

BAB 4 ANALISIS KEBUTUHAN

Bab ini menjelaskan tentang analisis kebutuhan sistem *self-service reservation* pada Everyday Smart Hotel Malang yang terdiri dari identifikasi kebutuhan sudut pandang bisnis dan sudut pandang pengguna. Kemudian dilakukan pemodelan *use case diagram* beserta *use case scenario*. Berdasarkan *use case scenario* dapat dibuat *activity diagram*.

4.1 Business Perspective

Business Perspective merupakan sudut pandang yang berfokus pada fungsi dan proses bisnis yang dibutuhkan untuk memenuhi tujuan dan strategis bisnis sehingga didapatkan kebutuhan yang dapat mendukung tujuan bisnis.

4.1.1 Proses Bisnis (As-is vs To-Be)

Analisis proses bisnis (*as-is vs to-be*) dilakukan untuk mengetahui proses bisnis yang sedang berjalan (*as-is*), kemudian dilakukan analisis untuk memahami dan mengetahui masalah dalam proses bisnis *as-is* sehingga didapatkan solusi yang dimodelkan dalam proses bisnis *to-be*.

4.1.1.1 Pemodelan Proses Bisnis As-Is Pemesanan Kamar

Pemodelan proses bisnis *as-is* digunakan untuk memodelkan proses bisnis yang sedang berjalan di Everyday Smart Hotel yaitu proses awal pemesanan *on the spot* dari awal hingga akhir proses pemesanan. Berdasarkan hasil wawancara terhadap *manager E-Commerce* dan IT operasional Everyday Smart Hotel Malang serta pengamatan secara langsung didapatkan proses bisnis yang berjalan pada pemesanan kamar di Everyday Smart Hotel seperti yang ditunjukkan pada Gambar 4.1 berikut ini :

1. Pelanggan datang ke *lobby* hotel.
2. *Receptionist* menjelaskan tipe kamar, fasilitas, serta harga yang tersedia kepada pelanggan.
3. Pelanggan memilih kamar, jumlah beserta waktu lama yang diinginkan.
4. *Receptionist* memberikan *form* biodata pemesanan kepada pelanggan.
5. Pelanggan melakukan pengisian data diri pelanggan di *form* pemesanan dan memberikan *form* pemesanan kepada *receptionist*.
6. *Receptionist* memasukkan data pelanggan ke dalam sistem.
7. Pelanggan melakukan pembayaran sesuai yang telah dipesan.
8. *Receptionist* memberikan kunci kamar dan memberikan bukti pembayaran.
9. Pelanggan menerima kunci kamar dan menerima bukti pembayaran.

4.1.1.2 Pemodelan Proses Bisnis As-Is Pemesanan Laundry

Pemodelan proses bisnis *as-is* digunakan untuk memodelkan proses bisnis yang sedang berjalan di Everyday Smart Hotel yaitu proses awal pemesanan *on the spot* dari awal hingga akhir proses pemesanan. Berdasarkan hasil wawancara terhadap *manager E-Commerce* dan IT operasional Everyday Smart Hotel Malang serta pengamatan secara langsung didapatkan proses bisnis yang berjalan pada pemesanan laundry di Everyday Smart Hotel seperti yang ditunjukkan pada Gambar 4.2 berikut ini :

1. Pelanggan datang ke *lobby* hotel.
2. *Receptionist* memberikan form laundry kepada pelanggan.
3. Pelanggan mengisi pemesanan laundry yang diinginkan.
4. *Receptionist* memberikan *form* biodata pemesanan kepada pelanggan.
5. Pelanggan melakukan pengisian data diri pelanggan di *form* pemesanan dan memberikan *form* pemesanan kepada *receptionist*.
6. *Receptionist* memasukkan data pelanggan ke dalam sistem.
7. Pelanggan melakukan pembayaran sesuai yang telah dipesan.
8. *Receptionist* memberikan bukti pembayaran.

4.1.1.3 Analisis Permasalahan

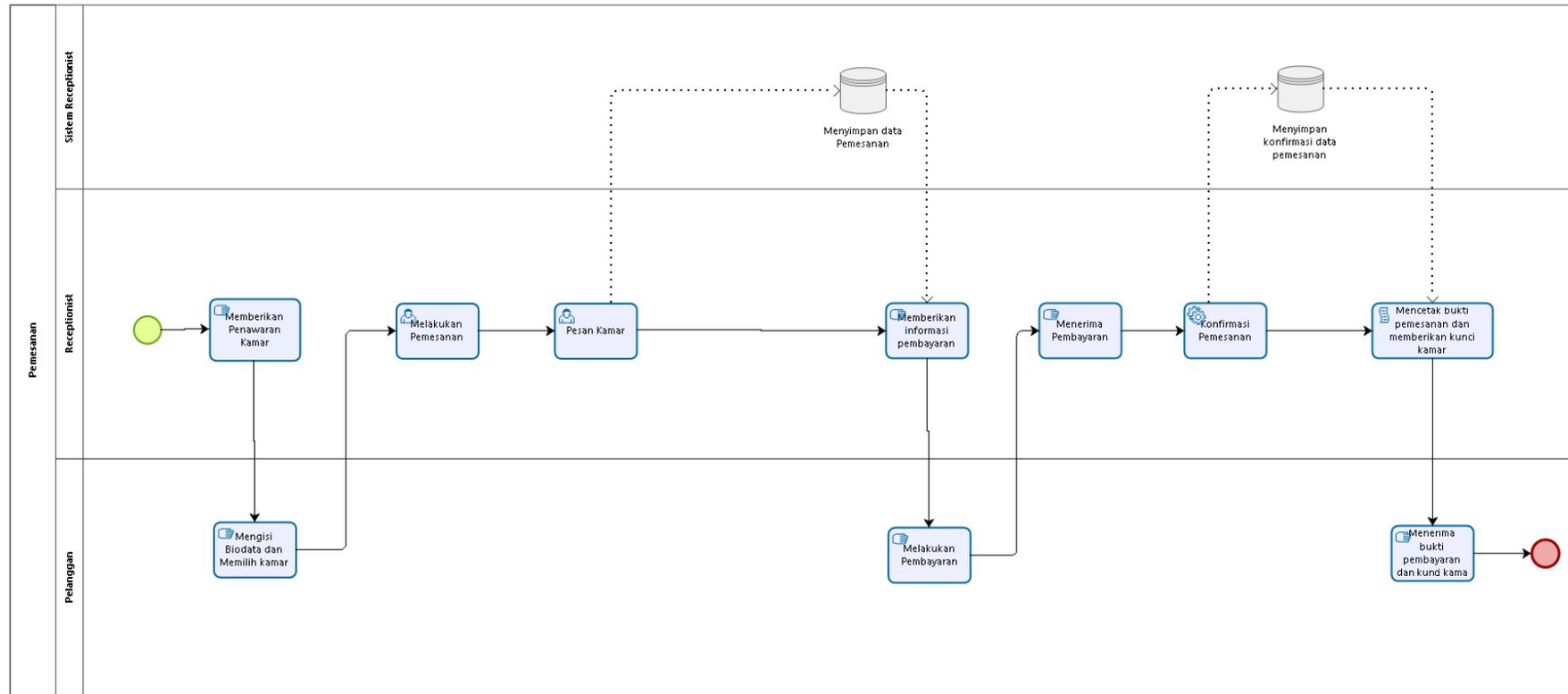
Analisis permasalahan menjelaskan masalah yang terjadi saat ini dalam proses pemesanan di Everyday Smart Hotel Malang sehingga ditemukan solusi yang dapat mengatasi masalah tersebut. Analisis permasalahan dilakukan berdasarkan hasil wawancara dengan bagian *e-commerce* serta proses bisnis *as-is*. Tabel 4.1 menjelaskan tentang analisis permasalahan pemesanan.

Tabel 4.1 Analisis Permasalahan Pemesanan

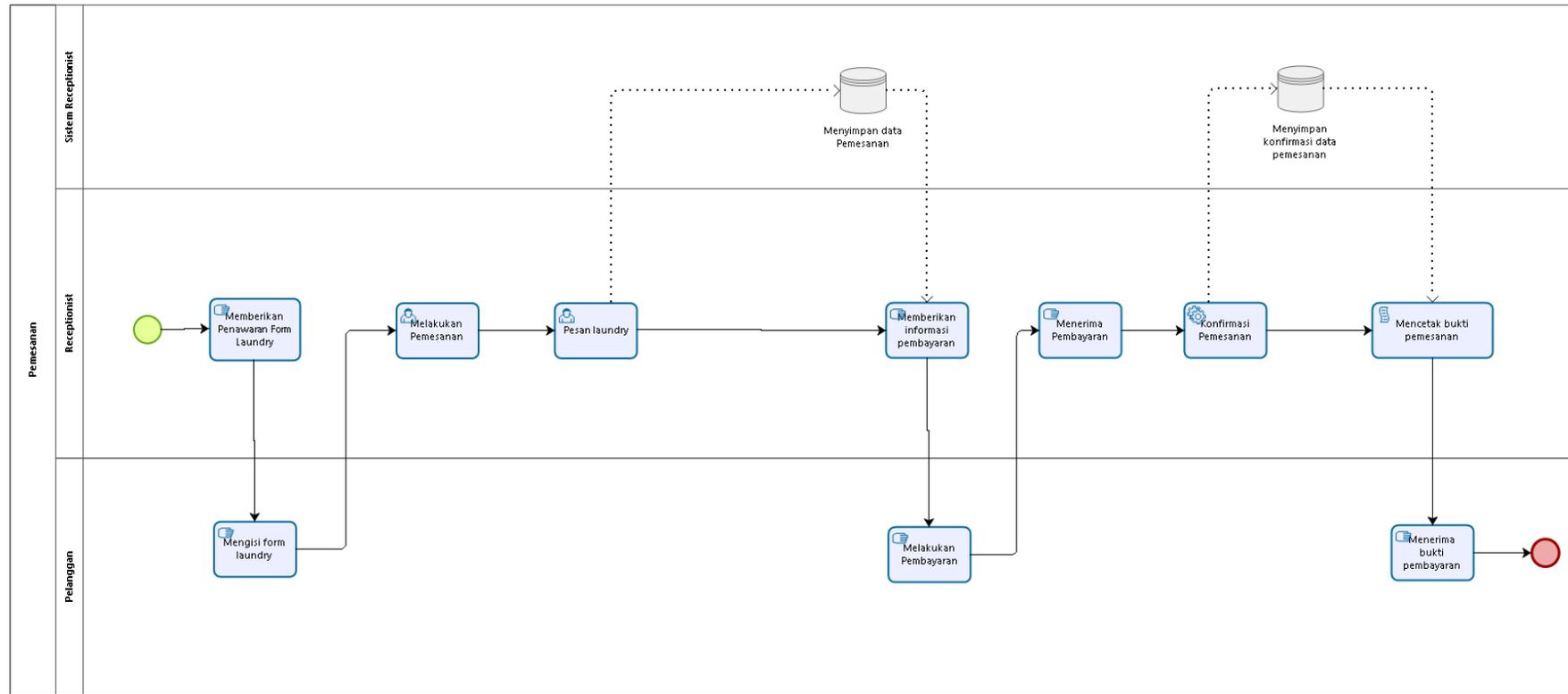
Masalah	<ol style="list-style-type: none">1. Pelanggan mengisi data masih secara manual pada formulir yang disediakan kemudian diinputkan kembali oleh <i>receptionist</i> kedalam sistem sehingga membutuhkan waktu yang lama.2. Pelanggan tidak mengetahui informasi secara detail kamar dan fasilitasnya sehingga <i>receptionist</i> selalu menjelaskan kepada pelanggan.3. Ketika pelanggan yang cukup banyak pada <i>event</i> tertentu, maka adanya antrian pada <i>lobby</i> hotel.
Mempengaruhi	Pelanggan

Tabel 4.1 Analisis Permasalahan Pemesanan (lanjutan)

Dampak	<ol style="list-style-type: none">1. Beberapa pelanggan yang <i>complaint</i> akibat tidak sesuai dengan ekspektasi mereka.2. <i>Receptionist</i> selalu menjelaskan informasi kamar yang membutuhkan waktu yang lama.3. Adanya antrian yang panjang mengakibatkan pelanggan yang beralih ke hotel sekitar dan meninggalkan Everyday Smart Hotel Malang
Solusi	<ol style="list-style-type: none">1. Menyediakan sistem yang dapat memberikan informasi kamar dan fasilitas beserta harganya.2. Menyediakan sistem yang dapat digunakan oleh pelanggan untuk melakukan pemesanan.3. Menyediakan sistem yang dapat mengisi biodata pelanggan.4. Menyediakan sistem informasi transaksi yang telah dipesan.5. <i>Receptionist</i> hanya melakukan konfirmasi yang telah dilakukan oleh pelanggan.



Gambar 4.1 Proses Bisnis Pemesanan Kamar As-Is



Gambar 4.2 Proses Bisnis Pemesanan Laundry As-Is

Tabel 4.2 menjelaskan tentang analisis permasalahan pemesanan saat ini dan waktu yang diperlukan untuk melakukan proses pemesanan. Data yang diterima berdasarkan observasi dan wawancara.

Tabel 4.2 Analisis Masalah Pemesanan Kamar dan Waktu Yang Dibutuhkan

No	Identifikasi Masalah	Rata-rata Waktu yang diperlukan (menit)
1	Pelanggan tidak mengetahui informasi secara detail kamar dan fasilitasnya sehingga <i>receptionist</i> selalu menjelaskan kepada pelanggan	02.00.00
2	Pelanggan mengisi data masih secara manual pada formulir yang disediakan kemudian diinputkan kembali oleh <i>receptionist</i> kedalam sistem sehingga membutuhkan waktu yang lama.	02.00.00
3	<i>Receptionist</i> melakukan konfirmasi pemesanan	00.30.00
4	Ketika pelanggan yang cukup banyak pada <i>event</i> tertentu, maka adanya antrian pada <i>lobby</i> hotel.	05.00.00

4.1.1.4 Pemodelan Proses Bisnis Pemesanan Kamar *To-Be*

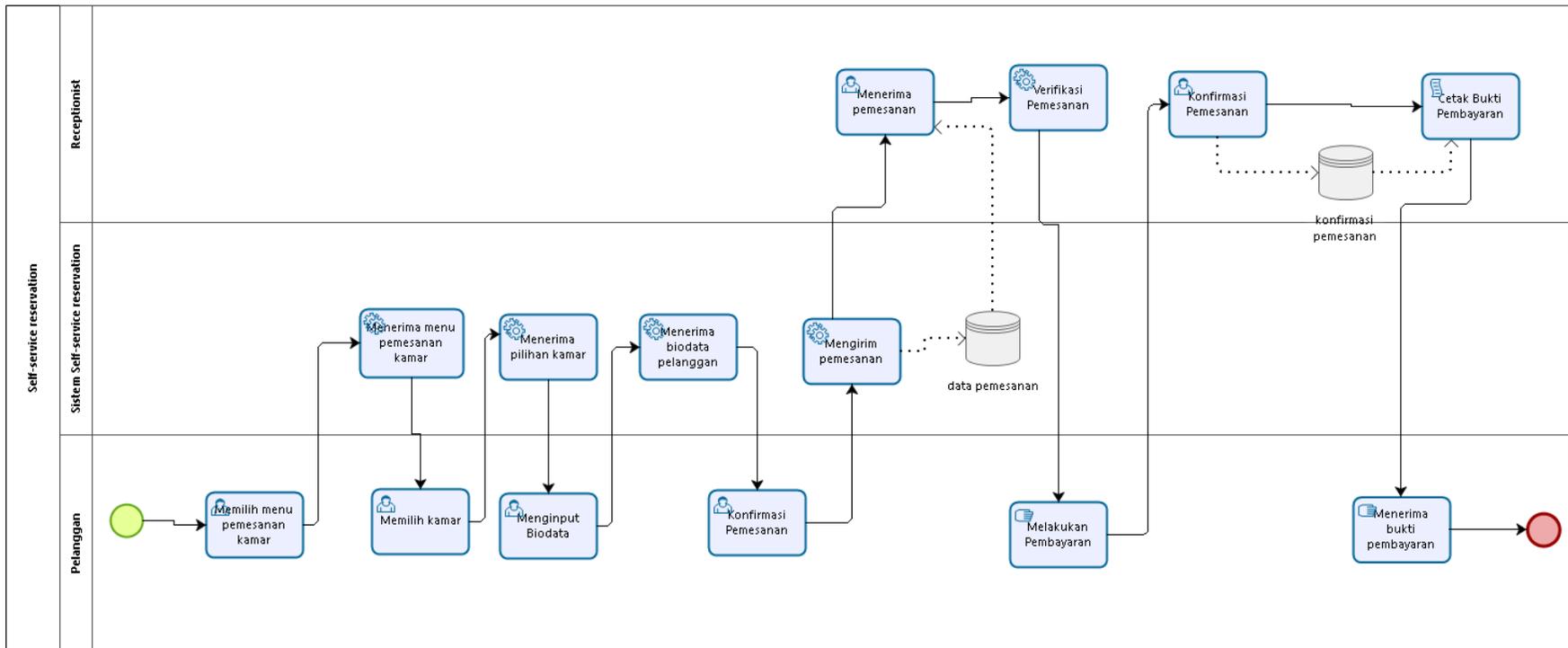
Permasalahan proses bisnis *to-be* digunakan untuk memodelkan proses bisnis usulan di Everyday Smart Hotel Malang yaitu pemesanan berdasarkan solusi yang terdapat pada analisis permasalahan. Alur proses pemesanan *to-be* dijelaskan pada sesuai Gambar 4.3 adalah :

1. Pelanggan datang ke *lobby* hotel.
2. Pelanggan dapat melakukan pemesanan melalui sistem yang tersedia di *lobby* hotel.
3. Pelanggan menerima informasi kamar dan biaya serta fasilitasnya pada sistem
4. Pelanggan melakukan pengisian biodata di dalam sistem
5. Pelanggan melakukan pemesanan dan melakukan konfirmasi pemesanan sesuai yang diinginkan oleh pelanggan
6. *Receptionist* menerima pemesanan yang telah dikonfirmasi oleh pelanggan
7. Pelanggan melakukan pembayaran ke *receptionist*.
8. *Receptionist* melakukan konfirmasi pemesanan
9. *Receptionist* memberikan bukti pembayaran dan memberikan kunci kamar yang telah dipesan oleh pelanggan.

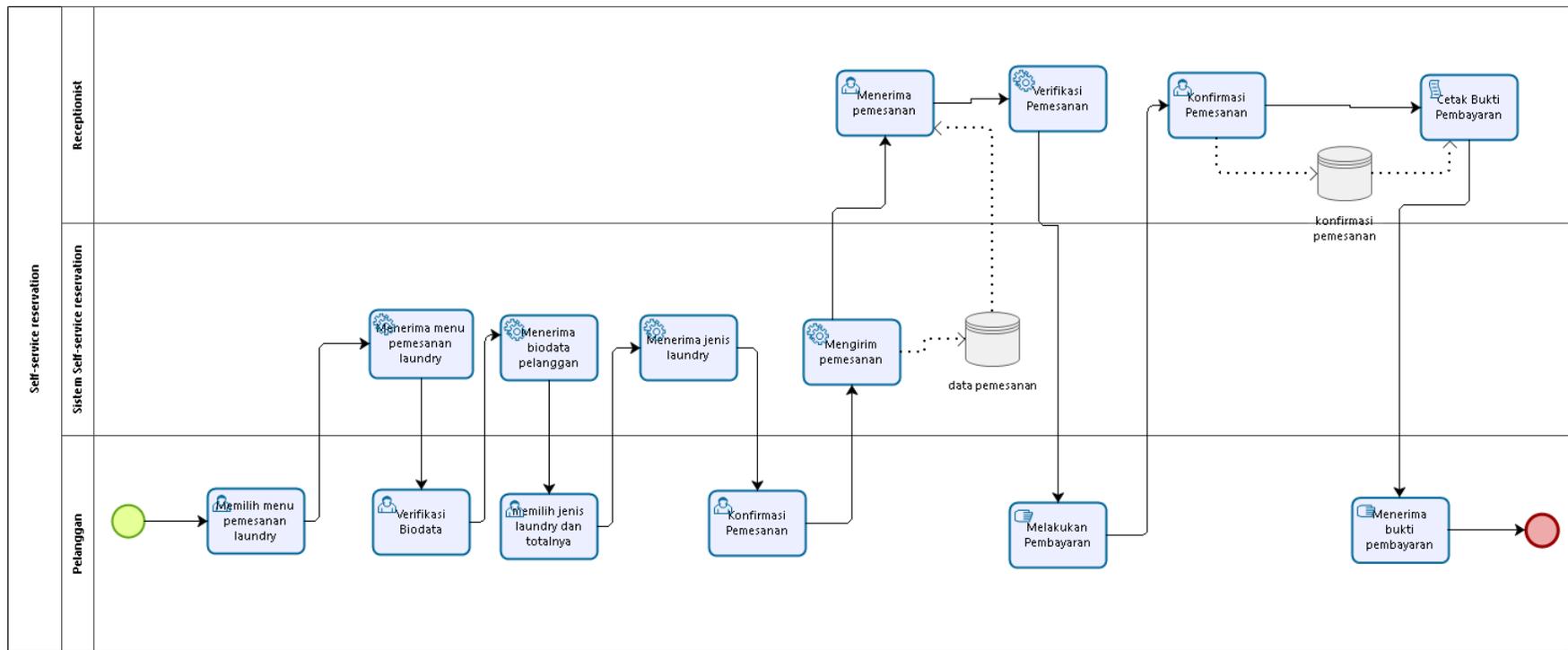
4.1.1.5 Pemodelan Proses Bisnis Pemesanan Laundry *To-Be*

Permasalahan proses bisnis *to-be* digunakan untuk memodelkan proses bisnis usulan di Everyday Smart Hotel Malang yaitu pemesanan berdasarkan solusi yang terdapat pada analisis permasalahan. Alur proses pemesanan *to-be* dijelaskan pada sesuai Gambar 4.4 adalah :

1. Pelanggan datang ke *lobby* hotel.
2. Pelanggan dapat melakukan pemesanan melalui sistem yang tersedia di *lobby* hotel.
3. Pelanggan melakukan verifikasi data pelanggan di dalam sistem
4. Pelanggan menerima informasi laundry dan biaya pada sistem
5. Pelanggan melakukan pemesanan dan melakukan konfirmasi pemesanan sesuai yang diinginkan oleh pelanggan
6. *Receptionist* menerima pemesanan yang telah dikonfirmasi oleh pelanggan
7. Pelanggan melakukan pembayaran ke *receptionist*.
8. *Receptionist* melakukan konfirmasi pemesanan
9. *Receptionist* memberikan bukti pembayaran kepada pelanggan.



Gambar 4.3 Proses Bisnis Pemesanan Kamar To-Be



Gambar 4.4 Proses Bisnis Pemesanan Laundry To-Be

4.1.2 Analisis Kebutuhan Pemangku Kepentingan

Analisis kebutuhan pemangku kepentingan dilakukan untuk mengetahui apa yang diinginkan oleh pemangku kepentingan untuk mendukung proses bisnis (tujuan, target, dan solusi) sehingga ditemukan solusi yang dapat mendukung serta target bisnis. Langkah-langkah yang dilakukan adalah sebagai berikut:

4.1.2.1 Tipe Pemangku Kepentingan

Tipe pemangku kepentingan merupakan kumpulan pemangku kepentingan yang memiliki karakteristik dan hubungan yang sama dengan sistem atau proyek yang menghasilkan sistem. Tabel 4.3 merupakan hasil dari tipe pemangku kepentingan :

Tabel 4.3 Tipe Pemangku Kepentingan

No	Tipe Pemangku Kepentingan	Deskripsi Pemangku Kepentingan
1	Pelanggan	Seseorang yang melakukan pemesanan kamar, <i>ballroom</i> dan <i>laundry</i> yang tersedia di Everyday Smart Hotel Malang.
2	<i>Receptionist</i>	Seseorang yang melayani pelanggan untuk melakukan pemesanan kamar, <i>ballroom</i> dan <i>laundry</i> serta pembayaran fasilitas lain yang tersedia di Everyday Smart Hotel.

4.1.2.2 Peran dan Perwakilan Pemangku Kepentingan

Tipe peran dan perwakilan pemangku kepentingan merupakan kumpulan pemangku kepentingan yang memiliki peran dan tanggung jawab yang sama pada proyek. Dengan mengetahui peran, maka dapat diambil perwakilan dari pemangku kebutuhan. Tabel 4.4 dapat menjelaskan masing-masing peran dan perwakilan pemangku kepentingan

Tabel 4.4 Peran dan Perwakilan Pemangku Kepentingan

Peran Pemangku Kepentingan	Deskripsi
Pelanggan	Orang yang dapat melakukan pemesanan kamar, <i>laundry</i> , dan <i>ballroom</i> di Everyday Smart Hotel
<i>Receptionist</i>	Orang yang melayani pelanggan serta dalam pembayaran dan konfirmasi pemesanan serta sebagai <i>customer service</i> di <i>front office</i> Everyday Smart Hotel

4.1.2.3 Peran Pengguna

Peran pengguna merupakan peran yang dijalankan pengguna saat melakukan interaksi dengan sistem. Tabel 4.5 merupakan peran pengguna di dalam sistem *self-service reservation* di Everyday Smart Hotel Malang

Tabel 4.5 Peran Pengguna

Peran Pengguna	Deskripsi
<i>User</i>	<i>User</i> merupakan pengguna sistem yang dapat melakukan <i>login</i> sesuai dengan hak akses
<i>Receptionist</i>	Merupakan pengguna yang dapat melakukan konfirmasi pemesanan pelanggan
Pelanggan	Merupakan pengguna yang dapat melakukan pemesanan

4.1.2.4 Kebutuhan Pemangku Kepentingan dan Pengguna

Peran pengguna merupakan peran yang dijalankan pengguna saat melakukan interaksi dengan kebutuhan pemangku kepentingan dan pengguna merupakan apa yang diinginkan dari pemangku kepentingan dan pengguna dilakukan prioritas untuk mengetahui kebutuhan yang menjadi fokus dalam penyelesaian masalah pemangku kepentingan dan pengguna. Kemudian mendefinisikan situasi saat ini yang sedang berjalan sehingga dapat mengetahui solusi dari kebutuhan. Prioritas kebutuhan pemangku kepentingan dan pengguna ditentukan berdasarkan hasil prioritas kebutuhan pemangku kepentingan dan pengguna ditentukan berdasarkan hasil prioritas kebutuhan. Tabel 4.6 dapat menjelaskan kebutuhan pemangku kepentingan dan pengguna untuk sistem *self-service reservation* pada Everyday Smart Hotel Malang

Tabel 4.6 Kebutuhan Pemangku Kepentingan dan Pengguna

Kebutuhan	Pemangku Kepentingan	Prioritas	Situasi Saat Ini	Solusi Yang Ditawarkan
Kemudahan melihat daftar kamar dan <i>ballroom</i>	Pelanggan	Mo	Daftar kamar dan <i>ballroom</i> hanya bisa dilihat oleh <i>receptionist</i>	Daftar kamar dan <i>ballroom</i> bisa dilihat oleh pelanggan

Tabel 4.6 Kebutuhan Pemangku Kepentingan dan Pengguna (lanjutan)

Kebutuhan	Pemangku Kepentingan	Prioritas	Situasi Saat Ini	Solusi Yang Ditawarkan
Menyediakan fasilitas untuk melakukan pemesanan kamar, <i>laundry</i> dan <i>ballroom</i>	Pelanggan	Mo	Pemesanan kamar, <i>laundry</i> dan <i>ballroom</i> hanya bisa dilakukan oleh <i>receptionist</i>	Pelanggan dapat melakukan pemesanan kamar, <i>laundry</i> dan <i>ballroom</i> melalui sistem.
Menyediakan fasilitas untuk melihat informasi kamar, <i>ballroom</i> , dan <i>laundry</i>	Pelanggan	Mo	<i>Receptionist</i> selalu menjelaskan berulang-ulang kepada seluruh pelanggan	Pelanggan dapat melihat informasi layanan detail kamar, <i>laundry</i> dan <i>ballroom</i> melalui sistem.
Menyediakan fasilitas untuk konfirmasi pemesanan kamar, <i>ballroom</i> dan <i>laundry</i> yang melakukan pemesanan oleh pelanggan secara <i>on the spot</i>	<i>Receptionist</i>	Mo	<i>Receptionist</i> mengelola pemesanan kamar dengan menggunakan sistem yang telah ada saat ini	<i>Receptionist</i> dapat melakukan konfirmasi pemesanan yang telah dilakukan oleh pelanggan.

4.2 User Perspective

User Perspective merupakan sudut pandang pengguna dimana didapatkan solusi bagi pengguna untuk mengatasi masalah bisnis yang ada di *business perspective*. Langkah-langkah yang dilakukan dalam *user perspective* adalah sebagai berikut :

4.2.1 Fitur

Fitur merupakan suatu kemampuan sistem yang dapat menyediakan nilai bagi pengguna. Fitur akan dilakukan prioritas menggunakan metode MosCoW untuk mengetahui fitur yang menjadi fokus dalam pengembangan sistem. Fitur sistem didapatkan berdasarkan kebutuhan pemangku kepentingan dan pengguna. Tabel 4.7 merupakan fitur dari sistem *Customer Relationship Management (CRM) Self-service reservation* pada Everyday Smart Hotel Malang

Tabel 4.7 Fitur Sistem

Kode Fitur	Fitur	Deskripsi	Prioritas
F-01	<i>Log In</i>	Sistem menyediakan fasilitas bagi <i>user</i> untuk mengakses sistem sesuai dengan hak akses.	Mo
F-02	Pemesanan Kamar	Sistem menyediakan fasilitas untuk melakukan pemesanan kamar	Mo
F-03	Pemesanan <i>Ballroom</i>	Sistem menyediakan fasilitas untuk melakukan pemesanan <i>ballroom</i>	Mo
F-04	Pemesanan <i>Laundry</i>	Sistem menyediakan fasilitas untuk melakukan pemesanan <i>laundry</i>	Mo
F-05	Infomasi <i>History</i> Transaksi	Sistem menyediakan fasilitas untuk melihat informasi <i>history</i> transaksi meliputi transaksi pemesanan	S
F-06	Konfirmasi Pemesanan	Sistem menyediakan fasilitas untuk melakukan konfirmasi pemesanan.	Mo
F-07	<i>Compatibility</i>	Sistem dapat dijalankan dari berbagai <i>browser</i> .	Mo

4.2.2 Spesifikasi Kebutuhan

Spesifikasi kebutuhan dalam sistem *Customer Relationship Management (CRM) Self-service reservation*. Berdasarkan analisis fitur sistem, maka dapat diuraikan kebutuhan fungsional dan kebutuhan non fungsional yang ada di dalam sistem. Spesifikasi kebutuhan dibagi menjadi dua yaitu kebutuhan fungsional dan non fungsional.

4.2.2.1 Kebutuhan Fungsional

Kebutuhan fungsional merupakan suatu aksi yang dapat dilakukan oleh sistem. Tabel 4.8 menjelaskan kebutuhan fungsional *user*. Tabel 4.9 menjelaskan kebutuhan fungsional pelanggan. Tabel 4.10 menjelaskan kebutuhan fungsional *receptionist*.

1. User

Tabel 4.8 Kebutuhan Fungsional User

Kode Fitur	Kode Fungsi	Nama Fungsi	Deskripsi
F-01	SKPL-F-01	<i>Log In.</i>	Sistem menyediakan fasilitas bagi <i>user</i> untuk mengakses sistem sesuai dengan hak akses.
	SKPL-F-02	<i>Log Out.</i>	Sistem menyediakan fasilitas bagi <i>user</i> untuk keluar dari sistem.

2. Pelanggan

Tabel 4.9 Kebutuhan Fungsional Pelanggan

Kode Fitur	Kode Fungsi	Nama Fungsi	Deskripsi
F-02	SKPL-F-03	Melihat daftar kamar.	Sistem menyediakan fungsi untuk melihat daftar kamar yang tersedia.
	SKPL-F-04	Melihat <i>detail</i> fasilitas kamar dan harga kamar.	Sistem menyediakan fungsi untuk melihat <i>detail</i> fasilitas kamar dan harga.
	SKPL-F-05	Melakukan pemesanan kamar.	Sistem menyediakan fungsi untuk melakukan pemesanan kamar.
F-03	SKPL-F-06	Melihat daftar <i>ballroom</i> .	Sistem menyediakan fungsi untuk melihat daftar <i>ballroom</i> yang tersedia.
	SKPL-F-07	Melihat <i>detail</i> fasilitas <i>ballroom</i> dan harganya.	Sistem menyediakan fungsi untuk melihat <i>detail</i> fasilitas <i>ballroom</i> dan harga.
	SKPL-F-08	Melakukan pemesanan <i>ballroom</i> .	Sistem menyediakan fungsi untuk melakukan pemesanan <i>ballroom</i> .
F-04	SKPL-F-09	Melihat daftar <i>laundry</i> .	Sistem menyediakan fungsi untuk melihat daftar <i>laundry</i> beserta jenisnya.
	SKPL-F-10	Melihat <i>detail</i> <i>laundry</i> dan harganya.	Sistem menyediakan fungsi untuk melihat <i>detail</i> fasilitas <i>laundry</i> dan harga.
	SKPL-F-11	Melakukan pemesanan <i>laundry</i> .	Sistem menyediakan fungsi untuk melakukan pemesanan <i>ballroom</i> .

3. Receptionist

Tabel 4.10 Kebutuhan Fungsional Receptionist

Kode Fitur	Kode Fungsi	Nama Fungsi	Deskripsi
F-05	SKPL-F-12	Melihat daftar transaksi pemesanan kamar.	Sistem menyediakan fungsi untuk melihat daftar transaksi kamar yang telah dikonfirmasi.
	SKPL-F-13	Melihat daftar transaksi pemesanan <i>ballroom</i> .	Sistem menyediakan fungsi untuk melihat daftar transaksi <i>ballroom</i> yang telah dikonfirmasi.
	SKPL-F-14	Melihat daftar transaksi pemesanan <i>laundry</i> .	Sistem menyediakan fungsi untuk melihat daftar transaksi <i>laundry</i> yang telah dikonfirmasi.
F-06	SKPL-F-15	Melihat daftar pemesanan baru.	Sistem menyediakan fungsi untuk melihat daftar pemesanan baru.
	SKPL-F-16	Melakukan konfirmasi pemesanan	Sistem menyediakan fungsi untuk melakukan konfirmasi pemesanan baru.

4.2.2.2 Kebutuhan Non-Fungsional

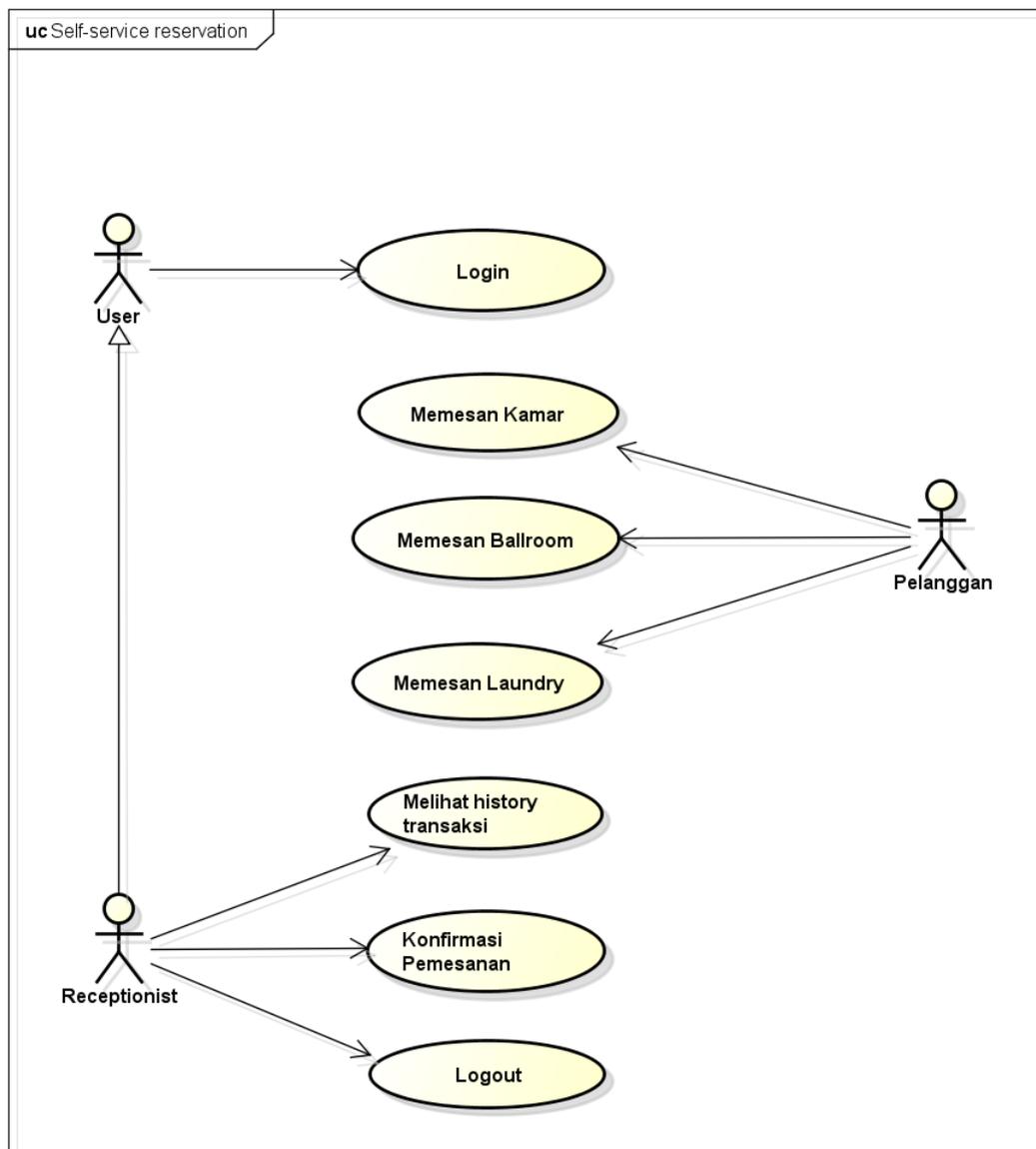
Kebutuhan non-fungsional merupakan kebutuhan yang menjelaskan kondisi sistem atau bagaimana sistem akan berjalan. Tabel 4.11 menjelaskan tentang kebutuhan non-fungsional.

Tabel 4.11 Kebutuhan Non Fungsional

Kode Fitur	Kode Fungsi	Nama Fungsi	Deskripsi
F-07	SKPL-NF-01	<i>Compatibility</i> .	Sistem dapat berjalan di berbagai <i>browser</i> yang berbeda.

4.3 Pemodelan Use Case Diagram

Use case diagram merupakan diagram *Unified Modelling Language* (UML) yang merepresentasikan aktor-aktor yang terlibat dalam sistem beserta peran (berupa *use case*) yang bisa dijalankan melalui sistem tersebut, pada *use case diagram* ini mengacu pada kebutuhan fungsionalitas sistem yang telah peneliti jabarkan sebelumnya. Kemudian dipetakan berdasarkan aktor yang dapat melakukan fungsionalitas tersebut. Tiap *use case* yang didefinisikan akan dijelaskan lebih *detail* menggunakan *use case scenario*. *Use case scenario* akan dijelaskan dalam bentuk tabel yang berisi nama *use case*, tujuan *use case*, aktor yang terlibat, skenario utama, dan skenario alternatif. Berikut ini merupakan *usecase diagram* yang ditunjukkan pada Gambar 4.3 berikut ini :



powered by Astah

Gambar 4.5 Use Case Diagram

Berikut penjelasan dari diagram *use-case* pada Gambar 4.5 :

Pada *usecase login* aktor dapat melakukan *login* untuk mendapatkan otorisasi sebagai *receptionist*. Selanjutnya, pada *usecase* memesan kamar aktor dapat melakukan transaksi pemesanan kamar yang tersedia di dalam sistem. Kemudian, pada *usecase* memesan *ballroom* aktor dapat melakukan pemesanan *ballroom* yang terdapat 3 *ballroom* yang masing-masing memiliki fasilitas yang berbeda. Selanjutnya pada *usecase* memesan *laundry*, aktor dapat melakukan pemesanan *laundry* yang sesuai kebutuhan dan sistem menampilkan informasi masing-masing harga yang disediakan oleh Everyday Smart Hotel Malang.

Pada *usecase* melihat data pelanggan, aktor dapat melihat keseluruhan pelanggan yang telah melakukan pemesanan beserta data diri pelanggan. Kemudian pada *usecase* melihat *history* transaksi, aktor dapat melihat seluruh

transaksi yang telah dilakukan oleh pelanggan yang terdiri transaksi pemesanan kamar, *laundry* dan *ballroom*. Selanjutnya pada *usecase* konfirmasi pemesanan, aktor dapat melakukan validasi pemesanan yang dilakukan oleh *receptionist*, ketika pelanggan telah melakukan pembayaran sesuai dengan transaksinya yang dikonfirmasi. Kemudian pada *usecase logout*, aktor dapat keluar dari hak akses sistem.

4.3.1 Skenario Use Case

Berikut ini merupakan skenario dari masing-masing *use case* yang telah dijabarkan sebelumnya pada Gambar 4.3 mengenai *usecase diagram* sistem *self service reservation* pada Everyday Smart Hotel Malang :

Tabel 4.12 Skenario Use Case Login

<i>Flow of Events</i> untuk use case melakukan <i>login</i>	
<i>Brief Description</i>	Use case ini menggambarkan bagaimana aktor dapat <i>login</i> untuk mendapatkan hak akses ke dalam sistem
<i>Actor</i>	<i>Receptionist</i>
<i>Pre-conditions</i>	<ol style="list-style-type: none"> 1. Aktor telah memiliki akun dan telah terdaftar di dalam sistem 2. Mengakses halaman <i>login receptionist</i>
<i>Basic flow of Events</i>	<ol style="list-style-type: none"> 1. <i>Use case</i> di mulai ketika aktor mengunjungi halaman website <i>receptionist</i> 2. Aktor memasukkan <i>email</i> dan <i>password</i> 3. Aktor menjalankan fungsi <i>login</i> 4. Sistem melakukan autentikasi dan menampilkan halaman awal <i>receptionist</i>
<i>Alternative Flows</i>	<ol style="list-style-type: none"> 1. <i>Email</i> dan <i>password</i> salah <p>Jika, di <i>basic flow</i>, <i>guest</i> memasukkan <i>email</i> dan <i>password</i> tidak valid, maka sistem akan memberikan informasi dan kembali kehalaman <i>login</i></p>
<i>Subflow</i>	Tidak ada
<i>Key Scenarios</i>	Aktor berhasil masuk ke dalam sistem
<i>Post-conditions</i>	Aktor masuk ke dalam sistem dan akan di- <i>redirect</i> ke halaman <i>home receptionist</i>

Tabel 4.13 Skenario Use Case Pemesanan Kamar

<i>Flow of Events</i> untuk <i>use case</i> melakukan pemesanan kamar	
<i>Brief Description</i>	<i>Use case</i> ini menggambarkan bagaimana aktor dapat melakukan pemesanan kamar yang tersedia di Everyday Smart Hotel Malang
<i>Actor</i>	Pelanggan
<i>Pre-conditions</i>	Kamar tersedia
<i>Basic flow of Events</i>	<ol style="list-style-type: none"> 1. Aktor memilih fungsi menu pemesanan kamar 2. Aktor menentukan jadwal <i>check-in</i> dan <i>check-out</i> kemudian memilih fungsi pemesanan jadwal. 3. Sistem menampilkan daftar kamar yang tersedia di Everyday Smart Hotel. 4. Aktor memilih kamar dan detail kamar yang diinginkan, lalu aktor memilih fungsi pemesanan kamar. 5. Sistem menampilkan <i>form</i> biodata pelanggan 6. Aktor mengisi biodata pada <i>form</i> biodata pelanggan, lalu menjalankan fungsi menyimpan biodata 7. Sistem menampilkan keseluruhan informasi transaksi yang dipilih oleh aktor 8. Aktor melakukan konfirmasi sesuai pada transaksi, lalu memilih fungsi konfirmasi transaksi kamar. 9. Sistem menampilkan informasi pemesanan kamar dan menampilkan <i>ID Customer</i>
<i>Alternative Flows</i>	<ol style="list-style-type: none"> 1. Aktor mengisi biodata pada <i>form</i> tidak lengkap Pada pengisian biodata, apabila aktor tidak mengisi salah satu data yang harus terisi maka sistem akan menampilkan pesan bahwa terdapat salah satu data yang belum terisi. 2. Kamar tidak tersedia Jika kamar tidak tersedia pada waktu yang telah ditentukan oleh pelanggan, maka sistem menampilkan informasi bahwa “Kamar tidak tersedia”
<i>Subflow</i>	Tidak ada

<i>Key Scenarios</i>	<ol style="list-style-type: none"> 1. Aktor berhasil melakukan pemesanan kamar 2. Kamar telah di pesan lebih dari satu
<i>Post-conditions</i>	<ol style="list-style-type: none"> 1. Aktor berhasil melakukan memesan kamar 2. Data pemesanan bertambah 3. Kamar yang tersedia berkurang

Tabel 4.14 Skenario Use Case Pemesanan *Ballroom*

<i>Flow of Events</i> untuk <i>use case</i> melakukan pemesanan <i>ballroom</i>	
<i>Brief Description</i>	<i>Use case</i> ini menggambarkan bagaimana aktor dapat melakukan pemesanan <i>ballroom</i>
<i>Actor</i>	Pelanggan
<i>Pre-conditions</i>	<i>Ballroom</i> tersedia
<i>Basic flow of Events</i>	<ol style="list-style-type: none"> 1. <i>Use case</i> di mulai ketika aktor memilih menu pemesanan <i>ballroom</i> 2. Aktor menentukan jadwal <i>check-in</i> dan <i>check-out</i> kemudian memilih fungsi pemesanan jadwal 3. Sistem menampilkan daftar <i>ballroom</i> yang tersedia di Everyday Smart Hotel 4. Aktor memilih <i>ballroom</i> dan detail <i>ballroom</i> yang diinginkan, lalu memilih fungsi pemesanan <i>ballroom</i> 5. Sistem menampilkan <i>form</i> biodata pelanggan 6. Aktor mengisi biodata pada <i>form</i> biodata pelanggan, lalu memilih fungsi penyimpanan biodata 7. Sistem menampilkan keseluruhan informasi transaksi yang dipilih oleh aktor 8. Aktor melakukan konfirmasi sesuai pada transaksi, lalu memilih fungsi konfirmasi pemesanan <i>ballroom</i>
<i>Alternative Flows</i>	<ol style="list-style-type: none"> 1. Aktor mengisi biodata pada <i>form</i> tidak lengkap <p>Pada pengisian biodata, apabila aktor tidak mengisi salah satu data yang harus terisi maka sistem akan menampilkan pesan bahwa terdapat salah satu data yang belum terisi.</p>

	<p>2. <i>Ballroom</i> tidak tersedia</p> <p>Jika <i>ballroom</i> tidak tersedia pada waktu yang telah ditentukan oleh pelanggan, maka sistem menampilkan informasi bahwa "<i>Ballroom</i> tidak tersedia"</p>
<i>Subflow</i>	Tidak ada
<i>Key Scenarios</i>	<ol style="list-style-type: none"> 1. Aktor berhasil melakukan pemesanan <i>ballroom</i> 2. <i>Ballroom</i> telah di pesan lebih dari satu hari
<i>Post-conditions</i>	<ol style="list-style-type: none"> 1. Aktor berhasil melakukan memesan <i>ballroom</i> 2. Data pemesanan bertambah 3. <i>Ballroom</i> yang tersedia berkurang

Tabel 4.15 Skenario Use Case Pemesanan Laundry

<i>Flow of Events</i> untuk use case melakukan pemesanan Laundry	
<i>Brief Description</i>	<i>Use case</i> ini menggambarkan bagaimana aktor dapat melakukan pemesanan laundry
<i>Actor</i>	Pelanggan
<i>Pre-conditions</i>	<ol style="list-style-type: none"> 1. Aktor memiliki ID <i>Customer</i> 2. <i>Login</i> status sukses
<i>Basic flow of Events</i>	<ol style="list-style-type: none"> 1. Use case di mulai ketika aktor memilih fungsi menu pemesanan laundry 2. Aktor memasukkan ID <i>Customer</i> dan memilih fungsi validasi pelanggan 3. Sistem menampilkan seluruh detil laundry dengan dua jenis yaitu <i>Ladies</i> dan <i>Gentlements</i> 4. Aktor memilih dan menentukan jumlah laundry 5. Sistem menampilkan perhitungan biaya keseluruhan 6. Aktor memilih fungsi pemesanan laundry 7. Sistem menampilkan informasi transaksi yang dilakukan oleh aktor. 8. Aktor memilih fungsi konfirmasi pemesanan laundry
<i>Alternative Flows</i>	Tidak ada
<i>Subflow</i>	Tidak ada

<i>Key Scenarios</i>	1. Aktor berhasil melakukan pemesanan <i>laundry</i>
<i>Post-conditions</i>	1. Aktor berhasil melakukan memesan <i>laundry</i> 2. Data pemesanan bertambah

Tabel 4.16 Skenario Use Case Melihat *History* Transaksi

<i>Flow of Events</i> untuk use case melihat <i>history</i> transaksi	
<i>Brief Description</i>	<i>Use case</i> ini menggambarkan bagaimana aktor dapat melihat seluruh informasi transaksi pemesanan
<i>Actor</i>	<i>Receptionist</i>
<i>Pre-conditions</i>	<i>Login</i> status sukses
<i>Basic flow of Events</i>	<ol style="list-style-type: none"> 1. <i>Use case</i> dimulai ketika aktor berhasil <i>login</i> ke sistem 2. Aktor memilih menu fungsi pemesanan 3. Sistem menampilkan pilihan <i>reservation</i> 4. Aktor memilih fungsi melihat <i>history</i> transaksi 5. Sistem menampilkan informasi seluruh transaksi yang telah dilakukan
<i>Alternative Flows</i>	Tidak ada
<i>Subflow</i>	Tidak ada
<i>Key Scenarios</i>	1. Aktor berhasil melihat sleuruh informasi transaksi pemesanan
<i>Post-conditions</i>	Aktor berhasil seluruh melihat informasi transaksi

Tabel 4.17 Skenario Use Case Konfirmasi Pemesanan

<i>Flow of Events</i> untuk use case melakukan konfirmasi pemesanan	
<i>Brief Description</i>	<i>Use case</i> ini menggambarkan bagaimana aktor dapat melakukan konfirmasi pemesanan
<i>Actor</i>	<i>Receptionist</i>
<i>Pre-conditionz</i>	<ol style="list-style-type: none"> 1. <i>Login</i> status sukses 2. Pelanggan telah melakukan pemesanan
<i>Basic flow of Events</i>	<ol style="list-style-type: none"> 1. <i>Use case</i> dimulai ketika aktor berhasil <i>login</i> ke sistem 2. Aktor memilih fungsi <i>reservation</i>

	<ol style="list-style-type: none"> 3. Sistem menampilkan pilihan <i>reservation</i> 4. Aktor memilih fungsi <i>order</i> yang baru dimasukkan oleh pelanggan 5. Sistem menampilkan informasi bahwa terdapat pemesanan baru 6. Aktor memilih fungsi konfirmasi pemesanan 7. Sistem menampilkan data pemesanan pada halaman <i>reservation history</i>
<i>Alternative Flows</i>	Tidak ada
<i>Subflow</i>	Tidak ada
<i>Key Scenarios</i>	<ol style="list-style-type: none"> 1. <i>Receptionist</i> dapat melakukan validasi pemesanan
<i>Post-conditions</i>	<ol style="list-style-type: none"> 1. Aktor berhasil melakukan memesan kamar 2. Data pemesanan bertambah 3. Kamar yang tersedia berkurang

Tabel 4.18 Use Case Skenario Logout

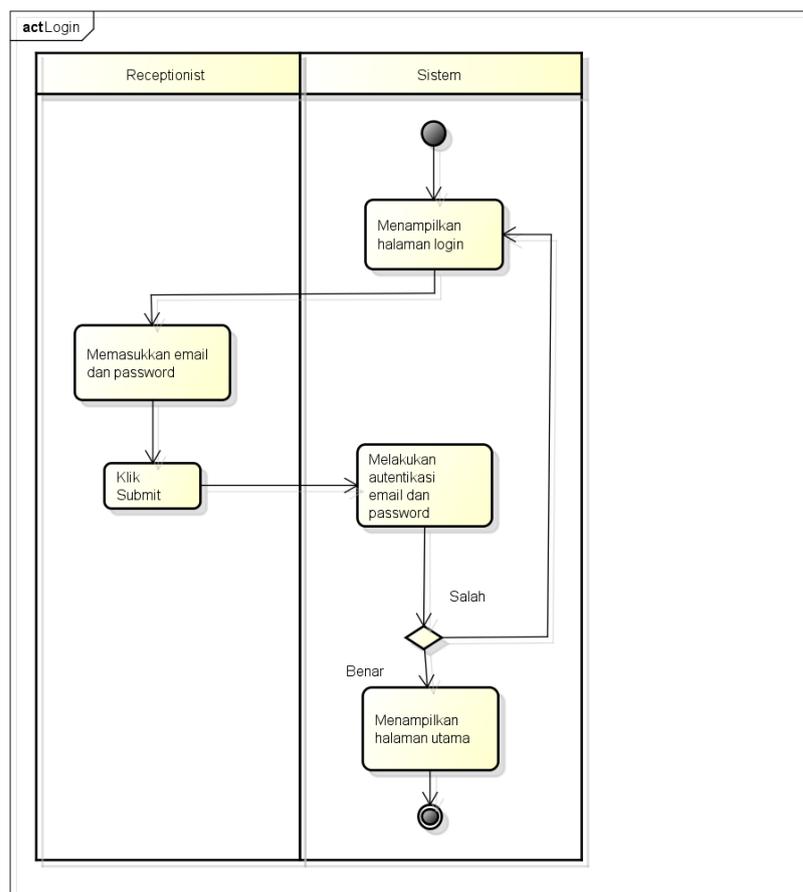
<i>Flow of Events</i> untuk use case melakukan <i>logout</i>	
<i>Brief Description</i>	<i>Use case</i> ini menggambarkan bagaimana aktor dapat melakukan konfirmasi pemesanan
<i>Actor</i>	<i>Receptionist</i>
<i>Pre-conditionz</i>	<ol style="list-style-type: none"> 1. <i>Login</i> status sukses 2. Aktor berada di dalam sistem
<i>Basic flow of Events</i>	<ol style="list-style-type: none"> 1. Aktor memilih fungsi <i>logout</i> 2. Sistem menampilkan halaman awal <i>receptionist</i>
<i>Alternative Flows</i>	Tidak ada
<i>Subflow</i>	Tidak ada
<i>Key Scenarios</i>	Aktor berhasil keluar dari sistem
<i>Post-conditions</i>	Aktor keluar dari sistem

4.4 Pemodelan Activity Diagram

Perancangan *activity diagram* berisi penggambaran aliran kerja dari sistem. Berikut ini merupakan *activity diagram* sistem *self-service reservation* Everyday Smart Hotel Malang :

1. Activity Diagram Login

Activity diagram login dimulai ketika aktor mengakses halaman *login* dan sistem menampilkan halaman *login*, kemudian aktor mengisi *email* dan *password* dan sistem mengecek masukan dari aktor jika data yang dimasukkan oleh aktor benar maka aktor akan masuk ke dalam sistem. Berikut ini merupakan *activity diagram login* yang ditunjukkan pada Gambar 4.6 dibawah ini:



powered by Astah

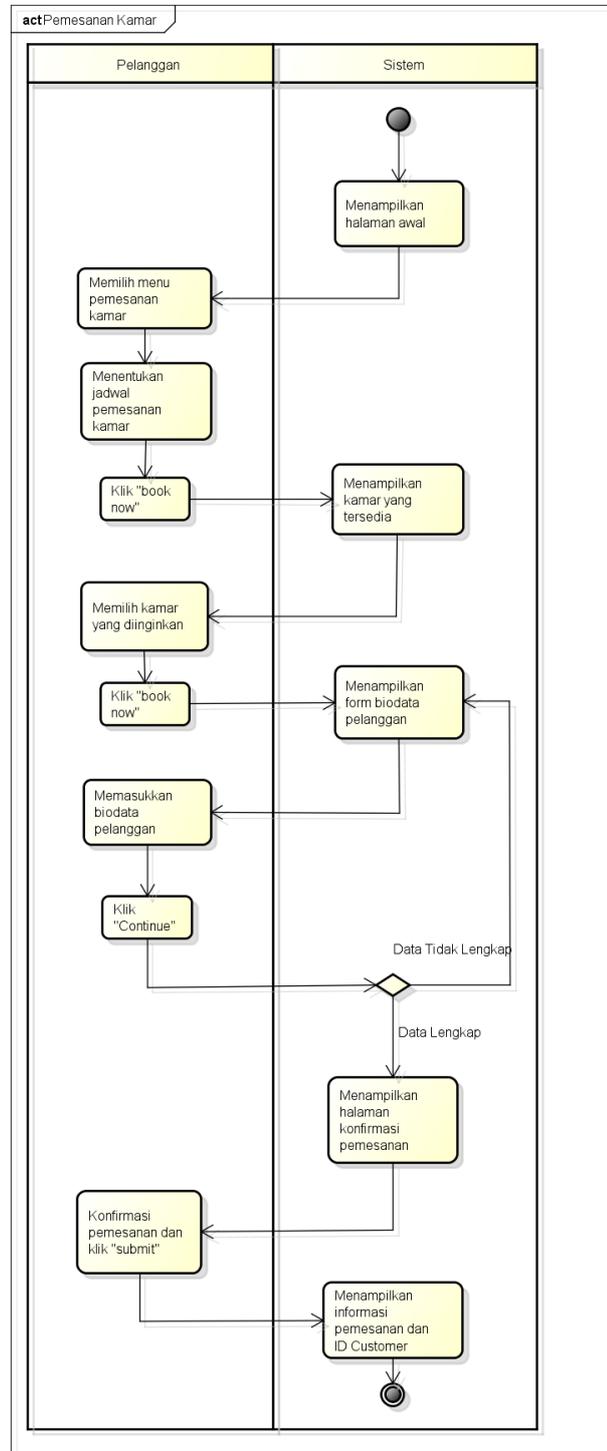
Gambar 4.6 Activity Diagram Login

Gambar 4.6 menjelaskan bahwa proses *login* dimulai pada aktor masuk ke dalam halaman *receptionist* dan sistem menampilkan *form login* yaitu *email* dan *password*, *receptionist* memasukkan *email* dan *password* serta menekan tombol submit untuk masuk ke dalam sistem.

2. Activity Diagram Pemesanan Kamar

Activity diagram pemesanan kamar dimulai ketika aktor mengakses halaman web dan aktor memilih menu pemesanan kamar pada sistem dan sistem

menampilkan daftar kamar yang ada di Everyday Smart Hotel. Berikut ini merupakan *activity diagram* pemesanan kamar yang ditunjukkan pada Gambar 4.7 dibawah ini :



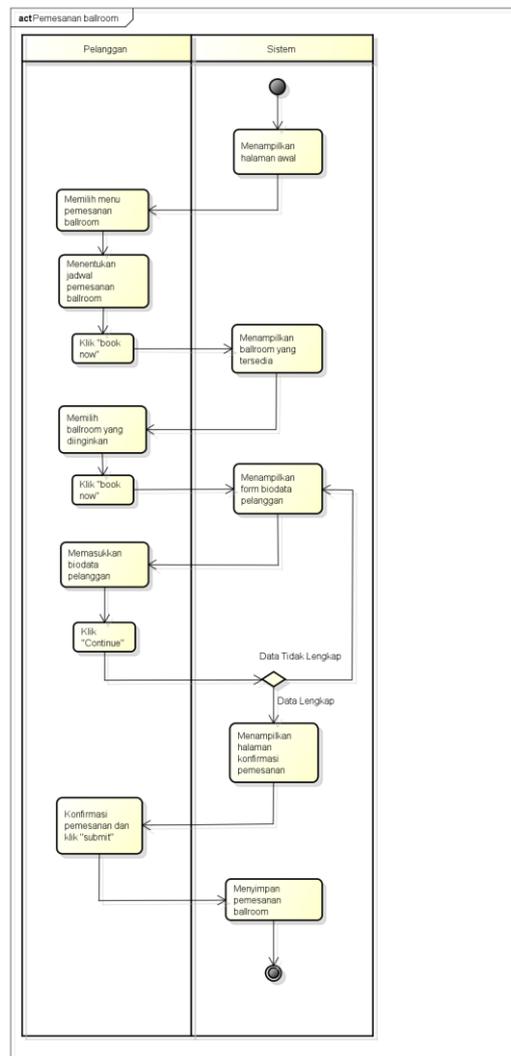
Gambar 4.7 Activity Diagram Pemesanan Kamar

Gambar 4.7 menjelaskan bahwa proses pemesanan dimulai dari aktor memilih menu pemesanan kamar, memilih jadwal dan memilih fungsi pemesanan.

Kemudian sistem menampilkan seluruh kamar yang tersedia pada jadwal yang dipilih oleh aktor, lalu aktor memasukkan biodata pada *form* yang tersedia dan melakukan konfirmasi pemesanan serta aktor mendapatkan *ID Customer*.

3. Activity Diagram Pemesanan Ballroom

Activity diagram pemesanan *ballroom* dimulai ketika aktor mengakses halaman web dan aktor memilih menu pemesanan *ballroom* pada sistem dan sistem menampilkan daftar *ballroom* yang ada di Everyday Smart Hotel. Berikut ini merupakan *activity diagram* pemesanan *ballroom* yang ditunjukkan pada Gambar 4.8 dibawah ini :

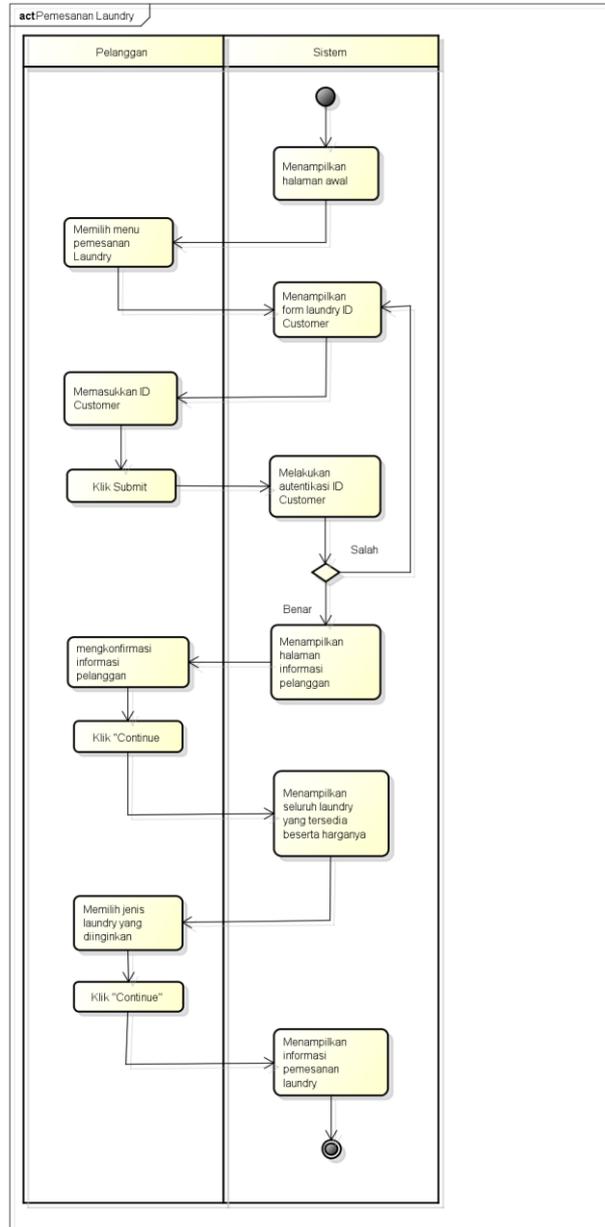


Gambar 4.8 Activity Diagram Pemesanan Ballroom

Gambar 4.8 menjelaskan bahwa proses pemesanan *ballroom* dimulai dari aktor memilih menu pemesanan *ballroom*, memilih jadwal dan memilih fungsi pemesanan. Kemudian sistem menampilkan seluruh *ballroom* yang tersedia pada jadwal yang dipilih oleh aktor, lalu aktor memasukkan biodata pada form yang tersedia dan melakukan konfirmasi pemesanan.

4. Activity Diagram Pemesanan Laundry

Activity diagram pemesanan laundry dimulai ketika aktor mengakses halaman web dan aktor memilih menu pemesanan laundry pada sistem dan sistem menampilkan daftar harga laundry yang ada di Everyday Smart Hotel. Berikut ini merupakan *activity diagram* pemesanan laundry yang ditunjukkan pada Gambar 4.9 dibawah ini:

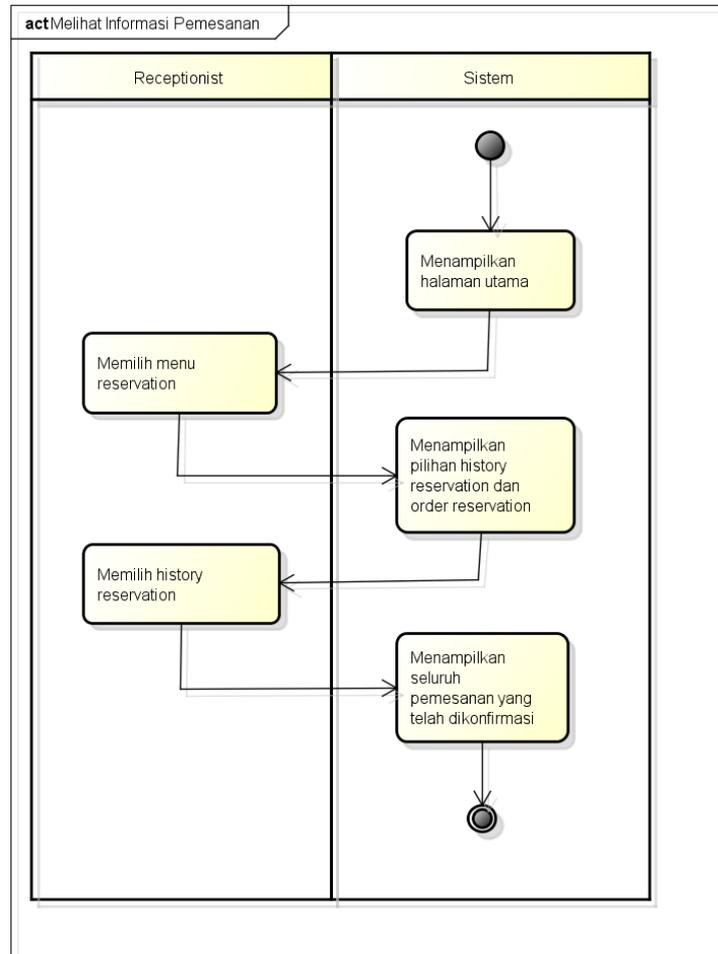


Gambar 4.9 Activity Diagram Pemesanan Laundry

Gambar 4.9 menjelaskan bahwa proses pemesanan laundry dimulai dari aktor memilih menu pemesanan laundry, lalu aktor memasukkan ID Customer yang dapat digunakan untuk melakukan pemesanan laundry sesuai kebutuhan beserta harganya dan konfirmasi pemesanan laundry.

5. Activity Diagram Melihat *History* Transaksi

Activity diagram melihat *history* transaksi dimulai ketika aktor mengakses halaman web dan aktor memilih menu transaksi pada sistem. berikut ini merupakan *activity diagram* melihat *history* pemesanan yang ditunjukkan pada Gambar 4.10 dibawah ini

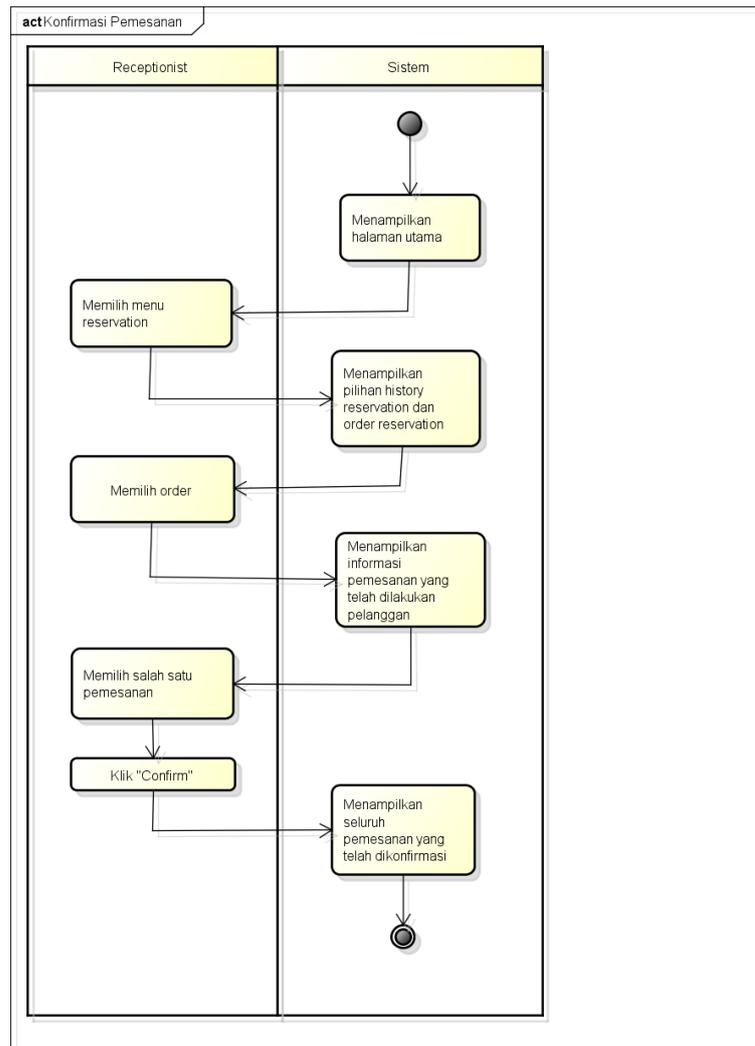


Gambar 4.10 Activity Diagram Melihat *History* Transaksi

Gambar 4.10 menjelaskan bahwa melihat *history* transaksi pemesanan dimulai dari *receptionist* masuk ke dalam sistem dan memilih menu *reservation*, kemudian *receptionist* memilih menu *history reservation* untuk melihat seluruh pemesanan yang telah dilakukan dan dikonfirmasi oleh *receptionist*

6. Activity Diagram Konfirmasi Pemesanan

Activity diagram melakukan konfirmasi pemesanan dimulai ketika aktor mengakses halaman web dan aktor memilih menu order pemesanan pada sistem. Berikut ini merupakan *activity diagram* konfirmasi pemesanan yang ditunjukkan pada Gambar 4.11 dibawah ini



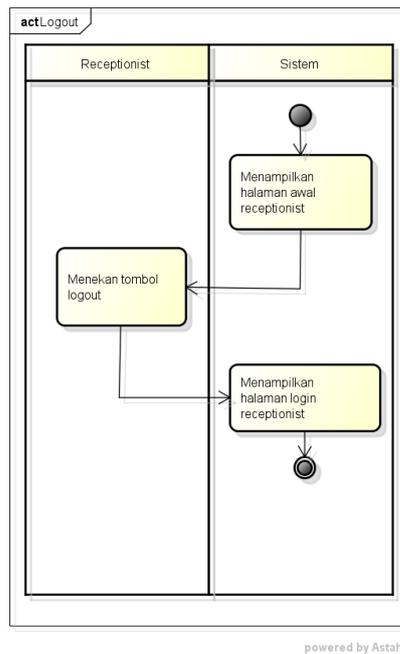
powered by Astah

Gambar 4.11 Activity Diagram Konfirmasi Pemesanan

Gambar 4.11 menjelaskan bahwa konfirmasi pemesanan dimulai dari *receptionist* masuk ke dalam sistem dan memilih menu *reservation*, kemudian *receptionist* memilih menu *orders* untuk melihat seluruh pemesanan yang telah dilakukan oleh pelanggan tetapi belum dikonfirmasi, untuk melakukan konfirmasi *receptionist* dapat menekan tombol *confirm* yang terdapat di halaman *order* untuk mengkonfirmasi pemesanan.

7. Activity Diagram Logout

Activity diagram melakukan *logout* dimulai ketika aktor mengakses halaman berada di dalam sistem. Berikut ini merupakan *activity diagram logout* yang ditunjukkan pada Gambar 4.12 dibawah ini

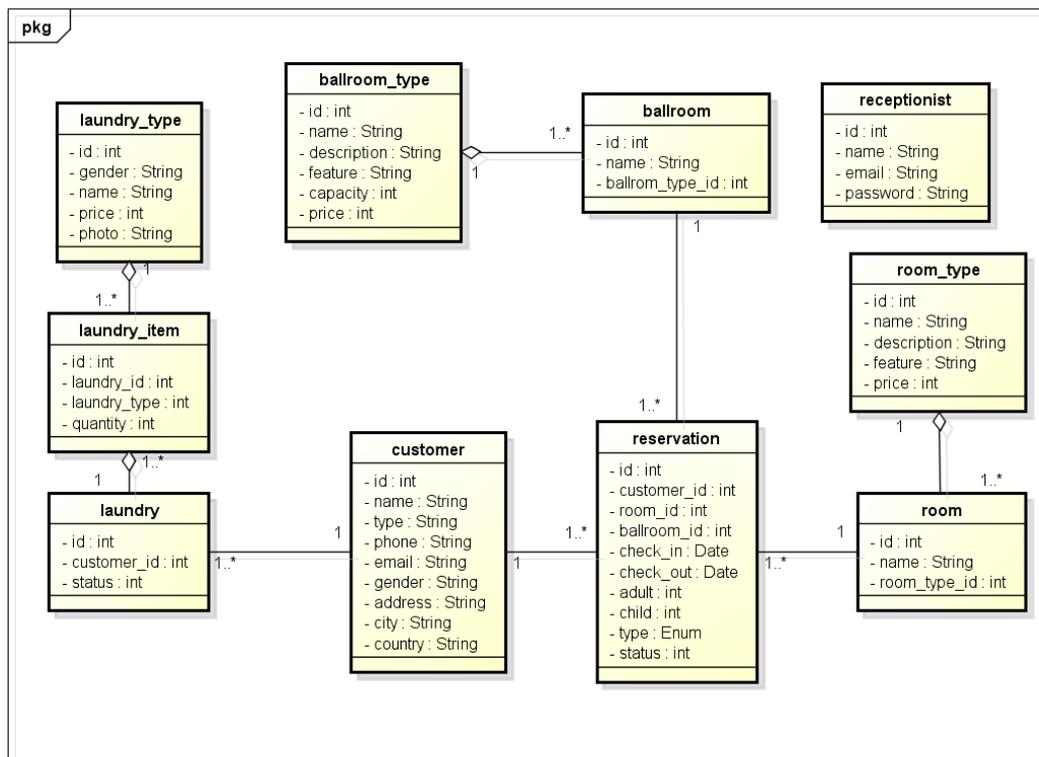


Gambar 4.12 Activiry Diagram Logout

Gambar 4.12 menjelaskan bahwa konfirmasi pemesanan dimulai dari *receptionist* berada di dalam sistem dan memilih fungsi *logout*, kemudian sistem akan kembali menampilkan halaman *login receptionist*.

4.5 Analysis-Level Class Diagram

Class diagram digunakan untuk merepresentasikan *class*, atribut, fungsi, beserta hubungan antar *class* yang ada pada sistem *Customer Relationship Management (CRM) Self-service reservation*. Pada level analisis, *class diagram* hanya menggambarkan *class* beserta atribut. Fungsi dalam *class diagram* dimasukkan pada level perancangan. Gambar 4.13 merupakan *analysis-level class diagram* dari sistem *Customer Relationship Management (CRM) Self-service reservation*.



powered by Astah

Gambar 4.13 Analysis-Level Class Diagram

Pada Gambar 4.13 menjelaskan bahwa *Analysis-Level Class Diagram* untuk implementasi sistem *Customer Relationship Management (CRM) Self-service Reservation* memiliki 10 class yang terdiri dari class *laundry_type*, *laundry_item* dan *laundry* berfungsi untuk pemesanan *laundry* yang terhubung dengan class *customer* dikarenakan pemesanan *laundry* dapat digunakan setelah mempunyai *ID Customer*. Kemudian pada class *reservation* yang terhubung dengan class *ballroom* dan *ballroom_type* berfungsi untuk melakukan pemesanan *ballroom* pada sistem. Selanjutnya class *reservation* terhubung dengan class *room* dan *room_type_id* berfungsi untuk melakukan pemesanan pada sistem.