

DPPL-07

DESKRIPSI PERANCANGAN PERANGKAT LUNAK

Travellight

untuk:

Seluruh Masyarakat

Dipersiapkan oleh:

Kelompok 7 IF-44-02

Hendrawan Aulia Rahman (1301200458)

Gisella Vania Dwipayana (1301204469)

Moch Ibni Hasyim Irwanto (1301204015)


Mahrnunissa Azmima Fitria (1301204016)

Salsabila Awwaliyyah Lestari (1301204257)

Program Studi S1 Informatika

Fakultas Informatika, Universitas Telkom

Jl. Telekomunikasi 1, Dayeuhkolot Bandung

 Telkom University	Prodi S1- Informatika Universitas Telkom	Nomor Dokumen		Halaman
		<i>DPPL-07</i>		<i>54</i>
		Revisi	<i>A</i>	<i>Tgl: 15 Januari 2023</i>

DAFTAR PERUBAHAN

Revisi	Deskripsi
A	Perancangan Rinci
B	
C	
D	
E	
F	
G	

INDEX TGL	-	A	B	C	D	E	F	G
Ditulis oleh		15 Januari 2023						
Diperiksa oleh								
Disetujui oleh								

Daftar Halaman Perubahan

Halaman	Revisi	Halaman	Revisi
8	A		

Daftar Isi

Daftar Isi	3
1. Pendahuluan	6
1.1 Tujuan Penulisan Dokumen	6
1.2 Lingkup Masalah	6
1.3 Definisi dan Istilah	6
1.4 Referensi	6
2 Perancangan Global	7
2.1 Rencana Lingkungan Implementasi	7
2.2 Deskripsi Arsitektur Perangkat Lunak	7
3 Perancangan Rinci	8
3.1 Realisasi Use Case	8
3.1.1 Use Case #1 <Pendaftaran Akun>	8
3.1.1.1 Use Case Scenario #1 <Pendaftaran Akun>	8
3.1.1.2 UI Design dan Deskripsi Objek UI #1 <Pendaftaran Akun>	9
3.1.1.3 Identifikasi Object dan Tipe nya #1 <Pendaftaran Akun>	11
3.1.1.4 Sequence Diagram #1 <Pendaftaran Akun>	12
3.1.2 Use Case #2 <Login>	12
3.1.2.1 Use Case Scenario #2 <Login>	12
3.1.2.2 UI Design dan Deskripsi Objek UI #2 <Login>	12
3.1.2.3 Identifikasi Object Baru & Tipe Kelas #2 <Login>	14
3.1.2.4 Sequence Diagram #2 <Login>	15
3.1.3 Use Case #3 <Update Profile>	15
3.1.3.1 Use Case Scenario #3<Update Profile>	15
3.1.3.2 UI Design dan Deskripsi Objek UI #3 <Edit Profile>	15
3.1.3.3 Identifikasi Object Baru & Tipe Kelas #3 <Update Profile>	17
3.1.3.4 Sequence Diagram #3 <Update Profile>	18
3.1.4 Use Case #4 <Manajemen Produk>	18
3.1.4.1 Use Case Scenario #4<Manajemen Produk>	18
3.1.4.2 UI Design dan Deskripsi Objek UI #4 <Manajemen Produk>	19
3.1.4.3 Identifikasi Object Baru & Tipe Kelas #4 <Manajemen Produk>	23
3.1.4.4 Sequence Diagram #4 <Manajemen Produk>	24
3.1.5 Use Case #5 <Pencarian Informasi Produk>	24
3.1.5.1 Use Case Scenario #5 <Pencarian Informasi Produk>	24
3.1.5.2 UI Design dan Deskripsi Objek UI #5 <Pencarian Informasi Produk>	24
3.1.5.3 Identifikasi Object Baru & Tipe Kelas #5 <Pencarian Informasi Produk>	26
3.1.5.4 Sequence Diagram #5 <Pencarian Informasi Produk>	27
3.1.6 Use Case #6 <Melihat informasi produk>	27
3.1.6.1 Use Case Scenario #6 <Melihat informasi produk>	27
3.1.6.2 UI Design dan Deskripsi Objek UI 6 <Melihat informasi produk>	27

3.1.6.3 Identifikasi Object dan Tipe nya #6 <Melihat informasi produk>	29
3.1.6.4 Sequence Diagram #6 <Melihat informasi produk>	30
3.1.7 Use Case #7 <Booking Hotel>	30
3.1.7.1 Use Case Scenario #7 <Booking Hotel>	30
3.1.7.2 UI Design dan Deskripsi Objek UI #7 <Booking Hotel>	30
3.1.7.3 Identifikasi Object dan Tipe nya #7 <Booking Hotel>	33
3.1.7.4 Sequence Diagram #7 <Booking Hotel>	34
3.1.8 Use Case #8<Melakukan Pembayaran>	34
3.1.8.1 Use Case Scenario #8<Melakukan Pembayaran>	34
3.1.8.2 UI Design dan Deskripsi Objek UI #8<Melakukan Pembayaran>	34
3.1.8.3 Identifikasi Object dan Tipe nya #8<Melakukan Pembayaran>	36
3.1.8.4 Sequence Diagram #8<Melakukan Pembayaran>	36
3.1.9 Use Case #9<Memverifikasi Pembayaran>	36
3.1.9.1 Use Case Scenario #9<Memverifikasi Pembayaran>	36
3.1.9.2 UI Design dan Deskripsi Objek UI #9<Memverifikasi Pembayaran>	37
3.1.9.3 Identifikasi Object dan Tipe nya #9<Memverifikasi Pembayaran>	38
3.1.9.4 Sequence Diagram #9<Memverifikasi Pembayaran>	38
3.2 Diagram Kelas Keseluruhan	39
3.3 Perancangan Data / Basis Data	41
3.4 Perancangan Algoritma dan/atau Query	42
4. Matriks Keruntutan (Requirement Traceability Matrix)	54

1. Pendahuluan

1.1 Tujuan Penulisan Dokumen

Tujuan penulisan dokumen ini adalah untuk menjelaskan mengenai deskripsi perancangan perangkat lunak yang akan dibuat atau dikembangkan, baik berupa penjelasan secara umum maupun secara detail. Pada dokumen DPPL ini, perangkat lunak yang dibuat adalah aplikasi Travellight, yaitu aplikasi yang bertujuan untuk mempermudah customer dalam memesan hotel saat bepergian. Dokumen ini digunakan oleh pengembang perangkat lunak sebagai acuan teknis untuk pengembangan perangkat lunak ini. Dengan adanya dokumen ini, diharapkan pengembangan perangkat lunak akan menjadi lebih terarah.

1.2 Lingkup Masalah

Aplikasi “Travellight” adalah perangkat lunak berupa sistem informasi yang digunakan oleh customer, pemilik hotel, dan administrator. Perangkat lunak ini digunakan untuk membantu customer dalam memesan kamar hotel. Tujuan produk ini untuk mempermudah customer dalam bepergian. Sasaran aplikasi ini kepada masyarakat yang berusia 17 - 50 tahun.

1.3 Definisi dan Istilah

Berikut adalah daftar definisi dan istilah penting yang digunakan dalam dokumen DPPL ini:

1. DPPL

DPPL adalah singkatan dari Deskripsi Perancangan Perangkat Lunak yang merupakan dokumen hasil analisis yang berisi deskripsi perancangan perangkat lunak.

2. SKPL-F-TVL-xx

SKPL-F-TVL-xx merupakan kode yang digunakan untuk merepresentasikan kebutuhan fungsional perangkat lunak yang akan dikembangkan. Kode xx merupakan nomor kebutuhan.

1.4 Referensi

SKPL Travellight Kelompok 7 IF-44-02

2 Perancangan Global

2.1 Rencana Lingkungan Implementasi

Untuk implementasi dari aplikasi Travellight yaitu aplikasi ini berbasis website yang dimana aplikasi ini dapat diakses baik dari mobile maupun desktop. Dalam pengembangan website tentunya membutuhkan perangkat lunak dan bahasa pemrograman pendukung seperti HTML, CSS, JavaScript, SQL, PHP.

Kode pemrograman HTML digunakan untuk membuat kerangka bentuk website, kode pemrograman CSS digunakan untuk menghias dan mempercantik halaman website, kode pemrograman JavaScript digunakan agar website lebih dinamis, kode pemrograman SQL digunakan untuk akses dan kontrol database, dan kode pemrograman PHP digunakan untuk pemrosesan yang terjadi pada website.

2.2 Deskripsi Arsitektur Perangkat Lunak

No	Nama Komponen	Keterangan
1	database server	sebagai aplikasi tempat untuk menyimpan data hotel, user, admin, dll
2	aplikasi Travellight	aplikasi booking hotel
3	browser	untuk mengakses aplikasi Travellight
4	host	sebagai penyedia layanan website baik itu domain, server, database, dll

3 Perancangan Rinci

3.1 Realisasi Use Case

Berisi TABEL USE CASE sebagai berikut :

No	Nama Use Case	Deskripsi Use Case
#1	Pendaftaran Akun	Merupakan proses untuk pendaftaran akun oleh customer atau pemilik hotel
#2	Login	Merupakan proses untuk login user baik customer, pemilik hotel, maupun administrator
#3	Update profile	Mengubah data profil customer
#4	Manajemen produk	Merupakan proses untuk menambahkan produk baru, mengubah informasi produk, atau menghapus produk yang dilakukan oleh pemilik hotel
#5	Pencarian Informasi Produk	Merupakan proses untuk pencarian informasi produk oleh customer
#6	Melihat Informasi Produk	Merupakan proses untuk melihat informasi produk oleh customer
#7	Booking Hotel	Merupakan proses untuk memesan hotel yang diinginkan oleh customer
#8	Melakukan Pembayaran	Merupakan proses untuk memverifikasi pembayaran customer terhadap suatu pesanan yang dilakukan oleh administrator
#9	Memverifikasi Pembayaran	Merupakan proses untuk memverifikasi pembayaran customer terhadap suatu pesanan yang dilakukan oleh administrator

3.1.1 Use Case #1 <Pendaftaran Akun>

3.1.1.1 Use Case Scenario #1 <Pendaftaran Akun>

Skenario Use Case #1 :

i. Pre-Condition

Customer / pemilik hotel berada pada halaman login website Travellight dan belum memiliki akun

ii. Use Case Description

a. Primary Flow

1. Customer / pemilik hotel memilih menu Daftar
2. Customer / pemilik hotel akan melihat halaman form pendaftaran akun
3. Customer / pemilik hotel memasukkan data pendaftaran akun (Nama, nomor HP, alamat email, alamat tempat tinggal, foto profil, username, password)
4. Customer / pemilik hotel menunggu data pendaftaran akun divalidasi oleh controller SignUp saat customer/pemilik hotel klik tombol sign up
5. Customer / pemilik hotel menunggu data pendaftaran akun disimpan pada database customer / pemilik hotel
6. Customer / pemilik melihat tampilan page login

b. Alternative Flow

1. Pada tahap 3, customer / pemilik hotel memasukkan data pendaftaran yang tidak valid

2. Customer / pemilik hotel melihat tampilan pesan data tidak valid dan kembali menjalankan tahap 2
- iii. Post-Condition
 - Entri untuk informasi customer / pemilik hotel tersimpan dalam database
 - Tercipta akun baru untuk customer / pemilik hotel

3.1.1.2 UI Design dan Deskripsi Objek UI #1 <Pendaftaran Akun>

Tabel Deskripsi Objek UI

ID. LAYAR	NAMA LAYAR	DESKRIPSI
L01	Page Login Customer	Merupakan halaman yang tampil setelah customer mengklik tombol As Customer di Landing Page
L02	Page Login Pemilik Hotel	Merupakan halaman yang tampil setelah pemilik hotel mengklik tombol As Hotel Owner di Landing Page
L03	Page Form Registrasi Customer	Merupakan halaman dimana customer mengisi data registrasi
L04	Page Form Registrasi Pemilik Hotel	Merupakan halaman dimana pemilik hotel mengisi data registrasi

Antarmuka L01:

TRAVELLIGHT

Login As Customer

Username

txt

Password

txt

LOGIN

Belum punya akun? [Daftar](#)

Page Login Customer

Id_Objek	JENIS	LABEL*	Keterangan**
B01	Button	Login	Jika diklik, akan mengaktifkan Function yang menampilkan dashboard customer
LK01	Link	Daftar	Jika diklik akan mengaktifkan Function yang menampilkan Page Form Registrasi Customer
TF01	Text Field	Username	Tempat customer memasukkan username
TF02	Text Field	Password	Tempat customer memasukkan password

Antarmuka L02:

TRAVELLIGHT

Login As Hotel Owner

Username

txt

Password

txt

LOGIN

Belum punya akun? [Daftar](#)

Page Login Pemilik Hotel

Id_Objek	JENIS	LABEL*	Keterangan**
B02	Button	Login	Jika diklik, akan mengaktifkan Function yang menampilkan dashboard pemilik hotel
LK02	Link	Daftar	Jika diklik akan mengaktifkan Function yang menampilkan Page Form Registrasi Pemilik Hotel
TF03	Text Field	Username	Tempat pemilik hotel memasukkan username
TF04	Text Field	Password	Tempat pemilik hotel memasukkan password

Antarmuka L03:

Travellight

Sign Up as Customer

Name

Phone Number

Email Address

Address

Username

Password

Confirm Password

Sign Up

Page Form Registrasi Customer

Id_Objek	JENIS	LABEL*	Keterangan**
B03	Button	Sign Up	Jika diklik, akan mengaktifkan Function save customer
TF05	Text Field	Name	Tempat customer mengisi nama
TF06	Text Field	Phone Number	Tempat customer mengisi nomor HP
TF07	Text Field	Email Address	Tempat customer mengisi alamat email
TF08	Text Field	Address	Tempat customer mengisi alamat tempat tinggal

Id_Objek	JENIS	LABEL*	Keterangan**
TF09	Text Field	Username	Tempat customer mengisi username
TF10	Text Field	Password	Tempat customer mengisi password
TF11	Text Field	Confirm Password	Tempat customer mengisi password untuk konfirmasi

Antarmuka L04:

Page Form Registrasi Pemilik Hotel

Id_Objek	JENIS	LABEL*	Keterangan**
B04	Button	Sign Up	Jika di klik, akan mengaktifkan Function save owner
TF12	Text Field	Name	Tempat pemilik hotel mengisi nama
TF13	Text Field	Phone Number	Tempat pemilik hotel mengisi nomor HP
TF14	Text Field	Email Address	Tempat pemilik hotel mengisi alamat email
TF15	Text Field	Bank Account	Tempat pemilik hotel mengisi nomor rekening bank
TF16	Text Field	Username	Tempat pemilik hotel mengisi username
TF17	Text Field	Password	Tempat pemilik hotel mengisi password
TF18	Text Field	Confirm Password	Tempat pemilik hotel mengisi password untuk konfirmasi

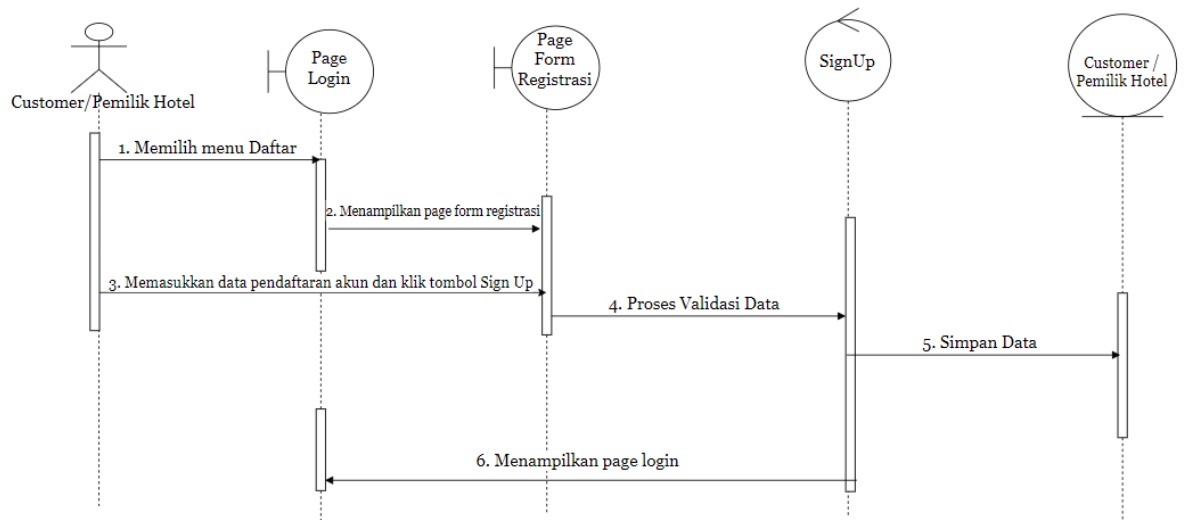
3.1.1.3 Identifikasi Object dan Tipe nya #1 <Pendaftaran Akun>

TABEL OBJECT PERANCANGAN

No	Nama Object Baru	Jenis / Tipe Kelas
1	Page Login Customer	Boundary(Interface)
2	Page Login Pemilik Hotel	Boundary(Interface)
3	Page Form Registrasi Customer	Boundary(Interface)
4	Page Form Registrasi Pemilik Hotel	Boundary(Interface)
3	SignUp	Controller
4	Customer	Entity(Database)
5	Pemilik Hotel	Entity(Database)

*Tipe kelas diisi dengan Boundary(Interface), Entity(Database), Controller

3.1.1.4 Sequence Diagram #1 <Pendaftaran Akun>



3.1.2 Use Case #2 <Login>

3.1.2.1 Use Case Scenario #2 <Login>

Skenario Use Case #2 :

i. Pre-Condition

Customer / pemilik hotel telah mendaftar akun dan berada pada page login.
Administrator telah berada pada page login

ii. Use Case Description

a. Primary Flow

1. Customer / pemilik hotel / administrator memasukkan username dan password.
2. Customer / pemilik hotel / administrator menunggu username dan password akun divalidasi oleh controller Login saat customer/pemilik hotel / administrator klik tombol Login
3. Customer / pemilik hotel / administrator menunggu proses validasi username, password dari database customer / pemilik hotel/ administrator

b. Alternative Flow

1. Pada tahap 1, customer / pemilik hotel memasukan username dan password yang salah.
2. Customer / pemilik hotel melihat pesan username dan password salah dan kembali menjalankan tahap 1.

iii. Post-Condition

User berhasil login

3.1.2.2 UI Design dan Deskripsi Objek UI #2 <Login>

Tabel Deskripsi Objek UI

ID. LAYAR	NAMA LAYAR	DESKRIPSI
L01	Page Login Customer	Merupakan halaman yang tampil setelah customer mengklik tombol As Customer di Landing Page

L02	Page Login Pemilik Hotel	Merupakan halaman yang tampil setelah pemilik hotel mengklik tombol As Hotel Owner di Landing Page
L05	Page Login Administrator	Merupakan halaman yang tampil saat administrator mengakses website Travellight

Antarmuka L01:

TRAVELLIGHT

Login As Customer

Username

Password

LOGIN

Belum punya akun? [Daftar](#)

Page Login Customer

Id_Objek	JENIS	LABEL*	Keterangan**
B01	Button	Login	Jika diklik, akan mengaktifkan Function yang menampilkan dashboard customer
LK01	Link	Daftar	Jika diklik akan mengaktifkan Function yang menampilkan Page Form Registrasi Customer
TF01	Text Field	Username	Tempat customer memasukkan username
TF02	Text Field	Password	Tempat customer memasukkan password

Antarmuka L02:

TRAVELLIGHT

Login As Hotel Owner

Username

Password

LOGIN

Belum punya akun? [Daftar](#)

Page Login Pemilik Hotel

Id_Objek	JENIS	LABEL*	Keterangan**
B02	Button	Login	Jika diklik, akan mengaktifkan Function yang menampilkan dashboard pemilik hotel

Id_Objek	JENIS	LABEL*	Keterangan**
LK02	Link	Daftar	Jika diklik akan mengaktifkan Function yang menampilkan Page Form Registrasi Pemilik Hotel
TF03	Text Field	Username	Tempat pemilik hotel memasukkan username
TF04	Text Field	Password	Tempat pemilik hotel memasukkan password

Antarmuka L05:

TRAVELLIGHT

Login As Administrator

Username

Password

LOGIN

Page Login Administrator

Id_Objek	JENIS	LABEL*	Keterangan**
B05	Button	Login	Jika diklik, akan mengaktifkan Function yang menampilkan dashboard administrator
TF19	Text Field	Username	Tempat administrator memasukkan username
TF20	Text Field	Password	Tempat administrator memasukkan password

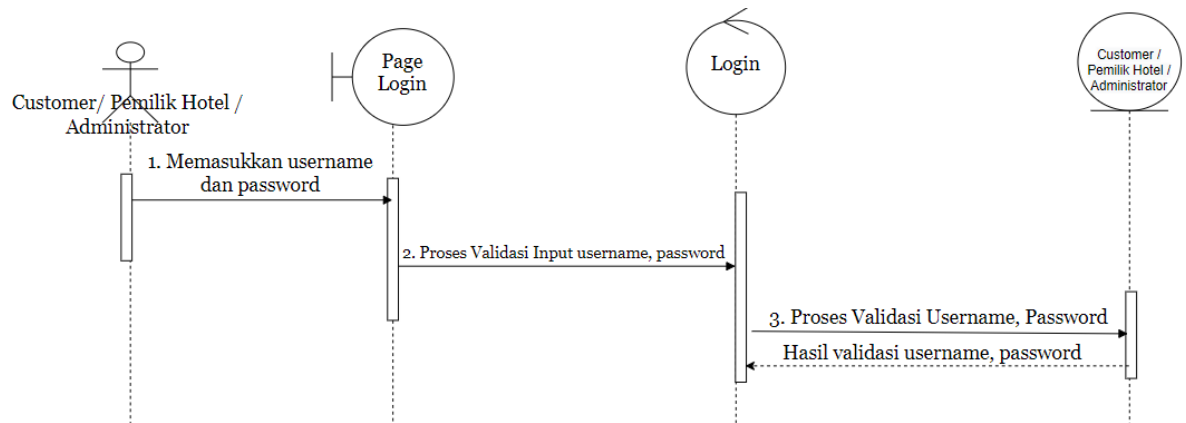
3.1.2.3 Identifikasi Object Baru & Tipe Kelas #2 <Login>

TABEL OBJECT PERANCANGAN

No	Nama Object Baru	Jenis / Tipe Kelas
1	Page Login Customer	Boundary(Interface)
2	Page Login Pemilik Hotel	Boundary(Interface)
3	Page Login Administrator	Boundary(Interface)
4	Login	Controller
5	Customer	Entity(Database)
6	Pemilik Hotel	Entity(Database)
7	Administrator	Entity(Database)

*Tipe kelas seperti Boundary(Interface), Entity(Database), Controller

3.1.2.4 Sequence Diagram #2 <Login>



3.1.3 Use Case #3 <Update Profile>

3.1.3.1 Use Case Scenario #3<Update Profile>

Skenario Use Case #3 :

- i. Pre-Condition
 - Customer telah memiliki akun dan berada pada dashboard
- ii. Use Case Description
 - a. Primary Flow
 1. Customer memilih menu edit profile
 2. Customer melihat page edit profile
 3. Customer memasukkan data profil baru yang ingin diubah
 4. Customer menunggu data profil baru divalidasi oleh controller Profile saat customer klik tombol Edit
 5. Customer menunggu data pendaftaran akun di-update pada database customer
 - b. Alternative Flow
 1. Pada tahap 3, customer mengindikasikan bahwa ia memasukkan informasi data profil baru yang salah.
 2. Kembali menjalankan tahap 2.
 - c. Alternative Flow
 1. Pada tahap 4, terdeteksi terdapat data yang tidak valid.
 2. Ditampilkan pesan data tidak valid dan kembali menjalankan tahap 2.
- iii. Post-Condition
 - Data profil dari customer telah diperbaharui pada database customer

3.1.3.2 UI Design dan Deskripsi Objek UI #3 <Edit Profile>

Tabel Deskripsi Objek UI

ID. LAYAR	NAMA LAYAR	DESKRIPSI
L06	Dashboard Customer	Merupakan halaman yang tampil saat customer berhasil login
L07	Page Edit Profile	Merupakan halaman yang tampil saat sudah klik button edit profile di dashboard

Antarmuka L06:

Page Dashboard Customer

Id_Objek	JENIS	LABEL*	Keterangan**
DM01	Dropdown Menu	Nama Customer	Jika diklik, akan menampilkan 3 menu yaitu Edit Profile, List Pesanan, dan Logout
DI01	Dropdown Item	Edit Profile	Jika diklik, akan mengaktifkan Function yang menampilkan Page Edit Profile
DI02	Dropdown Item	List Pesanan	Jika diklik, akan mengaktifkan Function yang menampilkan Page List Pesanan
DI03	Dropdown Item	Logout	Jika diklik, akan mengaktifkan Function logout
TF21	Text Field	Lokasi Pencarian	Tempat customer memasukkan keyword terkait lokasi dari hotel yang ingin dicari
TF22	Text Field	Banyak Kamar	Tempat customer memasukkan banyak kamar yang ingin dipesan saat pencarian hotel
DP01	Date Picker	Waktu Check In	Tempat customer memasukkan waktu check in pemesanan saat pencarian hotel
DP02	Date Picker	Waktu Check Out	Tempat customer memasukkan waktu check out pemesanan saat pencarian hotel
C01	Card	Hotel	Jika diklik, akan mengaktifkan Function yang menampilkan info hotel lebih lengkap
B06	Button	Search	Jika diklik, akan mengaktifkan Function cari

Antarmuka L07

Edit Profile

Name

.....

Phone Number

.....

Email Address

.....

Address

.....

Username

.....

Edit

Back

Page Edit Profile

Id_Objek	JENIS	LABEL*	Keterangan**
TF23	Text Field	Name	Tempat customer memasukkan nama
TF24	Text Field	Username	Tempat customer memasukkan username
TF25	Text Field	Address	Tempat customer memasukkan alamat tempat tinggal
TF26	Text Field	Email Address	Tempat customer memasukkan alamat email
TF27	Text Field	Phone Number	Tempat customer memasukkan No Hp
B07	Button	Edit	Jika diklik akan mengaktifkan function update profile
B08	Button	Back	Jika diklik, akan mengaktifkan Function kembali ke halaman sebelumnya

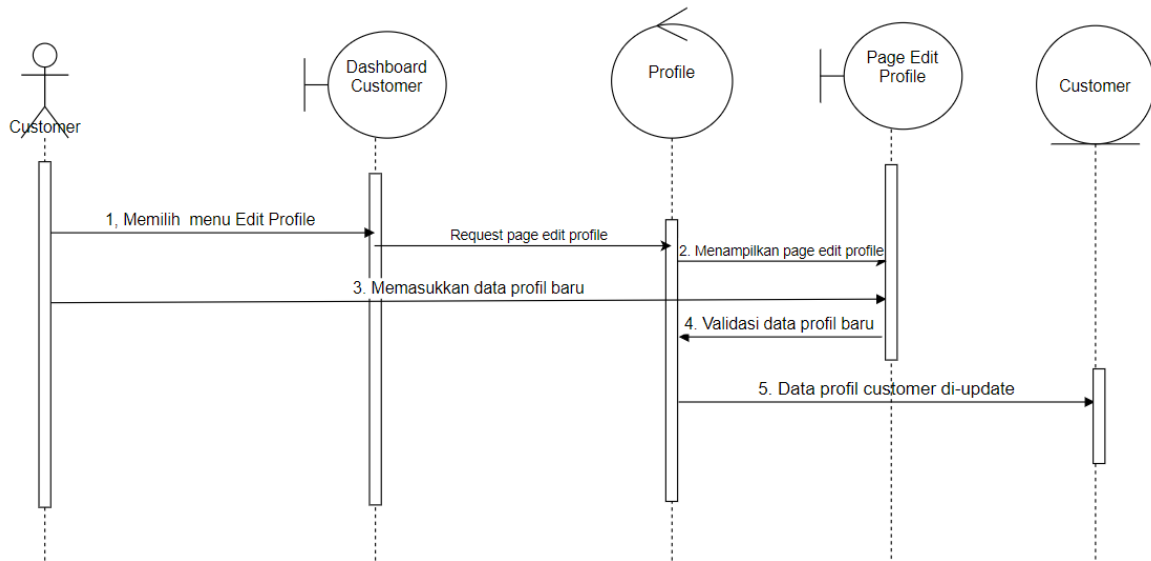
3.1.3.3 Identifikasi Object Baru & Tipe Kelas #3 <Update Profile>

TABEL OBJECT PERANCANGAN

No	Nama Object Baru	Jenis / Tipe Kelas
1	Dashboard Customer	Boundary(Interface)
2	Page Edit Profile	Boundary(Interface)
3	Profile	Controller
4	Customer	Entity(Database)

*Tipe kelas seperti Boundary(Interface), Entity(Database), Controller

3.1.3.4 Sequence Diagram #3 <Update Profile>



3.1.4 Use Case #4 <Manajemen Produk>

3.1.4.1 Use Case Scenario #4<Manajemen Produk>

Skenario Use Case #4 :

- i. Pre-Condition
 - Pemilik hotel melihat tampilan dashboard pemilik hotel
- ii. Use Case Description
 - a. Alternative Flow 1
 1. Pemilik hotel memilih menu "Add Product"
 2. Pemilik hotel melihat page add product
 3. Pemilik hotel mengisi data produk (nama hotel, lokasi hotel, deskripsi hotel, gambar hotel)
 4. Pemilik hotel menunggu data produk divalidasi oleh controller Product saat pemilik hotel klik tombol Add
 5. Pemilik hotel menunggu data produk disimpan pada database hotel
 - b. Alternative Flow 2
 1. Pemilik hotel mengklik tombol "Edit"
 2. Pemilik hotel melihat page edit product
 3. Pemilik hotel mengisi data produk baru
 4. Pemilik hotel menunggu data produk baru divalidasi oleh controller Product saat pemilik hotel klik tombol Edit
 5. Pemilik hotel menunggu data produk di-update pada database hotel
 - c. Alternative Flow 3
 1. Pemilik hotel mengklik tombol "Delete"
 2. Pemilik hotel mengklik Tombol "OK" pada dialog box konfirmasi hapus produk
 3. Pemilik hotel menunggu data produk dihapus dari database hotel
 - d. Alternative Flow 4
 1. Pemilik hotel memilih Tombol "tambah kamar"
 2. Pemilik hotel melihat page tambah kamar
 3. Pemilik hotel mengisi data kamar (jenis kamar, harga, stok, dan status)
 4. Pemilik hotel menunggu data kamar divalidasi oleh controller Room saat pemilik hotel klik tombol Tambah
 5. Pemilik hotel menunggu data kamar disimpan pada database kamar
 - e. Alternative Flow 5
 1. Pemilik hotel mengklik tombol "Edit (kamar)"
 2. Pemilik hotel melihat page edit kamar

3. Pemilik hotel mengisi data kamar baru
4. Pemilik hotel menunggu data kamar baru divalidasi oleh controller Room saat pemilik hotel klik tombol “Edit”
5. Pemilik hotel menunggu data kamar di-update pada database kamar
- f. Alternative Flow 6
 1. Pemilik hotel mengklik tombol “Delete (kamar)”
 2. Pemilik hotel mengklik Tombol “OK” pada dialog box konfirmasi hapus kamar
 3. Pemilik hotel menunggu data kamar dihapus dari database kamar

iii. Post-Condition

Data produk atau kamar hasil perubahan telah berhasil disimpan di database

3.1.4.2 UI Design dan Deskripsi Objek UI #4 <Manajemen Produk>

Tabel Deskripsi Objek UI

ID. LAYAR	NAMA LAYAR	DESKRIPSI
L08	Dashboard Pemilik Hotel	Merupakan halaman yang tampil saat pemilik hotel berhasil login
L09	Page Add Product	Merupakan halaman untuk mengisi data terkait penambahan produk
L10	Page Edit Product	Merupakan halaman untuk mengisi data terkait edit produk
L11	Page Tambah Kamar	Merupakan halaman untuk mengisi data terkait penambahan kamar
L12	Page Edit Kamar	Merupakan halaman untuk mengisi data terkait edit kamar

Antarmuka L08:

Page Dashboard Pemilik Hotel

Id_Objek	JENIS	LABEL*	Keterangan**
LK03	Link	Add Product	Jika diklik, akan mengaktifkan Function yang menampilkan page add product
LK04	Link	List Pesanan	Jika diklik akan mengaktifkan Function yang menampilkan page list pesanan
LK05	Link	Logout	Jika diklik akan mengaktifkan Function logout

Id_Objek	JENIS	LABEL*	Keterangan**
B09	Button	Tambah Kamar	Jika diklik akan mengaktifkan Function yang menampilkan page tambah kamar
B10	Button	Edit	Jika diklik akan mengaktifkan Function yang menampilkan page edit product
B11	Button	Delete	Jika diklik akan mengaktifkan Function delete product
B12	Button	Edit (kamar)	Jika diklik akan mengaktifkan Function yang menampilkan page edit kamar
B13	Button	Delete (kamar)	Jika diklik akan mengaktifkan Function delete room

Antarmuka L09:

Page Add Product

Id_Objek	JENIS	LABEL*	Keterangan**
LK03	Link	Add Product	Jika diklik, akan mengaktifkan Function yang menampilkan page add product
LK04	Link	List Pesanan	Jika diklik akan mengaktifkan Function yang menampilkan page list pesanan
LK05	Link	Logout	Jika diklik akan mengaktifkan Function logout
TF28	Text Field	Name	Tempat pemilik hotel mengisi nama hotel
TF29	Text Field	Location	Tempat pemilik hotel mengisi lokasi hotel
TA01	Text Area	Description	Tempat pemilik hotel mengisi deskripsi hotel
FI01	File Input	Add Hotel Picture	Tempat pemilik hotel mengupload gambar hotel
B14	Button	Add	Jika diklik akan mengaktifkan Function save product
B08	Button	Back	Jika diklik, akan mengaktifkan Function kembali ke halaman sebelumnya

Antarmuka L10:

Travellight
Edit Hotel
Add Product List Pesanan Logout

Name

.....

Location

.....

Description

.....

Add Hotel Picture

Choose File

Edit

Back

Page Edit Product

Id_Objek	JENIS	LABEL*	Keterangan**
LK03	Link	Add Product	Jika diklik, akan mengaktifkan Function yang menampilkan page add product
LK04	Link	List Pesanan	Jika diklik akan mengaktifkan Function yang menampilkan page list pesanan
LK05	Link	Logout	Jika diklik akan mengaktifkan Function logout
TF30	Text Field	Name	Tempat pemilik hotel mengisi nama hotel
TF31	Text Field	Location	Tempat pemilik hotel mengisi lokasi hotel
TA02	Text Area	Description	Tempat pemilik hotel mengisi deskripsi hotel
FI02	File Input	Add Hotel Picture	Tempat pemilik hotel mengupload gambar hotel
B15	Button	Edit	Jika diklik akan mengaktifkan Function edit product
B08	Button	Back	Jika diklik, akan mengaktifkan Function kembali ke halaman sebelumnya

Antarmuka L11:

Travellight
Add Kamar
Add Product
List Pesanan
Logout

Jenis Kamar

Harga

Stok

Status

Tambah

Back

Page Tambah Kamar

Id_Objek	JENIS	LABEL*	Keterangan**
LK03	Link	Add Product	Jika diklik, akan mengaktifkan Function yang menampilkan page add product
LK04	Link	List Pesanan	Jika diklik akan mengaktifkan Function yang menampilkan page list pesanan
LK05	Link	Logout	Jika diklik akan mengaktifkan Function logout
TF32	Text Field	Jenis Kamar	Tempat pemilik hotel mengisi jenis kamar
TF33	Text Field	Harga	Tempat pemilik hotel mengisi harga kamar
TF34	Text Field	Stok	Tempat pemilik hotel mengisi stok Kamar
TF35	Text Field	Status	Tempat pemilik hotel mengisi status Kamar
B16	Button	Tambah	Jika diklik akan mengaktifkan Function save room
B08	Button	Back	Jika diklik, akan mengaktifkan Function kembali ke halaman sebelumnya

Antarmuka L12:

Travellight
Edit Kamar
Add Product List Pesanan Logout

Jenis Kamar

Harga

Stok

Status

Edit

Back

Page Edit Kamar

Id_Objek	JENIS	LABEL*	Keterangan**
LK03	Link	Add Product	Jika diklik, akan mengaktifkan Function yang menampilkan page add product
LK04	Link	List Pesanan	Jika diklik akan mengaktifkan Function yang menampilkan page list pesanan
LK05	Link	Logout	Jika diklik akan mengaktifkan Function logout
TF36	Text Field	Jenis Kamar	Tempat pemilik hotel mengisi jenis kamar
TF37	Text Field	Harga	Tempat pemilik hotel mengisi harga kamar
TF38	Text Field	Stok	Tempat pemilik hotel mengisi stok Kamar
TF39	Text Field	Status	Tempat pemilik hotel mengisi status Kamar
B17	Button	Edit	Jika diklik akan mengaktifkan Function edit room
B08	Button	Back	Jika diklik, akan mengaktifkan Function kembali ke halaman sebelumnya

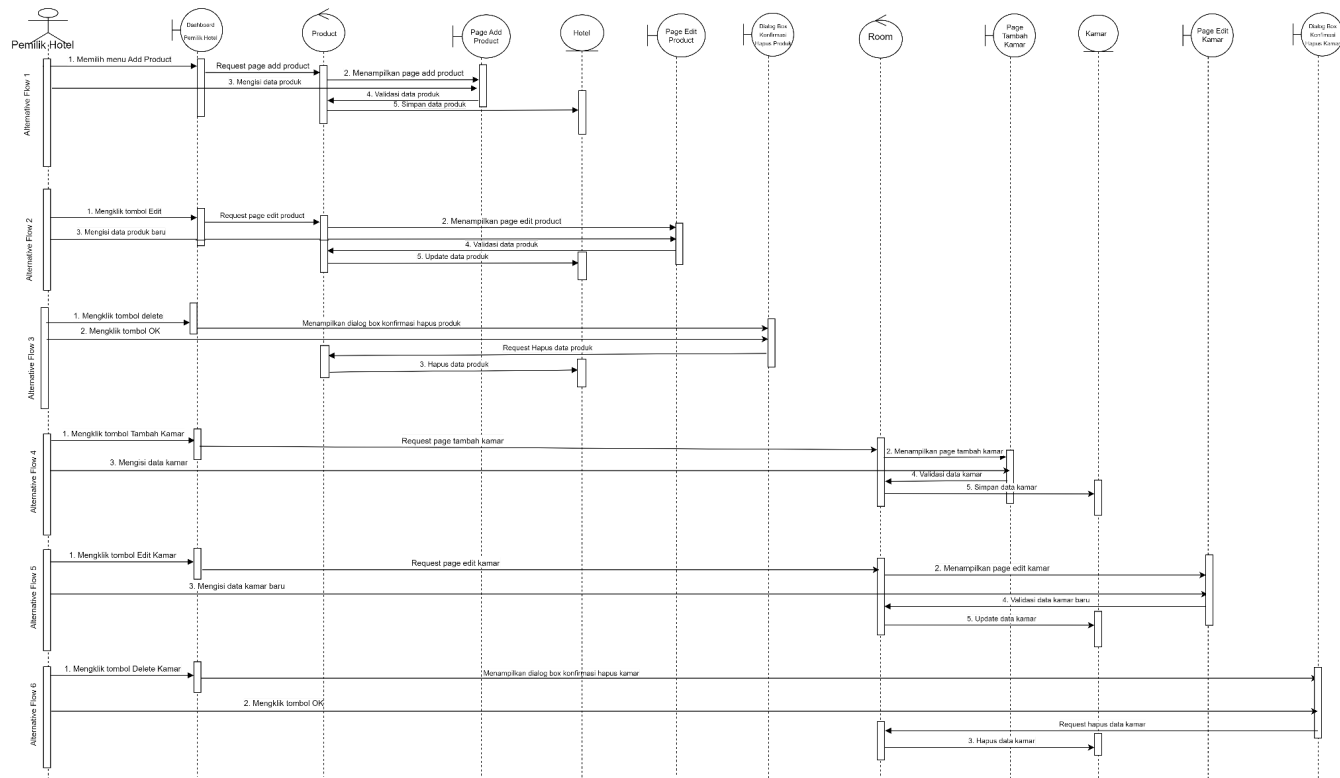
3.1.4.3 Identifikasi Object Baru & Tipe Kelas #4 <Manajemen Produk>

TABEL OBJECT PERANCANGAN

No	Nama Object Baru	Jenis / Tipe Kelas
1	Dashboard Pemilik Hotel	Boundary(Interface)
2	Page Add Product	Boundary(Interface)
3	Page Edit Product	Boundary(Interface)
4	Page Tambah Kamar	Boundary(Interface)
5	Page Edit Kamar	Boundary(Interface)
6	Room	Controller
7	Product	Controller
8	Kamar	Entity(Database)
9	Hotel	Entity(Database)

*Tipe kelas seperti Boundary(Interface), Entity(Database), Controller

3.1.4.4 Sequence Diagram #4 <Manajemen Produk>



3.1.5 Use Case #5 <Pencarian Informasi Produk>

3.1.5.1 Use Case Scenario #5 <Pencarian Informasi Produk>

Skenario Use Case #4 :

- i. Pre-Condition
 - Customer berada pada dashboard website Travellight
- ii. Use Case Description
 - a. Primary Flow
 1. Customer memasukkan nama daerah, mengisi tanggal check in, check out dan banyaknya kamar
 2. Customer menunggu data yang diinputkan divalidasi oleh controller Searching saat customer klik tombol “Search”
 3. Customer menunggu data produk diambil dari database hotel
 4. Customer melihat produk hasil pencarian.
- iii. Post-Condition
 - Ditampilkan informasi produk hasil pencarian

3.1.5.2 UI Design dan Deskripsi Objek UI #5 <Pencarian Informasi Produk>


Tabel Deskripsi Objek UI


ID. LAYAR	NAMA LAYAR	DESKRIPSI
L06	Dashboard Customer	Merupakan halaman yang tampil saat customer berhasil login
L13	Page Hasil pencarian	Merupakan halaman yang tampil setelah melakukan pencarian produk

Antarmuka L06:

Travellight
Selamat Datang
Nama Customer ↓

Mau Ke mana nih ??

Waktu Check in


Waktu Check out


Banyaknya Kamar
.....

Search

Hasil Rekomendasi

Gambar

Nama Hotel
Deskripsi
Produk

Gambar

Nama Hotel
Deskripsi
Produk

Gambar

Nama Hotel
Deskripsi
Produk

Page Dashboard Customer

Id_Objek	JENIS	LABEL*	Keterangan**
DM01	Dropdown Menu	Nama Customer	Jika diklik, akan menampilkan 3 menu yaitu Edit Profile, List Pesanan, dan Logout
DI01	Dropdown Item	Edit Profile	Jika diklik, akan mengaktifkan Function yang menampilkan Page Edit Profile
DI02	Dropdown Item	List Pesanan	Jika diklik, akan mengaktifkan Function yang menampilkan Page List Pesanan
DI03	Dropdown Item	Logout	Jika diklik, akan mengaktifkan Function logout
TF21	Text Field	Lokasi Pencarian	Tempat customer memasukkan keyword terkait lokasi dari hotel yang ingin dicari
TF22	Text Field	Banyak Kamar	Tempat customer memasukkan banyak kamar yang ingin dipesan saat pencarian hotel
DP01	Date Picker	Waktu Check In	Tempat customer memasukkan waktu check in pemesanan saat pencarian hotel
DP02	Date Picker	Waktu Check Out	Tempat customer memasukkan waktu check out pemesanan saat pencarian hotel
C01	Card	Hotel	Jika diklik, akan mengaktifkan Function yang menampilkan info hotel lebih lengkap
B06	Button	Search	Jika diklik, akan mengaktifkan Function cari

Antarmuka L13:

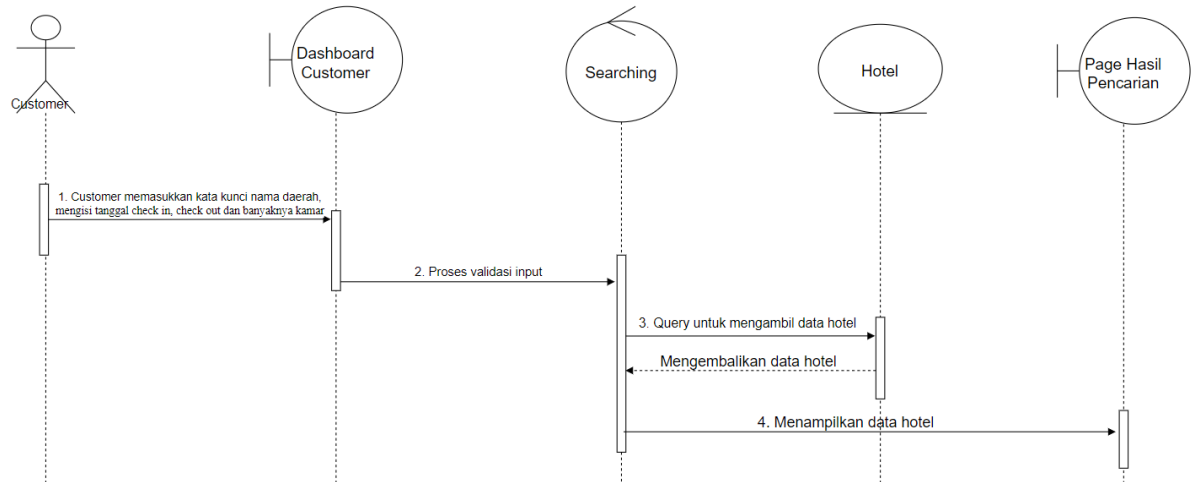
Prodi S1 Informatika Tel-U

DPPL-07

Halaman 25 dari 54

Template dokumen ini dan informasi yang dimilikinya adalah milik Prodi S1 Informatika Tel-U dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Prodi S1 Informatika Tel-U.

3.1.5.4 Sequence Diagram #5 <Pencarian Informasi Produk>



3.1.6 Use Case #6 <Melihat informasi produk>

3.1.6.1 Use Case Scenario #6 <Melihat informasi produk>

Skenario Use Case #1 :

- i. Pre-Condition
Customer berada pada dashboard website Travellight
- ii. Use Case Description
 - a. Primary Flow
 1. Customer mengklik card hotel yang ingin dilihat
 2. Customer menunggu data hotel dan kamar diambil dari database hotel dan kamar
 3. Customer melihat informasi data hotel (nama hotel, lokasi hotel, gambar hotel, deskripsi hotel, kamar tersedia pada hotel tersebut)
- iii. Post-Condition
Informasi Produk ditampilkan

3.1.6.2 UI Design dan Deskripsi Objek UI 6 <Melihat informasi produk>


Tabel Deskripsi Objek UI


ID. LAYAR	NAMA LAYAR	DESKRIPSI
L06	Dashboard Customer	Merupakan halaman yang tampil saat customer berhasil login
L14	Page Informasi Hotel	Merupakan halaman untuk melihat informasi hotel

Antarmuka L06:

Travellight
Selamat Datang
Nama Customer ↓

Mau Ke mana nih ??

Waktu Check in


Waktu Check out


Banyaknya Kamar
.....

Search

Hasil Rekomendasi

Gambar

Nama Hotel
Deskripsi
Produk

Gambar

Nama Hotel
Deskripsi
Produk

Gambar

Nama Hotel
Deskripsi
Produk

Page Dashboard Customer

Id_Objek	JENIS	LABEL*	Keterangan**
DM01	Dropdown Menu	Nama Customer	Jika diklik, akan menampilkan 3 menu yaitu Edit Profile, List Pesanan, dan Logout
DI01	Dropdown Item	Edit Profile	Jika diklik, akan mengaktifkan Function yang menampilkan Page Edit Profile
DI02	Dropdown Item	List Pesanan	Jika diklik, akan mengaktifkan Function yang menampilkan Page List Pesanan
DI03	Dropdown Item	Logout	Jika diklik, akan mengaktifkan Function logout
TF21	Text Field	Lokasi Pencarian	Tempat customer memasukkan keyword terkait lokasi dari hotel yang ingin dicari
TF22	Text Field	Banyak Kamar	Tempat customer memasukkan banyak kamar yang ingin dipesan saat pencarian hotel
DP01	Date Picker	Waktu Check In	Tempat customer memasukkan waktu check in pemesanan saat pencarian hotel
DP02	Date Picker	Waktu Check Out	Tempat customer memasukkan waktu check out pemesanan saat pencarian hotel
C01	Card	Hotel	Jika diklik, akan mengaktifkan Function yang menampilkan info hotel lebih lengkap
B06	Button	Search	Jika diklik, akan mengaktifkan Function cari

Antarmuka L14:

Prodi S1 Informatika Tel-U

DPPL-07

Halaman 28 dari 54

Template dokumen ini dan informasi yang dimilikinya adalah milik Prodi S1 Informatika Tel-U dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Prodi S1 Informatika Tel-U.

Travellight
Nama Customer ▼

NAMA HOTEL

alamat

Gambar

Fasilitas

Fasilitas

Fasilitas

Fasilitas

Kamar Tersedia

Jenis kamar

Harga

Sisa kamar

....

....

....

Pesan

Back

Page Infomasi Hotel

Id_Objek	JENIS	LABEL*	Keterangan**
DM01	Dropdown Menu	Nama Customer	Jika diklik, akan menampilkan 3 menu yaitu Edit Profile, List Pesanan, dan Logout
DI01	Dropdown Item	Edit Profile	Jika diklik, akan mengaktifkan Function yang menampilkan Page Edit Profile
DI02	Dropdown Item	List Pesanan	Jika diklik, akan mengaktifkan Function yang menampilkan Page List Pesanan
DI03	Dropdown Item	Logout	Jika diklik, akan mengaktifkan Function logout
B18	Button	Pesan	jika ditekan akan mengaktifkan function pesan yang menampilkan page booking hotel
B08	Button	Back	Jika diklik, akan mengaktifkan Function kembali ke halaman sebelumnya

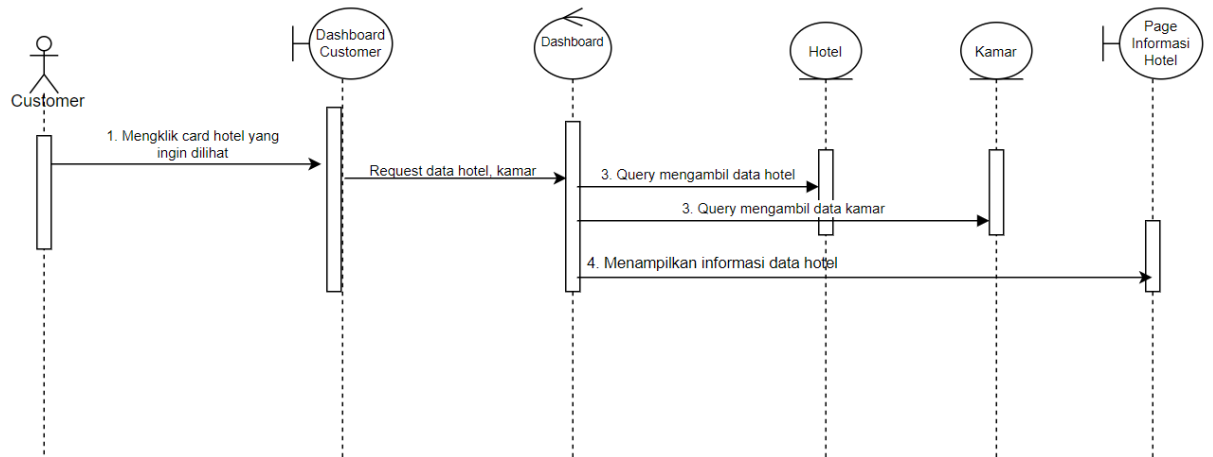
3.1.6.3 Identifikasi Object dan Tipe nya #6 <Melihat informasi produk>

TABEL OBJECT PERANCANGAN

No	Nama Object Baru	Jenis / Tipe Kelas
1	Dashboard Customer	Boundary
2	Page Informasi Hotel	Boundary
3	Hotel	Entity
4	Kamar	Entity
5	Dashboard	Controller

*Tipe kelas diisi dengan Boundary(Interface), Entity(Database), Controller

3.1.6.4 Sequence Diagram #6 <Melihat informasi produk>



3.1.7 Use Case #7 <Booking Hotel>

3.1.7.1 Use Case Scenario #7 <Booking Hotel>

Skenario Use Case #1 :

i. Pre-Condition

Customer berada pada page informasi hotel

ii. Use Case Description

a. Primary Flow

1. Customer mengklik tombol “Pesan” pada kamar yang diinginkan
2. Customer melihat page booking hotel
3. Customer memasukkan data pesanan (banyak kamar, waktu check in, dan waktu check out)
4. Customer menunggu data pesanan divalidasi oleh controller Booking saat customer klik tombol “Submit”
5. Customer menunggu data pesanan disimpan pada database pesanan
6. Customer melihat tampilan page list pesanan

iii. Post-Condition

Data pesanan tersimpan dalam database

3.1.7.2 UI Design dan Deskripsi Objek UI #7 <Booking Hotel>

Tabel Deskripsi Objek UI

ID. LAYAR	NAMA LAYAR	DESKRIPSI
L14	Page Informasi Hotel	Merupakan halaman untuk melihat informasi hotel
L15	Page Booking Hotel	Merupakan halaman untuk membooking hotel
L16	Page List Pesanan	Merupakan halaman yang menampilkan daftar pesanan yang dimiliki customer

Antarmuka L14:

Travellight
Nama Customer ▼

NAMA HOTEL

alamat

Gambar

Fasilitas

Fasilitas

Fasilitas

Fasilitas

Kamar Tersedia

Jenis kamar

Harga

Sisa kamar

....

....

....

Pesan

Back

Page Infomasi Hotel

Id_Objek	JENIS	LABEL*	Keterangan**
DM01	Dropdown Menu	Nama Customer	Jika diklik, akan menampilkan 3 menu yaitu Edit Profile, List Pesanan, dan Logout
DI01	Dropdown Item	Edit Profile	Jika diklik, akan mengaktifkan Function yang menampilkan Page Edit Profile
DI02	Dropdown Item	List Pesanan	Jika diklik, akan mengaktifkan Function yang menampilkan Page List Pesanan
DI03	Dropdown Item	Logout	Jika diklik, akan mengaktifkan Function logout
B18	Button	Pesan	jika ditekan akan mengaktifkan function pesan yang menampilkan page booking hotel
B08	Button	Back	Jika diklik, akan mengaktifkan Function kembali ke halaman sebelumnya

Antarmuka L15:

Travellight
Nama customer

Pesan Kamar

Tipe :

Harga

Sisa Kamar :

Banyaknya kamar yang ingin dipesan :

Waktu Check in

mm/dd/yyyy

Waktu Check Out

mm/dd/yyyy

Submit

Back

Page Booking Hotel

Id_Objek	JENIS	LABEL*	Keterangan**
DM01	Dropdown Menu	Nama Customer	Jika diklik, akan menampilkan 3 menu yaitu Edit Profile, List Pesanan, dan Logout
DI01	Dropdown Item	Edit Profile	Jika diklik, akan mengaktifkan Function yang menampilkan Page Edit Profile
DI02	Dropdown Item	List Pesanan	Jika diklik, akan mengaktifkan Function yang menampilkan Page List Pesanan
DI03	Dropdown Item	Logout	Jika diklik, akan mengaktifkan Function logout
TF40	Text Field	Banyak kamar yang ingin dipesan	Tempat customer mengisi banyak kamar yang ingin dipesan
DP03	Date time picker	Waktu check in	Tempat customer mengisi Waktu check in
DP04	Date time picker	Waktu Check Out	Tempat customer mengisi Waktu Check Out
B19	Button	Submit	Jika diklik, akan mengaktifkan Function save pesan
B08	Button	Back	Jika diklik, akan mengaktifkan Function kembali ke halaman sebelumnya

Antarmuka L16:

Travellight	List Pesanan	Nama Customer
<div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin-bottom: 10px;"> <p>Pesanan AU001</p> <p>Total Harga: Rp. 500.000,00</p> <p>Tenggat Waktu Pesanan: 21/10/2022 23.59</p> <div style="text-align: right; margin-top: 10px;"> <input type="button" value="Bayar"/> </div> </div> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px;"> <p>Pesanan AU002</p> <p>Total Harga: Rp. 600.000,00</p> <p>Tenggat Waktu Pesanan: 22/10/2022 23.59</p> <div style="text-align: right; margin-top: 10px;"> <input type="button" value="Bayar"/> </div> </div> <div style="margin-top: 10px;"> <input type="button" value="Back"/> </div>		

Page List Pesanan

Id_Objek	JENIS	LABEL*	Keterangan**
DM01	Dropdown Menu	Nama Customer	Jika diklik, akan menampilkan 3 menu yaitu Edit Profile, List Pesanan, dan Logout
B20	Button	Bayar	jika ditekan akan mengaktifkan function bayar
B08	Button	Back	Jika diklik, akan mengaktifkan Function kembali ke halaman sebelumnya

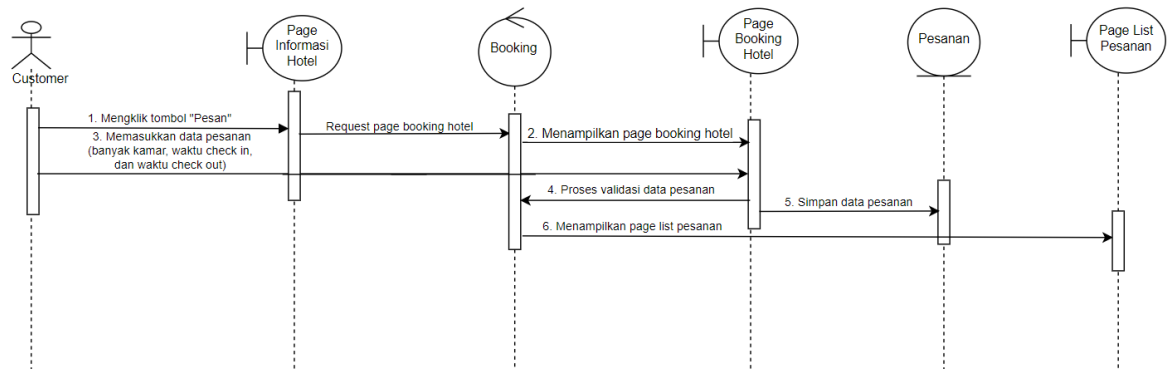
3.1.7.3 Identifikasi Object dan Tipe nya #7 <Booking Hotel>

TABEL OBJECT PERANCANGAN

No	Nama Object Baru	Jenis / Tipe Kelas
1	Page Informasi Hotel	Boundary
2	Page Booking Hotel	Boundary
3	Page List Pesanan	Boundary
4	Booking	Controller
5	Pesanan	Entity

*Tipe kelas diisi dengan Boundary(Interface), Entity(Database), Controller

3.1.7.4 Sequence Diagram #7 <Booking Hotel>



3.1.8 Use Case #8<Melakukan Pembayaran>

3.1.8.1 Use Case Scenario #8<Melakukan Pembayaran>

Skenario Use Case #1 :

- i. Pre-Condition
 - Customer berada pada page list pesanan
- ii. Use Case Description
 - a. Primary Flow
 1. Customer mengklik tombol “Bayar” pada pesanan yang ingin dibayar
 2. Customer melihat page pembayaran
 3. Customer mengupload gambar bukti transfer pembayaran
 4. Customer menunggu file gambar divalidasi oleh controller Payment saat customer mengklik tombol “Submit”
 5. Customer menunggu gambar bukti transfer pembayaran disimpan pada database pembayaran
 6. Customer menunggu status pesanan di-update menjadi “paid”
 7. Customer melihat page list pesanan
- iii. Post-Condition
 - Data pembayaran tersimpan dalam database
 - Entri status pesanan pada database pesanan berubah menjadi “paid”

3.1.8.2 UI Design dan Deskripsi Objek UI #8<Melakukan Pembayaran>

Tabel Deskripsi Objek UI

ID. LAYAR	NAMA LAYAR	DESKRIPSI
L16	Page List Pesanan	Merupakan halaman yang menampilkan daftar pesanan yang dimiliki customer
L17	Page Pembayaran	Merupakan halaman untuk mengisi bukti transfer pembayaran

Antarmuka L16:

Travellight	List Pesanan	Nama Customer
<div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin-bottom: 10px;"> Pesanan AU001 Total Harga: Rp. 500.000,00 Tenggat Waktu Pesanan: 21/10/2022 23.59 <div style="float: right; border: 1px solid #ccc; border-radius: 5px; padding: 2px 10px;">Bayar</div> </div> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px;"> Pesanan AU002 Total Harga: Rp. 600.000,00 Tenggat Waktu Pesanan: 22/10/2022 23.59 <div style="float: right; border: 1px solid #ccc; border-radius: 5px; padding: 2px 10px;">Bayar</div> </div> <div style="margin-top: 10px; border: 1px solid #ccc; border-radius: 5px; padding: 5px 10px; display: inline-block;">Back</div>		

Page List Pesanan

Id_Objek	JENIS	LABEL*	Keterangan**
DM01	Dropdown Menu	Nama Customer	Jika diklik, akan menampilkan 3 menu yaitu Edit Profile, List Pesanan, dan Logout
B20	Button	Bayar	jika ditekan akan mengaktifkan function bayar
B08	Button	Back	Jika diklik, akan mengaktifkan Function kembali ke halaman sebelumnya

Antarmuka L17:

Travellight	List Pesanan	Nama Customer
<div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin-bottom: 10px;"> Pembayaran ID Pesanan : Total Harga: Rp. 500.000,00 Tenggat Waktu Pesanan: 21/10/2022 23.59 Dipesan pada tanggal 18/10/2022 Status: Ordered </div> <p>Upload bukti transfer pembayaran</p> <div style="border: 1px solid #ccc; border-radius: 5px; padding: 5px; margin-bottom: 10px;"> <div style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 10px; display: inline-block;">Browse</div> <div style="margin-left: 10px;">No file selected</div> </div> <div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 10px;">Submit</div> <div style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 10px;">Back</div> </div>		

Page Pembayaran

Id_Objek	JENIS	LABEL*	Keterangan**
<i>FI03</i>	<i>File Input</i>	<i>Bukti Transfer Pembayaran</i>	<i>Tempat customer untuk mengupload file gambar bukti transfer</i>
<i>B21</i>	<i>Button</i>	<i>Submit</i>	<i>Jika diklik akan mengaktifkan function save bayar</i>
<i>B08</i>	<i>Button</i>	<i>Back</i>	<i>Jika diklik, akan mengaktifkan Function kembali ke halaman sebelumnya</i>

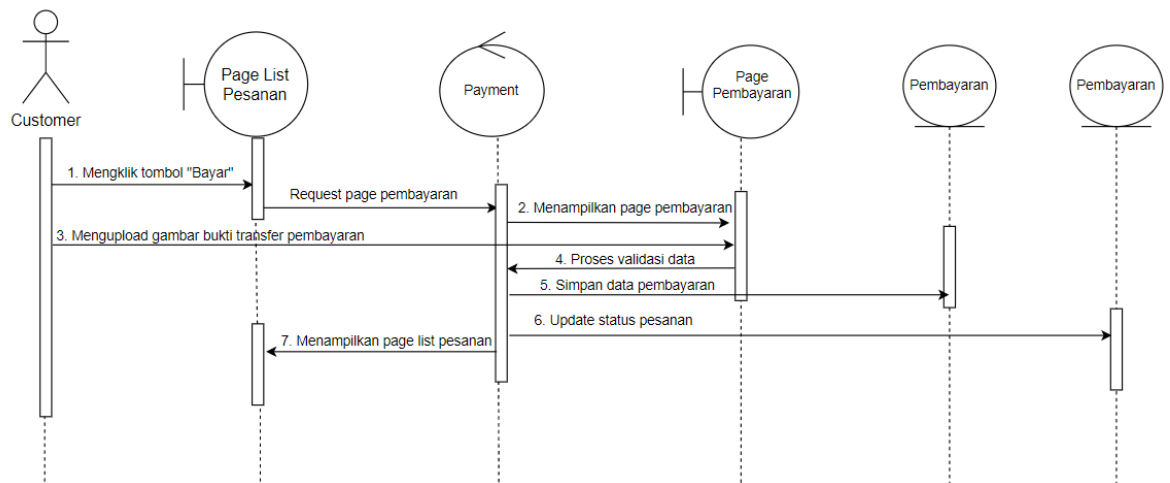
3.1.8.3 Identifikasi Object dan Tipe nya #8<Melakukan Pembayaran>

TABEL OBJECT PERANCANGAN

No	Nama Object Baru	Jenis / Tipe Kelas
<i>1</i>	<i>Page List Pesanan</i>	<i>Boundary (Interface)</i>
<i>2</i>	<i>Page Pembayaran</i>	<i>Boundary (Interface)</i>
<i>3</i>	<i>Payment</i>	<i>Controller</i>
<i>4</i>	<i>Pesanan</i>	<i>Entity(Database)</i>
<i>5</i>	<i>Pembayaran</i>	<i>Entity(Database)</i>

*Tipe kelas diisi dengan *Boundary(Interface)*, *Entity(Database)*, *Controller*

3.1.8.4 Sequence Diagram #8<Melakukan Pembayaran>



3.1.9 Use Case #9<Memverifikasi Pembayaran>

3.1.9.1 Use Case Scenario #9<Memverifikasi Pembayaran>

Skenario Use Case #1 :

- i. Pre-Condition
 - Administrator berada di page dashboard administrator
- ii. Use Case Description
 - a. Primary Flow
 1. Administrator mengklik tombol “Verifikasi” pada pembayaran yang ingin diverifikasi
 2. Administrator melihat tampilan page informasi pembayaran
 3. Administrator mengklik tombol “Verifikasi Pembayaran”
 4. Administrator menunggu status pembayaran di-update menjadi “verified” pada database pembayaran

5. Administrator menunggu status pesanan di-update menjadi “verified” pada database pesanan
6. Administrator melihat page dashboard administrator

iii. Post-Condition

Data pembayaran dan data pesanan yang berubah telah tersimpan dalam database (status pesanan dan status pembayaran menjadi terverifikasi).

3.1.9.2 UI Design dan Deskripsi Objek UI #9<Memverifikasi Pembayaran>

Tabel Deskripsi Objek UI

ID. LAYAR	NAMA LAYAR	DESKRIPSI
L18	Dashboard Administrator	Merupakan halaman untuk melihat daftar pembayaran yang belum di verifikasi
L19	Page Informasi Pembayaran	Merupakan halaman untuk melihat bukti pembayaran dan memverifikasi pembayaran

Antarmuka L18:

VERIFIKASI PEMBAYARAN

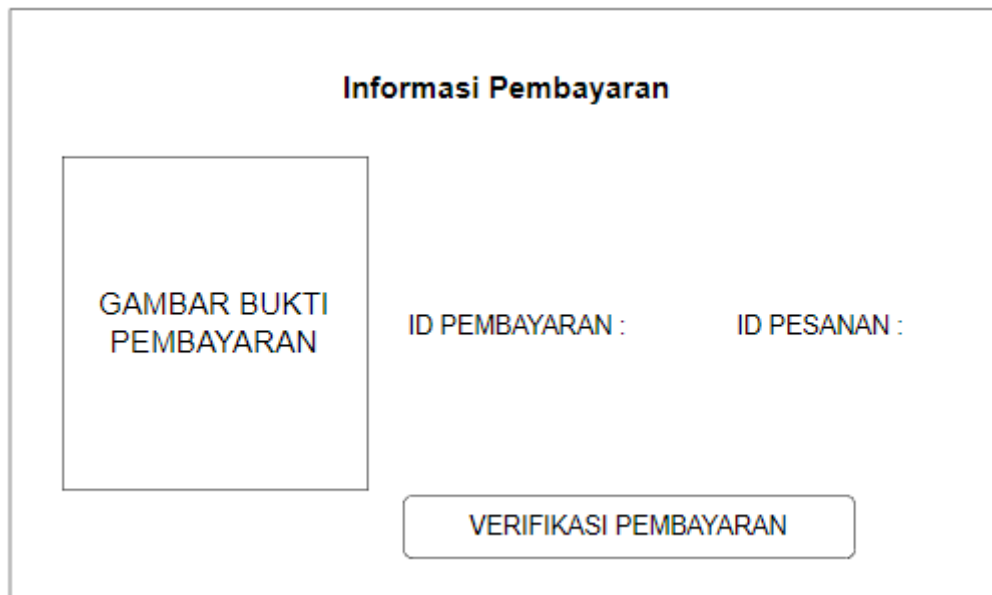
Logout

idPembayaran	idPesanan	URL Gambar Pembayaran	Status Pembayaran	
.....	Verifikasi
.....	Verifikasi
.....	Verifikasi
.....	Verifikasi
.....	Verifikasi

Page Dashboard Administrator

Id_Objek	JENIS	LABEL*	Keterangan**
B22	Button	Verifikasi	Jika di klik akan menampilkan page informasi pembayaran
B23	Button	Logout	Jika di klik akan mengaktifkan function logout

Antarmuka L19:



Page Informasi Pembayaran

Id_Objek	JENIS	LABEL*	Keterangan**
B24	Button	Verifikasi Pembayaran	Jika ditekan akan mengaktifkan function save verif

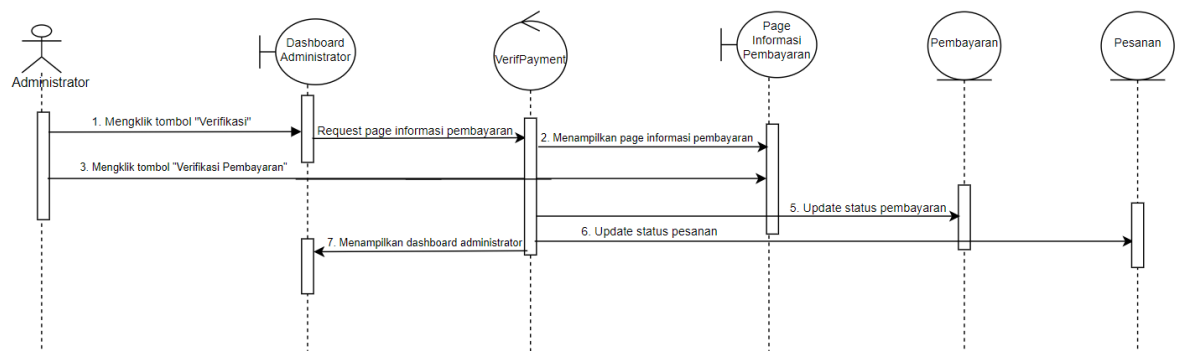
3.1.9.3 Identifikasi Object dan Tipe nya #9<Memverifikasi Pembayaran>

TABEL OBJECT PERANCANGAN

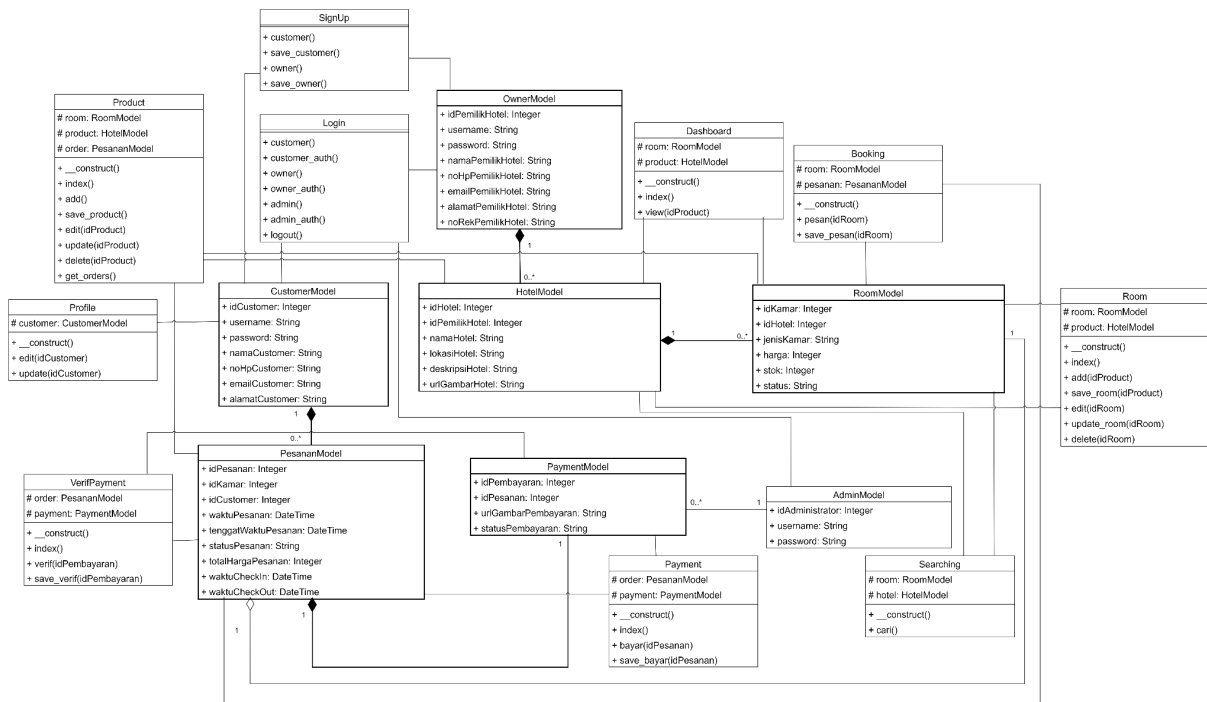
No	Nama Object Baru	Jenis / Tipe Kelas
1	Dashboard Administrator	Boundary
2	Page Informasi Pembayaran	Boundary
3	VerifPayment	Controller
4	Pembayaran	Entity
5	Pesanan	Entity

*Tipe kelas diisi dengan Boundary(Interface), Entity(Database), Controller

3.1.9.4 Sequence Diagram #9<Memverifikasi Pembayaran>



3.2 Diagram Kelas Keseluruhan



TABEL KELAS :

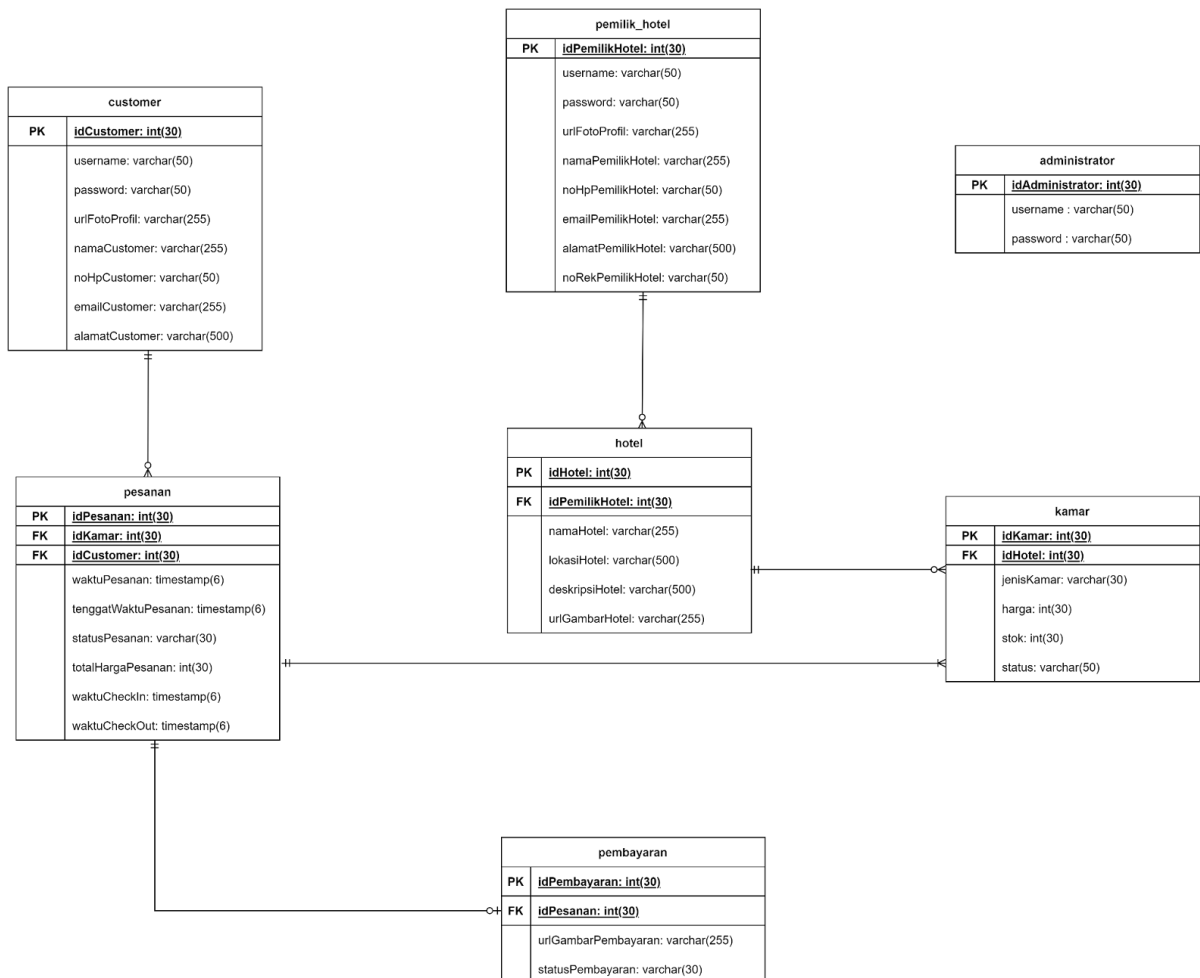
ID Kelas	Nama Kelas Perancangan	Atribute (visibility)	Method / Operation
01	CustomerModel	Public attribute: - idCustomer - username - password - namaCustomer - noHpCustomer - emailCustomer - alamatCustomer	-
02	OwnerModel	Public attribute: - idPemilikHotel - username - password - namaPemilikHotel - noHpPemilikHotel - emailPemilikHotel - alamatPemilikHotel - noRekPemilikHotel	-
03	AdminModel	Public attribute: - idAdministrator - username - password	-
04	HotelModel	Public attribute: - idHotel - idPemilikHotel	-

		<ul style="list-style-type: none"> - <i>namaHotel</i> - <i>lokasiHotel</i> - <i>deskripsiHotel</i> - <i>urlGambarHotel</i> 	
05	<i>RoomModel</i>	Public attribute: <ul style="list-style-type: none"> - <i>idKamar</i> - <i>idHotel</i> - <i>jenisKamar</i> - <i>harga</i> - <i>stok</i> - <i>status</i> 	-
06	<i>PaymentModel</i>	Public attribute: <ul style="list-style-type: none"> - <i>idPembayaran</i> - <i>idPesanan</i> - <i>urlGambarPembayaran</i> - <i>statusPembayaran</i> 	-
07	<i>PesananModel</i>	Public attribute: <ul style="list-style-type: none"> - <i>idPesanan</i> - <i>idKamar</i> - <i>idCustomer</i> - <i>waktuPesanan</i> - <i>tenggatWaktuPesanan</i> - <i>statusPesanan</i> - <i>totalHargaPesanan</i> - <i>waktuCheckIn</i> - <i>waktuCheckOut</i> 	-
08	<i>SignUp (Controller)</i>	-	<i>customer()</i> <i>save_customer()</i> <i>owner()</i> <i>save_owner()</i>
09	<i>Login (Controller)</i>	-	<i>customer()</i> <i>customer_auth()</i> <i>owner()</i> <i>owner_auth()</i> <i>admin()</i> <i>admin_auth()</i> <i>logout()</i>
10	<i>Profile (Controller)</i>	Protected attribute: <ul style="list-style-type: none"> - <i>customer</i> 	<i>__construct()</i> <i>edit(idCustomer)</i> <i>update(idCustomer)</i>
11	<i>Product (Controller)</i>	Protected attribute: <ul style="list-style-type: none"> - <i>room</i> - <i>product</i> - <i>order</i> 	<i>__construct()</i> <i>index()</i> <i>add()</i> <i>save_product()</i> <i>edit(idProduct)</i> <i>update(idProduct)</i> <i>delete(idProduct)</i> <i>get_orders()</i>
12	<i>Room (Controller)</i>	Protected attribute: <ul style="list-style-type: none"> - <i>room</i> - <i>product</i> 	<i>__construct()</i> <i>index()</i> <i>add(idProduct)</i> <i>save_room(idProduct)</i> <i>edit(idRoom)</i> <i>update room(idRoom)</i>

			<i>delete(idRoom)</i>
13	<i>Searching (Controller)</i>	Protected attribute: - room - hotel	<i>__construct()</i> <i>cari()</i>
14	<i>Dashboard (Controller)</i>	Protected attribute: - room - product	<i>__construct()</i> <i>index()</i> <i>view(idProduct)</i>
15	<i>Booking (Controller)</i>	Protected attribute: - room - pesanan	<i>__construct()</i> <i>pesan(idRoom)</i> <i>save_pesan(idRoom)</i>
16	<i>Payment (Controller)</i>	Protected attribute: - order - payment	<i>__construct()</i> <i>index()</i> <i>bayar(idPesanan)</i> <i>save_bayar(idPesanan)</i>
17	<i>VerifPayment (Controller)</i>	Protected attribute: - order - payment	<i>__construct()</i> <i>index()</i> <i>verif(idPembayaran)</i> <i>save_verif(idPembayaran)</i>

3.3 Perancangan Data / Basis Data

Entity-Relationship Diagram (ERD) basis data Travellight sebagai berikut:



3.4 Perancangan Algoritma dan/atau Query

3.4.1 Kelas SignUp

Algoritma #1

Nama Kelas : *SignUp (Controller)*
Nama Operasi : *Pendaftaran Akun Customer*
Algoritma : *save_customer*

```
public function save_customer()
{
    helper(['form']);

    $rules = [
        'name' => 'required|min_length[3]|max_length[255]',
        'phone_num' => 'required|min_length[3]|max_length[50]',
        'email' => 'required|min_length[3]|max_length[255]',
        'address' => 'required|min_length[3]|max_length[500]',
        'username' => 'required|min_length[3]|max_length[20]',
        'password' => 'required|min_length[3]|max_length[20]',
        'confpassword' => 'required|min_length[3]|max_length[20]',
        'confpassword' => 'matches[password]'
    ];

    if ($this->validate($rules)){
        $model = new CustomerModel();
        $data = [
            'namaCustomer' => $this->request->getVar('name'),
            'noHpCustomer' => $this->request->getVar('phone_num'),
            'emailCustomer' => $this->request->getVar('email'),
            'alamatCustomer' => $this->request->getVar('address'),
            'username' => $this->request->getVar('username'),
            'password' => $this->request->getVar('password')
        ];
        $model->save($data);
        return redirect()->to(base_url('login/customer'));
    } else {
        $data['validation'] = $this->validator;
        echo view('signup_customer', $data);
    }
}
```

Algoritma #2

Nama Kelas : *SignUp (Controller)*
Nama Operasi : *Pendaftaran Akun Pemilik Hotel*
Algoritma : *save_owner*

```
public function save_owner()
{
    helper(['form']);
    $rules = [
        'name' => 'required|min_length[3]|max_length[255]',
        'phone_num' => 'required|min_length[3]|max_length[50]',
        'email' => 'required|min_length[3]|max_length[255]',
        'address' => 'required|min_length[3]|max_length[500]',
        'bank_acc' => 'required|min_length[3]|max_length[50]',
        'username' => 'required|min_length[3]|max_length[20]',
        'password' => 'required|min_length[3]|max_length[20]',
    ];
```

```

        'confpassword' => 'required|min_length[3]|max_length[20]',
        'confpassword' => 'matches[password]'
    ];

    if ($this->validate($rules)){
        $model = new OwnerModel();
        $data = [
            'namaPemilikHotel' => $this->request->getVar('name'),
            'noHpPemilikHotel' => $this->request->getVar('phone_num'),
            'emailPemilikHotel' => $this->request->getVar('email'),
            'alamatPemilikHotel' => $this->request->getVar('address'),
            'noRekPemilikHotel' => $this->request->getVar('bank_acc'),
            'username' => $this->request->getVar('username'),
            'password' => $this->request->getVar('password')
        ];
        $model->save($data);
        return redirect()->to('/login/owner');
    } else {
        $data['validation'] = $this->validator;
        echo view('signup_owner', $data);
    }
}

```

3.4.2 Kelas Login

Algoritma #3

Nama Kelas : Login (Controller)
 Nama Operasi : Login Akun Customer
 Algoritma : customer_auth

```

public function customer_auth()
{
    $session = session();
    $model = new CustomerModel();
    $username= $this->request->getVar('username');
    $password = $this->request->getVar('password');
    $data = $model->where('username', $username)->first();
    if($data){
        $pass = $data['password'];
        if($password == $data['password']){
            $ses_data = [
                'idCustomer' => $data['idCustomer'],
                'username' => $data['username'],
                'password' => $data['password'],
                'urlFotoProfil' => $data['urlFotoProfil'],
                'namaCustomer' => $data['namaCustomer'],
                'noHpCustomer' => $data['noHpCustomer'],
                'emailCustomer' => $data['emailCustomer'],
                'alamatCustomer' => $data['alamatCustomer'],
                'logged_in' => 'customer'
            ];
            $session->set($ses_data);
            return redirect()->to(base_url('dashboard'));
        }else{
            $session->setFlashdata('msg', 'Wrong Password');
            return redirect()->to(base_url('login/customer'));
        }
    }
}

```

```

        $session→setFlashdata('msg', 'Email not Found');
        return redirect()→to(base_url('login/customer'));
    }
}

```

Algoritma #4

Nama Kelas : Login (Controller)
 Nama Operasi : Login Akun Pemilik Hotel
 Algoritma : owner_auth

```

public function owner_auth()
{
    $session = session();
    $model = new OwnerModel();
    $username= $this→request→getVar('username');
    $password = $this→request→getVar('password');
    $data = $model→where('username', $username)→first();
    if($data){
        $pass = $data['password'];
        if($password == $data['password']){
            $ses_data = [
                'idPemilikHotel'      ⇒ $data['idPemilikHotel'],
                'username'            ⇒ $data['username'],
                'password'            ⇒ $data['password'],
                'urlFotoProfil'       ⇒ $data['urlFotoProfil'],
                'namaPemilikHotel'    ⇒ $data['namaPemilikHotel'],
                'noHpPemilikHotel'    ⇒ $data['noHpPemilikHotel'],
                'emailPemilikHotel'   ⇒ $data['emailPemilikHotel'],
                'alamatPemilikHotel' ⇒ $data['alamatPemilikHotel'],
                'noRekPemilikHotel'   ⇒ $data['noRekPemilikHotel'],
                'logged_in'           ⇒ 'owner'
            ];
            $session→set($ses_data);
            return redirect()→to(base_url('products'));
        }else{
            $session→setFlashdata('msg', 'Wrong Password');
            return redirect()→to(base_url('login/owner'));
        }
    }else{
        $session→setFlashdata('msg', 'Email not Found');
        return redirect()→to(base_url('login/owner'));
    }
}

```

Algoritma #5

Nama Kelas : Login (Controller)
 Nama Operasi : Login Akun Administrator
 Algoritma : admin_auth

```

public function admin_auth()
{
    $session = session();
    $model = new AdminModel();
    $username= $this→request→getVar('username');
    $password = $this→request→getVar('password');
    $data = $model→where('username', $username)→first();
    if($data){

```

```
$pass = $data['password'];
if($password == $data['password']){
    $ses_data = [
        'idAdministrator'    ⇒ $data['idAdministrator'],
        'username'           ⇒ $data['username'],
        'password'           ⇒ $data['password'],
        'logged_in'          ⇒ TRUE
    ];
    $session→set($ses_data);
    return redirect()→to(base_url('verifpembayaran'));
}else{
    $session→setFlashdata('msg', 'Wrong Password');
    return redirect()→to(base_url('login/admin'));
}
}else{
    $session→setFlashdata('msg', 'Email not Found');
    return redirect()→to(base_url('login/admin'));
}
}
```

Algoritma #6

Nama Kelas : Login (Controller)

Nama Operasi : Logout

Algoritma : *logout*

```
public function logout()
{
    $session = session();
    $session->destroy();
    return redirect()->to(base_url('/'));
}
```

3.4.3 Kelas Profile

Algoritma #7

Nama Kelas : Profile (Controller)

Nama Operasi : Update Profile

Algoritma : *edit & update*

```
function edit($id)
{
    $dataCustomer = $this->customer->find($id);
    if (empty($dataCustomer)) {
        throw new \CodeIgniter\Exceptions\PageNotFoundException('Data
Customer tidak ditemukan');
    }
    $data['customer'] = $dataCustomer;
    return view('edit_profile', $data);
}

public function update($id)
{
    if (!$this->validate([
        'name' => [
            'rules' => 'required',
            'errors' => [
                'required' => '{field} harus diisi'
```

```

    ],
    'phone_num' => [
        'rules' => 'required',
        'errors' => [
            'required' => '{field} harus diisi'
        ]
    ],
    'email' => [
        'rules' => 'required',
        'errors' => [
            'required' => '{field} harus diisi'
        ]
    ],
    'address' => [
        'rules' => 'required',
        'errors' => [
            'required' => '{field} harus diisi'
        ]
    ],
    'username' => [
        'rules' => 'required',
        'errors' => [
            'required' => '{field} harus diisi'
        ]
    ],
    'username' => [
        'rules' => 'required',
        'errors' => [
            'required' => '{field} harus diisi'
        ]
    ],
    ]), {
        session()→setFlashdata('error', $this→validator→listErrors());
        return redirect()→back();
    }
    $this→customer→update($id, [
        'namaCustomer' => $this→request→getVar('name'),
        'noHpCustomer' => $this→request→getVar('phone_num'),
        'emailCustomer' => $this→request→getVar('email'),
        'alamatCustomer' => $this→request→getVar('address'),
        'username' => $this→request→getVar('username')
    ]);
    session()→setFlashdata('message', 'Update Data Customer Berhasil');
    return redirect()→to('/dashboard');
}

```

3.4.4 Kelas Product

Algoritma #8

Nama Kelas : Product (Controller)

Nama Operasi : Tambah Produk

Algoritma : save_product

```

public function save_product()
{

```

```

        helper(['form']);
        $id = session()→get('idPemilikHotel');

        $rules = [
            'name' ⇒ 'required|min_length[3]|max_length[255]',
            'location' ⇒ 'required|min_length[3]|max_length[500]',
            'desc' ⇒ 'required|min_length[3]|max_length[500]'
        ];
        $upload = $this→request→getFile('urlGambarHotel');
        $fileName = $upload→getName();
        $upload→move(' ../public/assets/imghotel/', $fileName);

        if ($this→validate($rules)){

            $data = [
                'idPemilikHotel' ⇒ $id,
                'namaHotel' ⇒ $this→request→getVar('name'),
                'lokasiHotel' ⇒ $this→request→getVar('location'),
                'deskripsiHotel' ⇒ $this→request→getVar('desc'),
                'urlGambarHotel' ⇒ $upload→getName()
            ];
            $this→product→save($data);
            return redirect()→to(base_url('/products'));
        } else {
            $data['validation'] = $this→validator;
            echo view('products_add', $data);
        }
    }
}

```

Algoritma #9

Nama Kelas : Product (Controller)

Nama Operasi : Edit Produk

Algoritma : edit & update

```

function edit($id)
{
    $dataProduct = $this→product→find($id);
    if (empty($dataProduct)){
        throw new \CodeIgniter\Exceptions\PageNotFoundException('Data Produk
tidak ditemukan');
    }
    $data['product'] = $dataProduct;
    return view('products_edit', $data);
}

public function update($id)
{
    if (!$this→validate([
        'name' ⇒ [
            'rules' ⇒ 'required',
            'errors' ⇒ [
                'required' ⇒ '{field} harus diisi'
            ]
        ],
        'location' ⇒ [
            'rules' ⇒ 'required',
            'errors' ⇒ [

```

```

        'required' => '{field} harus diisi'
    ],
    'desc' => [
        'rules' => 'required',
        'errors' => [
            'required' => '{field} harus diisi'
        ]
    ],
    )) {
        session()→setFlashdata('error', $this→validator→listErrors());
        return redirect()→back();
    }

    $this→product→update($id, [
        'namaHotel' => $this→request→getVar('name'),
        'lokasiHotel' => $this→request→getVar('location'),
        'deskripsiHotel' => $this→request→getVar('desc')
    ]);
    session()→setFlashdata('message', 'Update Data Produk Berhasil');
    return redirect()→to(base_url('/products'));
}

```

Algoritma #10

Nama Kelas : Product (Controller)
Nama Operasi : Hapus Produk
Algoritma : delete

```

function delete($id){
    $db = \Config\Database::connect();
    $builder = $db→table('kamar');
    $builder→delete(['idHotel' => $id]);
    $this→product→delete($id);
    session()→setFlashdata('message', 'Delete Data Produk Berhasil');
    return redirect()→to(base_url('/products'));
}

```

3.4.5 Kelas Room

Algoritma #11

Nama Kelas : Room (Controller)
Nama Operasi : Tambah kamar
Algoritma : add & save_room

```

public function add($id)
{
    $product = new HotelModel();
    $data['room'] = $product→find($id);
    return view('rooms_add', $data);
}

public function save_room($id)
{
    helper(['form']);
    if (!$this→validate([

```



```

        'jenisKmr' => [
            'rules' => 'required',
            'errors' => [
                'required' => '{field} harus diisi'
            ]
        ],
        'harga' => [
            'rules' => 'required',
            'errors' => [
                'required' => '{field} harus diisi'
            ]
        ],
        'stok' => [
            'rules' => 'required',
            'errors' => [
                'required' => '{field} harus diisi'
            ]
        ]
    )) {
        session()→setFlashdata('error', $this→validator→listErrors());
        return redirect()→back();
    }

    $rules = [
        'jenisKmr' => 'required|min_length[3]|max_length[30]',
        'harga' => 'required|min_length[3]|max_length[30]',
        'stok' => 'required|min_length[1]|max_length[30]'
    ];
    if ($this→validate($rules)){
        $model = new RoomModel();
        $data = [
            'idHotel' => $id,
            'jenisKamar' => $this→request→getVar('jenisKmr'),
            'harga' => $this→request→getVar('harga'),
            'stok' => $this→request→getVar('stok'),
            'status' => $this→request→getVar('status')
        ];
        $model→save($data);
        return redirect()→to(base_url('/products'));
    } else {
        $data['validation'] = $this→validator;
        echo view('rooms_add', $data);
    }
}

```

Algoritma #12

Nama Kelas : Room (Controller)
 Nama Operasi : Edit kamar
 Algoritma : edit & update_room

```

function edit($id)
{
    $dataProduct = $this→room→find($id);
    if (empty($dataProduct)){
        throw new \CodeIgniter\Exceptions\PageNotFoundException('Data Kamar
tidak ditemukan');
    }
}

```

```

        $data['room'] = $dataProduct;
        return view('rooms_edit', $data);
    }

    public function update_room($id)
    {
        if (!$this->validate([
            'jenisKamar' => [
                'rules' => 'required',
                'errors' => [
                    'required' => '{field} harus diisi'
                ]
            ],
            'harga' => [
                'rules' => 'required',
                'errors' => [
                    'required' => '{field} harus diisi'
                ]
            ],
            'stok' => [
                'rules' => 'required',
                'errors' => [
                    'required' => '{field} harus diisi'
                ]
            ]
        ])) {
            session()->setFlashdata('error', $this->validator->listErrors());
            return redirect()->to(base_url('/rooms/edit/'.$id));
        }

        $this->room->update($id, [
            'jenisKamar' => $this->request->getVar('jenisKamar'),
            'harga' => $this->request->getVar('harga'),
            'stok' => $this->request->getVar('stok'),
            'status' => $this->request->getVar('status')
        ]);
        session()->setFlashdata('message', 'Update Data Produk Berhasil');
        return redirect()->to(base_url('/products'));
    }
}

```

Algoritma #13

Nama Kelas : Room (Controller)

Nama Operasi : Hapus kamar

Algoritma : delete

```

function delete($id){
    $dataProduct = $this->room->find($id);
    if (empty($dataProduct)){
        throw new \CodeIgniter\Exceptions\PageNotFoundException('Data Kamar
tidak ditemukan');
    }
    $this->room->delete($id);
    session()->setFlashdata('message', 'Delete Data Produk Berhasil');
    return redirect()->to(base_url('/products'));
}

```

3.4.6 Kelas Searching

Algoritma #14

Nama Kelas : Searching (Controller)

Nama Operasi : Pencarian produk

Algoritma : cari

```
public function cari()
{
    helper(['form']);

    $rules = [
        'banyak' => 'required|min_length[1]',
        'lokasi' => 'required|min_length[1]',
        'checkin' => 'required',
        'checkout' => 'required'
    ];

    if ($this->validate($rules)){
        $bykPesanan = $this->request->getVar('banyak');
        if ($bykPesanan ≤ 0){
            session()->setFlashdata('underbook', 'Banyak kamar tidak valid');
            return redirect()->to(base_url('dashboard'));
        } else {
            if (($this->request->getVar('checkin') =
            $this->request->getVar('checkout')) ||
            ($this->request->getVar('checkin') > $this->request->getVar('checkout'))){
                session()->setFlashdata('invaliddate', 'Tanggal check in dan check out
                tidak valid');
                return redirect()->to(base_url('dashboard'));
            }
        }
        $db = \Config\Database::connect();
        $squery = 'SELECT * FROM kamar join hotel on kamar.idHotel=hotel.idHotel where
        kamar.stok ≥ ' . $this->request->getVar('banyak') .
        ' and LOWER(hotel.lokasiHotel) LIKE LOWER("%' .
        $this->request->getVar('lokasi') . '%" ) and kamar.status = "available" group by
        kamar.idHotel';
        $result = $db->query($squery);
        $data['hotel'] = $result->getResult('array');
        echo view('search_view', $data);
    } else {
        $data['validation'] = $this->validator;
        return redirect()->to(base_url('/dashboard'));
    }
}
```

3.4.7 Kelas Dashboard

Algoritma #15

Nama Kelas : Dashboard (Controller)

Nama Operasi : Melihat informasi produk

Algoritma : view

```
function view($id)
{
    $dataProduct = $this->product->find($id);
    if (empty($dataProduct)){
        throw new \CodeIgniter\Exceptions\PageNotFoundException('Data Produk
        tidak ditemukan');
    }
    $data['product'] = $dataProduct;
    $data['room'] = $this->room->findAll();
}
```

```

        return view('products_view', $data);
    }

```

3.4.8 Kelas Booking

Algoritma #16

Nama Kelas : Booking (Controller)

Nama Operasi : Pesan kamar

Algoritma : pesan & save_pesan

```

public function pesan($id)
{
    $dataRoom = $this->room->find($id);
    if (empty($dataRoom)){
        throw new \CodeIgniter\Exceptions\PageNotFoundException('Data Kamar tidak
ditemukan');
    }
    $data['room'] = $dataRoom;
    return view('book_view', $data);
}

public function save_pesan($id)
{
    helper(['form']);
    $idCustomer = session()->get('idCustomer');

    $rules = [
        'banyak' => 'required|min_length[1]',
        'checkin' => 'required',
        'checkout' => 'required'
    ];

    if ($this->validate($rules)){
        $model = new RoomModel();
        $data = [
            'idKamar' => $id,
            'idCustomer' => $idCustomer,
            'statusPesanan' => 'ordered',
            'waktuCheckIn' => $this->request->getVar('checkin'),
            'waktuCheckOut' => $this->request->getVar('checkout')
        ];
        $dataRoom = $this->room->find($id);
        // cek stok kamar ≥ banyak pesanan
        $bykPesanan = $this->request->getVar('banyak');
        if ($dataRoom['stok'] < $bykPesanan){
            session()->setFlashdata('overbook', 'Banyak kamar yang dipesan melebihi
kamar yang tersedia');
            return redirect()->to(base_url('book/'.$id));
        } else if ($bykPesanan ≤ 0){
            session()->setFlashdata('underbook', 'Banyak kamar yang dipesan tidak
valid');
            return redirect()->to(base_url('book/'.$id));
        } else {
            if (($data['waktuCheckIn'] = $data['waktuCheckOut']) ||
($this->request->getVar('checkin') > $this->request->getVar('checkout'))){
                session()->setFlashdata('invaliddate', 'Tanggal check in dan check out
tidak valid');
                return redirect()->to(base_url('book/'.$id));
            } else {
                // hitung total harga
                $wktCheckIn = new DateTime($this->request->getVar('checkin'));
                $wktCheckOut = new DateTime($this->request->getVar('checkout'));
                $selisih = $wktCheckOut->diff($wktCheckIn);
                $totalHarga = ($selisih->d) * $bykPesanan * $dataRoom['harga'];
            }
        }
    }
}

```

```

        $data['totalHargaPesanan'] = $totalHarga;
        // update sisa kamar
        $dataRoom['stok'] -= $bykPesanan;
        if ($dataRoom['stok'] == 0){
            $dataRoom['status'] = 'not available';
        }
        // waktu pesanan dan tenggat waktu pembayaran
        $waktu = new DateTime();
        $data['waktuPesanan'] = $waktu->format('Y-m-d H:i:s');
        $hours = 24;
        $tgWaktu = (clone $waktu)->add(new DateInterval("PT{$hours}H"));
        $data['tenggatWaktuPesanan'] = $tgWaktu->format('Y-m-d H:i:s');

        $model->save($dataRoom);
        $this->pesanan->save($data);
        return redirect()->to(base_url('/orders'));
    }
}
} else {
    $data['validation'] = $this->validator;
    echo view('book_view', $data);
}
}
}

```

3.4.9 Kelas Payment

Algoritma #17

Nama Kelas : Payment (Controller)
Nama Operasi : Melakukan pembayaran
Algoritma : bayar & save_bayar

```

function bayar($id) {
    $dataOrder = $this->order->find($id);
    if (empty($dataOrder)){
        throw new \CodeIgniter\Exceptions\PageNotFoundException('Data
Pesanan tidak ditemukan');
    }
    $data['order'] = $dataOrder;
    return view('pay_view', $data);
}

public function save_bayar($id)
{
    helper(['form']);
    $model = new PesananModel();
    $upload = $this->request->getFile('buktiTf');
    $fileName = $upload->getName();
    $upload->move('../public/assets/images/', $fileName);
    $data = [
        'idPesanan' => $id,
        'urlGambarPembayaran' => $upload->getName(),
        'statusPembayaran' => 'paid'
    ];
    $dataPesanan = $this->order->find($id);
    // cek waktu saat ini < tenggat waktu
    $waktu = new DateTime();
    $currTime = $waktu->format('Y-m-d H:i:s');
    if ($currTime > $dataPesanan['tenggatWaktuPesanan']){
        session()->setFlashdata('tenggatbyr', 'Pembayaran tidak dapat
dilakukan! - melebihi batas waktu pembayaran');
    }
}

```

```

        return redirect()→to(base_url('pay/'.$id));
    } else {
        // update status pesanan
        $dataPesanan['statusPesanan'] = 'paid';
        $model→save($dataPesanan);
        $this→payment→save($data);
        return redirect()→to(base_url('/orders'));
    }
}

```

3.4.10 Kelas VerifPayment

Algoritma #18

Nama Kelas : VerifPayment (Controller)

Nama Operasi : Verifikasi pembayaran

Algoritma : verif & save_verif

```

function verif($id) {
    $dataPembayaran = $this→payment→find($id);
    if (empty($dataPembayaran)){
        throw new \CodeIgniter\Exceptions\PageNotFoundException('Data
Pembayaran tidak ditemukan');
    }
    $data['payment'] = $dataPembayaran;
    return view('verif_view', $data);
}

function save_verif($id) {
    $modelPembayaran = new PaymentModel();
    $modelPesanan = new PesananModel();
    $dataPembayaran = $this→payment→find($id);
    $dataPesanan = $this→order→find($dataPembayaran['idPesanan']);
    $dataPembayaran['statusPembayaran'] = 'verified';
    $dataPesanan['statusPesanan'] = 'verified';
    $modelPembayaran→save($dataPembayaran);
    $modelPesanan→save($dataPesanan);
    return redirect()→to(base_url('/verifpembayaran'));
}

```

4. Matriks Keruntutan (Requirement Traceability Matrix)

Kode FR	Nama Functional Requirement	Nama Use Case
SKPL-F-TVL-01	Pendaftaran Akun	Pendaftaran Akun
SKPL-F-TVL-02	Login ke akun	Login
SKPL-F-TVL-03	Update profile	Update profile
SKPL-F-TVL-04	Manajemen produk	Manajemen produk
SKPL-F-TVL-05	Pencarian informasi produk	Pencarian informasi produk
SKPL-F-TVL-06	Melihat informasi produk	Melihat informasi produk
SKPL-F-TVL-07	Booking hotel	Booking hotel
SKPL-F-TVL-08	Melakukan pembayaran	Melakukan pembayaran
SKPL-F-TVL-09	Memverifikasi pembayaran	Memverifikasi pembayaran