan aplikasi CRM berbasis WAP dapat membantu perusahaan atau hotel tersebut dalam memberikan pilihan dan pelayanan kepada pelanggan secara online dimanapun berada sehingga dapat menekan waktu dan biaya,serta proses yang dilakukan lebih cepat dan efisien.

Maka dari itu penulis mencoba membangun dan mengembangkan aplikasi CRM berbasis WAP dengan mengangkat judul penelitian “**PERANCANGAN APLIKASI COSTUMER RELATIONSHIP MANAGEMENT BERBASIS WAP PADA THE ALIGA HOTEL**”

2.1 Rekayasa Perangkat Lunak

2.1.1 Definisi Rekayasa Perangkat Lunak

Istilah Rekayasa Perangkat Lunak (RPL) secara umum disepakati sebagai terjemahan dari istilah Software engineering. Istilah Software Engineering mulai dipopulerkan pada tahun 1968 pada software engineering Conference yang diselenggarakan oleh NATO. Sebagian orang mengartikan RPL hanya sebatas pada bagaimana membuat program komputer. Padahal ada perbedaan yang mendasar antara perangkat lunak (software) dan program komputer.

Perangkat lunak adalah seluruh perintah yang digunakan untuk memproses informasi. Perangkat lunak dapat berupa program atau prosedur. Program adalah kumpulan perintah yang dimengerti oleh komputer sedangkan prosedur adalah perintah yang dibutuhkan oleh pengguna dalam memproses informasi (O'Brien, 1999).

Pengertian RPL sendiri adalah suatu disiplin ilmu yang membahas semua aspek produksi perangkat lunak, mulai dari tahap awal yaitu analisa kebutuhan pengguna, menentukan spesifikasi dari kebutuhan pengguna, disain, pengkodean, pengujian sampai pemeliharaan sistem setelah digunakan. Dari pengertian ini jelaslah bahwa RPL tidak hanya berhubungan dengan cara pembuatan program komputer. Pernyataan ”semua aspek produksi” pada pengertian di atas, mempunyai arti semua hal yang berhubungan dengan proses produksi seperti manajemen proyek, penentuan personil, anggaran biaya, metode, jadwal, kualitas sampai dengan pelatihan pengguna merupakan bagian dari RPL.

2.1.2 Sejarah Rekayasa Perangkat Lunak

Istilah software engineering digunakan pertama kali pada akhir 1950-an dan awal 1960-an. Saat itu, masih terdapat perdebatan tajam mengenai aspek engineering dari pengembangan perangkat lunak. Pada tahun 1968 dan 1969, komite sains NATO mensponsori dua konferensi tentang rekayasa perangkat lunak, yang memberikan dampak kuat terhadap pengembangan rekayasa perangkat lunak. Banyak yang menganggap dua konferensi inilah yang menandai awal resmi profesi rekayasa perangkat lunak.

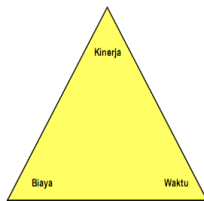
Pada tahun 1960-an hingga 1980-an, banyak masalah yang ditemukan para praktisi pengembangan perangkat lunak. Banyak project yang gagal, hingga masa ini disebut sebagai krisis perangkat lunak. Kasus kegagalan pengembangan perangkat lunak terjadi mulai dari project yang melebihi anggaran, hingga kasus yang mengakibatkan kerusakan fisik dan kematian. Salah satu kasus yang terkenal antara lain meledaknya roket Ariane akibat kegagalan perangkat lunak. Selama bertahun-tahun, para peneliti memfokuskan usahanya untuk menemukan teknik jitu untuk memecahkan masalah krisis perangkat lunak. Berbagai teknik, metode, alat, proses diciptakan dan diklaim sebagai senjata pamungkas untuk memecahkan kasus ini. Mulai dari pemrograman terstruktur, pemrograman berorientasi objek, perangkat pembantu pengembangan perangkat lunak (CASE tools), berbagai standar, UML hingga metode formal diagung-agungkan sebagai senjata pamungkas untuk menghasilkan software yang benar, sesuai anggaran dan tepat waktu. Pada tahun 1987, Fred Brooks menulis artikel No Silver Bullet, yang berproposisi bahwa tidak ada satu teknologi

atau praktek yang sanggup mencapai 10 kali lipat perbaikan dalam produktivitas pengembangan perangkat lunak dalam tempo 10 tahun.

Sebagian berpendapat, *no silver bullet* berarti profesi rekayasa perangkat lunak dianggap telah gagal. Namun sebagian yang lain justru beranggapan, hal ini menandakan bahwa bidang profesi rekayasa perangkat lunak telah cukup matang, karena dalam bidang profesi lainnya pun, tidak ada teknik pamungkas yang dapat digunakan dalam berbagai kondisi.

2.1.3 Tujuan Rekayasa Perangkat Lunak

Secara umum tujuan RPL tidak berbeda dengan bidang rekayasa yang lain. Hal ini dapat kita lihat pada gambar 2.1 dibawah ini :



Sumber : Rekayasa Perangkat Lunak, Aunur R. Mulyanto, 2008

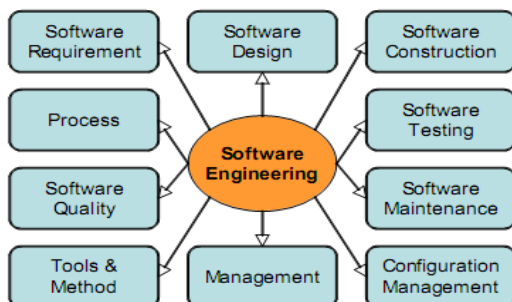
Gambar 2.1 Tujuan RPL

Dari Gambar di atas dapat diartikan bahwa bidang rekayasa akan selalu berusaha menghasilkan output yang kinerjanya tinggi, biaya rendah dan waktu penyelesaian yang tepat. Secara lebih khusus kita dapat menyatakan tujuan RPL adalah:

- Memperoleh biaya produksi perangkat lunak yang rendah.
- Menghasilkan perangkat lunak yang kinerjanya tinggi, andal dan tepat waktu.
- Menghasilkan perangkat lunak yang dapat bekerja pada berbagai jenis platform.
- Menghasilkan perangkat lunak yang biaya perawatannya rendah.

2.1.4 Ruang Lingkup

Sesuai dengan definisi yang telah disampaikan sebelumnya, maka ruang lingkup RPL dapat digambarkan pada gambar 2.2 berikut:



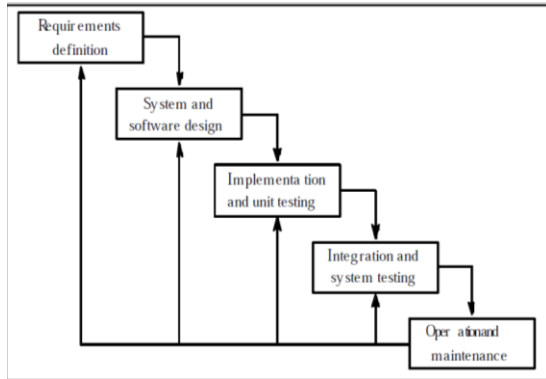
Sumber : Rekayasa Perangkat Lunak, Aunur R. Mulyanto, 2008

Gambar 2.2 Ruang Lingkup RPL

- Software Requirements berhubungan dengan spesifikasi kebutuhan dan persyaratan perangkat lunak .
- Software desain mencakup proses penampilan arsitektur, komponen, antar muka, dan karakteristik lain dari perangkat lunak.
- Software construction berhubungan dengan detail pengembangan perangkat lunak, termasuk algoritma, pengkodean, pengujian dan pencarian kesalahan .
- Software testing meliputi pengujian pada keseluruhan perilaku perangkat lunak.
- Software maintenance mencakup upaya-upaya perawatan ketika perangkat lunak telah dioperasikan.
- Software configuration management berhubungan dengan usaha perubahan konfigurasi perangkat lunak untuk memenuhi kebutuhan tertentu.
- Software engineering management berkaitan dengan pengelolaan dan pengukuran RPL, termasuk perencanaan proyek perangkat lunak.
- Software engineering tools and methods mencakup kajian teoritis tentang alat bantu dan metode RPL.
- Software engineering process berhubungan dengan definisi, implementasi pengukuran, pengelolaan, perubahan dan perbaikan proses RPL.
- Software quality menitik beratkan pada kualitas dan daur hidup perangkat lunak.

2.1.5 Modul Rekayasa Perangkat Lunak

a. Linear Sequential Model/ Waterfall Model
Model ini adalah model klasik yang bersifat sistematis, berurutan dalam membangun software. Berikut ini ada dua gambaran dari waterfall model. Sekalipun keduanya menggunakan nama-nama fase yang berbeda, namun sama dalam intinya. Fase-fase dalam Waterfall Model menurut referensi Sommerville pada gambar 2.3 berikut :



Sumber : Sommerville, Ian. "Software Engineering". 6th. Addison Wesley. 2001.
Gambar 2.3 Tahapan-Tahapan Model Waterfall

Keterangan:

1. Requirements analysis and definition : Mengumpulkan kebutuhan secara lengkap kemudian dianalisis dan didefinisikan kebutuhan yang harus dipenuhi oleh program yang akan dibangun. Fase ini harus dikerjakan secara lengkap untuk bisa menghasilkan desain yang lengkap.
2. System and software design : Desain dikerjakan setelah kebutuhan selesai dikumpulkan secara lengkap.
3. Implementation and unit testing : desain program diterjemahkan ke dalam kode-kode dengan menggunakan bahasa pemrograman yang sudah ditentukan. Program yang dibangun langsung diuji baik secara unit.
4. Integration and system testing : Penyatuan unit-unit program kemudian diuji secara keseluruhan (system testing).
5. Operation and maintenance : mengoperasikan program dilingkungannya dan melakukan pemeliharaan, seperti penyesuaian atau perubahan karena adaptasi dengan situasi sebenarnya.

Kekurangan yang utama dari model ini adalah kesulitan dalam mengakomodasi perubahan setelah proses dijalani. Fase sebelumnya harus lengkap dan selesai sebelum mengerjakan fase berikutnya.

Masalah dengan waterfall :

1. Perubahan sulit dilakukan karena sifatnya yang kaku.
2. Karena sifat kakunya, model ini cocok ketika kebutuhan dikumpulkan secara lengkap sehingga perubahan bisa ditekan sekecil mungkin. Tapi pada kenyataannya jarang sekali konsumen/pengguna yang bisa memberikan kebutuhan secara lengkap, perubahan kebutuhan adalah sesuatu yang wajar terjadi.
3. Waterfall pada umumnya digunakan untuk rekayasa sistem yang besar dimana proyek

dikerjakan di beberapa tempat berbeda, dan dibagi menjadi beberapa bagian sub-proyek.

NO	GAMBAR	NAMA	KETERANGAN
1		Actor	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan use case.
2		Dependency	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (independent) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (independent).
3		Generalization	Hubungan dimana objek anak (descendent) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (ancestor).
4		Include	Menspesifikasikan bahwa use case sumber secara eksplisit.
5		Extend	Menspesifikasikan bahwa use case target memperluas perilaku dari use case sumber pada suatu titik yang diberikan.
6		Association	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		System	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		Use Case	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
9		Collaboration	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		Note	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

2.2 Sekilas Tentang UML

2.2.1 Pengertian UML

Unified Modelling Language (UML) adalah sebuah “bahasa” yang telah menjadi standar dalam industri untuk menentukan, visualisasi, merancang dan mendokumentasikan artifact dari sistem software, untuk memodelkan bisnis dan sistem non software lainnya. *UML* merupakan suatu kumpulan teknik terbaik yang telah terbukti sukses dalam pemodelan sitem yang besar dan kompleks.

Dengan menggunakan *UML* kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun.

Jenis – Jenis Diagram UML

2.2.2 Use Case Diagram

a. Defenisi

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. *Use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan system untuk melakukan pekerjaan-pekerjaan tertentu. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya. *Use case diagram* dapat digunakan untuk :

1. Menyusun *requirement* sebuah sistem,
2. Mengkomunikasikan rancangan dengan klien, dan
3. Merancang *test case* untuk semua *feature* yang ada pada sistem.

Yang dapat dilihat pada table 2.1 berikut ini :

Tabel 2.1 Use case Diagram

Sumber : Pengenalan UML.pdf

2.2.3 Class Diagram

a. Defenisi

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan diantaranya :

1. Atribut/properti suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).
2. Menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain.

Class memiliki tiga area pokok :

1. Nama (dan stereotype)
2. Atribut
3. Metoda

Atribut dan metoda dapat memiliki salah satu sifat berikut :

1. *Private*, tidak dapat dipanggil dari luar *class* yang bersangkutan
2. *Protected*, hanya dapat dipanggil oleh *class* yang bersangkutan dan anak-anak yang mewarisinya
3. *Public*, dapat dipanggil oleh siapa saja

Class dapat merupakan implementasi dari sebuah *interface*, yaitu *class* abstrak yang hanya memiliki metoda. *Interface* tidak dapat langsung diinstansiasikan, tetapi harus diimplementasikan dahulu menjadi sebuah *class*. Dengan demikian *interface* mendukung resolusi metoda pada saat *run-time*. *Class* dapat dikelompokkan menjadi *package*. Kita juga dapat membuat diagram yang terdiri atas *package*.

Hubungan Antar *Class*

1. Asosiasi, yaitu hubungan statis antar *class*. Umumnya menggambarkan *class* yang memiliki atribut berupa *class* lain, atau *class* yang harus mengetahui eksistensi *class* lain. Panah *navigability* menunjukkan arah *query* antar *class*.
2. Agregasi, yaitu hubungan yang menyatakan bagian (“terdiri atas..”).
3. Pewarisan, yaitu hubungan hirarkis antar *class*. *Class* dapat diturunkan dari *class* lain dan mewarisi semua atribut dan metoda *class* asalnya dan menambahkan fungsionalitas baru, sehingga ia disebut anak dari *class* yang diwarisinya. Kebalikan dari pewarisan adalah generalisasi.
4. Hubungan dinamis, yaitu rangkaian pesan (*message*) yang di-*passing* dari satu *class* kepada *class* lain. Hubungan dinamis dapat digambarkan dengan menggunakan *sequence diagram* yang akan dijelaskan kemudian.

Yang dapat dilihat pada table 2.2 berikut ini :

Tabel 2.2 Class Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>)

			akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		<i>Association</i>	Apa yang menghubungkan objek satu dan lainnya

2.2.4 Activity Diagram

a. Defenisi

Activity diagrams menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir.

1. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.


2. *Activity diagram* merupakan *state diagram* khusus, di mana sebagian besar adalah *action* dan sebagian besar transisi di-trigger oleh selesainya *state* sebelumnya (*internal processing*).

3. *Activity diagram* dapat dibagi menjadi beberapa *object swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu.

Yang dapat dilihat pada table 2.3 berikut ini :

Tabel 2.3 Activity Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4		<i>Activity Final Node</i>	Bagaimana objek dibentuk dan dihancurkan

5		Fork Node	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran
---	---	-----------	--

Sumber : Pengenalan UML.pdf

			tentang aktifitas yang terjadi
--	--	--	--------------------------------

Sumber : Pengenalan UML.pdf

2.2.5 Sequence Diagram

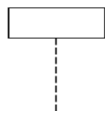


a. Defenisi

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, display, dan sebagainya) berupa message yang digambarkan terhadap waktu.

1. Sequence diagram terdiri antar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).
2. Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah event untuk menghasilkan output tertentu.

Yang dapat dilihat pada table 2.4 berikut ini :

Tabel 2.4 Sequence Diagram




NO	GAMBAR	NAMA	KETERANGAN
1		LifeLine	Objek entity, antarmuka yang saling berinteraksi.
2		Message	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
3		Message	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi

2.2.6 Collaboration Diagram

a. Defenisi

Collaboration diagram juga menggambarkan interaksi antar objek seperti sequence diagram, tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian message. Setiap message memiliki sequence number, di mana message dari level tertinggi memiliki nomor 1. Messages dari level yang sama memiliki prefiks yang sama, yang dapat dilihat pada table 2.5 berikut ini :

Tabel 2.5 Collaboration Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		Kelas peran	peran Kelas menjelaskan bagaimana objek berperilaku. Gunakan simbol UML objek untuk mengilustrasikan peran kelas, tetapi tidak daftar atribut objek.
2		Asosiasi peran	Asosiasi menggambarkan bagaimana peran asosiasi akan berperilaku diberikan situasi tertentu. Anda dapat menarik peran asosiasi menggunakan baris sederhana dilabeli dengan stereotip.
3		Pesan	Tidak seperti urutan diagram, diagram kolaborasi tidak memiliki cara eksplisit untuk menunjukkan waktu dan bukannya jumlah pesan dalam rangka eksekusi. urutan penomoran dapat menjadi bersarang menggunakan sistem desimal Dewey.

Sumber : Pengenalan UML.pdf

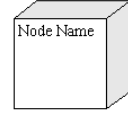
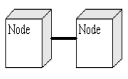
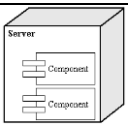
2.2.7 Deployment Diagram

a. Defenisi

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, yang dapat dilihat pada table 2.6 berikut ini :

Table 2.6 Deployment Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		Komponen	node adalah sumber daya fisik yang menjalankan kode komponen.
2		Asosiasi	Asosiasi mengacu pada koneksi fisik antara node, seperti Ethernet.
3		Komponen dan Nodes	komponen di dalam node yang menyebarkan mereka.

Sumber : *Pengenalan UML.pdf*

2.3 Konsep Dasar PHP

2.3.1 Sejarah PHP

Pada awalnya PHP merupakan kependekan dari *Personal Home Page* (Situs personal). PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama *Form Interpreted* (FI), yang

wujudnya berupa sekumpulan skrip yang digunakan untuk mengolah data formulir dari web.

Selanjutnya Rasmus merilis kode sumber tersebut untuk umum dan menamakannya PHP/FI. Dengan perilsan kode sumber ini menjadi sumber terbuka, maka banyak pemrogram yang tertarik untuk ikut mengembangkan PHP.

Pada November 1997, dirilis PHP/FI 2.0. Pada rilis ini, *interpreter* PHP sudah diimplementasikan dalam program C. Dalam rilis ini disertakan juga modul-modul ekstensi yang meningkatkan kemampuan PHP/FI secara signifikan.

Pada tahun 1997, sebuah perusahaan bernama Zend menulis ulang interpreter PHP menjadi lebih bersih, lebih baik, dan lebih cepat. Kemudian pada Juni 1998, perusahaan tersebut merilis interpreter baru untuk PHP dan meresmikan rilis tersebut sebagai PHP 3.0 dan singkatan PHP diubah menjadi akronim berulang *PHP: Hypertext Preprocessing*.

Pada pertengahan tahun 1999, Zend merilis interpreter PHP baru dan rilis tersebut dikenal dengan PHP 4.0. PHP 4.0 adalah versi PHP yang paling banyak dipakai pada awal abad ke-21. Versi ini banyak dipakai disebabkan kemampuannya untuk membangun aplikasi web kompleks tetapi tetap memiliki kecepatan dan stabilitas yang tinggi.

Pada Juni 2004, Zend merilis PHP 5.0. Dalam versi ini, inti dari interpreter PHP mengalami perubahan besar. Versi ini juga memasukkan model pemrograman berorientasi objek ke dalam PHP untuk menjawab perkembangan bahasa pemrograman ke arah paradigma berorientasi objek.

PHP adalah salah satu bahasa pemrograman di Internet. PHP bersifat cepat, gratis dan murah (gratis), selain itu PHP mendukung penggunaan database seperti MySQL, PostgreSQL, mSQL, Oracle etc.

Untuk dapat menjalankan PHP melalui browser, maka anda diharuskan terlebih dahulu menginstall web server (misalnya Apache, PWS, IIS) lalu menginstall PHP, sedangkan untuk menjalankan MySQL anda tidak perlu menginstall web server, hanya saja jika ingin dijalankan melalui browser, maka anda harus menginstall web server. Prosedur penginstallan web server, PHP dan MySQL dapat anda baca pada manual masing – masing. Adapun manual dan instaln Apache.

Jika anda tidak ingin direpotkan dengan prosedur penginstalan PHP, anda dapat menjalankan PHP di direktori *public_html* anda di *s.ee.itb.ac.id*. Hal itu dapat dilakukan karena PHP bersifat *server side* artinya script PHP di jalankan di server, bukan di computer client. Jadi semakin banyak client yang menjalankan script PHP di suatu situs, maka beban yang ditanggung oleh server semakin bertambah.

Sintaks – sintaks dalam PHP banyak yang sama dengan C++, dalam hal ini penulis tidak memberikan penjelasan penggunaan sintaks – sintaks tersebut mengingat para cakru telah memperoleh pengajaran tersebut di dalam kuliah maupun praktikum EL – 207 Komputer dan Pemrograman.

1. PHP singkatan dari “PHP Hypertext Preprocessor”
2. Dirancang untuk membentuk web dinamis
3. Memiliki fungsi yang sama dengan ASP, Cold Fusion atau Perl
4. Lahir dimulai dari Rasmus Lerdorf (1994) dari skrip Perl
5. Dapat berkomunikasi dengan database (misalnya mySQL)
6. PHP bersifat FREE
7. Awalnya dirancang untuk bekerja dengan Web Server Apache
8. Namun akhirnya dapat bekerja dengan PWS atau IIS
9. Kode PHP berposisi sebagai suatu TAG dalam dokumen HTML

Yaitu tag:

```
<?php  
    isi kode php  
>
```

Contoh program

Program Hello World

Program [Hello World](#) yang ditulis menggunakan PHP adalah sebagai berikut:

```
<?php  
    echo "Hello World";  
>
```

Program bilangan Fibonacci

Berikut ini adalah contoh program yang relatif lebih kompleks yang ditulis dengan menggunakan PHP. Contoh program ini adalah program untuk menampilkan 20 bilangan pertama dari deret [bilangan Fibonacci](#).

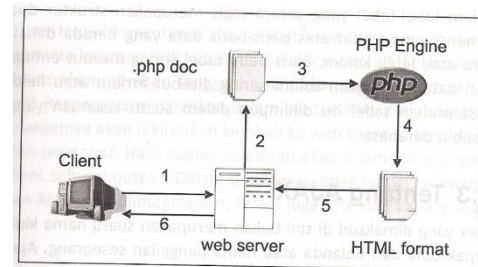
```
<?php
```

```
function fibonacci_seq( $panjang ) {  
    for( $l = array(0,1), $i = 2, $x = 0; $i < $panjang;  
        $i++ )  
        $l[] = $l[$x++] + $l[$x];  
    return $l;  
}
```

```
fibonacci_seq(20);  
// Angka "20" dapat diganti sesuai keinginan  
>
```

2.3.2 PHP Hypertext Preprocessor (PHP)

PHP merupakan bahasa pemrograman yang paling populer dan banyak digunakan untuk pemrograman web. PHP disebut juga pemrograman *server side*, artinya program dijalankan pada server. PHP sebenarnya merupakan program yang berjalan pada platform LINUX. Program ini bersifat freeware. Berikut struktur pembacaan dalam web server pada gambar 2.4 berikut ini :



Sumber : *Amazing News Website with PHP, AJAX, and MYSQL*, Mochamad Joko Adi Wirawan, 2008

Gambar 2.4 Struktur Pembacaan Web Server

Setiap script PHP harus didahului dengan pembuka PHP (<?php atau <?) dan diakhiri dengan penutup (>). Berikut contoh sederhana script PHP.

```
<? php  
    Echo "I will Win";  
>
```

Output script diatas adalah **I WILL WIN**.

2.3.3 Dasar PHP

PHP dijalankan dalam file berekstensi .php, .php3 atau .phtml, itu tergantung dengan settingan PHP anda, tetapi secara umum ekstensi file PHP adalah .php.

Kode PHP menyatu dengan tag – tag HTML dalam satu file. Kode PHP diawali dengan tag <? atau <?php dan ditutup dengan >.

Contoh :

```
<?  
    phpinfo();  
>
```

Struktur penulisan dalam PHP, sama seperti dalam C++, yaitu setiap pernyataan diakhiri oleh semicolon (;) dan bersifat *case sensitive* untuk penulisan nama variabel. Cara penulisan komentar dalam PHP juga sama dengan C++.

Contoh :

```
<?
  $nama = "divisi komputer hme itb"; // Huruf
kecil semua
  $NAMA = "DIVISI KOMPUTER HME
ITB"; // Huruf besar semua
  print "$nama □ variable menggunakan huruf
kecil.<br>";
  print "$NAMA □ variable menggunakan
huruf besar.";
?>
```

2.3.4 MySQL

MySQL merupakan database *multiuser* yang menggunakan bahasa *Structure Query Language* (SQL). MySQL dalam operasi client-server, melibatkan server-daemon MySQL di sisi server dan berbagai macam program, serta library yang berjalan di sisi client. MySQL mampu menangani data yang cukup besar. SQL adalah bahasa standar yang digunakan untuk mengakses server database. Bahasa ini awalnya dikembangkan IBM, tetapi telah diadopsi dan digunakan sebagai standar industry. Dengan menggunakan SQL, proses akses database menjadi lebih *friendly*. Dalam konteks bahasa SQL, informasi umumnya tersimpan dalam table-tabel yang secara logis merupakan struktur dua dimensi yang terdiri atas baris-baris data yang berada dalam satu atau lebih kolom. Baris pada table sering disebut entitas dari data, sedangkan kolom sering disebut atribut atau field. Keseluruhan table itu dihimpun dalam suatu kesatuan yang disebut database.

2.4 Sekilas Tentang CRM

2.4.1 Pengertian CRM

CRM merupakan sebuah pendekatan baru dalam mengelola hubungan korporasi dan pelanggan pada level bisnis sehingga dapat memaksimalkan komunikasi, pemasaran melalui pengelolaan berbagai kontak yang berbeda dengan pelanggan. Pendekatan ini memungkinkan untuk mempertahankan pelanggan dan memberikan nilai tambah terus menerus pada pelanggan, selain juga memperoleh keuntungan yang berkelanjutan.

Hal yang perlu dipahami adalah bahwa dari luar, pelanggan yang berinteraksi dengan perusahaan hanya memahami bisnis yang dilakukan oleh perusahaan sebagai satu entitas, tidak lebih; meskipun pelanggan juga berinteraksi dengan sejumlah pekerja yang berbeda peran dan departemennya. Semua itu tetap dianggap sebagai satu kesatuan. Dengan CRM, dukungan pada proses bisnis, informasi tentang pelanggan dan interaksinya dapat dimasukkan, disimpan, diakses oleh semua staf pada berbagai unit kerja dengan tujuan untuk meningkatkan layanan yang diberikan pada pelanggan, dan menggunakan informasi kontak pelanggan untuk target pemasaran.

CRM (customer relationship management) mengkombinasikan kebijakan, proses, dan strategi

yang diterapkan organisasi menjadi satu kesatuan yang digunakan untuk melakukan interaksi dengan pelanggan dan juga untuk menelusuri informasi pelanggan. Pada era saat ini, implementasi CRM selalu akan menggunakan teknologi informasi untuk menarik pelanggan baru yang menguntungkan, hingga mereka memiliki keterikatan pada perusahaan.

Konsep ini juga banyak digunakan sebagai sebuah terminologi pada industri informasi sebagai sebuah metodologi, piranti lunak, dan bahkan sebagai kapabilitas internet yang membantu korporasi mengelola hubungan pelanggan. Piranti ini memungkinkan korporasi untuk menjelaskan hubungan secara cukup rinci sehingga manajemen, staf penjualan, staf pelayanan, dan bahkan pelanggan dapat secara langsung mengakses informasi, menyesuaikan kebutuhan antara kebutuhan dan produk, pengingat pelanggan pada layanan tertentu, dan sebagainya. Dalam pengembangan piranti lunak CRM harus dilakukan pendekatan yang holistik. Terkadang inisiatif implementasi piranti lunak CRM gagal dilakukan karena hanya terbatas pada instalasi piranti saja, tanpa memperhatikan konteks, dukungan dan pemahaman staf, dan pemanfaatan sepenuhnya sistem informasi.

Banyak aspek yang tercakup dalam CRM. Aspek tersebut pada umumnya akan berhubungan langsung dengan salah satu aspek berikut:

1. Operasi Front office yang langsung berinteraksi dengan pelanggan seperti ketemu langsung, panggilan telepon, e-mail, layanan online, dll.
2. Operasi Back office yang sangat berpengaruh pada aktivitas pada layanan di front office seperti bagian pembayaran, perawatan, perencanaan, pemasaran, dll.
3. Hubungan bisnis, yaitu interaksi dengan perusahaan dan rekanan lain seperti suppliers/ vendors, outlet pengecer dan distribusi, jaringan industri. Jaringan eksternal ini akan mendukung aktivitas di front dan back office.
4. Data kunci dalam CRM dapat dianalisis dengan tujuan melakukan perencanaan kampanye pada target pemasaran, memahami strategi bisnis, dan memutuskan keberhasilan aktivitas CRM seperti pangsa pasar, karakteristik pelanggan, pendapatan dan keuntungan.

Implementasi CRM akan memberikan dampak manfaat bagi korporasi seperti akan memungkinkan untuk memperoleh keseimbangan dalam bagaimana menggunakan sumber daya untuk pemasaran agar mendapat nilai lebih pada segmen tertentu, meningkatkan interaksi pelanggan yang akan mendukung pengalaman branding pada pelanggan, mendongkrak produktivitas penjualan

melalui pemetaan proses menggunakan teknologi baru, menetapkan tujuan penjualan dan menentukan reward untuk pencapaiannya, fokus pada gab antara harapan pelanggan dan pengalaman layanan saat ini, dengan menggunakan alat analisis akan dapat ditingkatkan pemahaman organisasi tentang pelanggan.

2.4.2 Strategi dan Implementasi CRM

Sasaran untuk menjalankan strategi CRM harus mempertimbangkan situasi spesifik yang dihadapi perusahaan dan juga kebutuhan dan harapan pelanggan. Informasi yang didapat melalui inisiasi CRM dapat mendukung pengembangan strategi pemasaran dengan mengembangkan pengetahuan pada area-area tertentu seperti identifikasi segmen pelanggan, peningkatan retensi pelanggan, perbaikan produk yang disampaikan, juga identifikasi pelanggan blue chip (yang paling menguntungkan). Sasaran ini dilakukan dengan melakukan manipulasi informasi yang terkait yang selanjutnya dimanfaatkan bersama untuk ditransformasikan menjadi pengetahuan. Pengetahuan ini merupakan pengetahuan yang memungkinkan perusahaan dapat lebih memahami pelanggan, dan digunakan untuk menyesuaikan kapabilitas organisasi agar dapat memberikan nilai yang lebih baik bagi pelanggan. Strategi CRM bervariasi dalam besaran, kompleksitas, cakupannya. Hal ini akan berpengaruh pada tipe apa CRM dikembangkan. Strategi CRM yang efektif akan fokus pada berbagai saluran. Perusahaan harus secara efektif mengelola saluran ini untuk perbaikan pelayanan.

Kegagalan berbagai proyek CRM terutama terkait dengan kualitas dan availabilitas data pelanggan. Data yang baik merupakan isu penting. Ketika perusahaan yang menggunakan CRM untuk menelusuri siklus hidup pendapatan, pembiayaan, margin keuntungan, dan interaksi dengan pelanggan secara individual, maka semua ini harus tergambar dengan jelas dalam suatu bisnis proses. Data yang digunakan harus dapat diambil dari berbagai sumber data yang ada di tiap departemen atau bidang yang ada di perusahaan. Sistem yang komprehensif dengan struktur yang telah didefinisikan dengan baik akan meningkatkan kualitas data yang dipakai dalam pengambilan keputusan.

2.4.3 Manfaat CRM

CRM memberikan beberapa manfaat sebagai berikut :

1. Membantu perusahaan / organisasi sehingga memungkinkan bagian marketing untuk mengidentifikasi secara tepat pelanggan / relasi mereka, mengatur bagian *marketing* dengan tujuan dan

sasaran yang jelas serta meningkatkan kualitas bagian *marketing*.

2. Membantu perusahaan / organisasi untuk meningkatkan kinerja dengan mengoptimalkan *sharing* informasi oleh beberapa pegawai dan mempersingkat proses yang ada.
3. Memungkinkan bentuk hubungan personal dengan pelanggan / relasi dengan tujuan meningkatkan kepuasan pelanggan / relasi dan memaksimalkan keuntungan yang didapat; mengidentifikasi pelanggan / relasi yang potensial dan memberikan mereka *service* yang lebih dibandingkan pelanggan / relasi lainnya.
4. Memberikan informasi kepada pegawai lainnya untuk mengetahui secara pasti pelanggan / relasi mereka, mengetahui kebutuhannya dan membangun hubungan yang efektif antara perusahaan / organisasi, pelanggan / relasi dan partner pendukung lainnya.

Dengan kata lain, CRM dikembangkan untuk memberikan tingkat kepuasan pelanggan / relasi sehingga akan tetap kembali dengan *service* yang perusahaan / organisasi berikan dan membuka peluang bagi perusahaan / organisasi menjadi pemenang dalam berbisnis.

2.5 Sekilas Tentang WAP

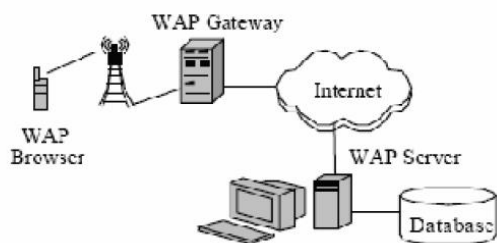
2.5.1 Pengertian WAP

Pengertian WAP (*Wireless application protocol*) adalah suatu protocol aplikasi yang memungkinkan internet dapat diakses oleh ponsel dan perangkat wireless lainnya. Wap membawa informasi secara online melewati internet langsung menuju ponsel atau klien WAP lainnya. Dengan adanya WAP berbagai informasi dapat diakses setiap saat hanya dengan menggunakan ponsel. Ada tiga bagian dalam WAP yaitu perangkat wireless yang mendukung WAP, WAP gateway sebagai perantara dan server sebagai sumber dokumen. Dokumen yang berada pada web server adalah dokumen HTML maupun WML. Dokumen WML khusus ditampilkan melalui browser dari perangkat WAP, sedangkan dokumen HTML yang seharusnya ditampilkan melalui web browser, sebelum dibaca melalui browser WAP diterjemahkan dulu oleh gateway agar dapat menyesuaikan dengan perangkat WAP.

Saat browser ingin meminta sebuah informasi yang ada di server, ponsel harus melewati WAP gateway dulu, begitu juga sebaliknya. Proses pengiriman informasi dari ponsel ke WAP gateway

atau sebaliknya menggunakan jaringan komunikasi nirkabel (wireless) yang masih memiliki keterbatasan, terutama pada kecilnya bandwidth yang ada. Kecilnya bandwidth tersebut tidak cocok jika dipergunakan untuk memproses informasi lewat protokol HTTP. Protokol HTTP berfungsi untuk mengatur pengiriman informasi dari client keserver dan sebaliknya. Untuk mengatasi kesenjangan ini, diciptakan WAP gateway. Fungsi WAP gateway adalah untuk meneruskan permintaan informasi dari ponsel menuju server lewat HTTP request dan sebaliknya dari server menuju ponsel melalui respon.

Dapat dilihat ilustrasi arsitektur umum WAP pada gambar 2.5 dibawah ini :



Sumber : *Pengertian WAP, pdf*

Gambar 2.5 : ilustrasi arsitektur umum

WAP

2.5.2 Sejarah WAP

Awal kemunculan WAP dimulai dari riset yang dilakukan *Ericson* pada tahun 1995 dengan mengembangkan protokol umum yang disebut *Intelligent Terminal Transfer Protokol (ITIP)* yang memungkinkan adanya nilai tambah pada handphone. Selama tahun 1996-1997, *Motorola*, *Nokia* dan *Planet Unwired* (sekarang *phone.com*) meluncurkan konsep serupa. *Planet Unwired* memperkenalkan *Handheld Device Markup Language (HDML)* dan *Handheld Device Transport Protocol (HDTP)* sama halnya dengan *HTML* yang digunakan pada *www*. *HDML* digunakan untuk memperlihatkan isi website atau sebagai user interface dan sangat sesuai digunakan untuk aplikasi wireless internet access dengan limit transfer data yang sangat kecil.

Bulan maret tahun 1997 *Nokia* memperkenalkan konsep *Smart Messaging* yang khusus dirancang untuk *GSM*, dimana komunikasi antara handphone dengan internet dilakukan melalui sms dan pemograman bahasa yang disebut *Tagget Text Markup Language (TTML)* sama dengan *HDML*. Maka pada 26 juni 1997 keempat perusahaan tersebut sepakat membentuk forum *Wireless Aplication Protocol (WAP)*.

2.5.3 Keuntungan dan Kelemahan WAP

Adapun keuntungan dari WAP adalah sebagai berikut :

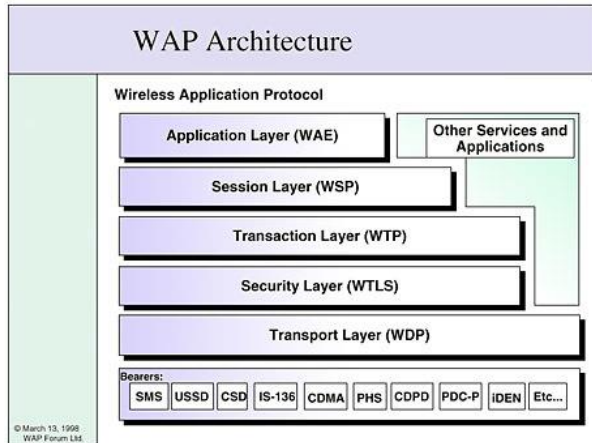
1. Tidak adanya kepemilikan metode dalam meangakses internet dengan standar WAP baik pada isi maupun layanan.
2. Network yang independen karena WAP bekerja pada seluruh jaringan seluler yang ada.
3. Metode WAP telah di adopsi hampir 95% produsen telepon seluler diseluruh dunia alam memanfaatkan *Wireless Internet Access*.
4. WAP adalah suatu standar protokol dan aplikasinya yakni WAP browser yang dapat digunakan pada seluruh sistem operasi terkenal termasuk *Palm OS*, *EPOC*, *Windows SE*, *FLEXOS*, *OS/9*, *java OS* dan sebagainya.
5. Dengan menggunakan teknologi *GPRS*, perhitunagn akses dihitung berdasarkan jumlah bit yang terkoneksi yang harganya relatif murah sekitar Rp.30 per kilo Byte tergantung masing-masing operator seluler.

Adapun kelemahan dari WAP adalah konfigurasi telepon seluler untuk service WAP masih termasuk sulit.

2.5.4 Konsep Model Jaringan Pada Teknologi WAP

Protokol dapat didefinisikan pada suatu cara atau aturan yang dibakukan untuk melakukan komunikasi antara satu peralatan jaringan (bias berupa computer, switch, hub dan lain-lain) dengan peralatan jaringan lainnya. Setiap protokol yang dibuat untuk aplikasi jaringan biasanya mengikuti standar untuk mendesainnya. Desain atau standar protokol tersebut merupakan arsitektur protokol yang ditetapkan secara internasional oleh *ISO (International Standard Organization)*.

Arsitektur tersebut dinamakan *Open System Reference Standar Architecture* seperti terlihat komponen arsitektur WAP pada gambar 2.6 dibawah ini :



Sumber : Pengertian WAP, pdf

Gambar 2.6 : komponen arsitektur

WAP

Keterangan :

1. WAE (*Wireless Application Environment*)
WAE mengadopsi pemodelan WWW, format content-nya serupa dengan standar internet, WAE User agent (perangkat wireless / emulator), format elemen-elemen seperti WML, WMLS, dan WBMP sebagai pembentuk content.
2. WSP (*Wireless Session Protocol*)
Layer untuk menangani inisialisasi protokol, konfigurasi dan error condition (seperti gagalnya hubungan karena mobile station berada diluar layanan coverage area). Selain itu juga mengatur pertukaran content aplikasi antar client dan server.
3. WTP (*Wireless Transaction Protocol*)
Layer untuk mengatur pertukaran data (*response/request*) dari suatu aplikasi, misalnya saat browsing, mekanisme *response/request* ini disebut *transaction*. Selain itu menjamin transaksi tersebut tidak mengalami *loss connection* serta menghindari duplikasi pesan.
4. WTLS (*Wireless Transport Layer Security*)
Layer sebagai interface untuk pengaturan (*creating and terminating*) koneksi yang aman, *privacy*, *data integrity*, autentifikasi, *handshaking*, *denial of service protection*, SSL (*Secure Socket Layer*) dan *compression*.
5. WDP (*Wireless Datagram Protocol*)
Membuat dukungan terhadap berbagai tipe network, pengalamatan aplikasi dengan *port number*, *optional error detection*.

Bearers sendiri merupakan teknologi pendukung untuk mentransmisikan data dengan signal-signal radio seperti routing data kepada penerima (*receiver*) yang cocok. Teknologi tersebut contohnya adalah CSD, CDPD, GSM, SMS, GPRS, FDMA, TDMA, CDMA, 3G dan lain lain.

3.3 Metode Penelitian

3.3.1 Metode Pengumpulan Data

Untuk menyelesaikan permasalahan yang ada, metode pengumpulan data yang saya lakukan sebagai penulis adalah dengan cara:

1. Tinjauan Lapangan (*Field Reseach*)

Pengumpulan data dan informasi dengan melakukan pertemuan dengan pimpinan atau pemilik *The Aliga Hotel*. Kemudian melakukan wawancara dengan beberapa pertanyaan yang berhubungan dengan topik yang dibahas kepada pihak-pihak bersangkutan. Penulis juga melakukan observasi dengan melakukan peninjauan dan pengamatan secara langsung sistem yang sedang berjalan di perusahaan tersebut serta mengumpulkan data atau informasi yang terkait dengan pelayanan-pelayanan yang tersedia. Penulis juga melakukan *sampling* dan *investigasi* yaitu dengan cara meminta contoh sampel untuk dokumen yang digunakan oleh perusahaan bersangkutan dengan masalah yang diteliti. Seperti brosur yang bersangkutan dengan pelayanan hotel..

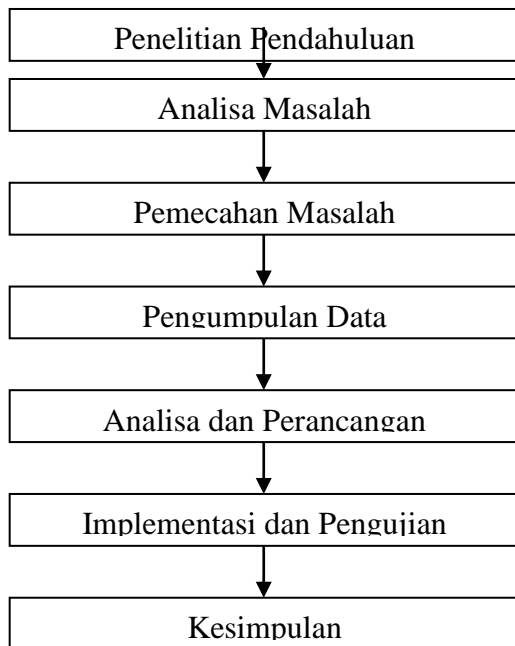
2. Tinjauan Perpustakaan (*Library Reseach*)

Dalam metode ini dilakukan pembacaan dan pembelajaran sumber-sumber yang akan mendukung penulisan penelitian ini. Sumber tersebut dapat berupa buku-buku dan hasil penelitian. Untuk hasil penelitian dapat berupa laporan penelitian, jurnal ilmiah dan penelitian. Termasuk juga kategori bahan-bahan yang dipublikasikan secara *on-line* (menggunakan internet).

3. Penelitian Laboratorium (*Laboratory Reseach*).

Penelitian ini dilakukan untuk mengumpulkan data yang dibantu dengan seperangkat *Personal Computer* (PC). Serta berbagai perangkat lunak yang akan membantu dalam penelitian.

3.3.2 Metode Perancangan



Gambar 3.1 Model Metode Perancangan

Pada gambar diatas memberikan bentuk dan urutan dari apa yang akan dilakukan dalam penelitian ini. Apa yang harus dilakukan terlebih dahulu, tidak boleh melewati langkah yang lainnya sebelum langkah sebelumnya selesai, semuanya harus berurutan. Untuk lebih jelasnya, penulis akan menguraikan satu persatu dari blok diagram pada gambar diatas, seperti dibawah ini.

1. Penelitian Pendahuluan

Penelitian pendahuluan ini merupakan langkah pertama dalam melakukan suatu penelitian. Penelitian ini dilakukan pada *The Aliga Hotel*. Pada penelitian pendahuluan ini yang harus dilakukan adalah mencari tahu bagaimana sistem yang ada saat ini di tempat dimana penelitian dilakukan yang bertujuan untuk mencari tahu dan menganalisa semua permasalahan yang ada. Penelitian pendahuluan ini dilakukan dengan cara mendatangi langsung objek yang menjadi pokok dari penelitian.

2. Analisa Masalah

Setelah dilakukan penelitian pendahuluan seperti yang dijelaskan diatas, maka dilakukan analisa terhadap masalah yang dihadapi oleh *The Aliga Hotel*,

sehingga didapatkan beberapa masalah yang akan dihadapi.

3. Pemecahan Masalah

Setelah dilakukan analisa masalah dari hasil penelitian pendahuluan, maka penulis harus menentukan hipotesis dari permasalahan tersebut. Dari beberapa hipotesis yang dikemukakan, penulis harus menentukan pemecahan masalah yang bisa menyelesaikan permasalahan tersebut.

4. Pengumpulan Data

Langkah selanjutnya yang harus dilakukan adalah mengumpulkan data yang mendukung seluruh kebutuhan sistem.

5. Analisa dan Perancangan

Setelah semua data yang dibutuhkan telah terkumpul dan telah dilengkapi, langkah selanjutnya adalah melakukan analisa terhadap kebutuhan variable sistem yang akan dibangun, baik dari segi hardware maupun software. Analisa juga dilakukan terhadap sistem yang sedang berjalan pada *The Aliga Hotel*.

Analisa ini bertujuan untuk mengetahui tingkat kebutuhan sistem dan seperti apa sistem itu akan dibangun. Disini juga akan didapat apa input dan apa output dari sistem tersebut. Sistem ini menggunakan Unified Modelling Language (UML) sebagai analisa pemodelan dari sistem yang akan dirancang. Analisa pemodelan sistem ini menggunakan beberapa diagram dari Unified Modelling Language (UML), yaitu:

a. Use Case Diagram

Use case adalah rangkaian/ uraian sekelompok yang saling terkait dan membentuk sistem secara teratur yang dilakukan atau diawasi oleh sebuah actor.

b. Class Diagram

Class diagram menggambarkan struktur dan deskripsi class, package dan objek beserta hubungan satu sama lain seperti containment, pewaris, asosiasi dan lain-lain.

c. Activity Diagram

Activity diagram merupakan state diagram khusus, dimana sebagian besar state adalah action dan sebagian besar transisi di-trigger oleh selesai state sebelumnya (*internal processing*).

d. Statechart Diagram

Statechart diagram menggambarkan transisi dan perubahan keadaan (dari satu state ke state lainnya) suatu objek pada sistem akibat dari stimuli yang diterima.

e. Sequence Diagram

Sequence diagram (*diagram urutan*) adalah suatu diagram yang memperlihatkan dan menampilkan interaksi-interaksi antar objek didalam sistem yang disusun pada sebuah urutan atau rangkaian waktu.

f. Collaboration Diagram

Collaboration diagram adalah perluasan dari objek dan diagram (objek diagram menunjukkan objek-objek dan hubungannya antara satu dengan yang lain).

g. Deployment Diagram

Deployment diagram menggambarkan detail bagaimana komponen di-deploy dalam infrastruktur sistem, dimana komponen akan terletak (pada mesin, server atau piranti keras), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi *server* dan hal-hal lain yang bersifat fisik.

Setelah analisa selesai dilakukan, langkah selanjutnya adalah melakukan perancangan terhadap sistem yang akan dibangun.

6. Implementasi dan Pengujian

Implementasi dapat dilakukan setelah analisa dan perancangan selesai dilakukan. Dengan menampilkan/memperlihatkan hasil dari analisa dan perancangan dari sistem yang dirancang.

7. Kesimpulan

Kesimpulan berisi tentang ringkasan dari seluruh isi penelitian dari tahap awal memulai penelitian sampai akhir penelitian.

4. HASIL DAN PEMBAHASAN

Untuk memperoleh data atau informasi dalam studi kasus tertentu, perlu dilakukan kegiatan pengumpulan data yang berguna sebagai penunjang. Teknik pengumpulan data terdiri dari;

- Berdasarkan *setting* (dengan mewawancarai *owner* perusahaan dan karyawan-karyawan).
- Berdasarkan Sumber data (sumber primer: sumber yang langsung memberikan data dan sumber sekunder: sumber yang tidak langsung memberikan data).
- Berdasarkan teknik pengumpulan data dibagi menjadi: observasi, wawancara, dokumentasi, dan triangulasi/gabungan.

Dalam mengumpulkan data, penulis menggunakan teknik wawancara dan dokumentasi yaitu pertemuan antara pengumpul data dengan *owner* perusahaan dan karyawan-karyawannya untuk bertukar informasi agar data yang diperoleh dapat diolah dan sumber datanya merupakan sumber data primer, yaitu data yang diberikan secara langsung.

4.1. Struktur Database

Aplikasi pemesanan fasilitas ini menggunakan MySQL sebagai *Database Management System (DBMS)*. Diperlukan satu *database* yang diberi nama **aligahotel** serta tabel-tabelnya. Tabel-tabel *database* yang digunakan untuk penyimpanan data pada sistem informasi terdiri dari beberapa tabel.

Berikut tabel-tabel *database aligahotel* yang diperlukan dalam penyelesaian pemesanan fasilitas hotel ini :

1. Tabel Users

Tabel *user* digunakan untuk menampung data-data lengkap yang berhubungan dengan *member* dan admin pada *The Aliga Hotel* .

Keterangan diatas dapat dilihat pada tabel 4.1 berikut ini :

Tabel 4.1 Tabel Users

No	Field	Type	Keterangan
1	username	Varchar (50)	Primary Key
2	password	Varchar (50)	
3	nama_lengkap	Varchar (100)	
4	email	Varchar (100)	
5	no_telp	Varchar (20)	
6	level	Varchar (20)	
7	blokir	enum ('Y','N')	
8	id_session	Varchar (100)	

2. Tabel Pemesanan

Tabel pemesanan digunakan untuk menampung data pemesanan fasilitas pelanggan. Keteranganannya dapat dilihat pada tabel 4.2 berikut ini:

Tabel 4.2 Tabel Pemesanan

No	Field	Type	Keterangan
1	id_pemesanan	int (5)	Primary Key, Auto Increment
2	Tgl_pemesanan	Date	
3	id_hotel	int (5)	
4	Username	varchar (20)	
5	Tgl	Date	
6	id_jam	int (11)	
7	Jml	varchar (3)	
8	Status	varchar (15)	

3. Tabel Jam

Tabel jam digunakan untuk menampung data jam pemesanan. Keteranganannya dapat dilihat pada tabel 4.3 berikut ini:

Tabel 4.3 Tabel Jam

No	Field	Type	Keterangan
1	id_jam	int (11)	Primary Key, Auto Increment
2	jam	time	

4. Tabel Hotel

Tabel hotel digunakan untuk menampung data judul, keterangan, gambar dan biaya pemesanan. Keteranganannya dapat dilihat pada tabel 4.4 berikut ini:

Tabel 4.4 Tabel Hotel

No	Field	Type	Keterangan
1	id_hotel	Int(5)	Primary Key, Auto Increment
2	judul	Varchar (25)	
3	biaya	Int (5)	
4	Keterangan	longtext	
5	gambar	Varchar (10)	

5. Tabel Detail

Tabel detail digunakan untuk menampung data jam dan jumlah pelanggan yang pesan yang sudah memesan fasilitas pada tanggal dan jam yang dipilih. Keteranganannya dapat dilihat pada tabel 4.5 berikut ini:

Tabel 4.5 Tabel Detail

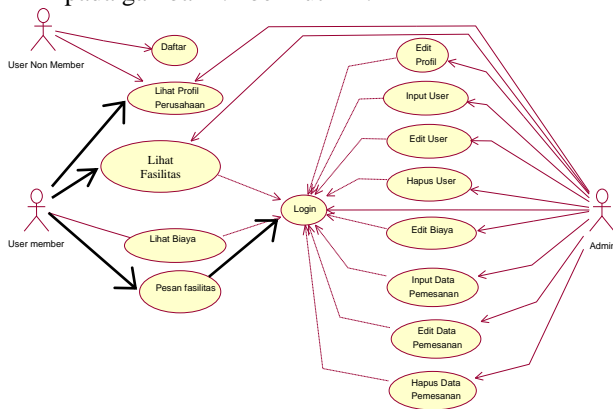
No	Field	Type	Keterangan
1	tgl	date	Primary Key
2	id_jam	int (3)	
3	jml	int (3)	

4.3. Analisa Perancangan Sistem

4.3.1. Use Case Diagram

Dalam *use case diagram* pemesanan fasilitas ini terdapat 3 aktor, diantaranya admin, *user member*, dan *user non-member*. Masing-masingnya memiliki hak akses sendiri dengan melakukan login terlebih dahulu atau tanpa harus melakukan login sekalipun.

Gambar *use case diagram* dari sistem pemesanan fasilitas hotel ini dapat dilihat pada gambar 4.1 berikut ini :

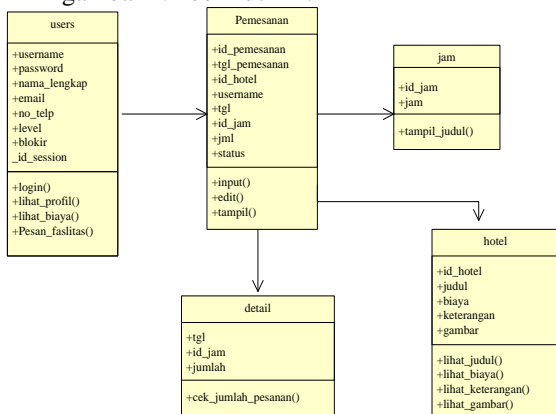


Gambar 4.1 Usecase Diagram

4.3.2. Class Diagram

Class diagram sistem pemesanan fasilitas ini mempunyai 5 kelas, yaitu *user*, Masing – masing *class* saling terkait dengan yang lainnya sesuai dengan sistem yang ada.

Gambar *class diagram* untuk sistem pemesanan fasilitas ini dapat dilihat pada gambar 4.2 berikut ini :



Gambar 4.2 Class Diagram

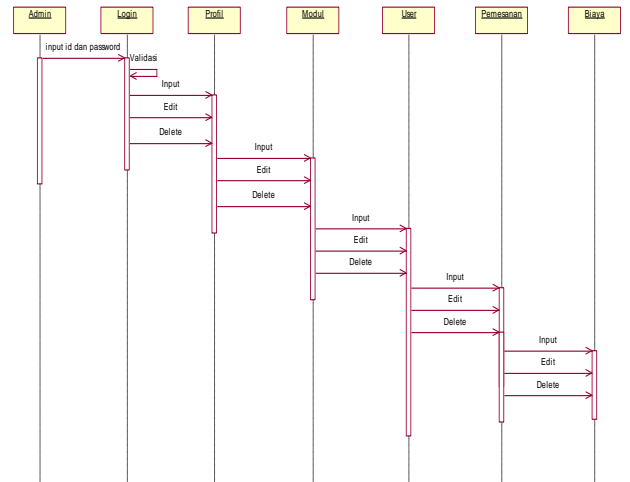
4.3.3. Sequence Diagram

Sequence diagram menjelaskan interaksi objek yang disusun berdasarkan urutan waktu. Secara mudahnya *sequence diagram* adalah gambaran tahap demi tahap yang seharusnya dilakukan untuk menghasilkan suatu sistem sesuai dengan *use case diagram*.

4.3.3.1. Sequence Diagram Admin

Dalam *sequence diagram* admin dapat dilihat interaksi-interaksi admin di dalam sistem pemesanan fasilitas ini. Interaksi admin dengan login sebagai validasi sebelum mengakses halaman lainnya.

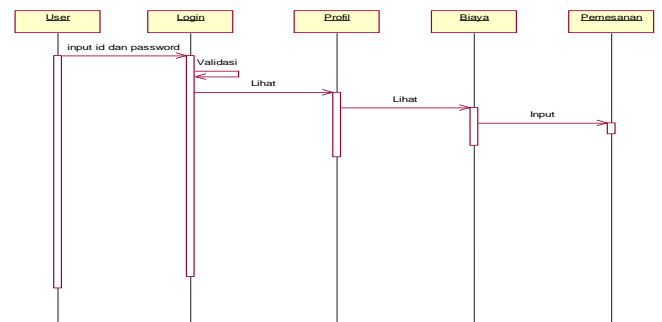
Keterangan diatas dapat dilihat pada gambar 4.3 berikut ini :



Gambar 4.3 Sequence Diagram Admin

4.3.3.2. Sequence Diagram User Member

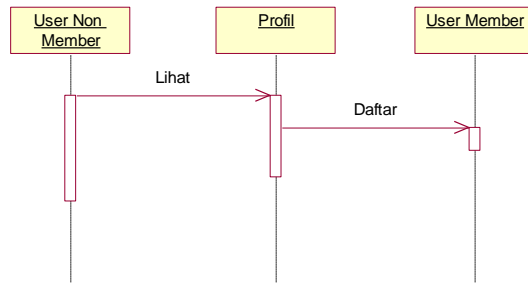
Interaksi-interaksi *user member* dalam sistem pemesanan fasilitas ini dapat dilihat pada *sequence diagram user member*. Interaksi *user member* dengan login sebagai validasi sebelum mengakses halaman pemesanan dan memberi komentar. Gambarnya dapat dilihat pada gambar 4.4 berikut ini :



Gambar 4.4 Sequence Diagram User Member

4.3.3.3. Sequence Diagram User Non-member

Interaksi-interaksi *user non-member* dalam sistem pemesanan fasilitas ini dapat dilihat pada *sequence diagram user non-member*. Interaksi yang terjadi hanya sebatas melihat profil, dan melakukan pendaftaran. Keterangan dapat dilihat pada gambar 4.5 berikut ini :



Gambar 4.5 Sequence Diagram User Non-member

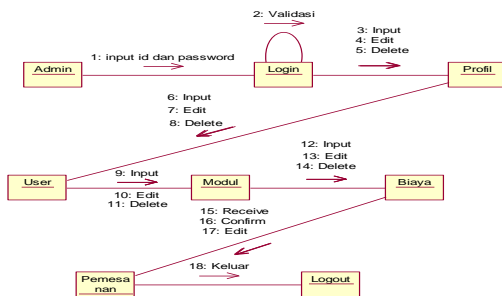
4.3.4. Collaboration Diagram

Collaboration diagram adalah cara alternative untuk mengetahui tahap-tahap terjadinya suatu aktivitas. Perbedaan antara collaboration dan sequence diagram adalah collaboration diagram memperlihatkan bagaimana hubungan antara beberapa objek, sedangkan sequence diagram memperlihatkan bagaimana urutan kejadian.

4.3.4.1. Collaboration Diagram Admin

Berikut adalah rangkaian hubungan objek yang dapat diakses admin, diantaranya login, input, edit dan hapus profil, jadwal, pemesanan, user, sampai logout.

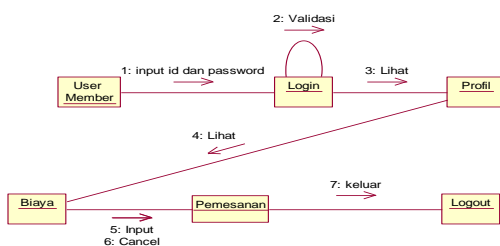
Keterangan diatas dapat dilihat pada gambar 4.6 berikut ini :



Gambar 4.6 Collaboration Diagram Admin

4.3.4.2. Collaboration Diagram User Member

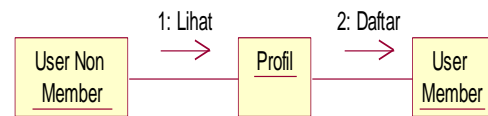
Rangkaian hubungan objek yang dapat diakses user member, diantaranya login, lihat profil, lihat jadwal, pesan fasilitas, beri komentar, dan logout untuk meninggalkan akunnya. Keterangan diatas dapat dilihat pada gambar 4.7 berikut ini :



Gambar 4.7 Collaboration Diagram User Member

4.3.4.3. Collaboration Diagram User Non-member

Rangkaian hubungan objek yang dapat diakses oleh user non-member hanya sebatas melihat profil perusahaan dan melakukan Pendaftaran. Keterangan diatas dapat dilihat pada gambar 4.8 berikut ini :



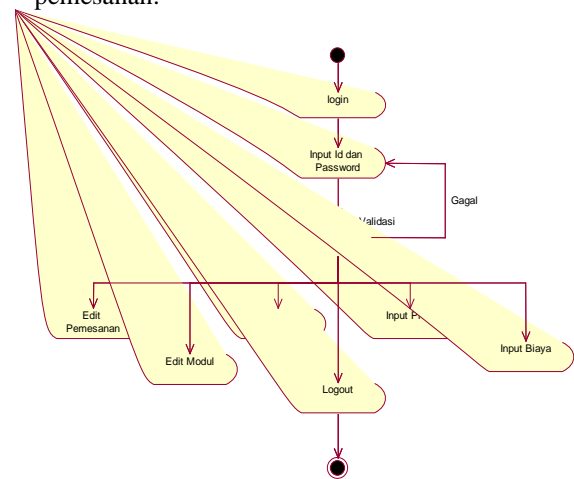
Gambar 4.8 Collaboration Diagram User Non-member

4.3.5. Activity Diagram

Activity diagram merupakan sebuah diagram dimana dalam diagram tersebut terdapat semua aktivitas dalam sistem pemesanan fasilitas ini. Dalam activity diagram digambarkan aktivitas dari setiap aktor yang ada.

4.3.5.1. Activity Diagram Admin

Activity diagram admin menggambarkan aktivitas-aktivitas yang dapat dilakukan oleh admin di dalam sistem pemesanan fasilitas ini. Pada activity diagram admin dibawah ini dapat kita lihat aktivitas dari admin yaitu admin harus login terlebih dahulu untuk dapat menginputkan data profil, jadwal, user dan pemesanan.



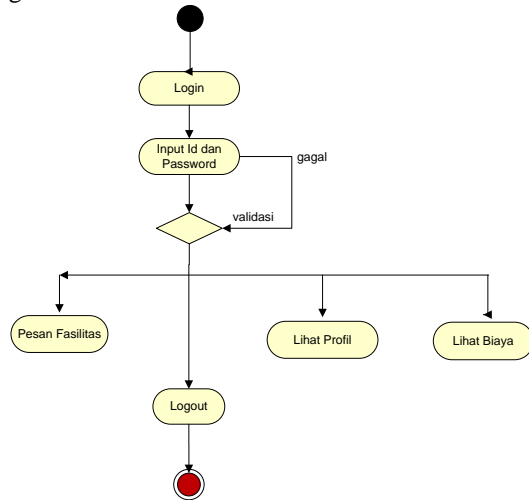
Gambar 4.9 Activity Diagram Admin

4.3.5.2. Activity Diagram User Member

Activity diagram user member menggambarkan aktivitas-aktivitas yang dapat dilakukan oleh user member di dalam sistem pemesanan fasilitas ini. Pada gambar activity diagram user member dibawah ini dapat kita lihat aktivitas dari user member yaitu user member harus login terlebih dahulu untuk

dapat melakukan pemesanan fasilitas, lihat profil, dan jadwal pemesanan.

Keterangan diatas dapat dilihat pada gambar 4.10 berikut ini :

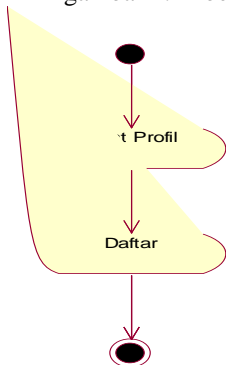


Gambar 4.10 Activity Diagram User Member

4.3.5.3. Activity Diagram User Non-member

Activity diagram user non-member menggambarkan aktivitas-aktivitas yang dapat dilakukan oleh pengguna yang tidak terdaftar di dalam sistem. Pada activity diagram user non-member dibawah ini dapat kita lihat aktivitas dari user non-member yaitu seorang pengguna yang tidak terdaftar hanya bisa melihat profil, melakukan pendaftaran.

Keterangan diatas dapat dilihat pada gambar 4.11 berikut ini :



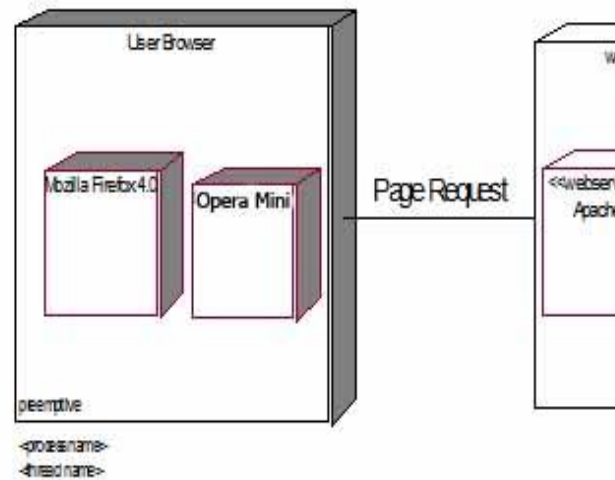
Gambar 4.11 Activity Diagram User Non-member

4.3.6 Deployment Diagram

Deployment/physical diagram menggambarkan detail bagaimana komponen di-deploy dalam infrastruktur sistem, dimana komponen akan terletak (pada mesin, server atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server, dan hal-hal lain yang bersifat fisik.

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.

Pada perancangan sistem pemesanan fasilitas hotel yang berbasis wap, server admin, webservice dan database yang digunakan adalah Apache dan MySQL. Sementara, user dapat melakukan akses dengan menggunakan browser yang ada di handphone dan komputer yang mereka gunakan. Keterangan diatas dapat dilihat pada gambar 4.12 berikut ini :



Gambar 4.12 Deployment Diagram 4.4. Desain Antarmuka Sistem

Desain antarmuka dirancang untuk mempertimbangkan keinginan pengguna, yaitu aplikasi yang memenuhi kebutuhan dan kemudahan untuk digunakan.

4.4.1 Desain Tampilan Halaman WAP

Desain ini menggambarkan semua tampilan dari halaman-halaman wap yang akan dibuat. Mulai dari halaman user non member sampai kepada halaman member.

4.4.1.1 Desain Halaman Awal

Halaman ini merupakan halaman yang pertama kali muncul ketika user mengakses website ini.

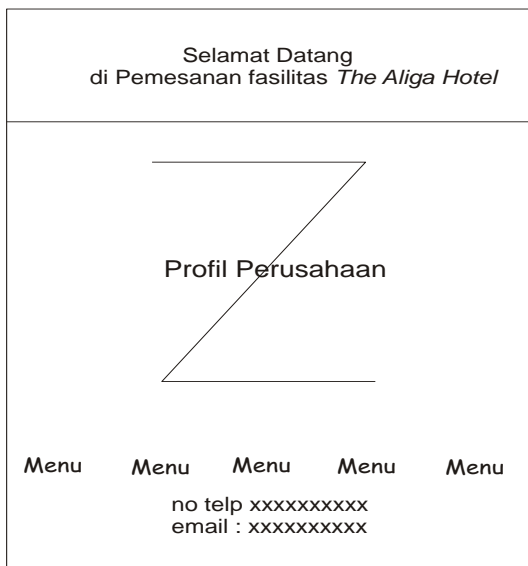
Keterangan diatas dapat dilihat pada gambar 4.13 berikut ini:



Gambar 4.13
Desain Halaman Awal WAP

4.4.1.2 Desain Halaman Profil Perusahaan

Pada halaman ini akan ditampilkan profil perusahaan, desain selengkapnya dapat dilihat pada tabel 4.14 berikut ini:



Gambar 4.14 Desain Halaman Profil Perusahaan

4.4.1.3 Desain Halaman Informasi Fasilitas

Pada halaman ini akan ditampilkan informasi fasilitas,

desain selengkapnya dapat dilihat pada tabel 4.15 berikut ini:



Gambar 4.15 Desain Halaman Informasi Fasilitas

4.4.1.4 Desain Halaman Cara Pemesanan

Pada halaman ini akan ditampilkan informasi mengenai cara pemesanan, desain selengkapnya dapat dilihat pada tabel 4.16 berikut ini:



Gambar 4.16 Desain Halaman Cara Pemesanan

4.4.1.5 Desain Halaman Daftar

Pada halaman ini akan ditampilkan desain daftar menjadi member, desain selengkapnya dapat dilihat pada tabel 4.17 berikut ini:

Gambar 4.17 Desain Halaman Daftar

4.4.1.6 Desain Halaman Login

Pada halaman ini akan ditampilkan form login, desain selengkapnya dapat dilihat pada tabel 4.18 berikut ini:

Gambar 4.18 Desain Halaman Login

4.4.1.7 Desain Halaman Member

Pada halaman ini akan ditampilkan halaman member, desain selengkapnya dapat dilihat pada tabel 4.19 berikut ini:

Gambar 4.19 Desain Halaman Member

4.4.1.8 Desain Halaman Pesan Fasilitas

Pada halaman ini akan ditampilkan halaman pemesanan fasilitas, desain selengkapnya dapat dilihat pada tabel 4.20 berikut ini:

Gambar 4.20 Desain Halaman Pesan Fasilitas

4.4.1.9 Desain Halaman Laporan Pemesanan

Pada halaman ini akan ditampilkan halaman laporan pemesanan fasilitas, desain selengkapnya dapat dilihat pada tabel 4.21 berikut ini:

Halaman Member Pemesanan Fasilitas <i>The Aliga Hotel</i>					
Laporan Pemesanan Fasilitas					
Nama	xxxxxxxxxx				
Fasilitas	xxxxxxxxxx				
Tanggal Pesan	xxxxxxxxxx				
Jam Pesan	xxxxxxxxxx				
Jumlah	xxxxxxxxxx				
Status	xxxxxxxxxx				
Menu	Menu	Menu	Menu	Menu	Menu
no telp xxxxxxxxxx email : xxxxxxxxxx					

Gambar 4.21 Desain Halaman Laporan Pemesanan

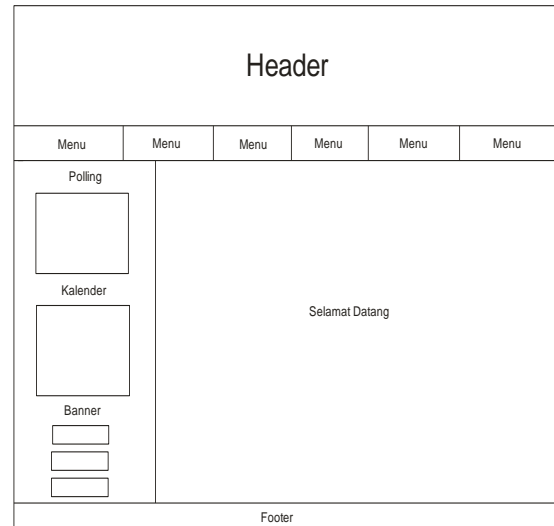
4.4.2 Desain Tampilan Halaman WEB

Desain ini menggambarkan semua tampilan dari halaman-halaman *web* yang akan dibuat. Didalamnya terdapat halaman user non member, member, dan halaman admin dengan fitur masing – masing.

4.4.2.1 Desain Halaman User Non Member

Merupakan tampilan awal ketika mengakses website yang akan dibuat. Berisikan informasi dan menu-menu yang dapat diakses oleh user.

Keterangan diatas dapat dilihat pada gambar 4.22 berikut ini:

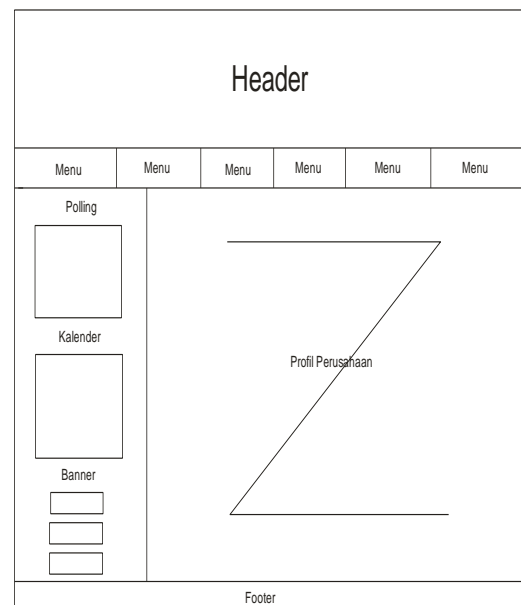


Gambar 4.22 Desain Halaman User Non Member

4.4.2.2 Desain Halaman Profil Perusahaan

Pada halaman ini akan ditampilkan profil perusahaan yang terdapat sejarah, visi dan misi dari perusahaan tersebut, sehingga user tahu dengan sejarah yang ada pada perusahaan tersebut.

Desain selengkapnya dapat dilihat pada tabel 4.23 berikut ini:

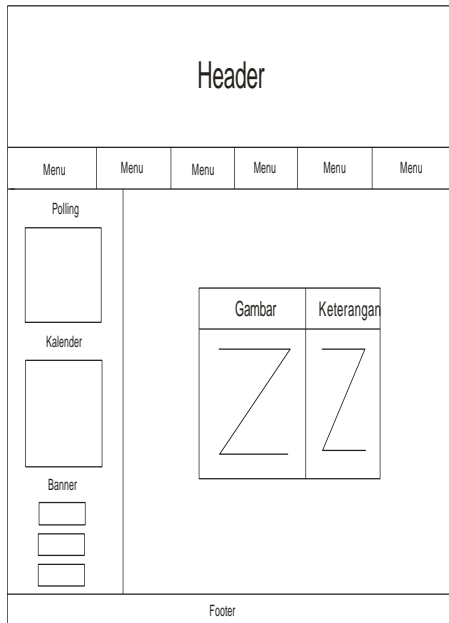


Gambar 4.23 Desain Halaman Profil Perusahaan

4.4.2.3 Desain Halaman Informasi Fasilitas

Pada halaman ini akan ditampilkan informasi fasilitas beserta biaya fasilitas itu sendiri sehingga user dapat menentukan fasilitas apa yang akan dipilihnya.

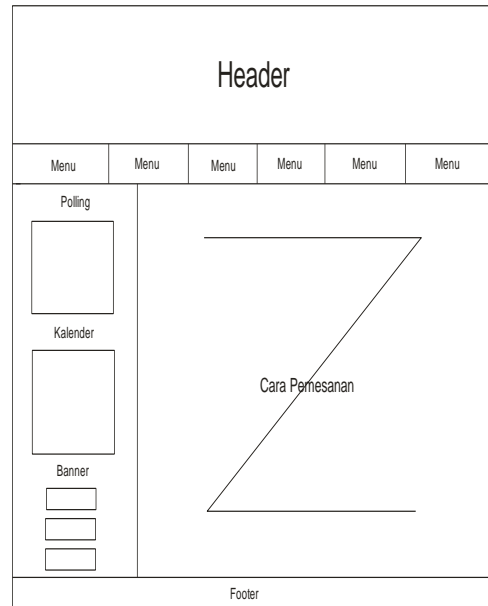
Desain selengkapnya dapat dilihat pada tabel 4.24 berikut ini:



Gambar 4.24 Desain Halaman Informasi Fasilitas

4.4.2.4 Desain Halaman Cara Pemesanan

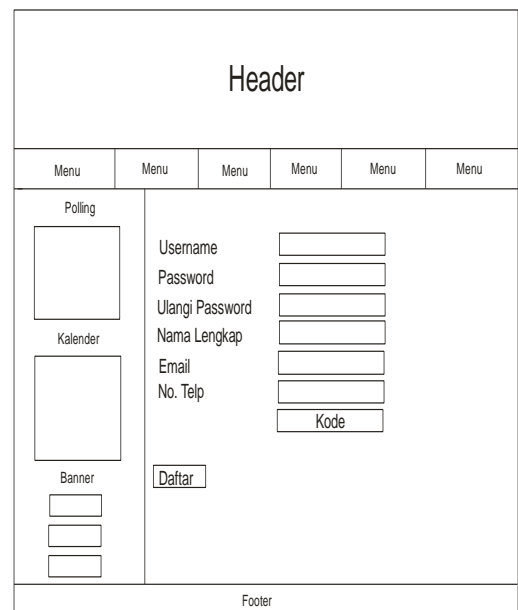
Pada halaman ini akan ditampilkan cara pemesanan, dimana user mengisi form yang disediakan untuk memilih fasilitas yang tersedia. Desain selengkapnya dapat dilihat pada tabel 4.25 berikut ini:



Gambar 4.25 Desain Halaman Cara Pemesanan

4.4.2.5 Desain Halaman Daftar

Pada halaman ini akan ditampilkan form daftar, dimana user apabila ingin memesan fasilitas harus registrasi dahulu kemudian user akan dapat melakukan pemesanan. Desain selengkapnya dapat dilihat pada tabel 4.26 berikut ini:



Gambar 4.26 Desain Halaman Daftar

4.4.2.6 Desain Halaman Login

Pada halaman ini akan ditampilkan form login, dimana user yang telah registrasi disilahkan untuk login agar user dapat menentukan fasilitas yang akan dipesan. Desain selengkapnya dapat dilihat pada tabel 4.27 berikut ini:

Gambar 4.27 Desain Halaman Login

4.4.2.7 Desain Halaman Pesan Fasilitas

Pada halaman ini akan ditampilkan form pemesanan fasilitas, dimana user mengisi form untuk pemesanan. Desain selengkapnya dapat dilihat pada tabel 4.28 berikut ini:

Gambar 4.28 Desain Halaman Pemesanan Fasilitas

4.4.2.8 Desain Halaman Laporan Pemesanan

Pada halaman ini akan ditampilkan form pemesanan fasilitas, dimana user setelah memesan dapat melihat hasil form yang telah diisi dan print.

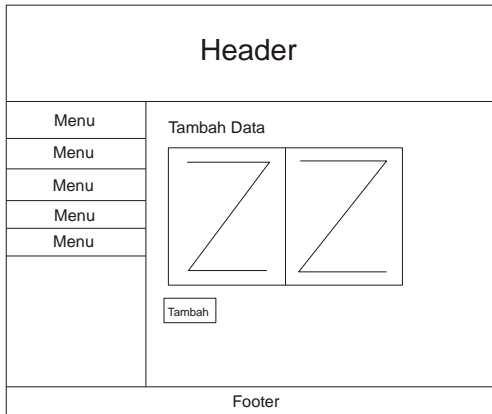
Desain selengkapnya dapat dilihat pada tabel 4.29 berikut ini:

Gambar 4.29 Desain Halaman Laporan Pemesanan Fasilitas

4.4.2.9 Desain Halaman Tambah Data

Pada halaman ini akan ditampilkan form tambah data pada tampilan admin, sehingga admin dapat mengedit, menghapus tanpa harus melalui database langsung.

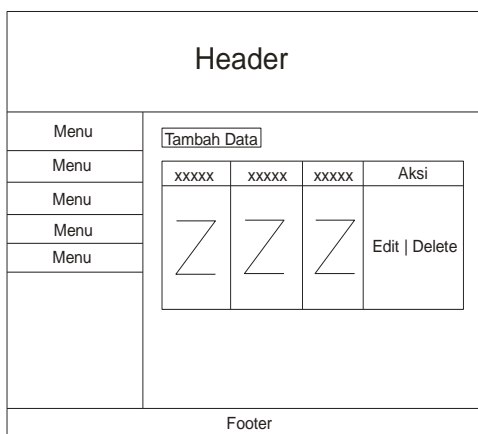
Desain selengkapnya dapat dilihat pada tabel 4.30 berikut ini:



Gambar 4.30 Desain Halaman Tambah Data

4.4.2.10 Desain Halaman Tampil Data

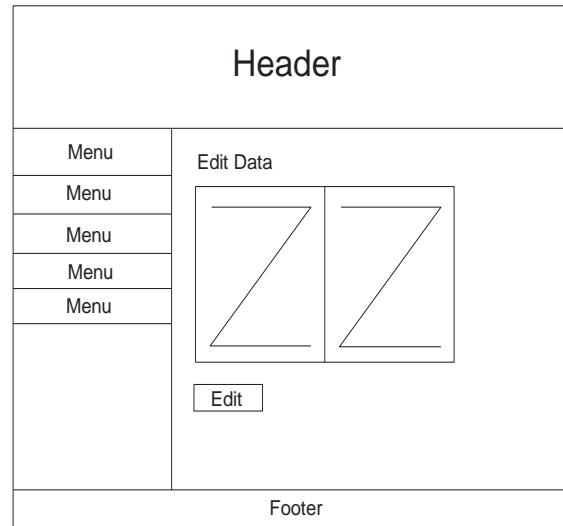
Pada halaman ini akan ditampilkan form tampil data pada tampilan admin, desain selengkapnya dapat dilihat pada tabel 4.31 berikut ini:



Gambar 4.31 Desain Halaman Tampil Data

4.4.2.11 Desain Halaman Edit Data

Pada halaman ini akan ditampilkan form edit data pada halaman admin, desain selengkapnya dapat dilihat pada tabel 4.32 berikut ini:



Gambar 4.32 Desain Halaman Edit Data

5. KESIMPULAN

Dari penulisan tugas akhir ini mulai dari tahapan analisa permasalahan yang ada hingga pengujian aplikasi sistem yang baru maka dapat diambil beberapa kesimpulan, yaitu:

1. Bahasa pemrograman PHP dapat memberikan kemudahan dalam pembangunan aplikasi CRM ini.
2. Program Adobe Dreamweaver CS5 sangat membantu dalam perancangan design sistem ini.
3. Database MySQL dapat menampung informasi data admin dan pelanggan di *The Aliga Hotel*.
4. Aplikasi *Pemesanan Fasilitas* ini dapat memudahkan hotel aliga untuk berintegrasi dengan pelanggan.
5. Aplikasi *Pemesanan Fasilitas* ini dapat memudahkan pelanggan dalam melakukan pemesanan.

5.1 Keterbatasan Sistem

Dari hasil analisa yang dilakukan, sistem yang dirancang masih memiliki keterbatasan, antara lain yaitu:

1. Sistem yang dibangun penulis pada intinya hanya memberikan informasi mengenai

profil, biaya, fasilitas yang tersedia dan lainnya.

2. Dalam sistem ini, pemesanan fasilitas hanya dapat dilakukan oleh member dan sifatnya hanya pemesanan.

Saran-saran

Agar sistem yang diusulkan dapat digunakan lebih optimal dan dapat berjalan sesuai dengan yang diharapkan, maka ada beberapa saran yang dapat dijadikan bahan pertimbangan untuk pihak *The Aliga Hotel*. Adapun saran yang penulis berikan adalah sebagai berikut :

1. Sistem yang dibangun penulis pada intinya hanya sebatas informasi mengenai *profil*, pemesanan fasilitas, dan informasi lainnya di hotel aliga. Sehingga diharapkan adanya pengembangan lagi untuk sistem yang lebih luas cakupannya.
2. Diharapkan ada pengembangan lainnya di aplikasi Pemesanan Fasilitas ini sehingga nantinya dapat meningkatkan nilai perusahaan di mata pelanggan.
3. Diperlukan *maintenace* terhadap program aplikasi yang telah dibuat, supaya dapat digunakan secara berkelanjutan selama kebutuhan terhadap informasi yang ada di hotel aliga

Anonymous (2008), *Sejarah UML* Diakses pada 2Nopember 2008 dari <http://www.omg.org>.

Anonymous (2008), *Sistem Informasi Akademik* Fakultas Farmasi UP. Komputer, Wahana (2005).

Enterprise, Jubilee, 2009, *Customer relationship management (CRM)* Dengan Blog, PT. Elexmedia Komputindo, Jakarta.

Fathansyah. Ir, *Buku Teks Ilmu Komputer Basis Data*, Cetakan Pertama, Penerbit Informatika, Bandung, 1999.

Romzi Imron Rozidi, *Membuat Sendidri SMS Gateway (ESME) Berbasis Protokol SMPP*. Andi Offset, Yogyakarta, 2004.

Analisis dan Perancangan Sistem Informasi dengan Metodologi Berorientasi Objek. Bandung: Informatika Bandung.

Faisal, *Sistem Informasi Akademik Berbasis Sms Gateway*, Juni 2010, www.gsmworld.com.

Azkal Fikri, *Aplikasi Short Message Service (sms) Gateway Untuk Layanan Informasi Registrasi Administrasi Mahasiswa*, 2009, <http://azkalfikri.blog.upi.edu>.

DAFTAR PUSTAKA