BAB IV

HASIL DAN PEMBAHASAN

4.1 Perancangan Sistem

Perancangan sistem ini dibuat sebagai tahapan untuk mempersiapkan proses implementasi sistem yang diinginkan, dan untuk menggambarkan secara jelas proses-proses yang diinginkan oleh pengguna. Sesuai dengan metode pendekatan yang akan digunakan adalah pendekatan *object oriented*, maka model yang digunakan untuk menggambarkan seluruh proses dan objek adalah dengan menggunakan UML.

4.1.1 Tujuan Perancangan Sistem

Tujuan dari perancangan sistem adalah sebagai gambaran umum pengembangan sistem usulan yang dapat memberikan spesifikasi sistem usulan perangkat lunak kepada pengguna dengan menganalisis kelemahan-kelemahan dalam sistem yang sudah ada yang mengembangkan sistem yang telah ada demi kepuasan, kenyamanan dan kemudahan.

Adapun tujuan perancangan sistem informasi reservasi berbasis web online ini adalah sebagai berikut:

a. Membuat sistem informasi reservasi berbasis web untuk memudahkan masyarakat mengakses pemesanan dan informasi mengenai cottage.
b. Membantu pihak cottage dalam melakukan proses reservasi cottage.
c. Dapat mempercepat proses pencarian kamar cottage yang kosong sehingga memudahkan dalam penyampaian informasi cottage yang ada di seaview

cottage tsb.

d. Dapat memberikan laporan yang dibutuhkan secara cepat, tepat, dan akurat kepada setiap bagian yang membutuhkan pimpinan apabila informasi diperlukan sewaktu-waktu.
e. Dapat mempermudah bagian front office dalam pembuatan laporan tentang total pembayaran yang harus dibayar oleh pengunjung.
f. Memudahkan para pengunjung yang hendak memesan cottage untuk menginap tidak perlu lagi datang langsung ke hotel karena bisa memesan cottage secara online.
g. Mempermudah bagian Front office untuk mengarsipkan data-data pengunjung yang menginap lebih terkomputerisasi sehingga resiko kehilangan data dapat diminimalisir, dan pencarian data apabila dibutuhkan akan cepat terpenuhi.

4.1.2 Gambaran Umum Sistem Yang diusulkan

Sistem informasi reservasi cottage berbasis website ini dapat digambarkan sebagai bentuk sebuah fasilitas yang menyediakan informasi-informasi cottage dari mulai informasi status kamar, fasilitas yang ada pada setiap kamar cottage, tarif dan type cottage, yang semuanya bisa diakses oleh masyarakat dan calon pengunjung yang hendak menginap.

Selain itu sistem informasi ini juga mendukung pemesanan cottage secara online dimana pengunjung yang hendak menginap bisa memesan kamar cottage untuk menginap melalui media internet asal memenuhi beberapa syarat dan ketentuan yang berlaku dari pihak cottage. Sehingga pengunjungtidak perlu datang

lagi untuk sekedar menanyakan informasi kamar atau memesan kamar.

Dengan adanya perancangan sistem informasi reservasi cottage berbasis online ini diharapkan akan memudahkan para pengujung yang ingin memesan kamar cottage untuk menginap, serta memudahkan dari pihak cottage dalam mengelola proses reservasi yang ada di Seaview Cottage Cirebon Waterland.

4.1.3 Perancangan Prosedur Yang Diusulkan

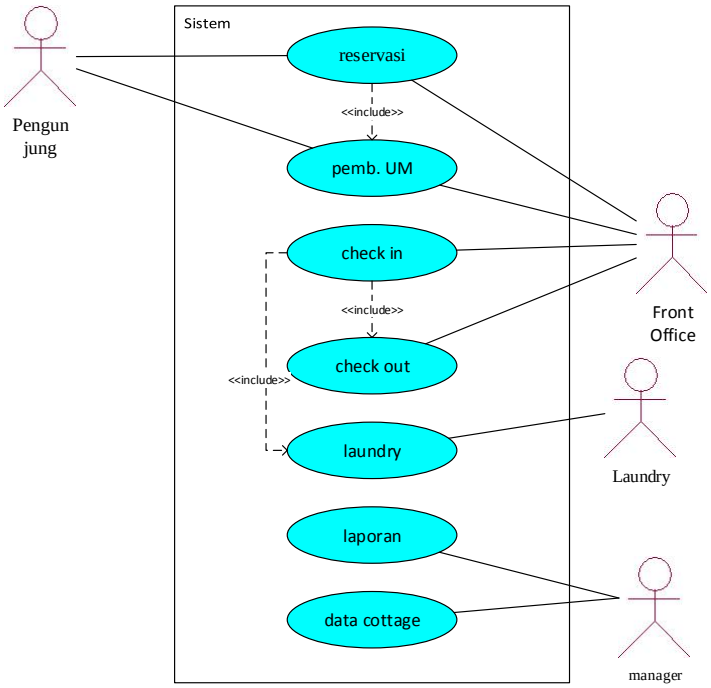
Setelah melihat sistem yang sedang berjalan dan telah mengevaluasi sistem, maka prosedur sistem yang diusulkan adalah:

4.1.3.1Use Case Diagram

*Use case*adalah deskripsi fungsi dari sebuah sistem dari perspektif/sudutpandang para pengguna istem. Untuk mengetahui gambaran interaksi antara system dan actor yang terlibat pada system yang di usulkan, maka digambarkan dengan menggunakan use case diagram sebagai berikut:

1. Use Case Sistem Reservasi Cottage yang diusulkan

Berikut ini penggambaran use case yang diusulkan untuk sistem reservasi *cottage*:



Gambar 4.1

Diagram *Use Case* yang diusulkan

a. Definisi Aktor dan deskripsinya

Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri.

Berikut adalah deskripsi pendefinisian aktor yang diusulkan pada sistem informasi reservasi *cottage*.

Tabel 4.1

Definisi Aktor dan Deskripsi yang Diusulkan

No.	Aktor	Deskripsi
1.	Pengunjung	Pihak yang melakukan reservasi.
2.	*Front Office*	Pihak yang bertugas melakukan dan memiliki hak akses untuk mengelola data reservasi, *check in* dan *check out*.
3.	*Laundry*	Pihak yang bertugas melakukan dan memiliki hak akses untuk mengelola data *laundry*.
4.	*Manager*	Pihak yang memiliki hak akses terkait data *cottage* dan laporan pemasukan dan laporan data reservasi.

b. Definisi Use Case dan Deskripsinya

Use Case merupakan fungsionalitas yang disediakan system sebagai unit-unit yang saling bertukar pesan antar unit atau actor. Berikut adalah deskripsi

pendefinisian *use case* yang diusulkan pada Sistem Informasi Reservasi *Cottage*.

Tabel 4.2

Definisi Use Case dan Deskripsi yang Diusulkan

No.	Aktor	Deskripsi
1.	Reservasi	Merupakan proses pemesanan Cottage yang dilakukan oleh Pengunjung dan dikelola oleh bagian Front Office untuk melakukan reservasi
2.	Pembayaran Uang Muka	Merupakan proses pembayaran uang muka yang dilakukan oleh pengunjung dan dikelola oleh bagian front office untuk melakukan pembayaran uang muka pemesanan Cottage
3.	Check in	Merupakan proses yang dilakukan oleh bagian Front Office untuk memeriksa dan menginput data pengunjung yang melakukan reservasi.
4.	Check out	Merupakan proses yang dilakukan oleh bagian Front Office untuk menginput data

		pembayaran setelah pengunjung menginap di cottage.
5.	Laundry	Merupakan proses yang dilakukan pihak laundry untuk mencatat data laundry.
6.	Laporan	Merupakan proses yang dilakukan oleh pihak manager untuk mengelola dan melihat data-data pemasukan yang sudah didapatkan cottage.
7.	Data Cottage	Merupakan proses yang dilakukan oleh pihak manager untuk menginputkan dan mengelola data-data terkait cottage.

Tabel 4.3

Skenario *Use Case* Reservasi yang di Usulkan

Identifikasi	
Nama *Use Case*	Reservasi
Tujuan	Memudahkan pengunjung untuk melakukan reservasi
Deskripsi	
Aktor	Pengunjung
Kondisi Awal	menampilkan halaman *Home*
Skenario Utama	
Aktor	Reaksi Sistem

1. Pengunjung melakukan reservasi	
2. Memilih tanggal *check in*dan *check out*	
	3. Menampilkan daftar *cottage* yang tersedia
4. Memilih jenis *cottage* yang ingin dipesan	
	5. Menampilkan *form* data diri terkait nama, *email* dan nomer telepon pengunjung
6. Memasukan data pemesanan terkait nama, *email,* dan no telepon dan memilih jumlah *cottage* yang akan dipesan	
	7. Sistem akan menampilkan popup

	pemberitahuan bahwa pengunjung harus melakukan pembayaran *DP*
	8. Sistem menampilkan pemberitahuan bahwa *email* telah dikirimkan ke *email* calon pengunjung
	9. Sistem mengirimkan email ke akun email aktor
	10. Sistem menampilkan pemberitahuan bahwa proses reservasi berhasil
	11. Sistem mencetak bukti reservasi yang telah dipesan aktor
12. Menerima bukti reservasi	

Tabel 4.4

Skenario *Use Case* Pembayaran Uang Muka

Identifikasi	
Nama *Use Case*	Pembayaran Uang Muka
Tujuan	Memudahkan pengunjung mengonfirmasi pembayaran uang muka
Deskripsi	

Aktor	Pengunjung
Kondisi Awal	Menampilkan menu pembayaran
Skenario Utama	
Aktor	Reaksi Sistem
1. Pengunjung telah mendapatkan nomer reservasi via email	
2. Pengunjung memilih menu form pembayaran uang muka	
	3. Menampilkan form pembayaran uang muka
4. Pengunjung memasukan nomor reservasi yang sebelumnya sudah dikirimkan melalui email	
5. Pengunjung memasukan data pembayaran ke	

form pembayaran uang muka	
6. Pengunjung mengupload bukti transferan	

Tabel 4.4

Skenario *Use Case* Pembayaran Uang Muka Lanjutan

	7. sistem mengupload bukti pembayaran
8. pengunjung menekan tombol *submit* apabila telah melengkapi form pembayaran serta mengupload bukti pembayaran	
	9. sistem menampilkan pemberitahuan bahwa telah berhasil melakukan *upload* bukti pembayaran
	10. sistem mencetak bukti pembayaran

Tabel 4.5

Use Case *Check In* yang Diusulkan

Identifikasi	
Nama *Use Case*	*Check In*
Tujuan	Memudahkan bagian front office dalam menginput data pengunjung yang melakukan reservasi *cottage*.
Deskripsi	
Aktor	*Front Office*
Kondisi Awal	Menampilkan menu *Check In*
Skenario Utama	
Aktor	Reaksi Sistem
1. Mengecek Nomor reservasi di menu Uang Muka	
	2. Sitem menampilkan bukti bahwa telah melakukan pembayaran
3. Memvalidasi	

dengan menekan aksi centang	
	4. Sistem menampilkan kelengkapan bukti transfer uang muka
5. Menyimpan bukti transfer	
	6. Sistem menampilkan pemberitahuan bahwa data telah di validasi
	7. Mencetak bukti pembayaran uang muka
8. Melakukan cek nomor reservasi di menu *check in* reservasi	
9. Memasukan nomor reservasi	
	10. Sistem menampilkan *form* data *check in*
11. Mengisi data *check in*	
12. Menekan aksi	

simpan apabila semua data telah di isi	
	13. Sistem menyimpan data *check in*
	14. Sistem menampilkan bahwa data tersebut telah disimpan di data *check in*
	15. Sistem mencetak bukti *check in*

Tabel 4.6

Use Case Check Out yang Diusulkan

Identifikasi	
Nama *Use Case*	*Check out*
Tujuan	Memudahkan bagian *front office* dalam menginput laporan data pembayaran dan pengunjung.
Deskripsi	
Aktor	*Front Office*
Kondisi Awal	Menampilkan menu *Check Out*
Skenario Utama	
Aktor	Reaksi Sistem
1. Mengecek Nomor	

check in	
	2. Sitem menampilkan data nomor *check in*
3. Memilih nomor *check in* sesuai dengan data check in pengunjung	
	4. Sistem menampilkan data keseluruhan dan berapa yang harus dibayarkan oleh pengunjung
5. Melakukan simpan data sebagai tanda data *check out* dengan menekan aksi simpan	
	6. Sistem menyimpan data *check out*
	7. Sistem menampilkan pemberitahuan bahwa data tersebut berhasil disimpan di data *check out*

	8. Sistem mencetak tanda bukti pembayaran yang harus dibayarkan.
	9. Sistem menampilkan status *check out* pengunjung

Tabel 4.7

***Use Case Laundry* yang Diusulkan**

Identifikasi	
Nama *Use Case*	*Laundry*
Tujuan	Memudahkan bagian laundry dalam menginput data laundry.
Deskripsi	
Aktor	*Laundry*
Kondisi Awal	Menampilkan menu *Laundry*
Skenario Utama	
Aktor	Reaksi Sistem
1. Mengecek Nomor *check in.*	
	2. Sistem menampilkan nomer *check in.*
3. Memilih nomor *cottage* yang ingin melakukan	

laundry.	
4. Mengisi data *laundry* pelanggan.	
5. Menyimpan data *laundry*.	
	6. Sistem menyimpan data *laundry*.
	7. Sistem menampilkan pemberitahuan bahwa data tersebut berhasil disimpan di data *laundry*.
	8. Sistem mencetak tanda bukti *laundry*.
	9. Sistem menampilkan status selesai atau belumnya *laundry*.

Tabel 4.8

***Use Case* Laporan yang Diusulkan**

Identifikasi	
Nama *Use Case*	Laporan
Tujuan	Memudahkan pihak manager dalam mengelola laporan pemasukan dan

	laporan data reservasi.
Deskripsi	
Aktor	*Manager*
Kondisi Awal	Menampilkan menu Laporan
Skenario Utama	
Aktor	Reaksi Sistem
1. Mengecek laporan transaksi	
	2. Sistem menampilkan menu periode transaksi
3. Memasukan periode tanggal transaksi	
	4. Sistem menampilkan laporan pemasukan
5. Mengecek laporan data reservasi	
6. Memasukan periode tanggal reservasi	
	7. Sistem menampilkan laporan data reservasi

Tabel 4.9

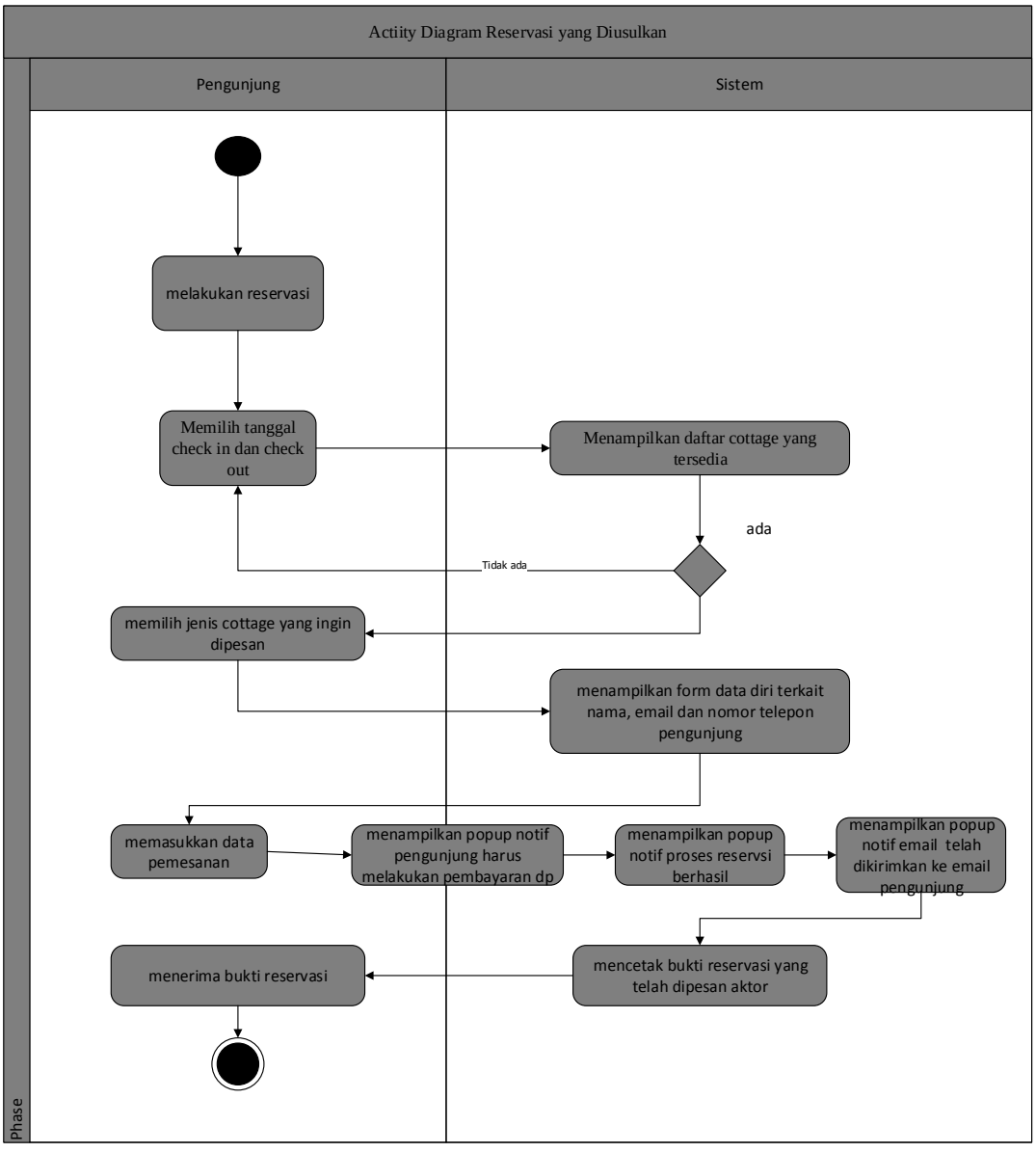
***Use Case* Data *Cottage* yang diusulkan**

Identifikasi	
Nama *Use Case*	Data *Cottage*
Tujuan	Memudahkan pihak *manager* dalam mengelola data *cottage* dan detail *cottage*.
Deskripsi	
Aktor	*Manager*
Kondisi Awal	Menampilkan menu Data *Cottage*
Skenario Utama	
Aktor	Reaksi Sistem
1. Menambahkan Data *Cottage*	
	2. Sistem menampilkan menu Data *Cottage*
3. Mengisidata-data *cottage*	

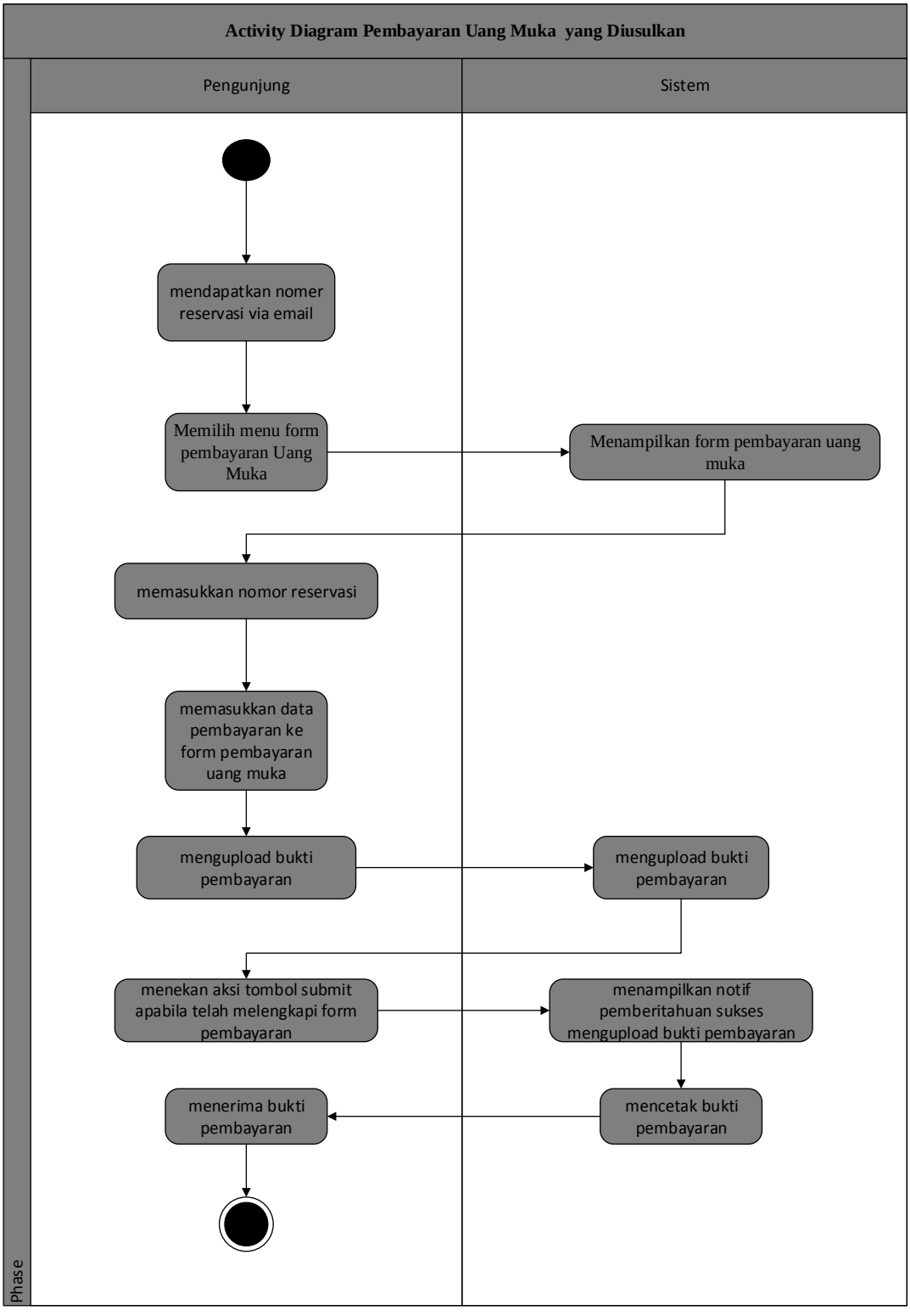
4. Menyimpan, meng*edit* data *cottage*	
	5. Sistem menampilkan pemberitahuan apabila data sudah disimpan, di edit maupun di hapus
	6. Sistem menyimpan data yg sudah di inputkan baik data baru ataupun data yang di*edit*
7. Melakukan hapus data *cottage* yang di inginkan	
	8. Sistem menghapus data

4.1.3.2Activity Diagram

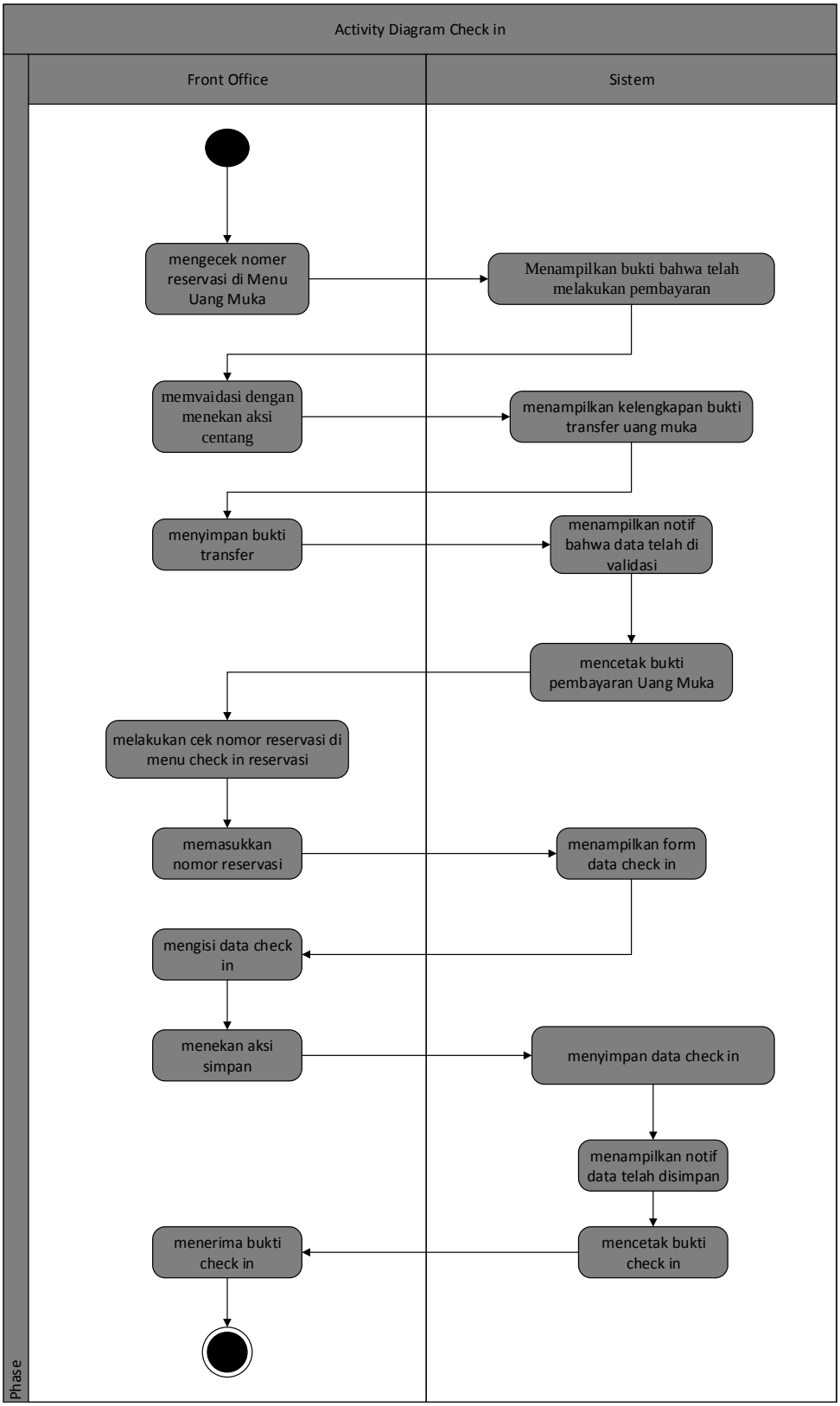
Diagram Aktivitas adalah cara lain untuk memodelkan aliran kejadian. Ada beberapa diagram aktivitas yang diusulkan yaitu sebagai berikut:



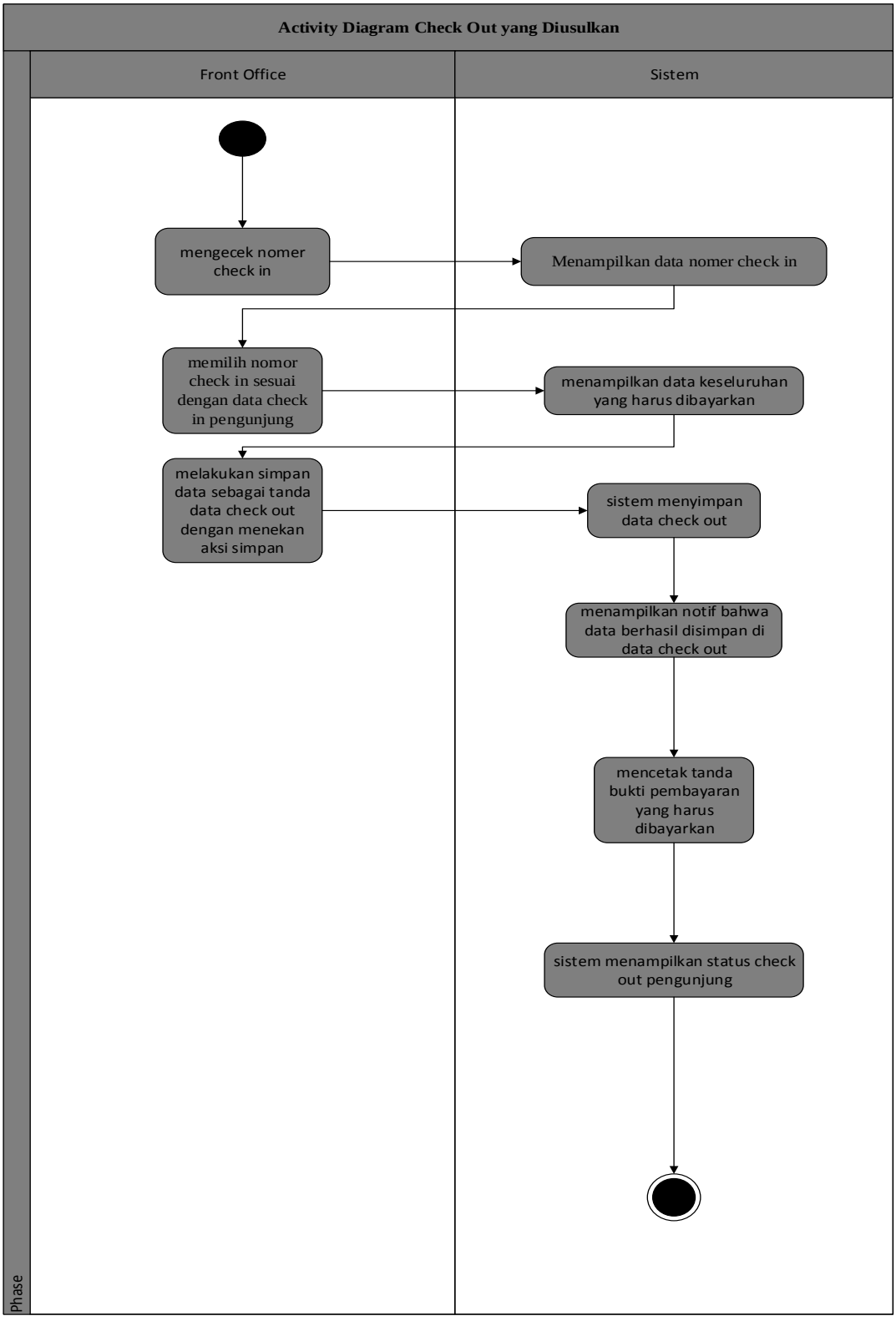
Gambar 4.2*ActivityDiagram* Reservasi



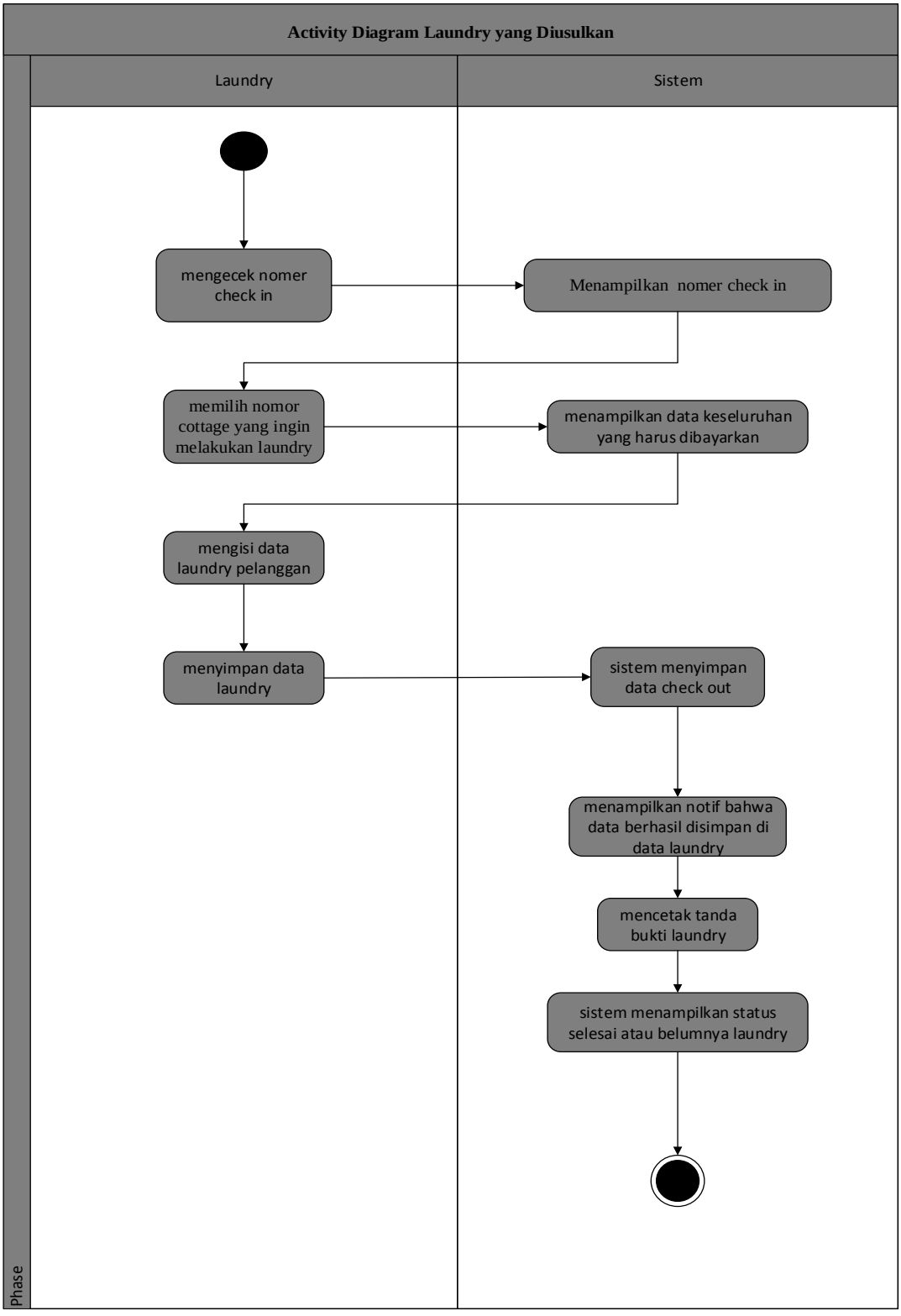
Gambar 4.3*Activity Diagram* Pembayaran Uang Muka



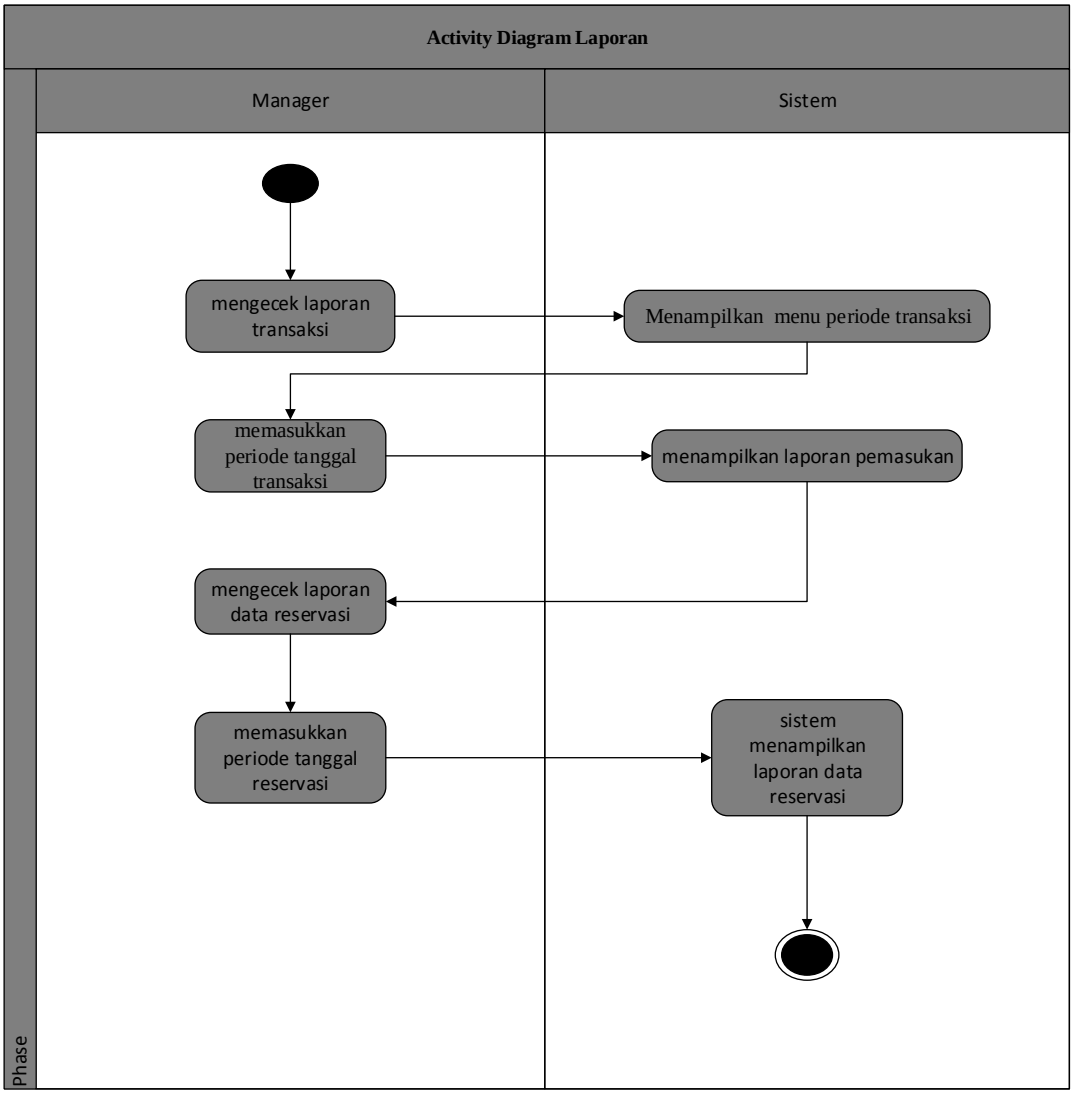
Gambar 4.4*Activity Diagram Check In*



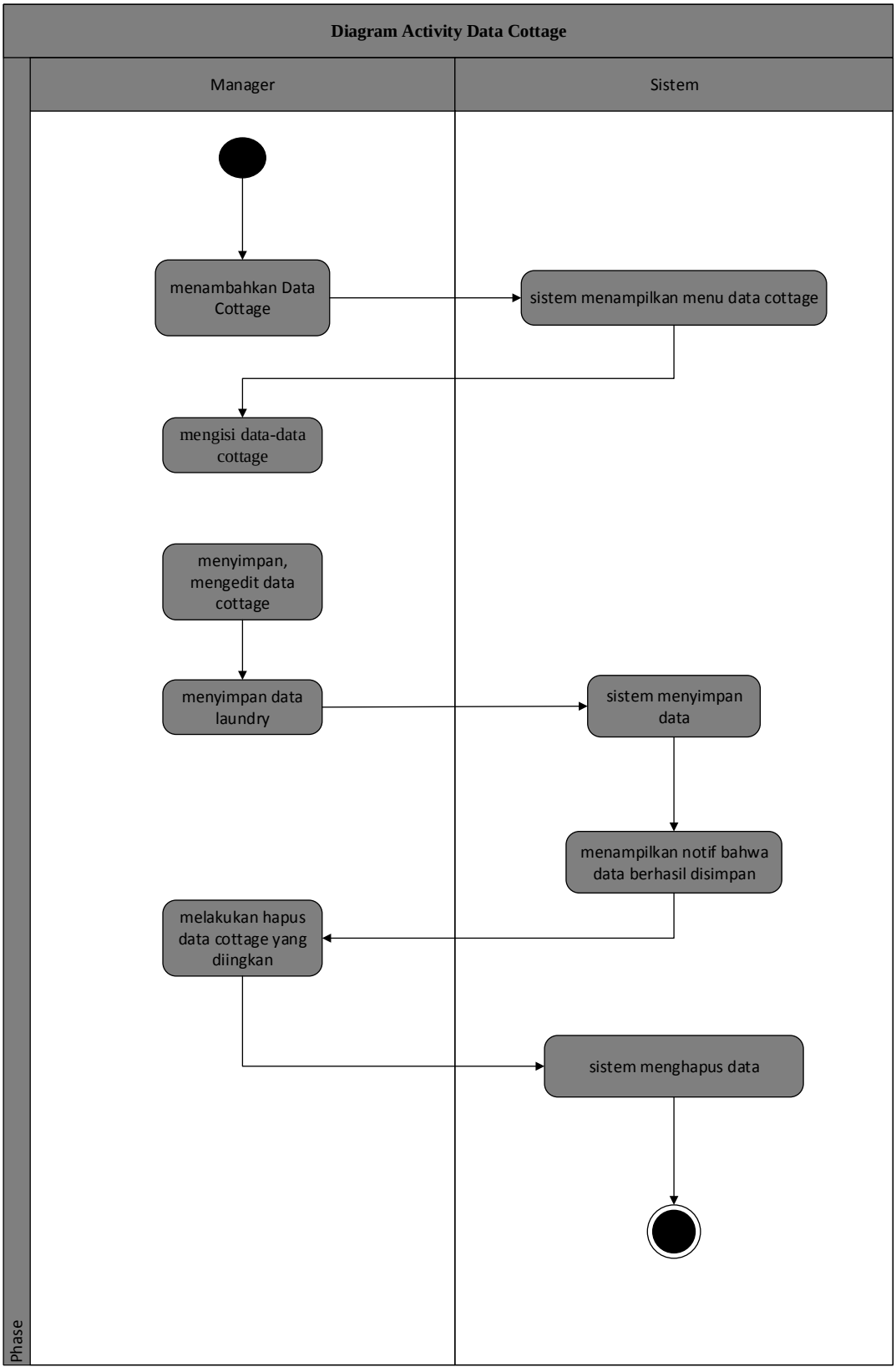
Gambar 4.5*Activity DiagramCheck Out*



Gambar 4.6*Activity DiagramLaundry* Yang Diusulkan



Gambar 4.7*Activity Diagram* Laporan Yang Diusulkan

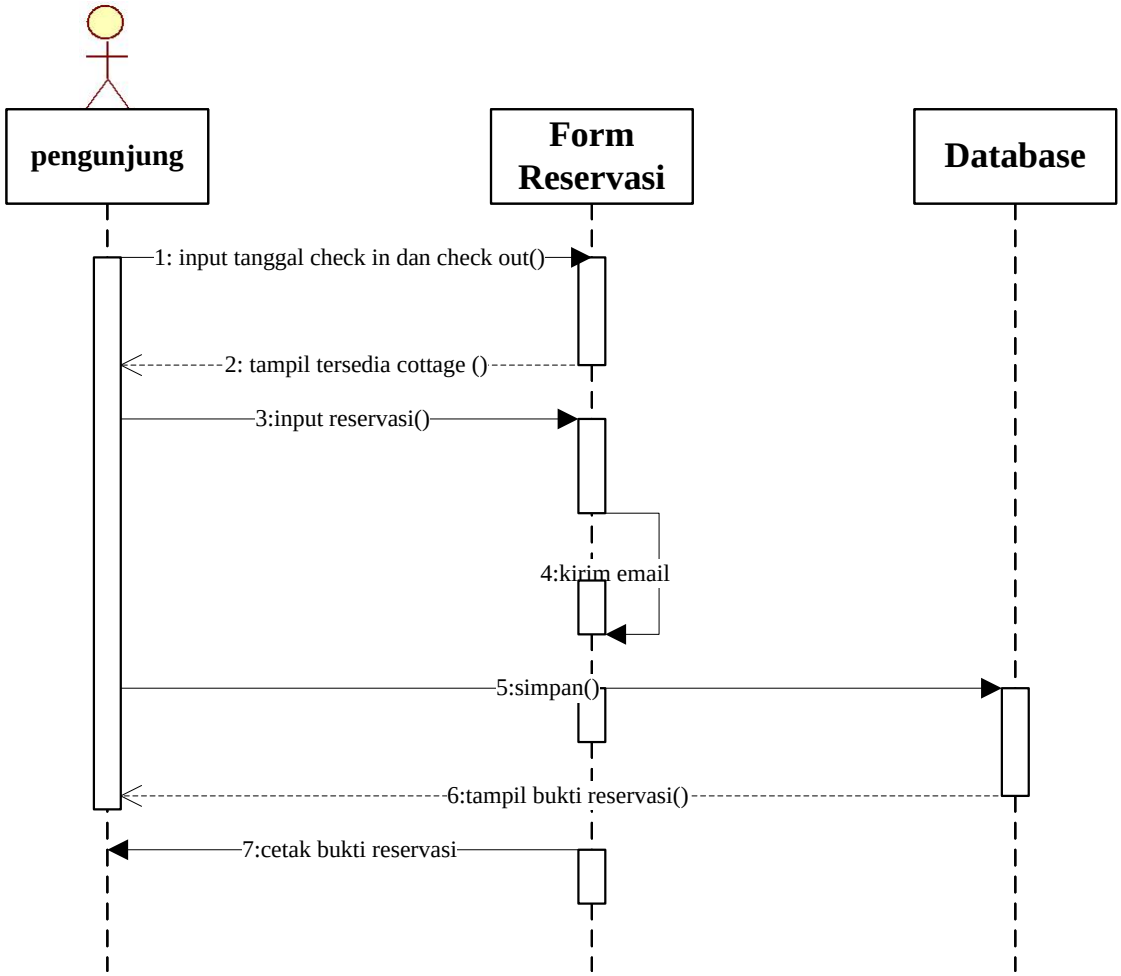

Diagram Activity Data Cottage
Manager
Sistem
menambahkan Data Cottage
sistem menampilkan menu data cottage
mengisi data-data cottage
menyimpan, mengedit data cottage
menyimpan data laundry
sistem menyimpan data
menampilkan notif bahwa data berhasil disimpan
melakukan hapus data cottage yang diingkan
sistem menghapus data
Phase

Gambar 4.8*ActivityDiagram* Data *Cottage* Yang Diusulkan

4.1.3.3 Sequence Diagram

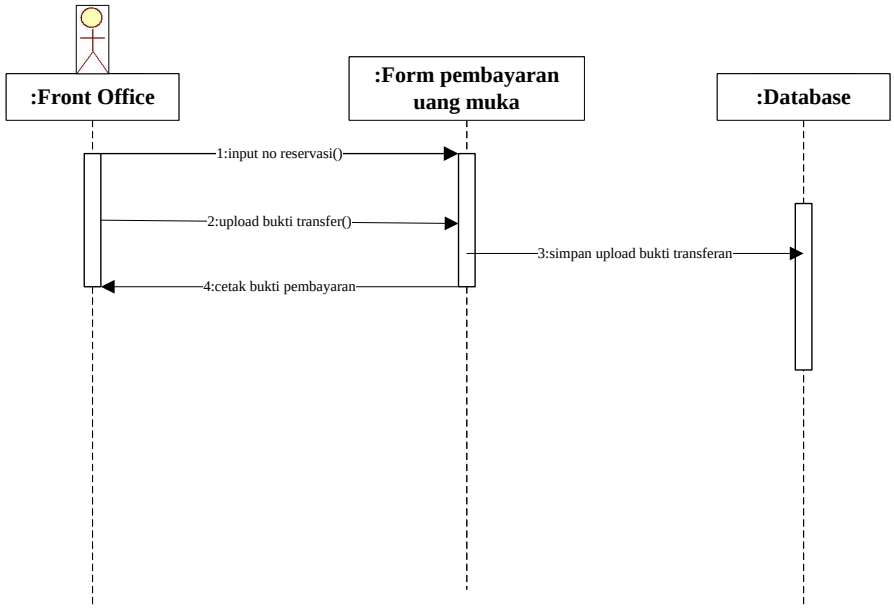
Diagram Sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Berikut adalah sekuen diagram yang diusulkan pada sistem reservasi *cottage*, yaitu:

1. *Sequence Diagram* Reservasi



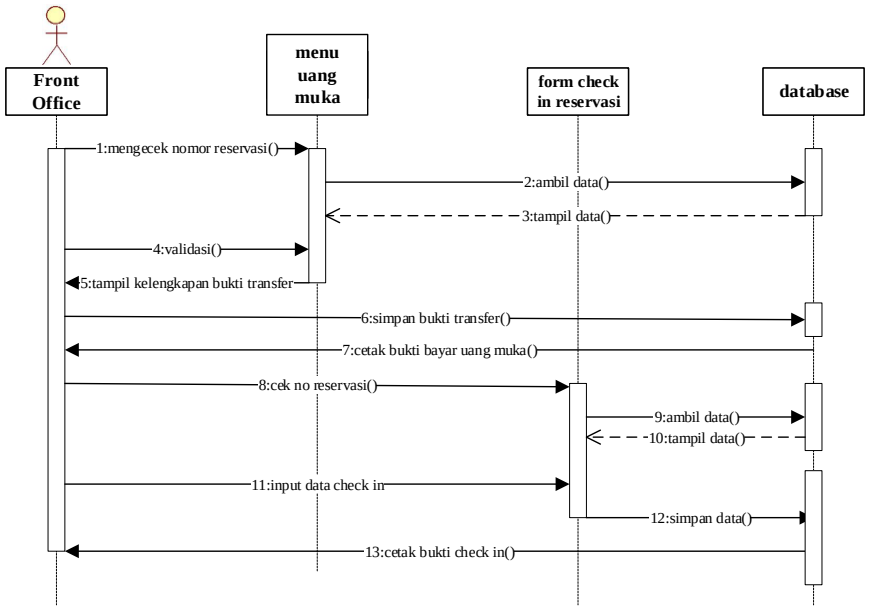
Gambar 4.9*Sequence Diagram* Reservasi Yang Diusulkan

2. *Sequence Diagram* Pembayaran Uang Muka



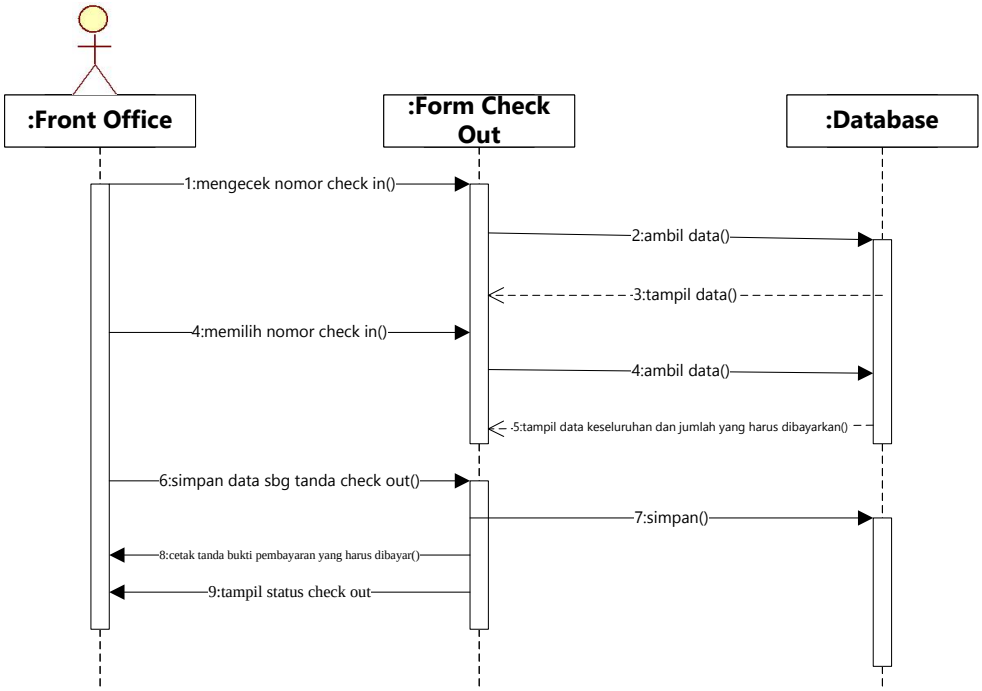
Gambar 4.10 *Sequence Diagram* Pembayaran Uang Muka

3. Sequence Diagram Check In



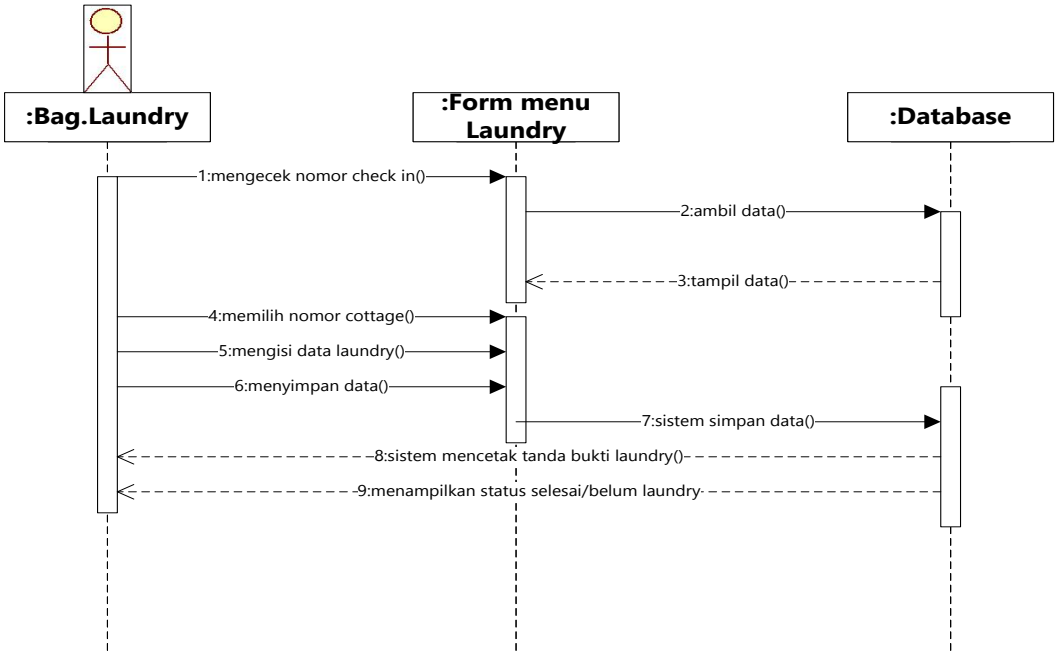
Gambar 4.11*Sequence Diagram Check In*

4. Sequence Diagram Check out



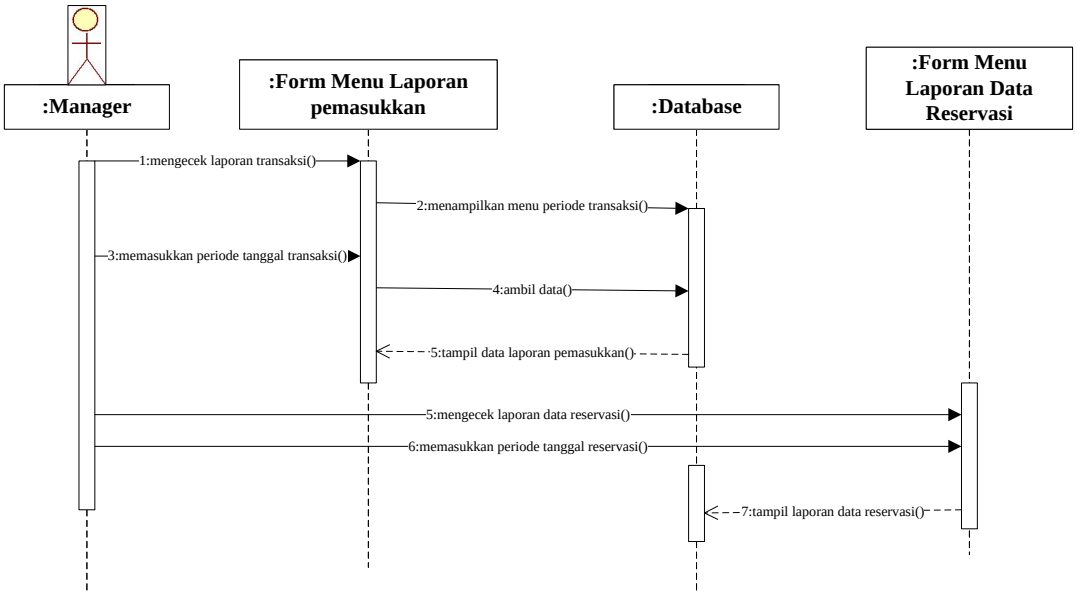
Gambar 4.11*Sequence Diagram Check Out*

5. Sequence Diagram Laundry



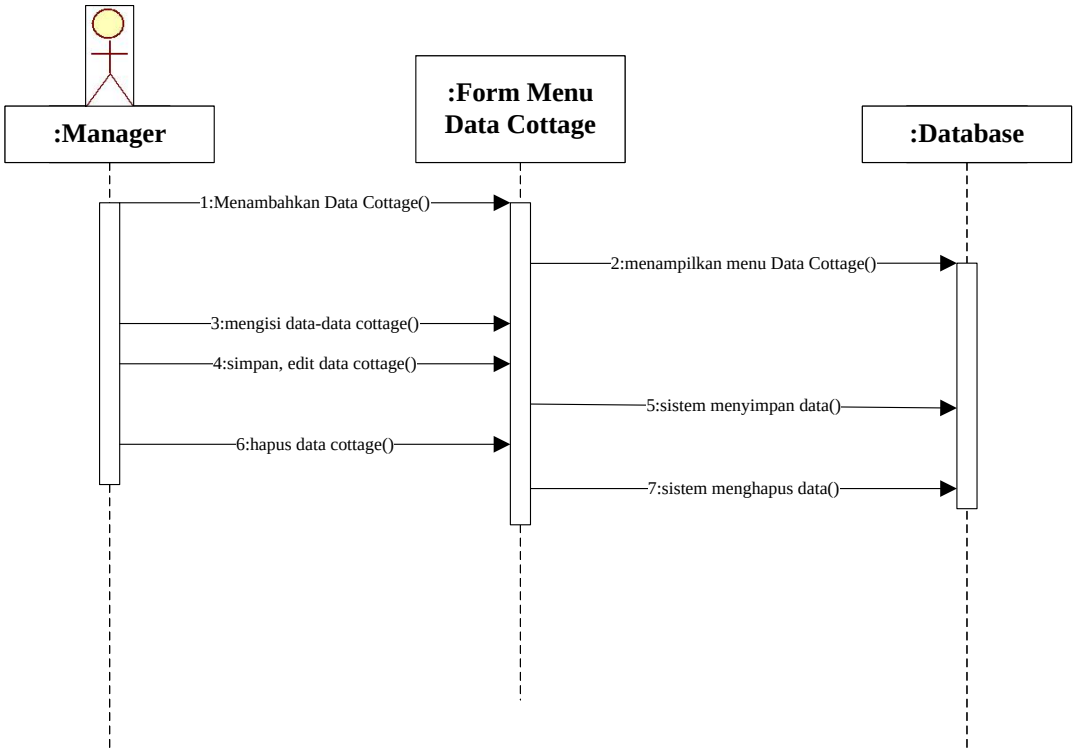
Gambar 4.13*Sequence Diagram Laundry*

6. Sequence Diagram Laporan



Gambar 4.14*Sequence Diagram* Laporan

7. *Sequence Diagram* Data *Cottage*

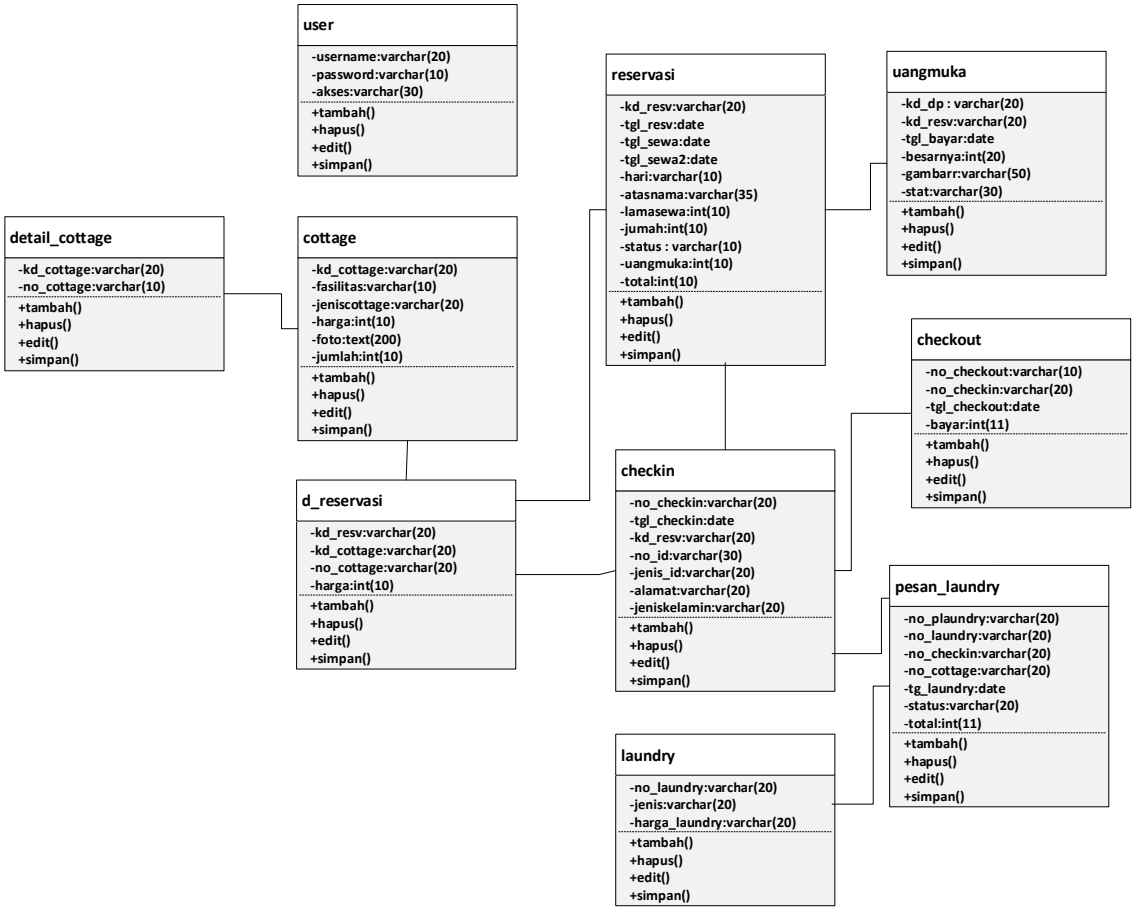


Gambar 4.15*Sequence Diagram* Data *Cottage*

4.1.4Perancangan Data

4.1.4.1 *Class Diagram*

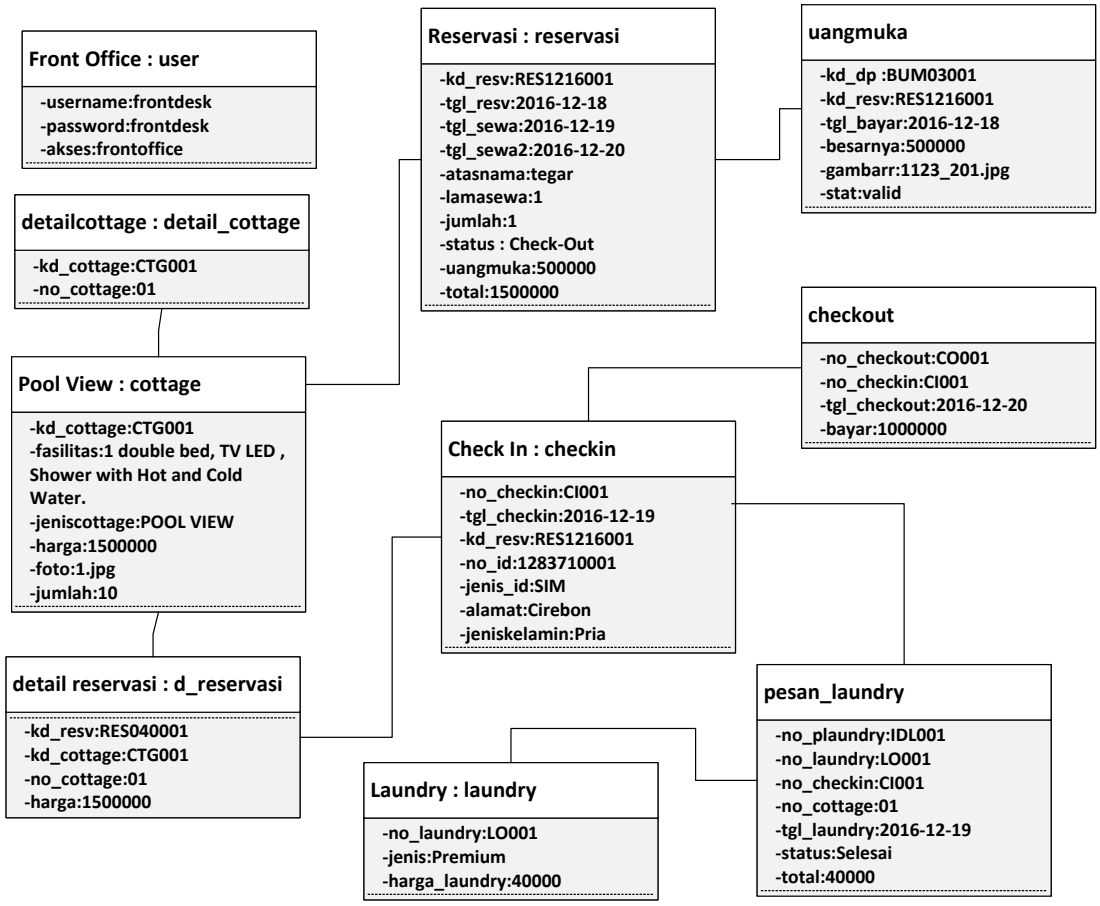
Pemodelan kelas menunjukkan kelas-kelas yang ada di sistem dan hubungan antar kelas-kelas itu atribut-atribut dan operasi-operasi di kelas-kelas. Diagram kelas menunjukkan aspek *static* sistem terutama untuk mendukung kebutuhan fungsional sistem. Berikut *class diagram* yang diusulkan:



Gambar 4.16*Class Diagram* Yang diusulkan

4.1.4.2 *Object Diagram*

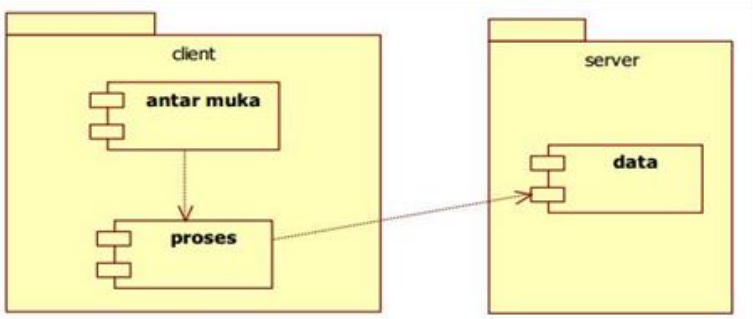
Berikut adalah diagram komponen dari Sistem Informasi reservasi:



Gambar 4.17*Object Diagram* Yang Diusulkan

4.1.4.3 *Component Diagram*

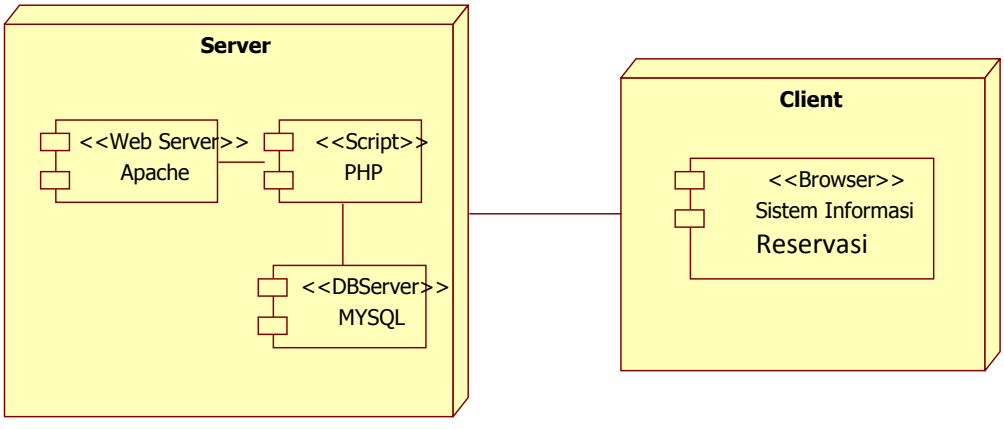
Diagram komponen atau *component diagram* dibuat untuk menunjukkan organisasi dan ketergantungan di antara kumpulan komponen dalam sebuah sistem. Diagram komponen fokus pada komponen sistem yang dibutuhkan dan ada di dalam sistem. Berikut adalah diagram komponen dari Sistem Informasi reservasi:



Gambar 4.18 *Component Diagram*

4.1.4.4*Deployment Diagram*

Deployment diagram menunjukkan tata letak sebuah sistem secara fisik, menampakkan bagian-bagian *software* yang berjalan pada bagian-bagian *hardware* yang digunakan untuk mengimplementasikan sebuah sistem dan keterhubungan antara komponen-komponen *hardware* tersebut. *Deployment diagram* dapat digunakan pada bagian-bagian awal proses perancangan sistem untuk mendokumentasikan arsitektur fisik sebuah sistem. Berikut *Deployment diagram* yang menggambarkan susunan fisik dan perangkat lunak dalam sistem:



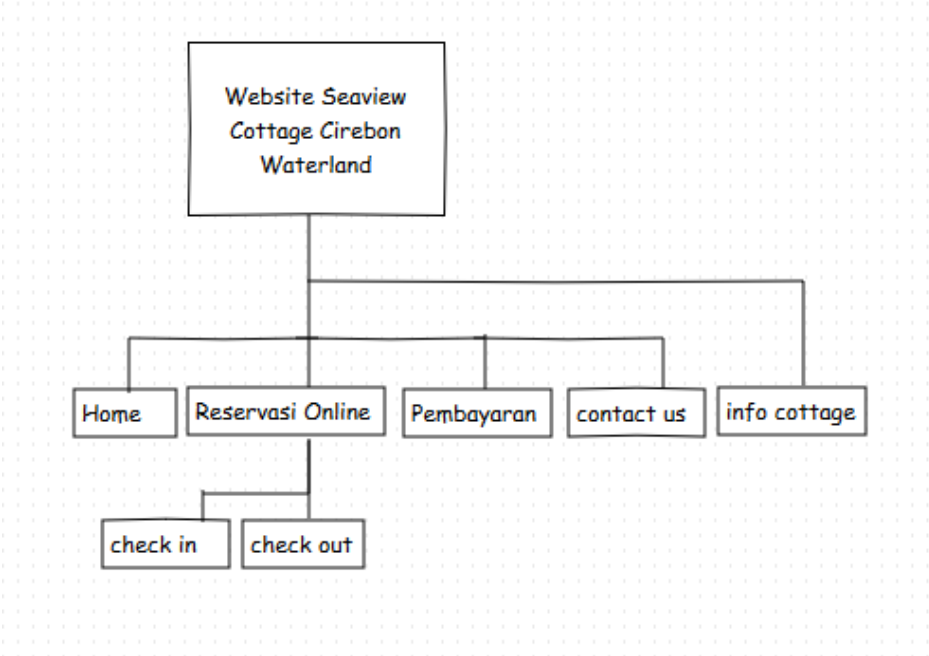
Gambar 4.19*Deployment Diagram*

4.2 Perancangan Antar Muka

Perancangan antar muka merupakan penggambaran tampilan yang digunakan secara langsung oleh pengguna, interaksi yang dapat dilakukan oleh pengguna dalam sistem. Adapun dalam antar muka ini terdapat beberapa bagian yang harus dilakukan, yaitu dari mulai menentukan struktur menu yang ada dalam program, tampilan input dan output pada setiap fungsi yang telah ditentukan, diuraikan sebagai berikut

4.2.1 Struktur Menu

Struktur menu adalah bentuk umum dari suatu rancangan program untuk memudahkan pemakai dalam menjalankan program komputer sehingga pada saat menjalankan program komputer, *user* tidak mengalami kesulitan dalam memilih menu-menu yang diinginkan.



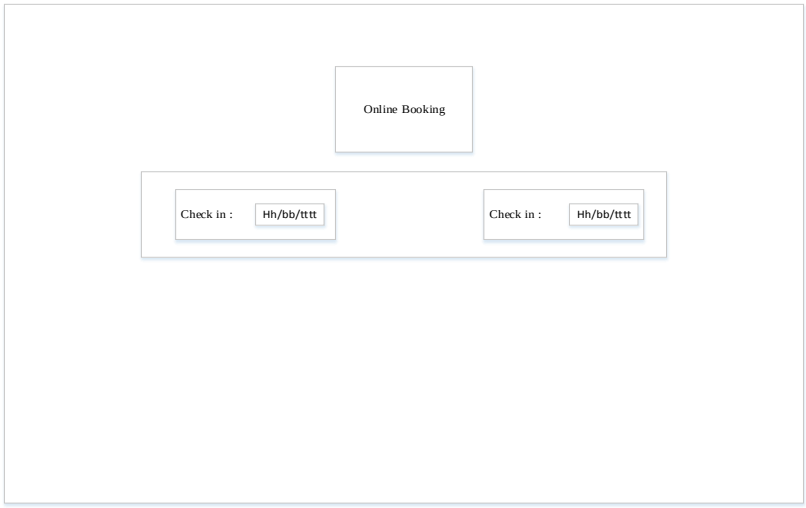
Gambar 4.20 Struktur Menu Sistem

4.2.2 Perancangan *Input*

Perancangan *input* yaitu sebuah tampilan yang dirancang sebagai tempat untuk memasukan data-data yang diberikan oleh pengguna sistem *(user)*, yang merupakan sumber untuk proses pengolahan yang akan menghasilkan sebuah informasi yang berguna bagi pengguna. Pengguna atau user dalam aplikasi dibagi ke dalam 4 tipe yaitu Pengunjung, *Front Office,Laundry*, *Manager*. Oleh karena itu berikut rancangan antar muka input secara umum berdasarkan tipe *user*.

Berikut perancangan input yang ada pada sistem yaitu sebagai berikut:

1. *Form* Reservasi



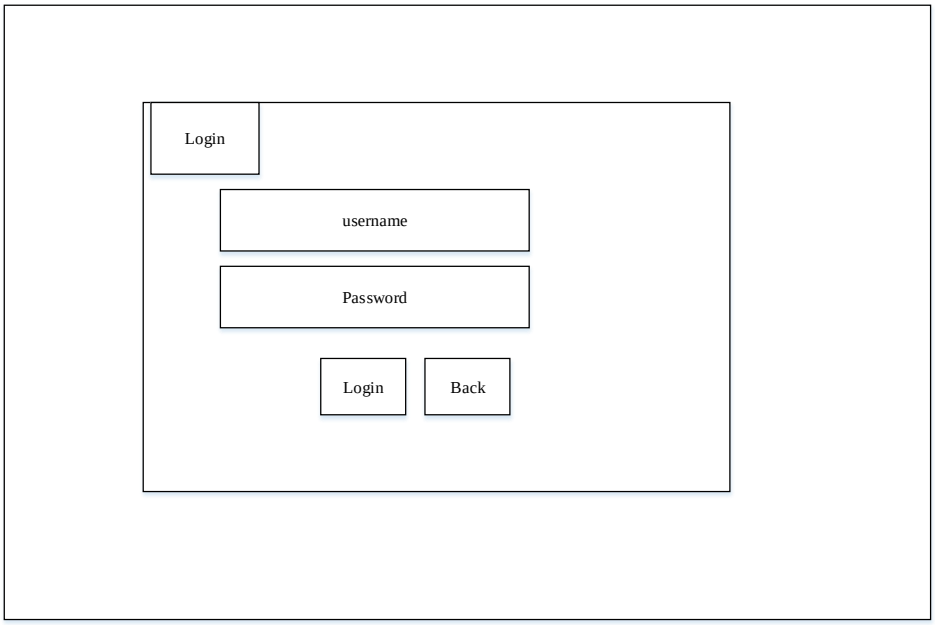
Gambar 4.21 Form Reservasi

2. *Form* menu utama



Gambar 4.22 *Form* menu Utama

3. Form user

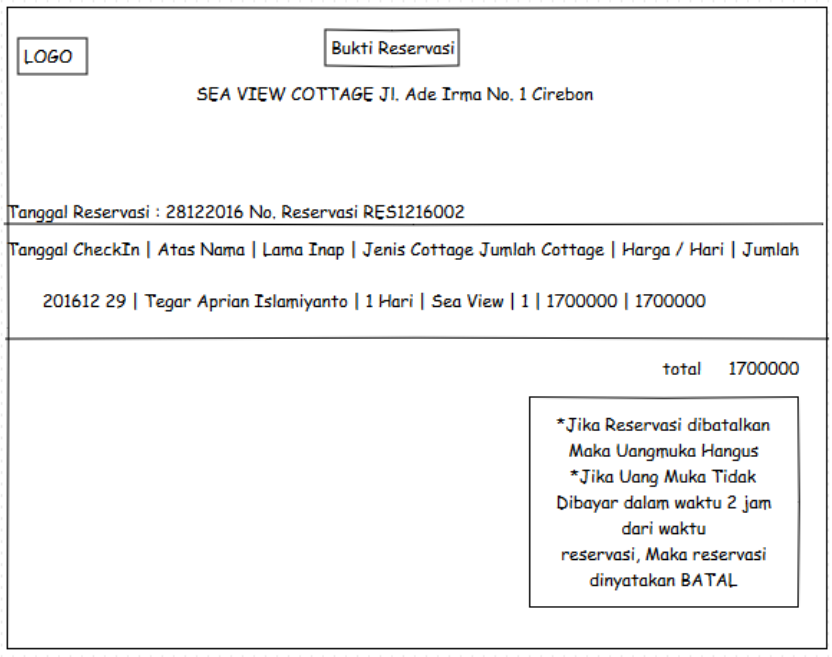


Gambar 4.23 Form User

4.2.3 Perancangan *Output*

Rancangan *output* adalah hasil dari proses yang dilakukan oleh sistem, hasil dari inputan yang nantinya akan di tampilkan pada keluaran dalam bentuk informasi.

a. Perancangan Output Bukti Reservasi



Gambar 4.24 Perancangan *Output* Bukti Reservasi

b. Perancangan*Output* Pembayaran Uang Muka

Bukti Pembayaran Uang Muka

CIREBON WATER LAND

No. BUM02001	**Tanggal Transaksi 2016-12-28**
Telah Terima Dari	**: Tegar Aprian Islamiyanto**
Pembayaran	**: Uang Muka Reservasi dengan No. Reservasi Cottage RES1216003**
Uang Sejumlah	**: Rp.500000,-**
Terbilang	**: Lima Ratus Ribu Rupiah**

***harap dibawa pada saat hari sewa cottage**

Gambar 4.25 Perancangan *output* pembayaran Uang Muka

c. Perancangan *Output* Bukti *Check In*

LOGO

Bukti Check In

SEA VIEW COTTAGE Jl. Ade Irma No. 1 Cirebon

Tanggal Reservasi : 28122016 No. Reservasi RES1216002

Tanggal CheckOut | Atas Nama | Lama Sewa | Jenis Cottage | Harga / Hari | Jumlah

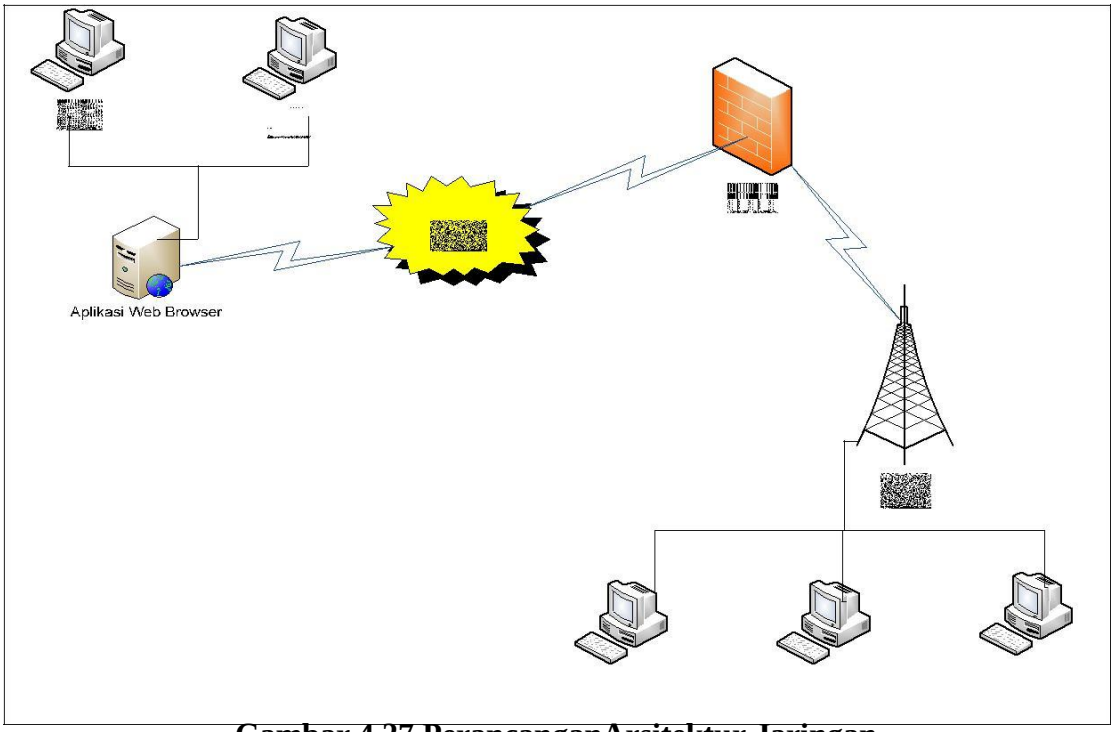
total

*Tunjukkan pada bagian Laundry apabila ingin melakukan Laundry

Gambar 4.26 Perancangan *Output* Bukti *Check In*

4.3 Perancangan Arsitektur Jaringan

Arsitektur jaringan bisa dikatakan sebagai gambaran secara fisik dari pola hubungan antara komponen-komponen jaringan, yang meliputi *server, workstation, hub* dan pengkabelannnya. Adapun Rancangan Arsitektur jaringan yaitu sebagai berikut:



Gambar 4.27 PerancanganArsitektur Jaringan

Arsitektur jaringan ini berskala WAN (*Wide Area Network*). Admin dan *server* mengirimkan informasi berupa aplikasi *web* ke internet, dan untuk para *user* terhubung pada sebuah media penghubung yang dikenal sebagai modem, dimana modem tersebut biasanya berupa lewat saluran telepon yang telah terkoneksi internet.

4.4 Pengujian

Adapun kriteria pengujian tersebut dimaksudkan untuk mengetahui kebenaran perangkat lunak dan kesesuaian dengan spesifikasinya dinyatakan berhasil jika aplikasi mampu menjalankan fungsi – fungsi berdasar kan spesifikasi

secara benar. Sebaliknya pengujian dinyatakan gagal jika terdapat fungsi dari perangkat lunak dalam spesifikasi yang tidak dapat dijalankan pada proses pengujian.

4.4.1 Rencana Pengujian

Metode pengujian yang digunakan pada perangkat lunak ini adalah metode pengujian *black box.* Rencana pengujiannya adalah sebagai berikut:

Tabel 4.9 Rencana Pengujian

Kelas Uji	Tingkat Uji
Login	*Username*
	Password
Input Data	Data *User Admin*
	Data *Cottage*
	Data *Laundry*
Proses	Transaksi *Check In*
	Transaksi Pembayaran
	Transaksi *Check Out*
	Transaksi *Laundry*
Output	Bukti Transaksi, Laporan-laporan

4.4.2 Kasus dan Hasil Pengujian

Tahap selanjutnya adalah kasus dan hasil pengujian, adalah sebagai berikut:

1. Pengujian *Login User*

   Pengujian ini dilakukan oleh petugas (yang terdiri dari *front office, manager*, dan *laundry*) dengan hak akses masing-masing.

Tabel 4.10 Pengujian *Login*

Kasus dan Hasil Pengujian Data Normal			
Data Masukan	Hasil yang Diharapkan	Pengamatan	Kesimpulan
Username: frontdesk *Password: frontdesk*	*Login* berhasil, hak akses sesuai menu *admin frontdesk*	Data *valid* dan bisa masuk ke sistem	[√] diterima [ ] ditolak
Username: manager *Password: manager*	*Login* berhasil, hak akses sesuai menu *admin manager*	Data *valid* dan bisa masuk ke sistem	[√] diterima [ ] ditolak
Username: laundry *Password: laundry*	*Login* berhasil, hak akses sesuai menu *admin laundry*	Data valid dan bisa masuk ke sistem	[√] diterima [ ] ditolak
Kasus dan Hasil Pengujian Data Salah			
Data Masukan	Hasil yang Diharapkan	Pengamatan	Kesimpulan
Username: frontdesk *Password: frontoffice*	*Login* gagal, tidak bias masuk ke sistem	Data tidak *valid* dan menampilkan pesan "*Username* dan	[√] diterima [ ] ditolak

		*Password*Anda Salah Atau Tidak Terdaftar”	

2. Pengujian *input* data *cottage*

Pengujian ini dilakukan oleh petugas yang memiliki hak akses sebagai *manager*.

Tabel 4.11 Pengujian *input* data *Cottage*

Kasus dan Hasil Pengujian Data Normal			
Data Masukan	Hasil yang Diharapkan	Pengamatan	Kesimpulan
Pilih menu reservasi	Menampilkan*form*reservasi	Menampilkan *form*reservasi	[√] diterima [ ] ditolak
Masukkan data reservasi dan cek ketersediaan	Jika data yang dimasukkan benar, maka data tersimpan dalam *database*	Data valid dan menampilkan pesan “data ada”	[√] diterima [ ] ditolak
Kasus dan Hasil Pengujian Data Salah			
Data Masukan	Hasil yang Diharapkan	Pengamatan	Kesimpulan
Data pada kolom tidak diisi	Data tidak tersimpan dalam *database*	Data tidak *valid* dan menampilkan pesan “harap isi	[√] diterima [ ] ditolak

		bidang ini"	

3. Pengujian *input* data reservasi

Pengujian ini dilakukan oleh petugas yang memiliki hak akses sebagai *front office.*

Tabel 4.12 Pengujian input data reservasi

Kasus dan Hasil Pengujian Data Normal		
Hasil yang Diharapkan	Pengamatan	Kesimpulan
Menampilkan *form* data *cottage*	Menampilkan *form* data *cottage*	[√] diterima [ ] ditolak
Jika data yang dimasukkan benar, maka data tersimpan dalam d*atabase*	Data *valid* dan menampilkan pesan "data tersimpan"	[√] diterima [ ] ditolak
Kasus dan Hasil Pengujian Data Salah		
Hasil yang Diharapkan	Pengamatan	Kesimpulan
Data tidak tersimpan dalam d*atabase*	Data tidak *valid* dan menampilkan pesan "harap isi bidang ini"	[√] diterima [ ] ditolak

4. Pengujian *input* data *check in*

Pengujian ini dilakukan oleh petugas yang memiliki hak akses sebagai *front office.*

Tabel 4.13 Pengujian *inputCheck In*

Kasus dan Hasil Pengujian Data Normal		
Hasil yang Diharapkan	Pengamatan	Kesimpulan
Menampilkan *form Check In*	Menampilkan *form Check In*	[√] diterima [ ] ditolak
Jika data yang dimasukkan benar, maka data tersimpan dalam *database*	Data valid dan menampilkan pesan "data tersimpan"	[√] diterima [ ] ditolak
Kasus dan Hasil Pengujian Data Salah		
Hasil yang Diharapkan	Pengamatan	Kesimpulan
Data tidak tersimpan dalam *database*	Data tidak valid dan menampilkan pesan "Data Tidak Ditemukan, periksa kembali nomor reservasi Anda"	[√] diterima [ ] ditolak

5. Pengujian *input check out*

Pengujian ini dilakukan oleh petugas yang memiliki hak akses sebagai *front office.*

Tabel 4.14 Pengujian *inputCheck Out*

Kasus dan Hasil Pengujian Data Normal		
Hasil yang Diharapkan	Pengamatan	Kesimpulan
Menampilkan *form Check Otut*	Menampilkan *form Check Out*	[√] diterima [ ] ditolak
Jika data yang dimasukkan benar, maka data tersimpan dalam *database*	Data valid dan menampilkan pesan "Berhasil menyimpan data *check out"*	[√] diterima [ ] ditolak
Kasus dan Hasil Pengujian Data Salah		
Hasil yang Diharapkan	Pengamatan	Kesimpulan
Data tidak tersimpan dalam *database*	Data tidak valid dan menampilkan pesan "Datagagal disimpan"	[√] diterima [ ] ditolak

6. Pengujian *input* data *Laundry*

Pengujian ini dilakukan oleh petugas yang memiliki hak akses sebagai *laundry*.

Tabel 4.15 Pengujian *input* data *Laundry*

Kasus dan Hasil Pengujian Data Normal		
Hasil yang Diharapkan	Pengamatan	Kesimpulan
Menampilkan *form* bagian *laundry*	Menampilkan *form* bagian *laundry*	[√] diterima [ ] ditolak
Jika data yang dimasukkan benar, maka data	Data valid dan menampilkan pesan "Berhasil	[√] diterima [ ] ditolak

tersimpan dalam *database*	menyimpan data *laundry*"	
Kasus dan Hasil Pengujian Data Salah		
Hasil yang Diharapkan	Pengamatan	Kesimpulan
Data tidak tersimpan dalam *database*	Data tidak *valid* dan menampilkan pesan "harap isi bidang ini"	[√] diterima [ ] ditolak

7. Pengujian *input* data *cottage*

Pengujian ini dilakukan oleh petugas yang memiliki hak akses sebagai *manager*.

Tabel 4.16 Pengujian *input* data *cottage*

Kasus dan Hasil Pengujian Data Normal		
Hasil yang Diharapkan	Pengamatan	Kesimpulan
Menampilkan *form* data *cottage*	Menampilkan *form* menu data *cottage*	[√] diterima [ ] ditolak
Jika data yang dimasukkan benar, maka data tersimpan dalam *database*	Data *valid* dan menampilkan pesan "Berhasil menyimpan data"	[√] diterima [ ] ditolak
Kasus dan Hasil Pengujian Data Salah		
Hasil yang Diharapkan	Pengamatan	Kesimpulan
Data tidak tersimpan dalam *database*	Data tidak *valid* dan menampilkan pesan "harap isi bidang ini"	[√] diterima [ ] ditolak

4.4.3 Kesimpulan Hasil Pengujian

Jika melihat dari hasil pengujian setiap kasus dengan input data-data yang benar dan sesuai maka hasilnya tidak terdapat kesalahan data. Oleh karena itu dapat diambil kesimpulan bahwa dalam proses menginputkan data reservasi, data *check in*, data *check out*, data *laundry* dan data *cottage* pada setiap form berfungsi dengan baik sehingga dapat menghasilkan output sesuai dengan yang diinputkan.

4.5 Implementasi

Implementasi adalah proses penerapan rancangan program yang telah dibuat pada bab sebelumnya atau aplikasi dalam melaksanakan sistem informasi pemrograman yang telah dibuat. Hasil dari tahapan implementasi ini adalah suatu sistem pengolahan data yang sudah dapat berjalan dengan baik. Dengan demikian dapat diketahui apakah perangkat lunah ini dapat menghasilkan sistem informasi reservasi yang sesuai dengan tujuan yang diharapkan. Implementasi pengembangan sistem informasi absensi dan penggajian ini dilakukan dengan menggunakan Bahasa pemrograman PHP. Adapun *database* yang digunakan yang digunakan sebagai media penyimpanan data menggunakan Bahasa *SQL* sebagai basis data dengan MySQL sebagai perangkat lunaknya.

4.5.1 Implementasi perangkat lunak

Berikut merupakan aplikasi, *database*, serta sistem operasi yang diperlukan untuk implementasi perangkat lunak sistem:

1. Sistem Operasi : *Microsoft Windows* 7
2. Editor : *Dreamweaver CS6*
4. Perangkat*Database Server* : XAMPP-3.1.0
5. Database Server : MySQL

4.5.2 Implementasi perangkat keras

Perangkat keras yang dibutuhkan berdasarkan *minimum requirement* yang harus dipenuhi dalam penerapan adalah sebagai berikut:

1. *Memory* 1 GB
2. Processor Intel Core i3 2.20 GHz
3. Intel HD Graphics 3000
4. *Monitor* 1366X768

4.5.3 Implementasi basis data

Implementasi basis data menggunakan MySQL 3.1.0, terdiri dari beberapa tabel yaitu sebagai berikut:

1. *User*

```
  `username` varchar(30) NOT NULL,
  `password` varchar(30) NOT NULL,
  `akses` varchar(30) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

2. *Check in*

```
CREATE TABLE IF NOT EXISTS `checkin` (
no_checkin` varchar(20) NOT NULL,
`tgl_checkin` date NOT NULL,
`kd_resv` varchar(20) NOT NULL,
`no_id` varchar(30) NOT NULL,
 `nama` varchar(40) NOT NULL,
`jenis_id` varchar(10) NOT NULL,
```

```
  `alamat` text NOT NULL,
  `jk` varchar(10) NOT NULL
  ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

3. data *Laundry*

```
CREATE TABLE IF NOT EXISTS `d_laundry` (
  `no_plaundry` varchar(12) NOT NULL,
  `jml_barang` int(12) NOT NULL,
  `jenis` varchar(50) NOT NULL,
  `harga` int(12) NOT NULL,
  `total` int(12) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

4. Reservasi

```
CREATE TABLE IF NOT EXISTS `reservasi` (
  `kd_resv` varchar(20) NOT NULL,
  `tgl_resv` date NOT NULL,
  `tgl_sewa` date NOT NULL,
  `tgl_sewa2` date NOT NULL,
  `atasnama` varchar(35) NOT NULL,
  `lamainap` int(10) NOT NULL,
  `jumlah` int(11) NOT NULL,
  `status` varchar(100) NOT NULL,
  `uangmuka` int(20) NOT NULL,
  `total` int(10) NOT NULL,
  `email` varchar(50) NOT NULL,
  `telp` varchar(20) NOT NULL,
  `jam_limit` varchar(20) NOT NULL,
  `kode_promo` varchar(20) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

5. Room

```
CREATE TABLE IF NOT EXISTS `room` (
  `kd_cottage` varchar(20) NOT NULL,
  `fasilitas` varchar(200) NOT NULL,
  `jenis_cottage` varchar(30) NOT NULL,
  `harga` int(11) NOT NULL,
  `gambar` varchar(200) NOT NULL,
  `jumlah` int(11) NOT NULL,

PRIMARY KEY (`kd_cottage`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

6. Promo

```
CREATE TABLE IF NOT EXISTS `promo` (
  `kode_promo` varchar(20) NOT NULL,
  `judul` varchar(50) NOT NULL,
  `desk` text NOT NULL,
  `status` varchar(20) NOT NULL,
  `besar_diskon` int(20) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

7. Uang Muka

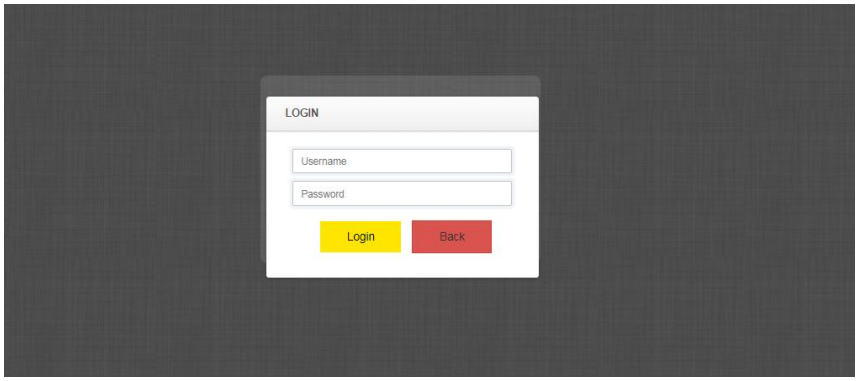
```
CREATE TABLE IF NOT EXISTS `uangmuka` (
  `kd_dp` varchar(20) NOT NULL,
  `kd_resv` varchar(20) NOT NULL,
  `tgl_bayar` date NOT NULL,
  `bank` varchar(5) NOT NULL,
  `nama` varchar(40) NOT NULL,
  `besarnya` int(11) NOT NULL,
  `stat` varchar(20) NOT NULL,
```

```
  `gambar` text NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

4.5.4 Implementasi antar muka

Antar muka dalam sebuah sistem merupakan bagian yang sangat penting, implementasi antar muka aplikasi website yang menggunakan Bahasa pemrograman PHP. Berikut merupakan implementasi antar muka dari halaman yang dibuat:

1. Form *login*



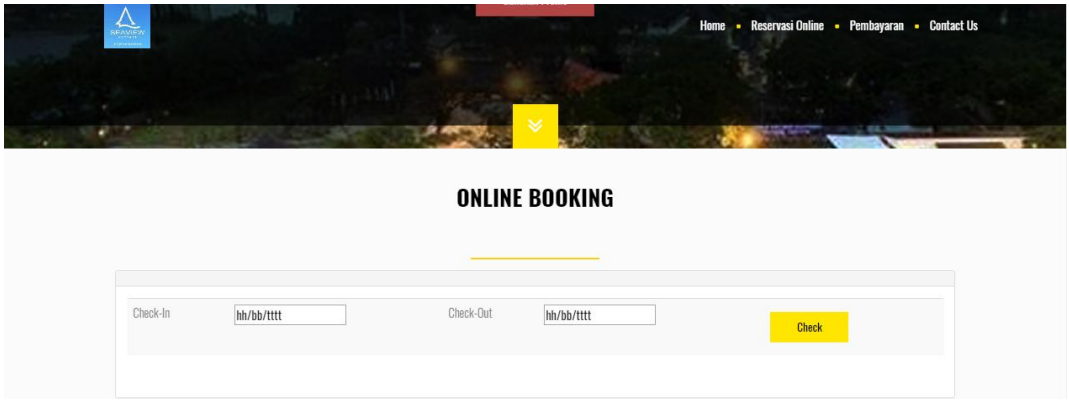
Gambar 4.28 Implementasi *Form login*

2. Menu Utama



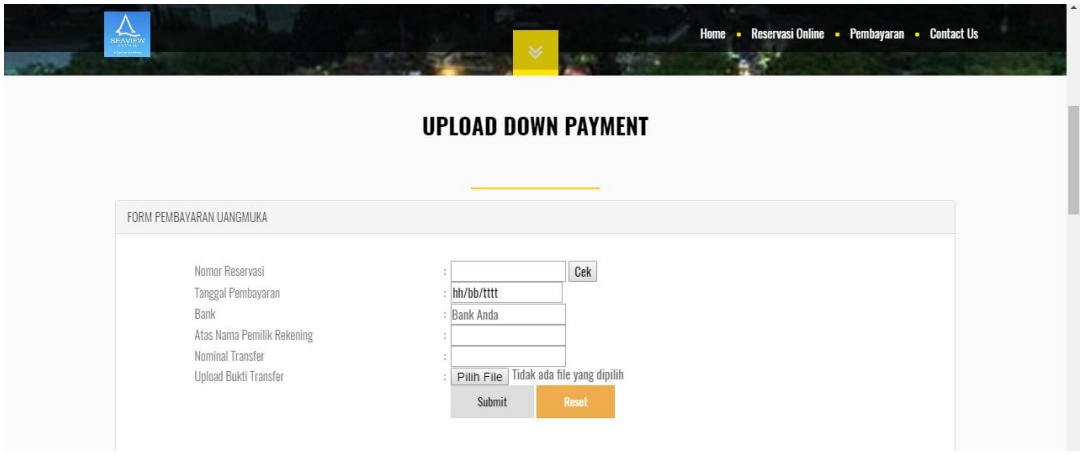
Gambar 4.29 Menu Utama

3. Menu Reservasi Online



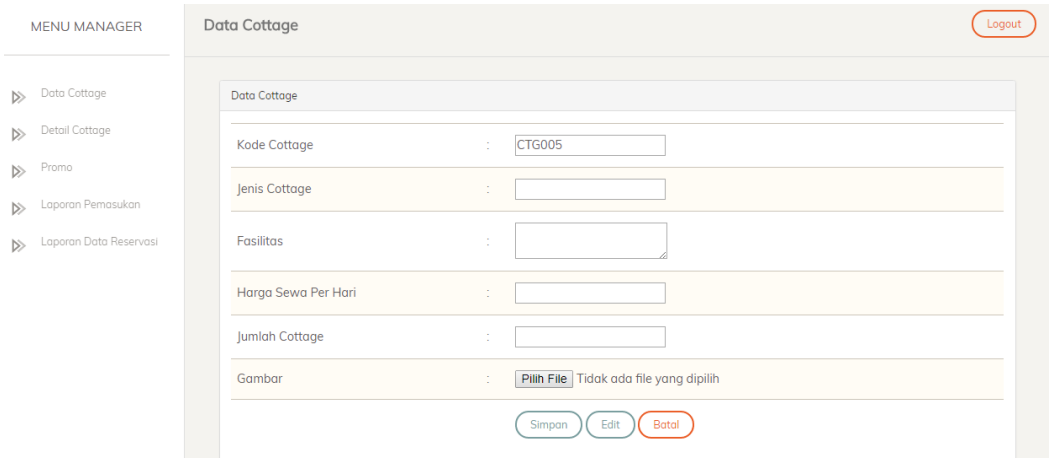
Gambar 4.30 Menu Reservasi Online

4. Form Pembayaran UangMuka



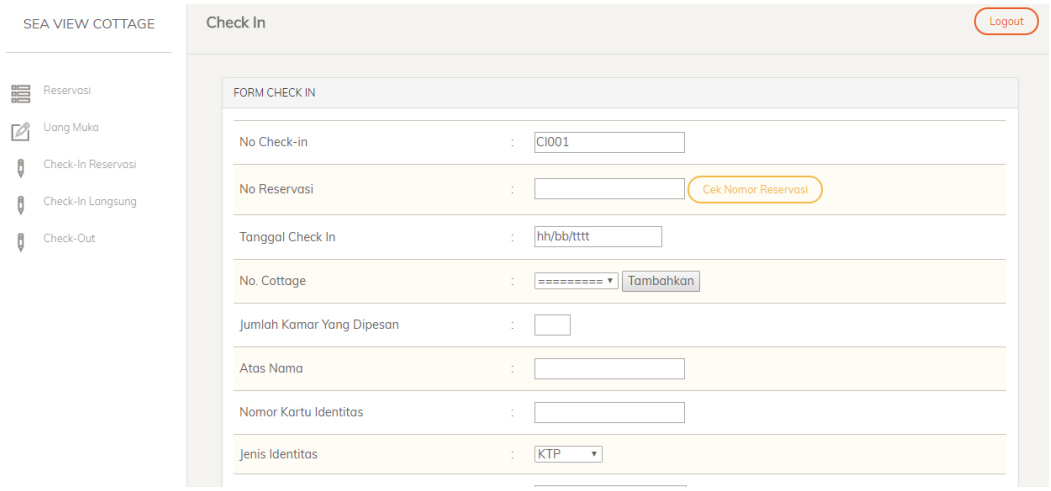
Gambar 4.31 *Form* Pembayaran Uang Muka

5. *Form* Data *Cottage*



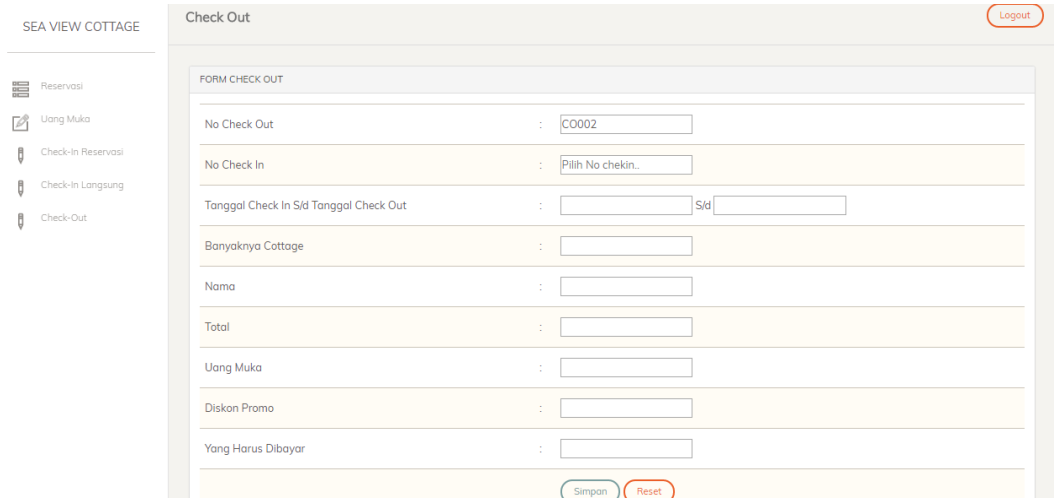
Gambar 4.32 Form Data Cottage

6. Form Check In



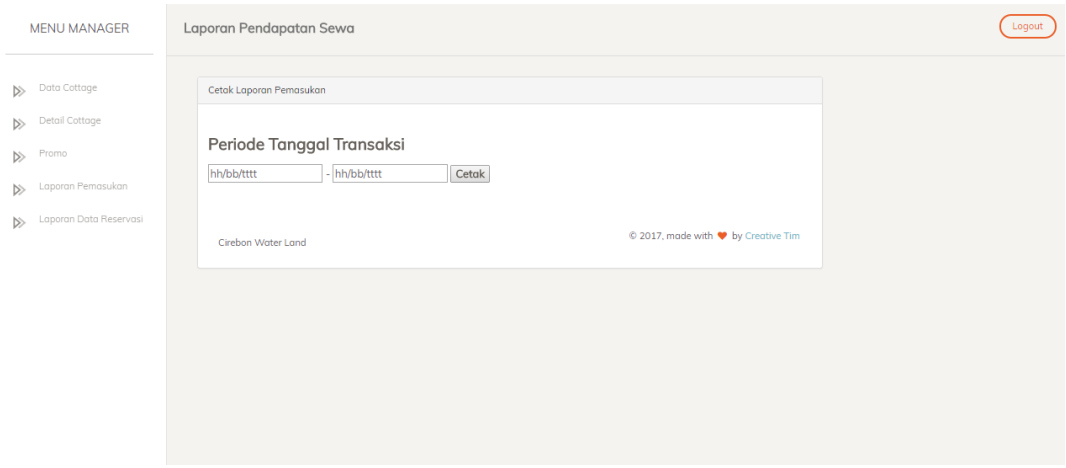
Gambar 4.33 Form Check In

7. *Form Check Out*



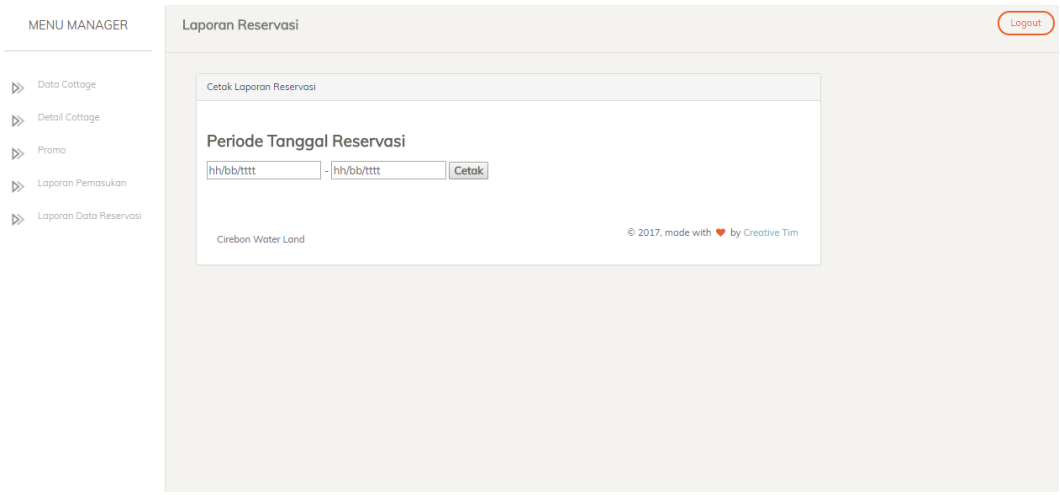
Gambar 4.34 *Form Check Out*

8. Form Laporan Pendapatan Sewa



Gambar 4.35 *Form* Laporan Pendapatan Sewa

9. Laporan Reservasi



Gambar 4.36 Laporan Reservasi

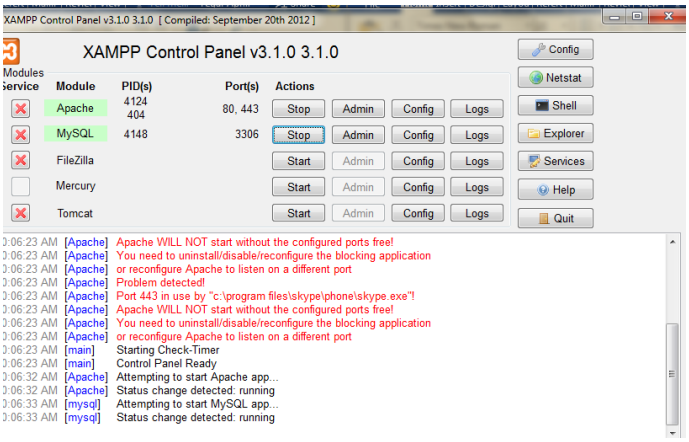
4.5.5 Implementasi instalasi program

Dalam menjalankan program aplikasi ini terdapat perangkat lunak dan file pendukung yang digunakan penulis gunakan diantaranya ialah:

1. Xampp version 3.5.2
2. Waterland1.sql (Database Sistem Informasi Resevasi Wisata Cottage Cirebon Waterland Ade Irma Suryani).

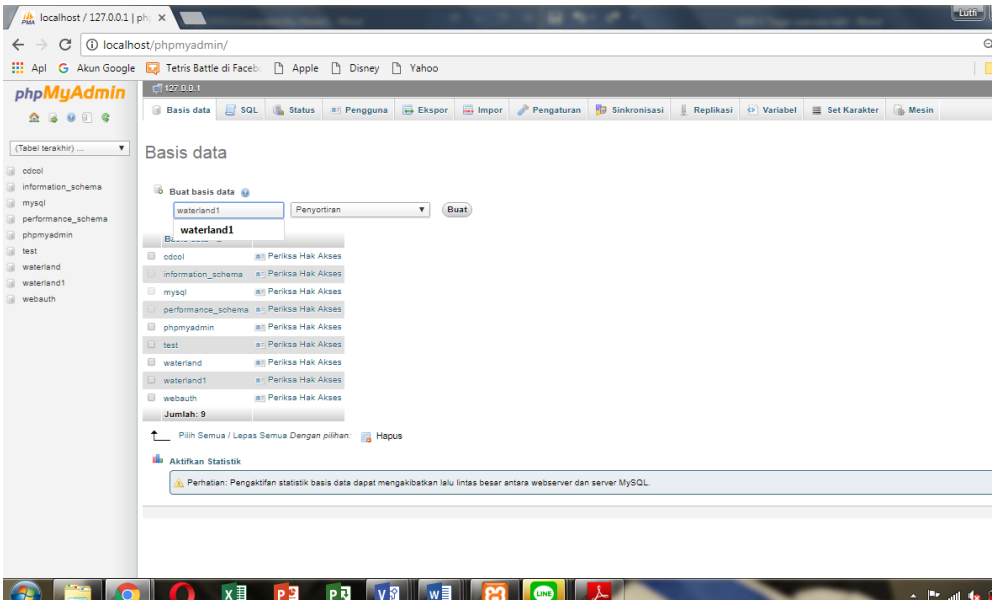
Pastikan aplikasi XAMPP telah ter-*install* pada perangkat yang akan digunakan.

1. Aktifkan XAMPP Control Panel dngan cara klik start > All Program > Xampp >pilih XAMPP Control Panel. Kemudian aktifkan *service* Apache dan MySQL seperti gambar dibawah ini.



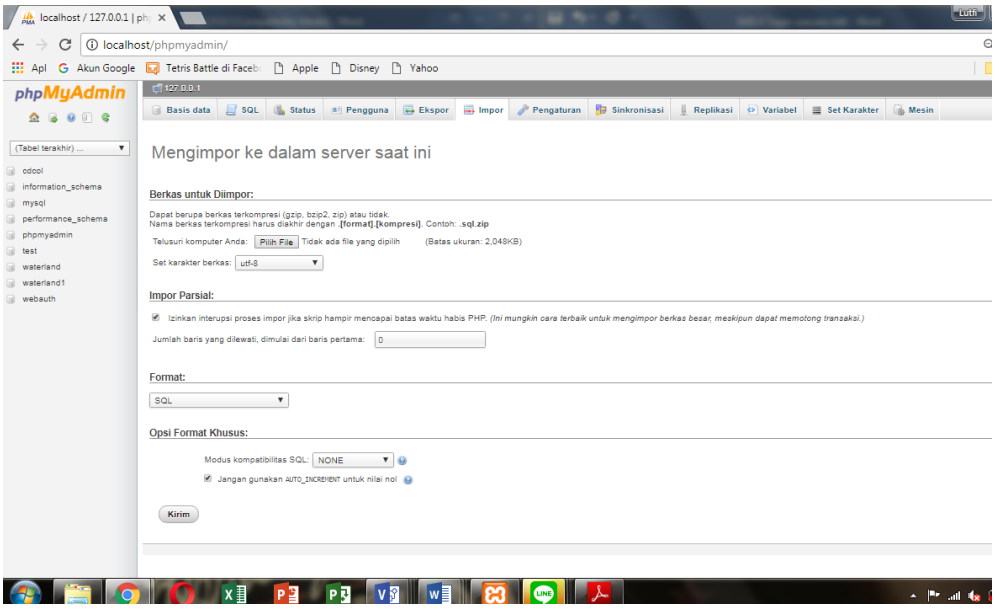
Gambar 4.37 Cara mengaktifkan *XAMPP Control Panel*

2. Buat *database* dengan nama waterland1 dengan cara, buka *browser* kemudian akses localhost/phpmyadmin pada *addressbar*, maka akan muncul seperti gambar dibawah ini kemudian klik “create/buat”



Gambar 4.38 Cara Membuat Basis Data

3. *Import database* dengan cara memilih menu “Import”, kemudian pilih filewaterland1.sql pada perangkat komputer lalu klik “kirim”.



Gambar 4.39 Cara *Import* basis data

4.5.6 Penggunaan Program

Sub bab ini berfungsi untuk menggambarkan cara penggunaan program yang telah dirancang dari hasil pembuatan Sistem Informasi Reservasi Wisata *Cottage*pada*Seaview Cottage* Cirebon *Waterland*. Cara pengoperasian sistemnya yaitu sebagai berikut:

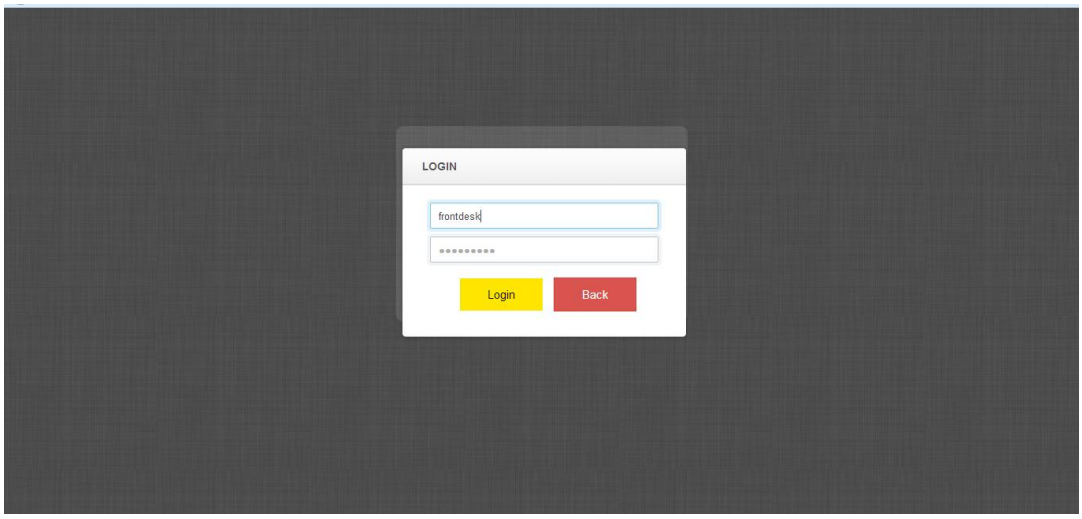
Hal pertama yang dilakukan agar bisa menggunakan sistem yaitu dengan membuka *www.seaviewcirebon.com*pada browser.



Gambar 4.40 Membuka Aplikasi Sistem Informasi Reservasi Cottage

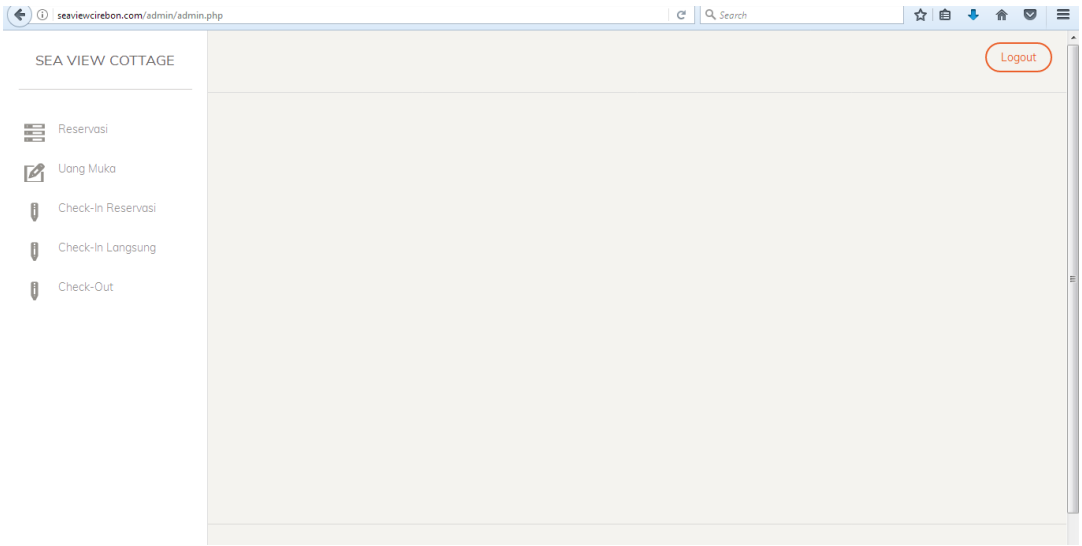
Untuk mengakses hak akses sebagai *admin* dapat dibuka pada alamat www.seaviewcirebon.com/admin lalu masukkan hak akses sebagai *admin front office* dengan memasukkan *username* frontdesk dan *password* frontdesk. Maka akan muncul tampilan untuk mengakses sebagai *front office*. Sama seperti halnya untuk mengakses hak akses *adminmanager* dan *laundry* hanya beda memasukkan *usarname* dan *password*nya.

Untuk *admin* sebagai *manager*, *username* dan *password*nya: manager dan manager. Dan untuk admin sebagai *laundry username* dan *password*nya menggunakan laundry dan laundry.



Gambar 4.41 tampilan halaman *login admin*

Untuk tampilan masuk hak akses sebagai *front office* sebagai berikut:



Gambar 4.42 tampilan menu halaman akses *front office*

untuk tampilan masuk hak akses sebagai *laundry* sebagai berikut:



Gambar 4.43 tampilan menu halaman akses *laundry*

Untuk tampilan masuk hak akses sebagai *manager* sebagai berikut:



Gambar 4.44 tampilan menu halaman akses *manager*