BAB IV

HASIL DAN PEMBAHASAN

4.1 Perancangan Sistem

Perancangan sistem dilakukan setelah tahap analisis sistem selesai mendapatkan gambaran dengan jelas apa yang harus dikerjakan. Berdasarkan hasil analisis sistem pada bab sebelumnya, maka diusulkan perancangan sistem baru untuk mengatasi beberapa permasalahan yang ada di sistem sebelumnya.

4.1.1 Tujuan Perancangan Sistem

Tujuan dari perancangan sistem secara umum adalah untuk memenuhi kebutuhan pemakai sistem dan untuk memberikan gambaran umum kepada pemakai terhadap sistem yang baru. Adapun tujuan perancangan dari sistem informasi reservasi *guest house* ini adalah :

a. Sistem informasi reservasi ini dapat memudahkan receptionist dalam seluruh proses penginputan dan penyimpanan data sehingga dapat memperoleh informasi data sebelumnya dengan mudah dan cepat.
b. Dengan adanya sistem informasi reservasi ini dapat mempercepat proses pengolahan data karena mudah dalam pencarian data.
c. Dapat mengetahui ketersediaan kamar.
d. Dapat mempermudah receptionist dalam melakukan proses biling sistem.

4.1.2 Gambaran Umum Sistem yang di usulkan

Sistem informasi yang diusulkan untuk sistem informasi reservasi kamar ini memiliki beberapa keunggulan dan perbedaan dari sistem yang sedang berjalan.

Sistem yang diusulkan telah terkomputerisasi, lebih mudah digunakan, integritas data terjaga, tidak akan memakan waktu yang lama dalam menginput dan mengolah data kamar, mengolah data menu, mengolah data barang laundry, transaksi reservasi, transaksi check in, transaksi restaurant, transaksi laundry, dan transaksi check out. Didalamnya telah menggunakan konsep client-server, dimana receptionist berperan memegang server yang dapat menginput, mengedit, menyimpan dan mencetak laporan tamu masuk, laporan restaurant, laporan laundry dan laporan tamu keluar sedangkan manager berperan sebagai client yang menerima hasil dari pengolahan transaksi tamu masuk, restaurant, laundry, dan tamu keluar.

Dengan adanya perancangan sistem informasi reservasi ini diharapkan dapat memudahkan receptionist dalam melakukan seluruh aktifitas yang berkaitan dengan kegiatan *guest house*.

4.1.3 Perancangan Prosedur Sistem yang Diusulkan

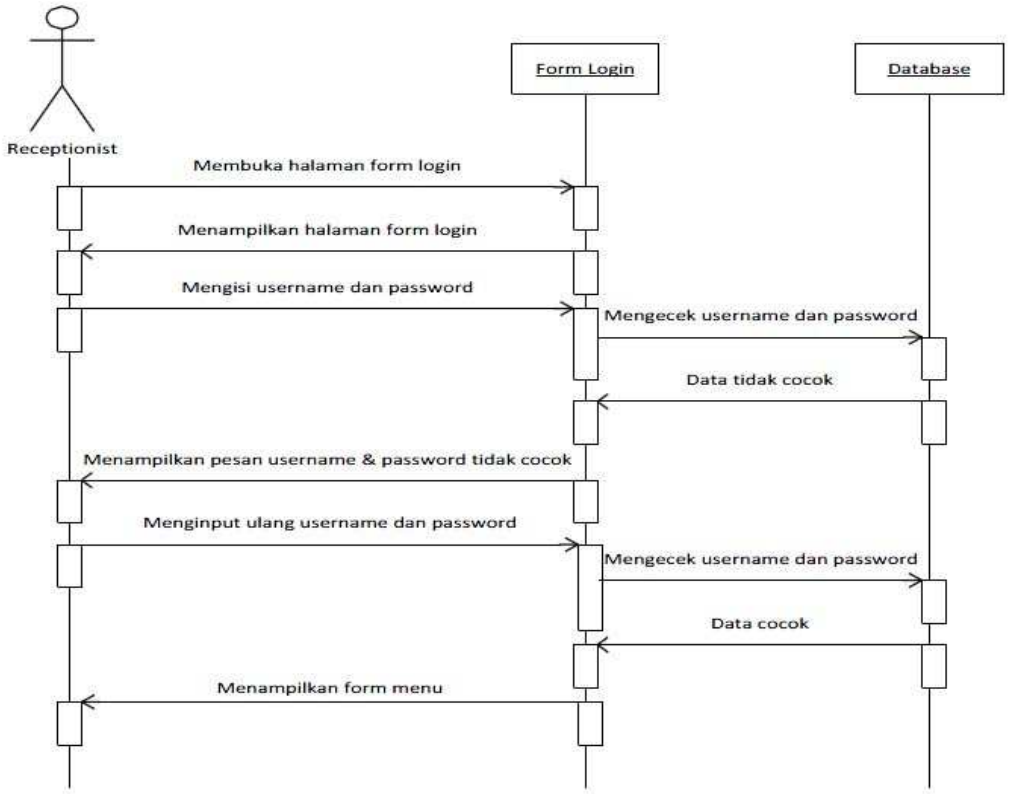
Perancangan prosedur merupakan tahap awal dari perancangan sistem informasi yang dilakukan sebagai pemecahan masalah yang ada pada proses sistem informasi reservasi kamar pada *Guest House* yang sedang berjalan.

4.1.3.1 Sequence Diagram

Sequence diagram adalah suatu diagram yang menggambarkan interaksi antar obyek dan mengindikasikan komunikasi diantara obyek-obyek tersebut. Diagram ini juga menunjukkan serangkaian pesan yang dipertukarkan oleh obyek-obyek yang melakukan suatu tugas atau aksi tertentu. Obyek-obyek tersebut kemudian diurutkan dari kiri ke kanan, aktor yang menginisiasi interaksi biasanya ditaruh di paling kiri dari diagram.

1. Sequence Diagram Login yang Diusulkan

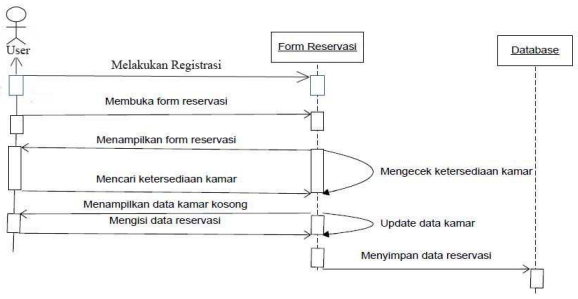
Berikut merupakan sequence diagram untuk Login. Adapun gambarnya sebagai berikut:



Gambar 4.1 Sequence Diagram Login yang Diusulkan

2. Sequence Diagram Reservasi yang Diusulkan

Berikut merupakan sequence diagram untuk mengelola reservasi. Adapun gambarnya sebagai berikut :

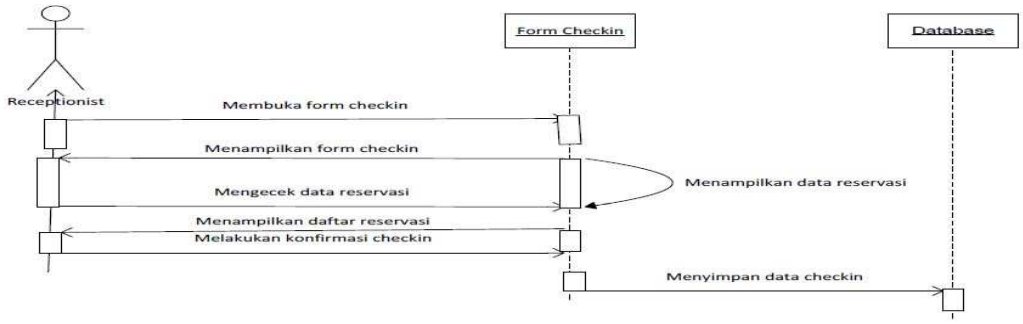


Gambar 4.2 Sequence Diagram Reservasi yang Diusulkan

3. Sequence Diagram Check In yang Diusulkan

Berikut merupakan sequence diagram untuk mengelola transaksi check in

Adapun gambarnya sebagai berikut :

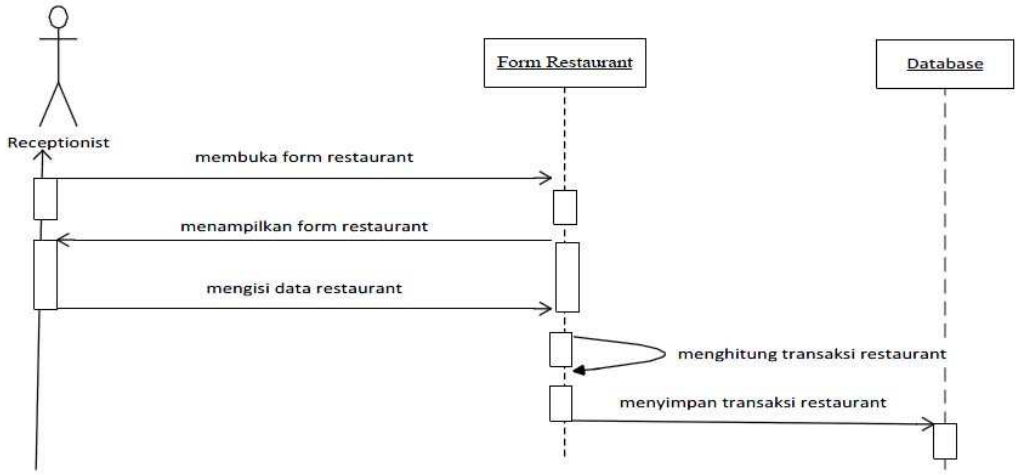


Gambar 4.3 Sequence Diagram Check In yang Diusulkan

4. Sequence Diagram Restauran yang Diusulkan

Berikut merupakan sequence diagram untuk mengelola transaksi restauran.

Adapun gambarnya sebagai berikut :

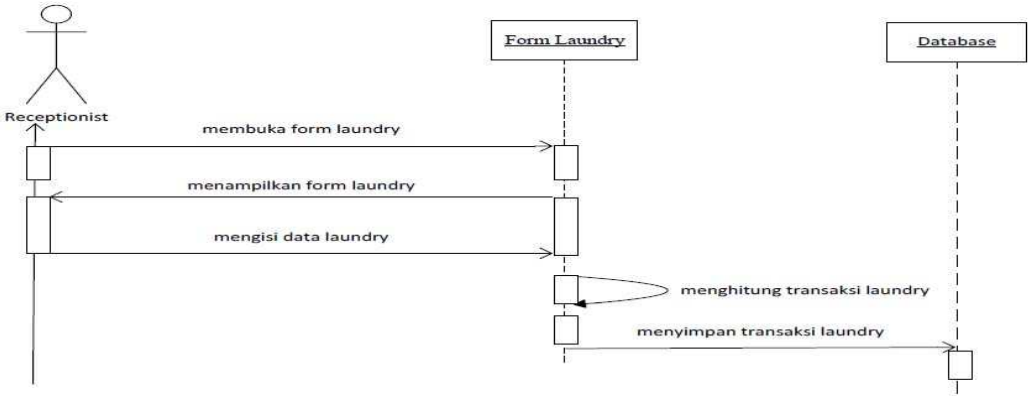


Gambar 4.4 *Sequence Diagram* Restaurant yang Diusulkan

5. Sequence Diagram Laundry yang Diusulkan

Berikut merupakan sequence diagram untuk mengelola transaksi check out.

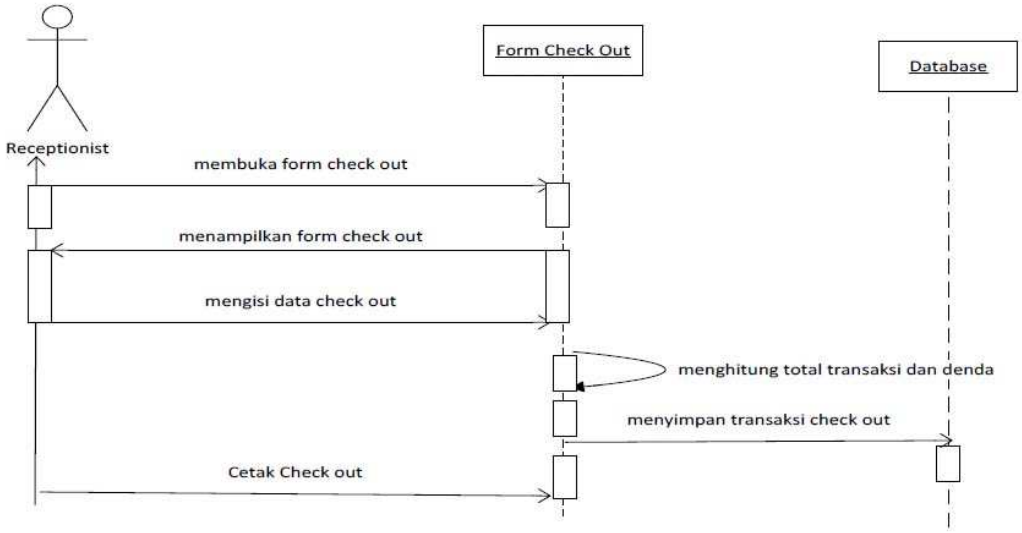
Adapun gambarnya sebagai berikut :



Gambar 4.5 *Sequence Diagram* Laundry yang Diusulkan

6. Sequence Diagram Check Out yang Diusulkan

Berikut merupakan sequence diagram untuk mengelola transaksi check out. Adapun gambarnya sebagai berikut :

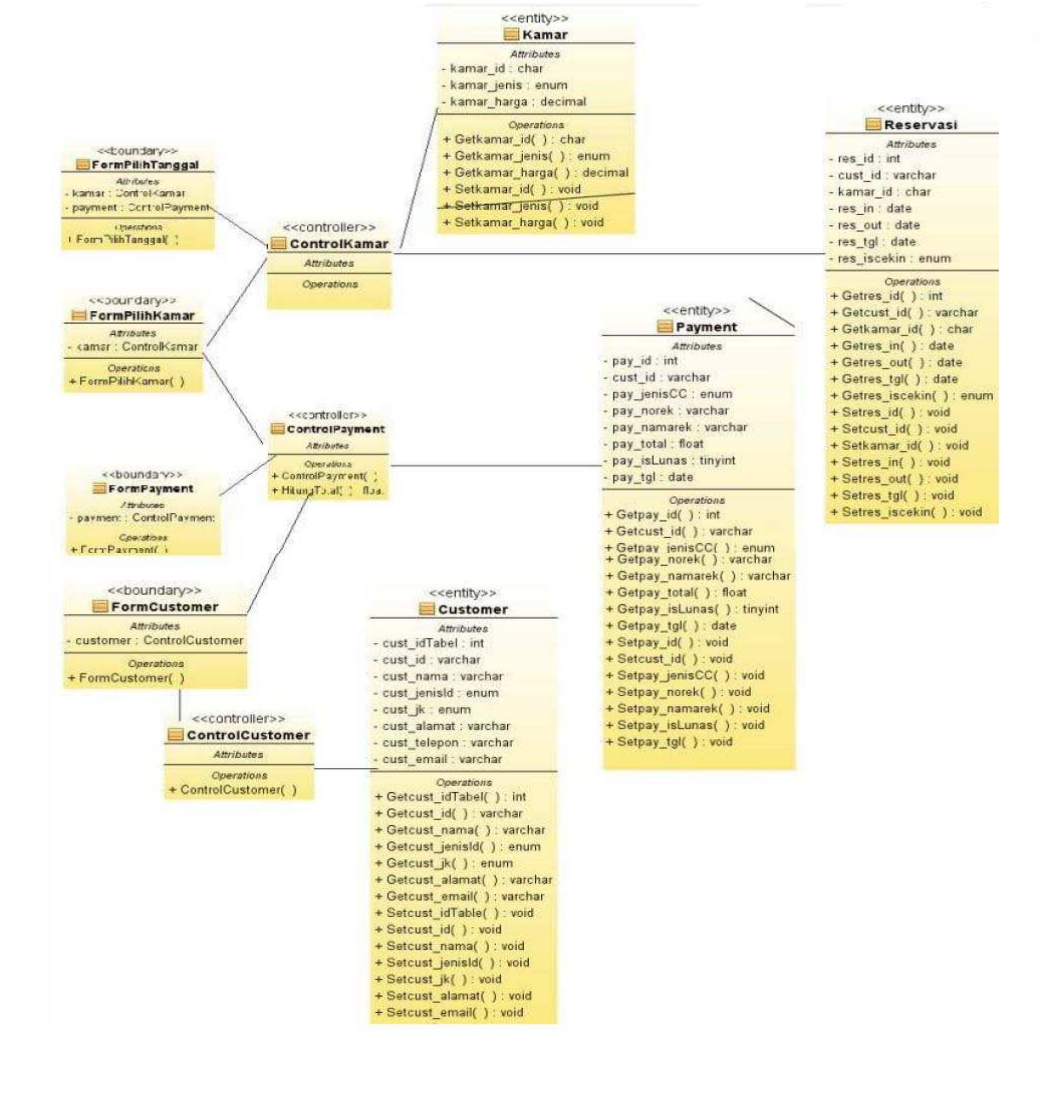


Gambar 4.6 *Sequence Diagram* Check Out yang Diusulkan

4.1.3.2 Class Diagram

Diagram kelas adalah diagram yang digunakan untuk menampilkan beberapa kelas serta paket-paket yang ada dalam sistem /perangkat lunak yang sedang kita kembangkan. Diagram kelas memberi gambaran/diagram statis tentang sistem/perangkat lunak dan relasi – relasi yang ada didalamnya.

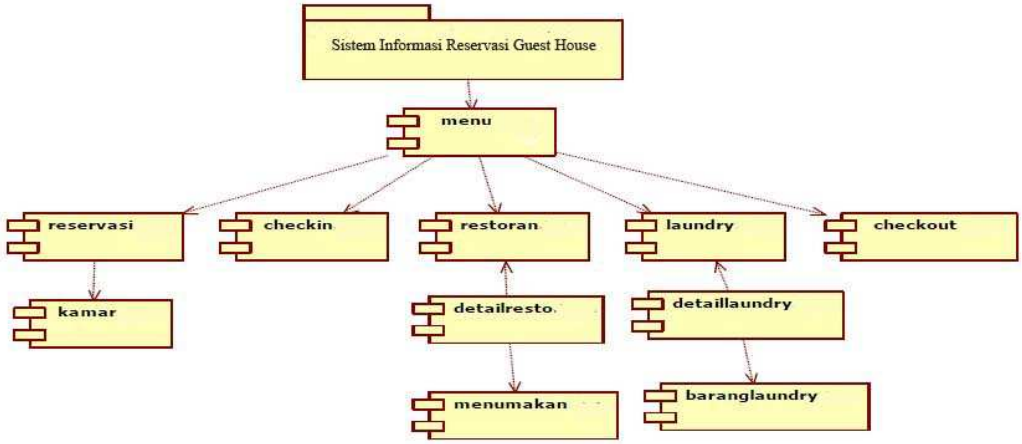
Adapun diagram class yang di usulkan adalah sebagai berikut :



Gambar 4.7 *Class Diagram* yang Diusulkan

4.1.3.3 Component Diagram

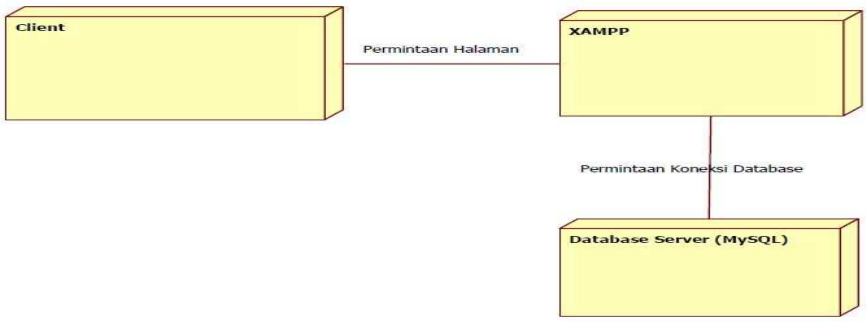
Component diagram adalah modul fisik program. Komponen – komponen yang dimaksud adalah pustaka – pustaka kode dan berkas – berkas yang ada saat program dijalankan. Adapun Component diagram yang di usulkan adalah sebagai berikut:



Gambar 4.8 *Component Diagram* yang Diusulkan

4.1.3.4 Deployment Diagram

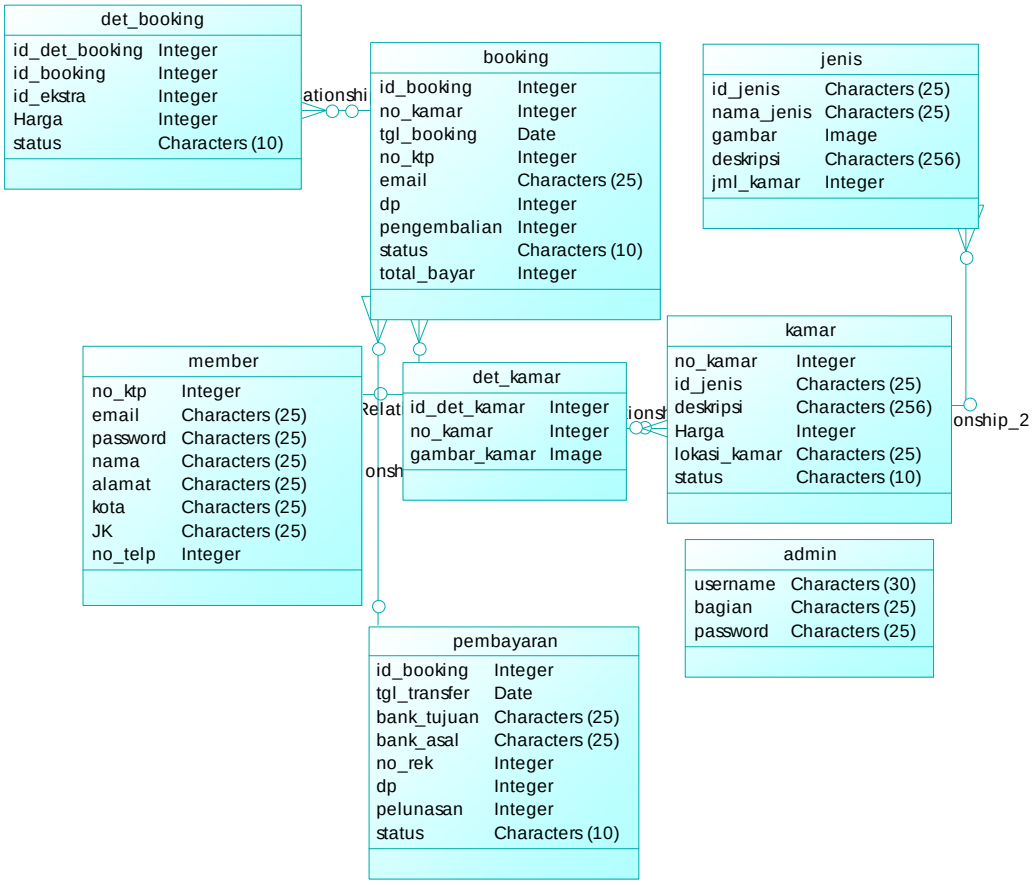
Memperlihatkan setiap simpul (node) dalam jaringan, hubungan – hubungan antara simpul itu sendiri serta proses – proses yang akan berjalan di masing – masing simpul. Adapun *Deployment diagram* yang di usulkan adalah sebagai berikut :



Gambar 4.9 Deployment Diagram yang Diusulkan

4.1.3.5 Relasi Tabel

Relasi antar tabel adalah suatu proses mengorganisasikan *file* untuk menghilangkan grup elemen yang berulang-ulang. Proses relasi antar tabel merupakan pengelompokan data menjadi tabel-tabel yang merupakan *entity* dan relasinya. Berfungsi mengakses data dan *item* sedemikian rupa sehingga database tersebut mudah dimodifikasi. Adapun bentuk relasi antar tabel dari sistem informasi yang diusulkan adalah sebagai berikut :

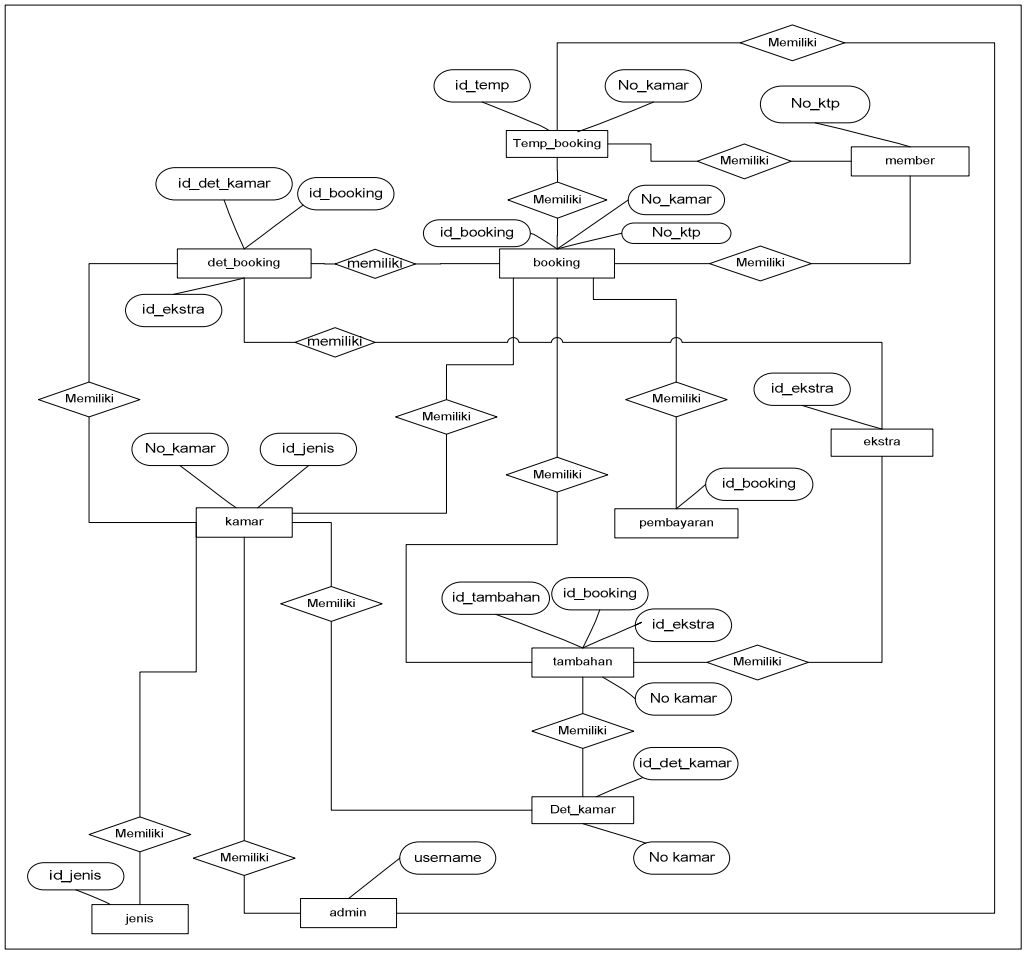


Gambar 4.10 Relasi Tabel

4.1.3.6 Entity Relationship Diagram (ERD)

Komponen utama ERD adalah entitas, atribut dan relasi. Entitas merupakan individu yang mewakili sesuatu yang nyata dapat dibedakan dari yang lain. Relasi

merupakan adanya hubungan diantara sejumlah entitas yang berasal dari entitas yang berbeda. Entity Relationship Diagram pada aplikasi sistem informasi penjualan ini yaitu sebagai berikut:

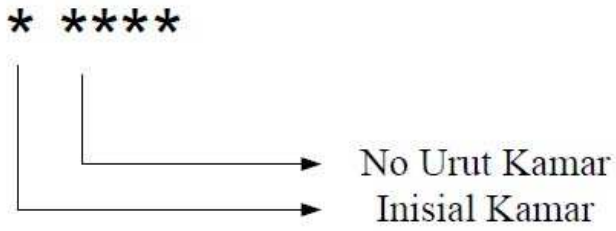


Gambar 4.11 Entity Relationship Diagram (ERD)

4.1.3.7 Kodifikasi

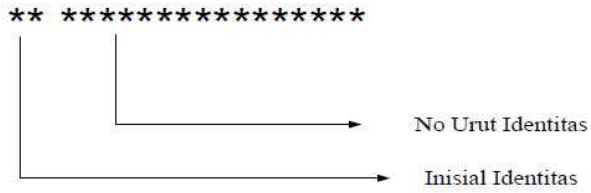
Sistem pengkodean ini dibuat guna mengidentifikasi suatu objek secara singkat. Dengan adanya sistem pengkodean ini diharapkan dapat mengklasifikasikan data. Kode dapat dibentuk dari kumpulan huruf, angka, dan karakter khusus.

1. no_kamar



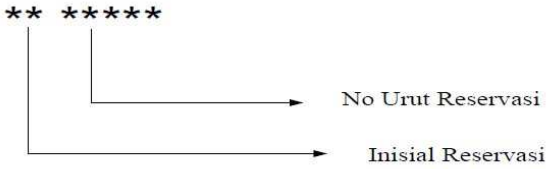
Contoh: K0001

2. no_identitas



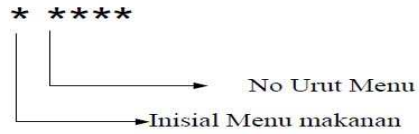
Contoh : ID3211085306890002

3. no_reservasi



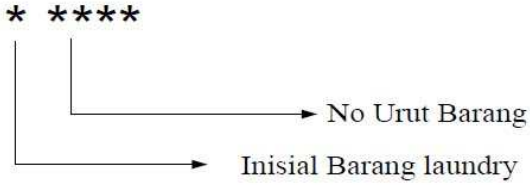
Contoh: CI00001

4. kode_menu



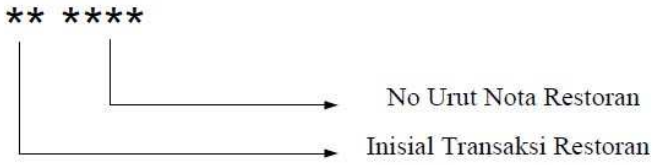
Contoh: R0001

5. kode_laundry



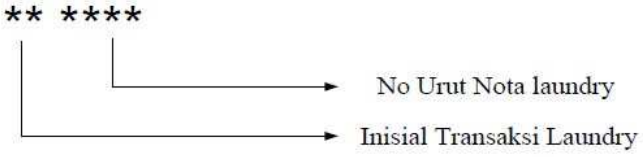
Contoh: L0001

6. no_nota_restoran



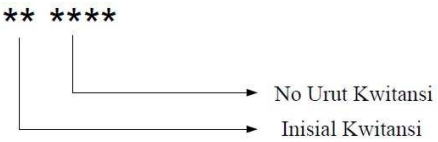
Contoh: TR0001

7. no_nota_laundry



Contoh: TL0001

8. no_kwitansi



Contoh: KW0001

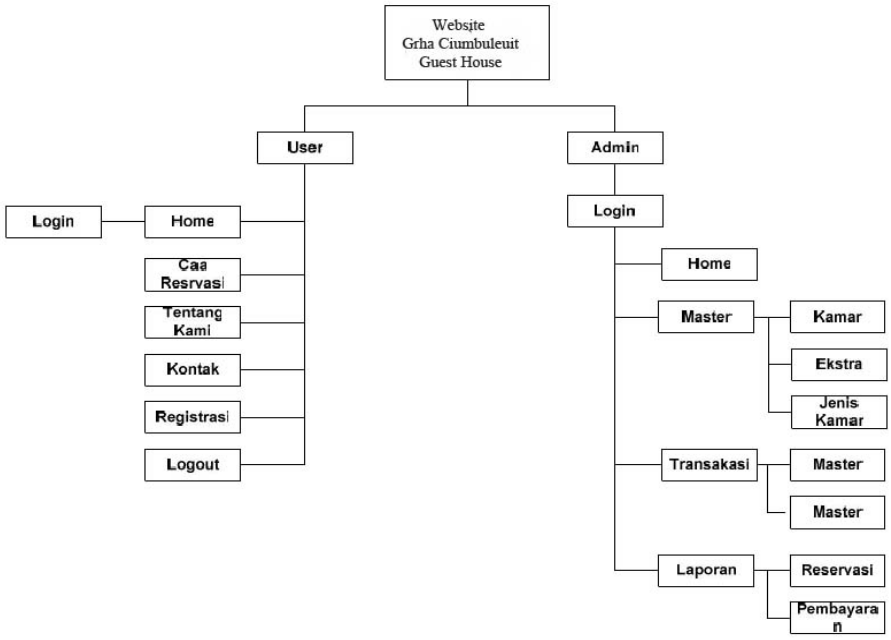
4.2. Perancangan Antar Muka

Perancangan antar muka merupakan tahapan untuk membuat tampilan atau desain dari sistem yang akan dibuat. Perancangan antar muka pemakai sangat penting untuk memenuhi criteria yang mudah digunakan, menarik dan nyaman digunakan oleh pemakai. Oleh karena itu dibuatlah rancangan antar muka untuk memudahkan pemakai. Rancangan tampilan yang dibuat meliputi rancangan struktur menu, rancangan input dan rancangan output dari sistem yang akan dibuat

4.2.1. Struktur Menu

Perancangan menu digunakan untuk memudahkan penelusuran serta alur program ketika kita menjalankan program yang kita buat. Struktur menu akan dibagi menjadi 2 yaitu struktur menu user dan struktur menu admin. Berikut ini

adalah gambar struktur menu usulan penjualan barang yang dapat dilihat pada gambar 4.10.



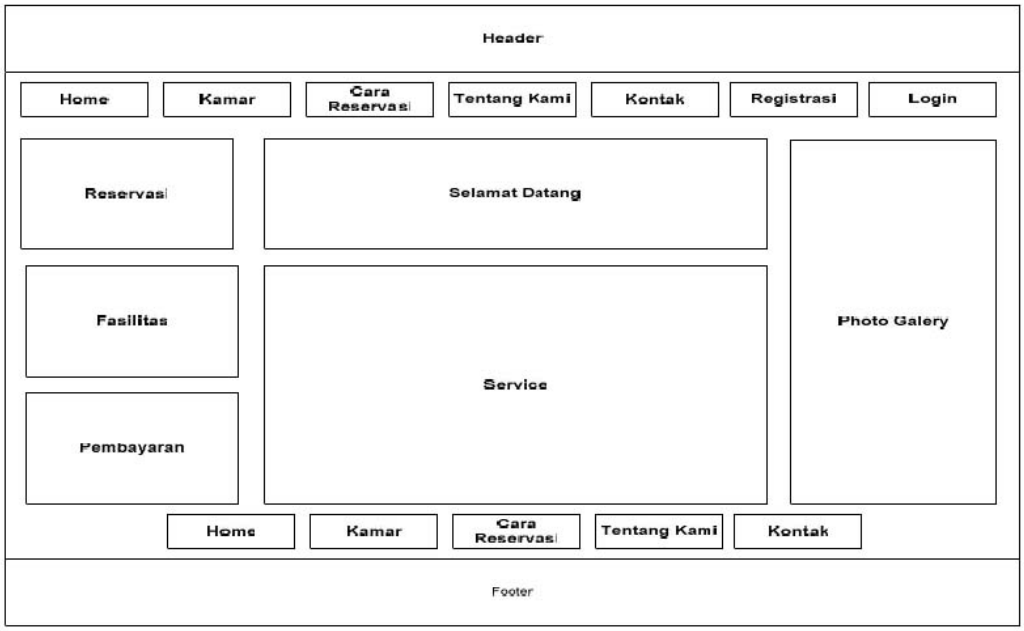
Gambar 4.12 Struktur Menu yang Dirancang

4.2.2. Perancangan Input

Dalam penbuatan website Grha Ciumbuleuit Guest House, penulis membuat 2 rancangan interface yaitu tampilan admin dan tampilan user yang dijelaskan sebagai berikut:

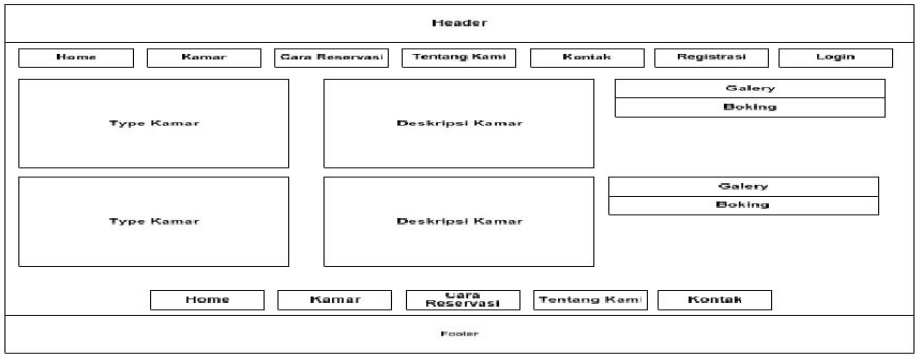
1. Tampilan User

a. Tampilan home user



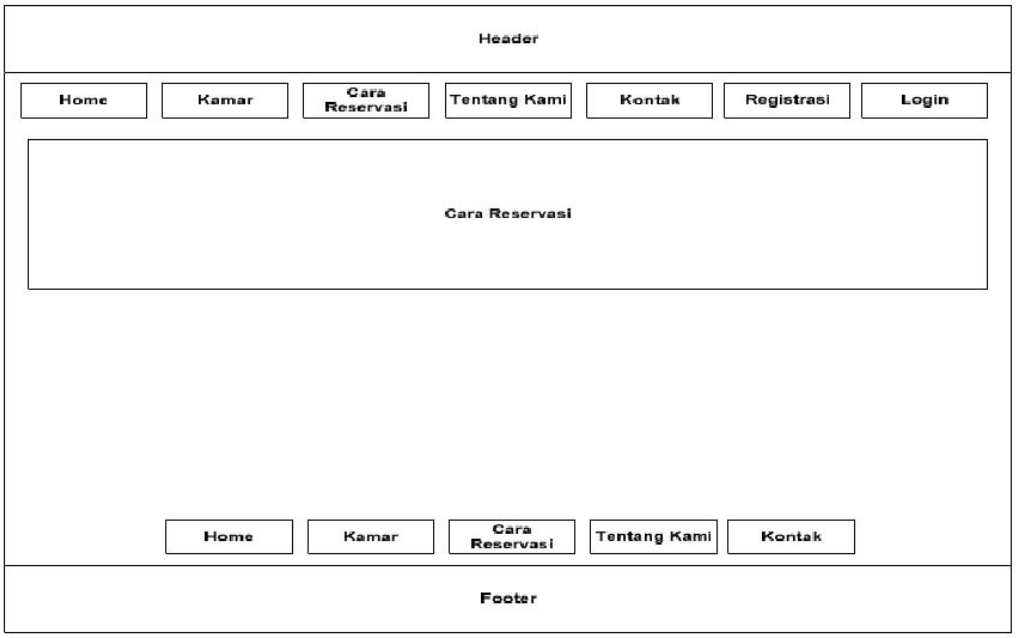
Gambar 4.13 Tampilan home user

b. Tampilan Kamar User



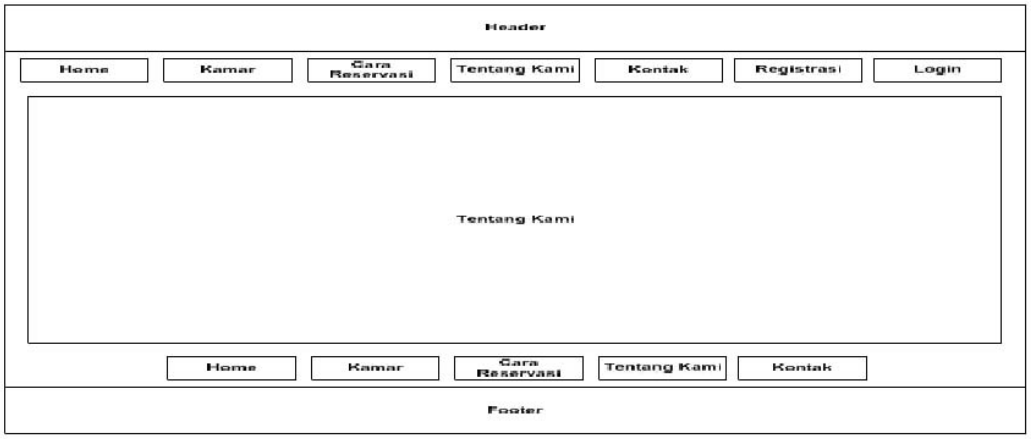
Gambar 4.14 Tampilan Kamar User

c. Tampilan cara reservasi User



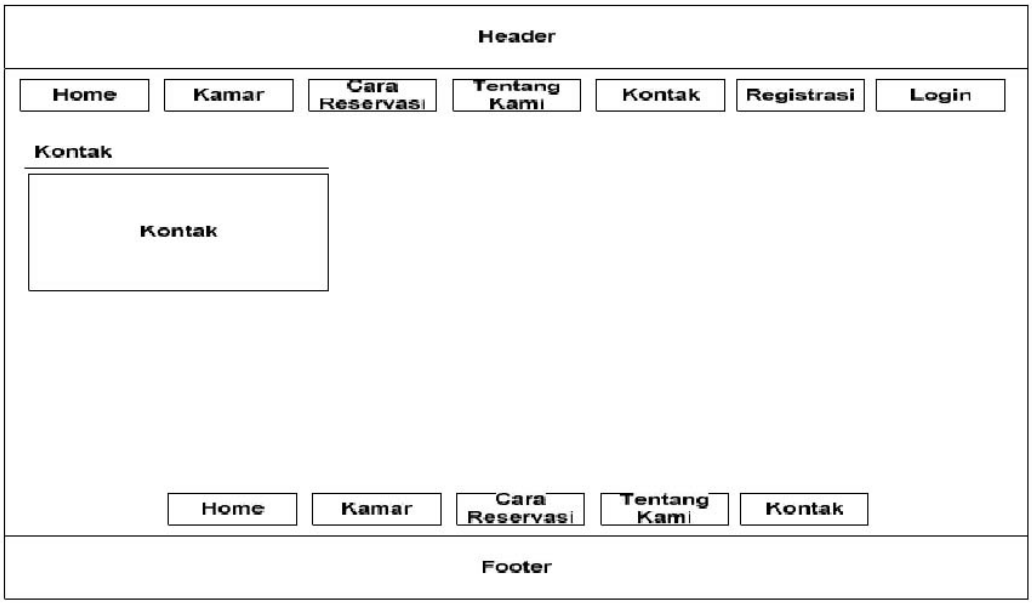
Gambar 4.15 Tampilan Cara Reservasi User

d. Tampilan Tentang Kami User



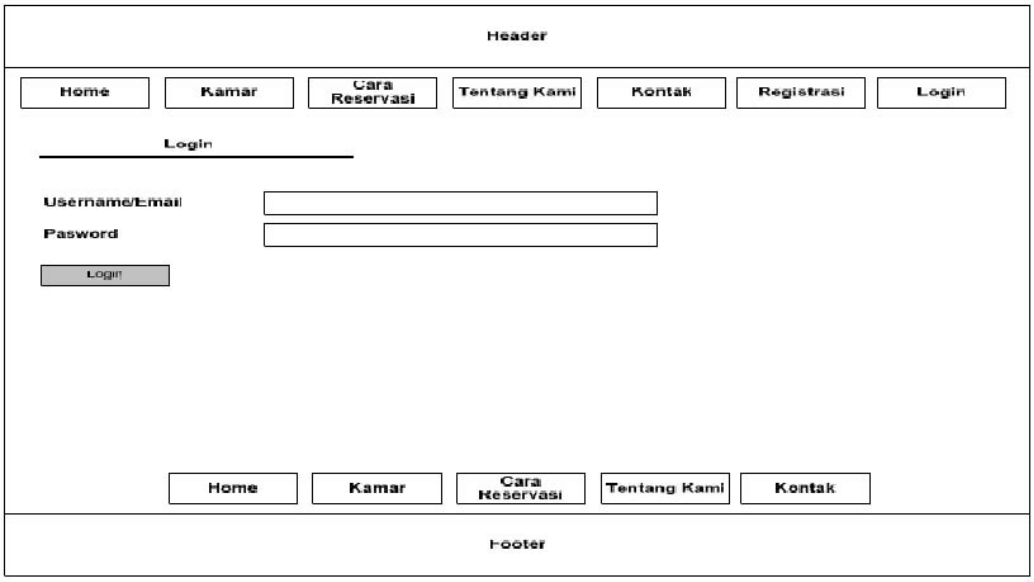
Gambar 4.16 Tampilan Tentang Kami User

e Tampilan Kontak User



Gambar 4.17 Tampilan Kontak User

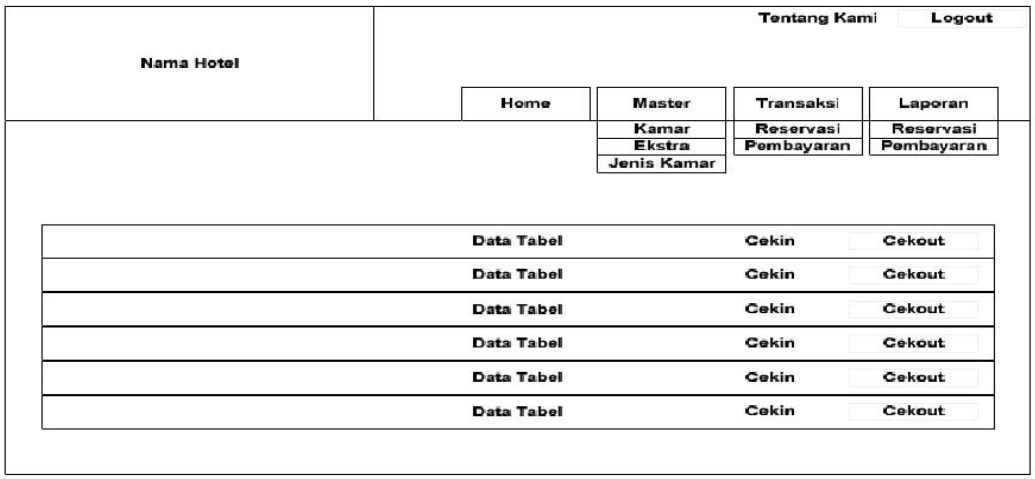
f. Tampilan Login User



Gambar 4.18 Tampilan Login User

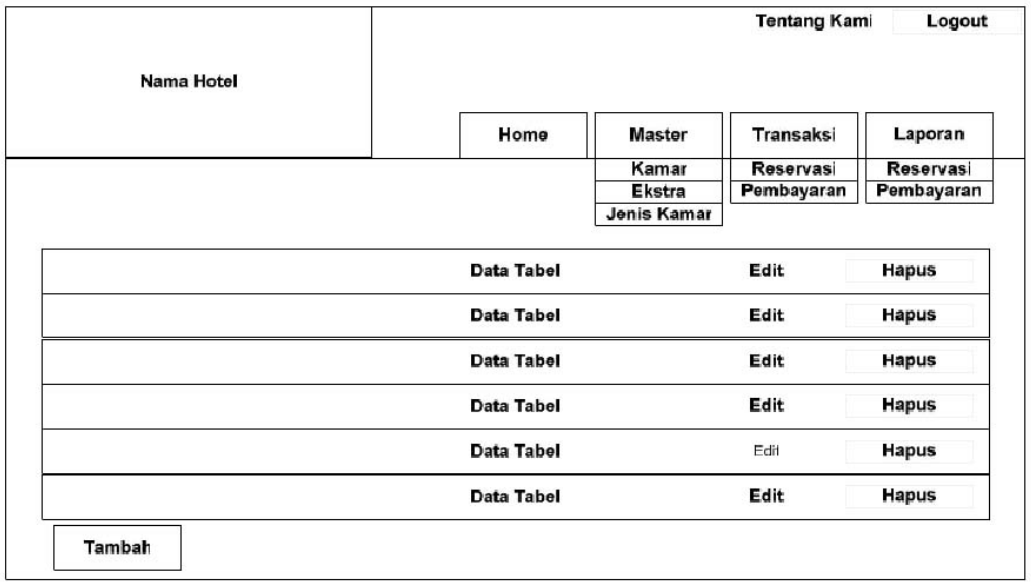
2. Tampilan Admin

a. Tampilan home Admin



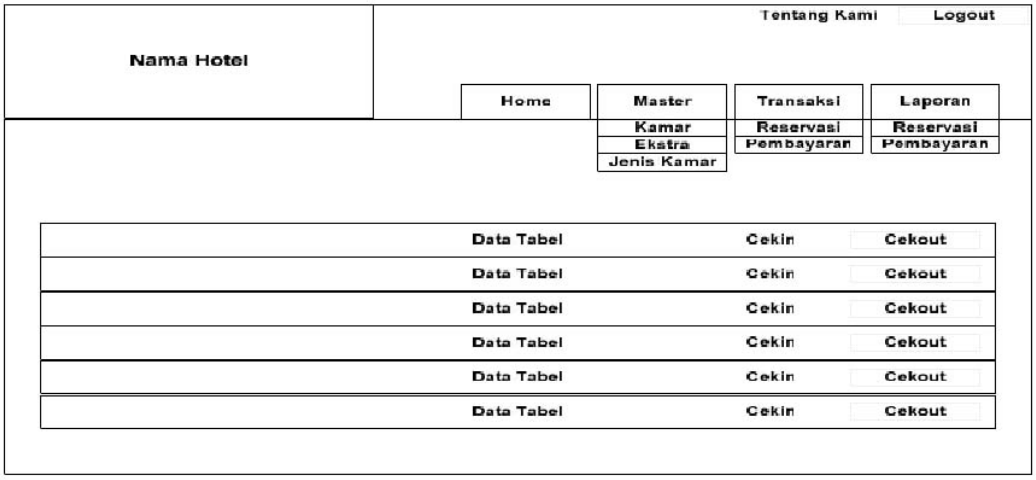
Gambar 4.19 Tampilan home Admin

b. Tampilan Kamar Admin



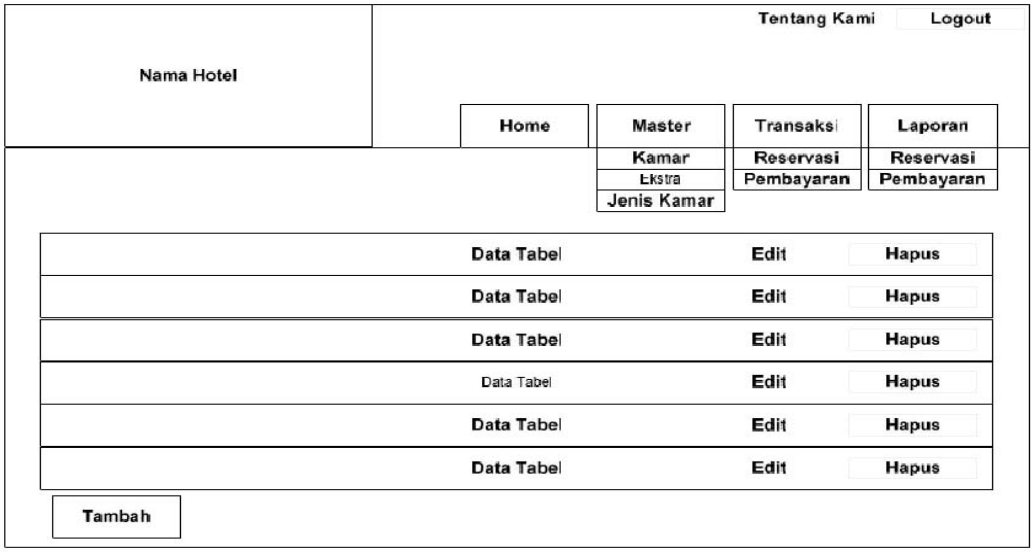
Gambar 4.20 Tampilan Kamar Admin

c. Tampilan Reservasi Admin



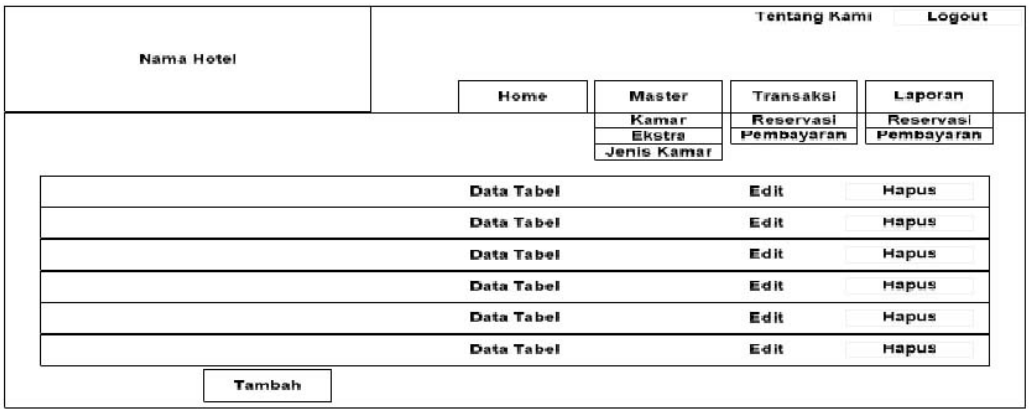
Gambar 4.21 Tampilan reservasi Admin

e. Tampilan Extra Admin



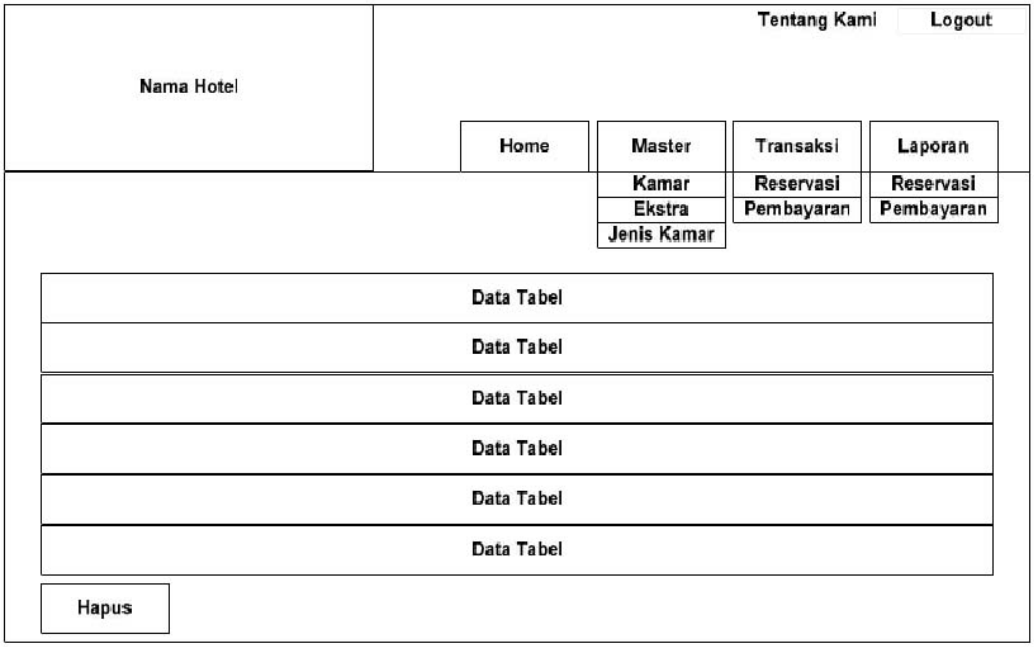
Gambar 4.22 Tampilan Extra Admin

f. Tampilan Jenis Kamar Admin



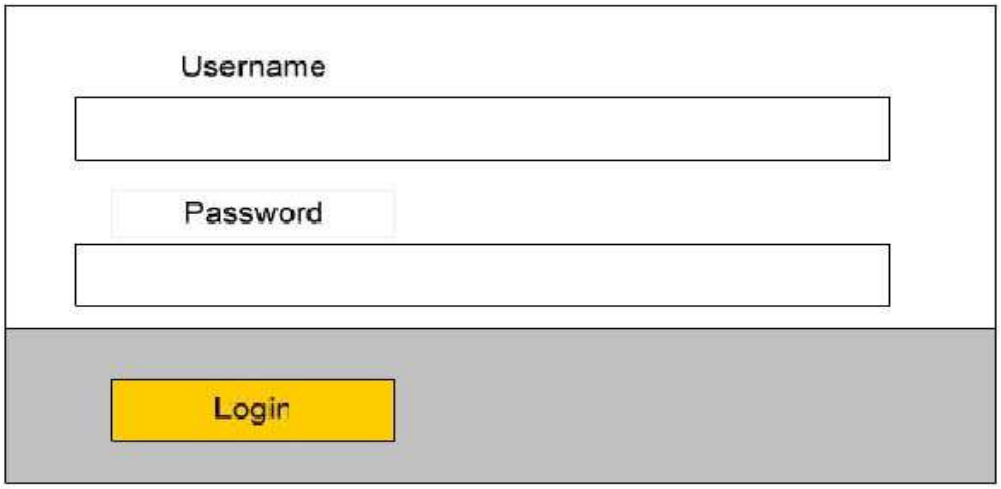
Gambar 4.23 Tampilan Jenis Kamar Admin

e. Tampilan Pembayaran Admin



Gambar 4.24 Tampilan PembayaranAdmin

g. Tampilan Login Admin

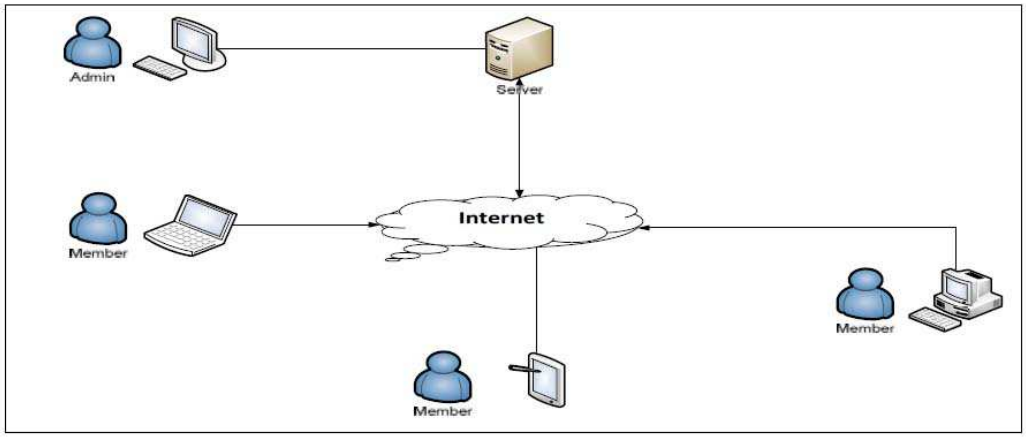


Gambar 4.25 Tampilan Login Admin

4.2.3 Perancangan Arsitektur Jaringan

Arsitekrur jaringan merupakan sebuah sistem dimana sistem tersebut terdiri atas komputer (dapat berupa PC, laptop, handphone), software (sistem operasi, aplikasi) dan perangkat jaringan lainnya yang bekerja bersama-sama untuk mencapai suatu tujuan yang sama. Suatu jaringan ini akan saling berhubungan untuk menghasilkan informasi yang dibutuhkan oleh *user*.

Arsitektur jaringan yang digunakan yaitu jaringan internet, dimana jaringan internet ini merupakan jaringan komputer yang bisa dikategorikan sebagai WAN, menghubungkan berjuta komputer diseluruh dunia, tanpa batas negara, dimana setiap orang yang memiliki komputer dapat bergabung ke dalam jaringan ini hanya dengan melakukan koneksi ke penyedia layanan internet (internet service provider / ISP). Berikut ini perancangan arsitektur jaringan yang digunakan pada aplikasi Grha Ciumbuleuit Guest House



Gambar 4.26 Perancangan Arsitektur Jaringan

4.3 Implementasi

Implementasi *prototype* Sistem Informasi Reservasi Kamar Berbasis Web pada Grha Ciumbuleuit Guest House Bandung dilakukan dengan menggunakan bahasa pemrograman PHP (PHP Hypertext Preprotocol) dan menggunakan web server Apache friend. Adapun *database* yang digunakan dalam implementasi Sistem Informasi Reservasi Kamar Berbasis Web pada Grha Ciumbuleuit Guest House Bandung ini adalah MySQL. Adapun sistem operasi yang digunakan adalah Windows 7.

4.3.1 Batasan Implementasi

Dalam mengimplementasikan perangkat lunak pemantau dan pengendalian aplikasi ini. Ada beberapa hal yang perlu menjadi batasan implementasi yaitu :

1. Basis data yang digunakan dalam pengimplementasian ini adalah MySQL.
2. Kebutuhan system dibangun berdasarkan kebutuhan tamu dan kebutuhan operasional dari pelayanan kamar.

4.3.2 Implementasi Perangkat Lunak

Untuk implementasi perangkat lunak ini menggunakan Apachefriends Xampp 1.5 WIN 32, Adobe Dreamweaver CS5.5 dan MySQL 1.4. Apachefriends Xampp 1.5 WIN 32 dipilih sebagai perangkat lunak untuk webserver karena faktor kecepatan, kinerja yang stabil, dan performansi. PHP dan dreamweaver 8 dipilih sebagai perangkat lunak pengembang karena menyediakan fasilitas yang memadai dan membuat perangkat lunak yang berbasis web. Sementara itu MySQL digunakan sebagai pengembang dalam pembuatan basis data.

4.3.3 Implementasi Perangkat Keras

Untuk dapat menjalankan *website* yang dirancang maka dibutuhkan suatu perangkat keras sebagai penunjangnya. Adapun perangkat kebutuhan perangkat keras yang dibutuhkan tersebut adalah sebagai berikut :

1. *Server* :

   a. Processor Minimal Pentium IV atau yang sekelasnya.
   b. Menggunakan minimal RAM 256 MB.
   c. *Harddisk* 40 Gb.
   d. *Keyboard*, Monitor dan *Mouse.*

2. *Client*

   a. Processor Minimal Pentium III atau yang sekelasnya.
   b. Menggunakan minimal RAM 256 MB.
   c. *Harddisk* 40 Gb.
   d. *Keyboard*, Monitor dan *Mouse.*

4.3.4 Implementasi Basis Data

Implementasi basis data yang digunakan pada aplikasi reservasi kamar pada Hotel Karmila Bandung dilakukan dengan menggunakan MySql. Implementasinya adalah sebagai berikut :

```
-- Database: `graha_db`
CREATE DATABASE ` graha_db ` DEFAULT CHARACTER SET latin1
COLLATE
latin1_swedish_ci;
USE `karmila`;
--------------------------------------------------------------------------------------------------
----
CREATE TABLE `admin` (
`username` varchar(50) NOT NULL default '',
`bagian` varchar(50) default NULL,
`password` varchar(100) default NULL,
PRIMARY KEY (`username`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
--------------------------------------------------------------------------------------------------
----
CREATE TABLE `booking` (
`id_booking` int(11) NOT NULL auto_increment,
`tgl_pesan` date default NULL,
`no_ktp` varchar(40) default NULL,
`email` varchar(255) default NULL,
```

```
`total_bayar` double(12,0) default NULL,
`total_bayar_ekstra` double(12,0) default NULL,
PRIMARY KEY (`id_booking`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
--------------------------------------------------------------------------------------------------------
----
CREATE TABLE `det_booking` (
`id_det_booking` int(11) NOT NULL auto_increment,
`id_booking` int(11) default NULL,
`no_kamar` varchar(10) default NULL,
`tgl_booking1` date default NULL,
`tgl_booking2` date default NULL,
`jam_checkin` time default NULL,
`jam_checkout` time default NULL,
`dp` double(12,0) default NULL,
`pengembalian` double(12,0) default NULL,
`status` varchar(20) default NULL,
PRIMARY KEY (`id_det_booking`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
--------------------------------------------------------------------------------------------------------
----
CREATE TABLE `det_kamar` (
`id_det_kamar` int(11) NOT NULL auto_increment,
`no_kamar` varchar(10) default NULL,
```

```
`gambar_kamar` varchar(255) default NULL,
PRIMARY KEY (`id_det_kamar`)
) ENGINE=InnoDB AUTO_INCREMENT=31 DEFAULT CHARSET=latin1;
--------------------------------------------------------------------------------------------------------
----
CREATE TABLE `ekstra` (
`id_ekstra` int(11) NOT NULL auto_increment,
`jenis_ekstra` varchar(100) default NULL,
`nama_ekstra` varchar(100) default NULL,
`harga_ekstra` double(12,0) default NULL,
PRIMARY KEY (`id_ekstra`)
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=latin1;
--------------------------------------------------------------------------------------------------------
----
CREATE TABLE `jenis` (
`id_jenis` varchar(10) NOT NULL default '',
`nama_jenis` varchar(100) default NULL,
`gambar` varchar(255) default NULL,
`deskripsi` text,
`jml_kamar` int(11) default NULL,
PRIMARY KEY (`id_jenis`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
--------------------------------------------------------------------------------------------------------
----
```

```
CREATE TABLE `kamar` (
`no_kamar` varchar(10) NOT NULL default '',
`id_jenis` varchar(10) default NULL,
`deskripsi` text,
`harga` double(12,0) default NULL,
`lokasi_kamar` int(11) default NULL,
`status` varchar(20) default NULL,
PRIMARY KEY (`no_kamar`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
--------------------------------------------------------------------------------------------------------
----
CREATE TABLE `member` (
`no_ktp` varchar(40) NOT NULL default '',
`email` varchar(255) default NULL,
`password` varchar(100) default NULL,
`nama` varchar(100) default NULL,
`alamat` varchar(255) default NULL,
`kota` varchar(100) default NULL,
`jk` varchar(20) default NULL,
`no_telp` bigint(20) default NULL,
PRIMARY KEY (`no_ktp`),
UNIQUE KEY `email` (`email`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
--------------------------------------------------------------------------------------------------------
----
CREATE TABLE `pembayaran` (
`id_booking` int(11) NOT NULL default '0',
`tgl_transfer` date default NULL,
`bank_tujuan` varchar(100) default NULL,
`bank_asal` varchar(100) default NULL,
`no_resi` int(11) default NULL,
`no_rek` bigint(20) default NULL,
`an_rek` varchar(100) default NULL,
`dp` double(12,0) default NULL,
`pelunasan` double(12,0) default NULL,
`status` varchar(100) default NULL,
PRIMARY KEY (`id_booking`),
KEY `id_booking` (`id_booking`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
--------------------------------------------------------------------------------------------------------
----
CREATE TABLE `tambahan` (
`id_tambahan` int(11) NOT NULL auto_increment,
`id_booking` int(11) default NULL,
`no_kamar` varchar(10) default NULL,
`id_ekstra` int(11) default NULL,
`harga` double(12,0) default NULL,
```

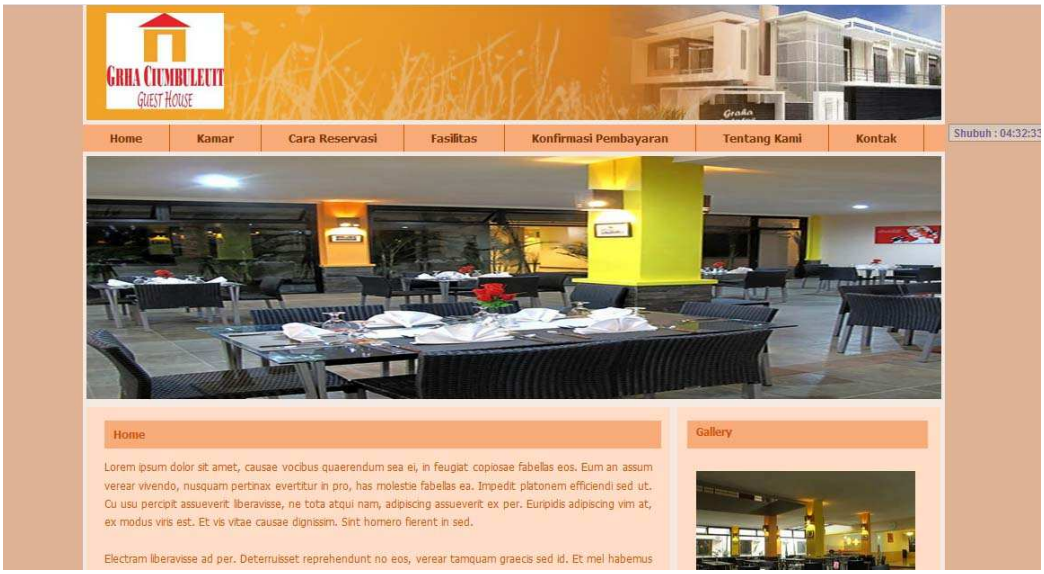
```
PRIMARY KEY (`id_tambahan`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
--------------------------------------------------------------------------------------------------------
----
CREATE TABLE `temp_booking` (
`id_temp` int(11) NOT NULL auto_increment,
`no_kamar` varchar(10) default NULL,
`tgl_booking1` date default NULL,
`tgl_booking2` date default NULL,
`email` varchar(100) default NULL,
PRIMARY KEY (`id_temp`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

4.3.5 Implementasi Antar Muka

Dalam PHP, implementasi antarmuka dilakukan dengan sebuah webpage yang

berekstensi PHP. Setiap halaman dalam perangkat lunak dibuat file program yang ditulis dengan berekstensi.php. Berikut ini adalah implementasi dari setiap halaman yang dibuat dan dibedakan berdasarkan Kebutuhan User dan Administrator Grha Ciumbuleuit Guest House Bandung

4.3.5.1 Implementasi Halaman Utama



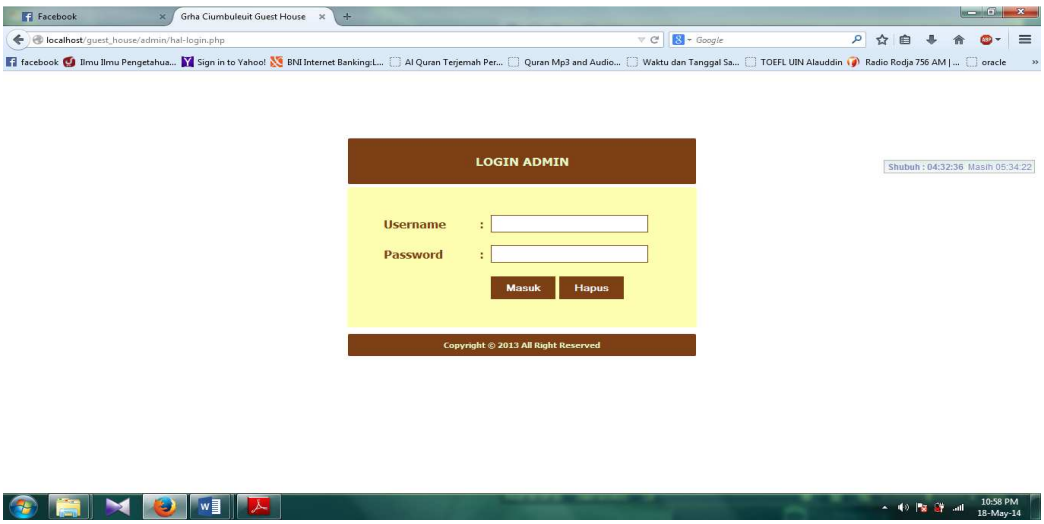
Gambar 4.27 Halaman Utama

Tabel 4.1 Implementasi Antarmuka User

Menu	**Deskripsi**	**Nama File**
Home	File program untuk menangani halaman utama	Index.php
Kamar	File program untuk menangani informasi tentang kamar dan harga kamar	Index.php
Cara reservasi	File program yang menangani informasi tentang cara reservasi	Index.php
Fasilitas	File program yang	Index.php

	menangani informasi tentang guest house	
Konfirmasi Pembayaran	File program yang Menangani konfirmasi pembayaran dari user	Index.php
Tentang Kami	File program yang memberikan informasi tentang sejarah guest house	Index.php
Kontak	File program yang menangani informasi kontak guest house yang bisa dihubungi	Index.php

4.3.5.2 Implementasi Halaman login Admin



Gambar 4.28 Implementasi Halaman login Admin

4.3.5.3 Implementasi Halaman Admin setelah login



Gambar 4.29 Implementasi Halaman login Admin

4.3.6 Implementasi Instalasi Program

Berikut ini adalah implementasi program

4.3.6.1 Implementasi Localhost

Berikut ini adalah implementasi instalasi program untuk menggunakan program Sistem

Informasi Pelayanan Kamar berbasis *web* pada Saidah pada localhost. Langkah-langkahnya adalah sebagai berikut :

1. Pilih menu control panel Apache Friend dengan cara klik start –Apache friend – xampp – XAMPP Control Panel.



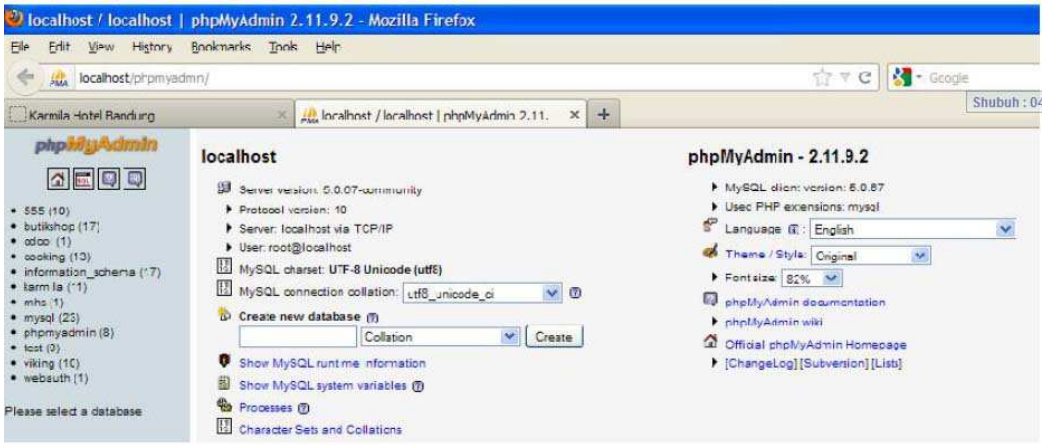
Gambar 4.30 XAMPP Control Panel

1. Buka web browser, lalu ketikan alamat http://localhost/guest_house



Gambar 4.31 Memilih URL

2. Import database



Gambar 4.32 Halaman phpMyAdmin

4.3.7 Penggunaan Program

Penggunaan program merupakan kegiatan yang bertujuan untuk menerangkan secara singkat bagaimana penggunaan program Sistem Informasi Reservasi Kamar berbasis web pada Grha Ciumbuleuit Guest House Bandung. Adapun cara pengoperasiannya seperti yang akan dijelaskan berikut ini :

4.3.7.1 Penggunan Program User

Berikut adalah petunjuk penggunaan program Sistem Informasi Pelayanan Kamar berbasis web pada Grha Ciumbuleuit Guest House Bandung bagi user.

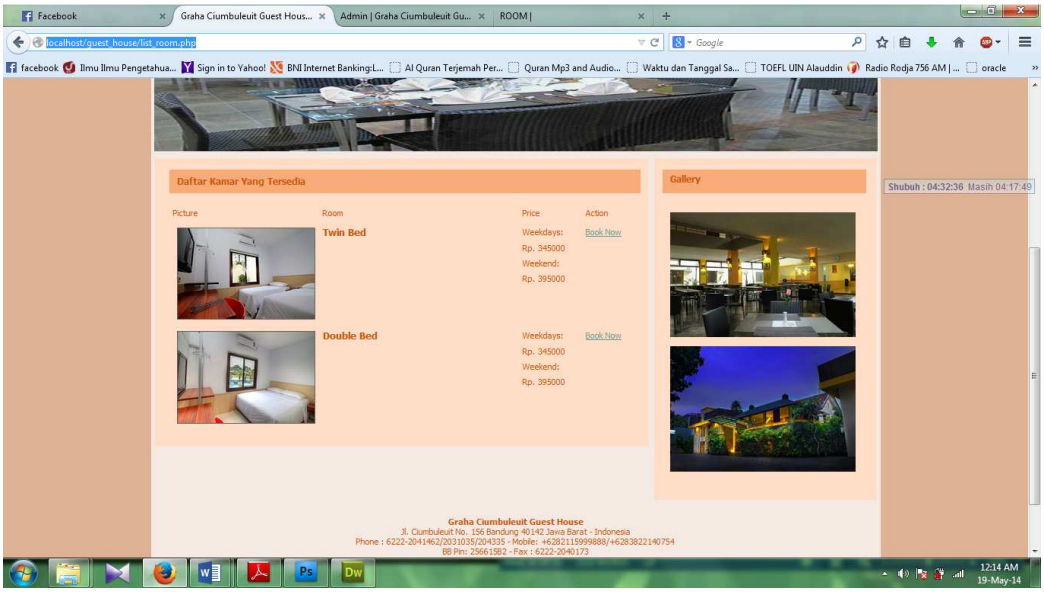
1. Home

Form Menu Utama, merupakan form yang pertama kalinya tampil pada saat program di running. Bentuk tampilannya adalah sebagai berikut.



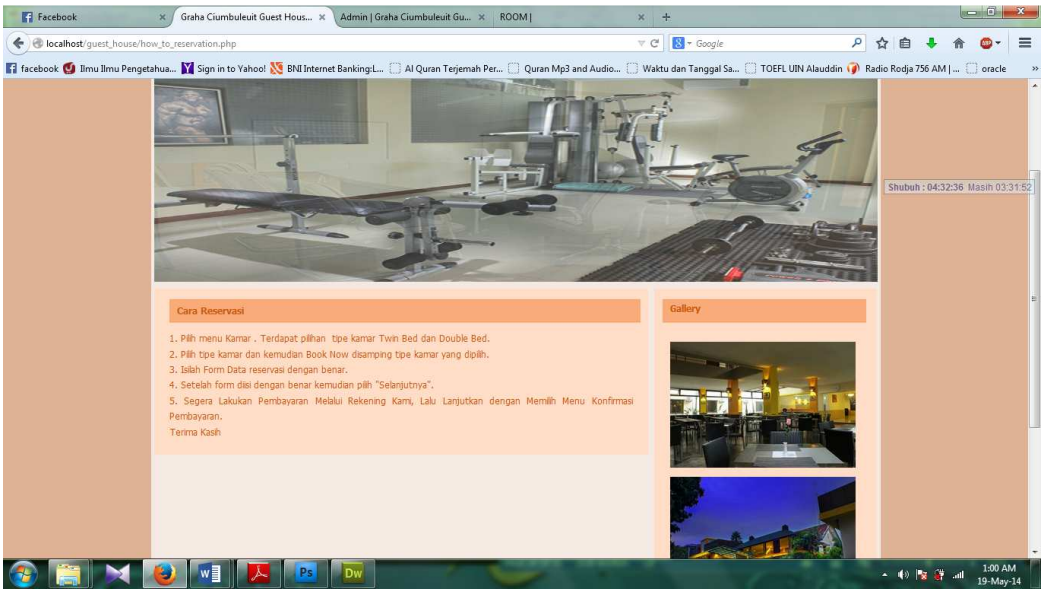
Gambar 4.33 Halaman Utama

2. Form Kamar



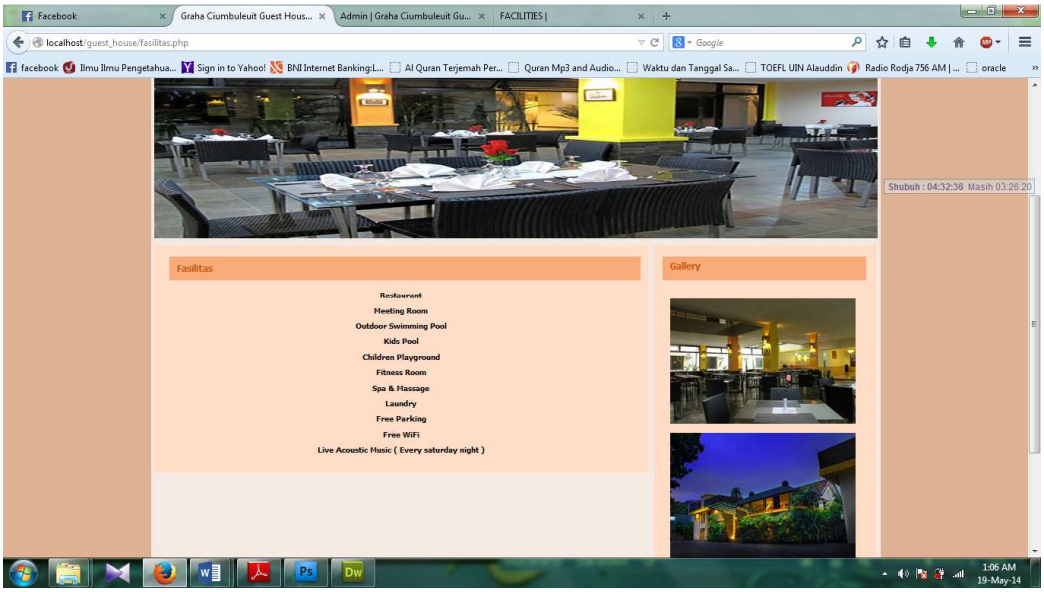
Gambar 4.34 Form Kamar

2. Form cara reservasi



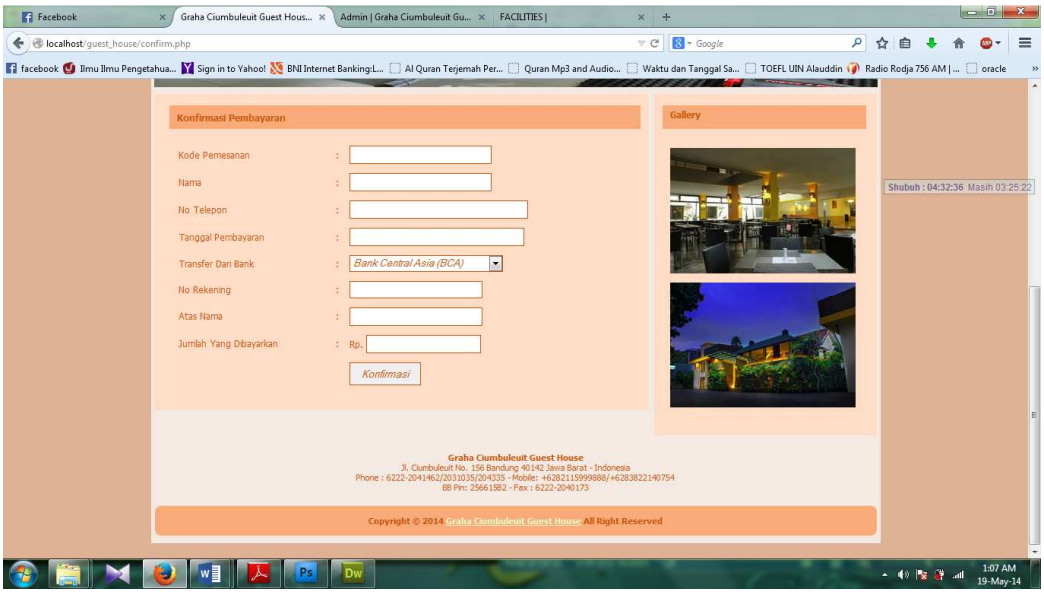
Gambar 4.35 Form cara reservasi

3. Form Fasilitas



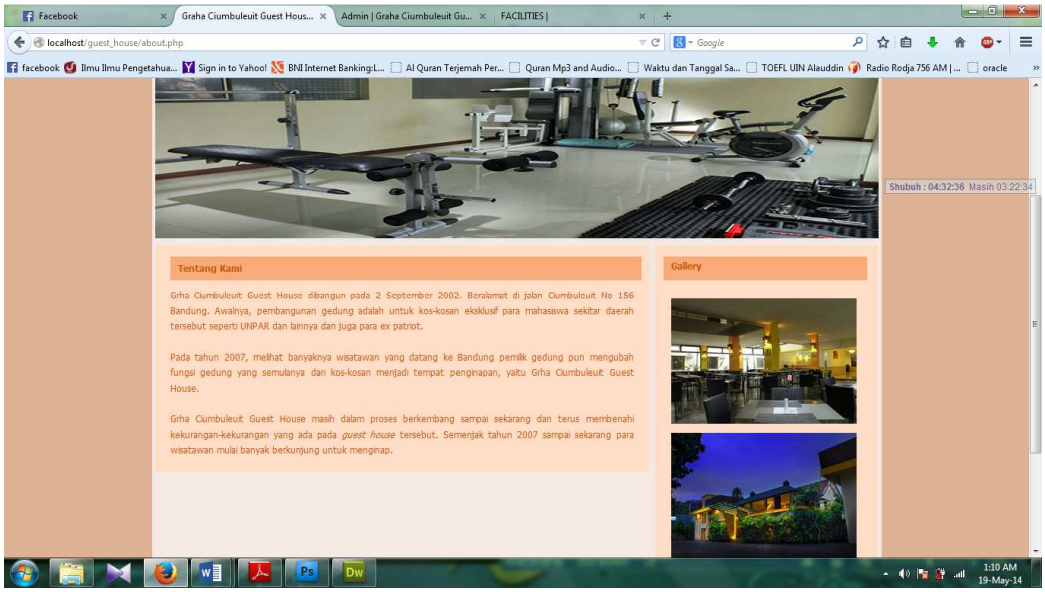
Gambar 4.36 Form Fasilitas

4. Form Konfirmasi Pembayaran



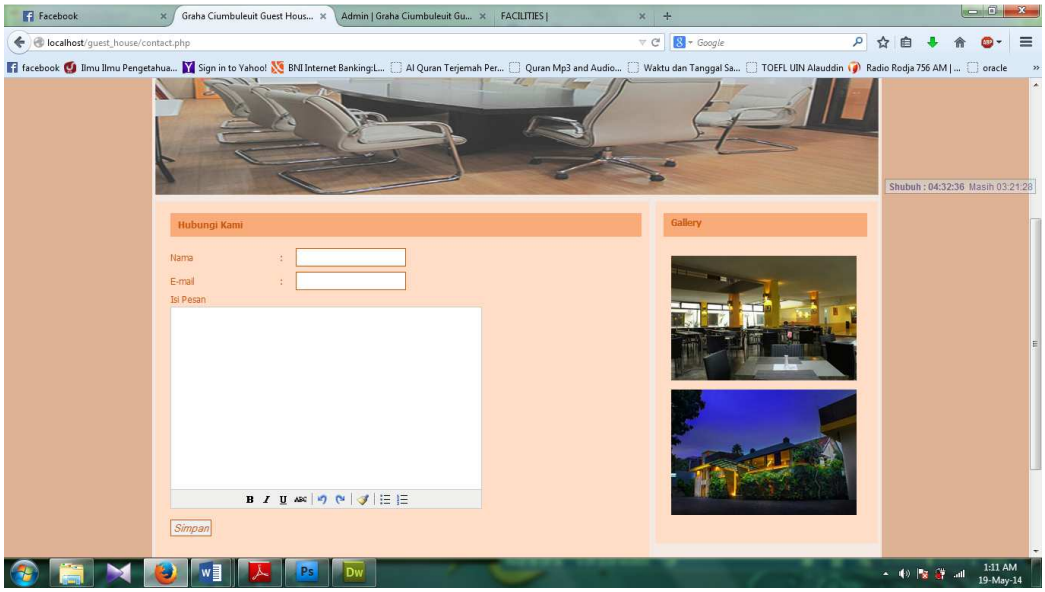
Gambar 4.37 Form Konfirmasi Pembayaran

5. Form Tentang Kami



Gambar 4.38 Form Tentang Kami

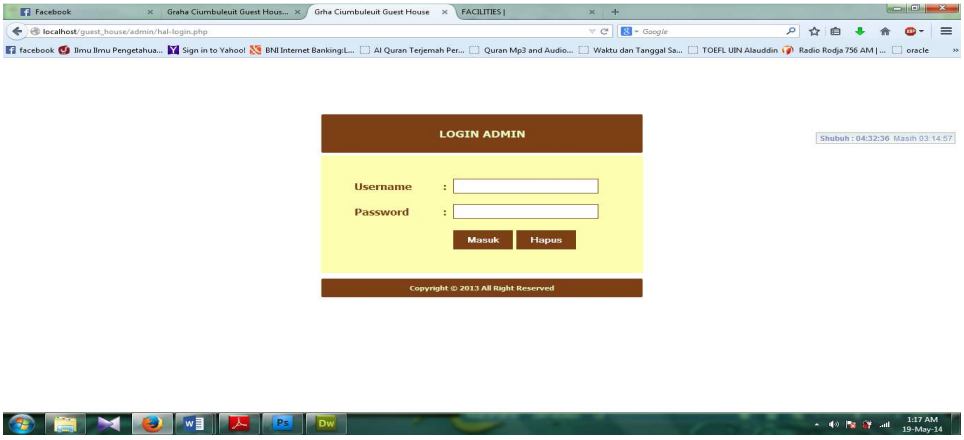
6. Form Kontak



Gambar 4.39 Form Kontak

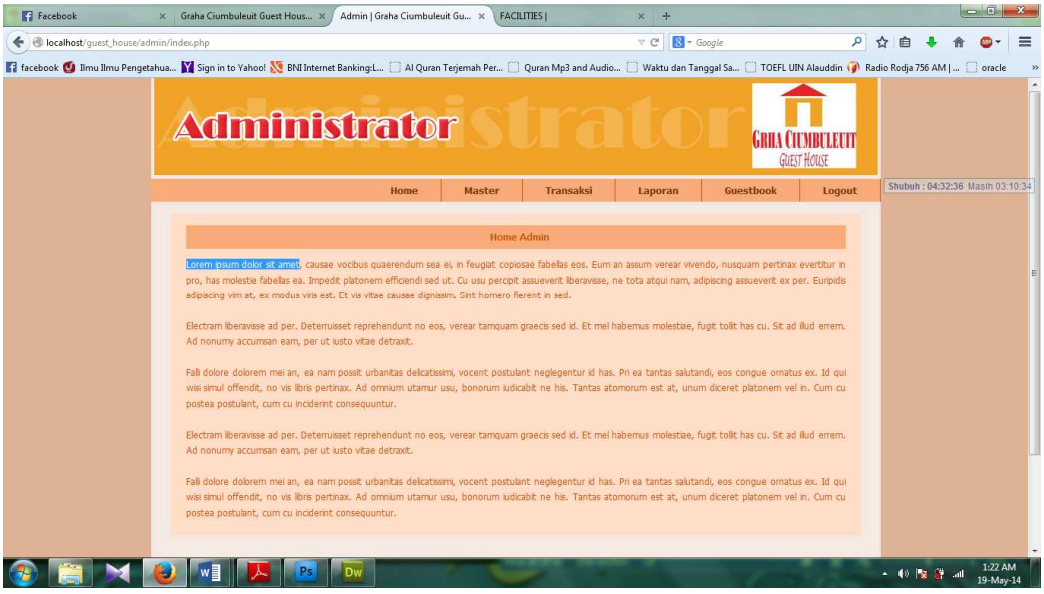
4.3.7.2 Penggunan Program Admin

1. Form Login Admin



Gambar 4.40 Form login dmin

2. Tampilan home admin setelah login



Gambar 4.41 Form tampilan Home Admin setelah login

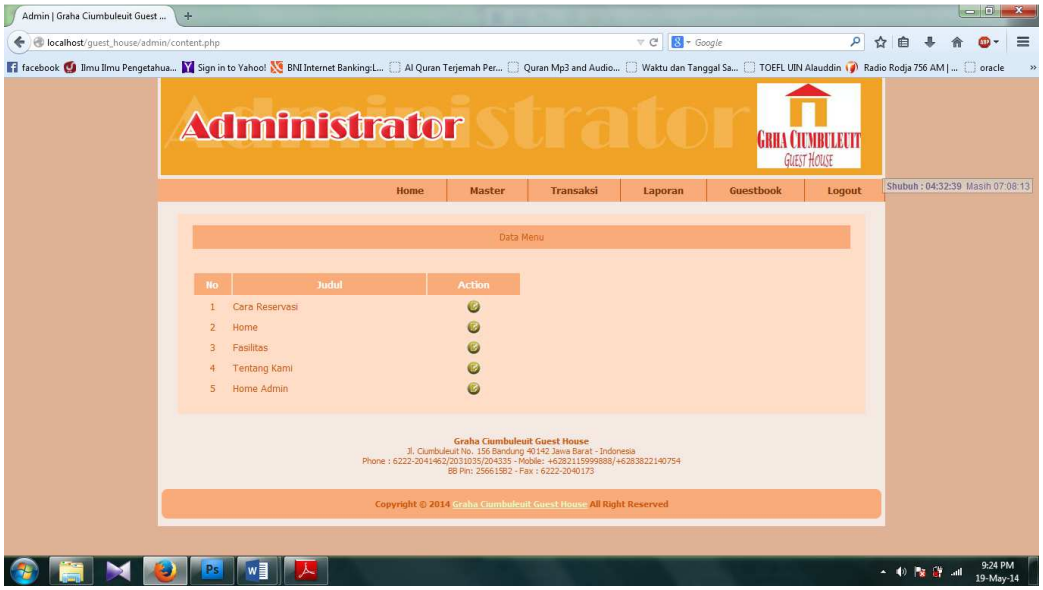
3. Tampilan Master Admin

a. Tampilan Data Admin



Gambar 4.42 Form Tampilan Data Admin

b. Tampilan Data Menu



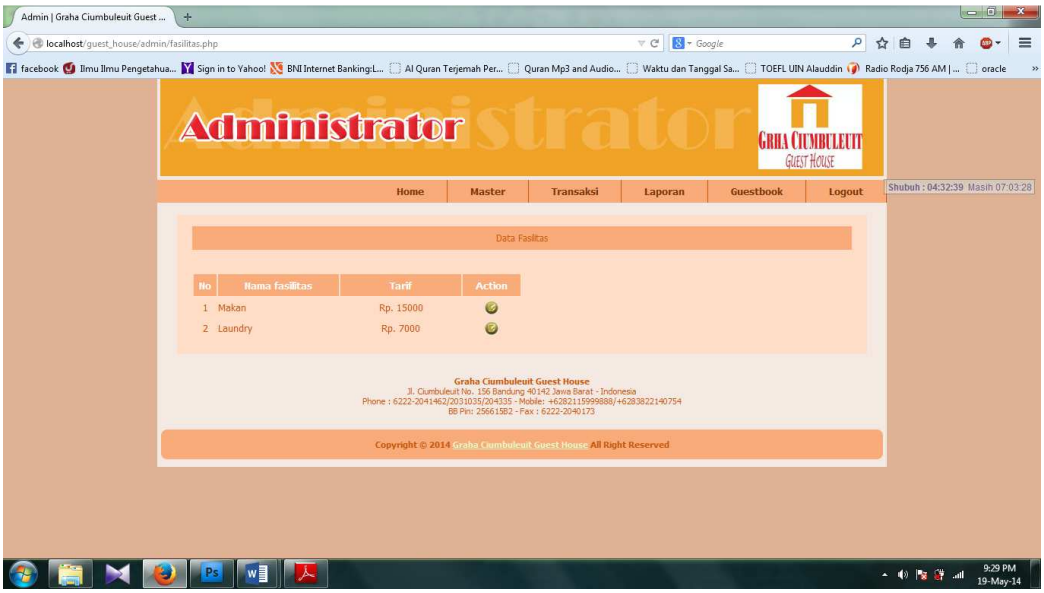
Gambar 4.43 Form Tampilan Data Menu

c. Tampilan Data Kamar



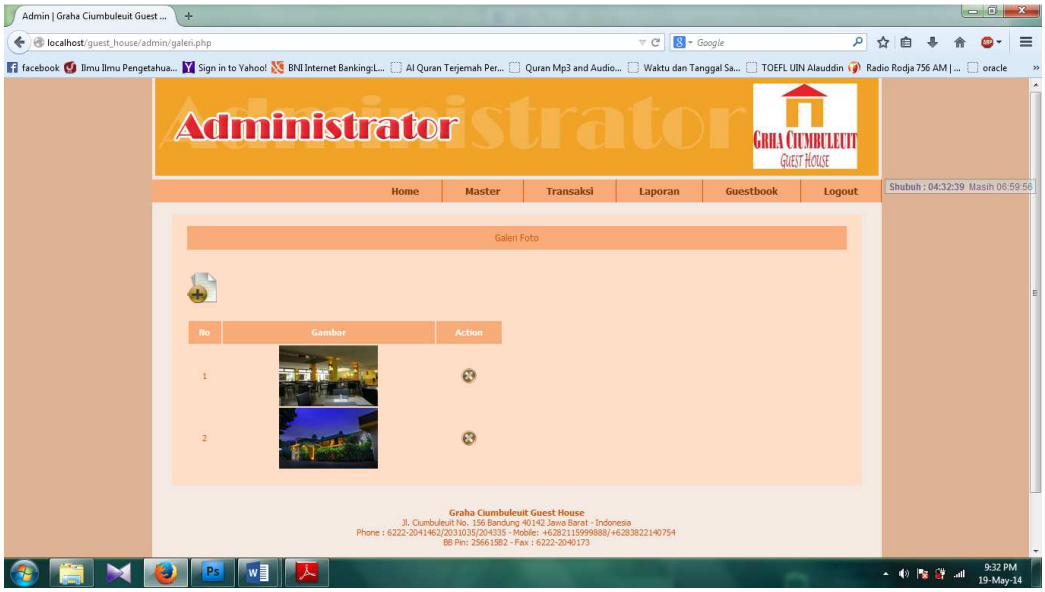
Gambar 4.44 Form Tampilan Data Kamar

d. Tampilan Data Fasilitas



Gambar 4.45 Form Tampilan Data Fasilitas

e. Tampilan Galeri Foto



Gambar 4.46 Form Tampilan Galeri Foto

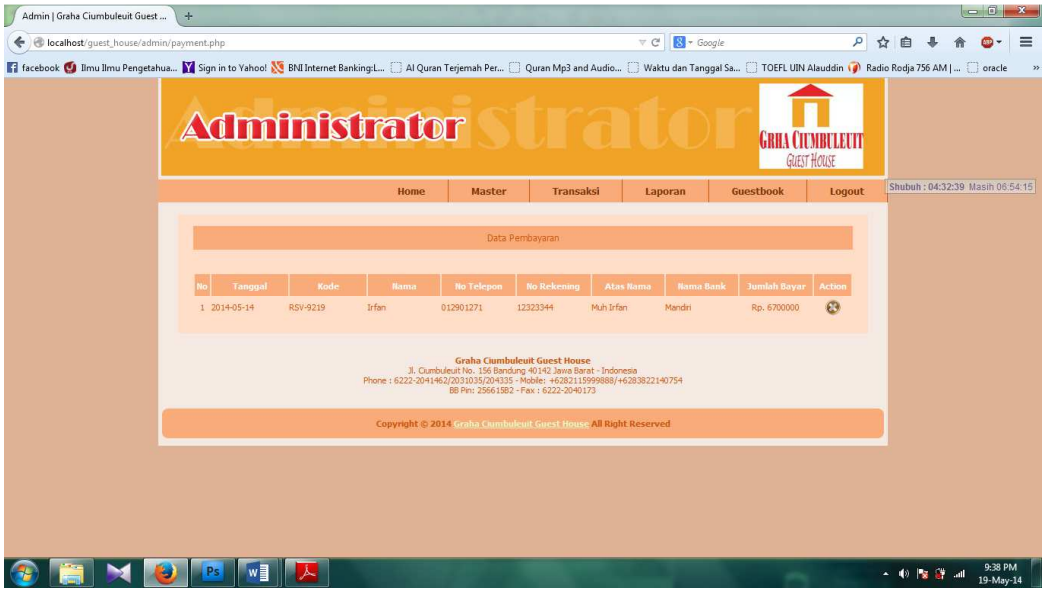
4. Tampilan Transaksi Admin

a. Tampilan Reservasi Admin



Gambar 4.47 Form Tampilan Reservasi Admin

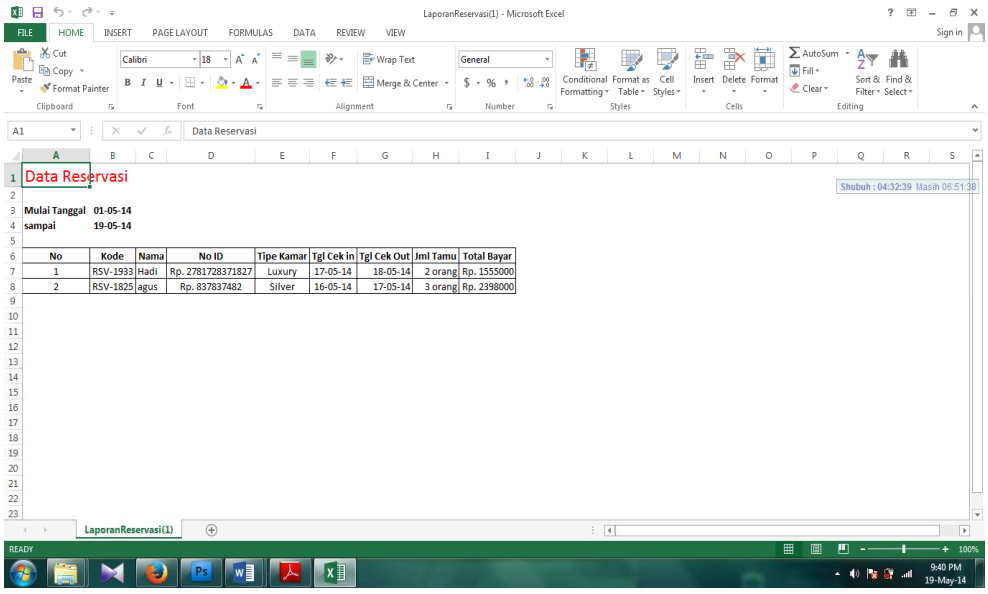
b. Tampilan Pembayaran Admin



Gambar 4.48 Form Tampilan Pembayaran Admin

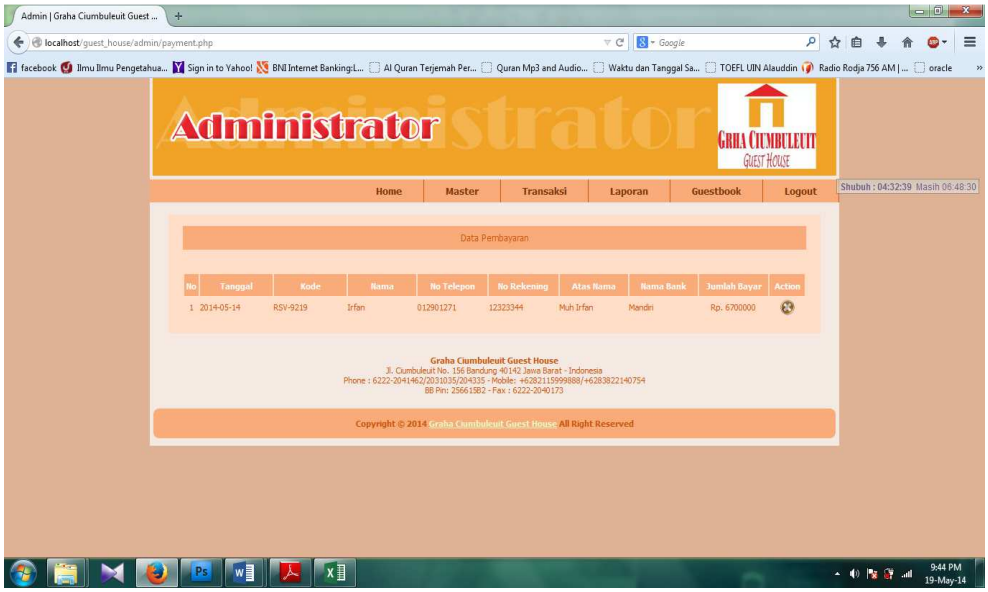
5. Tampilan Laporan Admin

a. Laporan Data Reservasi



Gambar 4.49 Form Tampilan Laporan Data Reservasi

b. Laporan Data Pembayaran



Gambar 4.50 Form Tampilan Laporan Data Pembayaran

4.4 Pengujian

Pengujian merupakan bagian yang penting dalam siklus pembangunan perangkat lunak. Tujuan dari pengujian adalah untuk menjamin perangkat lunak yang di bangun memilki kualitas yang handal, yaitu mampu mempresentasikan kajian pokok dari spesifikasi, analisis, perancangan dan pengkodean dari perangkat lunak itu sendiri. Dalam analisis dan perancangan ini, penulis menggunakan 3 (tiga) faktor pengujian yang dilakukan secara internal terhadap *Web* Grha Ciumbuleuit Guest House Bandung pada *module* reservasi *online*.

4.4.1 Rencana Pengujian

Pengujian perangkat lunak Sistem Informasi Pelayanan Kamar pada Grha Ciumbuleuit Guest Bandung ini menggunakan metode *black box*. Pengujian *black box* berfokus pada persyaratan fungsional perangkat lunak yang dibuat.

Tabel 4.2. Rencana Pengujian pada Sistem Informasi Reservasi Kamar pada Grha Ciumbuleuit Guest House Bandung

Kelas Uji	Butir Uji	Tingkat Pengujian	Jenis Pengujian
Login	Hak akses pengguna program	Modul	*Black box*
kamar	Menampilkan informasi harga dan tipe kamar	Modul	*Black box*

4.4.2 Kasus dan Hasil Pengujian

Kasus dan hasil pengujian dibuat untuk mengetahui apakah Sistem Informasi Pelayanan Kamar berbasis web pada Hotel Karmila Bandung dapat berjalan dengan baik atau tidak.

4.4.2.1 Kelas Uji Login

Berikut ini adalah tabel pengujian *login Admin* untuk pengecekkan *user* yang telah terdaftar :

Tabel 4.3 Pengujian Login Admin

Kasus dan Hasil Uji (Data normal)			
Data Masuk	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
User name : admin *Password* : admin	Dapat Login dan masuk ke halaman utama admin	Dapat login dan masuk ke halaman utama admin	[ ] Diterima [ ] Ditolak
Login	Dapat masuk ke halaman utama admin	Tombol Login dapat berfungsi sesuai yang diharapkan	[ ] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data salah)			
Data Masuk	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
User	Tidak dapat	*User* tidak dapat	[ ] Diterima
name : salah *Password* : salah	Login dan Menampilkan pesan kesalahan	login jika salah Username dan atau Password	[ ] Ditolak

4.4.2.2 Kelas Uji Check in

Berikut adalah tabel pengujian *check in* kamar :

Tabel 4.4 Pengujian check in kamar

Kasus dan Hasil Uji (Data Normal)			
Data Masuk	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Memilih menu link check in	Menampilkan form check in	Link berfungsi	[ ] Diterima [ ] Ditolak
Memilih nomor kamar	Menyimpan data check in ke dalam database	Tombol check in berfungsi	[ ] Diterima [ ] Ditolak
Mencetak bukti check in	Menampilkan bukti check in	Link berfungsi dengan baik	[ ] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data salah)			
Data Masuk	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Check box kosong	Menampilkan kotak pesan kesalahan	Sesuai dengan yang diharapkan	[ ] Diterima [ ] Ditolak
Jumlah Kamar tidak sesuai dengan data reservasi	Menampilkan kotak pesan kesalahan	Sesuai dengan yang diharapkan	[ ] Diterima [ ] Ditolak

4.4.2.3 Kelas Uji Check out

Berikut adalah tabel pengujian *check out* kamar :

Tabel 4.5 Pengujian check out kamar

Kasus dan Hasil Uji (Data Normal)			
Data Masuk	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Memilih link check out	Menyimpan data check out kedalam database	Link berfungsi	[ ] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data salah)			
Data Masuk	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Bukti check out tidak tampil	Menampilkan kotak pesan kesalahan	Sesuai dengan yang diharapkan	[ ] Diterima [ ] Ditolak

4.4.2.4 Kelas Uji Tanggal Reservasi

Berikut adalah tabel pengujian check tanggal reservasi :

Tabel 4.6 Pengujian tanggal reservasi

Kasus dan Hasil Uji (Data normal)			
Data Masuk	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Tanggal check in dan tanggal check out	Menampilkan data kamar *available*	Menampilkan data kamar *available*	[ ] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data salah)			
Data Masuk	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Tanggal check in kurang dari satu hari	Tidak dapat melanjutkan reservasi dan Menampilkan pesan kesalahan	*User* tidak dapat melanjutkan proses	[ ] Diterima [ ] Ditolak
Tanggal Check in lebih kecil dari tanggal check out	Tidak dapat melanjutkan reservasi dan Menampilkan pesan kesalahan	*User* tidak dapat melanjutkan proses	[ ] Diterima [ ] Ditolak

4.4.3 Kesimpulan Hasil Pengujian

Berdasarkan hasil pengujian dengan kasus uji *sample* di atas, dapat ditarik kesimpulan bahwa perangkat lunak bebas dari kesalahan sintaks dan secara fungsional mengeluarkan hasil yang sesuai dengan yang diharapkan.