

**MEET & EAT**

**DOSEN PENGAMPU :**

**Helen Sasty Pratiwi S.T., M.Eng.**

**Nama Kelompok ABCD :**

<b>Rino Ramadhan</b>	<b>H1101171017</b>
<b>Willson</b>	<b>H1101171019</b>
<b>Elsye Amanda Gloria</b>	<b>H1101171038</b>
<b>Delpia Amanda</b>	<b>H1101171047</b>



**UNIVERSITAS TANJUNGPURA**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**

**SISTEM INFORMASI**

**PONTIANAK**

**2019**

## **KATA PENGANTAR**

Puji syukur kami panjatkan kepada Tuhan Yang Maha Esa karena atas rahmat dan segala karunia-Nya, kami dapat menyelesaikan tugas akhir yang berjudul “Meet & Eat”. Tujuan dari penyusunan makalah ini adalah untuk memenuhi tugas akhir kelompok mata kuliah Teknologi Bergerak. Kami sangat bersyukur karena tugas ini dapat diselesaikan dengan tepat waktu.

Pada kesempatan ini, kami juga ingin mengucapkan terima kasih kepada pihak-pihak yang telah turut serta berkontribusi secara langsung maupun tidak langsung dalam pengerjaan tugas akhir ini. Kami ingin mengucapkan terima kasih kepada Helen Sasty Pratiwi S.T., M.Eng selaku dosen pengampu mata kuliah Teknologi Bergerak atas segala bimbingannya yang sangat berharga sehingga kami dapat menyelesaikan tugas ini.

Semoga tugas ini dapat memberikan manfaat yang berharga ,terutama untuk Mahasiswa/i Program Studi Sistem Informasi Universitas Tanjungpura mengenai tahapan dalam membuat sebuah aplikasi android, baik dari tahap analisis, perancangan, hingga pengimplementasian hasil rancangan kedalam kode program. Kami menyadari bahwa pengerjaan tugas akhir ini masih jauh dari kata sempurna.Oleh karena itu, Kami selaku penulis sangat mengharapkan kritik dan saran yang membangun demi perbaikan dimasa yang akan datang.

Pontianak,19 Desember 2019

Penulis

## DAFTAR ISI

Halaman

<b>KATA PENGANTAR.....</b>	<b>i</b>
<b>DAFTAR ISI.....</b>	<b>ii</b>
<b>BAB I .....</b>	<b>1</b>
<b>PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Batasan Masalah.....	2
1.3 Tujuan .....	2
<b>BAB II .....</b>	<b>3</b>
<b>LANDASAN TEORI.....</b>	<b>3</b>
2.1 Android .....	3
2.2 Android Studio .....	5
2.3 Android Software Development Kit.....	5
2.4 Java.....	6
2.5 XML.....	7
2.6 JSON.....	8
2.7 Use Case Diagram.....	8
2.8 Use Case Narrative.....	9
<b>BAB III .....</b>	<b>10</b>
<b>METODE PENELITIAN DAN PERANCANGAN .....</b>	<b>10</b>
3.1 Metodologi Analisis .....	10
3.2 Analisis Kebutuhan .....	10

3.3 Use Case Diagram.....	12
3.4 Perancangan User Interface.....	36
<b>BAB IV .....</b>	<b>43</b>
<b>HASIL DAN PEMBAHASAN .....</b>	<b>43</b>
4.1 Hasil .....	43
4.2 Pembahasan.....	43
<b>BAB V .....</b>	<b>11</b>
<b>PENUTUP .....</b>	<b>56</b>
5.1 Simpulan .....	56
5.2 Saran.....	56

## **BAB 1**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

Di era globalisasi yang berkembang pesat saat ini telah mendorong manusia untuk memanfaatkan kemajuan teknologi dalam membantu aktifitas kehidupan sehari-hari. Platform digital kini telah bertindak sebagai enabler pada konsep sharing economy dengan membantu mempertemukan berbagai pihak sesuai dengan kebutuhannya serta menimbulkan efisiensi. Hal tersebut dapat berpotensi menciptakan pemerataan baru, khususnya dalam bidang ekonomi.

Meet&Eat merupakan salah satu platform yang akan mewujudkan tujuan tersebut, yakni dengan mempertemukan supply dan demandnya secara efektif dan efisien melalui konsep sharing makanan. Melalui platform Meet&Eat, orang-orang yang memiliki hobi memasak akan dipertemukan dengan pihak yang ingin menikmati hidangan rumahan dengan harga yang terjangkau. Dengan menggunakan Meet&Eat, para host dapat memonetisasi hobinya serta mencoba berbagai resep baru dengan mengundang orang lain untuk menikmati hidangan makanan yang telah disajikan dirumahnya.

Saat ini, jumlah pengguna internet di Indonesia telah mencapai lebih dari 171 juta jiwa. Dari jumlah tersebut rupanya pengguna internet di Indonesia paling banyak menggunakan internet melalui smartphone. Dalam hal ini, Meet&Eat yang merupakan salah satu mobile app yang ingin dikembangkan dapat dengan efektif melakukan mobile marketing seperti melalui berbagai sosial media seperti facebook dan instagram yang dengan rutin menyebarkan konten yang menarik agar layanan yang ditawarkan pada aplikasi ini dapat dikenal oleh khalayak ramai.

## **1.2 Batasan Masalah**

Batasan masalah dalam perancangan Meet & Eat adalah:

1. Membuat rancangan awal Meet & Eat.
2. User yang terlibat antara lain : Guest dan Host

## **1.4 Tujuan**

Meet&Eat menawarkan win-win solution bagi users baik itu host maupun guest. Tidak hanya menawarkan keuntungan finansial, namun memiliki daya tarik tersendiri bahwa aplikasi ini dapat membantu dalam menemukan suasana makan bersama yang tidak didapat dari rumah makan biasa.

## **BAB 2**

### **LANDASAN TEORI**

#### **2.1 Android**

Menurut Lee (2012, pp. 2), Android adalah sistem operasi mobile yang didasari oleh Linux yang telah dimodifikasi. Awalnya Android dikembangkan oleh perusahaan startup yang bernama Android, Inc. Pada tahun 2005, sebagai awal memasuki dunia mobile, Google membeli Android dan mengambil alih pengembangan. Google menginginkan Android menjadi terbuka dan bebas. Pengembangan Android yang sederhana membuat Android sangat interaktif. Untuk mengembangkan Android dibentuklah Open Handset Alliance yang terdiri dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, TMobile, dan Nvidia.

Terdapat dua jenis distributor sistem operasi Android. Pertama, Google Mail Service yang mendapat dukungan penuh dari Google. Kemudian yang kedua adalah Open Handset Distribution (OHD) yang distribusinya tanpa dukungan langsung Google. Keuntungan dari Android adalah memberikan kemudahan pada pengembangan Android. Developer hanya perlu mengembangkan aplikasi pada platform Android, dan aplikasi tersebut akan dapat berjalan pada berbagai perangkat, yang tentunya mempunyai sistem operasi Android. Selain itu Android merupakan platform yang lengkap, baik dari sistem operasi, aplikasi serta tools untuk mengembangkan aplikasi.

##### **2.1.1 Fitur Pada Android**

Android merupakan sistem operasi open source dan bebas untuk di-customize. Oleh karena itu, tidak ada konfigurasi standar untuk software dan hardware pada perangkat berbasis Android, namun, Android mendukung beberapa fitur dasar, yaitu:

- a. Storage – menggunakan SQLite, karena SQLite merupakan database yang ringan untuk penyimpanan data.

- b. Connectivity – mendukung GSM/EDGE, IDEN, CDMA, EVDO, UMTS, Bluetooth, WiFi, LTE, WiMAX.
- c. Messaging – mendukung SMS dan MMS
- d. Web Browser - berdasarkan pada open source webkit, bersama dengan chrome's v8 Javascript engine.
- e. Media Support - mendukung beberapa media seperti AAC, HE-AAC, MPEG-4 SP, AMR, AMR-WB, WAV, MP3, JPG, PNG
- f. Hardware Support - accelerometer sensor, kamera, digital kompas, proximity sensor, GPS
- g. Multi-touch - mendukung multi-touch screens
- h. Multi-tasking – mendukung aplikasi multi-tasking
- i. Flash support – Android 2.3 mendukung flash 10.1
- j. Tethering - mendukung untuk sharing internet.

#### 2.1.2 Versi Android

Dari awal dimunculkan Android terdapat perubahan yang terus menerus dilakukan. Hingga saat ini versi Android terbaru bernama Marshmallow.

Tabel 2.1 Tabel Versi Android

(Sumber : [www.Android.com](http://www.Android.com))

Versi	Nama
Android 6.0	Marshmallow
Android 5.0	Lollipop
Andorid 4.4	KitKat
Android 4.1	Jelly Bean
Android 4.0	Ice Cream Sandwich
Android 3.0	Honeycomb



## **2.2 Android Studio**

Android Studio adalah Integrated Development Environment (IDE) untuk mengembangkan aplikasi Android. Android Studio berbasis pada “IntelliJ IDEA” Java-IDE dari JetBrains dan diperkenalkan oleh Google. Android Studio ini diumumkan pada Mei 2013. (Hohensee, 2014, pp. 1) Android Studio direncanakan akan menggantikan Eclipse sebagai IDE resmi untuk mengembangkan aplikasi pada platform Android. Android Studio memiliki beberapa fitur baru dibandingkan dengan Eclipse, diantaranya adalah :

- a. Menggunakan Gradle-based build sistem yang fleksibel.
- b. Bisa melakukan build pada beberapa APK
- c. Layout editor yang lebih bagus.
- d. Built-in support untuk Google Cloud Platform, sehingga mudah untuk integrasi dengan Google Cloud Messaging dan App Engine.
- e. Import library langsung dari Maven repository.

## **2.3 Android Software Development Kit ( SDK)**

Android SDK menurut Safaat (2012, pp. 5) merupakan tools API yang digunakan dalam mengembangkan aplikasi pada platform Android yang menggunakan bahasa pemrograman Java. Menurut Meier (2012, pp. 14), Android SDK sudah termasuk tools yang dibutuhkan dalam mengembangkan aplikasi, melakukan pengujian, dan melakukan debug seperti :

- a. Android API

Android SDK merupakan inti dari Android API library yang menyediakan para pengembang untuk mengakses kumpulan library pada Android. Libraries yang digunakan oleh Google merupakan libraries yang sama yang digunakan untuk membuat native application.

- b. Development Tools

Pada SDK sudah termasuk beberapa tools yang digunakan untuk melakukan compile dan debug sehingga aplikasi dapat dijalankan pada perangkat Android.

c. Android Virtual Device Manager dan Emulator

Pada SDK sudah termasuk emulator untuk mensimulasikan aplikasi yang dikembangkan sebelum di-install pada perangkat.

d. Full Documentation

Android SDK menyediakan informasi dari setiap package dan cara menggunakannya dalam bentuk dokumentasi di dalam source code. Selain itu, terdapat juga Android reference documentation dan developer guide yang menjelaskan cara memulai dalam mengembangkan aplikasi Android, menyertakan best practices, dan menyediakan pembahasan framework secara mendalam.

e. Sample Code

Android SDK menyediakan sampel program yang menunjukkan beberapa fungsi yang bisa dilakukan pada Android dan cara menggunakan fitur pada API.

f. Online Support

Android memiliki komunitas untuk para pengembang aplikasi. Google Groups merupakan salah satu forum dari kumpulan para pengembang aplikasi Android. Selain itu, Stack Overflow merupakan tempat di mana para pengembang aplikasi mendapatkan jawaban khususnya bagi programmer pemula.

## **2.4 Java**

Menurut Liang (2013, pp. 10), Java merupakan bahasa pemrograman yang dikembangkan oleh James Gosling berserta tim di Sun Microsystem pada tahun 1991. Awalnya Java disebut dengan “Oak”. Namun pada tahun 1995, nama “Oak” diganti menjadi Java. Bahasa pemrograman Java dirancang untuk menjadi bahasa pemrograman multi-platform yang cukup aman dan tangguh. Java memiliki beberapa karakteristik yaitu simpel, berorientasi object, high performance, multithreaded, dinamis, interpreted, serta portabel.

Pada mulanya Java biasa digunakan untuk mengembangkan aplikasi berbasis desktop yang lebih dikenal dengan nama J2SE. Kemudian muncul dua versi berikutnya yaitu J2EE yang diarahkan untuk mengembangkan aplikasi skala besar, aplikasi berbasis network, dan aplikasi berbasis web; dan J2ME yang diarahkan untuk mengembangkan aplikasi pada device yang kecil dan terbatas memorinya, misalnya pada perangkat Blackberry dan perangkat berbasis Symbian.

Pada tahun 2007, J2ME dijadikan basis untuk mengembangkan system operasi Android oleh Google. Pada tahun 2010, perusahaan Oracle membeli Sun Microsystem dan menjadi perusahaan penyedia layanan untuk Bahasa pemrograman Java. Dalam awal masa kepemilikannya, Oracle mengajukan tuntutan pada Google atas penggunaan teknologi J2ME pada platform Android, namun mereka dinyatakan kalah di pengadilan. Pada tahun 2011, Oracle merilis Java 7 yang perkembangannya sangat signifikan dari Java versi sebelumnya yaitu Java 6.

## **2.5 XML**

XML merupakan singkatan dari eXtensible Markup Language. Teknologi XML merupakan turunan dari SGML yang telah dikembangkan pada awal 80-an dan telah banyak digunakan pada dokumentasi teknis proyek berskala besar. XML dikembangkan mulai tahun 1996 dan dua tahun kemudian W3C mengumumkan versi terbaru sebagai suatu rekomendasi pada tanggal 10 Februari 1998. (Burd, 2014, pp. 17)

Bahasa pemrograman Java digunakan untuk menentukan apa yang akan dilakukan aplikasi, sedangkan XML digunakan untuk mendeskripsikan apa yang seharusnya. Kegunaan XML pada Android adalah untuk mendiskripsikan aplikasi, maksudnya adalah untuk mendiskripsikan layout pada layar, selain itu dengan XML suatu aplikasi dapat diatur untuk mempunyai lebih dari satu jenis bahasa, dan kegunaan lainnya mengenai data. Kegunaan kedua adalah untuk mendiskripsikan aplikasi itu sendiri. Setiap aplikasi Android pasti mempunyai file AndroidManifest.xml, yang merupakan suatu dokumen untuk mendiskripsikan fitur pada aplikasi.

## **2.6 JSON**

Menurut Smith (2015, pp. 1), JSON atau Javascript Object Notation adalah format pertukaran data yang ringan. JSON merupakan bagian dari bahasa pemrograman Javascript. Walaupun JSON merupakan bagian dari Javascript, tetapi JSON merupakan suatu format untuk pertukaran data. Selain itu, JSON juga tidak bergantung pada suatu bahasa pemrograman. JSON dikembangkan oleh Douglas Crockford. Crockford mendokumentasikan JSON pada tahun 2001, dan berikutnya JSON dapat digunakan sebagai alternative pengganti XML. Format ini memiliki kelebihan diantaranya mudah dibaca dan ditulis oleh manusia, berbasis teks, ringan, dan mudah untuk pertukaran data antara server dan klien. JSON terdiri dari dua struktur yaitu kumpulan pasangan nama atau nilai dan daftar nilai yang terurutkan.

## **2.7 Use Case Diagram**

Menurut Whitten & Bentley (2007, pp. 246), Use case diagram adalah suatu alat yang digunakan dalam permodelan use case untuk mengidentifikasi dan mendeskripsikan fungsi dari sebuah sistem. Use case diagram menggambarkan siapa yang akan menggunakan system dan bagaimana cara user untuk dapat berinteraksi dengan system tersebut.

Elemen-elemen yang terdapat dalam use case diagram:

### **a. Use case**

Use case diwakili oleh bentuk elips dengan nama use case yang terletak didalam elips itu sendiri. Sebuah use case akan merepresentasikan tujuan dari sistem.

### **b. Actor**

Aktor diwakili oleh bentuk stick figure yang merepresentasikan pengguna yang akan berinteraksi dengan sistem. Aktor yang dimaksud tidak selalu mewakili manusia, namun dapat juga mewakili sebuah organisasi.

#### c. Use Case Relationship

Relationship diwakili oleh sebuah garis antara dua simbol pada use case diagram. Pengertian dari relationship dapat berbeda tergantung oleh garis yang digambarkan dan jenis simbol yang terhubung.

#### d. The Boundary

The boundary adalah daerah yang membatasi antara use case dengan aktor dan diwakili oleh bentuk persegi panjang. The boundary digunakan untuk menggambarkan cakupan dari sistem.

### **2.8 Use Case Narrative**

Menurut Whitten & Bentley (2007, pp. 257), Use Case Narrative adalah sebuah deskripsi dari kejadian bisnis dan bagaimana user akan berinteraksi dengan sistem untuk menyelesaikan sebuah tugas. Use case narrative memiliki beberapa bagian yaitu:

#### 1. Use case name

Merupakan nama yang merepresentasikan tujuan yang ingin dicapai dari use case tersebut.

#### 2. Actor

Aktor menyatakan stakeholder yang melakukan interaksi dengan sistem dan berperan penting dalam melaksanakan use case.

#### 3. Description

Merupakan penjelasan yang menggambarkan tujuan dari usecase dan aktivitasnya.

#### 4. Precondition

Merupakan suatu kondisi tertentu yang harus dicapai sebelum menjalankan suatu use case.

#### 5. Flow of events

Merupakan langkah-langkah yang dilakukan oleh actor dalam menjalankan use case.

## **BAB III**

### **METODE PENELITIAN DAN PERANCANGAN**

#### **3.1 Metodologi Analisis**

Analisis kebutuhan bertujuan untuk menentukan kebutuhan yang diperlukan pada sebuah aplikasi. Analisis kebutuhan juga berfungsi sebagai bagaimana cara kerja aplikasi yang dibangun dan tampilan antarmuka yang sesuai dengan aplikasi tersebut. Metode yang digunakan dalam membangun aplikasi ini adalah metode pengumpulan data, perancangan model bisnis, analisis pengguna dan fungsionalitas, kebutuhan input dan output, analisis antarmuka dan perancangan perangkat lunak.

#### **3.2 Analisis Kebutuhan**

Hasil analisis kebutuhan yang dibutuhkan dalam pembuatan aplikasi ini adalah sebagai berikut.

##### **3.2.1 Analisis Pengguna dan Fungsionalitas**

Dalam aplikasi ini akan terdapat dua pengguna yang dalam prakteknya. Pengguna pertama (Host) merupakan pemilik rumah yang memiliki hobi memasak dan ingin mendapatkan penghasilan tambahan dengan memanfaatkan hobi tersebut. Pengguna kedua (Guest) merupakan orang-orang yang tidak memiliki waktu maupun kompetensi memasak namun tetap ingin menikmati makanan rumahan serta berkenalan dengan orang-orang baru.

Pada aplikasi ini akan terdapat perbedaan cara mengakses bagi pengguna. Untuk Guest, media yang digunakan dalam mengakses aplikasi ini untuk memesan dapat menggunakan smartphone berbasis Android, begitu juga dengan Host menambahkan menu dengan smartphone Android, serta memanipulasi data menu dan harga yang dapat diakses melalui aplikasi Meet & Eat.

### 3.2.2 Perangkat Lunak yang Dibutuhkan

Perangkat lunak yang dibutuhkan dalam pembuatan aplikasi Meet & Eat adalah sebagai berikut:

#### a) Android Studio

Android studio merupakan IDE (Integrated Development Environment) official IntelliJ IDEA untuk membuat aplikasi android.

#### b) Java Development Kit (JDK)

JDK berfungsi untuk mengkompilasi kode – kode java menjadi aplikasi java.

### 3.2.3 Perangkat Keras yang Dibutuhkan

Perangkat keras juga dibutuhkan dalam pembuatan aplikasi pencarian lokasi bengkel dan pemesanan adalah sebagai berikut:

#### a) Laptop

Laptop digunakan untuk membuat aplikasi serta pengujian aplikasi.

#### b) Smartphone

Smartphone digunakan untuk melakukan uji coba aplikasi android.

#### c) Kabel Data

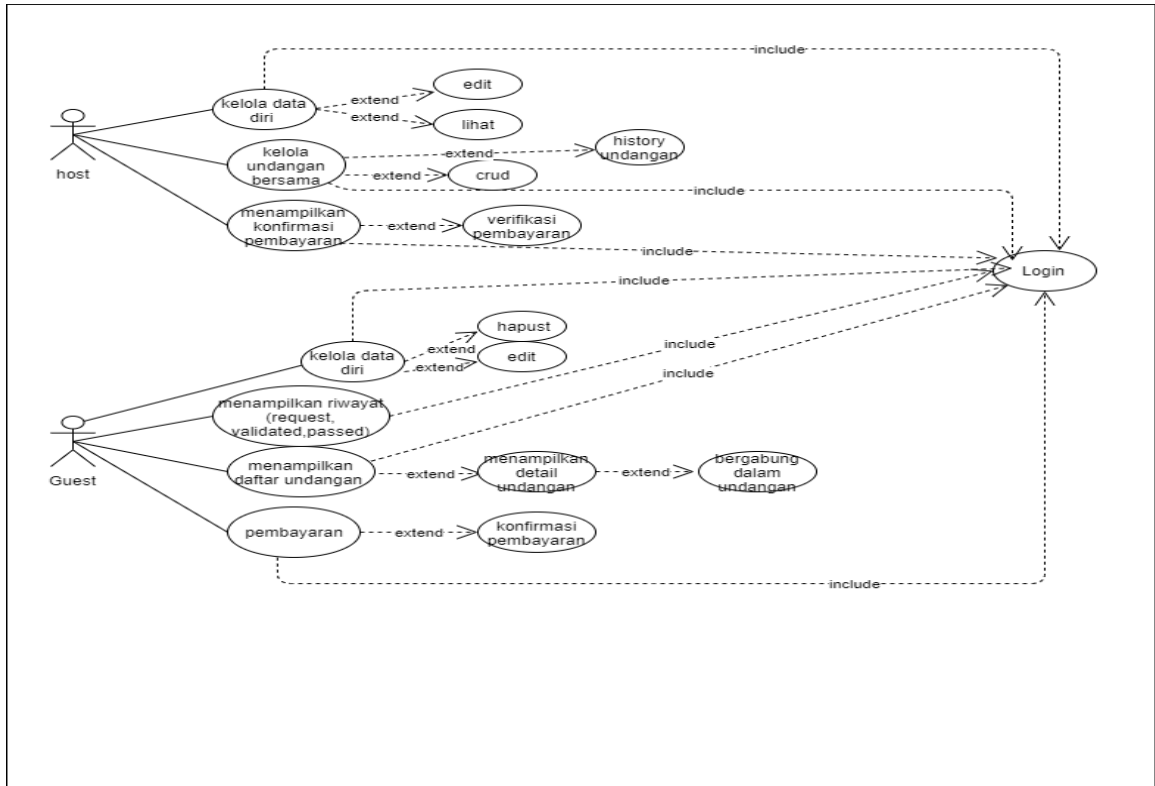
Kabel data digunakan untuk media penghubung smartphone dan Android Studio untuk melakukan debugging.

### 3.2.4 Analisis Kebutuhan Masukan (Input)

Dalam aplikasi ini terdapat beberapa input berdasarkan penggunaannya, yaitu guest yang dapat melakukan pemesanan, kemudian host yang menerima, serta mengelola undangan untuk guest agar dapat bergabung ke dalam undangan makan.

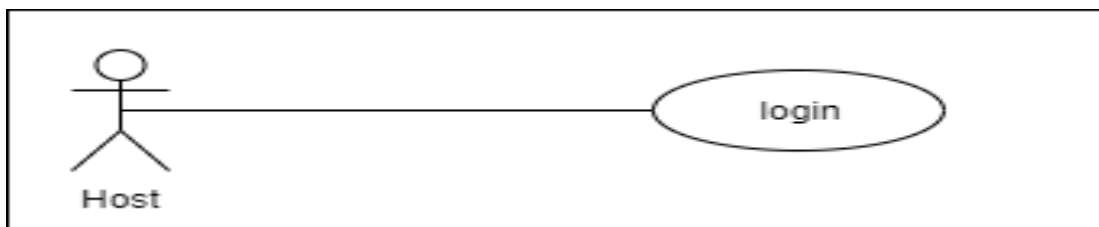
### 3.3 Use Case Diagram

#### 3.3.1 Usecase Umum



#### 3.3.2 Use case Spesification

##### 1. Login Host

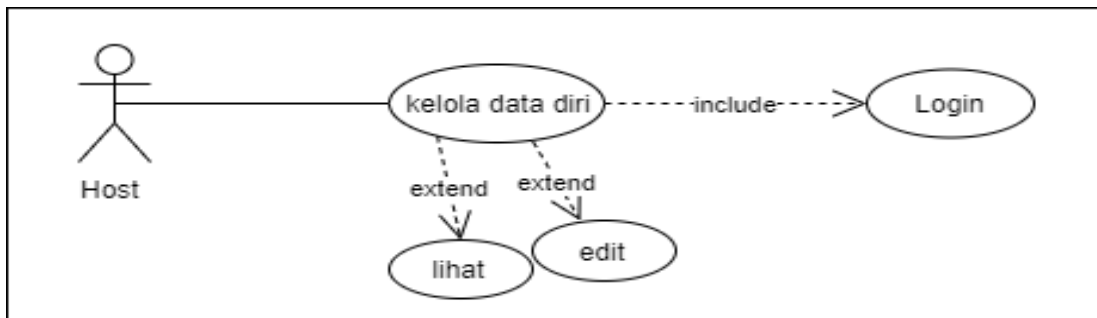


Nama Use Case	Login
---------------	-------



Aktor	Host
Aktor Pendukung	-
Deskripsi Singkat	Use case ini digunakan oleh Host untuk memperoleh akses masuk ke dalam sistem. Untuk dapat login ke dalam sistem maka host harus memasukkan username dan password yang berupa rangkaian karakter.
Kondisi Awal	Host belum masuk ke dalam sistem
Alur Dasar	<ol style="list-style-type: none"> <li>1. Use case ini dimulai ketika host memilih untuk melakukan login.</li> <li>2. Sistem menampilkan antarmuka untuk login.</li> <li>3. Host memasukkan username dan password.</li> <li>4. Sistem memeriksa username dan password yang diinputkan host <ol style="list-style-type: none"> <li>a. E-1 username dan password salah atau tidak sesuai.</li> </ol> </li> <li>5. Sistem memberikan akses ke host</li> <li>6. Use case ini selesai.</li> </ol>
Alur Alternatif	-
Alur Kesalahan	<ol style="list-style-type: none"> <li>1. E-1 username dan password salah atau tidak sesuai: <ol style="list-style-type: none"> <li>a. Sistem menampilkan peringatan bahwa username dan password salah atau tidak sesuai</li> <li>b. Kembali ke alur dasar langkah ke-2</li> </ol> </li> </ol>
Kondisi Sesudah	Host berhasil masuk kedalam sistem dan sistem menampilkan halaman utama

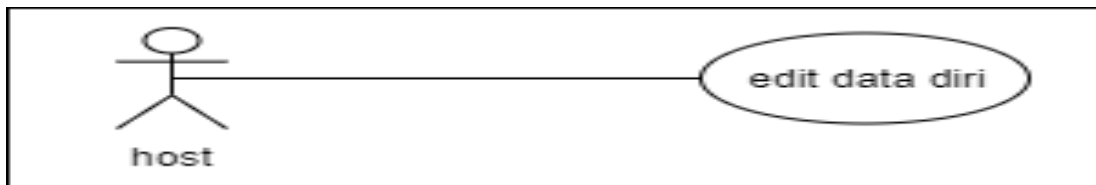
## 2. Kelola Data Diri



Nama Use Case	Kelola Data Diri
Aktor	Host
Aktor Pendukung	-
Deskripsi Singkat	Use case ini digunakan oleh host untuk mengelola data diri. Pada usecase ini host dapat melakukan fungsi lihat dan edit data diri
Kondisi Awal	Host sudah login kedalam sistem
Alur Dasar	<ol style="list-style-type: none"> <li>1. Use case ini dimulai ketika host memilih fungsi profile</li> <li>2. Sistem menampilkan halaman profile yang terdapat data diri host</li> <li>3. Host dapat melihat dan mengedit data diri</li> </ol>

	4. Usecase ini selesai
Alur Alternatif	-
Alur Kesalahan	-
Kondisi Sesudah	Host berhasil mengelola data diri

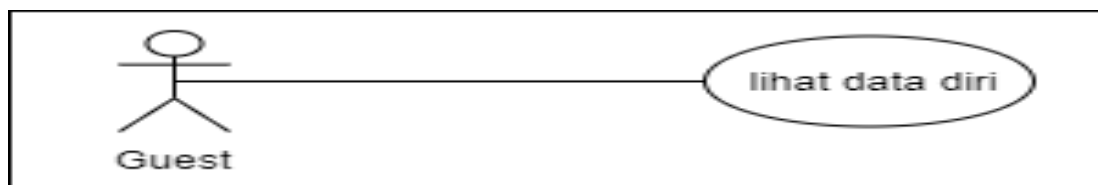
### 3. Edit data Diri



Nama Use Case	Edit data diri
Aktor	Guest
Aktor Pendukung	-
Deskripsi Singkat	Use case ini digunakan oleh guest untuk mengedit data diri

Kondisi Awal	<ol style="list-style-type: none"> <li>1. Guest sudah login kedalam sistem</li> <li>2. Guest sudah mengakses profile</li> </ol>
Alur Dasar	<ol style="list-style-type: none"> <li>1. Use case ini dimulai ketika guest memilih fungsi edit</li> <li>2. Sistem menampilkan halaman form untuk mengedit data diri</li> <li>3. Guest mengubah dan menyimpan data diri yang telah diubah</li> <li>4. Usecase ini selesai</li> </ol>
Alur Alternatif	-
Alur Kesalahan	-
Kondisi Sesudah	Guest berhasil melakukan edit data diri

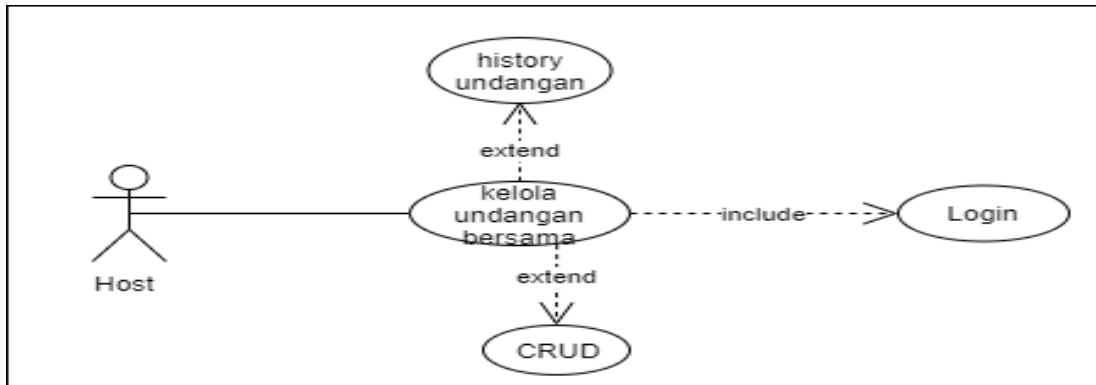
#### 4. Lihat Data Diri



Nama Use Case	Lihat Data Diri
---------------	-----------------

Aktor	Guest
Aktor Pendukung	-
Deskripsi Singkat	Use case ini digunakan oleh guest untuk melihat data diri.
Kondisi Awal	<ol style="list-style-type: none"> <li>1. Guest sudah login kedalam sistem</li> <li>2. Guest sudah mengakses profile</li> </ol>
Alur Dasar	<ol style="list-style-type: none"> <li>1. Use case ini dimulai ketika guest memilih fungsi lihat profile</li> <li>2. Sistem menampilkan halaman detail profile yang terdapat data diri guest</li> <li>3. Guest dapat melihat data diri</li> <li>4. Usecase ini selesai</li> </ol>
Alur Alternatif	-
Alur Kesalahan	-
Kondisi Sesudah	Guest berhasil melihat data diri

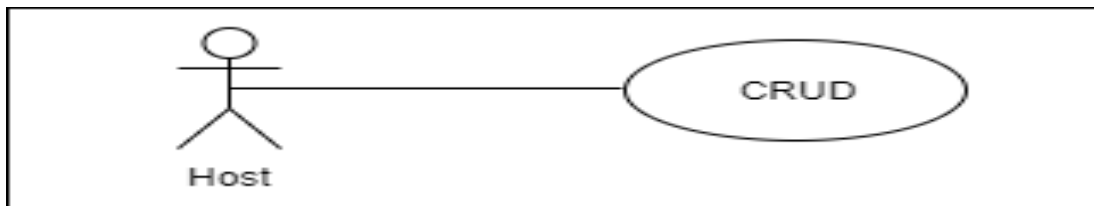
## 5. Kelola Undangan Bersama



Nama Use Case	Kelola Undangan Bersama
Aktor	Host
Aktor Pendukung	-
Deskripsi Singkat	Use case ini digunakan oleh host untuk mengelola undangan bersama. Pada usecase ini host dapat melakukan CRUD (create, read, update, delete) undangan serta dapat melihat history undangan yang sudah dibuat
Kondisi Awal	Host sudah login kedalam sistem
Alur Dasar	<ol style="list-style-type: none"> <li>1. Use case ini dimulai ketika host memilih fungsi kelola undangan bersama</li> <li>2. Sistem menampilkan halaman kelola undangan bersama</li> <li>3. Host dapat mengelola undangan bersama</li> </ol>

	4. Usecase ini selesai
Alur Alternatif	-
Alur Kesalahan	-
Kondisi Sesudah	Host berhasil mengelola undangan bersama

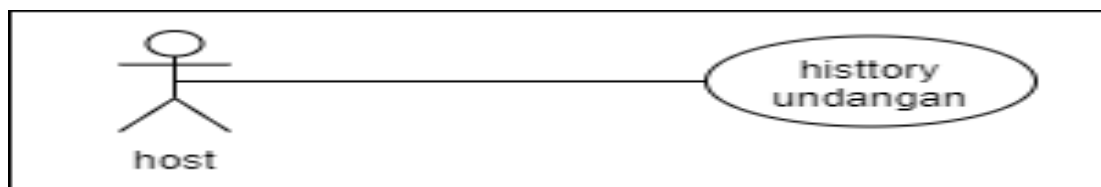
#### 6. CRUD Kelola Undangan Bersama



Nama Use Case	Kelola Undangan Bersama
Aktor	Host
Aktor Pendukung	-
Deskripsi Singkat	Use case ini digunakan oleh host dapat melakukan CRUD (create, read, update, delete) undangan

Kondisi Awal	<ol style="list-style-type: none"> <li>1. Host sudah login kedalam sistem</li> <li>2. Host sudah mengakses kelola undangan</li> </ol>
Alur Dasar	<ol style="list-style-type: none"> <li>1. Use case ini dimulai ketika host memilih fungsi CRUD</li> <li>2. Sistem menampilkan form untuk mengelola undangan</li> <li>3. Host dapat mengelola undangan bersama dengan melakukan CRUD</li> <li>4. Usecase ini selesai</li> </ol>
Alur Alternatif	-
Alur Kesalahan	-
Kondisi Sesudah	Host berhasil melakukan CRUD

## 7. History Undangan



Nama Use Case	History undangan
---------------	------------------



Aktor	Host
Aktor Pendukung	-
Deskripsi Singkat	Use case ini digunakan oleh host untuk melihat history undangan yang telah dibuat
Kondisi Awal	<ol style="list-style-type: none"> <li>1. Host sudah login kedalam sistem</li> <li>2. Host sudah membuat undangan</li> </ol>
Alur Dasar	<ol style="list-style-type: none"> <li>1. Use case ini dimulai ketika host memilih fungsi lihat history undangan</li> <li>2. Sistem menampilkan halaman history undangan</li> <li>3. Host dapat melihat history undangan</li> <li>4. Usecase ini selesai</li> </ol>
Alur Alternatif	-
Alur Kesalahan	-
Kondisi Sesudah	Host berhasil melihat history undangan

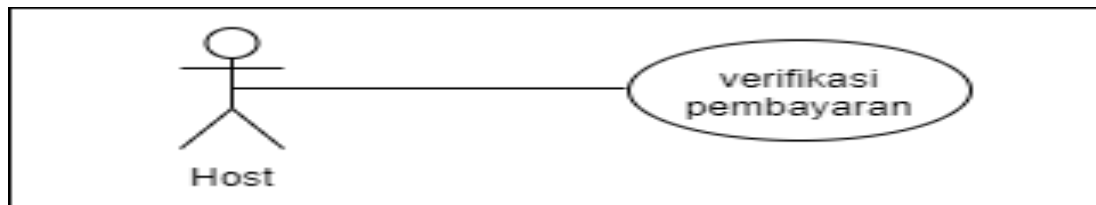
## 8. Konfirmasi Pembayaran



Nama Use Case	Konfirmasi pembayaran
Aktor	Host
Aktor Pendukung	-
Deskripsi Singkat	Use case ini digunakan oleh host untuk mengkonfirmasi pembayaran yang sudah dilakukan oleh guest
Kondisi Awal	Host sudah login kedalam sistem
Alur Dasar	<ol style="list-style-type: none"> <li>1. Use case ini dimulai ketika host memilih fungsi konfirmasi pembayaran</li> <li>2. Sistem menampilkan halaman berupa detail bayaran yang sudah dilakukan oleh guest</li> <li>3. Host dapat mengkonfirmasi pembayaran</li> <li>4. Usecase ini selesai</li> </ol>

Alur Alternatif	-
Alur Kesalahan	-
Kondisi Sesudah	Host berhasil melakukan konfirmasi pembayaran

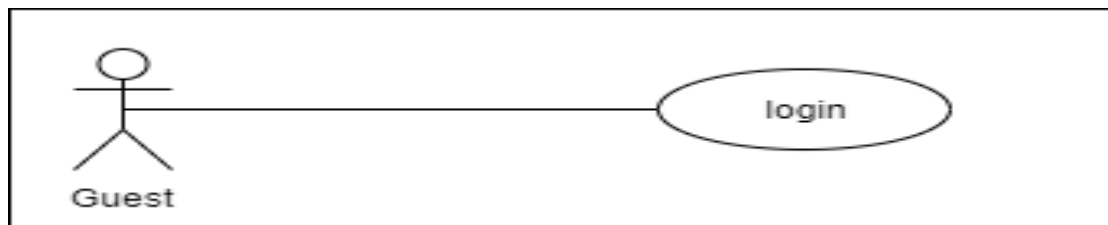
## 9. Verifikasi Pembayaran



Nama Use Case	Verifikasi pembayaran
Aktor	Host
Aktor Pendukung	-
Deskripsi Singkat	Use case ini digunakan oleh host memverifikasi pembayaran
Kondisi	Host sudah login kedalam sistem

Awal	
Alur Dasar	<ol style="list-style-type: none"> <li>1. Use case ini dimulai ketika host memilih fungsi verifikasi pembayaran</li> <li>2. Sistem menampilkan halaman verifikasi pembayaran</li> <li>3. Host dapat melakukan verifikasi pembayaran</li> <li>4. Usecase ini selesai</li> </ol>
Alur Alternatif	-
Alur Kesalahan	-
Kondisi Sesudah	Host berhasil memverifikasi undangan

#### 10. Login Guest

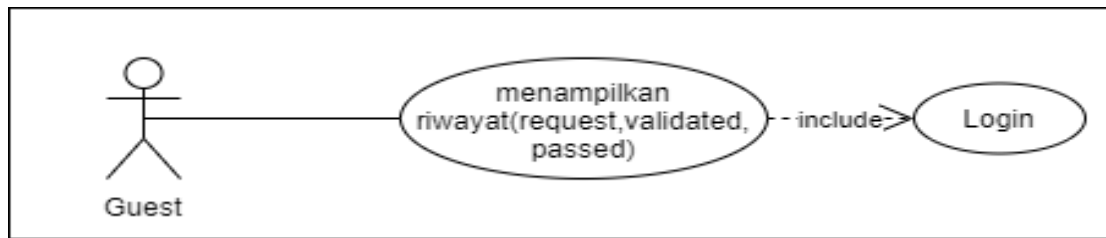


Nama Use Case	Login
---------------	-------

Aktor	Guest
Aktor Pendukung	-
Deskripsi Singkat	Use case ini digunakan oleh guest untuk memperoleh akses masuk ke dalam sistem. Untuk dapat login ke dalam sistem maka guest harus memasukkan username dan password yang berupa rangkaian karakter.
Kondisi Awal	Guest belum masuk ke dalam sistem
Alur Dasar	<p>7. Use case ini dimulai ketika guest memilih untuk melakukan login.</p> <p>8. Sistem menampilkan antarmuka untuk login.</p> <p>9. Guest memasukkan username dan password.</p> <p>10. Sistem memeriksa username dan password yang diinputkan host</p> <p>b. E-1 username dan password salah atau tidak sesuai.</p> <p>11. Sistem memberikan akses ke guest</p> <p>12. Use case ini selesai.</p>
Alur Alternatif	-
Alur Kesalahan	<p>2. E-1 username dan password salah atau tidak sesuai:</p> <p>c. Sistem menampilkan peringatan bahwa username dan password salah atau tidak sesuai</p> <p>d. Kembali ke alur dasar langkah ke-2</p>
Kondisi Sesudah	Guest berhasil masuk kedalam sistem dan sistem menampilkan halaman utama



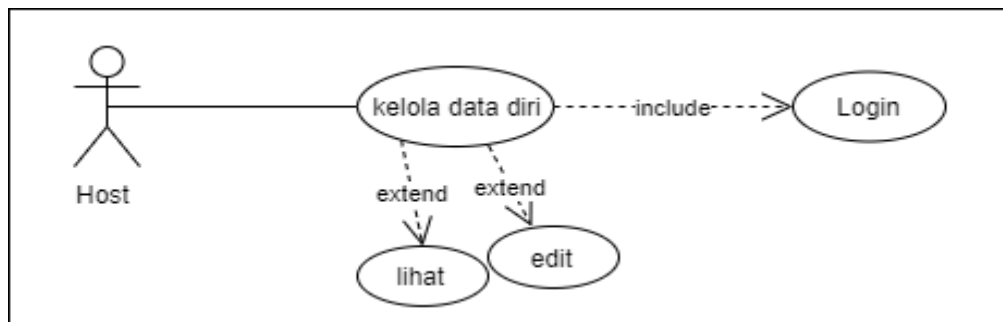
## 11. Riwayat Pemesanan



Nama Use Case	Riwayat Pemesanan
Aktor	Guest
Aktor Pendukung	-
Deskripsi Singkat	Use case ini digunakan oleh guest untuk melihat riwayat pemesana yang didalamnya terdapat request, validated, dan passed
Kondisi Awal	Guest sudah masuk ke dalam sistem
Alur Dasar	<ol style="list-style-type: none"><li>1. Use case ini dimulai ketika host memilih fungsi riwayat pemesanan</li><li>2. Sistem menampilkan halaman riwayat pemesanan</li><li>3. Guest dapat melihat riwayat pemesanan</li><li>4. Usecase ini selesai</li></ol>

Alur Alternatif	-
Alur Kesalahan	-
Kondisi Sesudah	Guest berhasil melihat riwayat pemesanan

## 12. Kelola Data Diri (Guest)

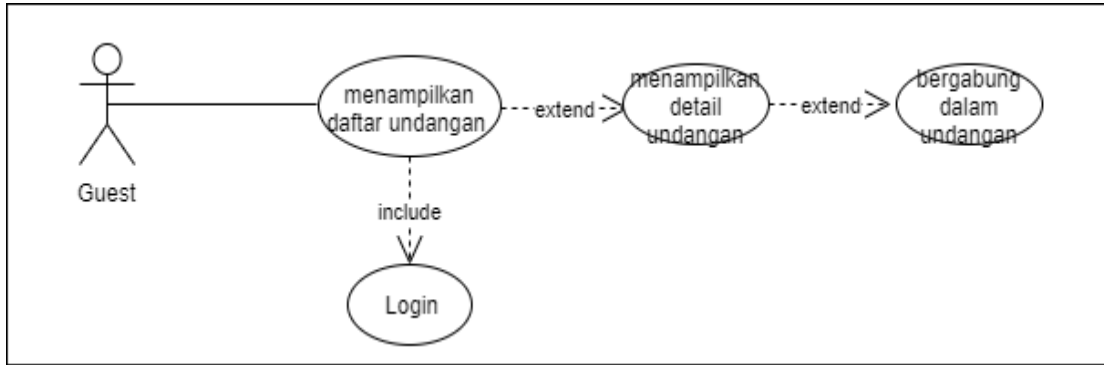


Nama Use Case	Kelola Data Diri
Aktor	Guest
Aktor Pendukung	-
Deskripsi	Use case ini digunakan oleh guest untuk mengelola data diri. Pada

Singkat	usecase ini guest dapat melakukan fungsi lihat dan edit data diri
Kondisi Awal	Guest sudah login kedalam sistem
Alur Dasar	<ol style="list-style-type: none"> <li>5. Use case ini dimulai ketika guest memilih fungsi profile</li> <li>6. Sistem menampilkan halaman profile yang terdapat data diri guest</li> <li>7. Guest dapat melihat dan mengedit data diri</li> <li>8. Usecase ini selesai</li> </ol>
Alur Alternatif	-
Alur Kesalahan	-
Kondisi Sesudah	Guest berhasil mengelola data diri



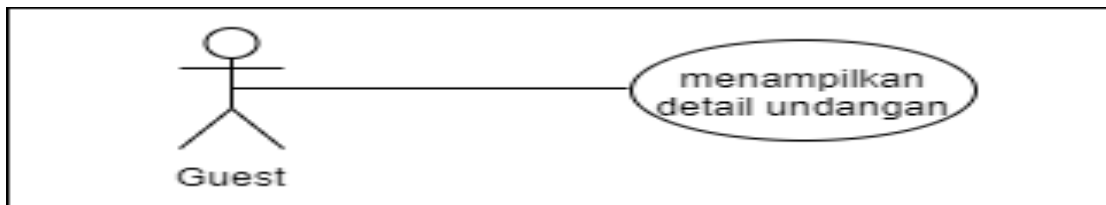
### 13. Daftar Undangan



Nama Use Case	Daftar Undangan
Aktor	Guest
Aktor Pendukung	-
Deskripsi Singkat	Use case ini digunakan oleh guest untuk melihat daftar undangan yang berisi list undangan yang tersedia
Kondisi Awal	Guest sudah masuk ke dalam sistem
Alur Dasar	<ol style="list-style-type: none"> <li>1. Use case ini dimulai ketika host memilih fungsi daftar undangan</li> <li>2. Sistem menampilkan halaman daftar undangan</li> <li>3. Guest dapat melihat daftar undangan</li> <li>4. Usecase ini selesai</li> </ol>

Alur Alternatif	-
Alur Kesalahan	-
Kondisi Sesudah	Guest berhasil melihat daftar undangan

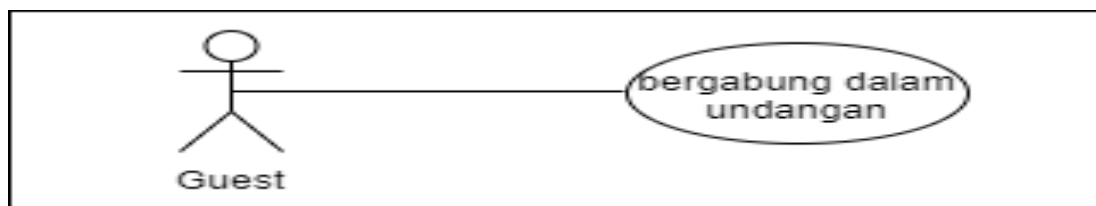
#### 14. Detail Undangan



Nama Use Case	Detail Undangan
Aktor	Guest
Aktor Pendukung	-
Deskripsi Singkat	Use case ini digunakan oleh guest untuk melihat undangan yang dipilih
Kondisi	1. Guest sudah masuk ke dalam sistem

Awal	2. Guest sudah mengakses daftar undangan
Alur Dasar	1. Use case ini dimulai ketika host memilih fungsi detail undangan 2. Sistem menampilkan halaman detail undangan 3. Guest dapat melihat detail undangan 4. Usecase ini selesai
Alur Alternatif	-
Alur Kesalahan	-
Kondisi Sesudah	Guest berhasil melihat detail undangan

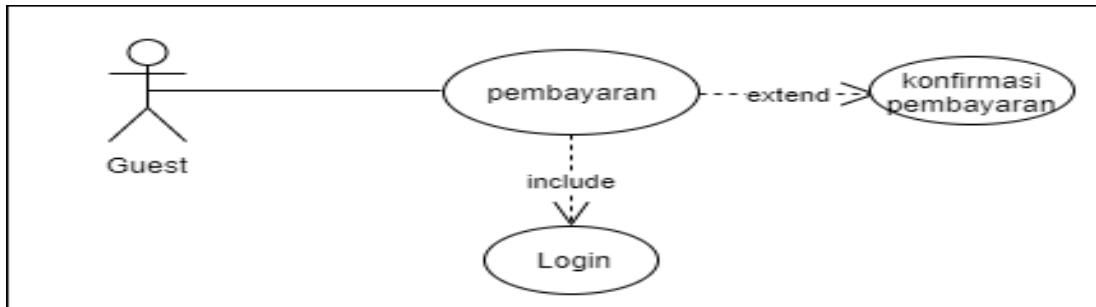
### 15. Bergabung Dalam Undangan



Nama Use Case	Bergabung dalam Undangan
Aktor	Guest

Aktor Pendukung	-
Deskripsi Singkat	Use case ini digunakan oleh guest untuk bergabung dalam undangan yang telah dipilih
Kondisi Awal	<ol style="list-style-type: none"> <li>1. Guest sudah masuk ke dalam sistem</li> <li>2. Guest sudah mengakses daftar undangan</li> <li>3. Guest sudah mengakses detail undangan</li> </ol>
Alur Dasar	<ol style="list-style-type: none"> <li>1. Use case ini dimulai ketika host memilih fungsi bergabung dalam undangan</li> <li>2. Sistem menampilkan halaman gabung dalam undangan</li> <li>3. Guest dapat melakukan bergabung dalam undangan</li> <li>4. Usecase ini selesai</li> </ol>
Alur Alternatif	-
Alur Kesalahan	-
Kondisi Sesudah	Guest berhasil berhasil bergabung dalam undangan

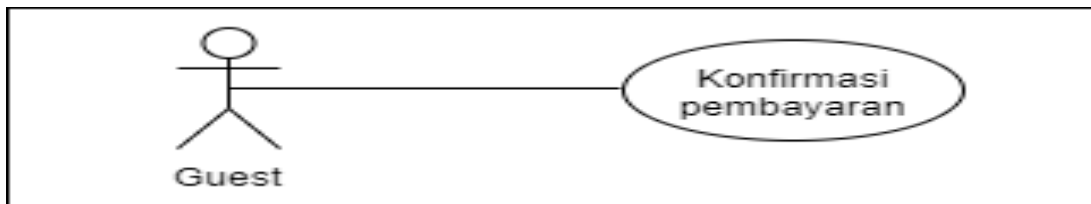
## 16. Pembayaran



Nama Use Case	Pembayaran
Aktor	Guest
Aktor Pendukung	-
Deskripsi Singkat	Use case ini digunakan oleh guest untuk melakukan pembayaran
Kondisi Awal	Guest sudah masuk ke dalam sistem
Alur Dasar	<ol style="list-style-type: none"> <li>1. Use case ini dimulai ketika host memilih fungsi pembayaran</li> <li>2. Sistem menampilkan halaman pembayaran</li> <li>3. Guest dapat melakukan pembayaran</li> <li>4. Usecase ini selesai</li> </ol>
Alur	-

Alternatif	
Alur	-
Kesalahan	
Kondisi Sesudah	Guest berhasil melakukan pembayaran

#### 17. Konfirmasi Pembayaran



Nama Use Case	Konfirmasi pembayaran
Aktor	Guest
Aktor Pendukung	-
Deskripsi Singkat	Use case ini digunakan oleh guest untuk melakukan konfirmasi pembayaran setelah mentransfer uang
Kondisi Awal	<ol style="list-style-type: none"> <li>1. Guest sudah masuk ke dalam sistem</li> <li>2. Guest sudah melakukan pembayaran</li> </ol>

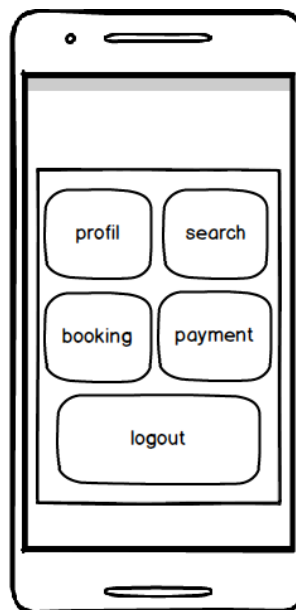
Alur Dasar	<ol style="list-style-type: none"> <li>1. Use case ini dimulai ketika host memilih fungsi konfirmasi pembayaran</li> <li>2. Sistem menampilkan halaman konfirmasi pembayaran</li> <li>3. Guest dapat melakukan konfirmasi pembayaran dengan mengupload bukti pembayaran</li> <li>4. Usecase ini selesai</li> </ol>
Alur Alternatif	-
Alur Kesalahan	-
Kondisi Sesudah	Guest berhasil melakukan konfirmasi pembayaran

### 3.4 Perancangan User Interface

#### 3.4.1 Guest

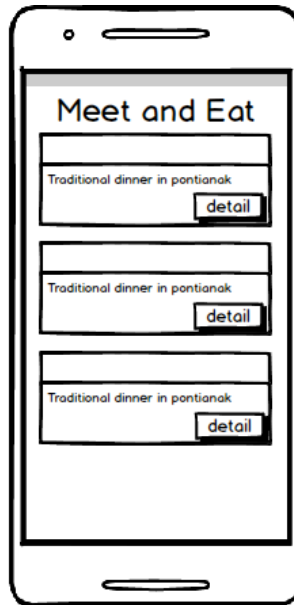


Gambar 3.4.1.1 Tampilan Login

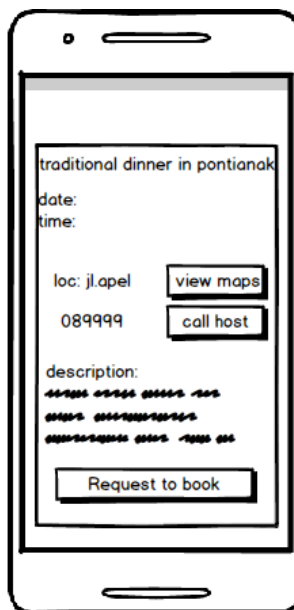


Gambar 3.4.1.2 Tampilan Dashboard guest

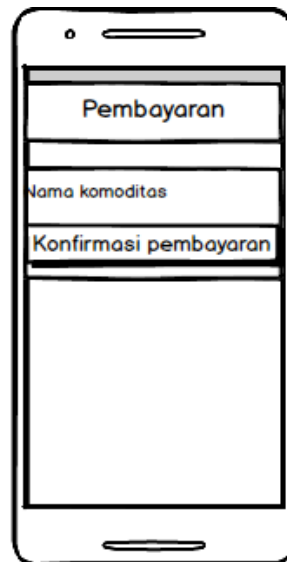




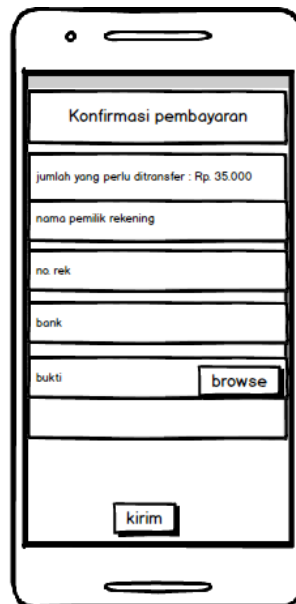
Gambar 3.4.1.3 Tampilan Menu Search



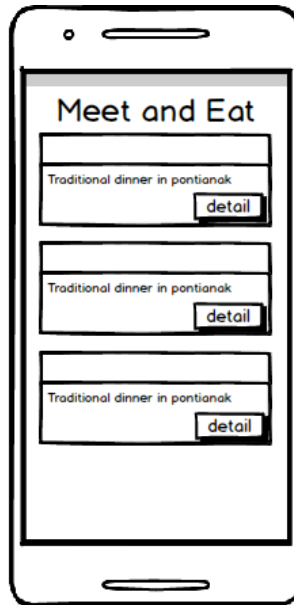
Gambar 3.4.1.4 Tampilan Detail Search



Gambar 3.4.1.5 Tampilan Menu Payment



Gambar 3.4.1.6 Tampilan Konfirmasi Pembayaran

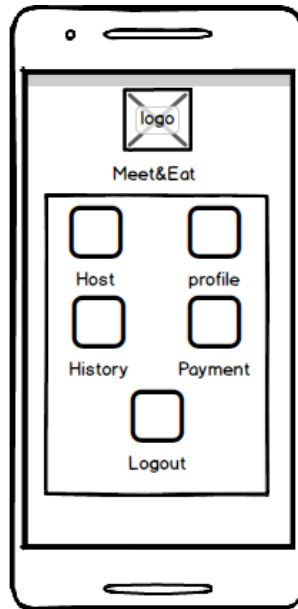


Gambar 3.4.1.7 Tampilan Menu Booking

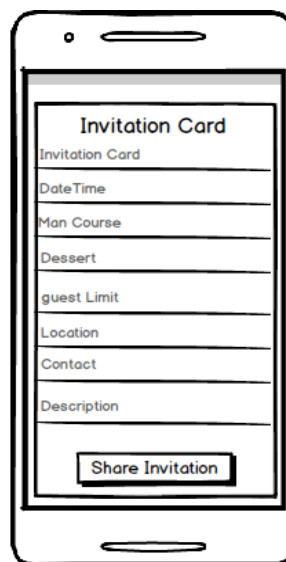
### 3.4.2 Host



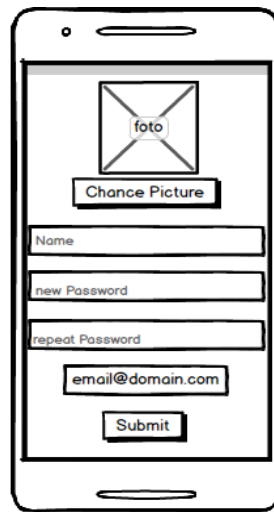
Gambar 3.4.2.1 Tampilan Login



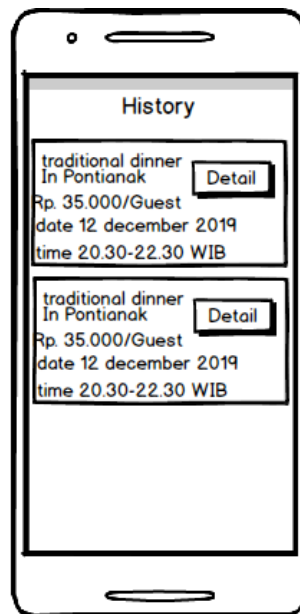
Gambar 3.4.2.2 Tampilan Menu Utama



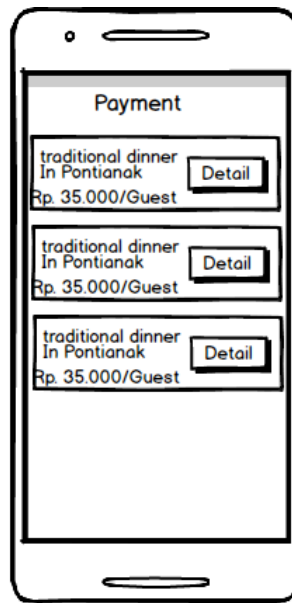
Gambar 3.4.2.3 Tampilan menu Host



Gambar 3.4.2.4 Tampilan Profil



Gambar 3.4.2.5 Tampilan menu History



Gambar 3.4.2.6 Tampilan menu Payment

## **BAB 4**

### **HASIL DAN PEMBAHASAN**

#### **4.1 Hasil**

Setelah melakukan analisis kebutuhan, perancangan, dan berakhir dengan pembuatan program yang sesungguhnya, maka hasil yang dicapai penulis adalah sebuah platform dengan nama “Meet & Eat”. Meet&Eat ingin mempertemukan demand dan supplynya secara langsung dengan mudah dan cepat dengan memanfaatkan teknologi. Melalui Meet&Eat, orang yang kecanduan memasak dan kerap menghabiskan pendapatannya untuk membeli beragam bahan masakan dan akibatnya banyak makanan yang terbuang sia-sia akan dipertemukan dengan mereka yang tidak memiliki waktu ataupun kemampuan memasak tetapi tetap ingin menyantap hidangan makanan rumahan. Melalui platform ini, para guest maupun host yang tidak saling mengenal sebelumnya akan dapat menjalin hubungan sosial satu sama lain. Meet&Eat dapat memiliki keunggulan kompetitif sebagai satu-satunya platform yang menawarkan layanan ini di Indonesia untuk saat ini.

#### **4.2 Pembahasan**

Host maupun guest dalam platform ini dapat dipertemukan melalui sebuah aplikasi berbasis mobile yang dapat di download langsung oleh pengguna. Melalui aplikasi berbasis mobile, pengguna dapat dengan mudah menggunakan platform ini melalui smartphone serta dengan visual yang lebih menarik. Para host dapat menggunakan aplikasi ini dengan melakukan registrasi terlebih dahulu. Setelah itu, para host dapat mengundang para guest untuk menikmati hidangan makanan di rumahnya dengan memasukkan informasi terkait nama, alamat, jenis makanan yang akan dihidangkan, jumlah guest yang akan diundang, beserta harga. Para guest yang ingin menggunakan aplikasi ini juga diharuskan untuk melakukan registrasi terlebih dahulu. Setelah itu, guest dapat melihat berbagai penawaran aktivitas ekonomi berbagai makanan yang dilakukan oleh para host.

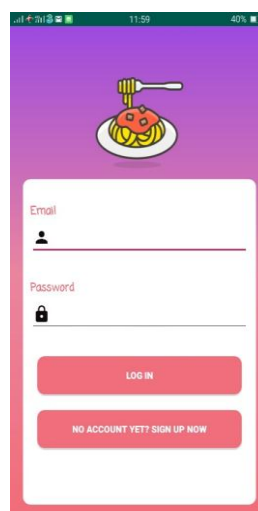
## 4.2.1 Tampilan Halaman Aplikasi Guest

### 4.2.1.1 Tampilan Registrasi

A mobile application registration screen with a purple gradient background. At the top, there is a white rounded rectangle containing a cartoon illustration of a plate of spaghetti with a fork. Below the illustration are four white input fields with labels: 'Fullname', 'Email', 'Password', and 'Rehype Password'. At the bottom of the white area are two red buttons: 'SIGN UP' and 'ALREADY HAVE AN ACCOUNT? LOGIN NOW'.

Gambar ini merupakan tampilan untuk melakukan registrasi menjadi Guest dalam sistem. Tanpa registrasi, seorang Guest tidak akan dapat melakukan login kedalam sistem. Registrasi dilakukan dengan memasukkan nama lengkap, email, password, dan konfirmasi password.

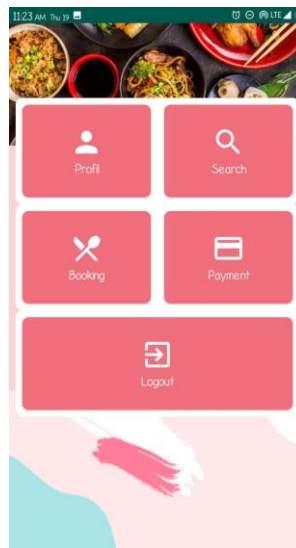
### 4.2.1.2 Tampilan Login

A mobile application login screen with a purple gradient background. At the top, there is a white rounded rectangle containing a cartoon illustration of a plate of spaghetti with a fork. Below the illustration are two white input fields with labels: 'Email' and 'Password'. Below the 'Password' field is a small black padlock icon. At the bottom of the white area are two red buttons: 'LOG IN' and 'NO ACCOUNT YET? SIGN UP NOW'.



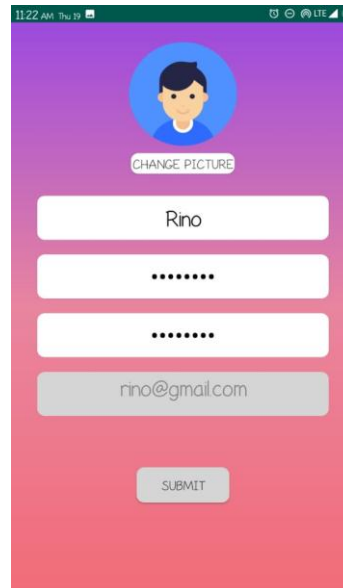
Gambar ini merupakan tampilan untuk melakukan login ke akun Guest. Login hanya bisa dilakukan apabila Guest telah melakukan registrasi akun. Guest harus memasukkan email dan kata sandi untuk dapat login kedalam sistem. Jika data yang dimasukkan tidak sesuai dengan form validation yang telah diterapkan, maka sistem akan menampilkan peringatan kesalahan. Jika login berhasil, maka sistem akan menampilkan halaman utama sistem.

#### 4.2.1.3 Tampilan Menu Utama



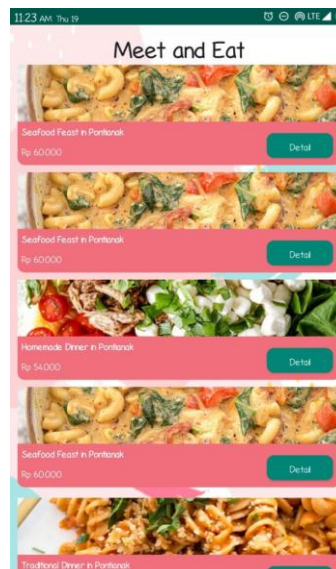
Gambar ini merupakan tampilan halaman utama dari Guest setelah berhasil melakukan login dalam sistem. Dalam halaman utama sistem terdapat menu profil untuk menampilkan dan mengubah data diri pengguna, search untuk menampilkan daftar undangan dari para host, booking untuk menampilkan daftar undangan yang telah di booking, payment untuk melakukan konfirmasi pembayaran, serta logout untuk keluar dari sistem.

#### 4.2.1.4 Tampilan Menu Profil



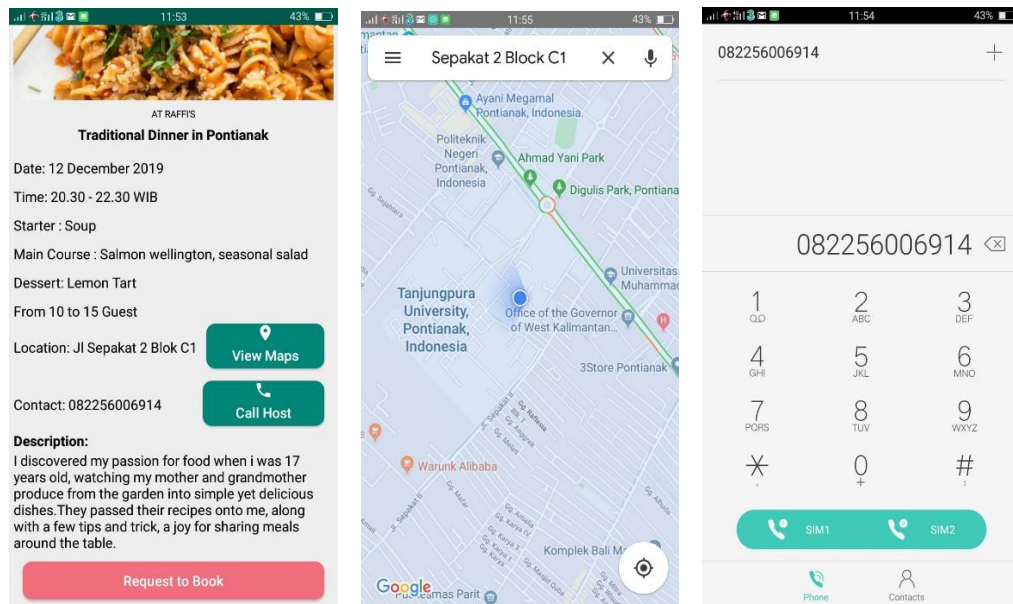
Gambar ini merupakan tampilan ketika Guest memilih menu profil pada halaman utama sistem. Pada halaman ini, Guest dapat melihat dan mengubah data dirinya seperti foto, nama, password, dan email.

#### 4.2.1.5 Tampilan Menu Search



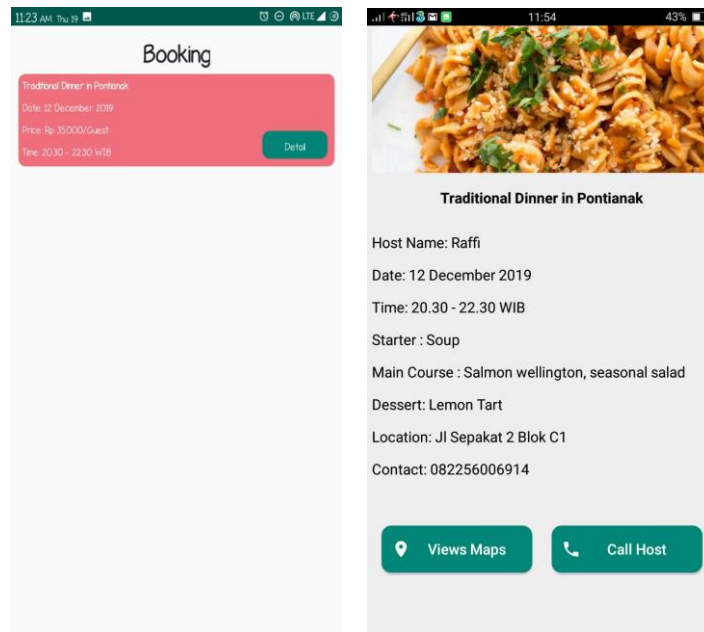
Gambar ini merupakan tampilan ketika Guest memilih menu search pada halaman utama sistem setelah berhasil melakukan login. Sistem akan menampilkan daftar undangan yang dapat dibooking oleh guest. Guest juga dapat melihat informasi detail dari undangan-undangan tersebut.

#### 4.2.1.6 Tampilan Menu Detail Undangan



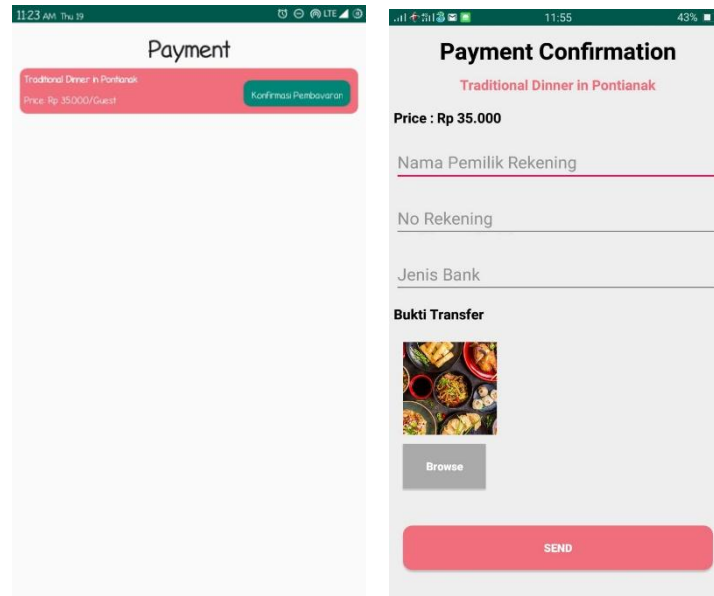
Gambar ini merupakan tampilan ketika Guest memilih menu detail pada salah satu undangan yang telah dipilih. Pada halaman ini akan terdapat berbagai informasi yang dapat dilihat oleh Guest terkait undangan seperti tanggal, waktu, hidangan (*starter, main course, dessert*), jumlah guest, lokasi, kontak, serta deskripsi yang ditulis oleh host. Pada tampilan ini juga dilengkapi Implicit Intent berfungsi untuk memanggil fungsi activity yang sudah ada di fungsi internal android seperti Dial Number, Open Browser dan lainnya. Dalam hal ini, Guest dapat menampilkan lokasi (*view maps*) dan nomor telepon(*call host*) dari host.

#### 4.2.1.7 Tampilan Menu Booking



Gambar ini merupakan tampilan ketika Guest memilih menu Booking pada halaman utama sistem. Sistem akan menampilkan daftar undangan dari guest yang telah dilakukan konfirmasi pembayaran serta telah diverifikasi oleh host. Guest juga dapat menampilkan informasi detail terkait undangan tersebut seperti nama host, tanggal, waktu, hidangan (*starter, main course, dessert*), lokasi, dan kontak dari host.

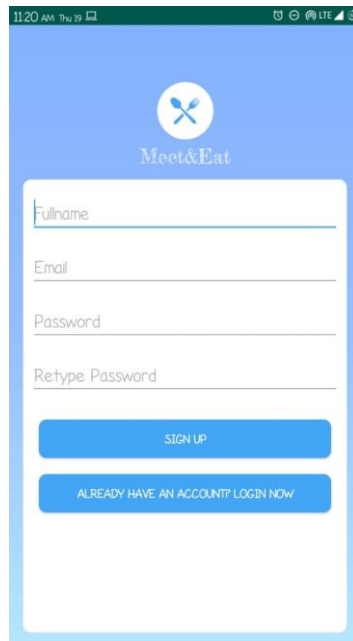
#### 4.2.1.8 Tampilan Menu Payment



Gambar ini merupakan tampilan ketika Guest memilih menu payment pada halaman utama sistem. Sistem akan menampilkan daftar undangan yang sebelumnya sudah diboooking namun belum dilakukan konfirmasi pembayaran. Guest dapat melakukan konfirmasi pembayaran dengan memasukkan nama pemilik rekening, nomor rekening, serta mengupload bukti transfer.

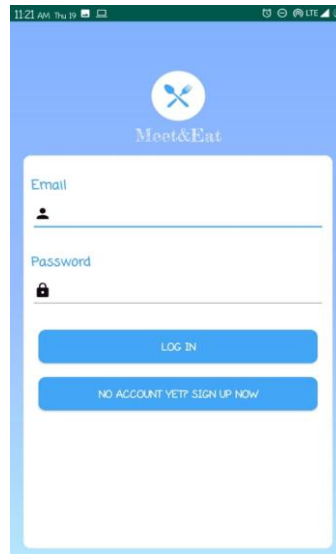
## 4.2.2 Tampilan Halaman Aplikasi Host

### 4.2.2.1 Tampilan Registrasi

The image shows a mobile application interface for registration. At the top, there is a status bar with the time 11:20 AM, the day Thursday, and various system icons. Below the status bar is a blue header with a white circular logo containing a crossed fork and knife, and the text "Meet&Eat" below it. The main content area is white and contains a registration form. The form has four input fields: "Fullname", "Email", "Password", and "Retype Password". Below the input fields are two blue buttons. The first button is labeled "SIGN UP" and the second button is labeled "ALREADY HAVE AN ACCOUNT? LOGIN NOW".

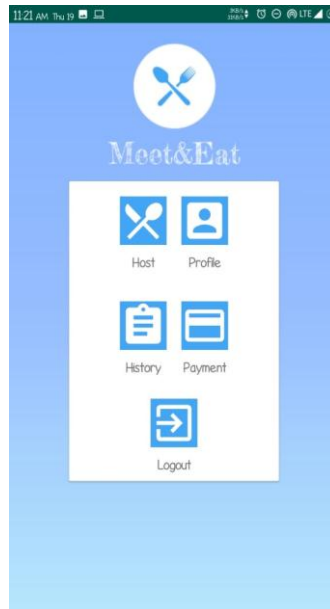
Gambar ini merupakan tampilan untuk melakukan registrasi menjadi Host dalam sistem. Tanpa registrasi, seorang Host tidak akan dapat melakukan login kedalam sistem. Registrasi dilakukan dengan memasukkan nama lengkap, email, password, dan konfirmasi password. Jika data yang dimasukkan calon host tidak sesuai dengan form validation yang telah diterapkan, maka sistem akan menampilkan peringatan kesalahan. Sebaliknya, apabila data yang diinputkan sudah benar, maka sistem akan menampilkan halaman login.

#### 4.2.2.2 Tampilan Login



Gambar ini merupakan tampilan untuk melakukan login ke akun Host. Login hanya bisa dilakukan apabila Host telah melakukan registrasi akun. Host harus memasukkan email dan kata sandi untuk dapat login kedalam sistem. Jika data yang dimasukkan tidak sesuai dengan form validation yang telah diterapkan, maka sistem akan menampilkan peringatan kesalahan. Jika login berhasil, maka sistem akan menampilkan halaman utama sistem.

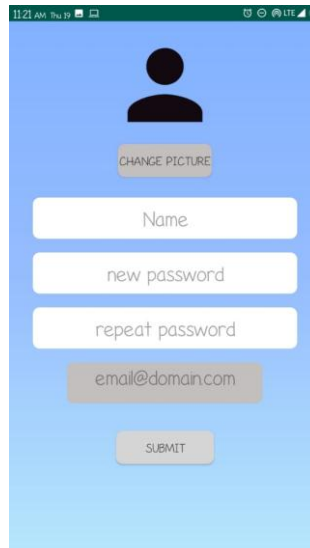
#### 4.2.2.3 Tampilan Menu Utama



Gambar ini merupakan tampilan halaman utama dari Host setelah berhasil melakukan login dalam sistem. Dalam halaman utama sistem terdapat menu profil untuk menampilkan dan mengubah data diri pengguna, host untuk menginputkan undangan, history untuk menampilkan daftar undangan yang telah diinputkan, payment untuk melakukan verifikasi pembayaran, serta logout untuk keluar dari sistem.

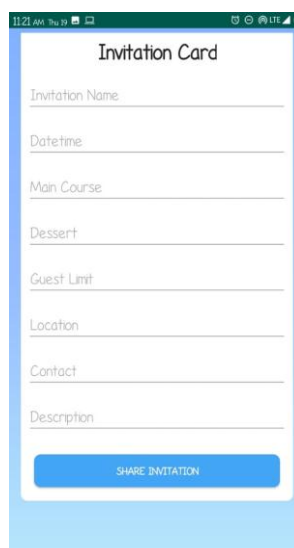


#### 4.2.4.4 Tampilan Menu Profil

A screenshot of a mobile application's profile menu. The background is a light blue gradient. At the top, there is a black silhouette of a person's head and shoulders. Below it is a button labeled "CHANGE PICTURE". Underneath are four input fields: "Name", "new password", "repeat password", and "email@domain.com". At the bottom is a button labeled "SUBMIT". The status bar at the top shows the time as 11:21 AM, the date as Thu, 19, and various system icons.

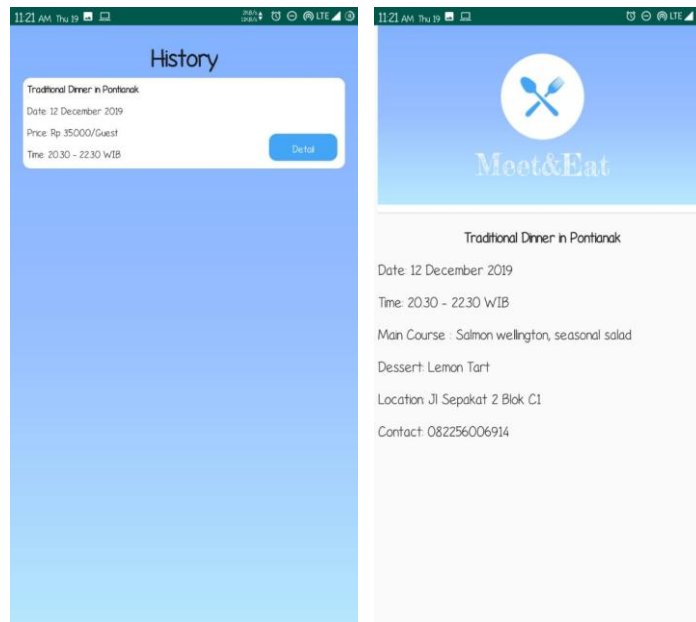
Gambar ini merupakan tampilan ketika Host memilih menu profil pada halaman utama sistem. Pada halaman ini, Host dapat melihat dan mengubah data dirinya seperti foto, nama, password, dan email. Segala perubahan yang dilakukan pada menu profil akan diproses oleh sistem setelah host menekan tombol submit.

#### 4.2.4.5 Tampilan Menu Host

A screenshot of a mobile application's invitation card form. The background is a light blue gradient. The form is titled "Invitation Card" and contains several input fields: "Invitation Name", "Datetime", "Main Course", "Dessert", "Guest Limit", "Location", "Contact", and "Description". At the bottom is a button labeled "SHARE INVITATION". The status bar at the top shows the time as 11:22 AM, the date as Thu, 19, and various system icons.

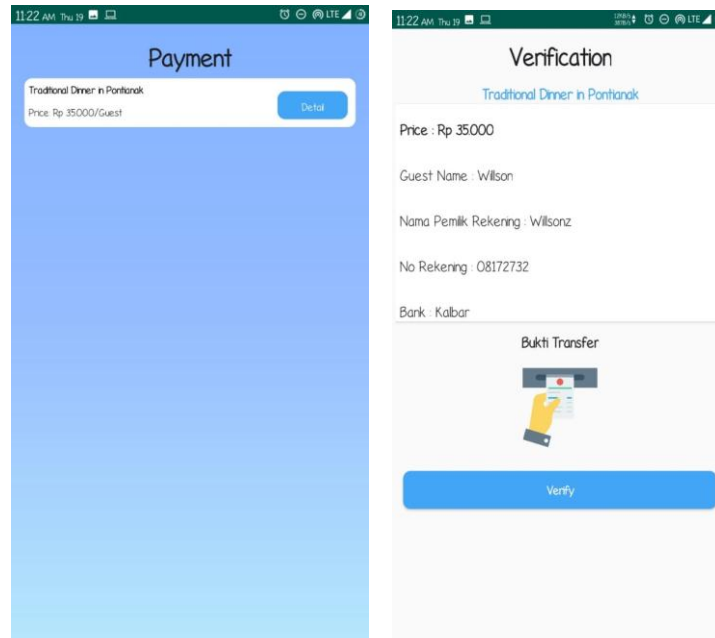
Gambar ini merupakan tampilan ketika host memilih menu host pada halaman utama sistem. Sistem akan menampilkan form untuk diisi oleh host yang meliputi nama undangan, waktu, tanggal, hidangan (*starter, main course, dessert*), jumlah guest, lokasi, kontak, dan deskripsi.

#### 4.2.4.6 Tampilan Menu History



Gambar ini merupakan tampilan ketika host memilih menu history pada halaman utama sistem. Pada halaman ini akan ditampilkan daftar undangan yang telah diinputkan oleh Host. Host juga dapat melihat informasi detail terkait undangan tersebut.

#### 4.2.4.6 Tampilan Menu Payment



Gambar ini merupakan tampilan ketika host memilih menu payment pada halaman utama sistem. Sistem akan menampilkan konfirmasi pembayaran yang telah diinputkan oleh Guest untuk selanjutnya diverifikasi oleh Host.

## **BAB 5**

### **PENUTUP**

#### **5.1 Kesimpulan**

Platform digital kini telah bertindak sebagai enabler pada konsep sharing economy dengan membantu mempertemukan berbagai pihak sesuai dengan kebutuhannya serta menimbulkan efisiensi. Hal tersebut dapat berpotensi menciptakan pemerataan baru, khususnya dalam bidang ekonomi. Meet&Eat merupakan salah satu platform yang akan mewujudkan tujuan tersebut, yakni dengan mempertemukan supply dan demandnya secara efektif dan efisien melalui konsep sharing makanan. Dengan menggunakan Meet&Eat, Para host dapat memonetisasi hobinya serta mencoba berbagai resep baru dengan mengundang orang lain untuk menikmati hidangan makanan yang telah disajikan dirumahnya.

#### **5.2 Saran**

Terdapat beberapa saran yang dapat dilakukan untuk pengembangan sistem selanjutnya, yaitu:

1. Sistem ini sebaiknya dilengkapi dengan fitur review serta pemberian rating kepada para host untuk meningkatkan kepercayaan(*trust*) dari guest terkait dengan undangan yang telah dipilih.
2. Sistem ini sebaiknya dilengkapi dengan fitur filter yang memungkinkan guest dapat lebih mudah mencari undangan sesuai dengan kriteria yang diinginkan seperti range harga, hidangan, lokasi, dan sebagainya.