

**APLIKASI ONLINE PENCATATAN TRANSAKSI KAMAR HOTEL
BERBASIS *ANDROID*
(STUDI KASUS: MOTTOMO INN)**

LAPORAN KERJA PRAKTIK

Ditulis untuk Memenuhi Persyaratan Mata Kuliah kerja Praktik

(INF2417401)

Disusun oleh:

Christian Micah Ranti

(21013015)



**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS KATOLIK DE LA SALLE MANADO
MANADO
2024**

**APLIKASI ONLINE PENCATATAN TRANSAKSI KAMAR HOTEL
BERBASIS *ANDROID*
(STUDI KASUS: MOTTOMO INN)**

LAPORAN KERJA PRAKTIK

Ditulis untuk Memenuhi Persyaratan Mata Kuliah kerja Praktik

(INF2417401)

Disusun oleh:

Christian Micah Ranti

(21013015)



**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS KATOLIK DE LA SALLE MANADO
MANADO
2024**

**LEMBAR PENGESAHAN
LAPORAN KERJA PRAKTIK**

Judul:
**APLIKASI ONLINE PENCATATAN TRANSAKSI KAMAR HOTEL
BERBASIS *ANDROID***

STUDI KASUS: MOTTOMO INN

Telah disetujui dan disahkan pada tanggal: 9 Desember 2024

Oleh:
MOTTOMO INN

The image shows a handwritten signature in red ink over the Mottomo Inn logo. The signature is written in a cursive style and appears to be 'Olivia Kasiman'. The logo itself consists of the word 'mottomo' in a red, lowercase, sans-serif font, with 'inn' in a smaller, red, lowercase, sans-serif font below it. A horizontal line with dots at both ends passes through the signature and the logo.

Olivia Kasiman
HOTEL MANAGER MOTTOMO INN

LEMBAR PERNYATAAN

Yang bertanda tangan dibawah ini

Nama : Christian Micah Ranti
NIM : 21013015
Tempat/Tanggal Lahir : Manado, 22 juli 2003
Fakultas/Program Studi : Teknik/Informatika

Menyatakan bahwa Laporan Kerja Praktik dan Aplikasi dengan judul "APLIKASI ONLINE PENCATATAN TRANSAKSI KAMAR HOTEL BERBASIS ANDROID (STUDI KASUS: MOTTOMO INN)" yang telah saya buat adalah benar hasil karya tulis saya dan bukan hasil karya tulis, kecuali dalam bentuk kutip yang telah disebutkan sumbernya.

Demikian surat pernyataan ini saya buat dengan sebenar-benarnya dan apabila ini tidak benar maka saya siap untuk menerima sanksi akademis sesuai dengan yang ditetapkan oleh fakultas, berupa pembatalan Kerja Praktik dan Hasilnya.

Manado, 16 Desember 2024

Yang Menyatakan,



Christian Micah Ranti

Menyetujui,

Dosen Pembimbing I

Dr. Liza Wikarsa, B.C.S., M.Comp

Dosen Pembimbing II

Steven Pandelaki, S.T., M.Sc

Ketua Program Studi

Vivie Deyby Kumenap, S.T., M.C.S

Mengetahui



Dekan Fakultas Teknik

Rachmadi, S.T., M.T



**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS KATOLIK DE LA SALLE MANADO**

FORM KP - 003

FORMULIR DATA UMUM PERUSAHAAN

NAMA MAHASISWA : Christian Micah Ranti
NIM : 21013015
NAMA PERUSAHAAN : PT. Manado Bungalo Sentosa
ALAMAT PERUSAHAAN : Jl. Sam Ratulangi No. 222 Ranotana Manado
DIDIRIKAN TAHUN : 2019
IZIN USAHA : 121222018758
BIDANG BISNIS : Jasa Perhotelan
JUMLAH KARYAWAN : 6
PEMILIK : Gliff Charles Kurniawan
DEWAN DIREKTUR : Lusye Kurniawan

WAKIL PERUSAHAAN

Tanggal : 9 Desember 2024
Nama : Olivia Kasiman
Jabatan : *Hotel Manager*

(Tanda tangan dan
Cap perusahaan)

: 



PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS KATOLIK DE LA SALLE MANADO

FORM KP - 004




FORMULIR PENILAIAN KEMAJUAN KERJA PRAKTEK

A. UMUM

NIM Mahasiswa : 21013015
Program Studi : Informatika
Dosen Pembimbing Akademik : Anthoni Rivai Pulakiang, S.Pd., M,Kom.
Topik/Rencana Bidang : Pemrograman Android
Pembimbing 1 : Dr. Liza Wikarsa, B.C.S., M.Comp.
Pembimbing 2 : Steven Pandelaki, S.T., M.Sc.
Terhitung Mulai : 7 Juni 2024
Target Selesai : 17 Juli 2024

B. KEGIATAN PELAKSANAAN KERJA PRAKTEK

No	Tanggal	Jenis Kegiatan	Paraf Pembimbing
1	11 Agustus 2024	Topik dan Lingkup KP	
2	12 Agustus 2024	Revisi Proposal KP	
3	11 Agustus 2024	Topik dan Lingkup KP	
4	12 Agustus 2024	Revisi Proposal KP	
5	19 Agustus 2024	Bab I – Bab III	
6	26 Agustus 2024	Bab IV – Fase Analisa	
7	19 Agustus 2024	Bab I – Bab III	
8	26 Agustus 2024	Bab IV – Fase Analisa	
9	2 september 2024	Bab IV – Revisi Fase Analisa dan Desain	
10	9 september 2024	Bab IV – Revisi Fase Desain Implementasi	
11	2 september 2024	Bab IV – Revisi Fase Analisa dan Desain	
12	9 september 2024	Bab IV – Revisi Fase Desain Implementasi	
13	16 september 2024	Revisi Fase Implementasi	
14	23 september 2024	Bab V	
15	16 september 2024	Revisi Fase Implementasi	
16	23 september 2024	Bab V	
17	7 oktober 2024	Demo Aplikasi dan Pengecekan Laporan	

18	17 oktober 2024	Revisi Laporan Akhir	
19	4 november 2024	Persiapan sidang KP	
20	2 desember 2024	Slide Presentasi	

Manado, 9 Desember 2024

Dosen Pembimbing KP



Dr. Liza Wikarsa, B.C.S., M.Comp.



**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS KATOLIK DE LA SALLE MANADO**

FORM KP - 005

FORMULIR PENILAIAN PELAKSANAAN KERJA PRAKTIK

NAMA MAHASISWA : Christian Micah Ranti
NIM : 21013015
NAMA PERUSAHAAN : PT. Manado Bungalo Sentosa
ALAMAT PERUSAHAAN : Jl. Sam Ratulangi No. 222 Ranotana Manado
TGL KERJA PRAKTEK : 7 Juni 2024 – 17 Juli 2024
TOPIK YANG DIBAHAS : Aplikasi *Online* Pencatatan Transaksi Kamar Hotel

Kerajinan =

50	60	70	80	90	100
----	----	----	----	----	-----

Kerajinan =

50	60	70	80	90	100
----	----	----	----	----	-----

Kerajinan =

50	60	70	80	90	100
----	----	----	----	----	-----

NILAI RATA-RATA : 95
Tanggal : 9 Desember 2024
NAMA PENILAI : Olivia Kasiman
JABATAN : *Hotel Manager*

(Tanda tangan dan cap perusahaan)

: 

KATA PENGANTAR

Pertama-tama, penulis mengucapkan syukur dan terima kasih kepada Tuhan Yang Maha Esa atas pertolongan dan rahmat-Nya sehingga dapat menyelesaikan kerja praktik di Mottomo Inn. Selama pembuatan Aplikasi Online Pencatatan dan penyusunan laporan menerima banyak arahan dan bantuan dari berbagai pihak, oleh sebab itu Penulis ingin mengucapkan terima kasih kepada:

1. Prof.Dr.Johanis Ohoitumur selaku Rektor dari Universitas Katolik De La Salle Manado.
2. Bapak Ronald Albert Rachmadi, S.T., M.T., selaku Dekan Fakultas Teknik.
3. Ibu Vivie D. Kumendap, S.T., M.C.S, selaku Ketua Program Studi Teknik
4. Informatika.
5. Bapak Anthoni Rivai Pulakiang S.Pd., M.Kom. Selaku Dosen Pembimbing Akademik.
6. Dr. Liza Wikarsa, B.C.S., M.Comp. Selaku Dosen Pembimbing I yang telah memberikan arahan dan saran selama pengerjaan.
7. Bapak Steven Pandelaki, S.T., M.Sc., Selaku Dosen Pembimbing II yang telah memberikan arahan dan saran selama pengerjaan.
8. Ibu Olivia Kasiman yang telah mengizinkan melakukan pelaksanaan magang dan Kerja Praktik pada Mottomo Inn.
9. Keluarga yang telah memberikan dukungan selama proses pengerjaan.
10. Teman-teman seperjuangan angkatan 2024.

Manado, Desember 2024



Penulis

DAFTAR ISI

FORMULIR DATA UMUM PERUSAHAAN	iii
FORMULIR PENILAIAN PELAKSANAAN KERJA PRAKTIK.....	vi
DAFTAR ISI	viii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR	xii
Daftar Lampiran	xiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan Kerja Praktik	3
1.4 Manfaat Kerja Praktik	3
1.5 Batasan Masalah.....	3
1.6 Sistematika Penulisan	3
BAB II DATA UMUM PERUSAHAAN	5
2.1 Sejarah Singkat Perusahaan	5
2.2 Lingkungan Pekerjaan Perusahaan	5
2.2.1 Visi	5
2.2.2 Misi	5
2.2.3 Struktur Perusahaan	6
2.3 Lingkup Pekerjaan Yang Dilakukan.....	6
BAB III LANDASAN TEORI.....	7
3.1 Teori Pendukung	7
3.1.1 Teori Manajemen Hotel.....	7
3.1.2 Pemrograman <i>Android</i>	7
3.2 Metodologi Pengembangan Perangkat Lunak	8
3.2.1 Metodologi <i>Waterfall</i>	8
3.2.2 Kakas Pemodelan yang Digunakan.....	9
3.3 Prosedur Pengumpulan dan Pengolahan Data	12
3.3.1 Pengumpulan Data	12

3.3.2	Pengolahan Data.....	13
BAB IV PEMBAHASAN.....		14
4.1	Analisis.....	14
4.1.1	Sumber Data.....	14
4.1.2	Daftar Pertanyaan Wawancara	14
4.1.3	Analisis Hasil Wawancara.....	15
4.1.4	Analisis Hasil Observasi	15
4.1.5	Pemodelan Sistem Sedang Berjalan.....	16
4.1.6	Identifikasi Masalah dan Kesempatan	20
4.1.7	Analisis Target Pengguna.....	20
4.1.8	Daftar Spesifikasi Persyaratan	21
4.2	Desain.....	21
4.2.1	Pemodelan.....	21
4.2.2	Pemodelan Antarmuka Aplikasi.....	28
4.2.3	Kamus Data.....	32
4.2.4	Pemodelan Modul Program.....	33
4.3	Implementasi	36
4.3.1	Lingkungan Implementasi.....	36
4.3.2	Manajemen Perubahan	36
4.3.3	Aturan Bisnis.....	37
4.3.3	Implementasi Antarmuka Aplikasi.....	38
4.3.4	Implementasi Basis Data.....	42
4.3.5	Implementasi Modul Program.....	43
4.4	Pengujian.....	46
4.4.1	Tujuan Pengujian.....	46
4.4.2	Kasus Pengujian	46
4.4.3	Pelaksanaan Pengujian	47
4.4.4	Analisis Hasil Pengujian	51
BAB V KESIMPULAN DAN SARAN.....		52
5.1	Kesimpulan	52
5.2	Saran.....	52
DAFTAR PUSTAKA		53

Lampiran A..... 54
Lampiran B..... 55
Lampiran C..... 56

DAFTAR TABEL

Tabel 3. 1 Daftar Simbol Class Diagram	10
Tabel 4. 1 Use Case Table Mengirim Laporan.....	17
Tabel 4. 2 Use Case Table Mengurus Check In Sistem Sedang Berjalan	17
Tabel 4. 3 Use Case Table Mengurus Check Out Sistem Sedang Berjalan	18
Tabel 4. 4 Use Case Table Memproses Booking Sistem Sedang Berjalan	18
Tabel 4. 5 Use Case Table Membatalkan Booking Sistem Sedang Berjalan	19
Tabel 4. 6 Use Case Table Memproses Booking Sistem Sedang Berjalan	19
Tabel 4. 7 Use Case Table Mengirim Booking Sistem Sedang Berjalan	20
Tabel 4. 8 Identifikasi Masalah Dan Kesempatan.....	20
Tabel 4. 9 Use Case Table Melakukan Pelaporan	22
Tabel 4. 10 Use Case Table Mengurus Proses Check In.....	23
Tabel 4. 11 Use Case Diagram Mengurus proses check out	23
Tabel 4. 12 Use Case Table Memproses Booking.....	23
Tabel 4. 13 Use Case Table Membatalkan Booking	24
Tabel 4. 14 Use Case Table Mengirim Booking	25
Tabel 4. 15 Kamus Data	32
Tabel 4. 16 Software Yang Dipakai.....	36
Tabel 4. 17 Hardware Yang Dipakai	36
Tabel 4. 18 Manajemen Perubahan	36
Tabel 4. 19 Pelaksanaan Pengujian Aplikasi.....	47

DAFTAR GAMBAR

Gambar 2. 1 Struktur Perusahaan.....	6
Gambar 3. 1 Wataerfall [7]	9
Gambar 3. 2 Use Case Diagram	10
Gambar 3. 3 Class Diagram [8]	10
Gambar 3. 4 Activity Diagram [9]	12
Gambar 4. 1 Activity Diagram Sistem Sedang Berjalan	16
Gambar 4. 2 Use Case Diagram Sistem Sedang Berjalan.....	17
Gambar 4. 3 Use Case Diagram	22
Gambar 4. 4 Class Diagram Sistem Baru	26
Gambar 4. 5 Activity Diagram Sistem Baru	27
Gambar 4. 6 Desain Halaman Masuk	28
Gambar 4. 7 Desain Halaman Daftar	28
Gambar 4. 8 Desain Halaman Aktivitas - Receptionist.....	29
Gambar 4. 9 Desain Halaman Riwayat - Receptionist.....	29
Gambar 4. 10 Desain Halaman Tambahkan Aktivitas	30
Gambar 4. 11 Desain Halaman Tentang Kamar.....	30
Gambar 4. 12 Desain Halaman Aktivitas - Manager	31
Gambar 4. 13 Desain Halaman Riwayat - Manager	31
Gambar 4. 14 Halaman Masuk.....	38
Gambar 4. 15 Halaman Daftar	39
Gambar 4. 16 Halaman Aktivitas - Receptionist.....	39
Gambar 4. 17 Halaman Riwayat	40
Gambar 4. 18 Halaman Tentang.....	40
Gambar 4. 19 Halaman Ubah/Tambah Data	41
Gambar 4. 20 Halaman Aktivitas - Manager	41
Gambar 4. 21 Database – History	42
Gambar 4. 22 Database - Kamar	42
Gambar 4. 23 Database - User	43

Daftar Lampiran

Lampiran 1 Hasil User Acceptance Test Manager	54
Lampiran 2 Hasil User Acceptance Test Receptionist	55
Lampiran 3 Wawancara receptionist	56
Lampiran 4 Wawancara Manager	57
Lampiran 5 Wawancara Housekeeping	57

BAB I

PENDAHULUAN

1.1 Latar Belakang

Mottomo Inn adalah sebuah budget hotel yang berlokasi di Kota Manado. Hotel ini berada di bawah naungan PT. Manado Bungalo Sentosa dan telah beroperasi sejak tahun 2019. Mottomo Inn memiliki 31 kamar yang dibedakan berdasarkan jenis, seperti *Cabin Room*, *Twin Room*, *Double Room*, *New Double Room* dan *Super Single Room*. Harga dan fasilitas dalam kamar bergantung pada jenis kamar tersebut. Oleh karena itu, diperlukan manajemen kamar yang baik untuk mengelola dan mengoordinasikan semua aspek terkait kamar, guna memastikan pengalaman tamu yang memuaskan serta kelancaran operasional hotel.

Manajemen kamar hotel yang efektif sangat penting bagi pihak manajemen karena secara langsung memengaruhi pelayanan, kepuasan tamu, dan keuntungan finansial hotel. Dengan manajemen yang baik, pengelola dapat memastikan kamar selalu siap ditempati, yang esensial untuk mempertahankan standar layanan tinggi dan mencegah keluhan tamu. Proses manajemen kamar meliputi pemesanan, pembersihan, perawatan, pengelolaan fasilitas, serta layanan tamu untuk memastikan kenyamanan dan kepuasan selama menginap.

Salah satu faktor paling penting dalam memaksimalkan pelayanan adalah proses check-in dan ketersediaan kamar. Hal ini menjadi salah satu poin krusial, di mana hotel dituntut memberikan pelayanan terbaik dan meminimalkan permasalahan, seperti kesalahan data kamar kosong. Permasalahan semacam ini sering terjadi bahkan di hotel besar sekalipun, yang pada akhirnya mengurangi tingkat kepuasan tamu dan dapat menimbulkan kerugian bagi pihak hotel.

Proses pemesanan kamar dapat dilakukan melalui dua cara, yaitu datang langsung (*walk-in*) atau pemesanan daring melalui mitra seperti Traveloka dan Tiket.com. Dengan metode *walk-in*, tamu dapat melakukan *check-in* langsung di Mottomo Inn dengan mengisi formulir yang mencakup data seperti nama lengkap, nomor telepon, jenis kamar, lama menginap, serta tanda tangan. Resepsionis akan mencatat informasi tersebut. Setelah pembayaran selesai, tamu akan menerima kunci kamar dan dapat langsung masuk ke kamar.

Untuk pemesanan melalui mitra, pengunjung dapat memesan kamar melalui aplikasi mitra, dan notifikasi pemesanan akan ditampilkan di aplikasi tersebut. Resepsionis mencatat detail pemesanan kamar *online* dalam sebuah buku. Karena pembayaran telah dilakukan melalui aplikasi, pelanggan hanya perlu menunjukkan bukti pemesanan kepada resepsionis untuk menerima kunci kamar. Selain itu, resepsionis juga merekap detail transaksi harian dan bulanan menggunakan *Microsoft Excel*.

Namun, beberapa masalah sering terjadi dalam proses bisnis tersebut. Data yang ditulis di atas kertas seringkali berserakan atau hilang, sehingga menyulitkan proses pencatatan dan pembuatan laporan. Selain itu, kesalahan pencatatan dan penyalinan data sering terjadi dalam proses ini. Pemindahan data dari kertas ke *Microsoft Excel* juga memakan waktu yang cukup lama. Oleh karena itu, pihak manajemen ingin mendigitalisasi proses bisnis dan data mereka guna meningkatkan kualitas layanan serta kepuasan tamu.

Berdasarkan permasalahan tersebut, ada peluang untuk mengembangkan Aplikasi *Online* Pencatatan Transaksi Kamar Hotel Berbasis *Android* khusus untuk Mottomo Inn. Aplikasi ini akan mengelola seluruh aspek pencatatan data transaksi hotel, mulai dari reservasi hingga check-in dan check-out. Aplikasi akan memiliki dua jenis pengguna, yaitu *Receptionist* dan *Manager*, masing-masing dengan fungsinya sendiri. Penggunaan aplikasi ini diharapkan dapat mengurangi kesalahan pencatatan dan penyalinan data dengan menyediakan sistem yang terkomputerisasi, akurat, dan konsisten. Selain itu, aplikasi ini akan memudahkan pembaruan informasi, meningkatkan keamanan data, serta menghemat waktu dalam pengolahan dan pembuatan laporan.

Pemilihan aplikasi berbasis *Android* didasarkan pada fakta bahwa resepsionis menggunakan perangkat *Android* untuk operasional hotel, termasuk aplikasi mitra dan *WhatsApp*. Dengan demikian, aplikasi ini diharapkan dapat mendukung operasional hotel secara lebih efisien dan profesional.

1.2 Rumusan Masalah

Bagaimana membangun Aplikasi *Online* Pencatatan Transaksi Kamar Hotel Berbasis *Android* untuk Mottomo Inn yang dapat memroses data pemesanan, *check-in*, dan proses *check-out* yang dilakukan oleh tamu secara *real-time*.

1.3 Tujuan Kerja Praktik

Membangun Aplikasi *Online* Pencatatan Transaksi Kamar Hotel Berbasis *Android* untuk Mottomo Inn yang dapat membantu memproses data pemesanan, *check-in*, dan proses *check-out* secara *real-time*.

Commented [LW1]: Perbaiki sesuai revisi rumusan masalah

1.4 Manfaat Kerja Praktik

a. Pihak Mottomo Inn:

1. Data pemesanan dan informasi kamar yang lebih akurat mengurangi risiko kesalahan dalam pencatatan, reservasi, dan alokasi kamar, sehingga pengalaman menginap lebih baik.
2. Proses *check-in* dan *check-out* yang lebih cepat dan efisien, mengurangi waktu tunggu serta menyederhanakan administrasi. Dapat meningkatkan efisiensi waktu dengan digitalisasi proses pencatatan, dan pengolahan data kamar.

b. Penulis:

1. Mendapatkan pengalaman kerja nyata.
2. Dapat penerapan pengetahuan yang telah diperoleh selama perkuliahan ke dalam situasi nyata di dunia bisnis.

1.5 Batasan Masalah

1. Aplikasi yang dibuat hanya dapat dijalankan pada perangkat berbasis *Android* dengan minimal sistem operasi *Android 8*.
2. Aplikasi yang dibangun tidak mencakup proses pembayaran sama sekali.
3. Aplikasi yang dibangun tidak dapat mengunggah gambar dikarenakan penggunaan basis data *firebase* yang harus membayar untuk mengakases fitur mengunggah gambar.

1.6 Sistematika Penulisan

Berikut adalah sistematika yang dipakai didalam penelitian kali ini:

BAB I: PENDAHULUAN

Pada bab ini berisi tentang permasalahan yang ditemukan selama melakukan kerja praktik. Bagian ini berisi latar belakang, rumusan masalah, tujuan Kerja Praktik, manfaat Kerja Praktik, batasan masalah, dan sistematika penulisan.

BAB II: DATA UMUM PERUSAHAAN

Pada bab ini membahas tentang data umum perusahaan yang terdiri dari sejarah perusahaan, lingkup pekerjaan perusahaan, dan lingkup pekerjaan yang dilakukan.

BAB III: LANDASAN TEORI

Pada bab ini membahas terkait topik KP, teknologi yang digunakan, metodologi pengembangan perangkat lunak, kaskas pemodelan, dan prosedur pengumpulan dan pengolahan data.

BAB IV: PEMBAHASAN

Bab ini akan mengkaji permasalahan yang dihadapi dan solusinya dengan mengikuti langkah-langkah metodologi pengembangan perangkat lunak.

BAB V: KESIMPULAN DAN SARAN

Pada bab ini berisi tentang kesimpulan dan saran untuk pengembangan selanjutnya.

BAB II

DATA UMUM PERUSAHAAN

2.1 Sejarah Singkat Perusahaan

Mottomo Inn merupakan *budget hotel* yang telah beroperasi sejak tahun 2019 di bawah naungan PT. Manado Bungalo Sentosa. Mottomo Inn didirikan di lokasi yang dianggap strategis, yaitu di jalan Sam Ratulangi, yang merupakan kawasan perkotaan yang dekat dengan berbagai lokasi penting seperti kantor perpajakan, kantor perwalikan Bank Indonesia, dan rumah sakit yang sangat memungkinkan untuk kedatangan tamu dari luar daerah.

2.2 Lingkungan Pekerjaan Perusahaan

Pada bagian ini akan membahas visi, misi, dan struktur dari Mottomo Inn:

2.2.1 Visi

Untuk memajukan pariwisata Kota Manado dan membuka lapangan pekerjaan.

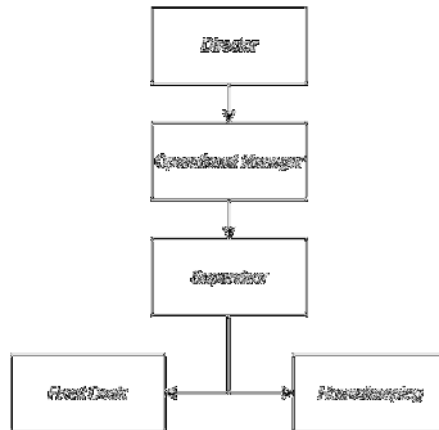
2.2.2 Misi

Berikut adalah misi dari PT. Manado Bungalo Sentosa:

1. Menjadi *budget hotel* pilihan wisatawan lokal dan domestic
2. Memperkerjakan dan melengkapi pekerja asli Manado dan sekitarnya
3. Berkontribusi dalam pembangunan Kota Manado

Commented [LW2]: Perbaiki formatting.
Jgn kasih jarak terlalu besar antara kalimat terakhir dan sub-bab berikutnya.

2.2.3 Struktur Perusahaan



Gambar 2. 1 Struktur Perusahaan

Untuk posisi yang ditempatkan pada saat magang adalah *front desk* yang bertugas untuk melayani tamu yang datang.

2.3 Lingkup Pekerjaan Yang Dilakukan

Selama melakukan Kerja Praktik di Mottomo Inn ditempatkan pada bagian *front desk* yang bertugas sebagai resepsionis hotel. Adapun aktivitas yang dilakukan selama Kerja Praktik:

1. Menyambut tamu yang datang
2. Melakukan proses *check-in* dan *check-out*
3. Melayani keluhan tamu
4. Melakukan pencatatan transaksi

BAB III

LANDASAN TEORI

3.1 Teori Pendukung

Pada bagian ini berisi penjelasan teori-teori yang digunakan sebagai dasar dalam pembangunan aplikasi *online* Pencatatan Transaksi Kamar Hotel Berbasis *Android*.

3.1.1 Teori Manajemen Hotel

Manajemen kamar hotel merupakan salah satu komponen yang sangat penting bagi pihak hotel karena secara langsung mempengaruhi efisiensi operasional, kepuasan tamu, dan profitabilitas hotel. Manajemen ruangan yang baik dapat membantu untuk menjaga ruangan dalam kondisi siap pakai setiap saat. Hal ini penting untuk menjaga standar pelayanan yang tinggi dan mencegah keluhan tamu dan mempertahankan reputasi hotel [1].

Komponen paling awal yang harus diperhatikan ketika ada pengunjung yang datang adalah pelayanan pada saat prosedur *check-in* yang kadang memakan waktu yang lama dan harus mengalami berbagai macam pencatatan dokumen yang bisa disederhanakan dengan formulir yang singkat dan efektif [1].

3.1.2 Pemrograman *Android*

Sistem operasi *Android* merupakan sebuah sistem operasi berbasis *Linux* yang dirancang khusus untuk perangkat seperti *smartphone* dan *tablet*. Arsitektur *Android* terdiri dari kernal linux dibagian bawah, *libraries* dibagian tengah, dan aplikasi dibagian atas. Setiap bagian saling terhubung dan bekerja sama untuk menjalankan fungsionalitas seperti *Storage Management*, *Processing*, dan pengoperasian perangkat keras [2].

A. Kotlin

Kotlin merupakan bahasa pemrograman moderen yang dirancang oleh perusahaan bernama *JetBrains* untuk pemrograman *Android*, dan didukung oleh pihak *google* secara resmi pada tahun 2017. Kotlin dirancang untuk berjalan di platform *Java Virtual Machine* (JVM), yang berarti dapat beroperasi bersamaan dengan bahasa Java dan menggunakan ekosistem pustaka yang sama [3].

B. *Firebase*

Firebase adalah layanan basis data yang disediakan oleh Google secara gratis dan menawarkan berbagai layanan *backend*, seperti *real-time database*, *Cloud Storage*, sistem keamanan, dan autentikasi. Keamanan *Firebase* tergolong baik untuk pengembangan aplikasi berskala kecil. Selain itu, *Firebase* menyediakan berbagai layanan keamanan tambahan yang dapat dimanfaatkan oleh pengembang [4].

Pemrograman menggunakan *Firebase* dapat dengan mudah dilakukan menggunakan bahasa pemrograman Kotlin dikarenakan sama-sama buatan google dan dapat diintegrasikan dengan mudah menggunakan *Android Studio*. Inilah yang membuat *Firebase* paling sering dipakai dalam pengembangan aplikasi *Android* [5].

3.2 Metodologi Pengembangan Perangkat Lunak

Pada bagian ini berisi penjelasan Metodologi pengembangan perangkat lunak yang dipakai dalam pembangunan aplikasi *online* Manajemen Kamar Hotel berbasis *Android*.

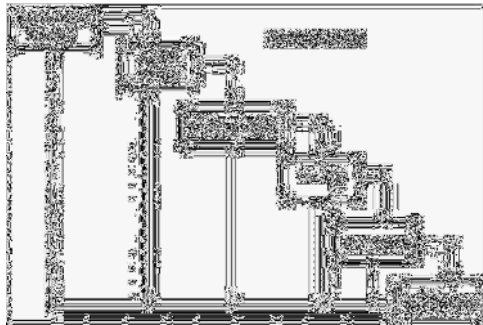
3.2.1 Metodologi *Waterfall*

Metodologi *waterfall* merupakan metodologi pengembangan perangkat lunak yang paling umum digunakan dikarenakan alur pengerjaannya yang linear dan berurutan yang mudah untuk dipahami. Metodologi ini dapat digunakan baik untuk pengerjaan proyek secara berkelompok maupun proyek tunggal yang hanya melibatkan 1 orang pengembang saja [6].

Waterfall terbagi dalam beberapa tahapan pengerjaan, yaitu tahap analisis, yang berisi analisis terhadap data yang telah dikumpulkan untuk mencari tahu apa yang harus dilakukan; tahapan desain, yang berisi desain dari aplikasi yang dirancang beserta desain antarmuka aplikasi; tahapan implementasi, yang berisi eksekusi dari hasil analisis dan desain; serta tahapan pengujian, untuk menguji apakah aplikasi yang dibangun telah berjalan dengan baik atau tidak. Setelah keempat tahapan berjalan dengan baik, barulah masuk pada tahap akhir, di mana perangkat lunak yang dibangun dapat digunakan. Tahap selanjutnya adalah *maintenance* atau perawatan yang dilakukan oleh pihak pengembang [6].

Kelemahan dalam metodologi ini adalah tidak fleksibel seperti metodologi *agile* yang lebih mudah untuk menerima perubahan seperti perubahan ruang

lingkup mendadak dan proses tersebut dapat mengganggu jalannya perkembangan proyek, namun masih bisa diatasi jika perubahan yang terjadi hanyalah perubahan kecil saja [7].



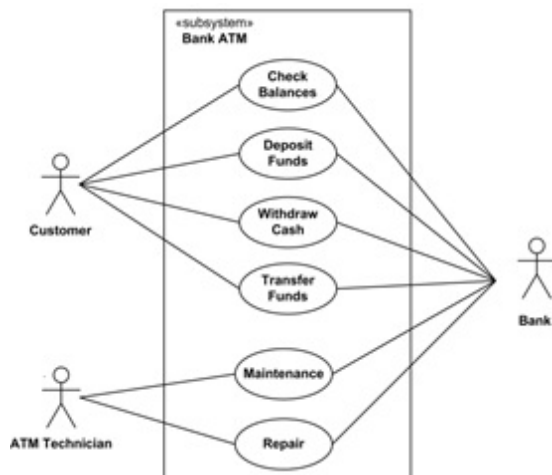
Gambar 3. 1 Waterfall [7]

3.2.2 Kakas Pemodelan yang Digunakan

Pada bagian ini akan berisi teori tentang kakas pemodelan yang dipakai yaitu, *Unified Modeling Language* atau UML. Jenis-jenis yang akan dipakai dalam memodelkan adalah *Use Case*, *Class Diagram*, dan *Activity Diagram*.

A. *Use Case Diagram*

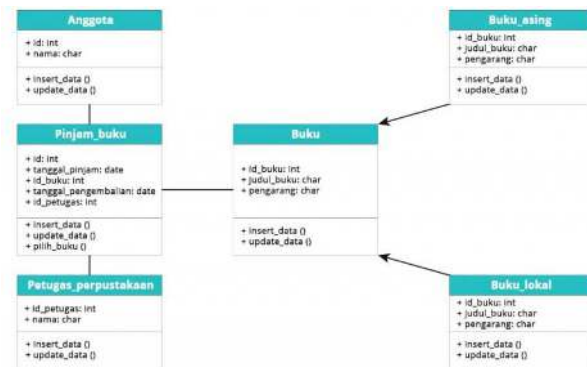
Use Case merupakan diagram yang digunakan untuk memodelkan fungsional sistem yang dilakukan oleh setiap *actor* dengan *case*. Diagram ini memiliki 2 bagian yaitu *Use Case Diagram* yang menggambarkan interaksi *actor* dan *case*, dan *Use Case Table* untuk menjelaskan [8].



Gambar 3. 2 Use Case Diagram

B. Use Case Diagram


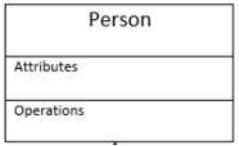


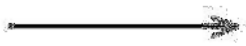
Class Diagram digunakan untuk memodelkan sebuah struktur sistem perangkat lunak dengan menggambarkan kelas yang berorientasi objek bersama dengan atribut dan hubungan antar kelas [9].



Gambar 3. 3 Class Diagram [8]

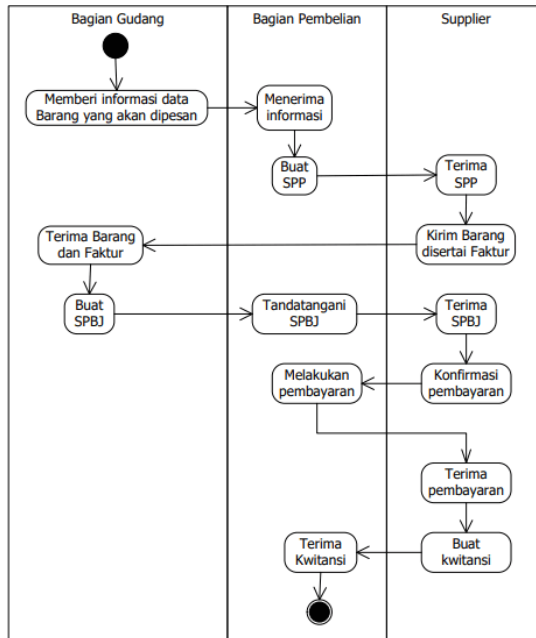
Tabel 3. 1 Daftar Simbol Class Diagram

No	Simbol	Nama	Keterangan
----	--------	------	------------

1		Generalisasi	Hubungan antara objek <i>descendent</i> dan <i>ancestor</i>
2		Kelas	Himpunan objek-objek beserta atriburnya serta operasi yang sama
3		Ketergantungan	Relasi antar kelas dengan makna ketergantungan antar kelas
4		Agregasi	Relasi antara kelas dengan makna semua bagian
5		Asosiasi berarah	Hubungan antar 2 atau lebih kelas dimana salah satunya digunakan oleh kelas lainnya.

C. *Activity Diagram*

Activity Diagram adalah salah satu jenis diagram UML yang terstruktur dan dapat memodelkan aliran aktivitas dalam sebuah sistem. Biasanya dipakai untuk memodelkan alur kerja bisnis dan logika kerjanya dari titik awal sampai titik akhir [9].



Gambar 3.4 Activity Diagram [9]

3.3 Prosedur Pengumpulan dan Pengolahan Data

Pada bagian ini akan membahas tentang cara pengumpulan dan pengolahan data yang dilakukan pada proyek.

3.3.1 Pengumpulan Data

Dalam proses mengumpulkan data menggunakan 2 teknik pengumpulan data, yaitu observasi dan wawancara.

A. Observasi

Melihat secara langsung proses bisnis yang terjadi pada Mottomo Inn yang mencakup *booking*, *check-in*, *check-out*, dan *cleaning*. Observasi berlangsung selama 250 jam magang terhitung mulai dari 7 Juni sampai 17 Juli 2024.

B. Wawancara

Mewawancarai langsung *manager*, *receptionist*, dan *housekeeping* untuk mencari tau masalah dan kebutuhan dari pihak Mottomo Inn.

3.3.2 Pengolahan Data

Hasil dari observasi dan wawancara akan dikumpulkan terlebih dahulu untuk dianalisis dan mencari apa saja yang dibutuhkan oleh pihak Mottomo Inn. Hasil wawancara digunakan untuk mencari tau apa saja proses yang terjadi didalam bisnis dan apa yang diperlukan untuk meningkatkan efisiensi, sedangkan hasil observasi akan digunakan untuk melihat apa saja masalah yang dihadapi selama proses bisnis sehari-hari.

BAB IV PEMBAHASAN

4.1 Analisis

Pada fase ini akan membahas data yang telah dikumpulkan dan diolah. Fase ini bertujuan untuk mencari tahu apa yang menjadi kekurangan dari sistem yang sedang berjalan, apa yang dibutuhkan oleh pihak perusahaan, dan aplikasi seperti apa yang diperlukan oleh perusahaan.

4.1.1 Sumber Data

Data yang didapatkan berasal dari wawancara dengan *manager, receptionist*, dan *housekeeping* untuk mencari tahu proses bisnis dan kendala apa saja dari tiga perspektif yang berbeda. Pertanyaan yang diajukan seputar proses bisnis yang terjadi mulai dari SOP, fakta lapangan, dan apa saja permasalahan yang sering terjadi.

Selain wawancara, observasi juga dilakukan untuk mengamati setiap proses bisnis yang dilakukan oleh perusahaan. Observasi dilakukan selama masa magang 250 jam. Poin yang diamati selama observasi meliputi bagaimana proses bisnis, permasalahan yang muncul, dan peluang apa yang bisa diselesaikan dengan pembuatan aplikasi.

4.1.2 Daftar Pertanyaan Wawancara

Berikut adalah pertanyaan yang ditanyakan kepada *maneger, receptionist*, dan *housekeeping*.

4.1.2.1 Daftar Pertanyaan Untuk Receptionist

1. Bagaimana prosedur pengunjung memesan kamar?
2. Apakah ada kendala saat pengunjung datang dan memesan kamar?
3. Apakah ada masalah ketika proses *check-out* dilakukan?
4. Bagaimana pencatatan pengunjung dilakukan?
5. Apakah ada kendala pada proses pencatatan?

4.1.2.2 Daftar Pertanyaan Untuk Manajer

1. Bagaimana cara melakukan perekapan laporan kepada atasan?
2. Apakah ada kendala dalam proses perekapan?

3. Apa peran pemilik perusahaan untuk dalam proses bisnis sehari-hari?
4. Apakah ada masalah dalam proses pemantauan sehari-hari?

4.1.2.3 Daftar Pertanyaan Untuk *Housekeeping*

1. Bagaimana proses pembersihan ruangan kamar?
2. Bagaimana proses pembersihan saat sebelum pengunjung melakukan *check-out*?
3. Apakah ada kendala dalam prosesnya?

4.1.3 Analisis Hasil Wawancara

Berdasarkan hasil wawancara yang dilakukan, dapat disimpulkan bahwa proses bisnis yang dilakukan sehari-hari sebagai berikut:

1. Pemilik perusahaan tidak ikut campur dalam proses bisnis sehari-hari dikarenakan kesibukan beliau dan hanya menerima laporan sesuai dengan format yang telah ditentukan sebelumnya.
2. *Manager* melakukan yang melakukan perekapan harian, mingguan, maupun bulanan terhadap setiap transaksi yang terjadi di Mottomo Inn.
3. Proses pencatatan dan perekapan oleh *manager* dilakukan menggunakan *Microsoft excel* sesuai prosedur yang berlaku di perusahaan dan tidak bisa diubah.
4. Permasalahan yang dihadapi manajer adalah pemantauan sehari-hari karena *manager* sendiri menangani beberapa usaha lainnya dan tidak selalu ada di lokasi.
5. Proses pencatatan data pengunjung oleh *receptionist* dilakukan menggunakan formulir kertas yang kemudian akan disalin kedalam *Microsoft excel*. Tahapan ini dapat dipermudah menggunakan aplikasi yang akan dibangun nantinya.

4.1.4 Analisis Hasil Observasi

Dalam observasi yang dilakukan, terdapat beberapa hal yang menjadi poin penting dalam pembuatan aplikasi.

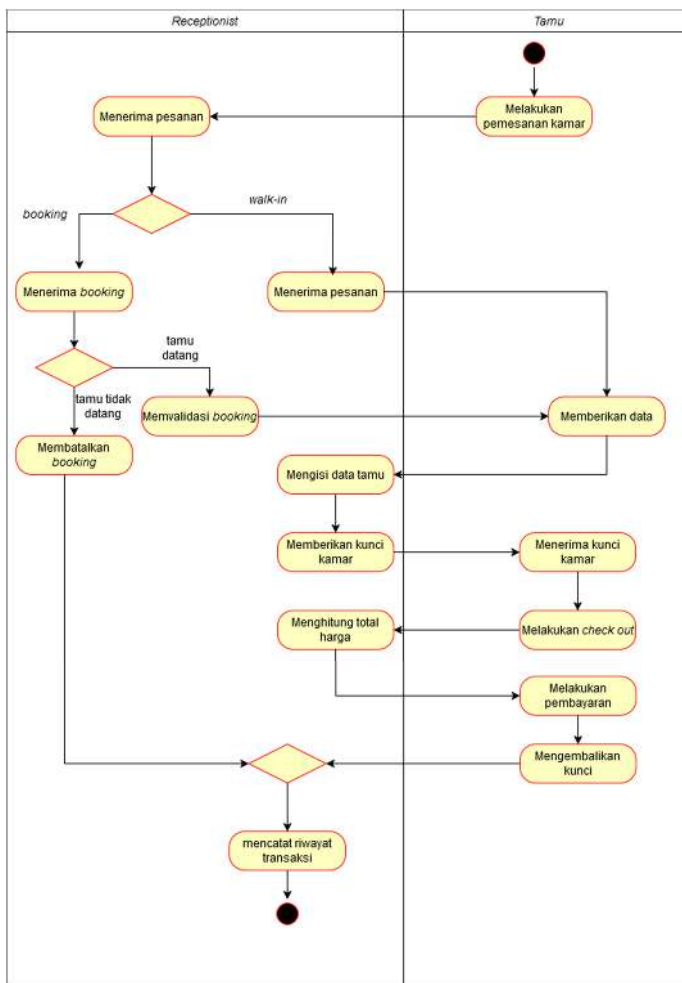
1. *Owner* tidak ikut campur dalam proses bisnis sehari-hari dan memercayakan semuanya kepada *manager*.
2. *Housekeeping* tidak memerlukan tipe pengguna sendiri dikarenakan *housekeeping* adalah *outsourse* yang bisa digantikan kapan saja.

3. *Manager* tidak memerlukan fitur perekapan data dan hanya memerlukan aplikasi untuk memantau status kamar secara *realtime*.
4. *Receptionist* hanya memerlukan fitur pencatatan data pengunjung saja.

4.1.5 Pemodelan Sistem Sedang Berjalan

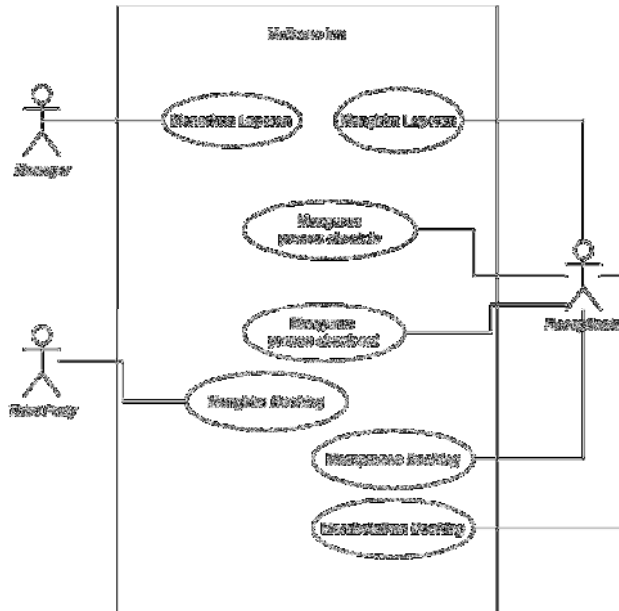
Berikut adalah penggambaran dari sistem sedang berjalan:

A. *Activity Diagram*



Gambar 4.1 *Activity Diagram* Sistem Sedang Berjalan

B. Use Case Diagram



Gambar 4.2 Use Case Diagram Sistem Sedang Berjalan

C. Use Case Table

Tabel 4.1 Use Case Table Mengirim Laporan

Use Case Name	Mengirim Laporan
Actor	Receptionist
Description	Pengguna (Receptionist) mengirimkan laporan transaksi yang tercatat.
Normal Course: (Actor's Action)	Pengguna (Receptionist) mengirim dokumen berisi laporan kepada Manager.
Alternate Course: (Actor's Action)	Pengguna (Receptionist) tidak mengirimkan laporan transaksi.
Pre-Condition	Pengguna (Receptionist) telah mencatat semua transaksi.
Post-Condition	Laporan diterima Manager.
Assumption	

Tabel 4.2 Use Case Table Mengurus Check In Sistem Sedang Berjalan

Use Case Name	Mengurus Proses Check In
Actor	Receptionist

Description	Pengguna (<i>Receptionist</i>) mengisi data yang diperlukan untuk melakukan <i>check in</i>
Normal Course: (Actor's Action)	<ol style="list-style-type: none"> 1. Pengguna (<i>Receptionist</i>) meminta data tamu yang diperlukan. 2. Pengguna (<i>Receptionist</i>) memasukan data tamu kedalam formulir.
Alternate Course: (Actor's Action)	Pengguna (<i>Receptionist</i>) tidak memasukan data tamu kedalam formulir.
Pre-Condition	Tamu memberikan data yang diperlukan.
Post-Condition	Formulir data tamu terisi.
Assumption	

Tabel 4. 3 Use Case Table Mengurus Check Out Sistem Sedang Berjalan

Use Case Name	Mengurus Proses <i>Check Out</i>
Actor	<i>Receptionist</i>
Description	Pengguna (<i>Receptionist</i>) mengurus prosedur <i>check out</i> sesuai dengan SOP yang berlaku.
Normal Course: (Actor's Action)	<ol style="list-style-type: none"> 1. Pengguna (<i>Receptionist</i>) mencatat <i>check out</i>. 2. Pengguna (<i>Receptionist</i>) meminta kunci kamar. 3. Pengguna (<i>Receptionist</i>) meminta bayaran dari tamu.
Alternate Course: (Actor's Action)	Pengguna (<i>Receptionist</i>) tidak melakukan prosedur <i>check out</i> .
Pre-Condition	Pengguna (<i>Receptionist</i>) harus meminta <i>housekeeping</i> untuk memeriksa kondisi kamar.
Post-Condition	Pengguna (<i>Receptionist</i>) mencatat transaksi sebagai transaksi yang telah selesai.
Assumption	

Tabel 4. 4 Use Case Table Memproses Booking Sistem Sedang Berjalan

Use Case Name	Menproses <i>Booking</i>
Actor	<i>Receptionist</i>
Description	Pengguna (<i>Receptionist</i>) menerima <i>booking</i> lewat pihak ketiga
Normal Course: (Actor's Action)	<ol style="list-style-type: none"> 1. Pengguna (<i>Receptionist</i>) mencatat informasi seputar <i>booking</i>. 2. Pengguna (<i>Receptionist</i>) menunggu konfirmasi tamu.

	<ol style="list-style-type: none"> 3. Sistem mengirim notifikasi <i>booking</i>. 4. Sistem memberikan informasi <i>booking</i>.
Alternate Course: (Actor's Action)	<ol style="list-style-type: none"> 1. Pengguna (<i>Receptionist</i>) tidak menerima notifikasi <i>booking</i> dari sistem. 2. Sistem tidak memberikan notifikasi <i>booking</i>.
Pre-Condition	Pengguna (<i>Receptionist</i>) mengaktifkan aplikasi pihak ketiga.
Post-Condition	Data <i>booking</i> berhasil ditambahkan dalam aplikasi.
Assumption	

Tabel 4. 5 Use Case Table Membatalkan *Booking* Sistem Sedang Berjalan

Use Case Name	Menatalkan <i>Booking</i>
Actor	<i>Receptionist</i>
Description	Pengguna (<i>Receptionist</i>) membatalkan <i>booking</i> yang telah dilakukan sebelumnya
Normal Course: (Actor's Action)	<ol style="list-style-type: none"> 1. Pengguna (<i>Receptionist</i>) menerima permintaan pembatalan <i>booking</i>. 2. Pengguna (<i>Receptionist</i>) membatalkan <i>booking</i>. 3. Sistem mengirim permintaan pembatalan <i>booking</i>.
Alternate Course: (Actor's Action)	Sistem tidak mengirim permintaan pembatalan <i>booking</i> .
Pre-Condition	Pengguna (<i>Receptionist</i>) menerima pembatalan <i>booking</i> dari tamu lewat aplikasi pihak ketiga.
Post-Condition	<i>Booking</i> berhasil dibatalkan.
Assumption	

Tabel 4. 6 Use Case Table Memproses *Booking* Sistem Sedang Berjalan

Use Case Name	Menerima Laporan
Actor	<i>Manager</i>
Description	Pengguna (<i>Manager</i>) menerima laporan yang diberikan <i>Receptionist</i> .
Normal Course: (Actor's Action)	Pengguna (<i>Manager</i>) menerima dokumen yang berisi laporan transaksi.
Alternate Course: (Actor's Action)	Pengguna (<i>Manager</i>) tidak menerima laporan dari <i>Receptionist</i> .

Pre-Condition	Laporan dikirim oleh <i>Receptionist</i> .
Post-Condition	Laporan diperiksa.
Assumption	

Tabel 4. 7 Use Case Table Mengirim Booking Sistem Sedang Berjalan

Use Case Name	Mengirim <i>Booking</i>
Actor	<i>Third Party</i>
Description	Pengguna (<i>Third Party</i>) mengirim <i>booking</i> yang dilakukan kepada pihak hotel.
Normal Course: (Actor's Action)	<ol style="list-style-type: none"> 1. <i>Third Party</i> menerima permintaan <i>booking</i> dari pengguna aplikasi <i>third party</i>. 2. <i>Third Party</i> mengirimkan permintaan pada pihak hotel.
Alternate Course: (Actor's Action)	<ol style="list-style-type: none"> 1. <i>Third Party</i> tidak menerima permintaan <i>booking</i>. 2. <i>Third Party</i> tidak mengirimkan permintaan pada pihak hotel.
Pre-Condition	<i>Third Party</i> melakukan kemitraan dengan hotel.
Post-Condition	Pihak hotel menerima permintaan <i>booking</i> .
Assumption	

4.1.6 Identifikasi Masalah dan Kesempatan

Tabel 4. 8 Identifikasi Masalah Dan Kesempatan

No	Masalah	Prioritas	Potensi Solusi/Kesempatan
1	Sering tercecemnya formulir data tamu	Tinggi	Membuat fitur aplikasi untuk mencatat setiap transaksi dan data pengunjung.
2	<i>Manager</i> tidak bisa memantau secara <i>realtime</i>	Tinggi	Membuat fitur aplikasi yang dapat menampilkan data secara <i>realtime</i> .

4.1.7 Analisis Target Pengguna

Untuk pengguna yang ditargetkan tentunya adalah karyawan dari Mottomo Inn, terutama *manager* dan *receptionist* dengan harapan dapat meningkatkan efektifitas kerja dan dapat menyelesaikan masalah yang telah disebutkan diatas.

Dengan cara kerja dari Mottomo Inn tidak diperlukan tipe pengguna lainnya seperti *housekeeping* dan *owner* karena dinilai tidak efektif dimana *housekeeping* adalah karyawan *outsourcing* yang bisa keluar kapan saja dan *owner* tidak berinteraksi langsung dengan proses bisnis sehari-hari.

4.1.8 Daftar Spesifikasi Persyaratan

Berikut adalah daftar spesifikasi persyaratan yang diperlukan baik dari pengguna maupun aplikasi yang dibuat.

- a. Daftar Spesifikasi Pengguna
 1. Pengguna memiliki akun *email*.
 2. Pengguna dapat melakukan *input* data.
 3. Pengguna dapat melakukan *delete* data.
- b. Daftar Spesifikasi Perangkat Lunak
 1. Perangkat Lunak dapat dengan mudah dipahami cara kerjanya oleh pengguna
 2. Perangkat Lunak dapat menyimpan data pengunjung
 3. Perangkat Lunak dapat menampilkan data *history* pemesanan kamar

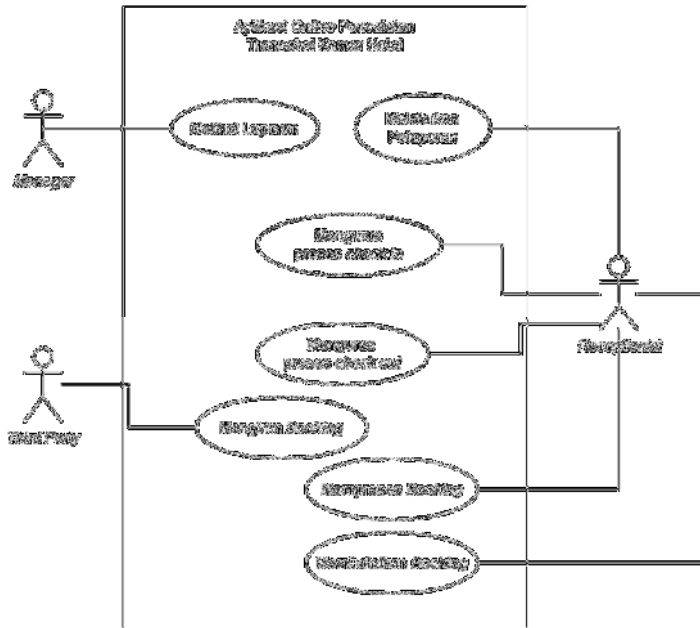
4.2 Desain

Pada fase ini akan berisi pembahasan seputar pemodelan dari aplikasi yang dibangun beserta *mockup* dari aplikasi yang dibuat.

4.2.1 Pemodelan

Pada bagian ini berisi pemodelan menggunakan *Unified Modelling Language* (UML) yaitu *Use Case Diagram*, *Class Diagram*, *Activity Diagram*.

A. Use Case Diagram



Gambar 4. 3 Use Case Diagram

B. Use Case Table

Tabel 4. 9 Use Case Table Melakukan Pelaporan

Use Case Name	Melakukan Pelaporan
Actor	Receptionist
Description	Pengguna (Receptionist) memasukkan data transaksi yang telah dilakukan
Normal Course: (Actor's Action)	Pengguna (Receptionist) memasukkan data sesuai format yang ada
Alternate Course: (Actor's Action)	Pengguna (Receptionist) tidak memasukkan data transaksi yang telah dilakukan
Pre-Condition	Pengguna (Receptionist) telah melakukan login sebelumnya
Post-Condition	Data tersimpan dalam database
Assumption	

Tabel 4. 10 Use Case Table Mengurus Proses Check In

Use Case Name	Mengurus Proses <i>Check In</i>
Actor	<i>Receptionist</i>
Description	Pengguna (<i>Receptionist</i>) mengisi data yang diperlukan untuk melakukan <i>check in</i>
Normal Course: (Actor's Action)	<ol style="list-style-type: none"> 1. Pengguna (<i>Receptionist</i>) meminta data tamu yang diperlukan. 2. Pengguna (<i>Receptionist</i>) memasukan data tamu kedalam aplikasi.
Alternate Course: (Actor's Action)	Pengguna (<i>Receptionist</i>) tidak memasukan data tamu kedalam aplikasi
Pre-Condition	Tamu memberikan data yang diperlukan
Post-Condition	Data tamu tersimpan
Assumption	

Tabel 4. 11 Use Case Diagram Mengurus proses check out

Use Case Name	Mengurus Proses <i>Check Out</i>
Actor	<i>Receptionist</i>
Description	Pengguna (<i>Receptionist</i>) mengurus prosedur <i>check out</i> sesuai dengan SOP yang berlaku dan menandai aktivitas kamar yang sedang berjalan sebagai "selesai"
Normal Course: (Actor's Action)	<ol style="list-style-type: none"> 1. Pengguna (<i>Receptionist</i>) mencatat <i>check out</i>. 2. Pengguna (<i>Receptionist</i>) meminta kunci kamar. 3. Pengguna (<i>Receptionist</i>) meminta bayaran dari tamu.
Alternate Course: (Actor's Action)	Pengguna (<i>Receptionist</i>) tidak melakukan prosedur <i>check out</i>
Pre-Condition	Pengguna (<i>Receptionist</i>) harus meminta <i>housekeeping</i> untuk memeriksa kondisi kamar
Post-Condition	Aktivitas kamar telah selesai dan dipindahkan ke bagian "Riwayat"
Assumption	

Tabel 4. 12 Use Case Table Memproses Booking

Use Case Name	Menproses <i>Booking</i>
Actor	<i>Receptionist</i>

Description	Pengguna (<i>Receptionist</i>) menerima <i>booking</i> lewat pihak ketiga
Normal Course: (Actor's Action)	<ol style="list-style-type: none"> 5. Pengguna (<i>Receptionist</i>) mencatat informasi seputar <i>booking</i>. 6. Pengguna (<i>Receptionist</i>) menunggu konfirmasi tamu. 7. Sistem mengirim notifikasi <i>booking</i>. 8. Sistem memberikan informasi <i>booking</i>.
Alternate Course: (Actor's Action)	<ol style="list-style-type: none"> 3. Pengguna (<i>Receptionist</i>) tidak menerima notifikasi <i>booking</i> dari sistem. 4. Sistem tidak memberikan notifikasi <i>booking</i>.
Pre-Condition	Pengguna (<i>Receptionist</i>) mengaktifkan aplikasi pihak ketiga.
Post-Condition	Data <i>booking</i> berhasil ditambahkan dalam aplikasi.
Assumption	

Tabel 4. 13 Use Case Table Membatalkan Booking

Use Case Name	Menatalkan <i>Booking</i>
Actor	<i>Receptionist</i>
Description	Pengguna (<i>Receptionist</i>) membatalkan <i>booking</i> yang telah dilakukan sebelumnya
Normal Course: (Actor's Action)	<ol style="list-style-type: none"> 1. Pengguna (<i>Receptionist</i>) menerima permintaan pembatalan <i>booking</i>. 2. Pengguna (<i>Receptionist</i>) membatalkan <i>booking</i>. 3. Sistem mengirim permintaan pembatalan <i>booking</i>.
Alternate Course: (Actor's Action)	Sistem tidak mengirim permintaan pembatalan <i>booking</i> .
Pre-Condition	Pengguna (<i>Receptionist</i>) menerima pembatalan <i>booking</i> dari tamu lewat aplikasi pihak ketiga.
Post-Condition	<i>Booking</i> berhasil dibatalkan.
Assumption	

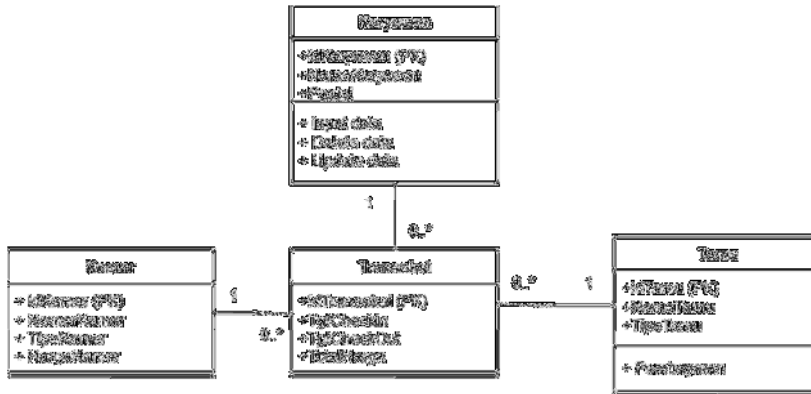
Use Case Name	Melihat Laporan
Actor	<i>Manager</i>
Description	Pengguna (<i>Manager</i>) melihat laporan aktivitas transaksi kamar yang terjadi

Normal Course: (Actor's Action)	<ol style="list-style-type: none"> 1. Pengguna (<i>Manager</i>) melihat aktivitas yang sedang berlangsung dan yang sudah selesai. 2. Sistem menampilkan aktivitas yang sedang berlangsung dan yang sudah selesai
Alternate Course: (Actor's Action)	<ol style="list-style-type: none"> 1. Pengguna (<i>Manager</i>) tidak melihat laporan. 2. Sistem tidak menampilkan laporan transaksi.
Pre-Condition	Pengguna (<i>Manager</i>) melakukan <i>login</i> .
Post-Condition	Laporan berhasil ditampilkan.
Assumption	

Tabel 4. 14 Use Case Table Mengirim Booking

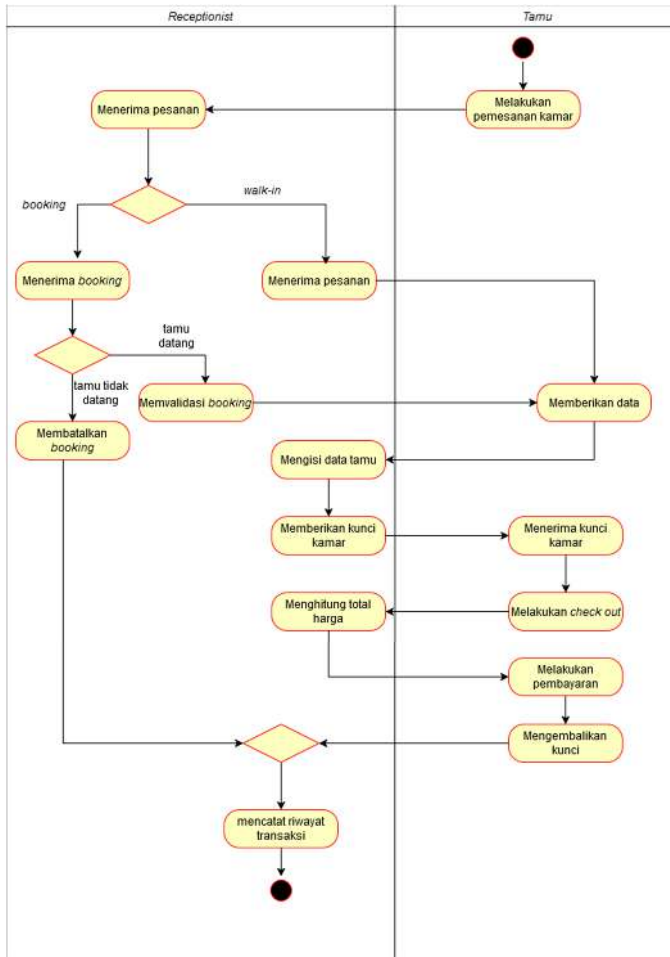
Use Case Name	Mengirim <i>Booking</i>
Actor	<i>Third Party</i>
Description	Pengguna (<i>Third Party</i>) mengirim <i>booking</i> yang dilakukan kepada pihak hotel.
Normal Course: (Actor's Action)	<ol style="list-style-type: none"> 1. <i>Third Party</i> menerima permintaan <i>booking</i> dari pengguna aplikasi <i>third party</i>. 2. <i>Third Party</i> mengirimkan permintaan pada pihak hotel.
Alternate Course: (Actor's Action)	<ol style="list-style-type: none"> 1. <i>Third Party</i> tidak menerima permintaan <i>booking</i>. 2. <i>Third Party</i> tidak mengirimkan permintaan pada pihak hotel.
Pre-Condition	<i>Third Party</i> melakukan kemitraan dengan hotel.
Post-Condition	Pihak hotel menerima permintaan <i>booking</i> .
Assumption	

C. Class Diagram



Gambar 4. 4 Class Diagram Sistem Baru

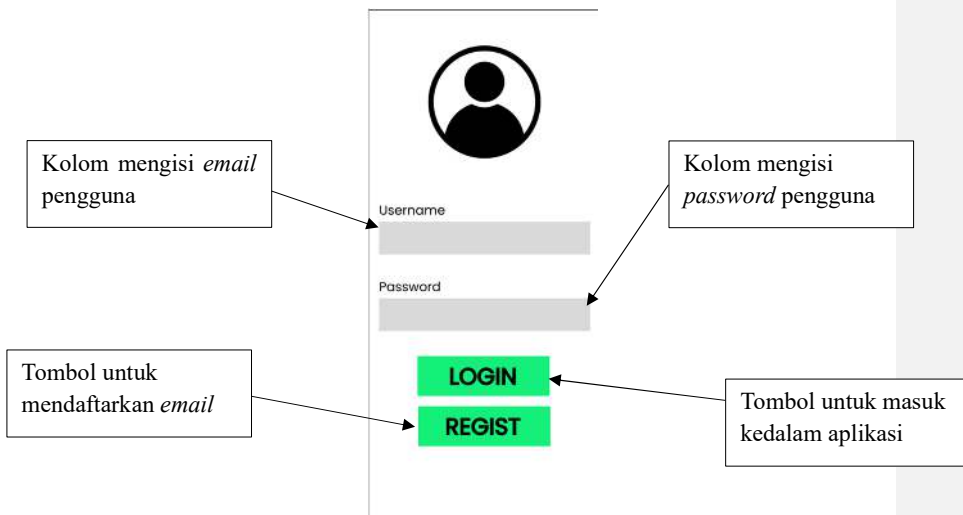
D. *Activity Diagram*



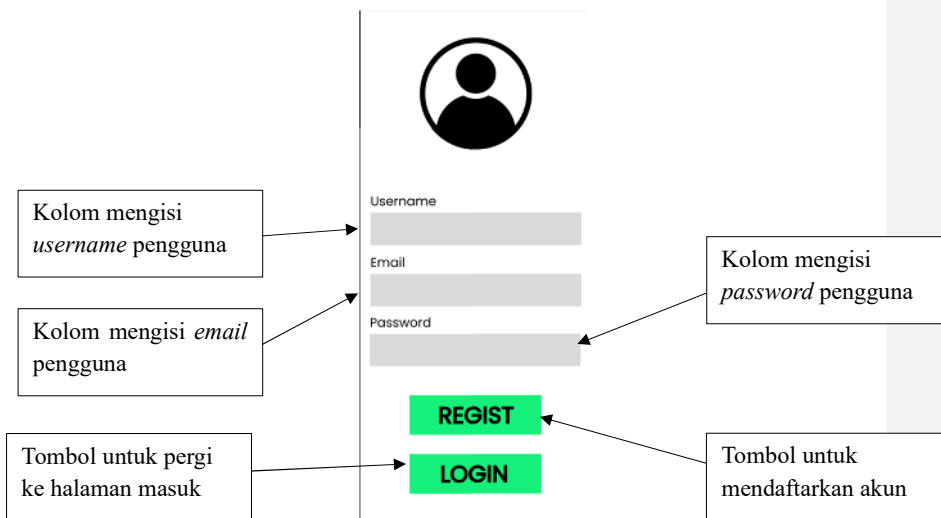
Gambar 4. 5 Activity Diagram Sistem Baru

4.2.2 Pemodelan Antarmuka Aplikasi

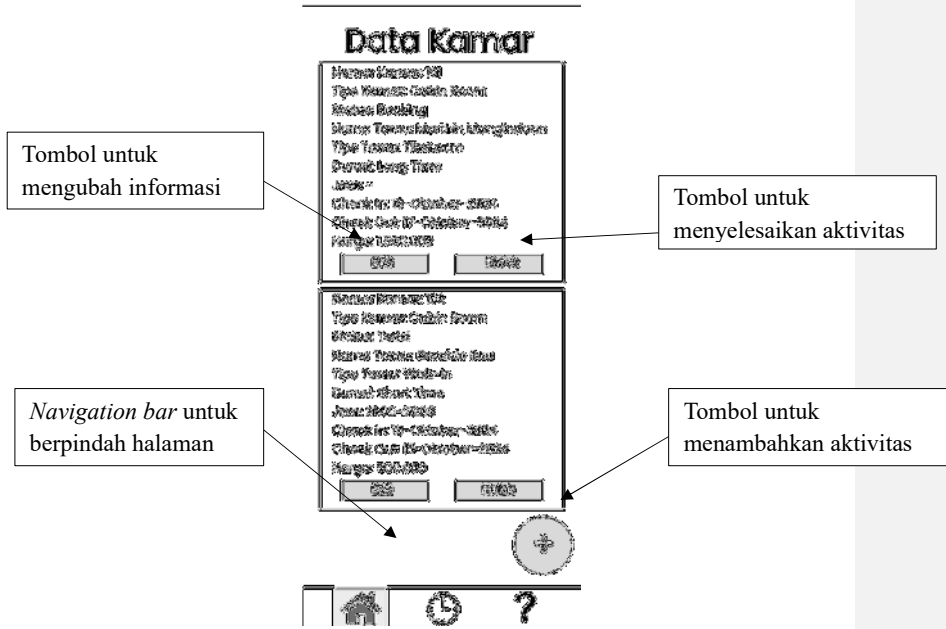
Berikut adalah desain antarmuka dari aplikasi yang akan dibangun untuk menunjang pencatatan kamar hotel.



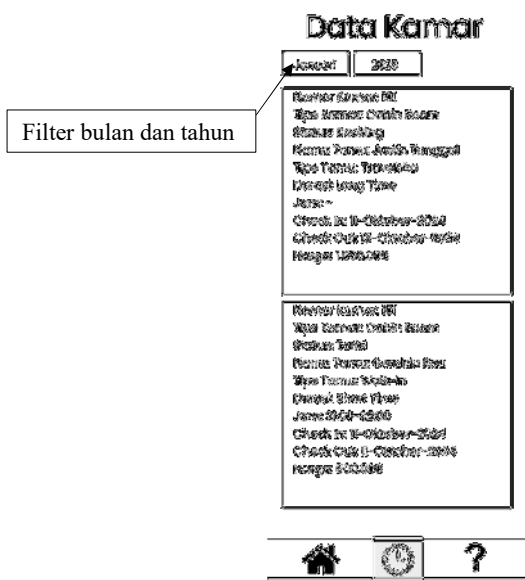
Gambar 4. 6 Desain Halaman Masuk



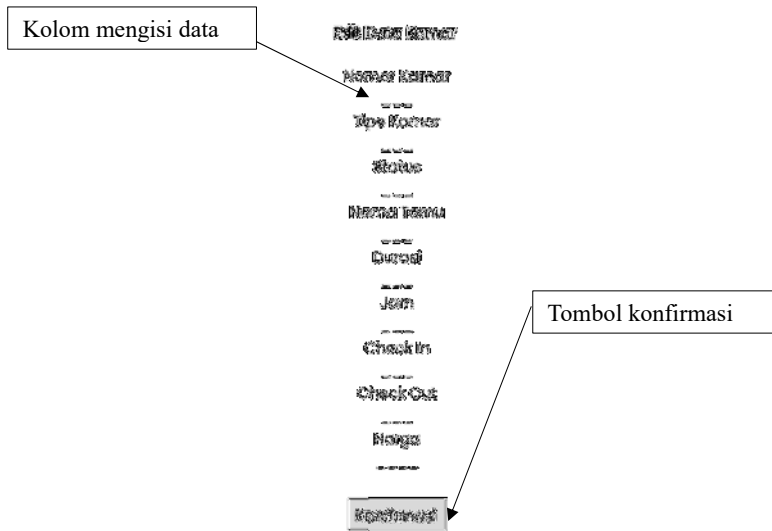
Gambar 4. 7 Desain Halaman Daftar



Gambar 4. 8 Desain Halaman Aktivitas - Receptionist



Gambar 4. 9 Desain Halaman Riwayat - Receptionist



Gambar 4. 10 Desain Halaman Tambahkan Aktivitas

Daftar Kamar

Kamar Double 101, 102, 103, 104, 201, 202	Harga: 400.000
Kamar Single 201, 202, 401, 402	Harga: 200.000
Kamar Double 301, 302, 401, 402, 501, 502	Harga: 300.000
Kamar Double 601, 602, 401, 402, 501, 502	Harga: 300.000
Kamar Double 701, 702, 401, 402, 501, 502	Harga: 300.000
Kamar Double 801, 802, 401, 402, 501, 502	Harga: 300.000

Home | Refresh | ?

Gambar 4. 11 Desain Halaman Tentang Kamar

4.2.3 Kamus Data

Berikut adalah kamus data untuk basis data yang dipakai:

Tabel 4. 15 Kamus Data

No	Tabel	Kolom	Tipe Data	Deskripsi
1	Kamar	ID	INT	ID unik kamar
2	Kamar	Nomor Kamar	INT	
3	Kamar	Tipe Kamar	String	
4	Kamar	Status Kamar	String	
5	Kamar	Nama Tamu	String	
6	Kamar	Tipe Tamu	String	<i>Booking,walk-in</i>
7	Kamar	Jam	Time stamp	<i>Jam check in-check out</i>
8	Kamar	Check In	Date	<i>Tanggal check in</i>
9	Kamar	Check Out	Date	<i>Tanggal check out</i>
10	Kamar	Durasi	String	<i>Short time/Long time</i>
11	Kamar	Harga	Double	Harga yang harus dibayarkan tamu
12	History	ID	INT	ID unik history
13	History	Nomor Kamar	INT	
14	History	Tipe Kamar	String	
15	History	Status Kamar	String	
16	History	Nama Tamu	String	
17	History	Tipe Tamu	String	<i>Booking,walk-in</i>
18	History	Jam	Time stamp	<i>Jam check in-check out</i>
19	History	Check In	Date	<i>Tanggal check in</i>
20	History	Check Out	Date	<i>Tanggal check out</i>
21	History	Durasi	String	<i>Short time/Long time</i>
22	History	Harga	Double	Harga yang harus dibayarkan tamu
23	User	ID	INT	ID unik user
24	User	Nickname	String	

4.2.4 Pemodelan Modul Program

Berikut adalah pemodelan modul program menggunakan *Pseudocode* untuk tiap fitur yang ada dalam aplikasi.

A. Login

```
1. BEGIN
2. DATABASE users = [
3.   {email: "user@example.com", password: "userpass", role: "user"},
4.   {email: "admin@example.com", password: "adminpass", role: "admin"}
5. ]
6. FUNCTION login()
7. WHILE TRUE
8.   DISPLAY "Enter your email:"
9.   INPUT email
10.  DISPLAY "Enter your password:"
11.  INPUT password
12.  // Validasi email dan password
13.  FOR user IN users
14.    IF user.email == email AND user.password == password THEN
15.      IF user.role == "admin" THEN
16.        DISPLAY "Welcome Admin! Redirecting to admin dashboard..."
17.        CALL adminDashboard()
18.      ELSE IF user.role == "user" THEN
19.        DISPLAY "Welcome User! Redirecting to user dashboard..."
20.        CALL userDashboard()
21.      END IF
22.      RETURN
23.    END IF
24.  END FOR
25.  DISPLAY "Invalid email or password. Please try again."
26. END WHILE
27. END FUNCTION
28. FUNCTION adminDashboard()
29.  DISPLAY "You are now in the admin dashboard."
30.  // Tambahkan fitur khusus untuk admin di sini
31. END FUNCTION
32. FUNCTION userDashboard()
33.  DISPLAY "You are now in the user dashboard."
34.  // Tambahkan fitur khusus untuk pengguna di sini
35. END FUNCTION
36. FUNCTION main()
37.  CALL login()
38. END FUNCTION
39. CALL main()
40. END
```

B. Register

```
1. BEGIN
2. DATABASE users = []
3. FUNCTION register()
4. WHILE TRUE
5.   DISPLAY "Enter your email:"
6.   INPUT email
7.   // Periksa apakah email sudah terdaftar
8.   FOR user IN users
9.     IF user.email == email THEN
10.      DISPLAY "Email is already registered. Please use a different email."
11.      RETURN
12.     END IF
13.   END FOR
```



```

14.  DISPLAY "Enter your username:"
15.  INPUT username
16.  DISPLAY "Enter your password:"
17.  INPUT password
18.  // Simpan data pengguna ke database
19.  ADD {email: email, username: username, password: password} TO users
20.  DISPLAY "Registration successful! You are now registered."
21.  RETURN
22.  END WHILE
23.  END FUNCTION
24.  FUNCTION main()
25.    DISPLAY "Welcome to the registration system!"
26.    CALL register()
27.  END FUNCTION
28.  CALL main()
29.  END

```

C. Menampilkan Aktivitas Yang Sedang Berlangsung

```

1.  BEGIN
2.  FUNCTION fetchDataFromFirestore()
3.    // Inialisasi instance Firebase Firestore
4.    firestore = FirebaseFirestore.getInstance()
5.    // Ambil koleksi "kamar" dari Firestore
6.    firestore.collection("kamar").get()
7.    .onSuccess(documents)
8.      FOR EACH document IN documents
9.        data = document.data.toMutableMap()
10.       data["documentId"] = document.id
11.       kamarList.add(data)
12.     END FOR
13.    kamarAdapter = KamarAdapter(kamarList, firestore)
14.    // Set adapter ke RecyclerView
15.    recyclerView.adapter = kamarAdapter
16.    // Beritahu adapter bahwa data telah berubah
17.    kamarAdapter.notifyDataSetChanged()
18.  END onSuccess
19.  .onFailure(exception)
20.    // Log kesalahan jika pengambilan data gagal
21.    LOG("Firestore", "Error", exception)
22.  END onFailure
23.  END FUNCTION
24.  END

```

D. Menampilkan Aktivitas Yang Telah Berakhir

```

1.  Function fetchHistoryData(adapter):
2.    Access "history" collection from Firestore
3.    Try to get data:
4.      If successful:
5.        For each document in the retrieved data:
6.          Extract document data
7.          Add data to historyList
8.        Notify adapter that data has been updated
9.      If failed:
10.       Log an error message with exception details

```

E. Menambahkan Aktivitas Baru

```

1.  Function updateData(documentId, data):
2.    Initialize Firestore instance

```

3. Access Firestore collection "kamar" and the specified document
4. Attempt to update the document with the given data:
5. If successful:
6. Log "Data berhasil diperbarui"
7. Close the activity
8. If failed:
9. Log "Error saat memperbarui data" with exception details
10. Function loadData(documentId):
11. Initialize Firestore instance
12. Access Firestore collection "kamar" and the specified document
13. Attempt to retrieve the document:
14. If successful:
15. If the document exists:
16. Set nomorKamarEditText to the value of "nomorKamar"
17. Set tipeKamarEditText to the value of "tipeKamar"
18. Set statusKamarEditText to the value of "statusKamar"
19. Set namaTamuEditText to the value of "namaTamu"
20. Set tipeTamuEditText to the value of "tipeTamu"
21. Set durasiEditText to the value of "durasi"
22. Set jamEditText to the value of "jam"
23. Set checkInEditText to the value of "checkIn"
24. Set checkOutEditText to the value of "checkOut"
25. Set hargaEditText to the value of "harga"
- 26.
27. Function pushDataWithAutoGeneratedId(data):
28. Initialize Firestore instance
29. Access Firestore collection "kamar"
30. Attempt to add the given data to the collection with an auto-generated ID:
31. If successful:
32. Log "Data berhasil disimpan dengan ID: <generated ID>"
33. Close the activity
34. If failed:
35. Log "Error saat menyimpan data" with exception details

F. Memindahkan Aktivitas

1. Function getItemCount():
2. Return the size of kamarList
3. Function deleteKamar(documentId, position):
4. Get kamar data from kamarList at the given position
5. Access Firestore collection "kamar" and document with documentId
6. Attempt to delete the document:
7. If successful:
8. Add the kamar data to the Firestore collection "history":
9. If successful:
10. Log "Data kamar berhasil disimpan ke history"
11. If failed:
12. Log "Error menyimpan data ke history" with exception details
13. Remove kamar data from kamarList at the given position
14. Notify that the item has been removed from the adapter
15. Log "Pesanan Selesai"
16. If failed:
17. Log "Error Menyelesaikan Pesanan" with exception details

G. Filter Bulan Dan Tahun

1. Function setupSpinners():
2. Create a list of years from 2024 to 2045 as strings
3. Create a list of months with their names
4. Set yearSpinner's adapter to use the list of years
5. Set monthSpinner's adapter to use the list of months
6. Define a listener for spinner item selection:

7.	On item selected:
8.	Call fetchHistoryData with the adapter
9.	On nothing selected:
10.	Do nothing
11.	Assign the listener to yearSpinner
12.	Assign the listener to monthSpinner

4.3 Implementasi

Berikut adalah pembahasa mengenai implementasi aplikasi yang telah dibuat.

4.3.1 Lingkungan Implementasi

Berikut adalah lingkungan implementasi yang mencakup perangkat lunak dan perangkat keras yang dipakai selama proses pembangunan dan pengembangan aplikasi.

4.3.1.1 Software

Tabel 4. 16 Software Yang Dipakai

Nama <i>Software</i>	Kegunaan
Android Studio versi 2023.3.1	Sebagai <i>compiler</i> untuk menjalankan pemrograman android
Figma 16.13.3	Sebagai media untuk mendesain antarmuk aplikasi
Draw.io 26.0.4	Sebagai media untuk membuat <i>diagram</i> untuk kepentingan pemodelan sistem
Firebase 13.0.2	Sebagai <i>database</i> yang menyimpan data dari aplikasi

4.3.1.2 Hardware

Tabel 4. 17 Hardware Yang Dipakai

Nama <i>Hardware</i>	Kegunaan
Lenovo Ideapad Slim 3	Sebagai alat untuk mengerjakan kode, desain, dan pembuatan laporan

4.3.2 Manajemen Perubahan

Selama proses pembuatan aplikasi, terjadi perubahan pada judul penelitian karena miskomunikasi.

Tabel 4. 18 Manajemen Perubahan

Perubahan yang dilakukan	Alasan
Pergantian judul KP	Terjadi miskomunikasi sehingga judul yang dipilih terlalu luas, sedangkan permintaan dan kebutuhan pihak

	Mottomo Inn tidak seluas judul sebelumnya.
--	--

4.3.3 Aturan Bisnis

Berikut adalah aturan bisnis yang diterapkan pada aplikasi yang telah dibuat.

1. *Login* dan *Register*

1. Tipe pengguna *Manager* akan memiliki akun khusus yang tidak perlu melakukan *register*.
2. Tipe pengguna *Receptionist* harus melakukan register terlebih dahulu.
3. Setiap tipe pengguna harus melakukan *login* kembali ketika menutup aplikasi.

2. Memasukkan data

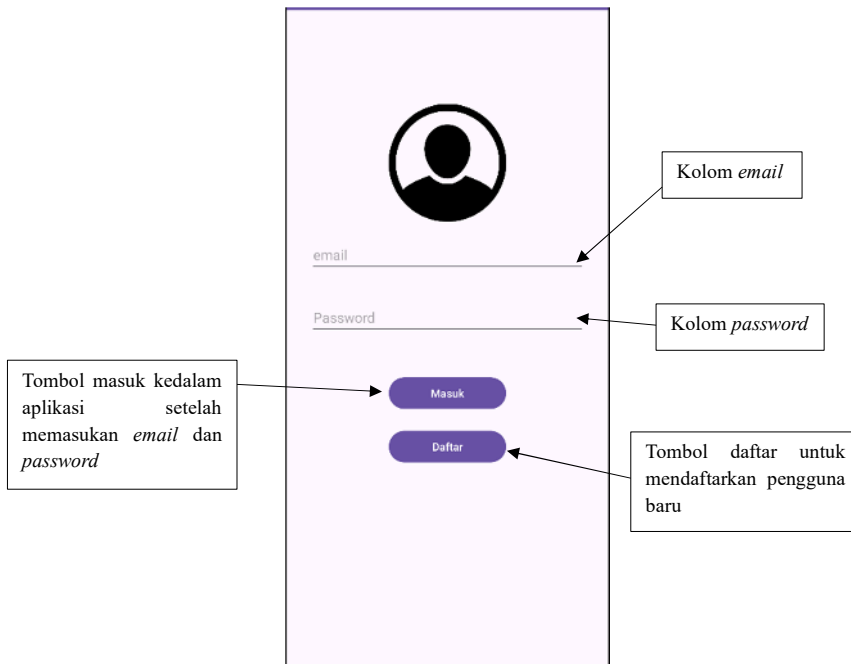
1. Tipe pengguna *Manager* tidak dapat memasukkan data kedalam sistem.
2. Tipe pengguna *Receptionist* dapat memasukkan data kedalam sistem.
3. Pemantauan aktivitas dan *history* Tipe pengguna *Manager* dan *Receptionist* dapat melihat aktivitas yang sedang berlangsung dan aktivitas yang telah selesai atau *history*.
4. Tipe pengguna *Receptionist* dapat melakukan mengubah data dan menyelesaikan aktivitas dan memindahkannya pada halaman *history*.

3. Menampilkan data

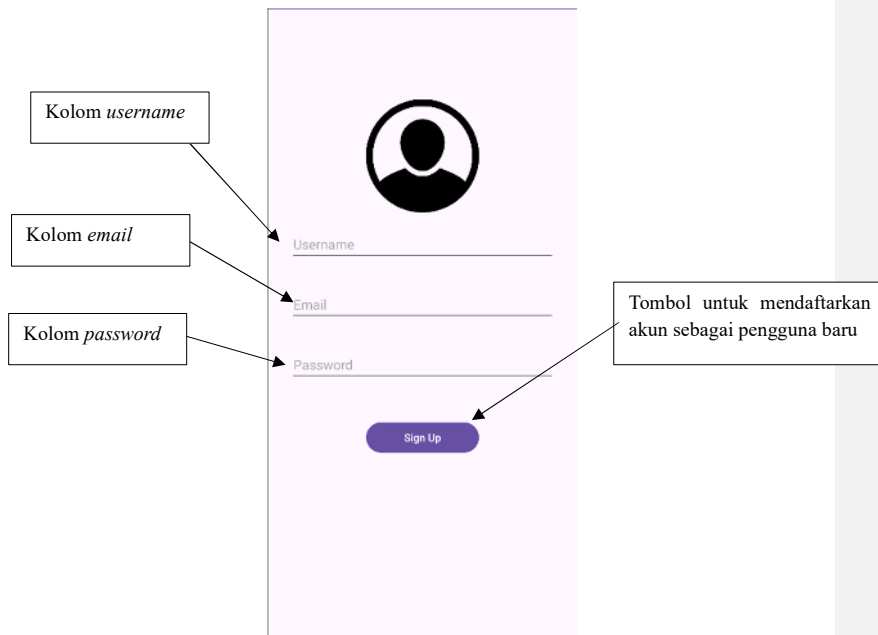
1. Tipe pengguna *Manager* dan *Receptionist* mendapatkan tampilan informasi data transaksi kamar yang telah dimasukan *Receptionist* sebelumnya.
2. Data yang ditampilkan akan berurut berdasarkan data yang dimasukan paling baru.

4.3.3 Implementasi Antarmuka Aplikasi

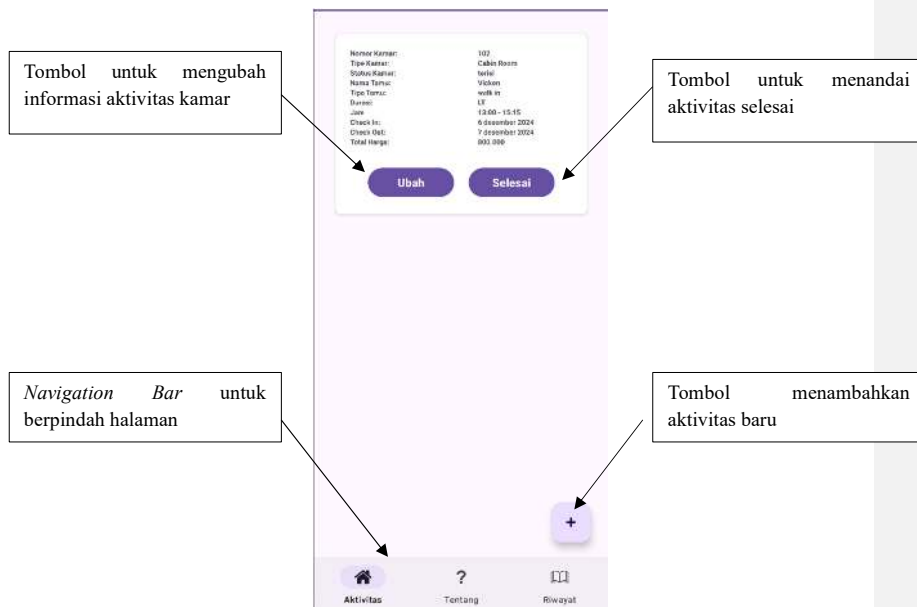
Berikut adalah implementasi antarmuka dari aplikasi yang telah dibangun:



Gambar 4. 14 Halaman Masuk



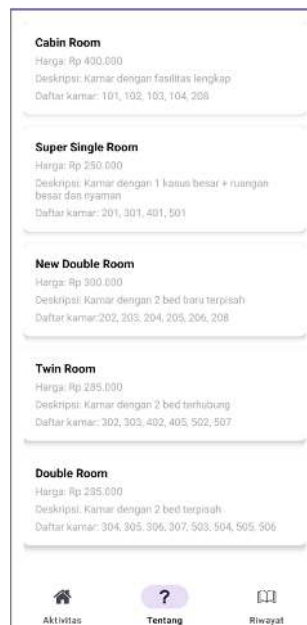
Gambar 4. 15 Halaman Daftar



Gambar 4. 16 Halaman Aktivitas - Receptionist



Gambar 4. 17 Halaman Riwayat



Gambar 4. 18 Halaman Tentang

Nomor Kamar

Tipe Kamar

Tipe Kamar

Status Kamar

Status Kamar

Nama Tamu

Nama Tamu

Tipe Tamu

Tipe Tamu

Durasi

Durasi Menginap

Jam

Jam masuk - keluar

Check In

Tanggal Check in

Check Out

Tanggal Check Out

Harga

Harga Yang Dibayarkan

Kirim

Formulir data transaksi

Tombol untuk mengirim informasi

Gambar 4. 19 Halaman Ubah/Tambah Data

Nomor Kamar:	102
Tipe Kamar:	Calit-Roson
Status Kamar:	Wolke
Nama Tamu:	Wolke
Tipe Tamu:	Wolke
Durasi:	12
Jam:	19:00 - 15:15
Check In:	6 Desember 2024
Check Out:	7 Desember 2024
Total Harga:	800.000

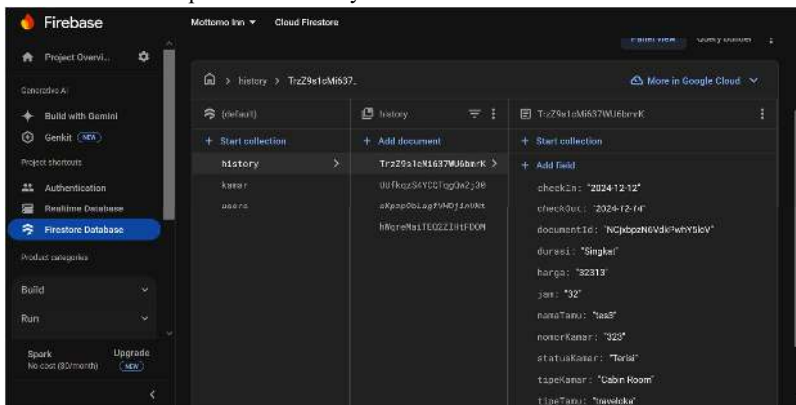
Aktivitas Tentang Riwayat

Gambar 4. 20 Halaman Aktivitas - Manager

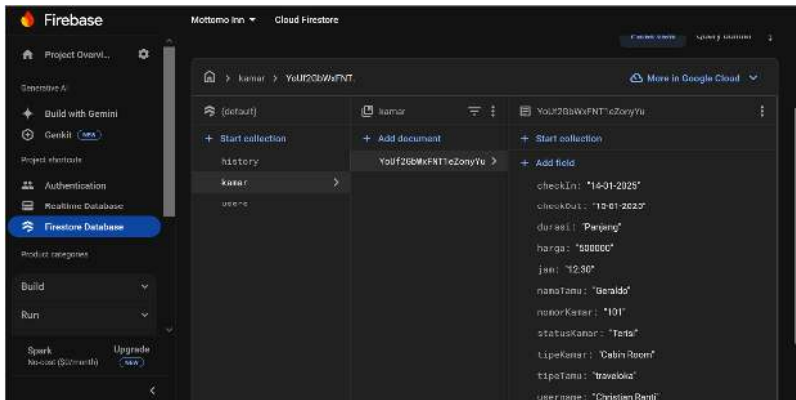
4.3.4 Implementasi Basis Data

Basis data yang digunakan pada aplikasi yang dibangun adalah *Firebase* yang merupakan layanan basis data gratis dari google. *Firebase* memiliki kekurangan dibagian fitur yang bisa diakses karena harus membayar, namun dengan fitur yang terbatas untuk digunakan yang dimaksimalkan dengan baik dapat membantu pengembang dalam mengembangkan aplikasi.

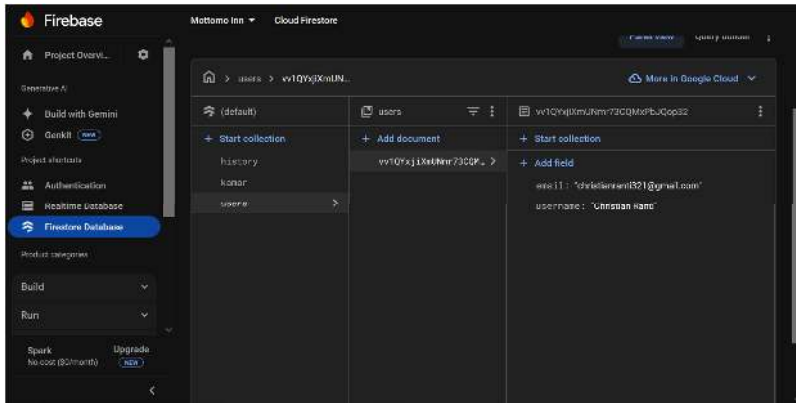
Fitur yang dipakai dalam aplikasi adalah *Firestore* yang memungkinkan penggunaan secara *real-time* dimana data tersebut disimpan didalam *cloud* yang dapat diakses selama memiliki koneksi internet yang stabil dari penggunanya. Berikut adalah tampilan basis datanya.



Gambar 4. 21 Database – History



Gambar 4. 22 Database - Kamar



Gambar 4. 23 Database - User

4.3.5 Implementasi Modul Program

Berikut adalah implementasi dari modul program yang telah dibuat.

A. Login

```

signup_button.setOnClickListener {
    val intent = Intent(this, SignUp::class.java)
    startActivity(intent)
}
login_button.setOnClickListener {
    val email = edit_email.text.toString().trim()
    val password = edit_password.text.toString().trim()
    if (email == "mottomoinnadmin@gmail.com" && password == "adminmottomoinn") {
        val intent = Intent(this, ManagerMain::class.java)
        startActivity(intent)
        finish() // Mengakhiri aktivitas saat ini
    } else {
        login(email, password)
    }
}
mAuth.signInWithEmailAndPassword(email, password)
    .addOnCompleteListener(this) { task ->
        if (task.isSuccessful) {
            val intent = Intent(this@LogIn, MainActivity::class.java)
            startActivity(intent)
            finish() // Mengakhiri aktivitas saat ini
        } else {
            Toast.makeText(this@LogIn, "User Does Not Exist", Toast.LENGTH_SHORT).show()
        }
    }
}
}

```

B. Register

```

signup_button.setOnClickListener {
    val email = edit_email.text.toString()
    val password = edit_password.text.toString()
    signup(email,password);
}
}
private fun signup(email: String, password: String){

```

```

mAuth.createUserWithEmailAndPassword(email, password)
    .addOnCompleteListener(this) { task ->
        if (task.isSuccessful) {
            val intent = Intent(this@SignUp, MainActivity::class.java)
            startActivity(intent)
        } else {
            Toast.makeText(this@SignUp, "Some Error Occurred", Toast.LENGTH_SHORT).show()
        }
    }
}
}
}

```

C. Menampilkan Aktivitas Yang Sedang Berlangsung

```

private fun fetchDataFromFirestore() {
    val firestore = FirebaseFirestore.getInstance()
    firestore.collection("kamar")
        .get()
        .addOnSuccessListener { documents ->
            for (document in documents) {
                val data = document.data.toMutableMap()
                data["documentId"] = document.id
                kamarList.add(data)
            }
            kamarAdapter = KamarAdapter(kamarList, firestore)
            recyclerView.adapter = kamarAdapter
            kamarAdapter.notifyDataSetChanged()
        }
        .addOnFailureListener { exception ->
            Log.w("Firestore", "Error saat mengambil data", exception)
        }
}
}
}

```

D. Menampilkan Aktivitas Yang Telah Berakhir

```

private fun fetchHistoryData(adapter: HistoryAdapter) {
    firestore.collection("history")
        .get()
        .addOnSuccessListener { documents ->
            for (document in documents) {
                val data = document.data
                historyList.add(data)
            }
            adapter.notifyDataSetChanged()
        }
        .addOnFailureListener { exception ->
            Log.e("HistoryActivity", "Error mendapatkan data dari history", exception)
        }
}
}
}

```

E. Menambahkan Aktivitas Baru

```

private fun updateData(documentId: String, data: Map<String, Any>) {
    val firestore = FirebaseFirestore.getInstance()
    firestore.collection("kamar")
        .document(documentId)
        .update(data)
        .addOnSuccessListener {
            Log.d("Firestore", "Data berhasil diperbarui")
            finish() // Tutup aktivitas
        }
}
}

```

```

        .addOnFailureListener { exception ->
            Log.w("Firestore", "Error saat memperbarui data", exception)
        }
    }
}
private fun loadData(documentId: String) {
    val firestore = FirebaseFirestore.getInstance()
    firestore.collection("kamar")
        .document(documentId)
        .get()
        .addOnSuccessListener { document ->
            if (document != null) {
                nomorKamarEditText.setText(document.getString("nomorKamar"))
                tipeKamarEditText.setText(document.getString("tipeKamar"))
                statusKamarEditText.setText(document.getString("statusKamar"))
                namaTamuEditText.setText(document.getString("namaTamu"))
                tipeTamuEditText.setText(document.getString("tipeTamu"))
                durasiEditText.setText(document.getString("durasi"))
                jamEditText.setText(document.getString("jam"))
                checkInEditText.setText(document.getString("checkIn"))
                checkOutEditText.setText(document.getString("checkOut"))
                hargaEditText.setText(document.getString("harga"))
            }
        }
    }
}
private fun pushDataWithAutoGeneratedId(data: Map<String, Any>) {
    val firestore = FirebaseFirestore.getInstance()
    firestore.collection("kamar")
        .add(data)
        .addOnSuccessListener { documentReference ->
            Log.d("Firestore", "Data berhasil disimpan dengan ID: ${documentReference.id}")
            finish()
        }
        .addOnFailureListener { exception ->
            Log.w("Firestore", "Error saat menyimpan data", exception)
        }
    }
}
}
}
}
}

```

F. Memindahkan Aktivitas

```

override fun getItemCount(): Int = kamarList.size
private fun deleteKamar(documentId: String, position: Int) {
    val kamar = kamarList[position]
    firestore.collection("kamar").document(documentId)
        .delete()
        .addOnSuccessListener {
            firestore.collection("history").add(kamar)
            .addOnSuccessListener {
                Log.d("Firestore", "Data kamar berhasil disimpan ke history")
            }
            .addOnFailureListener { exception ->
                Log.w("Firestore", "Error menyimpan data ke history: $exception")
            }
        }
        kamarList.removeAt(position)
        notifyItemRemoved(position)
        Log.d("Firestore", "Pesanan Selesai")
    }
    .addOnFailureListener { exception ->
        Log.w("Firestore", "Error Menyelesaikan Pesanan: $exception")
    }
}
}
}
}
}

```

G. Filter Bulan Dan Tahun

```
private fun setupSpinners() {
    val years = (2024..2045).map { it.toString() }
    val months = listOf("January", "February", "March", "April", "May", "June",
        "July", "August", "September", "October", "November", "December")
    yearSpinner.adapter = ArrayAdapter(this, android.R.layout.simple_spinner_item, years)
    monthSpinner.adapter = ArrayAdapter(this, android.R.layout.simple_spinner_item, months)
    val listener = object : AdapterView.OnItemClickListener {
        override fun onItemClick(parent: AdapterView<*>?, view: View?, position: Int, id: Long) {
            fetchHistoryData(adapter)
        }
    }
    override fun onNothingSelected(parent: AdapterView<*>?) {}
}
```

4.4 Pengujian

Pada tahap ini akan membahas tentang pengujian aplikasi yang dilakukan Yang berisi tujuan pengujian, kriteria pengujian, kasus pelaksanaan pengujian, dan analisis hasil pengujian yang telah dilakukan.

4.4.1 Tujuan Pengujian

Pengujian dilakukan untuk menguji apakah aplikasi yang dibuat dapat berjalan dengan baik yang dapat mencegah kesalahan yang bisa saja terjadi saat penggunaan sehari-hari.



4.4.2 Kasus Pengujian

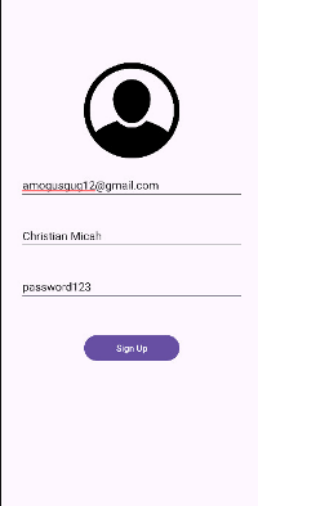
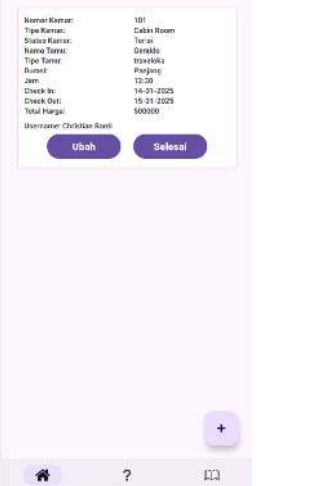
Berikut adalah kasus pengujian yang dilakukan untuk menguji apakah aplikasi yang dibangun telah memenuhi kriteria pengujian atau tidak:



1. Kasus Pengujian Masuk Ke Aplikasi – *receptionist*
2. Kasus Pengujian Masuk Ke Aplikasi – *manager*
3. Kasus Pengujian Daftar Akun – *receptionist*
4. Kasus Pengujian Menampilkan Aktivitas Yang Sedang Berlangsung
5. Kasus Pengujian Menampilkan Aktivitas Yang Telah Berakhir
6. Kasus Pengujian Menambahkan Aktivitas Baru
7. Kasus Pengujian Mengubah Data Aktivitas Yang Sedang Berjalan
8. Kasus Pengujian Menyelesaikan Aktivitas Yang Sedang Berjalan
9. Kasus Pengujian Filter Tahun Dan Bulan Pada Halaman Riwayat.


4.4.3 Pelaksanaan Pengujian

Tabel 4. 19 Pelaksanaan Pengujian Aplikasi

No	Gambar	Kasus Pengujian	Hasil Pengujian
1		Masuk Ke Aplikasi – <i>receptionist</i>	Berhasil masuk dengan akun tipe pengguna <i>receptionist</i> .
2		Masuk Ke Aplikasi – <i>Manager</i>	Berhasil masuk dengan akun tipe pengguna <i>manager</i> .

3		Daftar Akun – <i>receptionist</i>	Berhasil mendaftarkan akun tipe pengguna <i>receptionist</i> baru.
4		Menampilkan Aktivitas Yang Sedang Berlangsung	Berhasil menampilkan aktivitas yang sedang berlangsung.

<p>5</p>		<p>Menampilkan Data Aktivitas Yang Telah Berakhir</p>	<p>Berhasil menampilkan aktivitas yang telah berakhir.</p>
<p>6</p>		<p>Menambahkan Aktivitas Baru</p>	<p>Berhasil menambahkan aktivitas baru</p>

7	<p style="text-align: center;">Data Kamar</p> <p>Nomor Kamar 101</p> <p>Tipe Kamar Cabin Room</p> <p>Status Kamar Terisi</p> <p>Nama Tamu Geraldo</p> <p>Tipe Tamu traveloka</p> <p>Durasi Panjang</p> <p>Jam 12.30</p> <p>Check In 14-01-2025</p>	Mengubah Data Aktivitas Yang Sedang Berjalan	Berhasil mengubah data aktivitas yang sedang berjalan
8		Menyelesaikan Aktivitas Yang Sedang Berjalan	Berhasil menyelesaikan informasi tentang aktivitas yang sedang berjalan dan muncul di halaman riwayat.

9		Filter Bulan Dan Tahun Pada Halaman Riwayat	Berhasil menyaring aktivitas berdasarkan bulan dan tahun pada halaman riwayat.
---	---	---	--

4.4.4 Analisis Hasil Pengujian

Berikut adalah hasil analisis yang telah dilakukan setelah pelaksanaan pengujian mandiri yang telah dilakukan:

1. Aplikasi yang dibangun sesuai dapat berjalan dengan baik pada perangkat berbasis *Android*.
2. Fitur yang ada pada aplikasi berfungsi dengan baik dan mempermudah dalam proses pencatatan.
3. Aplikasi terhubung dengan *database* secara *real-time*.
4. Kecepatan aplikasi bergantung pada koneksi internet dari pengguna.

BAB V

KESIMPULAN DAN SARAN

Pada bab ini akan berisi kesimpulan dan saran oleh pengembang terkait penelitian yang dilakukan dan aplikasi yang telah dibangun.

5.1 Kesimpulan

Berikut adalah kesimpulan yang diperoleh setelah melakukan pengujian terhadap aplikasi yang telah dibangun dan hasil dari *User Acceptance Test* (UAT):

1. Fitur-fitur pada aplikasi dapat berjalan dengan baik
2. Kecepatan dari setiap fitur untuk diproses bergantung pada koneksi internet dari pengguna.
3. Aplikasi dapat dengan mudah digunakan dan dipahami oleh pihak hotel.
4. Pihak hotel merasa puas dengan aplikasi yang dibuat.
5. Aplikasi dapat menyelesaikan permasalahan yang dihadapi pihak hotel.

5.2 Saran

Adapun saran untuk pengembangan selanjutnya untuk Aplikasi *Online* Pencatatan Transaksi Kamar Hotel Berbasis *Android* adalah sebagai berikut:

1. Menambahkan fitur *Auto-refresh* yang dapat melakukan *refresh* secara otomatis ketika ada data baru yang ditambahkan.
2. Menambahkan fitur notifikasi ketika ada sebuah aktivitas yang sebentar lagi selesai.
3. Menambahkan perhitungan harga otomatis beserta perhitungan potongan harga.
4. Menambahkan fitur *search* untuk melakukan pencarian riwayat atau aktivitas.
5. Menambahkan fitur pengaturan kamar.
6. Menambahkan fitur penanggung jawab setiap aktivitas.
7. Membangun aplikasi yang bisa dijalankan pada *Operating System* lain seperti IOS.

DAFTAR PUSTAKA

- [1] R. C. Wood, *Hospitality Management: A Brief Introduction*, Los Angeles: SAGE Publication Ltd, 2015.
- [2] B. Phillips, C. Stewart and K. Marsicano, *Android Programming: The Big Nerd Ranch Guide 4th edition*, Atlanta: Big Nerd Ranch, 2019.
- [3] J. Skeen and D. Greenhalgh, *Kotlin Programming: The Big Nerd Ranch Guide*, Atlanta: Big Nerd Ranch, 2018.
- [4] N. Smyth, *Firestore Essentials - Android Edition*, South Carolina: Createspace Independent Pub, 2017.
- [5] P. Chougale, V. Yadav and A. Gaikwa, "Firestore - Overview And Usage," *International Research Journal of Modernization in Engineering Technology and Science*, vol. III, no. 12, pp. 1178-1183, 2021.
- [6] E. Pawan, R. Thamrin, P. Hasan, S. Bei and P. Matu, "Using Waterfall Method to Design Information System of SPMI STIMIK Sepuluh Nopember Jayapura," *International Journal of Computer and Information System (IJCIS)*, vol. II, no. 2, pp. 34-39, 2021.
- [7] V. S. Balasubramaniam, P. R. R. Thumati, P. Kanchi, R. Agarwal, O. Goel and E. A. Shrivastav, "Evaluating The Impact Of Agile And Waterfall Methodologies In Large Scale It Projects," *International Journal Of Progressive Reaserch Engineer Management And Science (IJPREAMS)*, vol. III, no. 12, pp. 398-412, 2023.
- [8] M. N. Arifin and D. Siahaan, "Structural and Semantic Similarity Measurement of UML Use Case Diagram," *Lontar Komputer*, vol. XI, no. 2, pp. 88-100, 2020.
- [9] M. Seidl, M. Scholz and C. Huemer, *UML @ Classroom: An Introduction to Object-Oriented Modeling*, New York: Springer, 2015.
- [10] D. McGreal and R. Jocham, *The Professional Product Owner: Leveraging Scrum as a Competitive Advantage*, Seattle: Addison-Wesley, 2018.
- [11] H. Alaidaros, M. Omar and R. Romli, "The state of the art of agile kanban method: challenges and opportunities," *Independent Journal of Management & Production*, vol. XII, no. 8, pp. 2535-2550, 2021.

Lampiran A

Nama Pengguna : OLIVIA KASIMAN Tanggal : 9 Desember 2024
Tipe Pengguna : Manager

Tabel pengujian fitur aplikasi:

Kasus pengujian	Diterima	Tidak diterima
Login	✓	
Menampilkan aktivitas yang sedang berlangsung	✓	
Menampilkan aktivitas yang telah selesai	✓	

No	Pertanyaan	Jawaban
1	Apakah aplikasi ini dapat menyelesaikan permasalahan yang dialami?	YA
2	Apakah aplikasi ini dapat dengan mudah dimengerti dan dipakai?	YA
3	Apa kesan pertama anda saat menggunakan aplikasi ini?	BAGUS, MEMBANTU
4	Apa kritik dan saran anda terhadap aplikasi ini?	BUATKAN DESAIN YANG LEBIH BAGUS

Tanda Tangan Penguji


OLIVIA KASIMAN

Lampiran B

Nama Pengguna : *Luis Lpam* Tanggal : *9 Desember 2024*
Tipe Pengguna : *Receptionist*

Tabel pengujian fitur aplikasi:

Kasus pengujian	Diterima	Tidak diterima
Masuk kedalam aplikasi	✓	
Daftar akun	✓	
Menambahkan data aktivitas	✓	
Mengubah data aktivitas	✓	
Menyelesaikan aktivitas	✓	
Menampilkan aktivitas yang sedang berlangsung	✓	
Menampilkan aktivitas yang telah selesai	✓	

No	Pertanyaan	Jawaban
1	Apakah aplikasi ini dapat menyelesaikan permasalahan yang dialami?	<i>Ya, Dapat menyelesaikan</i>
2	Apakah aplikasi ini dapat dengan mudah dimengerti dan dipakai?	<i>Lumayan</i>
3	Apa kesan pertama anda saat menggunakan aplikasi ini?	<i>Baik, does dikembangkan lagi</i>
4	Apa kritik dan saran anda terhadap aplikasi ini?	<i>memperbaiki desain aplikasi</i>

Tanda Tangan Penguji



Lampiran C

Wawancara: *Receptionist*

Berikut adalah wawancara yang dilakukan dengan *receptionist*:

- Penulis : Bagaimana prosedur pengunjung memesan kamar?
- Receptionist* : Bisa datang langsung bisa juga pesan lewat aplikasi seperti Traveloka dan Tiket.com, tapi hampir 80% pesan lewat aplikasi karena banyak promonya.
- Penulis : Apakah ada kendala saat pengunjung datang dan memesan kamar?
- Receptionist* : Kendala yang dialami biasanya soal pencatatan yang kadang berantakan, soalnya disini masih menggunakan formulis pendaftaran dan sering tercecer.
- Penulis : Apakah ada masalah ketika proses *check-out* dilakukan?
- Receptionist* : Biasanya sih kalau formulir tercecer ya jadi masalah besar. Kalo pengecekan kamar sih tinggal suruh orang *housekeeping* buat memeriksa terus hubungi kami disini buat konfirmasi.
- Penulis : Bagaimana pencatatan pengunjung dilakukan?
- Receptionist* : Kita catat dulu di formulir, lalu catat di *microsoft excel* buat *backup*. Nanti *sheet* nya kita kirim ke *manager* dalam kurun waktu tertentu.
- Penulis : Apakah ada kendala pada proses pencatatan?
- Receptionist* : Kendalanya ada pada formulir yang tercecer. Kalo yang lain biasanya kesulitan
- Penulis : Apakah ada kendala lain?
- Receptionist* : Tidak ada.

Wawancara: *Manager*

Berikut adalah wawancara yang dilakukan dengan *manager*:

- Penulis : Bagaimana cara melakukan perekapan laporan kepada atasan?
- Manager* : Kita ada prosedur sendiri di perusahaan untuk mengirimkan laporan dari anak perusahaan ke pusat dan tidak bisa diubah.
- Penulis : Apakah ada kendala dalam proses perekapan?

- Manager* : Masalah muncul kalau saja ada kesalahan dari pihak *receptionist*. Kalo dari bawahnya salah, keataspun jadi salah semua.
- Penulis : Apa peran pemilik perusahaan untuk dalam proses bisnis sehari-hari?
- Manager* : Pemilik tidak ikut campur dalam proses bisnis sehari-hari. Pemilik cuma menerika laporan perusahaan dari saya.
- Penulis : Apakah ada masalah dalam proses pemantauan sehari-hari?
- Manager* : Biasanya saya sering tidak di kantor karena harus mengurus permasalahan lain karena kebetulan saya pegang banyak lini bisnis jadinya tidak bisa memantau transaksi yang terjadi. Palingan tunggu rekapan harian dari *receptionist*.
- Penulis : Apa itu saja kendalanya?
- Manager* : Iya, itu saja.

Wawancara: Housekeeping

Berikut adalah wawancara yang dilakukan dengan *housekeeping*:

- Penulis : Bagaimana proses pembersihan ruangan kamar?
- Housekeeping* : Kita ada 2 jenis pemberishan, pembersihan rutin tiap pagi dan pembersihan kalau ada telepon dari tamu dikamar. Kalau kamar kosong biasanya cuma di bersikan seperlunya.
- Penulis : Bagaimana proses pembersihan saat sebelum pengunjung melakukan *check-out*?
- Housekeeping* : Kita biasanya langsung pergi ke kamar buat memeriksa terlebih dahulu. Jika tidak ada kerusakan tinggal telepon kebawah (*receptionist*) buat mengkonfirmasi. Kebetulan juga kan ruangan kita tepat disebelah meja *receptionist* jadinya cuma tinggal bilang aja.
- Penulis : Apakah ada kendala dalam prosesnya?
- Housekeeping* : Tidak ada.
- Penulis : Apakah ada kendala lain dalam proses pembersihan kamar?
- Receptionist* : Selain kekurangan personel ketika ramai saya rasa tidak ada.