

# System Proposal



## HOSPITALITYPLUS

**Disusun oleh:  
Kelompok B11**

**Anggota Kelompok:**

- 1. Kerina Bakarudin (2217051027)**
- 2. Ananda Sheva Hidayat (2217051096)**
- 3. Fitra Dwi Nugraha (2257051007)**

## 1. Executive Summary

HospitalityPlus merupakan hotel yang berfokus pada efisiensi operasional dan kualitas layanan. Sistem Manajemen Hotel sudah terintegrasi yang mencakup proses pemesanan yang rinci. Dalam sistem ini, pelanggan dapat melakukan pemesanan kamar, menelusuri fasilitas hotel, dan melakukan pembayaran. Proses pemesanan dimulai ketika pelanggan memilih jenis kamar dan layanan yang diinginkan, dan setelah pemesanan dikonfirmasi, sistem akan menghasilkan ID pemesanan unik dan merinci rincian produk serta total pembayaran. Pelanggan dapat melanjutkan untuk melakukan pembayaran menggunakan metode pembayaran yang telah disediakan dalam sistem. Pada sisi pegawai, mereka dapat mengakses daftar pemesanan yang masuk, mengelola ketersediaan kamar, dan memproses setiap pemesanan.

Pada studi kasus yang diberikan, metodologi System Development Life Cycle (SDLC) yang cocok untuk pengerjaannya adalah Agile. Implementasi SDLC Agile akan membantu HospitalityPlus dalam mengelola proyek pengembangan sistem manajemen hotel dengan efisien. Metode ini akan memungkinkan tim untuk berkolaborasi dan mengidentifikasi dan mengatasi masalah secara berkelanjutan, sehingga sistem yang dihasilkan memenuhi harapan pelanggan dan pegawai.

## 2. System Request

Untuk memastikan bahwa proyek ini dapat memenuhi kebutuhan bisnis organisasi, telah diusulkan sebuah System Request kepada Project Sponsor yang di dalamnya berisi informasi mengenai Project Sponsor, Business Needs, Business Requirements, Business Values, dan Special Issues/Constraints.

### 2.1. Project Sponsor

Project Sponsor pada pengembangan sistem informasi ini adalah sebagai berikut:

Nama	: Zulkfli Kurniawan Huang
Jabatan	: Chief Executive Officer
No. HP	: 080909090909
E-Mail	: zkfkfn@email.com
Alamat	: Kayangan Blok A1

### 2.2. Business Needs

Berdasarkan tujuan visi dan misi dari HospitalityPlus yang berfokus pada peningkatan operasional dan kualitas layanan, pada perusahaan ini memerlukan beberapa kebutuhan bisnis antara lain : Efisiensi Operasional seperti proses manajemen kamar dan proses pemesanan kamar. Kemudian memerlukan peningkatan Layanan Pelanggan seperti memberi pengalaman yang baik terhadap pelanggan untuk memberikan layanan menjelajahi produk hotel, memilih kamar hotel.

### 2.3. Business Requirements

Kebutuhan bisnis pada pengembangan sistem informasi ini terdiri dari kebutuhan non fungsional dan kebutuhan fungsional.

Kebutuhan non fungsional yaitu:

- Keamanan Sistem : Pada sistem ini harus membutuhkan keamanan tingkat tinggi untuk melindungi data seperti informasi pribadi dan data transaksi pembayaran milik pelanggan.
- Ketahanan Sistem : Pada sistem ini harus memiliki ketahanan terhadap tingginya permintaan saat proses pemesanan tanpa mengalami penurunan kinerja.
- Manajemen Pemeliharaan : Pada sistem ini memiliki kemudahan dalam pemeliharaan sistem dan menerapkan pembaruan perangkat lunak
- Usability : Pada sistem ini memiliki antarmuka yang ramah terhadap pengguna.

Sedangkan kebutuhan fungsional adalah sebagai berikut:

- Manajemen Pemesanan : Informasi detail kamar, fasilitas hotel dan produk hotel.
- Manajemen Pelanggan : Sistem mendapatkan informasi pelanggan.
- Manajemen Transaksi : Mencakup informasi pesanan, ketersediaan kamar dan pembayaran dari pelanggan dan memberikan ID pemesanan yang unik.
- Manajemen Pegawai : Pengelolaan daftar pesanan secara detail dan pemrosesan pesanan dari pelanggan serta pegawai dapat memantau dan memanajemen ketersediaan kamar dengan efisien.
- Pembayaran Online : Memberikan kemudahan bertransaksi yang aman dan andal.
- Notifikasi : Memberikan notifikasi seputar status pemesanan dan detail informasi tamu terhadap pelanggan dan pegawai

## 2.4. Business Value

Adapun nilai bisnis yang dapat diperoleh dengan adanya Sistem ini diuraikan menjadi 2 poin, yaitu tangible dan intangible.

Tangible:

- Peningkatan Pendapatan : Dengan hadirnya sistem ini diharapkan bisa meningkatkan pendapatan hotel dari efisiensi operasional yang jauh lebih baik dan meningkatkan jumlah pemesanan.
- Efisiensi Biaya Operasional : Dengan adanya sistem ini diharapkan bisa mengurangi biaya operasional terkait tugas manual, kesalahan yang dilakukan manusia dan manajemen ketersediaan kamar yang tidak efisien.

Intangible:

- Meningkatkan Kesetiaan Pelanggan : Dari proses pengalaman pemesanan yang baik dapat membantu membangun loyalitas pelanggan.
- Meningkatkan Citra Merek : Dengan adanya sistem ini yang efisien dan ramah pengguna dapat meningkatkan citra merek Hotel yang inovatif, modern dan peduli terhadap pelanggan

## 2.5. Special Issues / Constraints

Special issue atau constraint yang dihadapi dalam pengembangan sistem informasi ini adalah

- Keamanan Data dan Informasi : Memastikan perlindungan data yang optimal merupakan kewajiban yang harus dilakukan karena sistem menyimpan informasi sensitif pelanggan.
- Pelatihan Pegawai : Pengembangan sistem yang modern memungkinkan untuk diperlukannya pelatihan pegawai hotel agar dapat mengelola dan menggunakan sistem yang dibuat menjadi lebih efektif.

## 3. Requirement Definitions

### 3.1. Nonfunctional Requirements

- a) Operational Requirement:
  - Sistem harus bekerja secara penuh guna mendukung pemesanan yang dapat dilakukan kapan saja terkecuali pada saat jadwal pemeliharaan yang berlangsung.
  - Sistem harus dapat dipakai secara multi-platform untuk menjamin pengguna dapat mengakses layanan dari berbagai perangkat pintar.
- b) Performance Requirement:
  - Sistem harus mampu memberikan tanggapan dengan cepat untuk memastikan pengguna merasakan responsivitas yang optimal.
  - Sistem harus mampu menangani jumlah pengguna yang banyak tanpa mengalami penurunan performa.
- c) Reliability Requirement:
  - Sistem harus memiliki tingkat ketersediaan dan waktu operasional yang tinggi untuk mencegah kegagalan selama proses pemesanan.
- d) Safety Requirements:
  - Sistem harus memiliki sistem cadangan dan pemulihan data untuk menjaga keamanan informasi pemesanan dari pelanggan agar tidak kehilangan atau mengalami kerusakan
- e) Security Requirements:
  - Sistem memberikan perlindungan terhadap data pribadi pelanggan dengan mengimplmentasikan enkripsi dan firewall.
- f) Usability Requirements:
  - Sistem memberikan antarmuka yang dirancang secara sederhana dan responsif yang dapat mudah digunakan oleh pelanggan dan pegawai

### 3.2. Functional Requirements

Sistem informasi yang dibangun memiliki kebutuhan fungsional sebagai berikut:

- a) Manajemen Pelanggan  
Sistem dapat mencatat informasi lengkap data dari pelanggan.
- b) Manajemen Pegawai  
Sistem dapat mampu membantu pengelolaan pesanan dari pelanggan.

- c) Manajemen Transaksi  
Sistem mampu mencatat transaksi dengan baik seperti pembayaran hingga informasi pemesanan.
- d) Manajemen Pemesanan  
Sistem mampu memberikan informasi fasilitas hotel hingga produk hotel.
- e) Pembayaran Online  
Sistem mampu menyediakan fungsi pembayaran secara online untuk kemudahan para pelanggan.
- f) Notifikasi  
Sistem mampu memberikan notifikasi kepada pelanggan seperti status pesanan dan kepada pegawai seperti detail informasi tamu.

#### 4. Requirement Analysis Strategy

Terdapat beberapa kegiatan yang dapat dilakukan dalam mencari informasi mengenai kebutuhan sistem, yaitu Interview, Joint Application Development (JAD), Questionnaires, Document Analysis, dan Observasi. Dalam hal ini, kami memilih untuk melakukan kegiatan:

##### 4.1. Hasil Kegiatan Wawancara

Berdasarkan hasil wawancara ...

##### 4.2. Hasil Kegiatan Observasi

Berdasarkan hasil observasi ...

##### 4.3. Requirement Analysis Strategy (Strategi Analisis Kebutuhan)

Pemetaan Kebutuhan dengan Proses As-Is dan To-Be-System Berikut ini adalah pemetaan kebutuhan (baik fungsional maupun non fungsional) dengan proses As-Is dan proses To-Be.

**Tabel 4.3 Matriks Kebutuhan Fungsional As-Is dan To-Be**

Kebutuhan Fungsional		
No.	Proses berjalan (As-Is)	Sistem yang akan dibuat (To-Be)
1.	Merchant melakukan ...	Merchant dapat melakukan ...

**Tabel 4.4 Matriks Kebutuhan Non Fungsional As-Is dan To-Be**

<b>Kebutuhan Non-Fungsional</b>		
<b>No.</b>	<b>Proses berjalan (As-Is)</b>	<b>Sistem yang akan dibuat (To-Be)</b>
<b>1.</b>	<b>Operational Requirements</b>	
	Pelanggan memilih kamar atau memilih fasilitas	- Sistem harus bekerja secara penuh guna mendukung pemesanan yang dapat dilakukan kapan saja terkecuali pada saat jadwal pemeliharaan yang berlangsung.
<b>2</b>	<b>Performance Requirements</b>	
	Pelanggan memiliki jumlah banyak akun	- Sistem harus mampu menangani jumlah pengguna yang banyak tanpa mengalami penurunan performa.
	Pelanggan memilih jenis kamar dan fasilitas dalam kurun waktu yang cepat	- Sistem harus mampu memberikan tanggapan dengan cepat untuk memastikan pengguna merasakan responsivitas yang optimal.
<b>3</b>	<b>Reliability Requirements</b>	
	Sistem menerima banyak data masukan dari pelanggan dan pegawai	- Sistem harus memiliki tingkat ketersediaan dan waktu operasional yang tinggi untuk mencegah kegagalan selama proses pemesanan.
<b>4</b>	<b>Safety Requirements</b>	
	Sistem memiliki sistem cadangan dan pemlihan data	- Sistem harus memiliki sistem cadangan dan pemulihan data untuk menjaga keamanan informasi pemesanan dari pelanggan agar tidak kehilangan atau mengalami kerusakan
<b>5</b>	<b>Security Requirements</b>	
	Sistem menyimpan data pribadi pelanggan dan pegawai	- Sistem memberikan perlindungan terhadap data pribadi pelanggan dengan mengimplmentasikan enkripsi dan firewall.
<b>6</b>	<b>Usability Requirements</b>	
	sistem harus mempunyai antarmuka yang responsif	- Sistem memberikan antarmuka yang dirancang secara sederhana dan responsif yang dapat mudah digunakan oleh pelanggan dan pegawai

Dari gap antara proses yang berjalan dengan sistem yang akan dibuat, maka dapat dilakukan BPA/BPI/BPR, yaitu menyediakan sistem yang dapat meningkatkan efisiensi operasional dan kualitas layanan. Sistem yang akan dibuat harus dapat memenuhi kebutuhan pelanggan dan pegawai, serta dapat meningkatkan produktivitas dan efisiensi operasional hotel.

Berikut adalah beberapa hal yang dapat dilakukan untuk meningkatkan efisiensi operasional dan kualitas layanan dengan sistem yang baru:

1. Otomatisasi proses pemesanan. Sistem yang baru dapat mengotomatisasi proses pemesanan, sehingga pelanggan dapat melakukan pemesanan kamar dengan lebih mudah

dan cepat. Sistem juga dapat memberikan informasi yang lebih akurat tentang ketersediaan kamar dan harga.

2. Peningkatan layanan pelanggan. Sistem yang baru dapat meningkatkan layanan pelanggan dengan menyediakan fitur-fitur seperti live chat, chatbot, dan FAQ. Fitur-fitur ini dapat membantu pelanggan untuk mendapatkan informasi yang mereka butuhkan dengan lebih cepat dan mudah.
3. Peningkatan manajemen ketersediaan kamar. Sistem yang baru dapat membantu hotel untuk mengelola ketersediaan kamar secara lebih efektif. Sistem dapat memprediksi permintaan kamar dan menyesuaikan ketersediaan kamar secara real-time.

Pemilihan sistem yang tepat untuk Hospitality Plus harus disesuaikan dengan kebutuhan dan anggaran hotel. Sistem yang dipilih harus dapat memenuhi kebutuhan pelanggan dan pegawai, serta dapat meningkatkan produktivitas dan efisiensi operasional hotel.

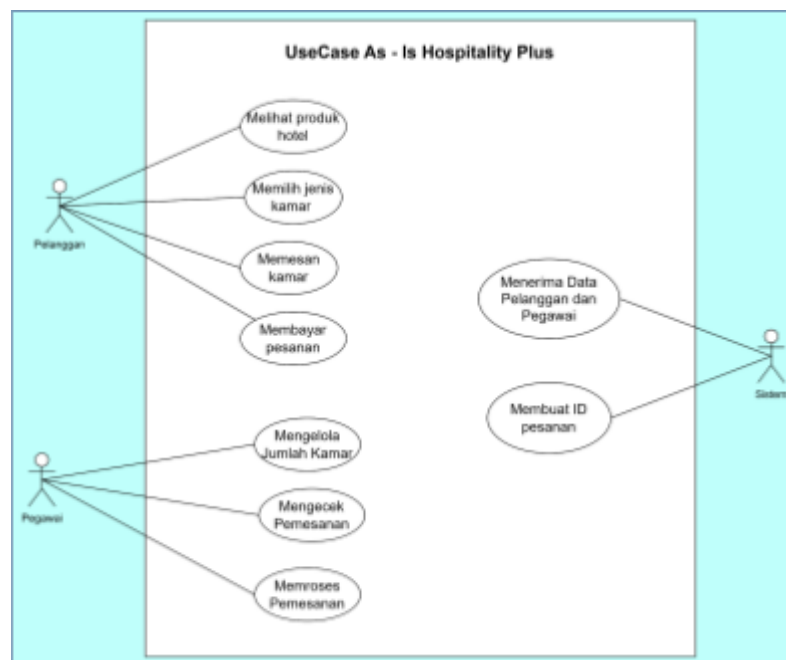
## 5. Functional Model

Pada bagian ini dijelaskan mengenai *use case diagram*, *activity diagram* dan *use case description* dari proyek sistem informasi yang dilakukan (Dennis et al, 2012).

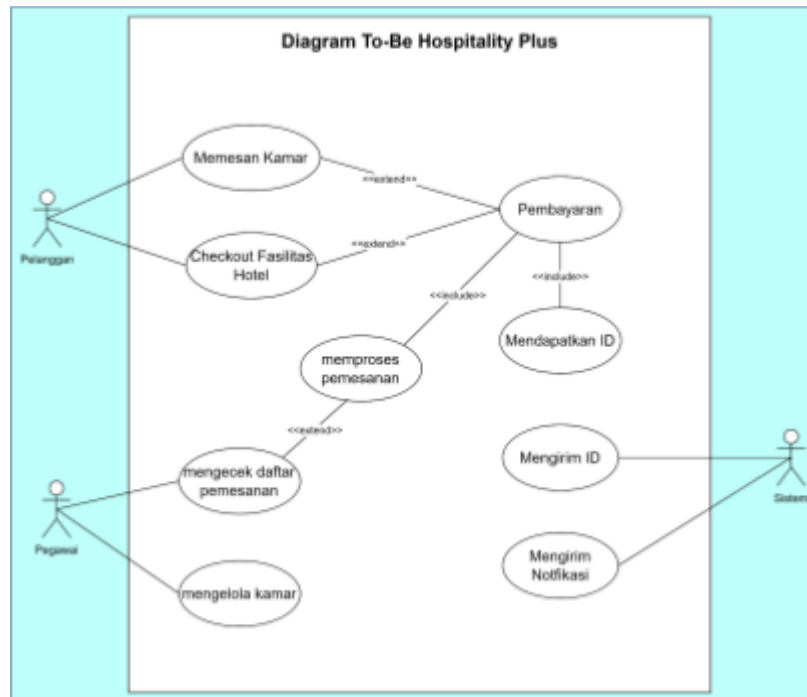
### 5.1 Use Case Diagram

*Use case diagram* dibagi menjadi 2 bagian yaitu *use case As-Is* dan *To-Be* yang dapat dilihat pada Gambar 11 dan Gambar 12:

**Gambar 5.1 Use Case Diagram As-Is**



**Gambar 5.1.2 Use Case Diagram To-Be**



## 5.2 Use Case Description

Deskripsi setiap *use case diagram* adalah sebagai berikut:

### Use Case Description

<b>Use Case Name: Pemesanan Kamar</b>	<b>ID: 001</b>	<b>Priority: High</b>
<b>Actor: Pelanggan</b>		
<b>Description:</b> Use case ini menggambarkan pelanggan yang ingin memesan kamar		
<b>Trigger:</b> Pelanggan mencari kamar		
<b>Preconditions:</b> Pelanggan sudah login ke dalam aplikasi		
<b>Normal Course:</b> <ol style="list-style-type: none"> <li>1. Pelanggan Membuka Aplikasi</li> <li>2. Pelanggan Melakukan Login</li> <li>3. Pelanggan Memesan kamar yang dipilih</li> </ol>		
<b>Postconditions:</b> Pelanggan berhasil memilih kamar yang diinginkan		
<b>Sub Flows:</b> Pelanggan bisa memilih kategori kamar yang akan dipilih		
<b>Alternate / Exceptional Flows: -</b> <ol style="list-style-type: none"> <li>1. Jika tidak ada hasil yang sesuai dengan kriteria pencarian, sistem memberikan notifikasi bahwa kamar tidak ditemukan.</li> <li>2. Jika terjadi kesalahan pada saat pencarian, sistem memberikan pesan kesalahan dan meminta pelanggan untuk mencoba lagi</li> <li>3. Jika hasil kamar sudah penuh, sistem memberikan pesan bahwa semua kamar telah diisi / penuh</li> </ol>		



<b>Use Case Name: Pemesanan Kamar</b>	<b>ID: 001</b>	<b>Priority: High</b>
<b>Jumlah Transaksi:</b>		

<b>Use Case Name: Checkout Fasilitas Hotel</b>	<b>ID : 002</b>	<b>Priority : High</b>
<b>Actor: Pelanggan</b>		
<b>Description:</b> Use case ini menggambarkan pelanggan yang ingin checkout fasilitas hotel		
<b>Trigger:</b> Pelanggan mencari fasilitas hotel		
<b>Preconditions:</b> Pelanggan sudah login ke dalam aplikasi		
<b>Normal Course:</b> <ol style="list-style-type: none"> <li>1. Pelanggan Melakukan pembayaran pada kamar yang dipesan</li> <li>2. Pelanggan Melakukan Login</li> <li>3. Pelanggan Memilih fasilitas yang ingin di checkout</li> </ol>		
<b>Postconditions:</b> Pelanggan berhasil memilih fasilitas hotel yang akan di checkout		
<b>Sub Flows:</b> Pelanggan bisa memilih lebih dari satu fasilitas hotel yang ingin di checkout		
<b>Alternate / Exceptional Flows: -</b> <ol style="list-style-type: none"> <li>1. Jika tidak ada hasil yang sesuai dengan kriteria pencarian, sistem memberikan notifikasi bahwa fasilitas hotel tidak ditemukan.</li> <li>2. Jika terjadi kesalahan pada saat pencarian, sistem memberikan pesan kesalahan dan meminta pelanggan untuk mencoba lagi.</li> </ol>		
<b>Jumlah Transaksi:</b>		

<b>Use Case Name: Mengecek Daftar Pemesanan</b>	<b>ID: 003</b>	<b>Priority: High</b>
<b>Actor: Pegawai</b>		
<b>Description:</b> Use case ini menggambarkan pegawai yang ingin mengecek pemesanan		
<b>Trigger:</b> Pegawai mengecek pemesanan		
<b>Preconditions:</b> Pegawai sudah login ke dalam aplikasi		
<b>Normal Course:</b> <ol style="list-style-type: none"> <li>1. Pegawai Membuka Aplikasi</li> <li>2. Pegawai Melakukan Login</li> <li>3. Pegawai Mengecek daftar pemesanan</li> </ol>		
<b>Postconditions:</b> Pegawai berhasil mengecek pemesanan dari pelanggan		
<b>Sub Flows:</b> Pegawai memilih daftar pemesanan satu per satu dari pelanggan		
<b>Alternate / Exceptional Flows: -</b> <ol style="list-style-type: none"> <li>1. Jika tidak ada hasil daftar pemesanan yang sesuai dengan kriteria pencarian, sistem memberikan notifikasi bahwa kamar tidak ditemukan.</li> <li>2. Jika terjadi kesalahan pada saat pengecekan, sistem memberikan pesan kesalahan dan meminta pegawai untuk mencoba lagi.</li> </ol>		
<b>Jumlah Transaksi:</b>		

<b>Use Case Name: Mengelola Kamar</b>	<b>ID: 004</b>	<b>Priority: High</b>
<b>Actor: Pegawai</b>		

<b>Use Case Name: Mengelola Kamar</b>	<b>ID: 004</b>	<b>Priority: High</b>
<b>Description:</b> Use case ini menggambarkan pegawai yang ingin mengelola kamar		
<b>Trigger:</b> Pegawai mengelola kamar		
<b>Preconditions:</b> Pegawai sudah login ke dalam aplikasi		
<b>Normal Course:</b> <ol style="list-style-type: none"> <li>1. Pegawai Melakukan Login</li> <li>2. Pegawai Mengelola kamar pelanggan</li> </ol>		
<b>Postconditions:</b> Pegawai berhasil mengelola kamar dari pelanggan		
<b>Sub Flows:</b> Pegawai mengecek dan mengelola kamar pelanggan satu per satu		
<b>Alternate / Exceptional Flows:</b> - <ol style="list-style-type: none"> <li>1. Jika tidak ada yang sesuai dengan kriteria pencarian, sistem memberikan notifikasi bahwa kamar tidak ditemukan.</li> <li>2. Jika terjadi kesalahan pada saat pengecekan, sistem memberikan pesan kesalahan dan meminta pegawai untuk mencoba melakukan pengecekan lagi.</li> </ol>		
<b>Jumlah Transaksi:</b>		

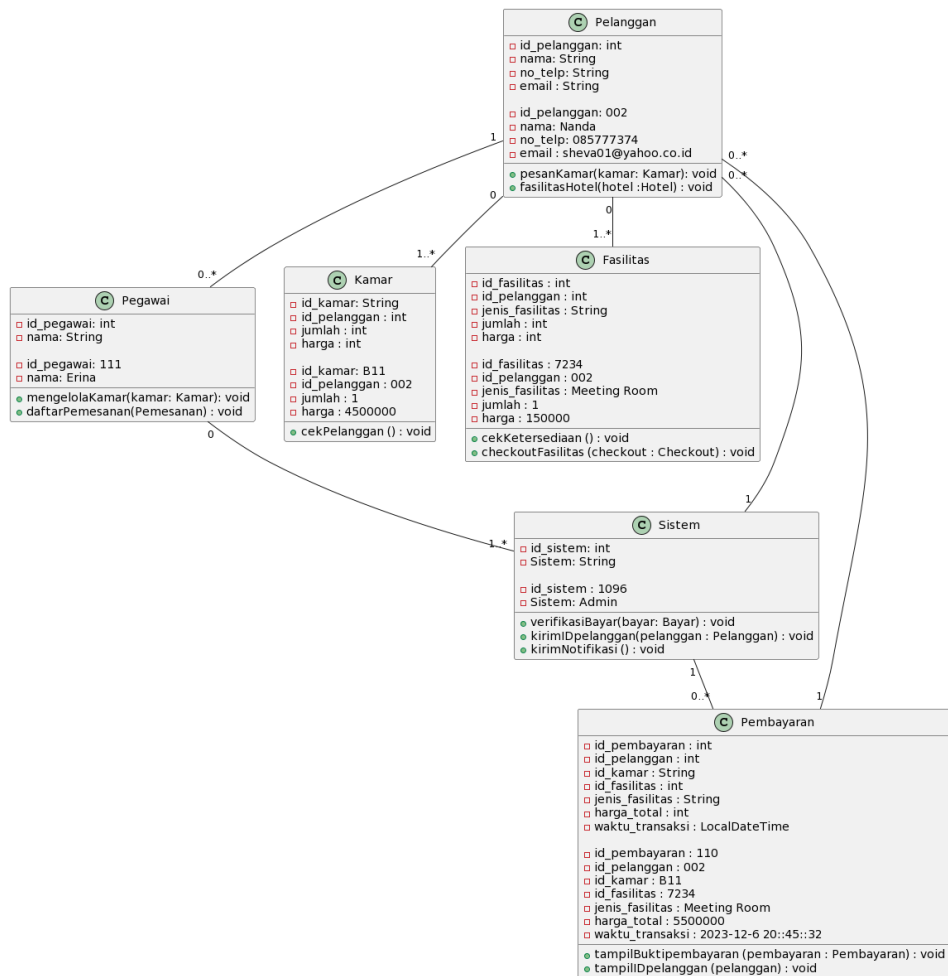
<b>Use Case Name: Checkout Fasilitas Hotel</b>	<b>ID : 005</b>	<b>Priority : Very High</b>
<b>Actor: Pelanggan</b>		
<b>Description:</b> Use case ini menggambarkan pelanggan yang ingin melakukan pembayaran		
<b>Trigger:</b> Pelanggan sudah memilih kamar atau checkout fasilitas hotel		
<b>Preconditions:</b> Pelanggan sudah login ke dalam aplikasi		
<b>Normal Course:</b> <ol style="list-style-type: none"> <li>1. Pelanggan Melakukan login</li> <li>2. Pelanggan sudah memilih kamar <ol style="list-style-type: none"> <li>2.1 Pelanggan sudah checkout fasilitas hotel</li> </ol> </li> <li>3. Pelanggan Melakukan Pembayaran</li> </ol>		
<b>Postconditions:</b> Pelanggan berhasil melakukan pembayaran		
<b>Sub Flows:</b> Pelanggan mendapatkan ID Pemesanan		
<b>Alternate / Exceptional Flows:</b> - <ol style="list-style-type: none"> <li>1. Jika terjadi kesalahan pada saat pembayaran, sistem memberikan pesan kesalahan dan meminta pelanggan untuk mencoba lagi.</li> </ol>		
<b>Jumlah Transaksi:</b>		

<b>Use Case Name: Mengirim ID</b>	<b>ID: 006</b>	<b>Priority: High</b>
<b>Actor: Sistem</b>		
<b>Description:</b> Use case ini menggambarkan sistem mengirim ID pemesanan		
<b>Trigger:</b> Sistem mengirim ID pesanan		
<b>Preconditions:</b> Pelanggan sudah melakukan pembayaran		
<b>Normal Course:</b> <ol style="list-style-type: none"> <li>1. Pelanggan Melakukan Login</li> <li>2. Pelanggan Memilih kamar yang dipesan <ol style="list-style-type: none"> <li>2.1 Pelanggan memilih fasilitas hotel yang dipesan</li> </ol> </li> </ol>		

<b>Use Case Name: Mengirim ID</b>	<b>ID: 006</b>	<b>Priority: High</b>
3. Pelanggan Melakukan pembayaran		
<b>Postconditions:</b> Pelanggan berhasil mendapatkan ID Pemesanan		
<b>Sub Flows:</b> Pelanggan mendapatkan ID pemesanan yang dikirim oleh sistem		
<b>Alternate / Exceptional Flows:</b> - 1. Jika tidak mendapatkan ID Pemesanan, sistem akan mengecek apakah pembayaran dilakukan sesuai prosedur yang berlaku . 2. Jika terjadi kesalahan pada ID pemesanan, pelanggan dapat melakukan complain melalui customer service / pegawai		
<b>Jumlah Transaksi:</b>		

### 5.3. Class Diagram

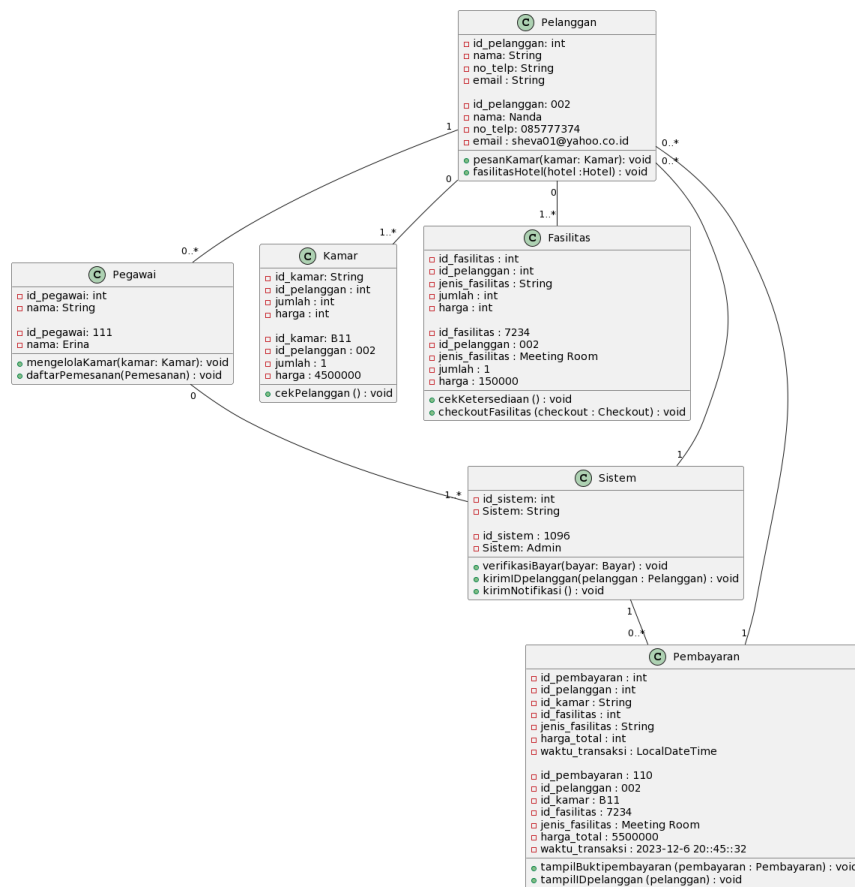
Selanjutnya dibuat Class Diagram berdasarkan kebutuhan sistem yang telah dibuat sebelumnya. Adapun Class Diagram adalah sebagai berikut



Gambar 5.3 Class Diagram

#### 5.4. Object Diagram

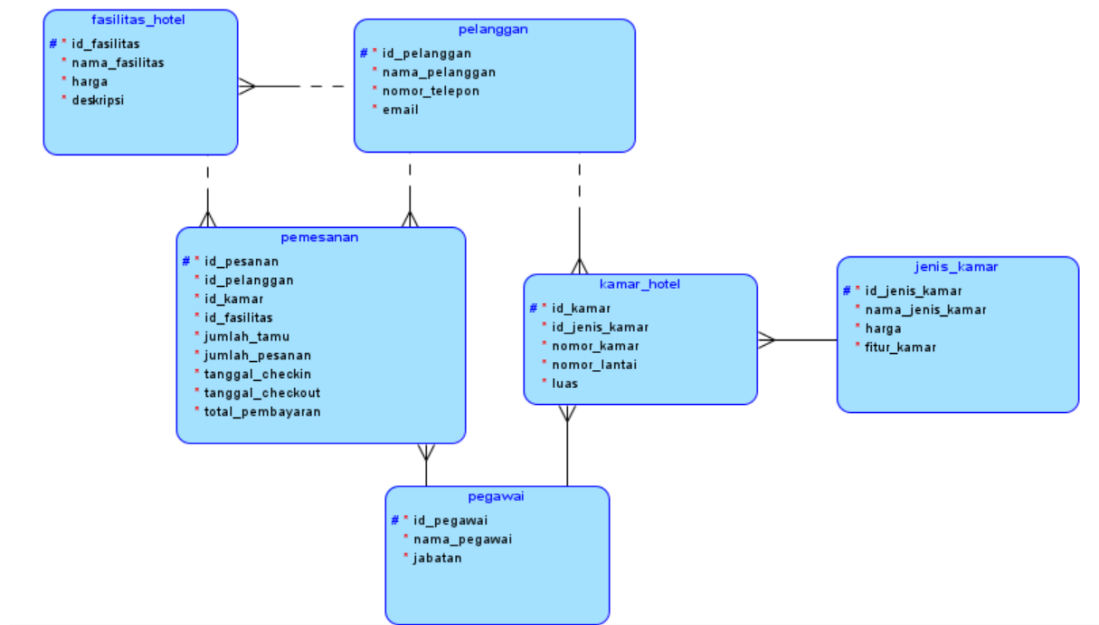
Selanjutnya dibuat Object Diagram berdasarkan Class Diagram yang telah dibuat sebelumnya. Adapun Object Diagram adalah sebagai berikut



Gambar 5.4 Object Diagram

### 5.5. Entity Relationship Diagram (ERD)

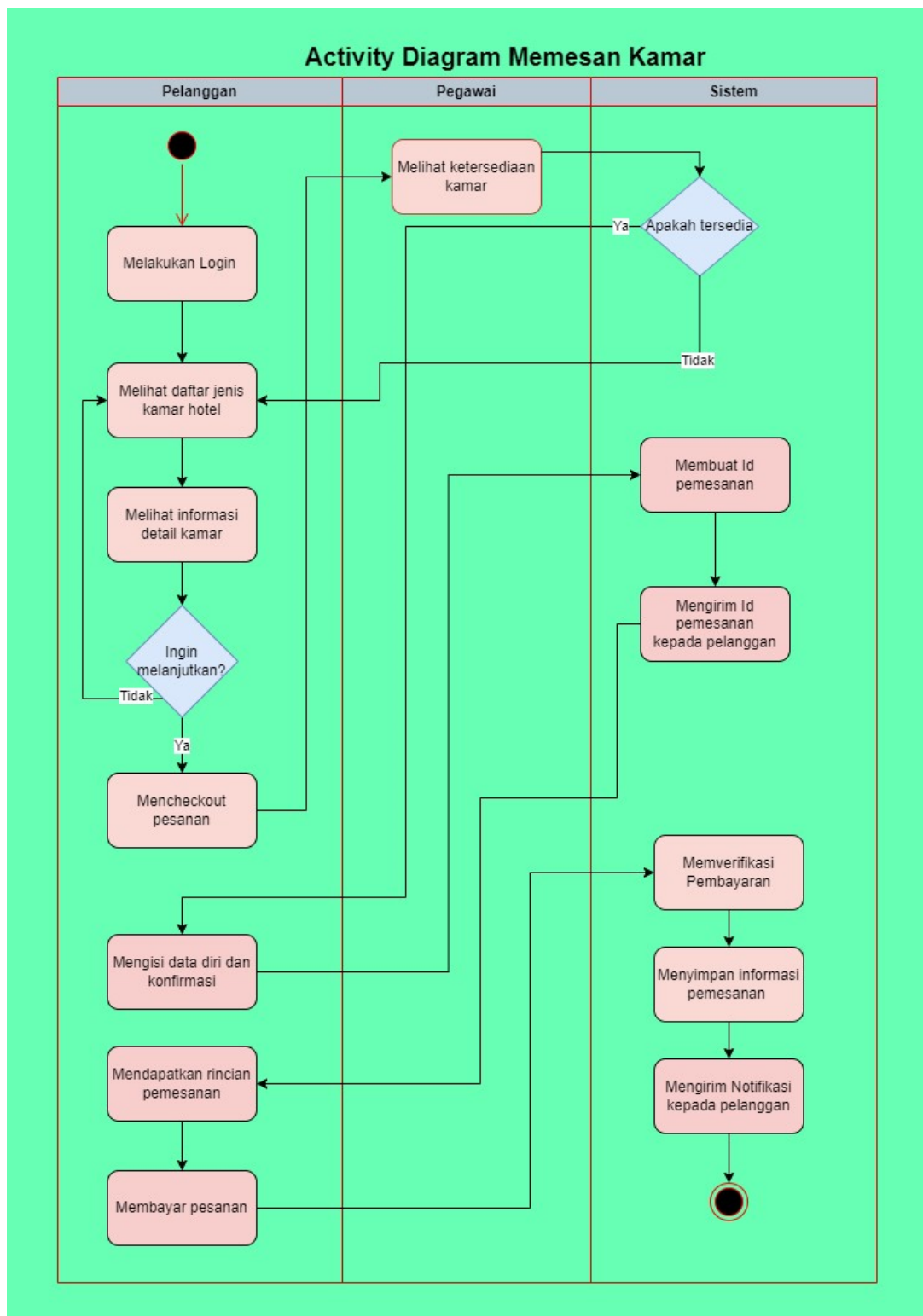
Selanjutnya dibuat ERD berdasarkan kebutuhan sistem yang telah dibuat sebelumnya. Adapun Entity Relationship Diagram adalah sebagai berikut



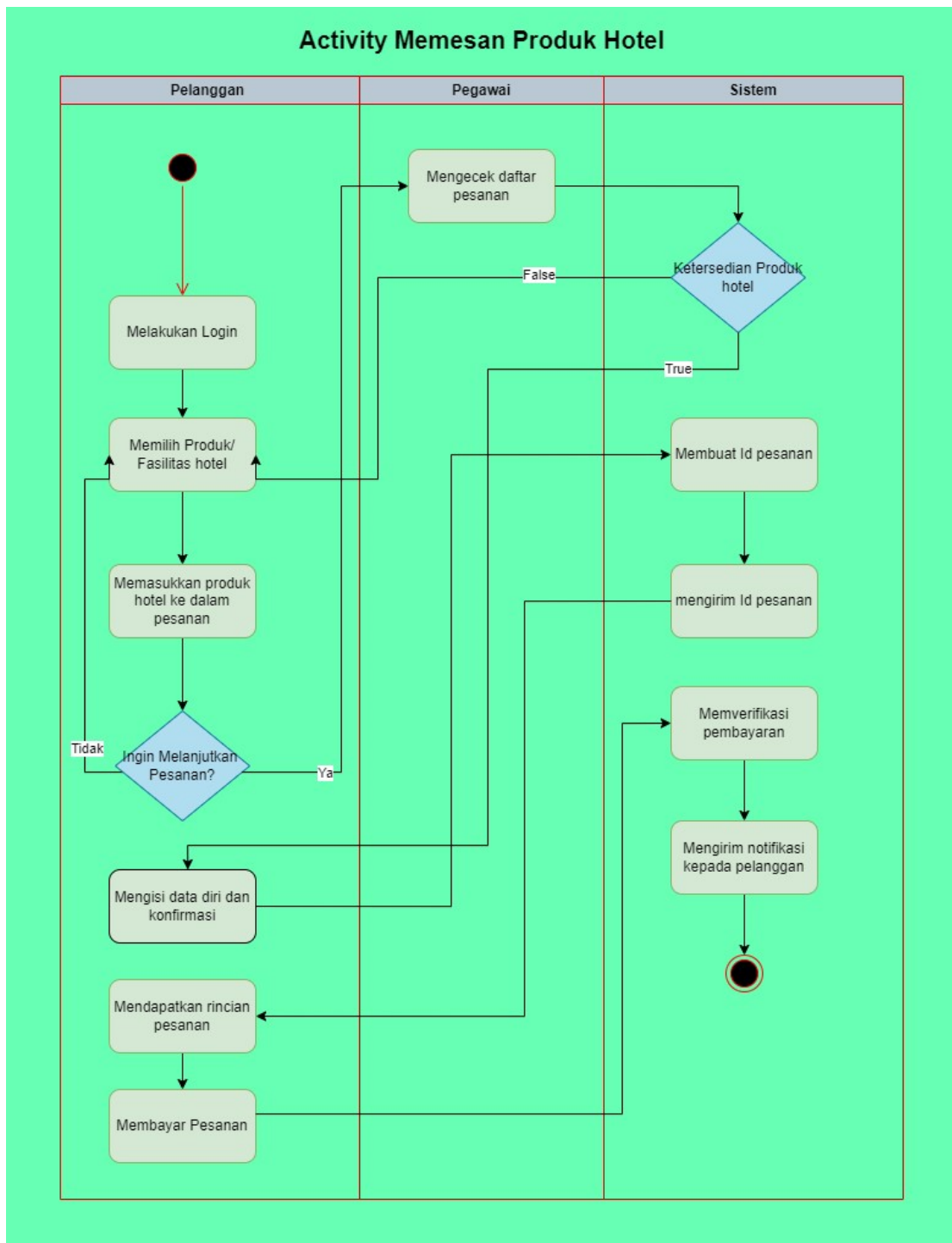
Gambar 5.5 ERD

## 5.6. Activity Diagram

Selanjutnya dibuat Activity Diagram berdasarkan Use Case yang telah dibuat sebelumnya. Adapun Activity Diagram adalah sebagai berikut

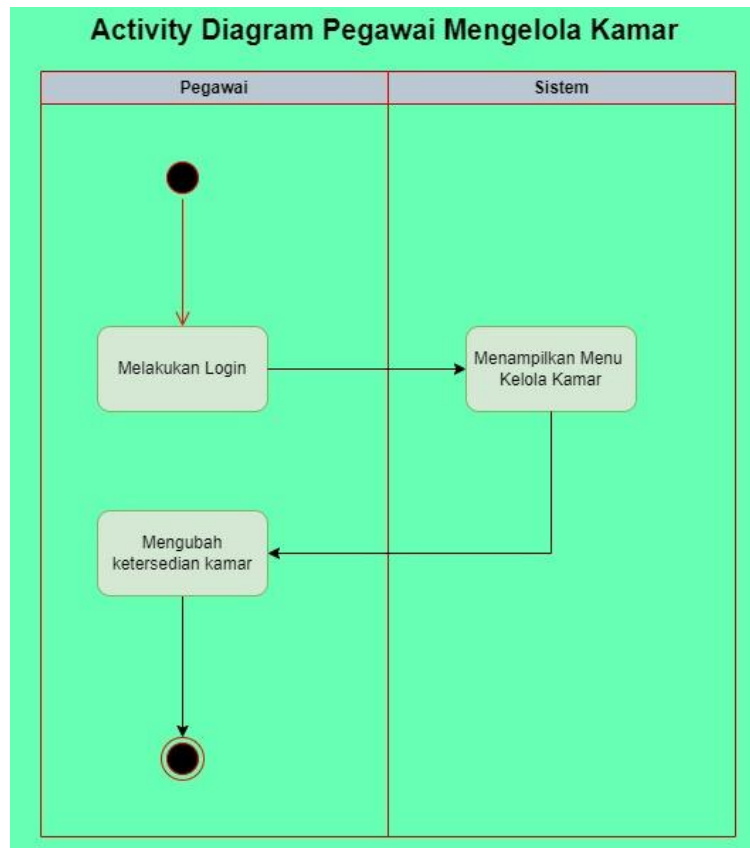


Gambar 5.6 Activity Diagram Memesan Kamar

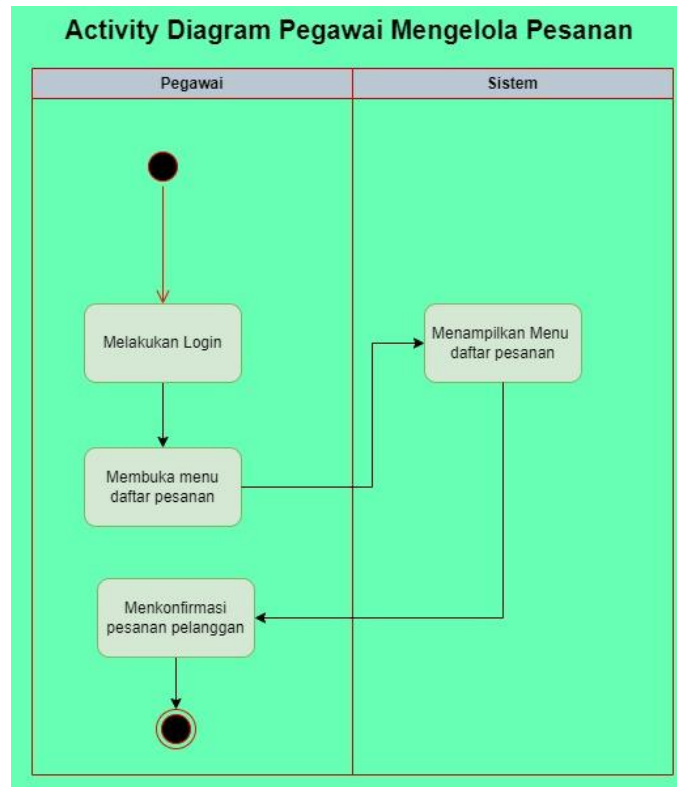


**Gambar 5.7 Activity Diagram Memesan Produk Hotel**





**Gambar 5.8 Activity Diagram Mengelola kamar**

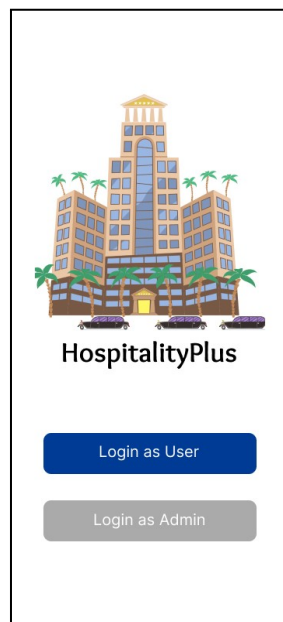


**Gambar 5.9 Activity Diagram Mengelola Pesanan**

## 6 User Interfaces

*User interface* merupakan bagian dari sistem yang berinteraksi dengan pengguna, meliputi tampilan layar yang menyediakan navigasi sistem, *screen* dan *form* serta *report* yang dihasilkan oleh sistem (Dennis et al, 2015). *User interfaces* yang dikembangkan pada sistem ini adalah sebagai berikut:

### 6.1 User Interface Home Page



**Gambar 6.1 UI Homepage**

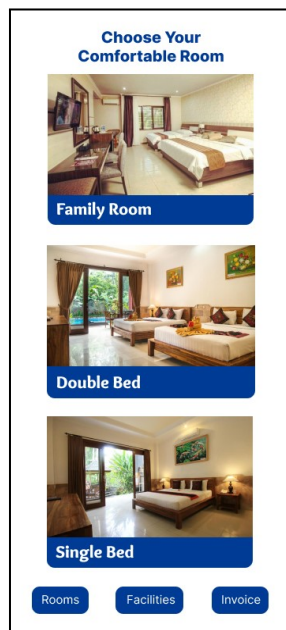
## 6.2 User Interface Login



A login form with two input fields and a button. The first input field is labeled "Username" and the second is labeled "Password". Below the input fields is a blue button labeled "Login".

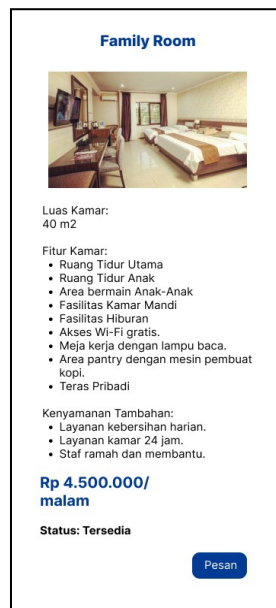
**Gambar 6.2 UI Login**

## 6.3 User Interface Jenis Kamar



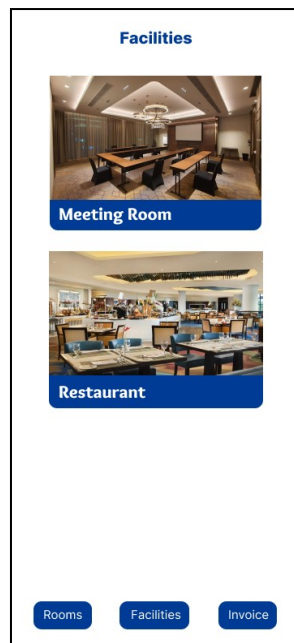
**Gambar 6.3 UI Jenis Kamar**

## 6.4 User Interface Detail Kamar



Gambar 6.4 UI Detail Kamar

## 6.5 User Interface Fasilitas Hotel



Gambar 6.5 UI Fasilitas

## 6.6 User Interface Detail Fasilitas

**Restaurant**  


Tersedia paket menu sarapan, makan siang dan makan malam.

**Rp 150.000/paket**

Pesan

**Gambar 6.6 UI Detail Fasilitas**

## 6.7 User Interface Form Pemesanan

**Pemesanan**  

Nama

Nomor Telepon

Email

Jumlah Tamu

Tanggal Check-in

Tanggal Check-out

Konfirmasi

**Gambar 6.7 UI Form Pemesanan**

## 6.8 User Interface Rincian Pesanan dan Pembayaran

**Rincian Pesanan**

**#ID117**

Nama

Nomor Telepon

Email

Room

Jumlah Tamu

Tanggal Check-in

Tanggal Check-out

Total Pembayaran  
**Rp 9.000.000**

**Bayar**

**Gambar 6.7 UI Form Rincian Pesanan dan Pembayaran**

## 6.9 User Interface Pembayaran Berhasil

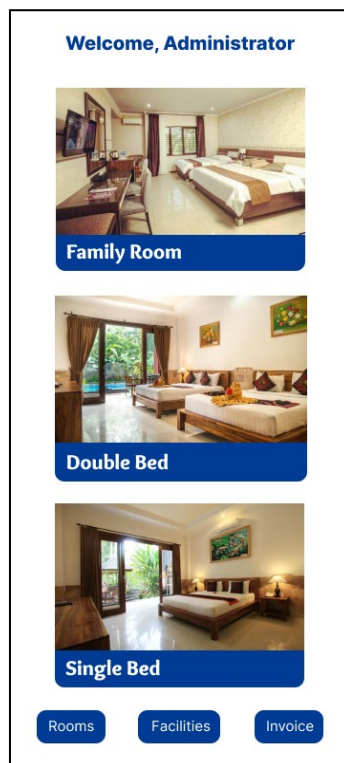
**PEMBAYARAN  
BERHASIL!**

Pesanan Anda akan segera kami proses

**OK**

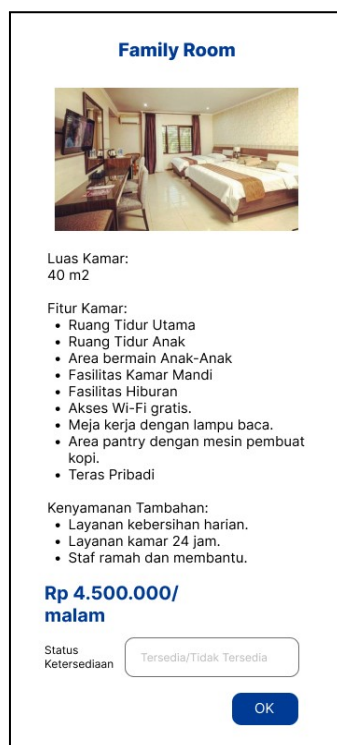
**Gambar 6.9 UI Pembayaran Berhasil**

## 6.10 User Interface Home Admin



Gambar 6.10 UI Home Admin

## 6.11 User Interface Kelola Ketersediaan Kamar



Gambar 6.11 UI Kelola Ketersediaan Kamar

## 6.12 User Interface Konfirmasi Pesanan

**Detail Pesanan**  
**#ID117**  
  
Nama   
  
Nomor  
Telepon   
  
Email   
  
Room   
  
Jumlah  
Tamu   
  
Tanggal  
Check-in   
  
Tanggal  
Check-out   
  
Total Pembayaran  
**Rp 9.000.000**

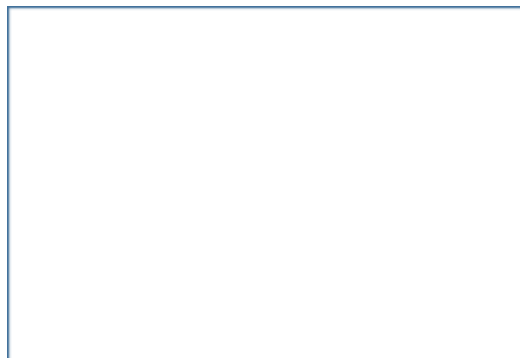
**Gambar 6.12 UI Konfirmasi Pesanan**

## 7 Physical Architecture Layer Design

Pada bagian ini dijelaskan mengenai *deployment diagram* dan *hardware-software specifications*.

### 7.1 Deployment Diagram

*Deployment diagram* merepresentasikan lingkungan untuk mengeksekusi sebuah *software* termasuk relasi antara komponen *hardware* yang digunakan dan komponen *software*. Rancangan *deployment diagram* untuk Markas Catering Information System adalah sebagai berikut:



**Gambar x Deployment Diagram**



## 7.2 Hardware-Software Specifications

*Hardware-Software specification* adalah sebuah dokumen yang menggambarkan spesifikasi *hardware* dan *software* yang dibutuhkan untuk mendukung sebuah aplikasi (Dennis et al, 2015). Spesifikasi dari *hardware* dan *software* pada pengembangan sistem ini dapat dilihat seperti berikut ini:

Tabel x Hardware-Software Specifications

Specification	Server Based	Client Based	Client-Server
Operating System	•		
Special Software	•		
Hardware	•		
Network	•		

## 8. System Prototype

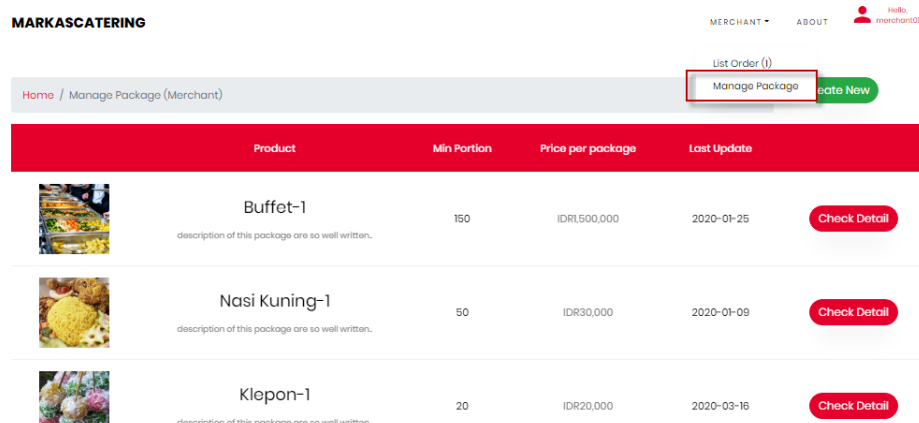
*System prototype* menyediakan fungsionalitas sistem yang akan digunakan oleh pengguna dan disetujui *project sponsor*. *System prototype* dari Sistem Manajemen Hotel pada HospitalityPlus dibagi berdasarkan fungsi yang ada.




Figma:

<https://www.figma.com/file/79hFdZGUFLpmLMFZ0K0EgO/HospitalityPlus?type=design&node-id=0%3A1&mode=design&t=U52g6BcJZDrbNKmm-1>

### 8.1 Manage Package

Fungsi Manage Package dimulai dengan Merchant pilih menu Manage Package:



MARKASCATERING					MERCHANT	ABOUT	Hella merchant01
Home / Manage Package (Merchant)					List Order (1)	Manage Package	State New
Product	Min Portion	Price per package	Last Update				
 Buffet-1 description of this package are so well written.	150	IDR1,500,000	2020-01-25	Check Detail			
 Nasi Kuning-1 description of this package are so well written.	50	IDR30,000	2020-01-09	Check Detail			
 Klepon-1 description of this package are so well written.	20	IDR20,000	2020-03-16	Check Detail			

Gambar 65 Menu Manage Package Merchant

## 9 Test Plan

*Testing* dimulai dengan pengembangan sebuah *test plan* yang menentukan serangkaian pengujian yang akan dilakukan. *Test plan* harus membahas semua produk yang dibuat selama pengembangan sistem (Dennis et al, 2015). Adapun *test plan* yang dilakukan pada pengembangan sistem ini terdiri dari *unit testing*, *integration testing*, *system testing* dan *acceptance testing*.

### 9.1 Acceptance Testing

Metode *acceptance testing* yang digunakan pada pengembangan sistem ini adalah *alpha testing* yaitu dengan data artificial. Pengujian dilakukan untuk mengkonfirmasi bahwa sistem yang dibangun memenuhi kebutuhan bisnis dan dapat diterima oleh pengguna.

**Tabel x Acceptance Testing**

No .	Testing ID	Objective	Tested By	Role	Date Tested	Actual Result	
						OK	NOT OK
1	UAT001	Merchant Menambah Package		Merchant			
2	UAT002						