

**PENERAPAN *USER CENTERED DESIGN* DALAM
PEMBANGUNAN APLIKASI INFORMASI HOSTEL BERBASIS
ANDROID**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Diah Shinta Dewi
NIM: 135150200111022



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018



PENGESAHAN

PENERAPAN *USER CENTERED DESIGN* DALAM PEMBANGUNAN APLIKASI INFORMASI
HOSTEL BERBASIS ANDROID

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Diah Shinta Dewi
NIM: 135150200111022

Skripsi ini telah diuji dan dinyatakan lulus pada
1 Agustus 2018
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I



Adam Hendra Brata, S.Kom., M.T., M.Sc.
NIK: 201607 900105 1 001

Dosen Pembimbing II



Lutfi Fanani, S.Kom., M.T., M.Sc.
NIK: 201607 890217 1 001

Mengetahui

Ketua Jurusan Teknik Informatika



Irfan Astoto Kurniawan, S.T, M.T, Ph.D
NIK: 19710518 200312 1 001

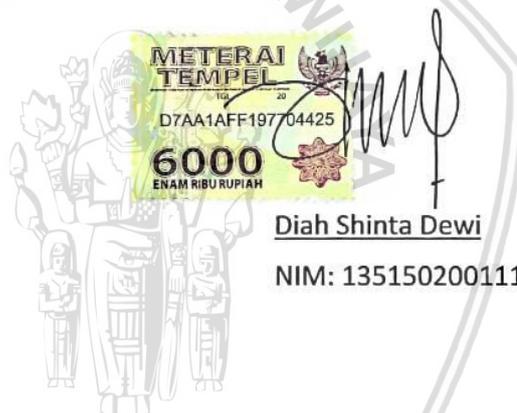


PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 6 Agustus 2018



Diah Shinta Dewi

NIM: 135150200111022

KATA PENGANTAR

Segala puji dan syukur kehadirat Allah SWT atas berkah, rahmat dan hidayah-Nya yang senantiasa dilimpahkan kepada penulis, sehingga bisa menyelesaikan skripsi dengan judul “PENERAPAN *USER CENTERED DESIGN* DALAM PEMBANGUNAN APLIKASI INFORMASI HOSTEL BERBASIS ANDROID” sebagai syarat untuk menyelesaikan Program Sarjana (S1) pada jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya Malang.

Dalam penyusunan skripsi ini banyak hambatan serta rintangan yang penulis hadapi namun pada akhirnya dapat menyelesaikannya dengan adanya bimbingan dan bantuan dari berbagai pihak secara moril dan materil. Untuk itu pada kesempatan ini penulis menyampaikan ucapan terimakasih kepada:

1. Kedua orang tua, papa Mohammad Zaini dan mama Dwi Isnanti yang telah memberikan dukungan baik moril dan materil serta doa yang tiada henti-hentinya kepada penulis.
2. Segenap keluarga yang telah memberikan doa dan semangat untuk menyelesaikan skripsi ini.
3. Bapak Wayan Firdaus Mahmudy, S.Si., M.T., Ph.D. selaku Dekan Fakultas Ilmu Komputer Universitas Brawijaya Malang.
4. Bapak Tri Astoto Kurniawan, S.T., M.T., Ph.D. selaku Ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya Malang.
5. Bapak Agus Wahyu Widodo, S.T., M.Sc. selaku Ketua Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya Malang.
6. Bapak Adam Hendra Brata, S.Kom., M.T., M.Sc. dan Bapak Lutfi Fanani, S.Kom., M.T., M.Sc. selaku Dosen Pembimbing yang selama ini telah banyak membantu dan membimbing penulis dalam proses penulisan skripsi ini.
7. Seluruh dosen Fakultas Ilmu Komputer Universitas Brawijaya yang telah memberikan ilmunya kepada penulis.
8. Saudara penulis Arie Krisnoanto, Lia Rosalina, Artiyan Prasetya dan Rusdja widya yang telah membantu tugas akhir penulis dalam bentuk apapun dan memberikan semangat agar cepat terselesaikan skripsi ini.
9. Ulul Albab Kholili atas bantuan dan dukungan yang diberikan setiap harinya kepada penulis saat bersama-sama menempuh skripsi untuk memperoleh gelar sarjana.

10. Rekan SABEB Entertainment dan rekan-rekan lainnya yang telah memberikan bantuan dan semangat kepada penulis.

11. Crew dari MAXX Corner Malang Town Square yang telah memberikan tempat yang nyaman dan minuman yang menyegarkan ketika penulis mengerjakan skripsi.

Rasa hormat dan terimakasih bagi semua pihak diatas atas segala dukungan dan doanya semoga Allah SWT membalas segala kebaikan yang telah diberikan kepada penulis.

Akhir kata, penulis mengharapkan skripsi ini dapat memberikan manfaat dan semoga Allah SWT memberi lindungan bagi kita semua.

Malang, 16 Juli 2018

Penulis

diahshintad@gmail.com



ABSTRAK

Kemajuan teknologi yang ada, khususnya pada perangkat bergerak dapat membantu kegiatan manusia. Pada saat ini, wisatawan mempunyai masalah terbatasnya akses informasi mengenai lokasi penginapan hostel dan informasi lengkap hostel tersebut. Berdasarkan data dari *website* Disbudpar Kota Malang, lonjakan wisatawan yang data pada Kota ini cukup signifikan. Hal ini berdampak pada tumbuhnya bisnis hostel di Kota tersebut. Adanya hostel yang memiliki harga murah dan fasilitas yang cukup nyaman membuat banyaknya wisatawan memilih hostel sebagai tempat menginap. Tetapi tidak adanya wadah khusus untuk menampung informasi lengkap penginapan tersebut. Dengan melihat kondisi ini, maka dibuatkanlah aplikasi sistem informasi khusus hostel di Kota Malang yang diimplementasikan pada *smartphone* Android dengan menggunakan metode *User Centered Design* yang mengacu pada *User Experience* para calon pengguna. Tujuan utama dari metode tersebut adalah untuk membuat suatu sistem informasi yang *user-friendly* dengan tingkat *usability* yang tinggi.

Berdasarkan tiga kriteria pengujian *usability* yaitu efektivitas, efisiensi dan tingkat kepuasan pengguna, maka dihasilkan nilai pada pengujian *usability* menurut efektivitas yaitu 91.7%, menurut efisiensi yaitu 100% dan menurut kepuasan pengguna sebesar 84.5%. Dari ketiga hasil tersebut maka dapat dihitung rata-rata tingkat *usability*-nya yaitu sebesar 92.06% dan dapat disimpulkan bahwa aplikasi ini mempunyai tingkat *usability* yang tinggi dan menjadi sebuah aplikasi *user-friendly*.

Kata Kunci: ucd, usability, android, hostel

ABSTRACT

The current technological developments, especially on mobile devices are very helpful to human activities. Travelers have problem of limited access to detail information about the location of hostel. Data from the Department of Culture and Tourism of Malang said, the surge of tourists who data on the city is quite significant. The impact is the growing business of hostels in the city. The cheap price and the complete facilities are the factors that make many travellers choose a hostel as a place to stay. However, there is no special container to accommodate the lodging's complete information. By looking at these conditions, the application of a special hostel information system in Malang was developed. This application is implemented on Android smartphones by using User Centered Design method that refers to the User Experience of potential users. The main purpose of the method is to create a user-friendly information system with a high usability level.

Based on three usability testing criteria are effectiveness, efficiency and user satisfaction level. Then the resulting value on usability testing according to effectiveness is 91.7%, according to efficiency is 100%, and according to user satisfaction of 84.5%. From these three results can be calculated average usability rate of 92.06%. So, it can be concluded that this application has a high level of usability and become a user-friendly application.

Keywords: ucd, usability, android, hostels

DAFTAR ISI

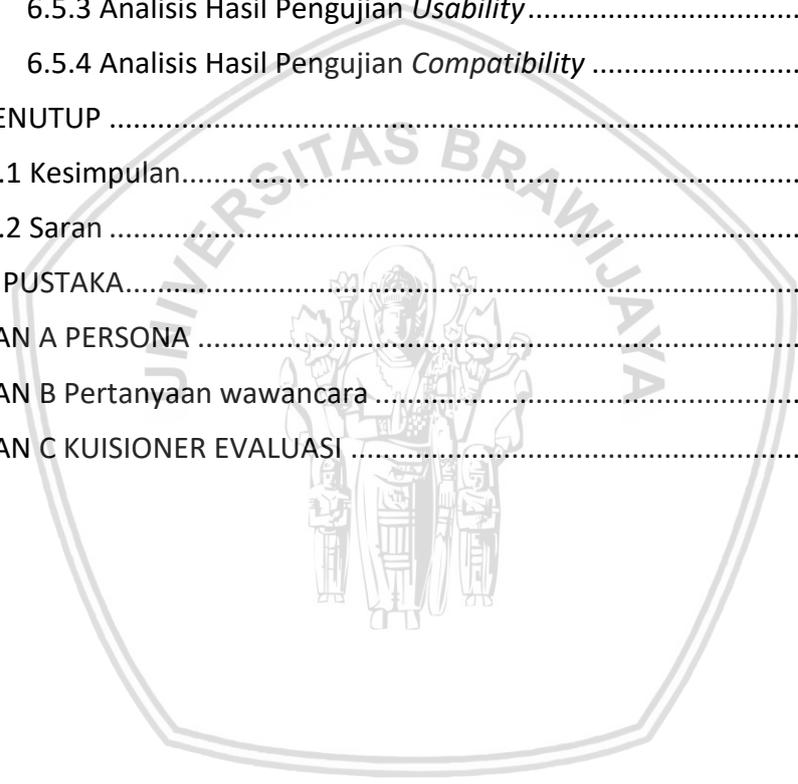
PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
ABSTRACT	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR.....	xiii
DAFTAR LAMPIRAN	xvi
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	3
1.3 Tujuan	3
1.4 Manfaat.....	3
1.5 Batasan masalah	3
1.6 Sistematika pembahasan.....	4
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Kajian Pustaka	5
2.2 Android	5
2.2.1 Jenis Android	5
2.3 API (Application Programming Interface).....	7
2.3.1 API Foursquare	7
2.4 LBS (Location Based Service)	7
2.4.1 Komponen LBS	7
2.5 UCD (User Centered Design).....	8
2.5.1 Prinsip Utama pada UCD	9
2.5.2 Target Pengguna UCD	9
2.5.3 Proses Perancangan metode <i>User Centered Design</i>	10
2.6 Usability Testing.....	10
2.6.2 Tujuan Usability Testing	12



2.6.3 Langkah Usability Testing.....	12
2.6.4 Jumlah Partisipan Usability Testing	13
2.7 <i>Post-Study System Usability Questionnaire</i> (PSSUQ).....	13
2.8 Skala Likert.....	14
2.9 Pengujian <i>Compatibility</i>	15
2.10 Pengujian <i>Validasi</i>	15
2.11 Pengujian <i>Unit</i>	15
BAB 3 METODOLOGI	17
3.1 Studi Literatur	17
3.2 Analisis Kebutuhan	18
3.3 Perancangan	18
3.4 Implementasi	19
3.5 Pengujian	20
3.6 Pengambilan Kesimpulan dan Saran	20
BAB 4 REKAYASA KEBUTUHAN.....	21
4.1 Analisis Kebutuhan	21
4.1.1 Analisis Kebutuhan Iterasi 1.....	21
4.1.2 Analisis Kebutuhan Iterasi 2.....	37
4.1.3 Analisis Kebutuhan Iterasi 3.....	41
BAB 5 PERANCANGAN DAN IMPLEMENTASI	45
5.1 Perancangan	45
5.1.1 Perancangan Sistem.....	45
5.1.2 Perancangan Perangkat Lunak.....	46
5.2 Implementasi	71
5.2.1 Spesifikasi Sistem	71
5.2.2 Batasan Implementasi.....	72
5.2.3 Implementasi Database	72
5.2.4 Implementasi Kode Program	72
5.2.5 Implementasi Antarmuka	76
5.2.6 Implementasi Screenflow Antarmuka	82
BAB 6 PENGUJIAN	84
6.1 Pengujian <i>Unit</i>	84



6.1.1 Pengujian <i>Method whatsappInstalledOrNot</i>	84
6.1.2 Pengujian <i>Method InitData</i>	86
6.2 Pengujian <i>Validasi</i>	87
6.3 Pengujian <i>Usability</i>	104
6.4 Pengujian <i>Compatibility</i>	110
6.5 Analisis Hasil Pengujian.....	111
6.5.1 Analisis Hasil Pengujian Unit	112
6.5.2 Analisis Hasil Pengujian Validasi	112
6.5.3 Analisis Hasil Pengujian <i>Usability</i>	112
6.5.4 Analisis Hasil Pengujian <i>Compatibility</i>	113
BAB 7 PENUTUP	114
7.1 Kesimpulan.....	114
7.2 Saran	114
DAFTAR PUSTAKA.....	116
LAMPIRAN A PERSONA	118
LAMPIRAN B Pertanyaan wawancara	120
LAMPIRAN C KUISIONER EVALUASI	130



DAFTAR TABEL

Tabel 2.1 Tabel Parameter API Foursquare	7
Tabel 2.2 Ukuran tingkat <i>Usability</i>	11
Tabel 2.3 Kuisisioner berdasarkan PSSUQ versi ketiga	13
Tabel 2.4 Tabel Skala Likert.....	15
Tabel 4.1 Identifikasi Aktor	22
Tabel 4.2 Rekap Kebutuhan Pengguna	22
Tabel 4.3 Kebutuhan Fungsional Non-Member	23
Tabel 4.4 Kebutuhan Fungsional Member	23
Tabel 4.5 Kebutuhan Non - Fungsional Member	24
Tabel 4.6 <i>Use Case Scenario Register</i>	25
Tabel 4.7 <i>Use Case Scenario Login</i>	26
Tabel 4.8 <i>Use Case Scenario</i> Menambah Daftar Keinginan	27
Tabel 4.9 <i>Use Case Scenario</i> Melihat List Hostel	28
Tabel 4.10 <i>Use Case Scenario</i> Melakukan Filter	28
Tabel 4.11 <i>Use Case Scenario</i> Melihat Informasi	29
Tabel 4.12 <i>Use Case Scenario</i> Melihat Rute	30
Tabel 4.13 <i>Use Case Scenario</i> Membagikan ke Media Sosial	30
Tabel 4.14 <i>Use Case Scenario</i> Menulis Komentar	31
Tabel 4.15 <i>Use Case Scenario</i> Melihat Komentar	32
Tabel 4.16 <i>Use Case Scenario</i> Menghapus Komentar	32
Tabel 4.17 <i>Use Case Scenario</i> Memberikan Rating	33
Tabel 4.18 <i>Use Case Scenario</i> Melihat <i>Point of Interest</i>	34
Tabel 4.19 <i>Use Case Scenario</i> Melihat Akun	34
Tabel 4.20 <i>Use Case Scenario</i> Melakukan Edit Profil	35
Tabel 4.21 <i>Use Case Scenario</i> Merubah Foto Profil	36
Tabel 4.22 <i>Use Case Scenario</i> Logout	36
Tabel 4.23 Kebutuhan Fungsional Member Iterasi Kedua.....	37
Tabel 4.24 <i>Use Case Scenario</i> Melihat Rating Hostel	39
Tabel 4.25 <i>Use Case Scenario</i> Menghapus Akun	39
Tabel 4.26 <i>Use Case Scenario</i> Menghubungi Pihak Hostel	40



Tabel 4.27 Kebutuhan Fungsional Member Iterasi Ketiga.....	41
Tabel 4.28 <i>Use Case Scenario</i> Menghapus Daftar Keinginan	42
Tabel 4.29 <i>Use Case Scenario</i> Melihat Gambar.....	43
Tabel 5.1 Tabel Keterangan Skema Basis Data <i>User</i>	52
Tabel 5.2 Tabel Keterangan Skema Basis Data <i>Hostel</i>	53
Tabel 5.3 Tabel Keterangan Skema Basis Data <i>Wishlist</i>	54
Tabel 5.4 Tabel Keterangan Skema Basis Data <i>Venue</i>	55
Tabel 5.5 Spesifikasi Perangkat Keras Komputer	71
Tabel 5.6 Spesifikasi Perangkat Lunak Komputer	71
Tabel 5.7 Tabel Penjelasan Method <code>whatsappInstalledOrNot</code>	73
Tabel 5.8 Tabel Penjelasan Method <code>initComponent</code>	74
Tabel 5.9 Tabel Penjelasan Method <code>initData</code>	75
Tabel 6.1 Kasus Uji Pengujian Unit <i>Method whatsappInstalledOrNot</i>	85
Tabel 6.2 Kasus Uji Pengujian Unit <i>Method InitData</i>	87
Tabel 6.3 Tabel Pengujian Validasi	93
Tabel 6.4 Tabel Hasil Efektivitas	104
Tabel 6.5 Tabel Hasil Efisiensi	106
Tabel 6.6 Tabel Kepuasan Pengguna	108
Tabel 6.7 Tabel Hasil Uji Tes <i>Compatibility</i>	110
Tabel 6.8 Presentasi Hasil Pengujian Kepuasan Pengguna	112



DAFTAR GAMBAR

Gambar 2.1 Proses <i>User Centered Design</i>	10
Gambar 3.1 Kerangka Metode Penelitian.....	17
Gambar 4.1 Diagram Use Case Iterasi Pertama	25
Gambar 4.2 Diagram Use Case Iterasi Kedua	38
Gambar 4.3 Diagram Use Case Iterasi Ketiga	42
Gambar 5.1 Perancangan Arsitektur Sistem	45
Gambar 5.2 Arsitektur Sistem Aplikasi Mobile	45
Gambar 5.3 Arsitektur Sistem Aplikasi Roommate.....	46
Gambar 5.4 Diagram <i>Sequence</i> Menampilkan Daftar Hostel	46
Gambar 5.5 Diagram <i>Sequence</i> Melihat Informasi	47
Gambar 5.6 Diagram <i>Sequence</i> Melihat <i>Point Of Interest</i>	48
Gambar 5.7 Perancangan <i>Class Diagram</i>	48
Gambar 5.8 Perancangan <i>Class Diagram</i> <i>authPackage</i>	49
Gambar 5.9 Perancangan <i>Class Diagram</i> <i>accountPackage</i>	49
Gambar 5.10 Perancangan <i>Class Diagram</i> <i>listPackage</i>	50
Gambar 5.11 Perancangan <i>Class Diagram</i> <i>detailsPackage</i>	51
Gambar 5.12 Perancangan <i>Class Diagram</i> <i>poiPackage</i>	51
Gambar 5.13 Perancangan <i>Class Diagram</i> <i>wishlistPackage</i>	52
Gambar 5.14 Struktur json <i>schema User</i> pada Firebase	52
Gambar 5.15 <i>Authentication email</i> dan <i>password</i>	52
Gambar 5.16 Struktur json <i>schema Hostel</i> pada Firebase.....	53
Gambar 5.17 Struktur json <i>schema Wishlist</i> pada Firebase	54
Gambar 5.18 Struktur json <i>schema Venue</i> pada Firebase	55
Gambar 5.19 ER Diagram	56
Gambar 5.20 Algoritme menghubungi pihak hostel menggunakan whatsapp....	56
Gambar 5.21 Algoritme Membuka Aplikasi Whatsapp	57
Gambar 5.22 Algoritme Menampilkan Informasi Detail Hostel	58
Gambar 5.23 Perancangan Antarmuka Halaman <i>Register</i>	59
Gambar 5.24 Perancangan Antarmuka Halaman <i>Login</i>	59
Gambar 5.25 Perancangan Antarmuka Halaman Daftar Hostel.....	60

Gambar 5.26 Perancangan Antarmuka Halaman Informasi	60
Gambar 5.27 Perancangan Antarmuka Halaman Lihat Tipe Kamar	61
Gambar 5.28 Perancangan Antarmuka Halaman Wishlist	61
Gambar 5.29 Perancangan Antarmuka Halaman Akun	62
Gambar 5.30 Perancangan <i>Screenflows</i> Antarmuka Iterasi Pertama	62
Gambar 5.31 Perancangan Antarmuka Halaman <i>Splashscreen</i>	64
Gambar 5.32 Perancangan Antarmuka Halaman Utama.....	64
Gambar 5.33 Perancangan Antarmuka Halaman List Hostel.....	65
Gambar 5.34 Perancangan Antarmuka Halaman <i>Wishlist</i>	65
Gambar 5.35 Perancangan Antarmuka Halaman <i>Point Of Interest</i>	66
Gambar 5.36 Perancangan Antarmuka Halaman Akun	66
Gambar 5.37 Perancangan Antarmuka Halaman <i>Edit Profil</i>	67
Gambar 5.38 Perancangan Antarmuka Halaman <i>Login</i>	67
Gambar 5.39 Perancangan Antarmuka Halaman <i>Register</i>	68
Gambar 5.40 Perancangan Antarmuka Halaman Detail Informasi Hostel.....	68
Gambar 5.41 Perancangan <i>Screenflow</i> Antarmuka Iterasi Kedua.....	69
Gambar 5.42 Perancangan Antarmuka Halaman <i>Wishlist</i>	70
Gambar 5.43 Perancangan <i>Screenflow</i> Antarmuka Iterasi Kedua.....	70
Gambar 5.44 Implementasi Database	72
Gambar 5.45 Implementasi Method <code>whatsappInstalledOrNot</code>	73
Gambar 5.46 Implementasi Method <code>initComponent</code>	74
Gambar 5.47 Implementasi Method <code>initData</code>	75
Gambar 5.48 Implementasi Halaman <i>Splashscreen</i>	76
Gambar 5.49 Implementasi Halaman <i>Login</i>	77
Gambar 5.50 Implementasi Halaman <i>Register</i>	77
Gambar 5.51 Implementasi Halaman Menu Utama	78
Gambar 5.52 Implementasi Halaman List Hostel.....	78
Gambar 5.53 Implementasi Halaman Detail Hostel.....	79
Gambar 5.54 Implementasi Halaman <i>Wishlist</i>	79
Gambar 5.55 Implementasi Halaman Akun	80
Gambar 5.56 Implementasi Halaman <i>Edit Profile</i>	80
Gambar 5.57 Implementasi Halaman POI	81



Gambar 5.58 Implementasi Screenflow Aplikasi 82

Gambar 6.1 Kode *Method* whatsappInstalledOrNot 84

Gambar 6.2 *Flow Graph Method* whatsappInstalledOrNot 84

Gambar 6.3 Kode *Method* InitData 86

Gambar 6.4 *Flow Graph Method* InitData 86



DAFTAR LAMPIRAN

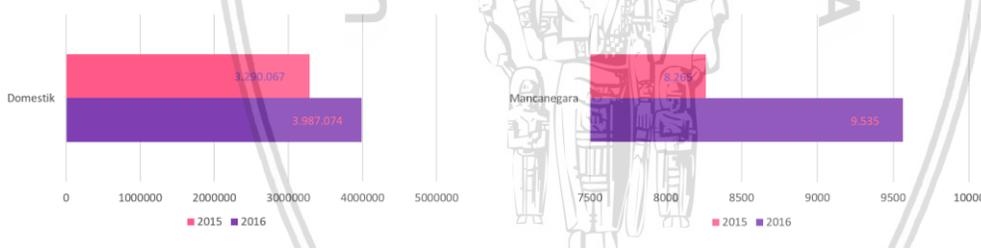
LAMPIRAN A PERSONA	118
LAMPIRAN B Pertanyaan wawancara	120
LAMPIRAN C KUISIONER EVALUASI	130



BAB 1 PENDAHULUAN

1.1 Latar belakang

Banyaknya tempat wisata yang ada, menjadikan Kota Malang sebagai Kota dengan tujuan para wisatawan untuk berlibur. Tidak hanya wisatawan lokal, wisatawan mancanegara juga datang mengunjungi Kota Malang. Kota Malang terletak pada sisi selatan provinsi Jawa Timur. Berada pada dataran tinggi dan dikelilingi oleh pegunungan menyebabkan Kota Malang mempunyai hawa sejuk. Beberapa Hotel besar telah berdiri untuk memberikan fasilitas penginapan bagi wisatawan yang berkunjung ke Kota Malang. Adanya bandara di Kota Malang memudahkan akses para wisatawan untuk mengunjungi Kota ini. Berdasarkan data dari Disbudpar Kota Malang, selama tahun 2015 dan 2016 terdapat peningkatan jumlah wisatawan yang cukup signifikan. Pada tahun 2015 tercatat jumlah wisatawan domestik yang masuk ke Kota Malang sebanyak 3.290.067, sedangkan untuk wisatawan mancanegara sebanyak 8.265 pengunjung. Pada tahun 2016 lonjakan wisatawan ke Kota Malang tergolong signifikan karena jumlahnya naik menjadi 3.987.074 untuk wisatawan domestik, dan 9.535 pengunjung untuk wisatawan mancanegara (Publik, 2017). Banyaknya wisatawan yang datang berarti fasilitas pendukung untuk para wisatawan juga harus tersedia cukup banyak. Hal ini berdampak dengan tumbuhnya bisnis hostel di Kota ini. Selain itu ada juga aplikasi *smartphone* yang menyediakan fitur untuk melakukan reservasi hotel langsung melalui aplikasi tersebut. Pada gambar 1.1 dibawah ini adalah grafik kenaikan jumlah wisatawan Kota Malang pada tahun 2015 dan 2016.



Gambar 1.1 Grafik Kenaikan Jumlah Wisatawan Domestik dan Mancanegara Kota Malang

Wisatawan yang datang ke Kota Malang memiliki variasi pilihan tempat untuk menginap yang lebih banyak, karena tidak semua wisatawan yang datang memiliki *budget* yang cukup untuk menginap di Hotel. Adanya hostel baru di Kota Malang ini memudahkan para wisatawan memiliki penginapan yang nyaman dan murah. Tetapi tidak ada wadah khusus yang berisi tentang informasi tentang hostel tersebut. Hal ini cukup memberi dampak negatif untuk pihak hostel dan para wisatawan. Pihak hostel tidak dapat memaksimalkan bisnisnya karena tidak adanya suatu media yang mempertemukan pihak hostel dan para wisatawan dalam satu wadah. Pihak hostel melakukan promosi dibantu dengan media sosial yang ada seperti Instagram, Facebook dan lainnya. Untuk beberapa orang yang tidak memiliki akun media sosial akan sedikit mengalami kesulitan untuk mengetahui promosi yang diadakan oleh pihak hostel.

Kemajuan teknologi yang ada khususnya pada perangkat bergerak atau *smartphone* dapat membantu kegiatan manusia. Pada saat ini, wisatawan mempunyai masalah terbatasnya

akses informasi mengenai lokasi penginapan yang berada di Kota Malang. Salah satu manfaat dari kemajuan teknologi yang ada adalah dapat membantu memberikan informasi dimana lokasi penginapan dengan konsep *backpacker* atau dengan nama lain hostel. Ketika para pengunjung tersebut berada pada daerah yang baru atau awam bagi mereka, maka akan menghabiskan waktu untuk mencari beberapa penginapan atau hostel terdekat dan rute untuk menuju ke lokasi tersebut.

Untuk membantu pembangunan aplikasi Informasi hostel dengan studi kasus Kota Malang, memanfaatkan teknologi *Location Based Service* yang artinya menggabungkan antara proses dari layanan mobile dengan posisi geografis dari penggunanya. Guna dari *Location Based Service* adalah untuk mencari lokasi fasilitas umum di sekitar pengguna. Aplikasi ini menggunakan data yang didapatkan dari Foursquare, dimana data tersebut akan terus berkembang seiring dengan banyaknya data yang dimasukkan oleh pengguna lain pada Foursquare. Perpaduan antara internet, dan teknologi GPS yang dikembangkan memberikan kemudahan bagi para wisatawan asing untuk mencari tempat-tempat wisata menarik (Sharma, et al., 2011).

Untuk membuat sebuah sistem informasi yang *user-friendly* dengan tingkat *usability* yang tinggi serta memberikan kesesuaian aplikasi dengan kebutuhan yang dibutuhkan oleh pengguna, maka digunakanlah metode UCD atau *User Centered Design*. UCD merupakan sebuah metode untuk membangun sebuah aplikasi yang mempunyai konsep, pengguna sebagai pusat dari pembangunan sistem. Dimulai dari tujuan sampai penentuan desain tergantung dari *experience* dari pengguna. UCD adalah proses yang interaktif dimana langkah perancangan dan evaluasi dibuat didalam permulaan proyek sampai implementasi (Devi, et al., 2012).

Memastikan bahwa tujuan sistem dengan tingkat *usability* yang tinggi, maka harus dilakukan evaluasi *usability* dari kebutuhan pengguna. Beberapa evaluasi yang dilakukan adalah efektivitas, efisiensi dan tingkat kepuasan pengguna. Untuk mengetahui apakah sistem dapat berjalan dengan baik pada berbagai *device* maka dilakukan pengujian *compatibility* dengan parameter uji yaitu resolusi layar, sistem operasi dan fitur sensor giroskop.

Berdasarkan latar belakang tersebut, akan dibuat aplikasi berbasis Android yang berjudul "Penerapan *User Centered Design* pada Pembangunan Aplikasi Informasi Hostel Berbasis Android". Aplikasi dengan nama "*Roommate*" ini dapat memberikan info pada penggunanya lokasi penginapan atau hostel yang ada di kota malang dengan data daftar hostel yang didapatkan dari Foursquare API dan rute untuk menuju lokasi tersebut. Selain itu aplikasi ini memiliki fitur memberikan rating dan komentar untuk kritik dan saran kepada hostel, memiliki fitur *points of interest* berisi tempat wisata atau tempat penting lainnya terdekat dengan lokasi penginapan.

1.2 Rumusan masalah

Berdasarkan uraian dari latar belakang diatas, maka dapat dirumuskan permasalahannya yaitu:

1. Bagaimana proses pembangunan sistem informasi hostel dengan menggunakan metode *User Centered Design*?

2. Bagaimana memanfaatkan API dari Foursquare untuk pembangunan aplikasi informasi hostel berbasis Android?
3. Bagaimana tingkat efektivitas, efisiensi dan kepuasan pengguna dalam penggunaan aplikasi informasi hostel berbasis Android?

1.3 Tujuan

1. Membangun sistem informasi hostel dengan menggunakan metode *User Centered Design*.
2. Menggunakan data dari API Foursquare untuk pembangunan aplikasi informasi Hostel berbasis Android.
3. Mengetahui tingkat efektivitas, efisiensi dan kepuasan pengguna dalam penggunaan aplikasi informasi Hostel berbasis Android.

1.4 Manfaat

Manfaat dari penelitian ini adalah dapat membantu para wisatawan yang ingin melakukan perjalanan di Kota Malang dengan memberikan informasi tentang penginapan dengan tipe hostel dan adanya *points of interest* terdekat dengan penginapan. Para pelaku usaha penginapan terutama hostel juga dapat memaksimalkan usahanya. Sehingga kedua belah pihak antara wisatawan dan pemilik hostel dapat sama-sama diuntungkan dengan adanya aplikasi ini.

1.5 Batasan masalah

Agar permasalahan yang telah dirumuskan lebih terfokus dan tidak terjadi pelebaran topik, maka penelitian ini dibatasi dalam hal:

1. Aplikasi ini akan dikembangkan pada perangkat *mobile* dengan platform Android.
2. Proses pembuatan aplikasi ini menggunakan Android Studio.
3. Data yang digunakan pada aplikasi ini didapatkan dari API Foursquare dan Openweathermap.
4. Basisdata yang digunakan untuk menyimpan data pada aplikasi ini menggunakan Firebase Realtime Database.
5. Koneksi dengan jaringan internet harus selalu tersedia.
6. Data yang didapatkan dari API beberapa tidak sesuai dengan kategori.
7. Informasi tambahan untuk hostel tidak lengkap.

1.6 Sistematika pembahasan

Sistematika pembahasan ditunjukkan untuk memberikan gambaran dan uraian dari penulisan skripsi ini secara garis besar yang meliputi beberapa bab, sebagai berikut:

BAB I : PENDAHULUAN

Memuat latar belakang, rumusan masalah, tujuan, batasan masalah, manfaat dan sistematika penulisan.

BAB II : LANDASAN KEPUSTAKAAN

Menguraikan tentang dasar teori yang mendasari pembuatan aplikasi informasi Hostel berbasis Android.

BAB III : METODOLOGI

Membahas metode yang digunakan dalam penelitian yang terdiri dari studi literature, perancangan sistem perangkat lunak, implementasi sistem perangkat lunak, pengujian dan analisis, serta penulisan laporan.

BAB IV : REKAYASA KEBUTUHAN

Membahas analisis kebutuhan informasi Hostel berbasis Android.

BAB V : PERANCANGAN DAN IMPLEMENTASI

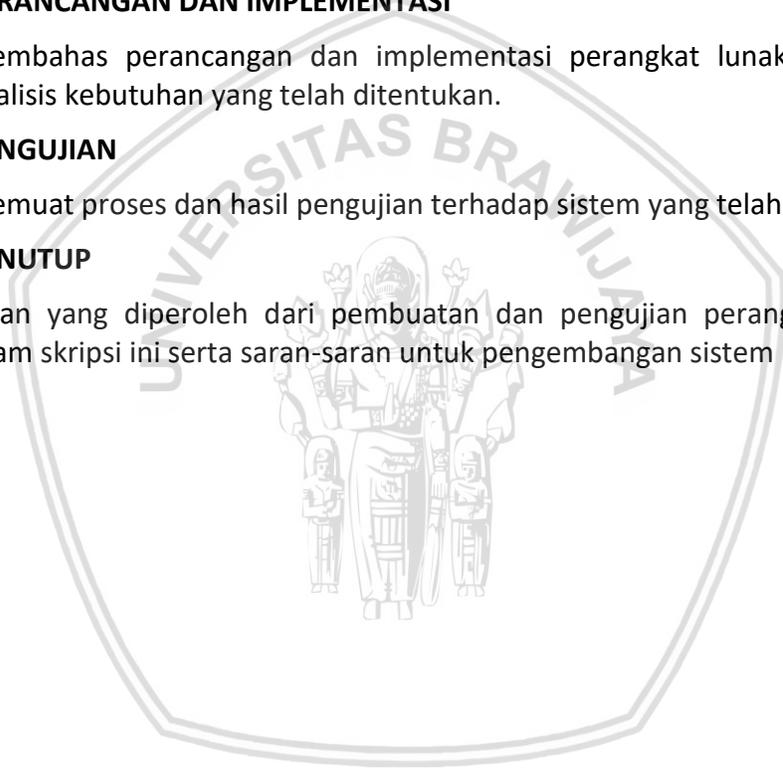
Membahas perancangan dan implementasi perangkat lunak sesuai dengan analisis kebutuhan yang telah ditentukan.

BAB VI : PENGUJIAN

Memuat proses dan hasil pengujian terhadap sistem yang telah direalisasikan.

BAB VII : PENUTUP

Memuat kesimpulan yang diperoleh dari pembuatan dan pengujian perangkat lunak yang dikembangkan dalam skripsi ini serta saran-saran untuk pengembangan sistem lebih lanjut.



BAB 2 LANDASAN KEPUSTAKAAN

Berdasarkan latar belakang dan rumusan masalah, maka bab ini penulis membahas landasan kepustakaan berupa kajian pustaka yang membahas tentang penelitian yang sudah pernah ada serta dasar teori yang digunakan untuk penyusunan penelitian. Kajian pustaka yang dijadikan sebagai acuan adalah penelitian yang berhubungan dengan penggunaan metode *user centered design*.

2.1 Kajian Pustaka

Penelitian sebelumnya yang dilakukan oleh Atikah Hayyu Mulia, membahas tentang bagaimana pengembangan antarmuka sebuah aplikasi dengan pendekatan UCD (*User Centered Design*). Mulai dari tujuan sampai dengan fase-fase dalam UCD diantaranya adalah melakukan analisis, membuat *task flows*, membuat desain hingga melakukan tahap evaluasi. Evaluasi yang dilakukan oleh peneliti adalah pengujian *usability*. Dari hasil pengujian yang dilakukan secara dua kali, nilai *usability* yang didapatkan meningkat sebesar 15% dari sebelumnya yang sebesar 78% menjadi 93%. (Mulia, et al., 2016)

Penelitian selanjutnya dilakukan oleh Intan Sandra Yatana Saputri, Mardhiah Fadhli dan Ibnu Surya yang membahas tentang penerapan metode UCD pada *E-Commerce* berbasis web. Pengujian yang dilakukan yaitu *black box testing*, *5 second testing* dan *usability testing*. Dari hasil pengujian *usability* yang dilakukan, berdasarkan *System Usability Scale* web ini berada pada *range excellent* yaitu sebesar 86.8% dan disimpulkan bahwa web tersebut berhasil membangun sistem yang *user-friendly* dengan tingkat *usability* yang tinggi. (Saputri, et al., 2017)

Penelitian terakhir yang dilakukan oleh Armadyah Amborowati yang diberi judul "Rancangan Sistem Pameran Online menggunakan Metode UCD (User Centered Design)" memberikan kesimpulan bahwa dengan menggunakan metode UCD untuk perancangan sistem, sistem yang dihasilkan lebih memberikan kepuasan bagi user dan meningkatkan kemanfaatan sistem tersebut. (Amadyah, 2008)

2.2 Android

Android adalah sebuah sistem operasi dari smartphone yang berbasis Linux. Android menyediakan platform yang ditujukan untuk para penganmbang jika ingin menciptakan aplikasi mereka sendiri untuk digunakan pada berbagai macam perangkat bergerak (Triadi, 2013).

2.2.1 Jenis Android

a. Astro

Android versi pertama ini dirilis pada 23 September 2008. Ponsel pertama yang menggunakan sistem operasi Android Astro adalah HTC Dream.

b. Bender

- Versi ini dirilis pada tanggal 9 Maret 2009.
- c. Cupcake
Versi ini dirilis pada tanggal 30 April 2009. Mulai dari versi inilah penamaan sistem operasi dari android menggunakan nama makanan pencuci mulut.
 - d. Donut
Versi yang dirilis pada September 2009 dipakai sebagai nama alias dari versi Android 1.6.
 - e. Enclair
Versi yang dirilis pada tanggal 9 Desember 2010 mendapatkan penambahan fitur untuk pengoptimalan hardware, browser baru dan lainnya.
 - f. Froyo (Frozen Yoghurt)
Versi ini dirilis pada tanggal 20 Mei 2010 sebagai nama alias dari sistem operasi Android versi 2.2
 - g. Gingerbread
Versi ini dirilis pada tanggal 6 Desember 2010 dan dipakai sebagai nama alias dari sistem operasi Android versi 2.3
 - h. Honeycomb
Versi yang dirilis pada tanggal 22 Februari 2011 ini dipakai sebagai nama alias dari sistem Android versi 3.0
 - i. Ice Cream Sandwich
Versi tersebut dirilis pada 19 Oktober 2011. Ice Cream Sandwich ini dipakai sebagai nama alias dari Android versi 4.0
 - j. Jelly Bean
Versi ini mendapatkan penambahan baru diantaranya meningkatkan input keyboard, desain baru, UI baru dan Voice Search yang lebih cepat.
 - k. Kitkat
Versi yang rilis pada tanggal 31 Oktober 2013 ini dipakai sebagai nama alias dari sistem Andrid versi 4.4.x
 - l. Lollipop
Versi ini muncul pada tanggal 15 oktober 2014 untuk sistem Android versi 5.0
 - m. Marsmellow
Tanggal rilis dari versi ini adalah 19 Agustus 2015 untuk android versi 6.0
 - n. Nougat
Versi terbaru dari android ini baru saja rilis pada tanggal 22 Agustus 2016 untuk sistem android versi 7.0

2.3 API (Application Programming Interface)

2.3.1 API Foursquare

API Foursquare menyediakan sebuah akses ke database aplikasi tersebut. Banyak dari para developer menggunakan API tersebut untuk membuat sebuah aplikasi berbasis lokasi. Data-data yang tersimpan pada database foursquare tersebut bukan dibuat oleh Foursquare, melainkan dibuat dan dirilis oleh pengguna aplikasi tersebut. Pada tabel 2.1 dijelaskan parameter yang diambil pada API Foursquare.

Tabel 2.1 Tabel Parameter API Foursquare

NAMA	DESKRIPSI
<i>ll</i>	Latitude dan Longitude dari Kota Malang.
<i>radius</i>	Batasan hasil hanya beberapa meter dari lokasi yang spesifik.
<i>categoryId</i>	Sebuah nomor identitas dari setiap kategori tempat pada API Foursquare.

2.4 LBS (Location Based Service)

Location Based Service (LBS) memberikan layanan bagi pengguna perangkat *mobile* yang disesuaikan dengan posisi mereka saat ini. LBS membuka pasar baru bagi pengembang, operator jaringan selular, dan penyedia layanan untuk mengembangkan dan memberikan nilai tambah layanan (Singhal & Shukla, 2012).

2.4.1 Komponen LBS

2.4.1.1 Piranti Mobile

Salah satu komponen yang berfungsi sebagai alat untuk penggunaanya mencari sebuah informasi. Piranti *mobile* bisa berupa Laptop maupun *smartphone*.

2.4.1.2 Jaringan Komunikasi

Komponen yang berfungsi sebagai jalur penghubung untuk pengiriman data-data yang dibutuhkan serta hasil tersebut dikirimkan kembali kepada pengguna. Komponen tersebut adalah jaringa komunikasi.

2.4.1.3 Komponen Positioning (Penunjuk Posisi/Lokasi)

Komponen ini berfungsi sebagai penentu lokasi pengguna saat ini. Posisi tersebut bisa didapatkan melalui jaringan komunikasi dan *Global Positioning System* (GPS).

2.4.1.4 Penyedia layanan dan aplikasi

Penyedia layanan adalah sebuah komponen LBS yang berfungsi untuk memberikan layanan yang digunakan oleh pengguna. Salah satu contohnya adalah ketika pengguna meminta untuk mengetahui posisi mereka saat ini, layanan dan aplikasi akan langsung memproses permintaan tersebut.

2.4.1.5 Penyedia data dan konten

Tidak semua penyedia layanan menyimpan data data yang telah diinputkan atau diterima oleh pengguna, karena bisa jadi jika data tersebut berasal dari pihak ketiga dan memiliki otoritas untuk menyimpannya. Sebagai contoh basis data geografis dan lokasi bisa saja berasal dari badan-badab milik pemerintah atau juga data-data perusahaan/bisnis/industry bisa saja berasal dari *Yellow Pages*, maupun perusahaan penyedia data lainnya.

2.5 UCD (User Centered Design)

Istilah User Centered Design pertama kali muncul di laboratorium University of California San Diego (UCSD) oleh Donald Norman's pada tahun 1980 (Abrams et al. 2004) dan menjadi terkenal setelah penerbitan buku yang berjudul "User-Centered System Design: New Perspectives on Human-Computer Interaction" (Norman & Draper 1986). Konsep dari metode ini adalah user sebagai pusat pengembangan sistem, dan tujuan/sifat-sifat, konteks dan lingkungan sistem semua didasarkan dari pengalaman pengguna (Amadyah, 2008).

Prinsip yang harus diperhatikan dalam UCD adalah sebagai berikut (Zahara, 2013):

1) Fokus pada pengguna

Perancangan harus berhubungan langsung dengan pengguna sesungguhnya atau calon pengguna melalui interview, survey, dan partisipasi dalam workshop perancangan. Tujuannya adalah untuk memahami kognisi, karakter, dan sikap pengguna serta karakteristik anthropometric. Aktivitas utamanya mencakup pengambilan data, analisis dan integrasinya ke dalam informasi perancangan dari pengguna tentang karakteristik tugas, lingkungan teknis, dan organisasi.

2) Perancangan terintegrasi

Perancangan harus mencakup antarmuka pengguna, sistem bantuan, dukungan teknis serta prosedur instalasi dan konfigurasi.

3) Pengujian pengguna

Satu-satunya pendekatan yang sukses dalam perancangan sistem yang berpusat pada pengguna adalah secara empiris dibutuhkan observasi tentang kelakuan pengguna, evaluasi umpan-balik yang cermat, wawasan pemecahan terhadap masalah yang ada, dan motivasi yang kuat untuk mengubah rancangan.

4) Perancangan interaktif

Sistem yang sedang dikembangkan harus didefinisikan, dirancang, dan dites berulang kali. Berdasarkan hasil test kelakuan dari fungsi, antarmuka, sistem bantuan, dokumentasi pengguna, dan pendekatan pelatihannya.

2.5.1 Prinsip Utama pada UCD

Berikut ini adalah prinsip yang harus diperhatikan dalam UCD: (Amadyah, 2008)

1. Fokus pada pengguna

Perancangan harus berhubungan langsung dengan pengguna sesungguhnya atau calon melalui *interview*, *survey* dan partisipasi dalam workshop perancangan. Tujuannya adalah untuk memahami kognisi, karakter dan sikap pengguna.

2. Perancangan terintegrasi

Perancangan harus mencakup antarmuka pengguna, sistem bantuan, dukungan teknis serta prosedur instalasi dan konfigurasi.

3. Dari awal berlanjut pada pengujian pengguna

Satu satunya pendekatan yang sukses dalam perancangan sistem yang berpusat pada pengguna adalah secara empiris dibutuhkan observasi tentang kelakuan pengguna, evaluasi, wawasan pemecahan terhadap masalah yang ada, dan motivasi kuat.

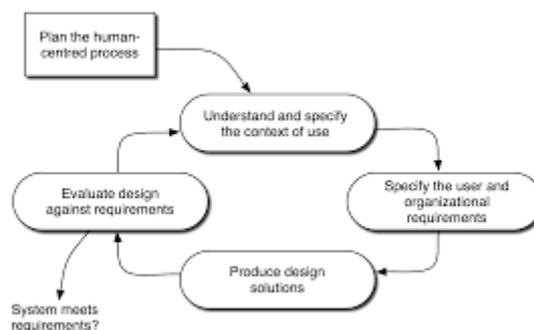
4. Perancangan interaktif

Sistem yang sedang dikembangkan harus didefinisikan, dirancang dan dites berulang kali. Berdasarkan hasil tes kelakuan dari fungsi, antarmuka, sistem bantuan, dokumentasi pengguna, dan pendekatan pelatihannya.

2.5.2 Target Pengguna UCD

Pengguna dapat berasal dari keluarga, teman atau orang asing, tidak perlu memiliki pengetahuan tentang aplikasi yang diteliti, tujuan penelitian dan sebagainya. Semakin dekat hubungan antar peneliti dan target, semakin lebih mudah mengetahui apakah target pengguna tersebut sudah memenuhi kriteria yang telah ditentukan atau belum. Selain itu, hasil yang didapatkan akan lebih realistis (Barnum, 2011)

2.5.3 Proses Perancangan metode *User Centered Design*



Gambar 2.1 Proses *User Centered Design*

Sumber : ISO 13409 (1999)

Dalam proses dari metode UCD, ada 4 langkah yang dilakukan secara iterasi seperti pada gambar 2.1. Keterangan dari metode diatas yaitu sebagai berikut:

1. *Specify the context of use*

Pada tahap ini dilakukan identifikasi siapa saja yang akan menggunakan sistem. Menjelaskan tentang produk apa yang akan dibuat dan dalam kondisi seperti apa mereka akan menggunakan produk ini.

2. *Specify User and Organizational Requirement*

Proses ini dilakukan untuk mengidentifikasi kebutuhan dari pengguna. Peneliti akan melakukan wawancara untuk mendapatkan informasi kebutuhan fungsional dan non fungsional yang akan diterapkan pada aplikasi. Untuk pertanyaan dari wawancara telah di lampirkan pada halaman Lampiran 1.

3. *Produce Design Solution*

Mulai membangun desain atau *prototype* sebelum benar-benar di implementasikan dari kebutuhan fungsional dan non fungsional yang telah didapat sebagai solusi dari sistem yang sedang dibangun.

4. *Evaluate Design*

Melakukan evaluasi terhadap desain pada tahap sebelumnya apakah tujuan atau kebutuhan dari calon pengguna sudah tercapai. Tahap evaluasi ini menggunakan teknik kuisiner yang disebar pada para calon pengguna.

2.6 Usability Testing

Usability digunakan untuk mengukur tingkat pengalaman pengguna ketika berinteraksi dengan produk baik itu *website*, *software*, *mobile phone*, maupun peralatan yang digunakan pengguna lainnya. Secara umum *usability*, adalah bagaimana pengguna bisa mempelajari dan menggunakan produk untuk memperoleh tujuannya dan seberapa puaskah mereka terhadap penggunaannya (Dumas & Fox, 2008)

Tabel 2.2 Ukuran tingkat *Usability*

No	Kriteria	Shackel (1990)	Nielson (1993)	ISO 9241-11 (1998)	Preece dkk. (2002)
1	Efektivitas	√		√	√
2	Efisiensi		√	√	√
3	<i>Learnability</i>	√	√		
4	<i>Memorability</i>	√	√		
5	Fleksibilitas	√			
6	Kesalahan		√		
7	Utilitas				√
8	<i>Safety</i> (keamanan)				√
9	Kepuasan		√	√	
10	<i>Attitude</i> (perilaku)	√			

Berikut ini adalah uraian kriteria pada Tabel 2.1 (Setia, 2012),

a. Efektivitas

Efektivitas berhubungan dengan keberhasilan pengguna mencapai tujuan dalam menggunakan suatu perangkat lunak.

b. Efisiensi

Untuk mengukur tingkat performa dalam menggunakan produk.

c. *Learnability*

Sebagai tingkat kemudahan sistem untuk dipelajari, diukur melalui waktu yang diperlukan untuk mempelajari penggunaan sistem hingga mencapai level kemahiran tertentu.

d. *Memorability*

Berhubungan dengan proses *recalling* cara pemakaian sistem setelah pengguna tidak berinteraksi dengan sistem tersebut selama beberapa waktu.

e. Fleksibilitas

Fleksibilitas berkaitan dengan variasi pengerjaan suatu *task* sistem.

f. *Error* (kesalahan)

Frekuensi kesalahan yang tinggi pada saat penggunaan sistem mengindikasikan rendahnya *usability* sistem yang bersangkutan.

g. Utilitas

Utilitas sebagai acuan tingkat fungsionalitas sebuah sistem yang dapat digunakan pengguna untuk menyelesaikan tugas.

h. *Safety* (Keamanan)

Sistem keamanan mencegah kerusakan fatal pada sistem dari kondisi yang tidak diinginkan.

i. *Satisfaction* (kepuasan)

Kepuasan pengguna terhadap sistem yang dipakai mengindikasikan bahwa sistem tersebut layak pakai.

j. *Attitude* (perilaku)

Mengukur kriteria perilaku sistem dari bagaimana pengguna menerima dan merasa puas dari sistem yang dipakai.

2.6.2 Tujuan Usability Testing

Beberapa tujuan *usability* sebagai berikut: (Rubin & Chisnell, 2008)

1. Untuk mengetahui dan memperbaiki kekurangan yang ada pada sebuah produk.
2. Untuk mencari permasalahan pada produk dan memperbaiki serta mengembangkan produk tersebut.
3. Memberikan kreasi produk sehingga mudah untuk dipelajari, memberikan kepuasan bagi pengguna, dan memiliki nilai *usability* yang tinggi.

2.6.3 Langkah Usability Testing

Langkah-langkah untuk melakukan *usability testing* adalah sebagai berikut (Nurhadryani, et al., 2011):

1. Menyiapkan sketsa, prototype yang ingin dilakukan pengujian.
2. Merancang alat ukur.

Alat ukur disesuaikan dengan komponen yaitu efektivitas, efisiensi dan kebutuhan pengguna. Pengukuran nilai efektivitas dan efisiensi menggunakan lembar observasi, sedangkan untuk penilaian kepuasan pengguna menggunakan *post-task* kuisisioner.

3. Melaksanakan pengujian

Pada saat observasi, responden melakukan langkah-langkah sesuai dengan elemen observasi. Jika responden berhasil maka diberi nilai "Ya" namun jika gagal maka diberi nilai "Tidak".

4. Menganalisis hasil

Tingkat efektivitas dan efisiensi diukur menggunakan tingkat keberhasilan pengguna dalam menyelesaikan elemen observasi dengan perhitungan sebagai berikut:

$$\text{Efektivitas, Efisiensi (\%)} = \frac{\sum_{i=1}^n x_i}{n} \times 100\% \quad (2.1)$$

Dimana x_i adalah nilai keberhasilan responden ke- i , $x_i = \{0,1\}$.

Sedangkan kepuasan pengguna didapat dari persentase perbandingan antara nilai kepuasan dengan perkalian bobot maksimum skala Linkert (5) dengan jumlah responden (n).

$$\text{Kepuasan (\%)} = \frac{\sum_{i=1}^n x_i}{n} \times 100\% \quad (2.2)$$

Dimana x_i adalah nilai keberhasilan responden ke- i , $x_i = \{0,1,2,3,4,5\}$

Sedangkan nilai *usability* sendiri didapat dari rata-rata efektivitas, efisiensi dan kepuasan pengguna.

$$\text{Usability (\%)} = \frac{(\text{efektivitas} + \text{efisiensi} + \text{kepuasan})}{3} \times 100\% \quad (2.3)$$

2.6.4 Jumlah Partisipan Usability Testing

Hasil terbaik untuk pengujian *usability* adalah tidak lebih dari 5 orang partisipan. Tidak peduli dengan perbedaan platform (Nielsen, 2012). Dengan menggunakan 5 partisipan, hasil yang didapatkan akan hampir mendekati rasio maksimum pengujian pengguna.

2.7 Post-Study System Usability Questionnaire (PSSUQ)

Merupakan kuisisioner untuk mengukur persepsi kepuasan pengguna terhadap sistem atau aplikasi. PSSUQ terdapat 3 versi dengan jumlah pertanyaan yang berbeda. Pada versi terakhir terdapat 16 pertanyaan dengan 4 sub skala. Dapat dilihat pertanyaan kuisisioner berdasarkan PSSUQ dibawah ini (Sauro & Lewis, 2014):

Tabel 2.3 Kuisisioner berdasarkan PSSUQ versi ketiga

No	Elemen Kuisisioner	Jawaban				
		SS	S	N	TS	STS
1	Saya merasa puas dengan betapa mudahnya menggunakan aplikasi ini					
2	Aplikasi ini mudah untuk digunakan					
3	Saya bisa menyelesaikan tugas dan scenario pada aplikasi ini					
4	Saya merasa nyaman menggunakan aplikasi ini					
5	Saya merasa aplikasi ini mudah untuk dipelajari					



No	Elemen Kuisiner	Jawaban				
		SS	S	N	TS	STS
6	Saya bisa menjadi produktif dengan cepat menggunakan aplikasi ini					
7	Aplikasi ini memberi pesan kesalahan yang jelas					
8	Setiap kali membuat kesalahan, saya memperbaiki dengan cepat dan mudah					
9	Informasi yang tersedia pada aplikasi ini sudah jelas					
10	Saya mudah untuk menemukan informasi yang saya butuhkan					
11	Informasi yang efektif membantu saya menyelesaikan tugas dan skenario					
12	Organisasi informasi pada aplikasi ini jelas					
13	Antarmuka pada aplikasi ini menyenangkan					
14	Saya suka menggunakan antarmuka aplikasi ini					
15	Aplikasi ini memiliki semua fungsi dan kemampuan yang saya harapkan					
16	Secara keseluruhan saya merasa puas dengan aplikasi ini					

2.8 Skala Likert

Skala likert mengukur sifat-sifat individu misalnya pengetahuan atau sikap dengan menggunakan skor total dari butir pertanyaan. Skala likert mempunyai 4 atau lebih butir pertanyaan yang dikombinasikan sehingga membentuk sebuah nilai yang merepresentasikan sikap individu. Penggunaan jumlah dari semua butir pertanyaan valid karena setiap butir pertanyaan adalah indicator dari variable yang direpresentasikannya (Budiaji, 2013).

Skala likert menggunakan beberapa butir pertanyaan untuk mengukur perilaku individu dengan merespon 5 titik pilihan pada setiap butir pertanyaan,

sangat setuju, setuju, tidak memutuskan (netral), tidak setuju dan sangat tidak setuju (Likert, 1932).

Tabel 2.4 Tabel Skala Likert

Sumber: (Bertram, n.d)

No	Interval	Nilai
1	0% - 19.99%	Sangat Buruk
2	20% - 39.99%	Buruk
3	40% - 59.99%	Cukup
4	60% - 79.99%	Baik
5	80% - 100%	Sangat Baik

2.9 Pengujian *Compatibility*

Pengujian *compatibility* adalah pengujian yang digunakan untuk memeriksa apakah sistem yang dikembangkan mampu berjalan dengan baik pada *hardware*, sistem operasi dan resolusi layar yang berbeda. Semakin sistem dapat berjalan pada banyak jenis perangkat yang berbeda, maka semakin baik aspek kompatibilitasnya. Dalam standar ISO 25020 mewakili aspek *compatibility*. (Liu, et al., 2014)

Sifat *open source* yang ada android menjadikan banyaknya perangkat yang menjadikan Android sebagai sistem operasinya. Perangkat tersebut tentunya memiliki konfigurasi perangkat yang berbeda seperti fitur, versi platform dan resolusi layar. Fitur pada setiap perangkat bisa jadi berbeda antara satu dengan yang lainnya. Beragamnya versi sistem operasi android juga harus diperhatikan agar aplikasi yang dikembangkan dapat berjalan dengan baik pada berbagai versi. (Anon., 2018)

2.10 Pengujian *Validasi*

Pengujian ini digunakan untuk mengetahui apakah sistem yang dibangun sudah benar sesuai dengan analisis kebutuhan. Item-item yang ada pada daftar kebutuhan dan hasil dari analisis kebutuhan yang menjadi acuan untuk melakukan pengujian validasi. Metode yang digunakan untuk pengujian validasi adalah metode *black box*.

2.11 Pengujian *Unit*

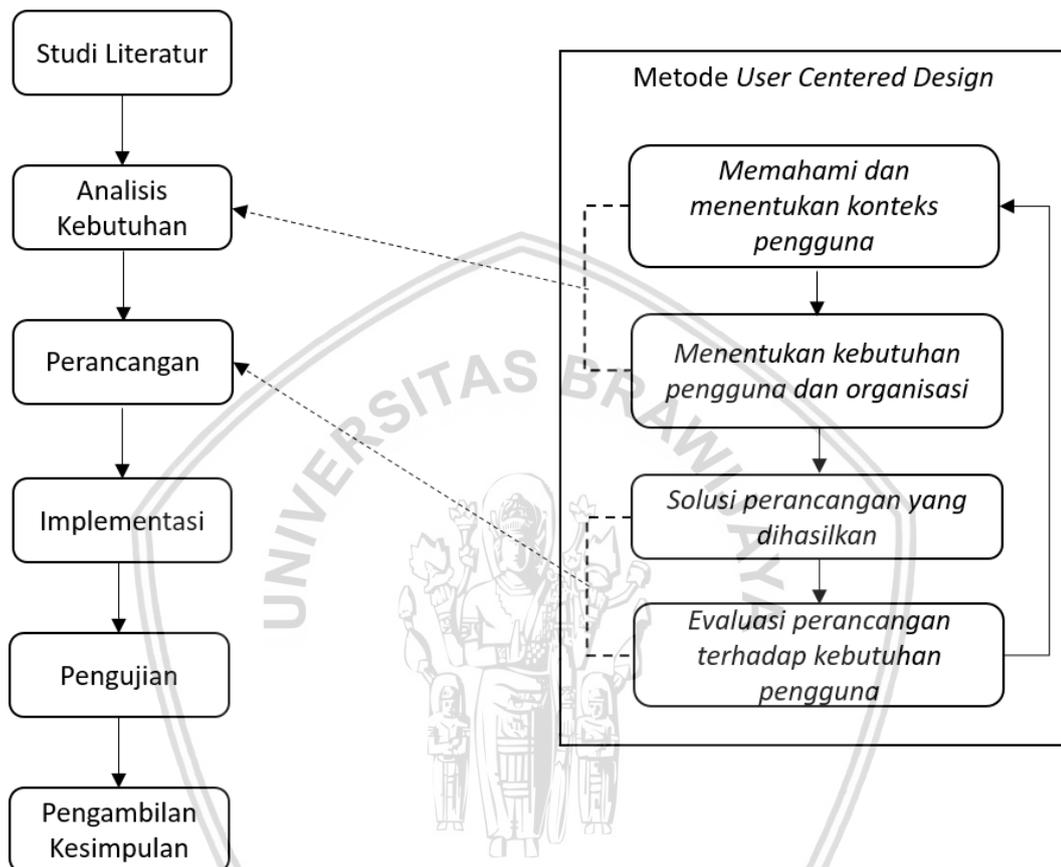
Pengujian ini adalah suatu metode verifikasi perangkat lunak dimana hasil pengujian tersebut adalah perangkat lunak ini layak atau tidaknya dipakai. Pengujian ini berfokus pada unit terkecil dalam sistem. Metode yang digunakan yaitu metode *white box*. Kelebihan dari metode ini adalah menggunakan sintax perulangan untuk mencari dan mendeteksi segala kondisi yang tidak sesuai serta menampilkan dan memonitor beberapa asumsi yang tidak sesuai dengan yang

diharapkan. Kelemahan dari metode pengujian ini adalah pada perangkat lunak yang jenisnya besar, karena metode pengujian ini dianggap boros karena melibatkan banyak melibatkan sumberdaya untuk melakukannya.



BAB 3 METODOLOGI

Pada bab ini dijelaskan mengenai prosedur dan kegiatan yang akan dilakukan dalam pengerjaan skripsi, yaitu studi literatur, analisis kebutuhan, perancangan, implementasi, pengujian dan terakhir mengambil kesimpulan dari pembuatan sistem aplikasi yang akan dibuat.



Gambar 3.1 Kerangka Metode Penelitian

3.1 Studi Literatur

Studi literatur berisi landasan teori yang berhubungan dengan penelitian. Hal ini dilakukan untuk menambah pengetahuan dasar dalam membangun suatu sistem dapat terpenuhi. Teori yang berkaitan dengan penelitian ini ada beberapa yaitu:

1. Android
2. API (*Application Programming Interface*)
3. LBS (*Location Based Service*)
4. UCD (*User Centered Design*)
 - Prinsip utama pada UCD
 - Target pengguna UCD

- Proses perancangan
- 5. *Usability testing*
- 6. *Post-Study System Usability Questionnaire*
- 7. Pengujian *compatibility*
- 8. Pengujian validasi
- 9. Pengujian unit

3.2 Analisis Kebutuhan

Pada sub bab ini adalah tahapan untuk mengidentifikasi kebutuhan dari sistem informasi hostel kota malang. Pada penelitian ini menggunakan metode UCD (*User Centered Design*) dimana ada dua alur untuk analisis kebutuhan yaitu *specify the context of use* dan *specify user and organizational requirement*.

Untuk alur *specify the context of use*, dilakukan identifikasi siapa saja calon pengguna yang akan menggunakan produk ini. Setelah identifikasi aktor, lalu menjalankan alur *specify user and organizational requirement* yaitu mengidentifikasi kebutuhan calon pengguna. Metode untuk melakukan alur ini adalah dengan wawancara kepada 5 calon pengguna yang sudah ditentukan pada alur sebelumnya. Pertanyaan dan hasil wawancara pada alur ini dapat dilihat pada halaman lampiran.

Setelah kedua alur tersebut sudah dilakukan maka mulai melakukan analisis data yang akan digunakan dalam pembangunan aplikasi sistem informasi ini. Dua langkah terakhir adalah membuat *Use Case Diagram* dan *Use Case Scenario*. Pembuatan *Use Case Diagram* digunakan untuk mengetahui fungsi apa saja yang ada dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi tersebut. Untuk menjabarkan alur kinerja atau langkah-langkah tiap *use case*, maka dibuat *Use Case Scenario* agar calon pengguna lebih mudah untuk memahami bagaimana cara kerja sistem.

Pada metode yang digunakan terdapat iterasi atau perulangan ketika mendapatkan evaluasi dari calon pengguna. Ketika selesai melakukan evaluasi dengan cara wawancara calon pengguna, maka hasil tersebut akan di analisis ulang untuk mengidentifikasi kebutuhan yang baru. Setelah mendapatkan analisis kebutuhan, dilanjutkan dengan membuat *use case diagram* dan *use case scenario* untuk kebutuhan yang baru dan atau kebutuhan yang diubah.

3.3 Perancangan

Tahapan selanjutnya adalah tahap perancangan. Tahap ini dibagi menjadi dua yaitu perancangan sistem dan perancangan perangkat lunak. Untuk perancangan sistem terdapat perancangan arsitektur secara umum serta perancangan arsitektur dalam sistem. Alur dari metode UCD (*User Centered Design*) lanjutan dari alur untuk analisis kebutuhan yaitu *product design solution*

dan *evaluate design against requirements* berada pada sub bab perancangan perangkat lunak.

Membuat perancangan antarmuka sebagai *design solution* berdasarkan dari hasil analisis kebutuhan. Pada tahap ini sangatlah penting dan harus diperhatikan karena mempunyai tujuan untuk memberikan kemudahan dalam penggunaannya. Perancangan antarmuka menggunakan *wireframe* dimana hanya sebatas *User Interface* dan beberapa fitur yang menggambarkan secara umum kerja sistem.

Perancangan ini dilakukan berdasarkan *object-oriented analys* dengan menggunakan pemodelan UML (*Unified Modelling Language*). Perancangan dimulai dari perancangan arsitektur sistem, perancangan diagram *Sequence*, pdiagram *class*, perancangan basis data, perancangan kode program dan perancangan antarmuka beserta *screenflow*.

Setelah dilakukan *design solution*, maka selanjutnya adalah memberikan desain tersebut kepada calon pengguna untuk dilakukan evaluasi terhadap tampilan dan fitur yang ada. Hasil dari evaluasi akan digunakan untuk memperbaiki sistem dengan melakukan iterasi. Iterasi dilakukan dengan cara wawancara pada 5 pengguna. Pengguna diberikan perancangan antarmuka lalu pengguna memberikan evaluasi dari perancangan antarmuka tersebut. Iterasi kembali dilakukan mulai dari analisis kebutuhan dari calon pengguna.

Ketika mendapatkan analisis kebutuhan baru, akan dilakukan *design solution* lagi untuk memperbaiki fitur sesuai kebutuhan pengguna yang baru. Setelah selesai dengan *design solution*, maka akan dikembalikan ke calon pengguna untuk dilakukan evaluasi dengan cara melakukan wawancara.

3.4 Implementasi

Setelah melakukan evaluasi terhadap *design solution* langkah selanjutnya adalah implementasi. Tahapan ini merupakan tahapan dimana perancangan sistem diubah menjadi bahasa pemrograman untuk menghasilkan suatu bentuk sistem yang dapat digunakan oleh pengguna. Implementasi aplikasi ini menggunakan software Android Studio dan menggunakan FireBase untuk menyimpan data-data pengguna. Untuk melakukan implementasi antarmuka berdasarkan dari perancangan yang telah dievaluasi oleh calon pengguna.

Implementasi ini diawali dengan spesifikasi apa saja yang dibutuhkan dalam membangun sistem tersebut. Adapun sistem yang dibutuhkan adalah perangkat keras dan perangkat lunak. Setelah itu dijelaskan batasan-batasan masalah pada implementasi sistem tersebut.

Pada bab implementasi terdapat tiga alur yaitu implementasi *database*, implementasi kode program dan implementasi antarmuka. Pada implementasi *database* digunakan Firebase Realtime Database untuk menyimpan data-data yang dibutuhkan dalam pembangunan sistem. Pada implementasi kode program dijelaskan *method* penting dalam pembangunan sistem ini. Lalu yang terakhir

adalah implementasi antarmuka, menerapkan hasil dari desain dari proses perancangan lalu diterapkan dalam kode xml.

3.5 Pengujian

Pengujian dilakukan untuk mengetahui apakah implementasi sistem sudah sesuai dengan perancangan dan analisis kebutuhan, juga untuk mengetahui apakah ada *error* atau *bug* pada sistem. Didalam proses pengujian ada beberapa metode yang digunakan yaitu pengujian unit, pengujian validasi, pengujian *usability* dan pengujian *compatibility*.

1. Pengujian unit digunakan untuk menguji beberapa metode yang ada pada kelas yang didapatkan dari hasil perancangan. Pengujian ini dilakukan dengan menggunakan metode *white box*. Jenis pengujian yang dilakukan adalah dengan menggunakan *basis path testing* untuk menguji kode program berdasarkan algoritma pada setiap metode.
2. Pengujian validasi merupakan pengujian yang digunakan untuk menguji seluruh kebutuhan fungsional sistem apakah sudah sesuai dengan kebutuhan yang didapatkan sebelumnya. Pengujian ini dilakukan dengan metode *black box*.
3. Pengujian *usability* merupakan pengujian yang digunakan untuk mengukur tingkat pengalaman pengguna ketika menggunakan sistem ini. Pengujian yang akan dilakukan adalah efektivitas, efisiensi dan tingkat kepuasan pengguna ketika menggunakan sistem ini. Untuk mendapatkan data, maka dilakukan pembagian kuisioner kepada calon pengguna. Untuk soal dan hasil kuisioner dari pengujian *usability* terlampir pada halaman lampiran.
4. Pengujian *compatibility* merupakan pengujian yang digunakan untuk mengetahui apakah aplikasi sudah berjalan dengan baik di berbagai *device* pengguna dengan spesifikasi yang berbeda. Pengujian dilakukan pada lima perangkat dengan parameter sistem operasi android yang berbeda, resolusi layar dan fitur sensor gyroscope. Analisis dilakukan dengan uji coba menjalankan aplikasi pada perangkat android. Selanjutnya dilakukan perhitungan skor presentase hasil pengujian dan dicocokkan dengan skala penilaian untuk mengetahui tingkat kualitas aspek *compatibility* aplikasi.

Dengan dilakukannya analisis pengujian maka akan didapatkan apa saja kekurangan dari sistem dan mengetahui apa saja yang harus diperbaiki dalam sistem, sehingga hasil dari analisis pengujian akan dijadikan sebagai isi dari kesimpulan dan saran.

3.6 Pengambilan Kesimpulan dan Saran

Tahap ini adalah tahap akhir dari penelitian ini. Kesimpulan diambil dari hasil pengujian dan analisis. Dengan adanya kesimpulan diharapkan dapat menjawab perumusan permasalahan yang ada. Sedangkan saran digunakan untuk memperbaiki kesalahan yang terjadi pada penelitian serta menjadi pertimbangan atas penelitian selanjutnya.

BAB 4 REKAYASA KEBUTUHAN

4.1 Analisis Kebutuhan

Pada tahap analisis kebutuhan ini menurut metode *User Centered Design* mencakup alur *Specify the context of use* dan *Specify user and organizational requirement*. Tahap ini berisi tentang gambaran umum sistem, identifikasi aktor, analisis kebutuhan fungsional, *use case diagram* dan *use case scenario*. Untuk metode UCD akan dilakukan iterasi sehingga pada tahap analisis kebutuhan ini dibagi menjadi beberapa iterasi sesuai dengan hasil evaluasi yang dilakukan pada bab 5.

4.1.1 Analisis Kebutuhan Iterasi 1

4.1.1.1 Gambaran Umum Sistem

Roommate merupakan sebuah aplikasi berbasis android yang di desain untuk membantu pengguna mendapatkan informasi penginapan *low budget* seperti Hostel dan Dormitory. Adanya fitur untuk melihat tempat populer yang diurutkan berdasarkan kategori cuaca pada saat itu memudahkan pengguna untuk mencari tempat rekomendasi yang bisa didatangi.

Aplikasi ini dapat diakses oleh pengguna yang sudah mendaftarkan akun pada sistem. Informasi yang diberikan adalah nama penginapan, alamat penginapan, nomor telepon penginapan, website, fasilitas yang diberikan dan tipe kamar yang ada pada penginapan tersebut. Pengguna juga dapat melihat gambar panorama penginapan dengan dukungan *augmented reality*. Aplikasi ini juga dilengkapi fitur *maps* untuk mempermudah pengguna menemukan rute untuk menuju ke penginapan yang diinginkan. Pengguna juga dapat membagikan informasi penginapan tersebut melalui *social media* pribadi sehingga lebih banyak teman yang mengetahui tentang informasi penginapan tersebut.

4.1.1.2 Deskripsi Sistem

Berikut ini merupakan proses utama yang terdapat pada sistem informasi hostel di Kota Malang:

1. Melihat list hostel
Operasi ini bertujuan untuk membantu pengguna mengetahui daftar hostel di Kota Malang.
2. Melihat detail informasi hostel
Operasi ini bertujuan untuk melihat informasi detail dari hostel yang dipilih oleh pengguna.
3. Melihat *popular venue*
Operasi ini dilakukan ketika pengguna membuka sistem dan melihat cuaca saat ini di Kota Malang, serta melihat info *popular venue* berdasarkan kategori yang dikeluarkan.

4.1.1.3 Identifikasi Aktor

Pengguna dari aplikasi ini terdiri dari *member* dan *non-member*. *Member* dapat menikmati semua fitur yang diberikan seperti melihat informasi dan memberikan komentar serta rating. Untuk menjadi *member*, pengguna *non-member* harus melakukan registrasi terlebih dahulu untuk mendapatkan alamat email dan *password* sehingga dapat melakukan login pada aplikasi.

Tabel 4.1 Identifikasi Aktor

Nama Aktor	Deskripsi
<i>Member</i>	Pengguna yang dapat melihat informasi lengkap tentang list hostel yang dipilih, menulis komentar sampai memberikan rating pada hostel yang dipilih.
<i>Non-Member</i>	Pengguna hanya dapat melakukan registrasi untuk menjadi member.

4.1.1.4 Analisis Kebutuhan Fungsional

Proses untuk melakukan analisis kebutuhan dilakukan dengan cara wawancara langsung terhadap calon pengguna aplikasi. Dari hasil wawancara didapatkan suatu kebutuhan pengguna yang disebutkan pada Tabel 4.3 dan 4.4 yaitu kebutuhan fungsional pengguna.

Pada lampiran terdapat contoh hasil wawancara yang telah dilakukan pada narasumber. Berikut merupakan hasil rekap wawancara yang telah dilakukan:

Tabel 4.2 Rekap Kebutuhan Pengguna

Nama Kebutuhan	Jumlah Orang	%
1. Pengguna membutuhkan fitur informasi hostel?	Setuju: 5 Orang Tidak setuju: 0 Orang	100%
2. Pengguna membutuhkan fitur <i>Point Of Interest</i> ?	Setuju: 5 Orang Tidak setuju: 0 Orang	100%
3. Pengguna membutuhkan fitur memberikan rating terhadap hostel?	Setuju: 5 Orang Tidak setuju: 0 Orang	100%
4. Pengguna membutuhkan fitur memberikan komentar terhadap hostel?	Setuju: 5 Orang Tidak setuju: 0 Orang	100%
5. Pengguna membutuhkan fitur membagikan ke <i>media social</i> ?	Setuju: 5 Orang Tidak setuju: 0 Orang	100%

a. Pengguna: Non-Member

Tabel 4.3 Kebutuhan Fungsional Non-Member

No	Kode Fungsi	Nama Fungsi	Deskripsi
1	RM-F-01	Register	Pengguna dapat melakukan registrasi untuk menjadi member pada sistem ini.
2	RM-F-02	Login	Memberlakukan login dengan memasukan <i>username</i> dan <i>password</i> .

b. Pengguna: Member

Tabel 4.4 Kebutuhan Fungsional Member

No	Kode Fungsi	Nama Fungsi	Deskripsi
1	RM-F-03	Menambah Daftar Keinginan	Pengguna dapat menyimpan informasi hostel yang diinginkan kedalam <i>wishlist</i> .
2	RM-F-04	Melihat List Hostel	Pengguna dapat melihat list hostel yang direkomendasikan oleh sistem.
3	RM-F-05	Melakukan Filter	Pengguna dapat mengisikan <i>budget</i> untuk dilakukan filter terhadap list.
4	RM-F-06	Melihat Informasi	Pengguna melihat informasi lengkap tentang penginapan yang telah dipilih.
5	RM-F-07	Melihat Rute	Pengguna dapat melihat rute menuju lokasi hostel.
6	RM-F-08	Membagikan ke Media Sosial	Pengguna dapat membagikan informasi penginapan tersebut kepada teman melalui media social yang ada.
7	RM-F-09	Menulis Komentar	Pengguna dapat memberikan komentar untuk dibagikan sesama member lain.
8	RM-F-10	Melihat Komentar	Pengguna dapat melihat komentar yang diberikan oleh pengguna lain.
9	RM-F-11	Menghapus Komentar	Pengguna dapat menghapus komentar yang telah ditulis sebelumnya.
10	RM-F-12	Memberikan Rating	Pengguna dapat memberikan rating kepada penginapan tersebut.
11	RM-F-13	Melihat <i>Point Of Interest</i>	Pengguna dapat mengetahui rekomendasi tempat wisata populer menurut cuaca di Malang.
12	RM-F-14	Melihat Akun	Pengguna dapat melihat halaman akun.
13	RM-F-15	Melakukan Edit Profil	Pengguna dapat melakukan edit pada profil pada halaman akun.

No	Kode Fungsi	Nama Fungsi	Deskripsi
14	RM-F-16	Merubah Foto Profil	Pengguna dapat merubah foto profil.
15	RM-F-17	Logout	Pengguna memberlakukan logout untuk keluar dari aplikasi.

4.1.1.5 Analisis Kebutuhan Non-Fungsional

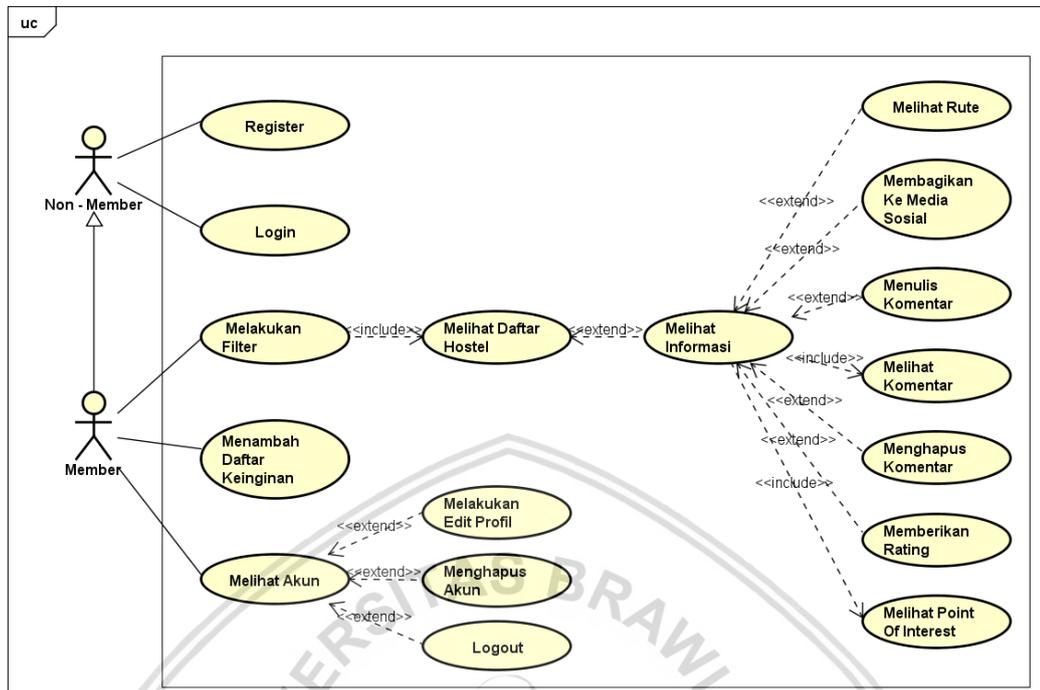
Tabel 4.5 Kebutuhan Non - Fungsional Member

No	Kode Fungsi	Nama Fungsi	Deskripsi
1	RM-NF-01	<i>Usability</i>	Sistem mempunyai tampilan yang mudah dimengerti untuk digunakan oleh semua orang.
2	RM-NF-02	<i>Compatibility</i>	Sistem dapat dijalankan di berbagai <i>device</i> dengan resolusi layar dan sistem tipe OS yang berbeda.

4.1.1.6 Analisis Data

Untuk membuat akun dan menjadi member pada aplikasi, maka data yang diperlukan adalah alamat email dan password. Alamat Email dan *password* diperlukan untuk melakukan login dan *authentication* pada sistem, adapun nama dan nomor telepon dapat diisikan untuk data tambahan pada akun. Untuk list hostel beserta informasi tentang hostel tersebut didapatkan dari API Foursquare dan dengan cara mencari pada *website* yang tersedia. Sedangkan info cuaca diambil dari Openweathermap API.

4.1.1.7 Use Case Diagram



Gambar 4.1 Diagram Use Case Iterasi Pertama

Fitur utama aplikasi ini adalah melihat list hostel yang ada di kota Malang. Pada setiap hostel terdapat informasi yang diberikan yaitu nama hostel, alamat hostel, nomor telepon hostel, website hostel, dan melihat gambar dari hostel. Pengguna juga dapat mengisi, menghapus dan melihat komentar yang ditujukan pada hostel.

4.1.1.8 Use Case Scenario

1. Use Case Scenario Register

Tabel 4.6 Use Case Scenario Register

Nama Use Case	Register
Kode Fungsi	RM-F-01
Aktor	Non-Member
Tujuan	Aktor dapat melakukan login pada aplikasi.
Deskripsi	Use case ini menjelaskan tentang bagaimana aktor akan melakukan registrasi.
Pre-condition	Aktor harus membuka aplikasi terlebih dahulu dan memilih pilihan registrasi.
Post-condition	Aktor melihat halaman login.
Alur Utama	

Aksi dari Aktor	Respons dari Sistem
<ol style="list-style-type: none"> 1. Aktor masuk ke halaman registrasi. 2. Aktor mengisikan alamat email dan password. 7. Aktor melihat halaman login. 	<ol style="list-style-type: none"> 8. Sistem menyimpan data ke dalam database. 9. Sistem melakukan autentikasi pada alamat email dan password. 10. Sistem menampilkan halaman login.

Pada tabel 4.6 merupakan gambaran *use case scenario* dari *register*. Pada *use case scenario register* sistem menyediakan *form* untuk mengisi identitas seperti alamat email dan *password*. Halaman ini hanya bisa diakses oleh pengunjung yang masih non-member. Alur dari skenario ini adalah pengunjung non-member mengisikan identitas kedalam form yang telah disediakan lalu sistem akan menyimpan data tersebut dan menampilkan halaman login. Kondisi setelahnya adalah pengunjung sudah menjadi member dan dapat login langsung kedalam sistem dengan alamat email dan *password*.

2. Use Case Scenario Login

Tabel 4.7 Use Case Scenario Login

Nama Use Case	Login	
Kode Fungsi	RM-F-02	
Aktor	Non – Member	
Tujuan	Aktor dapat melihat halaman menu utama aplikasi.	
Deskripsi	Use case ini menjelaskan tentang bagaimana aktor akan melakukan <i>login</i> setelah registrasi.	
Pre-condition	Aktor telah melakukan registrasi dan sudah masuk halaman login.	
Post-condition	Aktor melihat menu utama.	
Alur Utama		
	Aksi dari Aktor	Respons dari Sistem
	<ol style="list-style-type: none"> 1. Aktor masuk ke halaman login. 2. Aktor mengisikan alamat email dan password. 6. Aktor melihat halaman menu utama. 	<ol style="list-style-type: none"> 3. Sistem melakukan autentikasi pada alamat email dan password. 4. Sistem menampilkan halaman menu utama.
Alur Alternatif		
	Aksi dari Aktor	Respons dari Sistem



<p>8. Aktor mengisikan kembali alamat email dan password.</p> <p>11. Aktor melihat halaman menu utama.</p>	<p>7. Sistem menampilkan peringatan jika proses autentikasi gagal.</p> <p>9. Sistem melakukan autentikasi kembali.</p> <p>10. Sistem menampilkan halaman menu utama.</p>
--	--

Pada tabel 4.7 merupakan *use case scenario* dari *login*. Pada *use case scenario* login, sistem menyediakan halaman login yang berisi form untuk mengisikan *username* dan *password*. Alurnya adalah dengan pengunjung yang sudah menjadi member memasukkan *username* dan *password* kedalam form yang telah disediakan, lalu sistem akan melakukan validasi terlebih dahulu terhadap masukkan, jika kombinasi tersebut benar maka sistem akan menampilkan halaman utama dari aplikasi. Pada fitur login ini terdapat alur alternative yaitu jika member salah memasukkan *username* atau *password* maka sistem akan menampilkan halaman login kembali.

3. *Use Case Scenario* Menambah Daftar Keinginan

Tabel 4.8 Use Case Scenario Menambah Daftar Keinginan

Nama Use Case	Menambah Daftar Keinginan	
Kode Fungsi	RM-F-03	
Aktor	Member	
Tujuan	Aktor dapat menyimpan informasi hostel kedalam <i>wishlist</i> .	
Deskripsi	<i>Use case</i> ini akan menjelaskan tentang bagaimana aktor akan menyimpan hostel yang dipilih kedalam halaman <i>wishlist</i> .	
<i>Pre-condition</i>	Aktor memilih hostel dan membuka detail informasi hostel tersebut.	
<i>Post-condition</i>	Aktor akan menemukan list tersebut dalam halaman <i>wishlist</i> .	
Aliran Utama		
Aksi dari Aktor	Respons dari Sistem	
<p>1. Aktor memilih hostel.</p> <p>2. Aktor memilih icon <i>wishlist</i> pada halaman detail.</p> <p>6. Aktor melihat list yang disimpan pada halaman <i>wishlist</i>.</p>	<p>3. Sistem menampilkan halaman detail dengan informasi hostel tersebut.</p> <p>4. Sistem menyimpan informasi hostel ke dalam database.</p> <p>5. Sistem menampilkan list tersebut pada halaman <i>wishlist</i>.</p>	

Pada tabel 4.8 merupakan *use case scenario* menambah daftar keinginan. Pengguna yang dapat mengakses fitur ini adalah pengguna yang telah menjadi



member. Alurnya adalah pengguna dalam halaman informasi hostel, lalu pengguna memilih ikon *wishlist*, lalu sistem akan menyimpan informasi tersebut dalam database *wishlist* dan pengguna dapat melihat list hostel yang disimpan dalam halaman *wishlist*. Kondisi setelahnya adalah pengguna dapat menemukan informasi hostel yang disimpan pada halaman *wishlist* tanpa mencari dari pertama.

4. *Use Case Scenario* Melihat List Hostel

Tabel 4.9 Use Case Scenario Melihat List Hostel

Nama Use Case	Melihat List Hostel
Kode Fungsi	RM-F-04
Aktor	Member
Tujuan	Aktor dapat melihat semua informasi tentang hostel.
Deskripsi	Use case ini akan menjelaskan tentang bagaimana aktor bisa melihat informasi tentang hostel.
Pre-condition	Aktor memilih hostel pada list.
Post-condition	Aktor melihat detail informasi hostel tersebut.
Aliran Utama	
Aksi dari Aktor	Respons dari Sistem
3. Aktor memilih hostel. 4. Aktor memilih icon <i>wishlist</i> pada halaman detail. 7. Aktor melihat list yang disimpan pada halaman <i>wishlist</i> .	6. Sistem menampilkan halaman detail dengan informasi hostel tersebut. 7. Sistem menyimpan informasi hostel ke dalam database. 8. Sistem menampilkan list tersebut pada halaman <i>wishlist</i> .

Pada tabel 4.9 merupakan *use case scenario* lihat list hostel. Pada *use case scenario* ini sistem menyediakan halaman list hostel yang dapat dilihat oleh member. Alurnya adalah sistem akan menampilkan list hostel. Kondisi setelahnya adalah aktor dapat melihat list hostel.

5. *Use Case Scenario* Melakukan Filter

Tabel 4.10 Use Case Scenario Melakukan Filter

Nama Use Case	Melakukan Filter
Kode Fungsi	RM-F-05
Aktor	Member



Tujuan	Aktor dapat melihat list hostel dengan harga yang sudah ditentukan.	
Deskripsi	<i>Use case</i> ini akan menjelaskan tentang bagaimana aktor bisa melakukan filter terhadap list hostel.	
<i>Pre-condition</i>	Aktor berada pada halaman filter.	
<i>Post-condition</i>	Aktor melihat list hostel sesuai filter.	
Aliran Utama		
	Aksi dari Aktor	Respons dari Sistem
	1. Aktor mengisi harga untuk dilakukan filter.	2. Sistem menampilkan list hostel berdasarkan harga maksimum yang diisikan oleh aktor.

Pada tabel 4.10 merupakan *use case scenario* dari melakukan filter, hanya member yang dapat melakukan fungsi tersebut. Alurnya adalah aktor mengisi harga maksimal hostel yang diinginkan, lalu sistem akan menampilkan list hostel yang sudah dilakukan filter.

6. Use Case Scenario Melihat Informasi

Tabel 4.11 Use Case Scenario Melihat Informasi

Nama Use Case	Melihat Informasi	
Kode Fungsi	RM-F-06	
Aktor	Member	
Tujuan	Aktor dapat melihat semua informasi tentang hostel.	
Deskripsi	<i>Use case</i> ini akan menjelaskan tentang bagaimana aktor bisa melihat informasi tentang hostel.	
<i>Pre-condition</i>	Aktor memilih hostel pada list.	
<i>Post-condition</i>	Aktor melihat detail informasi hostel tersebut.	
Aliran Utama		
	Aksi dari Aktor	Respons dari Sistem
	1. Aktor memilih hostel dari list.	2. Sistem mengambil data informasi dari database sesuai dengan nama hostel yang dipilih. 3. Sistem menampilkan data informasi ke halaman detail.

Pada tabel 4.11 merupakan *use case scenario* melihat informasi. Hanya member yang dapat mengakses halaman ini. Alurnya adalah pengguna memilih hostel yang ada pada list, lalu sistem akan menampilkan informasi lengkap dari



hostel tersebut seperti alamat, *contact person* sampai fasilitas yang ada pada hostel tersebut.

7. *Use Case Scenario* Melihat Rute

Tabel 4.12 Use Case Scenario Melihat Rute

Nama <i>Use Case</i>	Melihat Rute
Kode Fungsi	RM-F-07
Aktor	Member
Tujuan	Aktor dapat melihat rute menuju lokasi hostel.
Deskripsi	<i>Use case</i> ini akan menjelaskan tentang bagaimana aktor bisa melihat rute menuju lokasi hostel.
<i>Pre-condition</i>	Aktor berada pada halaman detail hostel.
<i>Post-condition</i>	Aktor melihat rute menuju hostel tersebut.
Aliran Utama	
Aksi dari Aktor	Respons dari Sistem
1. Aktor memilih pilihan “ <i>See Maps</i> ” yang ada pada halaman detail.	2. Sistem akan mengambil data alamat dari hostel. 3. Sistem akan menampilkan <i>Google Maps</i> dengan rute menuju hostel yang dipilih.

Pada tabel 4.12 merupakan *use case scenario* melihat rute. Hanya member yang dapat mengakses halaman ini. Alurnya adalah setelah memilih hostel dari list yang sudah disediakan sistem, sistem akan menampilkan informasi dari hostel dan peta, ketika aktor memilih ikon lihat rute maka sistem akan menampilkan peta dan rute untuk menuju lokasi hostel tersebut.

8. *Use Case Scenario* Membagikan ke Media Sosial

Tabel 4.13 Use Case Scenario Membagikan ke Media Sosial

Nama <i>Use Case</i>	Membagikan ke Media Sosial
Kode Fungsi	RM-F-08
Aktor	Member
Tujuan	Aktor dapat membagikan aplikasi melalui media social.
Deskripsi	<i>Use case</i> ini akan menjelaskan tentang bagaimana aktor bisa membagikan aplikasi ke media social.
<i>Pre-condition</i>	Aktor berada pada halaman detail hostel.
<i>Post-condition</i>	Aktor berhasil membagikan aplikasi ke social media.

Aliran Utama	
Aksi dari Aktor	Respons dari Sistem
<ol style="list-style-type: none"> 1. Aktor memilih icon share pada bagian atas halaman detail. 3. Aktor memilih aplikasi social media yang dituju. 5. Aktor mengirim aplikasi. 	<ol style="list-style-type: none"> 2. Sistem akan menampilkan pilihan aplikasi social media. 4. Sistem membuka aplikasi social media yang dipilih oleh aktor.

Pada tabel 4.13 merupakan *use case scenario* membagikan ke media sosial. Pengguna yang dapat mengakses fitur ini adalah pengguna yang telah menjadi member. Alurnya adalah pengguna memilih social media yang diinginkan, sistem akan membuka social media tersebut lalu pengguna memilih bagian atau *share*. Kondisi setelahnya adalah teman dari *social media* pengguna dapat mengetahui informasi yang telah dibagikan.

9. *Use Case Scenario* Menulis Komentar

Tabel 4.14 Use Case Scenario Menulis Komentar

Nama <i>Use Case</i>	Menulis Komentar
Kode Fungsi	RM-F-09
Aktor	Member
Tujuan	Aktor dapat memberikan komentar untuk dibagikan dengan member lain.
Deskripsi	<i>Use case</i> ini akan menjelaskan tentang bagaimana aktor bisa menulis komentar untuk hostel.
<i>Pre-condition</i>	Aktor berada pada halaman detail hostel.
<i>Post-condition</i>	Aktor mengisi komentar untuk hostel.
Aliran Utama	
Aksi dari Aktor	Respons dari Sistem
<ol style="list-style-type: none"> 1. Aktor berada pada halaman detail hostel. 2. Aktor mengisi komentar pada kolom komentar yang sudah disediakan. 	<ol style="list-style-type: none"> 3. Sistem menyimpan komentar ke dalam database.

Pada tabel 4.14 merupakan *use case scenario* menulis komentar pada forum. Pengguna yang dapat mengakses fitur ini adalah pengguna yang telah menjadi member. Alurnya adalah pengguna mengisi komentar pada form yang sudah disediakan, sistem akan menyimpan ke dalam database lalu sistem akan



menampilkan komentar tersebut pada halaman forum. Kondisi setelahnya adalah pengguna lain dapat melihat komentar yang telah ditulis pada halaman forum.

10. *Use Case Scenario* Melihat Komentar

Tabel 4.15 *Use Case Scenario* Melihat Komentar

Nama <i>Use Case</i>	Melihat Komentar
Kode Fungsi	RM-F-10
Aktor	Member
Tujuan	Aktor dapat melihat komentar yang dikirimkan oleh pengguna lain.
Deskripsi	<i>Use case</i> ini akan menjelaskan tentang bagaimana aktor bisa melihat komentar yang telah dibuat oleh pengguna lain.
<i>Pre-condition</i>	Aktor berada pada halaman detail hostel.
<i>Post-condition</i>	Aktor melihat komentar untuk hostel.
Aliran Utama	
Aksi dari Aktor	Respons dari Sistem
<ol style="list-style-type: none"> 1. Aktor berada pada halaman detail hostel. 3. Aktor melihat komentar pada bagian bawah halaman detail. 	<ol style="list-style-type: none"> 2. Sistem menampilkan informasi lengkap hostel beserta semua komentar yang ada pada database.

Pada tabel 4.15 merupakan *use case scenario* melihat komentar. Alur utama dari scenario ini mulai dari pengguna membuka halaman detail dari hostel yang dipilih, lalu pada bagian bawah halaman langsung dapat melihat komentar-komentar yang telah diisikan oleh pengguna lainnya.

11. *Use Case Scenario* Menghapus Komentar

Tabel 4.16 *Use Case Scenario* Menghapus Komentar

Nama <i>Use Case</i>	Menghapus Komentar
Kode Fungsi	RM-F-11
Aktor	Member
Tujuan	Aktor dapat menghapus komentar yang pernah ditulis.
Deskripsi	<i>Use case</i> ini akan menjelaskan tentang bagaimana aktor bisa menghapus komentar pada halaman detail.
<i>Pre-condition</i>	Aktor berada pada halaman detail hostel.
<i>Post-condition</i>	Aktor menghapus komentar yang telah diisikan.



Aliran Utama	
Aksi dari Aktor	Respons dari Sistem
<ol style="list-style-type: none"> 1. Aktor berada pada halaman detail hostel. 2. Aktor menghapus komentar yang pernah diisi. 	<ol style="list-style-type: none"> 3. Sistem menghapus komentar pada database. 4. Sistem menampilkan komentar tanpa komentar yang telah dihapus.

Pada tabel 4.16 merupakan *use case scenario* menghapus komentar. Alur utama dari scenario ini mulai dari pengguna membuka halaman detail dari hostel yang dipilih, lalu mencari komentar milik pengguna tersebut kemudian menghapus dengan cara klik pada tombol *delete* yang tersedia.

12. Use Case Scenario Memberikan Rating

Tabel 4.17 Use Case Scenario Memberikan Rating

Nama Use Case	Memberikan Rating
Kode Fungsi	RM-F-12
Aktor	Member
Tujuan	Aktor dapat memberikan rating pada hostel.
Deskripsi	Use case ini akan menjelaskan tentang bagaimana aktor bisa memberikan rating pada hostel yang dipilih.
Pre-condition	Aktor berada pada halaman detail hostel.
Post-condition	Aktor melihat halaman detail hostel dengan rating yang sudah diakumulasikan.
Aliran Utama	
Aksi dari Aktor	Respons dari Sistem
<ol style="list-style-type: none"> 1. Aktor membuka halaman detail hostel. 2. Aktor mengisikan rating 1-5 bintang. 	<ol style="list-style-type: none"> 3. Sistem menyimpan hasil rating ke dalam database. 4. Sistem menampilkan hasil rating yang sudah diakumulasikan kedalam halaman detail.

Pada tabel 4.17 merupakan *use case scenario* dari memberikan rating. Pengguna yang dapat memberikan rating hanyalah pengguna yang telah terdaftar menjadi member. Alurnya adalah aktor memberikan rating pada hostel dari pengalaman menginap lalu sistem akan menyimpan rating tersebut dan menjadikan *summary* rating dari beberapa orang. Konsisi setelahnya adalah setiap pengguna yang melihat informasi hostel tersebut akan mengetahui *summary* rating yang telah dibuat oleh pengguna lainnya.



13. Use Case Scenario Point of Interest

Tabel 4.18 Use Case Scenario Melihat Point of Interest

Nama Use Case	Melihat Point Of Interest
Kode Fungsi	RM-F-13
Aktor	Member
Tujuan	Aktor dapat melihat cuaca saat ini serta rekomendasi tempat populer yang ada di kota malang berdasarkan rating pengunjung.
Deskripsi	Use case ini akan menjelaskan tentang bagaimana aktor bisa melihat Point Of Interest beserta cuaca saat ini.
Pre-condition	Aktor berada pada halaman menu utama.
Post-condition	Aktor mengetahui Point Of Interest beserta cuaca.
Aliran Utama	
Aksi dari Aktor	Respons dari Sistem
1. Aktor memilih menu POI pada halaman menu utama aplikasi.	2. Sistem menampilkan data cuaca dan rekomendasi tempat populer di kota malang.

Pada tabel 4.18 merupakan *use case scenario point of interest*. Hanya pengguna yang terdaftar menjadi member saja yang bisa melihat fitur ini. Alurnya adalah ketika pengguna melihat informasi hostel yang dipilih, maka sistem akan menampilkan halaman informasi hostel beserta *point of interest* beserta cuaca yang ada pada kota malang.

14. Use Case Scenario Melihat Akun

Tabel 4.19 Use Case Scenario Melihat Akun

Nama Use Case	Melihat Akun
Kode Fungsi	RM-F-14
Aktor	Member
Tujuan	Aktor dapat melihat halaman akun.
Deskripsi	Use case ini akan menjelaskan tentang bagaimana aktor bisa melihat akun yang berisi data pengguna.
Pre-condition	Aktor berada pada halaman menu utama.
Post-condition	Aktor melihat halaman akun.
Aliran Utama	
Aksi dari Aktor	Respons dari Sistem

1. Aktor berada pada halaman menu utama.	3. Sistem menampilkan halaman akun.
2. Aktor memilih menu "account".	

Pada tabel 4.19 merupakan *use case scenario* melihat akun. Pengguna yang dapat melihat akun profil adalah pengguna yang telah terdaftar menjadi member. Alurnya adalah memilih ikon akun pada bagian bawah aplikasi lalu sistem akan langsung menampilkan halaman akun profil dari pengguna. Pada halaman akun ini pengguna dapat melihat nama, nomor telepon dan alamat email yang terdaftar. Serta pengguna dapat melakukan edit profil dan keluar dari aplikasi.

15. Use Case Scenario Melakukan Edit Profil

Tabel 4.20 Use Case Scenario Melakukan Edit Profil

Nama Use Case	Melakukan Edit Profil	
Kode Fungsi	RM-F-15	
Aktor	Member	
Tujuan	Aktor dapat melakukan edit nama dan nomor telepon pada akun.	
Deskripsi	Use case ini akan menjelaskan tentang bagaimana aktor bisa mengubah nama dan nomor telepon yang telah terdaftar pada database.	
Pre-condition	Aktor berada pada halaman akun.	
Post-condition	Aktor merubah profil.	
Aliran Utama		
	Aksi dari Aktor	Respons dari Sistem
	<ol style="list-style-type: none"> 1. Aktor berada pada halaman akun. 2. Aktor memilih pilihan edit profil pada bagian bawah halaman. 4. Aktor mengisi <i>field</i> nama dan nomor telepon dengan data baru. 8. Aktor melihat nama dan nomor telepon baru pada halaman akun. 	<ol style="list-style-type: none"> 3. Sistem akan menampilkan halaman edit profil. 5. Sistem menyimpan data baru dan merubah yang ada pada database. 6. Sistem menampilkan halaman akun. 7. Sistem menampilkan data baru yang diambil dari database.

Pada tabel 4.20 berisi *use case scenario* melakukan edit profil. Pengguna yang dapat melakukan edit profil hanyalah pengguna yang sudah menjadi member. Alurnya adalah ketika aktor berada pada halaman akun, aktor memilih edit profil lalu aktor merubah nama dan nomor telepon lalu sistem akan menyimpannya pada database.



16. Use Case Scenario Merubah Foto Profil

Tabel 4.21 Use Case Scenario Merubah Foto Profil

Nama Use Case	Merubah foto profil
Kode Fungsi	RM-F-16
Aktor	Member
Tujuan	Aktor dapat merubah foto profil.
Deskripsi	Use case ini akan menjelaskan tentang bagaimana aktor bisa merubah foto profil.
Pre-condition	Aktor berada pada halaman edit profil.
Post-condition	Aktor merubah foto profil.
Aliran Utama	
Aksi dari Aktor	Respons dari Sistem
<ol style="list-style-type: none"> 1. Aktor berada pada halaman edit profil. 2. Aktor memilih pilihan edit foto profil. 3. Aktor memilih foto profil baru. 6. Aktor melihat halaman akun dengan foto profil yang telah diperbarui. 	<ol style="list-style-type: none"> 4. Sistem akan menyimpan foto pada database. 5. Sistem menampilkan foto terbaru pada halaman akun.

Pada tabel 4.21 berisi *use case scenario* merubah foto profil. Pengguna yang dapat melakukan ubah foto profil hanyalah pengguna yang sudah menjadi member. Alurnya adalah ketika aktor berada pada halaman akun, aktor memilih edit profil lalu aktor memilih pilihan untuk edit foto profil. Setelah pengguna memilih foto baru, maka sistem akan merubah data yang ada pada database lalu akan ditampilkan kedalam halaman akun.

17. Use Case Scenario Logout

Tabel 4.22 Use Case Scenario Logout

Nama Use Case	Logout
Kode Fungsi	RM-F-17
Aktor	Member
Tujuan	Aktor dapat keluar dari aplikasi.
Deskripsi	Use case ini akan menjelaskan tentang bagaimana aktor bisa keluar dari aplikasi.

<i>Pre-condition</i>	Aktor berada pada halaman akun.
<i>Post-condition</i>	Aktor kembali pada halaman login.
Aliran Utama	
Aksi dari Aktor	Respons dari Sistem
<ol style="list-style-type: none"> 1. Aktor berada pada halaman akun. 2. Aktor memilih tombol logout. 5. Aktor melihat halaman login. 	<ol style="list-style-type: none"> 3. Sistem menghentikan <i>session</i> aplikasi. 4. Sistem menampilkan halaman login setelah logout berhasil.

Pada tabel 4.22 berisi *use case scenario* logout. Pengguna yang dapat melakukan logout hanyalah pengguna yang sudah menjadi member. Alurnya adalah ketika aktor berada pada halaman akun, aktor memilih tombol logout lalu sistem akan menghentikan *session* aplikasi dan menampilkan halaman login.

4.1.2 Analisis Kebutuhan Iterasi 2

4.1.2.1 Analisis Kebutuhan Fungsional

Pada Iterasi pertama, dilakukan survey terhadap keinginan calon pengguna dengan metode quisioner kepada pada calon pengguna. Calon pengguna diminta untuk memberikan evaluasi dari tampilan dan fitur aplikasi. Hasil dari quisioner adalah jika fitur edit foto profil akan dihapus, melihat dan menghapus komentar juga tidak diperlukan, dan dihapusnya fitur untuk melakukan filter terlebih dahulu. Selain itu ada penambahan beberapa fitur untuk melengkapi sistem antara lain melihat rating hostel, menghapus akun dan menghubungi pihak hostel. Untuk hasil wawancara analisis kebutuhan iterasi pertama dapat dilihat pada halaman lampiran. Berikut perubahan dari hasil quisioner:

Tabel 4.23 Kebutuhan Fungsional Member Iterasi Kedua

No	Kode Fungsi	Nama Fungsi	Deskripsi
16	RM-F-20	Melihat Rating Hostel	Pengguna dapat melihat rating hostel menurut pengguna aplikasi.
17	RM-F-21	Menghapus Akun	Pengguna dapat menghapus akun yang telah dibuat.
18	RM-F-22	Menghubungi Pihak Hostel	Pengguna dapat tersambung melalui telepon dan aplikasi <i>whatsapp</i> dengan pihak hostel.

4.1.2.2 Use Case Diagram

Setelah dilakukan Iterasi yang pertama, adanya hasil dari quisioner evaluasi yang diberikan oleh peneliti yang membuat perubahan pada *use case diagram*.

Tabel 4.24 Use Case Scenario Melihat Rating Hostel

Nama Use Case	Melihat Rating Hostel
Kode Fungsi	RM-F-20
Aktor	Member
Tujuan	Aktor dapat melihat rating dari setiap hostel yang dipilih.
Deskripsi	Use case ini akan menjelaskan tentang bagaimana aktor bisa melihat rating dari hostel.
Pre-condition	Aktor berada pada halaman detail hostel.
Post-condition	Aktor mengetahui rating hostel.
Aliran Utama	
Aksi dari Aktor	Respons dari Sistem
1. Aktor berada pada halaman detail hostel. 3. Aktor melihat rating hostel.	2. Sistem menampilkan rating hostel yang sudah tersimpan pada database.

Pada tabel 4.24 berisi *use case scenario* melihat rating hostel. Pengguna yang dapat melihat rating hostel hanyalah pengguna yang sudah menjadi member. Alurnya adalah ketika aktor berada pada halaman detail informasi, aktor akan langsung melihat rating dari hostel tersebut.

2. Use Case Scenario Menghapus Akun

Tabel 4.25 Use Case Scenario Menghapus Akun

Nama Use Case	Menghapus Akun
Kode Fungsi	RM-F-21
Aktor	Member
Tujuan	Aktor dapat menghapus akun yang sudah terdaftar kedalam database.
Deskripsi	Use case ini akan menjelaskan tentang bagaimana aktor bisa menghapus akun yang telah dibuat.
Pre-condition	Aktor berada pada halaman akun.
Post-condition	Akun telah berhasil terhapus dari database.
Aliran Utama	
Aksi dari Aktor	Respons dari Sistem
1. Aktor berada pada halaman edit profil.	

<ul style="list-style-type: none"> 2. Aktor memilih tombol hapus akun. 5. Aktor kembali pada halaman login. 	<ul style="list-style-type: none"> 3. Sistem menghapus data akun pada database. 4. Sistem akan menampilkan halaman login.
---	---

Pada tabel 4.25 berisi use case scenario menghapus akun. Pengguna yang dapat melakukan fitur ini hanyalah pengguna yang sudah menjadi member. Alurnya adalah ketika aktor berada pada halaman akun, aktor memilih halaman edit profil lalu memilih tombol hapus akun. Sistem akan menghapus data akun pada database dan menampilkan halaman login kembali.

3. Use Case Scenario Menghubungi Pihak Hostel

Tabel 4.26 Use Case Scenario Menghubungi Pihak Hostel

Nama Use Case	Menghubungi Pihak Hostel	
Kode Fungsi	RM-F-22	
Aktor	Member	
Tujuan	Aktor dapat menghubungi pihak hostel melalui telepon atau aplikasi whatsapp.	
Deskripsi	Use case ini akan menjelaskan tentang bagaimana aktor bisa menghubungi pihak hostel secara langsung.	
<i>Pre-condition</i>	Aktor berada pada halaman detail informasi.	
<i>Post-condition</i>	Aktor menghubungi pihak hostel.	
Aliran Utama		
Aksi dari Aktor	Respons dari Sistem	
<ul style="list-style-type: none"> 1. Aktor berada pada halaman detail informasi hostel. 2. Aktor memilih icon telepon untuk menghubungi pihak hostel melalui telepon. 	<ul style="list-style-type: none"> 3. Sistem akan menyambungkan pengguna dengan nomor telepon pihak hostel. 4. Sistem akan menampilkan aplikasi telepon bawaan dari ponsel. 	
Aliran Alternatif		
Aksi dari Aktor	Respons dari Sistem	
<ul style="list-style-type: none"> 5. Aktor memilih icon whatsapp untuk menghubungi pihak hostel melalui aplikasi whatsapp. 	<ul style="list-style-type: none"> 6. Sistem akan menghubungkan aktor dengan nomor pihak hostel. 	



	7. Sistem akan menampilkan aplikasi whatsapp beserta nomor tujuan.
--	--

Pada tabel 4.26 berisi use case scenario menghubungi pihak hostel. Hanya aktor member yang bisa menggunakan fitur ini. Alurnya adalah aktor akan memilih akan menghubungi pihak hostel melalui telepon atau aplikasi whatsapp. Lalu sistem akan menyambungkan langsung dengan nomor pihak hostel melalui aplikasi telepon atau whatsapp.

4.1.3 Analisis Kebutuhan Iterasi 3

4.1.3.1 Analisis Kebutuhan Fungsional

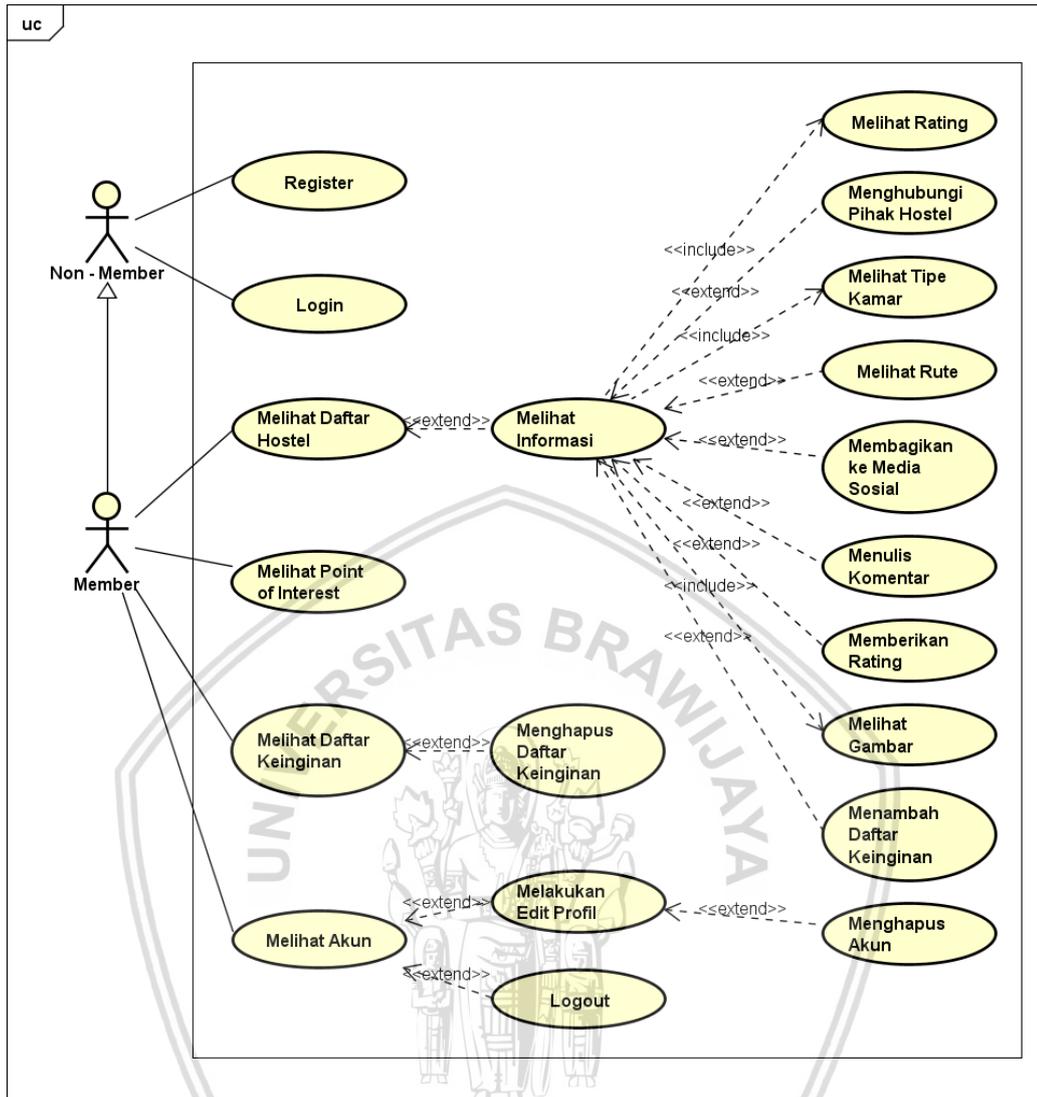
Pada iterasi kedua, dilakukan survey terhadap keinginan calon pengguna dengan metode kuisisioner. Pertanyaan dari kuisisioner dan tugas scenario yang sudah diberikan kepada pengguna telah dilampirkan pada halaman lampiran. Hasil dari kuisisioner adalah adanya penambahan fitur hapus daftar dari *wishlist* yang telah disimpan pengguna serta penambahan foto 360°. Selbihnya, calon pengguna telah menyetujui fitur-fitur pada aplikasi ini. Berikut adalah tabel kebutuhan fungsional dari hasil kuisisioner:

Tabel 4.27 Kebutuhan Fungsional Member Iterasi Ketiga

No	Kode Fungsi	Nama Fungsi	Deskripsi
19	RM-F-23	Menghapus Daftar Keinginan	Pengguna dapat menghapus daftar keinginan yang telah disimpan.
20	RM-F-24	Melihat Gambar	Pengguna dapat melihat foto 360° berisi interior dalam hostel.

4.1.3.2 Use Case Diagram

Setelah dilakukan iterasi yang kedua, adanya perubahan setelah dibagikan kuisisioner yang harus diisikan oleh pengguna sesuai dengan *task scenario* yang diberikan oleh observer. Hasil dari iterasi tersebut telah dilampirkan pada halaman lampiran. Adapun perubahan dari use case diagram adalah sebagai berikut:



Gambar 4.3 Diagram Use Case Iterasi Ketiga

4.1.3.3 Use Case Scenario

Dari hasil iterasi kedua, didapatkan penambahan fitur yaitu menghapus daftar keinginan dan melihat galeri foto 360° pada aplikasi. Adapun tambahan use case scenario adalah sebagai berikut:

1. Use Case Scenario Menghapus Daftar Keinginan

Tabel 4.28 Use Case Scenario Menghapus Daftar Keinginan

Nama Use Case	Menghapus daftar keinginan
Kode Fungsi	RM-F-23
Aktor	Member
Tujuan	Member dapat menghapus daftar keinginan yang telah disimpan.



Deskripsi	<i>Use case</i> ini akan menjelaskan tentang bagaimana <i>member</i> bisa menghapus daftar keinginan yang telah tersimpan pada database melalui sistem.	
<i>Pre-condition</i>	<i>Member</i> berada pada halaman <i>wishlist</i> .	
<i>Post-condition</i>	Daftar keinginan sudah terhapus dari database dan sistem.	
Aliran Utama		
	Aksi dari Aktor	Respons dari Sistem
	<ol style="list-style-type: none"> 1. <i>Member</i> berada pada halaman <i>wishlist</i>. 2. <i>Member</i> memilih daftar hostel mana yang akan dihapus. 3. <i>Member</i> memilih tombol hapus pada bagian kanan informasi hostel. 	<ol style="list-style-type: none"> 4. Sistem menghapus daftar pada <i>database</i>. 5. Sistem menampilkan daftar baru pada halaman.

Pada tabel 4.28 berisi *use case scenario* menghapus daftar *wishlist* pada sistem. Pengguna yang dapat menghapus hanyalah pengguna yang sudah menjadi *member*. Alurnya adalah ketika aktor berada pada halaman *wishlist*, aktor akan memilih daftar hostel mana yang akan dihapus, kemudian aktor akan memilih tombol hapus pada sisi kanan daftar maka sistem akan langsung menghapus daftar tersebut pada database dan menampilkan halaman *wishlist* yang baru.

2. *Use Case Scenario* Melihat Gambar

Tabel 4.29 Use Case Scenario Melihat Gambar

Nama <i>Use Case</i>	Melihat Gambar	
Kode Fungsi	RM-F-24	
Aktor	<i>Member</i>	
Tujuan	<i>Member</i> dapat melihat gambar foto 360° interior dari hostel.	
Deskripsi	<i>Use case</i> ini akan menjelaskan tentang bagaimana <i>member</i> dapat melihat gambar foto dari interior hostel.	
<i>Pre-condition</i>	<i>Member</i> berada pada halaman detail hostel.	
<i>Post-condition</i>	<i>Member</i> melihat foto 360° hostel.	
Aliran Utama		
	Aksi dari Aktor	Respons dari Sistem
	<ol style="list-style-type: none"> 1. <i>Member</i> berada pada halaman detail hostel. 	



	2. Sistem menampilkan informasi detail hostel.
--	--

Pada tabel 4.29 berisi *use case scenario* melihat gambar foto 360°. Pengguna yang dapat melihat hanyalah pengguna yang sudah menjadi member. Alurnya adalah ketika pengguna sedang dalam halaman detail hostel, pengguna akan langsung melihat gambar 360° yang berada pada bagian atas halaman.



BAB 5 PERANCANGAN DAN IMPLEMENTASI

5.1 Perancangan

Pada tahap perancangan ini menurut metode *User Centered Design* mencakup alur *Product design solution* dan *Evaluate design against user requirement*. Tahap ini berisi tentang perancangan sistem, perancangan perangkat lunak serta hasil evaluasi dari kebutuhan pengguna. Untuk metode UCD akan dilakukan iterasi sehingga pada tahap perancangan ini dibagi menjadi beberapa iterasi sesuai dengan hasil evaluasi dan hasil kebutuhan pengguna yang mengacu pada bab 4.

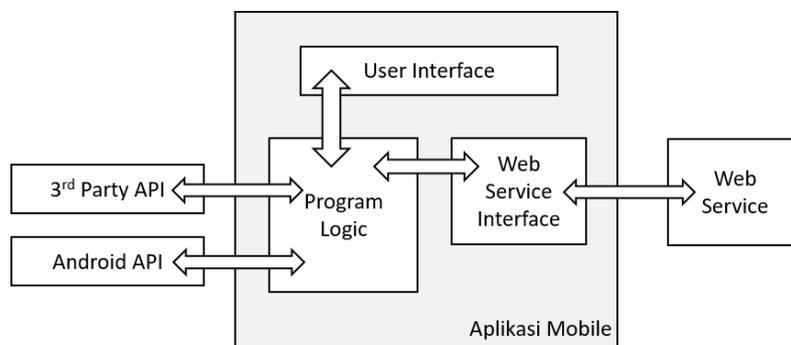
5.1.1 Perancangan Sistem

5.1.1.1 Perancangan Arsitektur Sistem

Perancangan arsitektur ini dibangun menggunakan konsep *mobile application* pada *platform mobile* dengan sistem operasi Android. Konsep ini dipilih karena dapat menggunakan fitur yang dimiliki oleh perangkat *smartphone*. Untuk database menggunakan *realtime database* yaitu Firebase. Data yang disimpan mencakup data pengguna dan data hostel serta komentar. Perancangan arsitektur sistem digambarkan pada Gambar 5.1 dibawah.



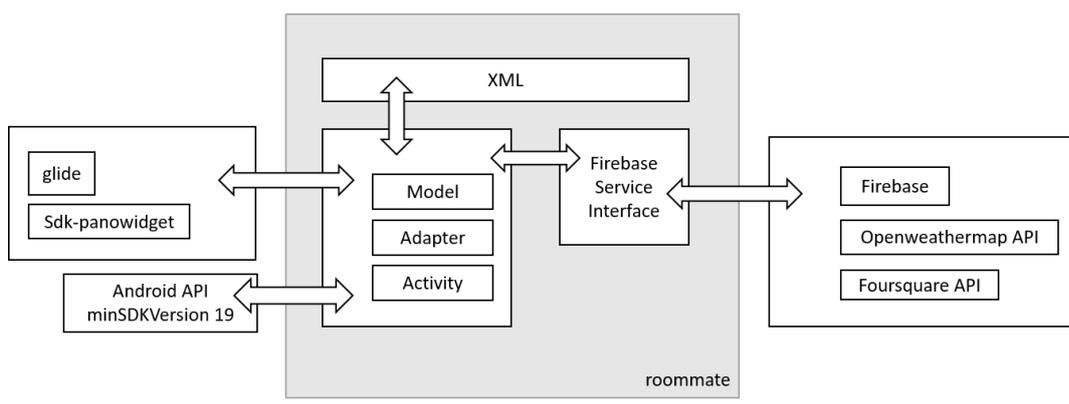
Gambar 5.1 Perancangan Arsitektur Sistem



Gambar 5.2 Arsitektur Sistem Aplikasi Mobile

Sumber : (STAPIC, 2013)





Gambar 5.3 Arsitektur Sistem Aplikasi Roommate

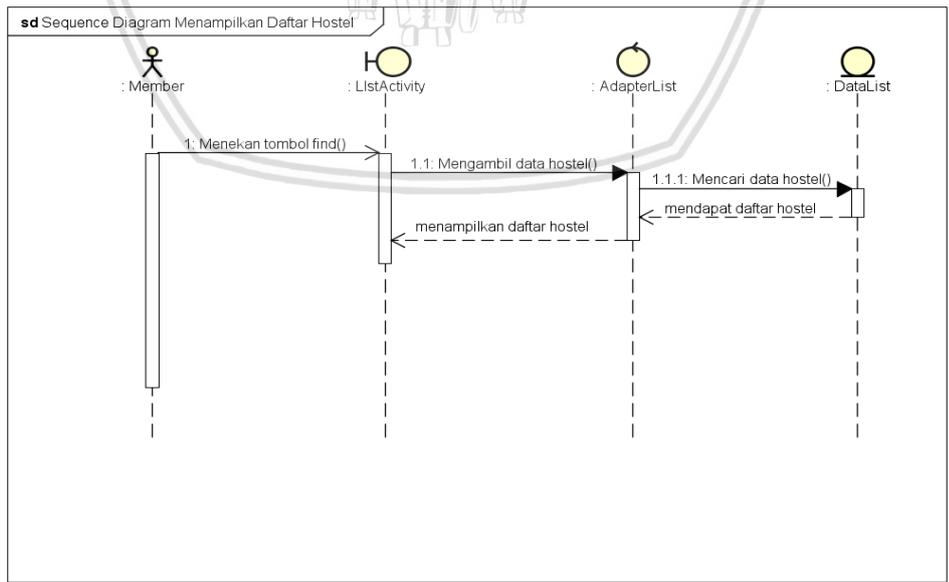
5.1.2 Perancangan Perangkat Lunak

5.1.2.1 Sequence Diagram

Diagram sequence menggambarkan perilaku *object* pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek (Hendini, 2016).

1. Diagram Sequence Menampilkan Daftar Hostel

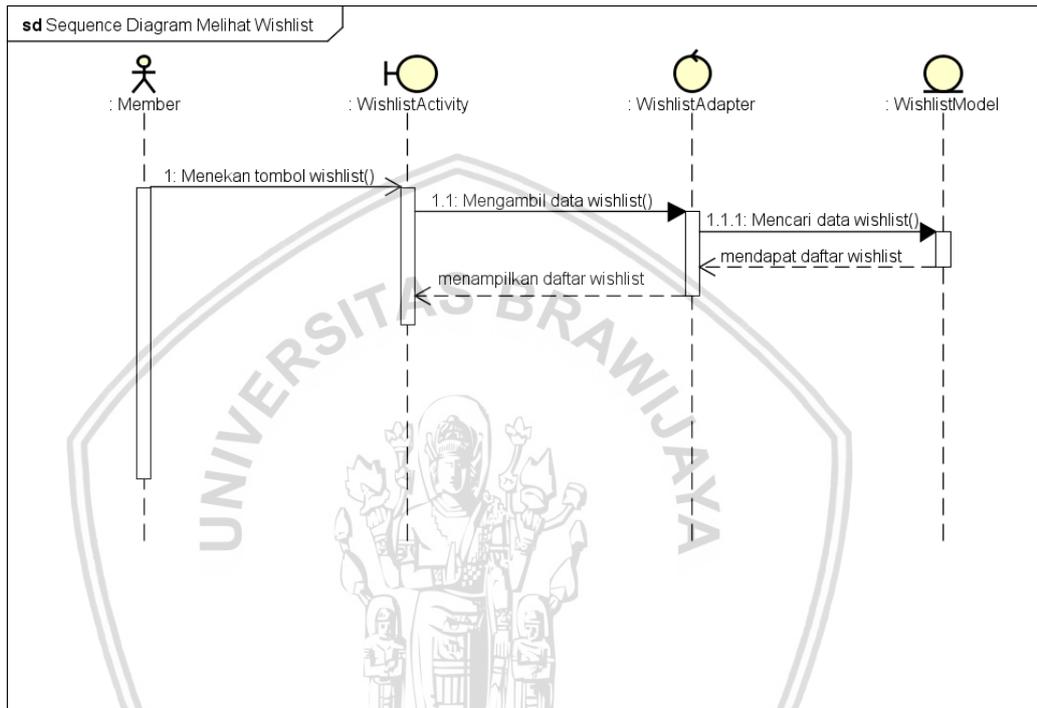
Gambar 5.4 dibawah ini merupakan diagram *sequence* yang menggambarkan interaksi ketika *member* melakukan proses melihat daftar hostel pada sistem. Alur untuk menampilkan daftar hostel adalah ketika member menekan tombol find pada halaman menu utama aplikasi, data akan diambil dari *DataList* dan data akan dikirim ke *AdapterList* agar dapat ditampilkan dalam bentuk list dan ditampilkan pada *ListActivity*.



Gambar 5.4 Diagram Sequence Menampilkan Daftar Hostel

2. Diagram *Sequence* Melihat *Wishlist*

Gambar 5.5 dibawah ini merupakan diagram *sequence* yang menggambarkan interaksi ketika *member* melakukan proses melihat informasi detail hostel yang dipilih. Alurnya adalah ketika member memilih daftar keinginan yang telah disimpan. Alurnya adalah ketika member menekan tombol *wishlist* maka data akan diambil dari *WishlistModel* lalu ditampilkan pada *WishlistActivity* dalam bentuk list melalui *WishlistAdapter*.

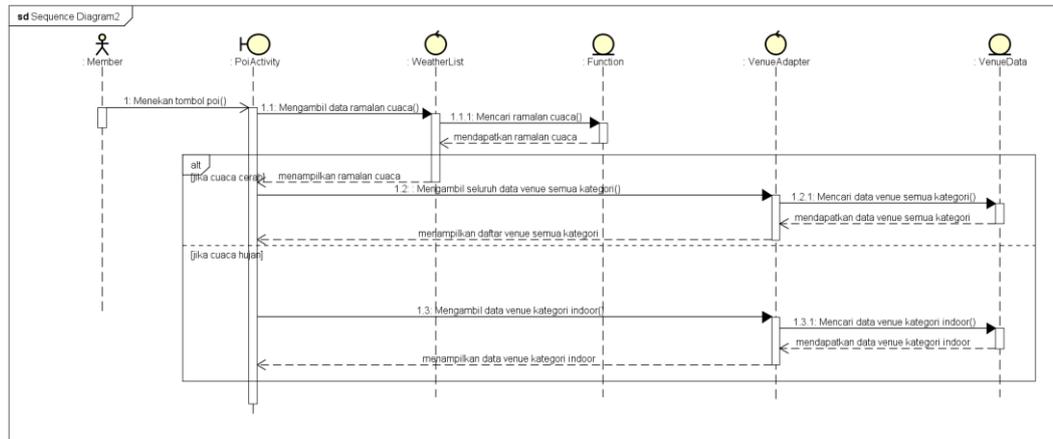


Gambar 5.5 Diagram *Sequence* Melihat Informasi

3. Diagram *Sequence* Melihat *Point Of Interest*

Pada gambar 5.6 merupakan diagram *sequence* ketika *member* melakukan proses untuk dapat melihat cuaca dan rekomendasi tempat populer. Alurnya adalah ketika member menekan tombol *poi* pada aplikasi, maka data cuaca akan dicari pada *Function* dan ditampilkan melalui *WeatherList*. Adanya suatu kondisi jika hasil dari cuaca adalah cerah maka akan diampikan rekomendasi tempat populer dengan semua kategori pada *PoiActivity* melalui *VenueData* dan *VenueAdapter* untuk menampilkan dalam bentuk list. Sedangkan jika kondisi cuaca hujan maka hanya akan menampilkan rekomentasi tempat populer dengan kategori *indoor*.

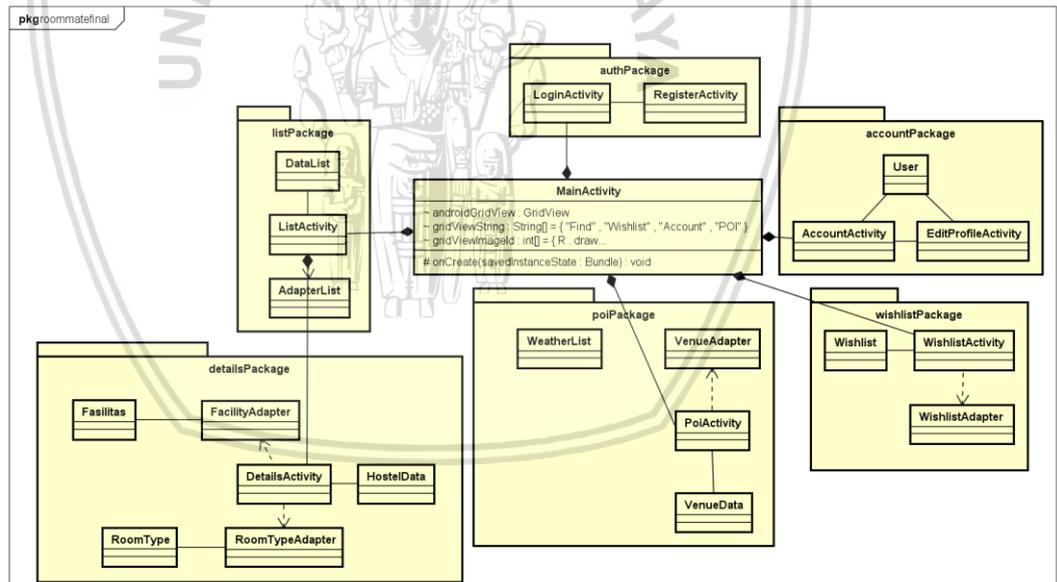




Gambar 5.6 Diagram Sequence Melihat Point Of Interest

5.1.2.2 Perancangan Class Diagram

Perancangan *class diagram* merupakan diagram yang digunakan untuk menunjukkan kelas dan relasi pada sistem. Kelas yang digunakan dikelompokkan berdasarkan fungsi-fungsinya dalam sebuah *package*. Perancangan kelas diagram sistem ini ditunjukkan pada gambar 5.7.



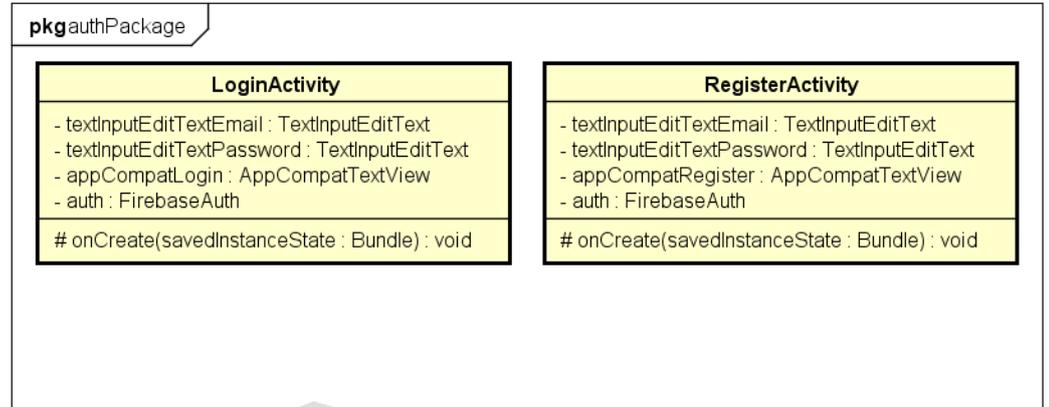
Gambar 5.7 Perancangan Class Diagram

1. Perancangan Class Diagram authPackage

Pada *authPackage* terdapat dua kelas yaitu *LoginActivity* dan *RegisterActivity*. Kelas *LoginActivity* digunakan untuk menampilkan *form* berisi email dan password yang berfungsi ketika pengguna ingin masuk pada sistem. Sedangkan untuk kelas *RegisterActivity* digunakan untuk menampilkan *form*



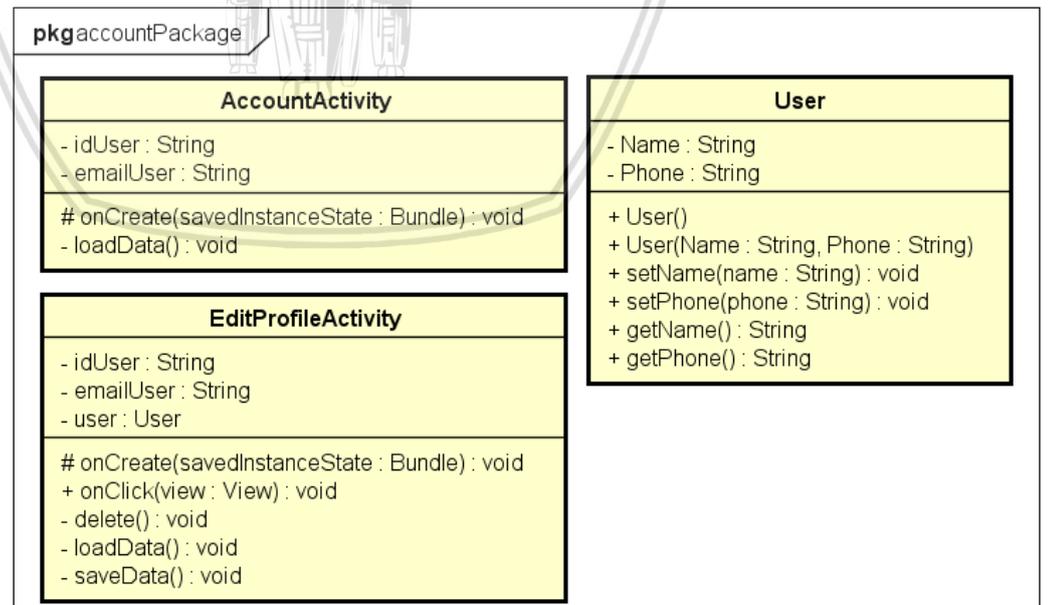
pendaftaran akun. Perancangan kelas diagram ini dapat dilihat pada gambar 5.8



Gambar 5.8 Perancangan Class Diagram authPackage

2. Perancangan Class Diagram accountPackage

Pada *accountPackage* terdapat tiga buah kelas yaitu *User*, *AccountActivity* dan *EditProfileActivity*. Kelas *User* adalah kelas model untuk menyimpan variabel data seperti *name* dan *phone* pengguna. Kelas *AccountActivity* digunakan untuk menampilkan data pengguna berisi *name* dan *phone*. Sedangkan untuk kelas *EditProfileActivity* digunakan untuk menampilkan *form* ketika pengguna ingin mengganti data profil serta menampilkan tombol pilihan untuk menghapus akun. Perancangan kelas diagram ini dapat dilihat pada gambar 5.9.

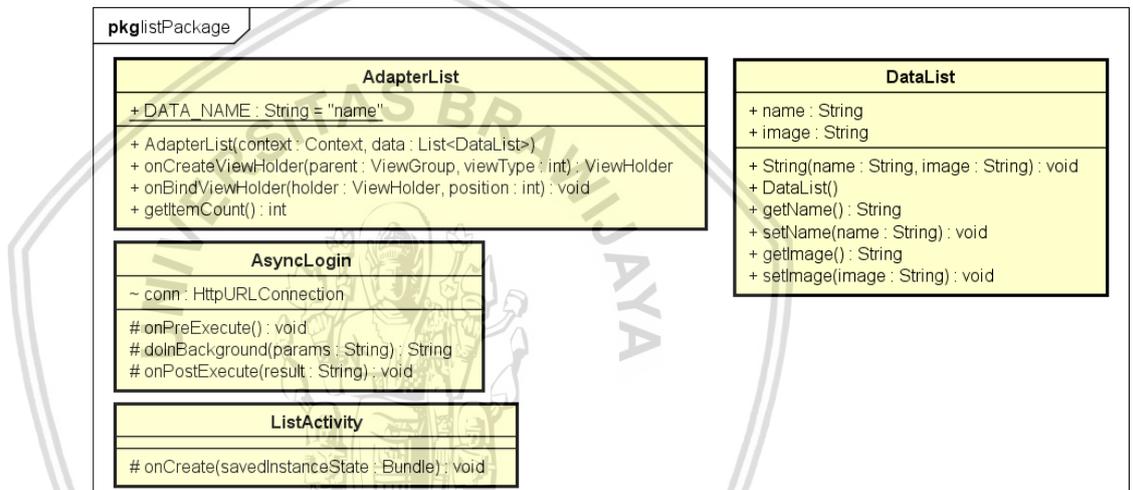


Gambar 5.9 Perancangan Class Diagram accountPackage



3. Perancangan Class Diagram listPackage

Pada *listPackage* terdapat empat kelas yaitu *AdapterList*, *AsyncLogin*, *ListActivity* dan *DataList*. Pada kelas *AsyncLogin* terdapat variabel *HttpURLConnection* yang berfungsi untuk membuat koneksi HTTP yang berguna untuk menerima data melalui Foursquare API. Kelas kedua yaitu *DataList* yang berfungsi sebagai kelas model untuk menyimpan data yang ada. Kelas ketiga yaitu *AdapterList* yang berfungsi untuk mengatur item pada listview dan mengatur data model dari setiap item listview. Kelas terakhir adalah kelas *ListActivity* yang berfungsi untuk menampilkan daftar hostel dengan tampilan listview. Perancangan kelas diagram ini dapat dilihat pada gambar 5.10



Gambar 5.10 Perancangan Class Diagram listPackage

4. Perancangan Class Diagram detailsPackage

Pada *detailsPackage* terdapat enam kelas yang terdiri dari kelas model yaitu *HostelData*, *Fasilitas* dan *RoomType*. Untuk kelas adapter yaitu *FacilityAdapter* dan *RoomTypeAdapter*. Kelas model berfungsi untuk menyimpan data yang akan ditampilkan pada item dari listview. Sedangkan kelas adapter berfungsi untuk mengatur item pada listview dan mengatur data model dari setiap item listview. Data yang disimpan akan ditampilkan pada kelas *DetailsActivity*. Perancangan kelas diagram ini dapat dilihat pada gambar 5.11

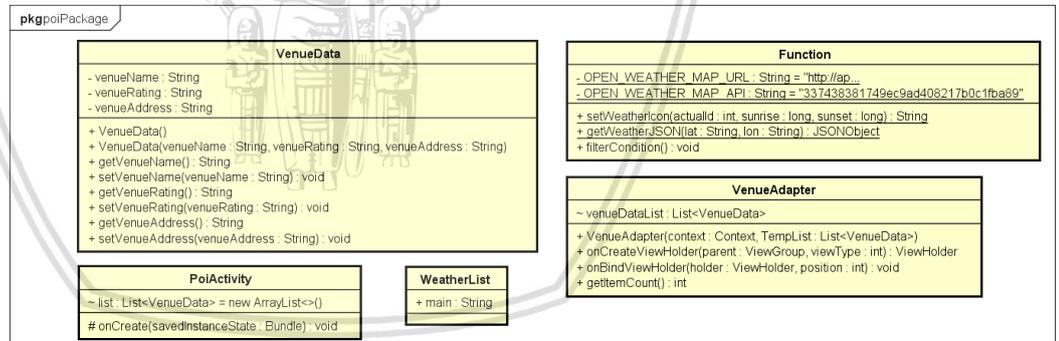




Gambar 5.11 Perancangan Class Diagram detailsPackage

5. Perancangan Class Diagram poiPackage

Pada poiPackage terdapat lima kelas. Kelas Function digunakan untuk mengambil data cuaca pada Openweathermap API dan akan disimpan dalam kelas WeatherList. Untuk tempat populer dengan atribut yang terdapat pada kelas model VenueData disimpan dan ditampilkan oleh kelas PoiActivity melalui kelas adapter VenueAdapter. Perancangan kelas diagram ini dapat dilihat pada gambar 5.12

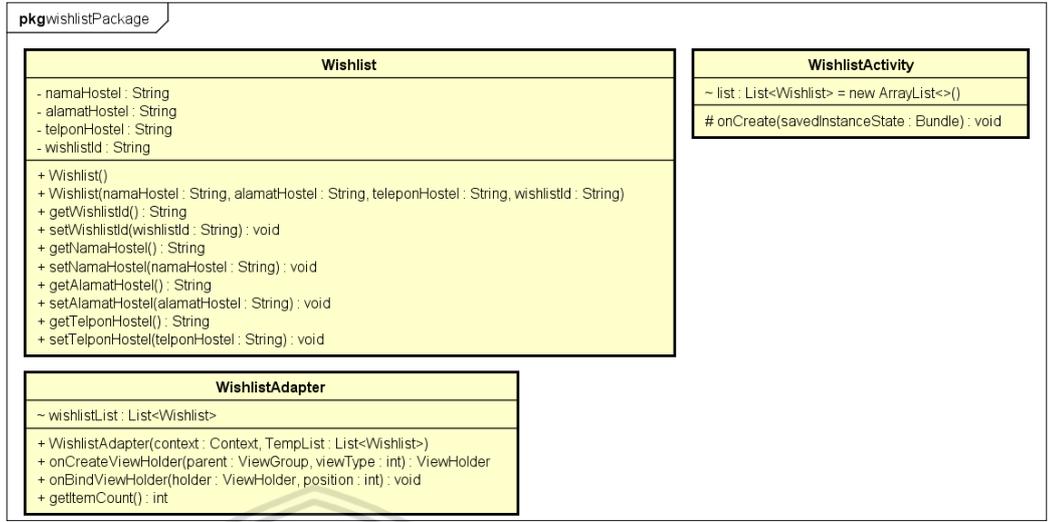


Gambar 5.12 Perancangan Class Diagram poiPackage

6. Perancangan Class Diagram wishlistPackage

Pada wishlistPackage terdapat tiga kelas yang terdiri dari kelas Wishlist, WishlistAdapter dan WishlistActivity. Kelas Wishlist berfungsi menjadi kelas model yaitu untuk menyimpan data seperti namaHostel, alamatHostel, telponHostel dan wishlistId. Untuk kelas WishlistAdapter berfungsi untuk mengatur item pada listview dan mengatur data model dari setiap item listview. Semua data akan ditampilkan pada halaman WishlistActivity dengan tampilan listview. Perancangan kelas ini dapat dilihat pada gambar 5.13





Gambar 5.13 Perancangan Class Diagram wishlistPackage

5.1.2.3 Perancangan Basis Data

Dalam perancangan basis data ini, setiap atribut dalam skema basis data akan dibuat berdasarkan atribut pada class diagram. Kemudian dilakukan pengelompokan atribut yang berhubungan. Terdapat empat skema yang digunakan yaitu *user*, *wishlist*, *hostel* dan *venue*.

Pada skema basis data pertama berisi informasi identitas pengguna yang diisikan pada saat pendaftaran akun dan ketika melakukan perubahan pada profil pengguna.

```

User : {
  userId : {
    Name :
    Phone :
  }
}
    
```

Gambar 5.14 Struktur json schema User pada Firebase

Identifer	Providers	Created	Signed In	User UID ↑
...com	✉	Dec 31, 2017	Jan 1, 2018	i5TPSYNGdKgBLGyglXUAI6XMzy03

Gambar 5.15 Authentication email dan password

Keterangan dari skema basis data *User* dapat dilihat pada tabel dibawah ini:

Tabel 5.1 Tabel Keterangan Skema Basis Data *User*

No	Nama Field	Type Data	Keterangan
1	userId	String	Untuk menyimpan id akun <i>user</i>
2	Name	String	Untuk menyimpan data nama <i>user</i>



3	<i>Phone</i>	String	Untuk menyimpan data nomor telepon <i>user</i>
---	--------------	--------	--

Pada skema basis data kedua berisi tentang informasi tambahan dari daftar hostel yang didapatkan dari Foursquare API.

```

HostelList : {
  alamatHostel :
  fasilitasHostel :
  gambar :
  gambar360 :
  namaHostel :
  ratingHostel : {
    ratingId : {
      userId : {
        rating :
        komentar :
      }
    }
  }
  telp2 :
  telpHostel :
  tipeKamar : {
    fasilitasKamar :
    harga :
    nama :
  }
  Website :
}
    
```

Gambar 5.16 Struktur json *schema* Hostel pada Firebase

Keterangan dari skema basis data Hostel dapat dilihat pada tabel dibawah ini:

Tabel 5.2 Tabel Keterangan Skema Basis Data Hostel

No	Nama Field	Tipe Data	Keterangan
1	alamatHostel	String	Untuk menyimpan data alamat hostel
2	fasilitasHostel	ArrayList<String>	Untuk menyimpan daftar fasilitas hostel
3	Gambar	String	Untuk menyimpan <i>link</i> gambar hostel
4	Gambar360	String	Untuk menyimpan <i>link</i> gambar panorama hostel
5	namaHostel	String	Untuk menyimpan data nama hostel
6	ratingId	String	Untuk menyimpan id rating



7	userId	String	Untuk menyimpan id akun <i>user</i> yang telah memberikan <i>review</i>
8	Rating	String	Untuk menyimpan data rating yang diberikan pengguna
9	Komentar	String	Untuk menyimpan data komentar yang diberikan oleh pengguna
10	Telp2	String	Untuk menyimpan data nomor telepon hostel yang kedua
11	telpHostel	String	Untuk menyimpan data nomor telepon utama hostel
12	fasilitasKamar	ArrayList<String>	Untuk menyimpan daftar fasilitas kamar pada masing-masing tipe kamar
13	Harga	String	Untuk menyimpan harga dari masing-masing tipe kamar
14	Nama	String	Untuk menyimpan data nama dari masing-masing tipe kamar
15	website	String	Untuk menyimpan data <i>link</i> website dari hostel

Untuk *schema* ketiga berisi tentang informasi data wishlist yang berisi `idWishlist`, `idUser`, `namaHostel`, `alamatHostel` dan `telponHostel` yang disimpan.

```

Wishlist : {
  userId : {
    wishlistId : {
      namaHostel :
      alamatHostel :
      telponHostel :
    }
  }
}
    
```

Gambar 5.17 Struktur json *schema* *Wishlist* pada Firebase

Keterangan dari skema basis data *Wishlist* dapat dilihat pada tabel dibawah ini:

Tabel 5.3 Tabel Keterangan Skema Basis Data *Wishlist*

No	Nama <i>Field</i>	Tipe Data	Keterangan
1	userId	String	Untuk menyimpan data berupa id <i>user</i>

2	wishlistId	String	Untuk menyimpan data dari id <i>wishlist</i> yang telah disimpan oleh pengguna
3	namaHostel	String	Untuk menyimpan data nama hostel yang disimpan oleh pengguna
4	alamatHostel	String	Untuk menyimpan data alamat hostel yang telah disimpan oleh pengguna
5	telponHostel	String	Untuk menyimpan data nomor telepon hostel yang telah disimpan oleh pengguna

Untuk *schema* keempat beris tentang informasi tempat populer dengan kategori tempat dan dipilih berdasarkan rating tertinggi dari Foursquare API.

```

Venue : {
  venueAddress :
  venueName :
  venueRating :
  venueCategory :
}
    
```

Gambar 5.18 Struktur json *schema Venue* pada Firebase

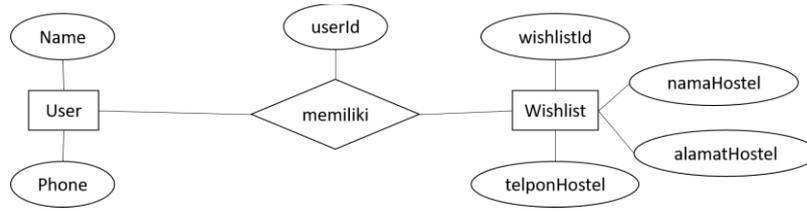
Keterangan dari skema basis data *Venue* dapat dilihat pada tabel dibawah ini:

Tabel 5.4 Tabel Keterangan Skema Basis Data *Venue*

No	Nama <i>Field</i>	Tipe Data	Keterangan
1	venueAddress	String	Untuk menyimpan data alamat tempat populer berdasarkan Foursquare API
2	venueName	String	Untuk menyimpan data nama tempat populer berdasarkan Foursquare API
3	venueRating	String	Untuk menyimpan data rating dari tempat populer berdasarkan Foursquare API
4	venueCategory	String	Untuk menyimpan kategori dari tempat populer berdasarkan Foursquare API



5.1.2.4 Perancangan ER Diagram



Gambar 5.19 ER Diagram

Pada gambar 5.19 terdapat ER Diagram dari relasi antara Wishlist dan User. Entitas pada diagram diatas adalah User dan Wishlist dengan mempunyai relasi User memiliki Wishlist. Untuk atribut pada entitas user yaitu *Name* dan *Phone* sedangkan pada entitas wishlist yaitu *wishlistId*, *namaHostel*, *alamatHostel* dan *telponHostel*. Atribut relasi pada diagram diatas yaitu *userId* yang berada pada entitas User dan Wishlist.

5.1.2.5 Perancangan Kode Program

1. Algoritme Menghubungi Pihak Hostel menggunakan whatsapp

```

    Algoritme rekomendasi tempat populer
    Deklarasi :
        - PackageManager
        - String : uri
        - Boolean : app_installed

    Deskripsi :
        - Proses :
        - Pengecekan pada PackageManager dengan nilai awal
          app_installed yaitu false
        - Dilakukan seleksi kondisi sebagai berikut:
            a. Jika pada PackageManager sudah
               terinstall aplikasi whatsapp maka nilai
               app_installed menjadi true
            b. Jika nama aplikasi tidak terdaftar pada
               PackageManager maka nilai app_installed
               menjadi false
        - Mengembalikan nilai app_installed berdasarkan hasil
          seleksi kondisi
    
```

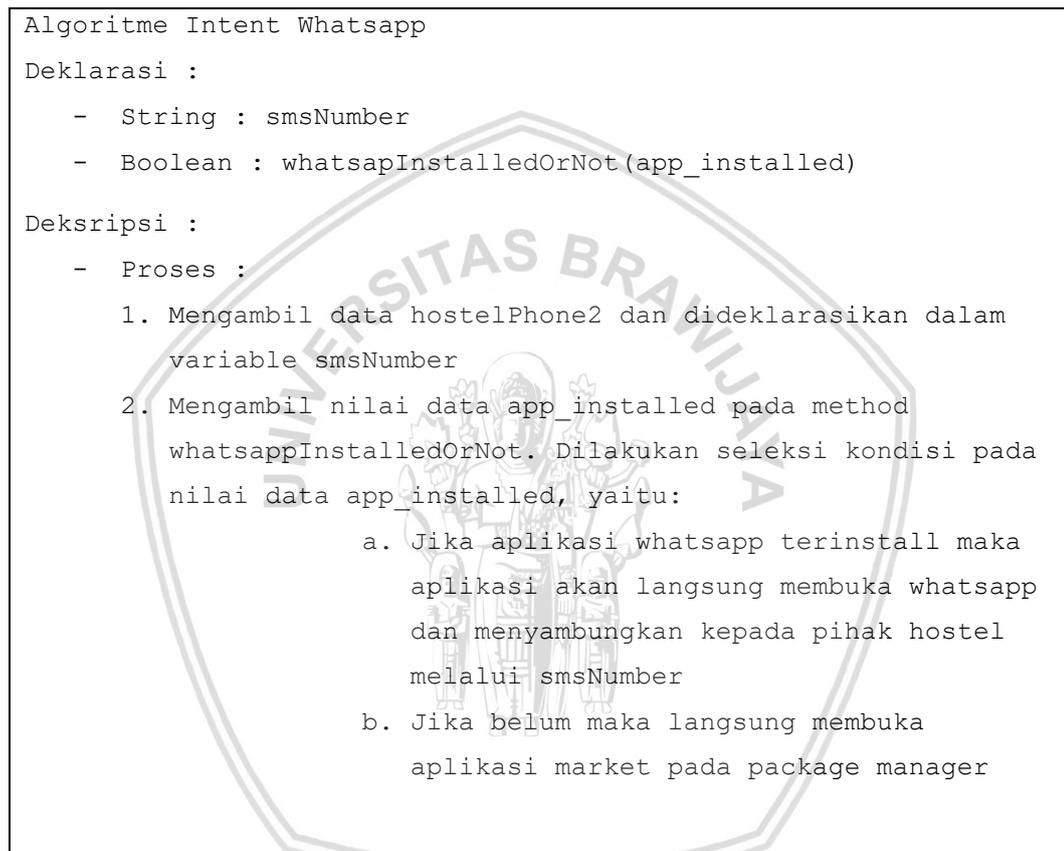
Gambar 5.20 Algoritme menghubungi pihak hostel menggunakan whatsapp

Pada gambar diatas berisi algoritme untuk menghubungi pihak hostel melalui aplikasi whatsapp. Untuk deklarasi terdapat tiga parameter yaitu



PackageManager, uri dengan tipe data String serta app_installed dengan tipe data Boolean. Untuk penjelasan deskripsi proses algoritme ini adalah yang pertama yaitu pengecekan pada PackageManager dengan nilai awal app_installed yaitu false. Lalu dilakukan perkondisian pada PackageManager, jika pada PackageManager telah terinstall aplikasi whatsapp maka nilai app_installed mejadi true, untuk sebaliknya jika belum terinstall maka nilai tetap false. Setelah dilakukan perkondisian maka nilai app_installed akan dikembalikan berdasarkan hasil seleksi perkondisian tersebut.

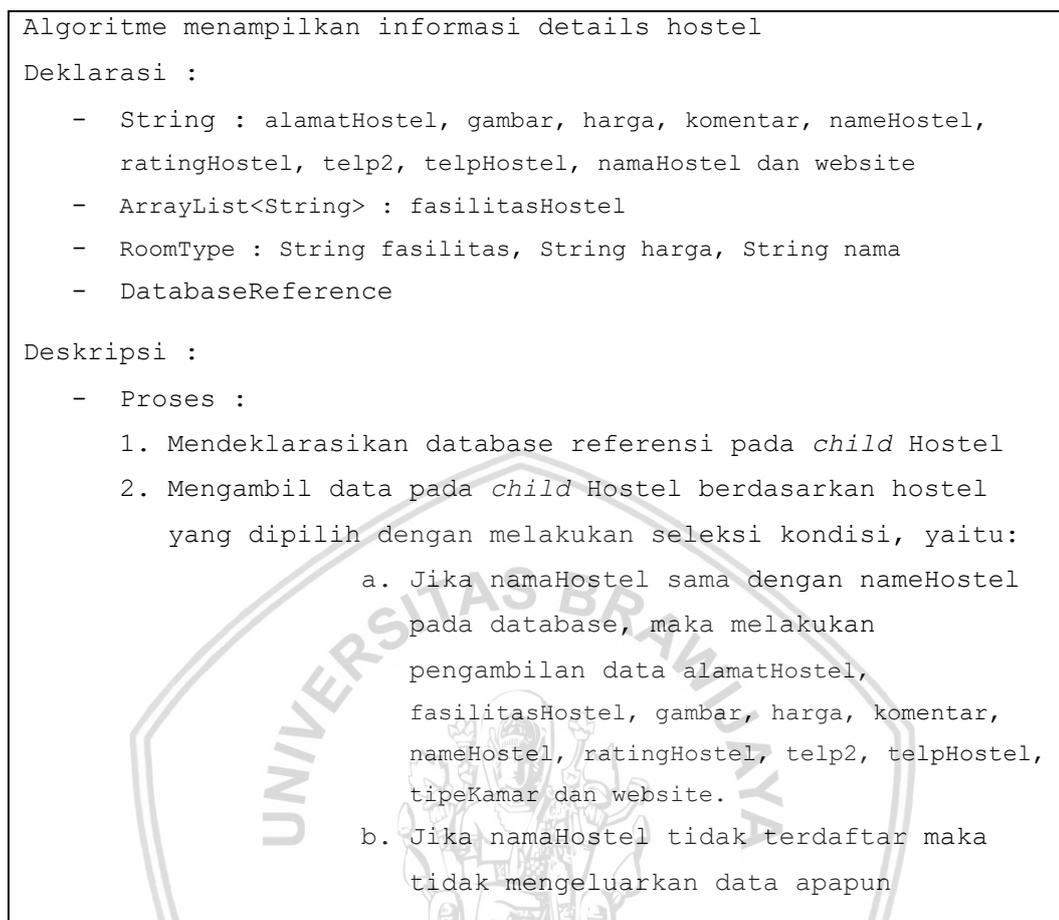
2. Algoritme Membuka Aplikasi Whatsapp



Gambar 5.21 Algoritme Membuka Aplikasi Whatsapp

Pada gambar diatas terdapat algoritme untuk membuka aplikasi whatsapp untuk melakukan panggilan atau mengirim pesan kepada pihak hostel. Pada algoritme ini terdapat dua variabel yaitu smsNumber dengan tipe data String dan nilai yang diambil dari method whatsappInstalledOrNot dengan tipe data Boolean. Untuk penjelasan deskripsi proses dimulai dari mengambil data hostelPhone2 dan dideklarasikan kedalam variabel smsNumber, lalu mengambil nilai data app_installed dan dilakukan seleksi kondisi pada nilai tersebut. Jika aplikasi whatsapp sudah terinstall maka sistem akan langsung membuka aplikasi whasapp dan terhubung dengan pihak hostel melalui smsNumber, sedangkan jika aplikasi whatsapp belum terinstall maka sistem akan membuka aplikasi market pada PackageManager.

3. Algoritme Menampilkan informasi details hostel



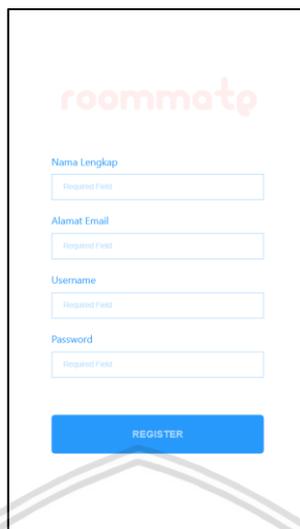
Gambar 5.22 Algoritme Menampilkan Informasi Detail Hostel

Pada gambar diatas berisi algoritme menampilkan informasi detail hostel. Terdapat beberapa variabel yang dibagi menjadi empat berdasarkan tipe datanya. Untuk tipe data String terdapat sepuluh variabel yaitu alamatHostel, gambar, harga, komentar, nameHostel, ratingHostel, telp2, telpHostel, namaHostel dan website. Untuk tipe data ArrayList<String> yaitu fasilitasHostel. Untuk variabel RoomType terdapat fasilitas, harga dan nama dengan tipe data String serta DatabaseReference yang berfungsi ketika membaca data pada Firebase. Deskripsi proses pada algoritme diatas adalah pertama-tama mendeklarasikan DatabaseReference untuk membaca data pada child Hostel berdasarkan hostel yang dipilih oleh pengguna pada halaman List Hostel sebelumnya.

5.1.2.6 Perancangan Antarmuka Iterasi 1

Bagian ini menjelaskan tentang perancangan antarmuka dari sistem informasi hostel. Perancangan antarmuka berisi tentang tampilan aplikasi yang akan digunakan oleh calon pengguna aplikasi. *Screenflow* digunakan untuk menggambarkan keseluruhan antarmuka sistem yang sesuai dengan kebutuhan pengguna.

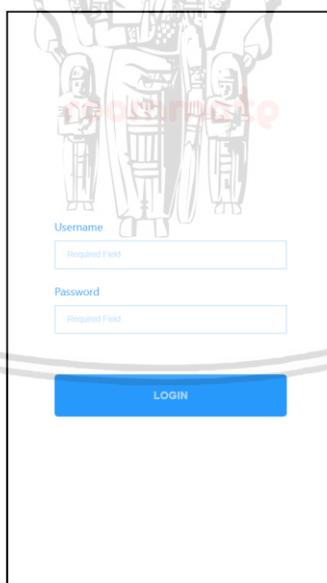
1. Halaman *register*

The image shows a registration form for 'roommate'. At the top, the logo 'roommate' is displayed in a light red font. Below the logo, there are four input fields, each with a label and a 'Required Field' indicator: 'Nama Lengkap', 'Alamat Email', 'Username', and 'Password'. Each field is a simple white box with a light blue border. At the bottom of the form is a blue button with the text 'REGISTER' in white capital letters.

Gambar 5.23 Perancangan Antarmuka Halaman *Register*

Pada gambar 5.22 terdapat perancangan antarmuka halaman *register* untuk pengguna *non-member* menjadi *member*. Pengguna *non-member* harus mengisi nama lengkap, alamat *email*, *username* dan *password* untuk menjadi *member*.

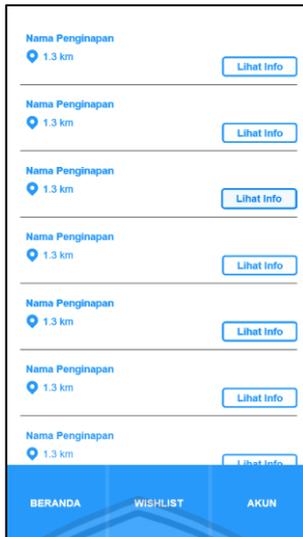
2. Halaman *login*

The image shows a login form for 'roommate'. At the top, the logo 'roommate' is displayed in a light red font. Below the logo, there are two input fields, each with a label and a 'Required Field' indicator: 'Username' and 'Password'. Each field is a simple white box with a light blue border. At the bottom of the form is a blue button with the text 'LOGIN' in white capital letters.

Gambar 5.24 Perancangan Antarmuka Halaman *Login*

Pada gambar 5.23 terdapat perancangan antarmuka halaman *login*. Ketika sudah melakukan *register* dan mempunyai *username* dan *password* maka pengguna dapat melakukan *login* pada aplikasi dengan mengisi *username* dan *password*.

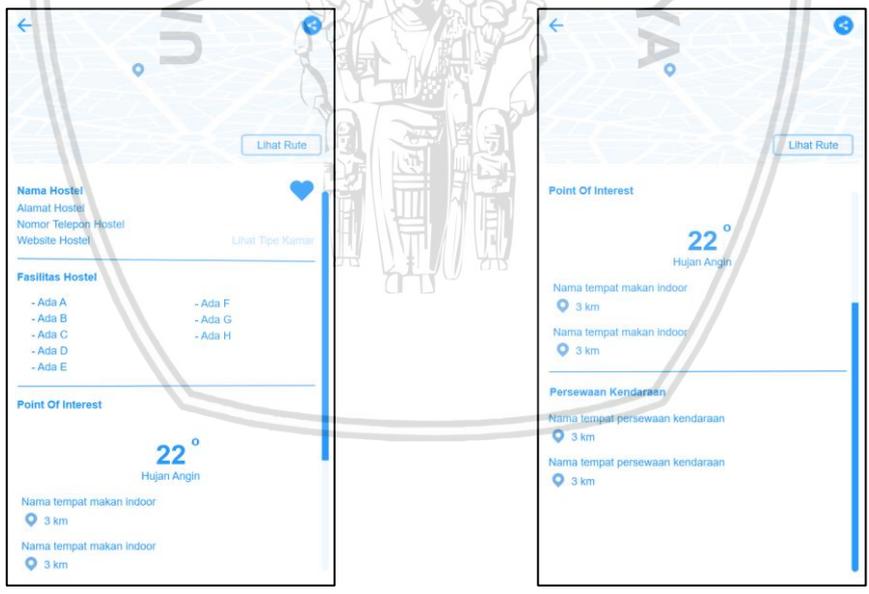
3. Halaman Daftar Hostel



Gambar 5.25 Perancangan Antarmuka Halaman Daftar Hostel

Pada gambar 5.24 terdapat perancangan antarmuka halaman daftar hostel. Setelah melakukan filter, maka sistem akan menampilkan halaman daftar hostel dengan rating dan harga kamar.

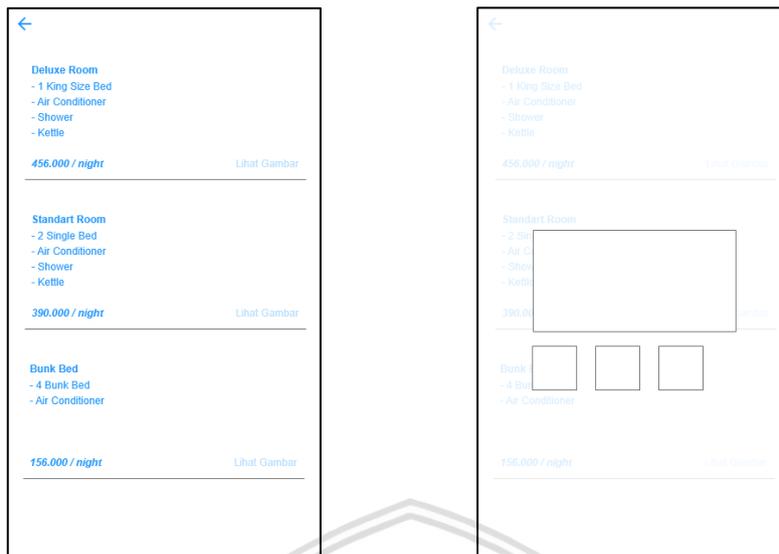
4. Halaman Informasi



Gambar 5.26 Perancangan Antarmuka Halaman Informasi

Pada gambar 5.25 terdapat perancangan antarmuka halaman informasi. Setelah pengguna memilih hostel dari daftar maka halaman informasi menampilkan *contact person* dari hostel, fasilitas, *point of interest*, persewaan kendaraan dan forum untuk berbagi komentar dan rating.

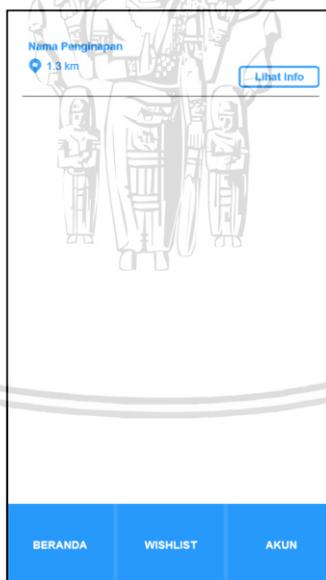
5. Halaman Lihat Tipe Kamar



Gambar 5.27 Perancangan Antarmuka Halaman Lihat Tipe Kamar

Pada gambar 5.26 terdapat perancangan antarmuka halaman tipe kamar. Halaman ini menampilkan tipe kamar yang tersedia serta gambar dari kamar tersebut.

6. Halaman Wishlist

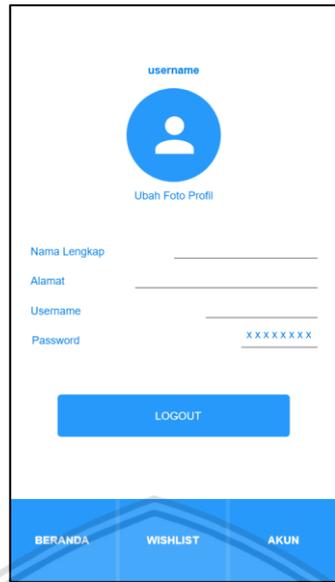


Gambar 5.28 Perancangan Antarmuka Halaman Wishlist

Pada gambar 5.27 terdapat perancangan antarmuka halaman *wishlist*. Ketika pengguna ingin menyimpan informasi tentang hostel yang dipilih maka *list* hostel tersebut akan tersimpan pada halaman *wishlist*.

7. Halaman Akun



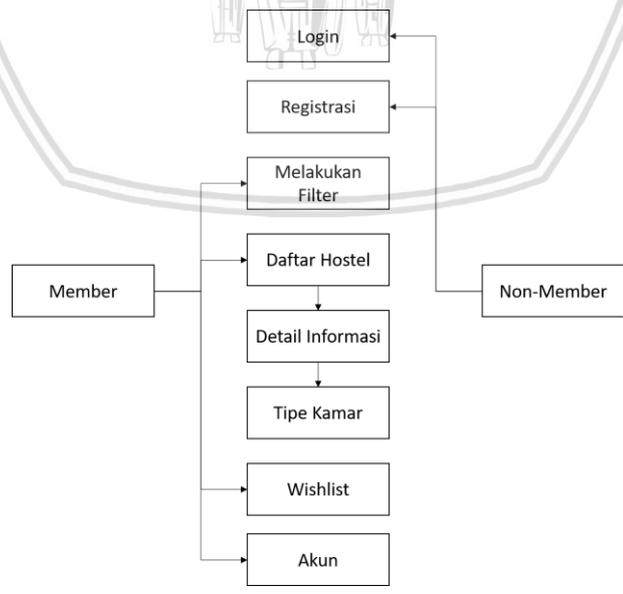


Gambar 5.29 Perancangan Antarmuka Halaman Akun

Pada gambar 5.28 terdapat perancangan antarmuka halaman akun. Pada halaman ini terdapat fitur untuk merubah foto profil dan data profil tersebut serta pilihan untuk keluar dari aplikasi.

5.1.2.7 Perancangan *Screenflows* Antarmuka

Pada tahap ini akan diberikan alur dari antarmuka aplikasi. Bagaimana dan menuju ke halaman manakah ketika memberikan aksi pada tombol.



Gambar 5.30 Perancangan *Screenflows* Antarmuka Iterasi Pertama

Penjelasan dari alur *screenflow* diatas adalah sebagai berikut:



1. Pengguna memilih halaman *register* untuk mendaftarkan akun lalu mengisi data yang dibutuhkan.
2. Pengguna kembali pada halaman *login* dan mengisi *form* sesuai data di database.
3. Pengguna melakukan *filter* harga hostel.
4. Pengguna melihat daftar hostel berdasarkan harga.
5. Pengguna melihat detail informasi hostel yang dipilih.
6. Pengguna melihat tipe kamar yang ada pada hostel tersebut.
7. Pengguna memilih halaman *wishlist*.
8. Pengguna memilih halaman akun.

5.1.2.8 Hasil Evaluasi Iterasi 1

Langkah awal evaluasi tahap pertama ini mendapatkan hasil yang sudah dilakukan pada pengujian. Dari hasil pengujian didapatkan beberapa masukan dan saran, serta kelebihan dan kekurangan dari masing masing fitur. Kuisisioner untuk melakukan pengujian sudah terlampir pada halaman lampiran C. Beberapa saran dan masukan bisa dilihat pada Tabel 5.1.

Dari hasil evaluasi terdapat beberapa halaman yang perlu perbaikan, berikut penjelasannya:

1. Halaman *Filter* dihapus, karena menurut calon pengguna dengan adanya halaman tersebut semakin membuat aplikasi menjadi rumit.
2. Tab Host yang berada dibawah diganti menjadi *icon* dengan ukuran lebih besar beserta keterangan *icon* agar calon pengguna langsung mengerti untuk apa *icon* tersebut digunakan.
3. Halaman *Login* dan Halaman *Register* menggunakan *field* alamat email dan password saja agar memudahkan proses autentikasi pada database yang digunakan.
4. Pada halaman *list* hostel hanya menampilkan nama hostel saja.
5. Untuk cuaca dan rekomendasi tempat dijadikan satu pada halaman POI, agar lebih memudahkan calon pengguna melihat rekomendasi beserta cuaca saat ini.
6. Pada halaman *edit profil*, bagian foto profil dihapus karena menurut calon pengguna tidak terlalu perlu digunakan. Penambahan data nama lengkap dan nomor telepon pengguna untuk data pada database.
7. Pada halaman detail ditambahkan fitur foro panorama 360 agar dapat melihat interior atau eksterior hotel lebih baik.
8. Penambahan fitur hapus akun jika pengguna akan menghapus akun yang telah dibuat.

5.1.2.9 Perancangan Antarmuka Iterasi 2

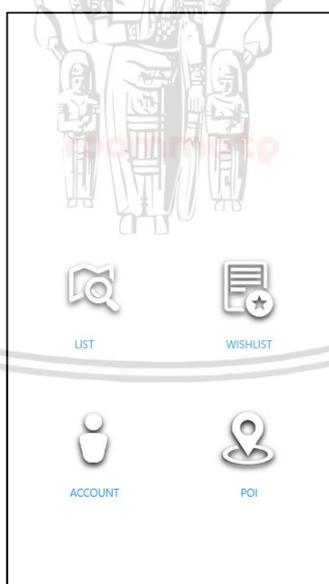
1. Halaman *Splashscreen*



Gambar 5.31 Perancangan Antarmuka Halaman *Splashscreen*

Pada gambar 5.30 menampilkan perancangan antarmuka halaman *splashscreen*. Pada halaman *splashscreen* terdapat satu komponen yaitu logo dari aplikasi roommate.

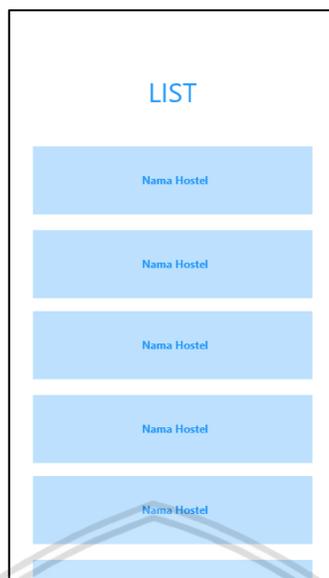
2. Halaman Menu Utama



Gambar 5.32 Perancangan Antarmuka Halaman Utama

Pada gambar 5.31 menampilkan perancangan halaman utama. Pada halaman ini berisi empat menu utama yaitu *list*, *wishlist*, *account* dan *point of interest (poi)*.

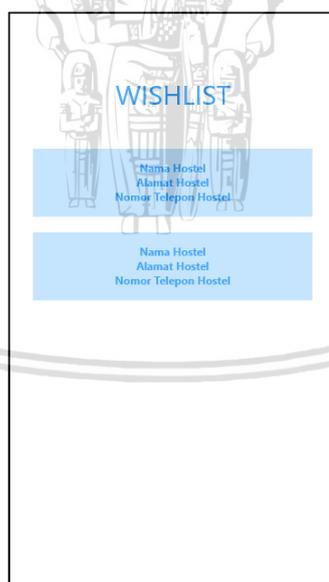
3. Halaman *List Hostel*



Gambar 5.33 Perancangan Antarmuka Halaman List Hostel

Pada gambar 5.32 terdapat perancangan antarmuka halaman *list* hostel. Pada halaman ini menampilkan *list* dari beberapa hostel yang diambil dari Foursquare API.

4. Halaman *Wishlist*



Gambar 5.34 Perancangan Antarmuka Halaman *Wishlist*

Pada gambar 5.33 merupakan perancangan halaman antamuka *wishlist*. Pada halaman ini berisi data hostel yang diinginkan pengguna untuk disimpan.

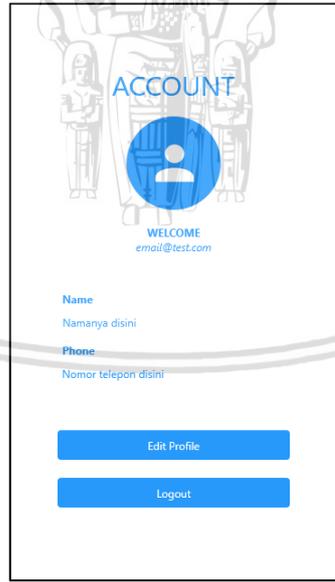
5. Halaman *Point Of Interest*



Gambar 5.35 Perancangan Antarmuka Halaman *Point Of Interest*

Pada gambar 5.34 adalah perancangan antarmuka halaman *point of interest*. Pada halaman ini terdapat informasi waktu, tanggal beserta cuaca saat ini. Serta adanya rekomendasi *popular venue* berdasarkan cuaca.

6. Halaman Akun

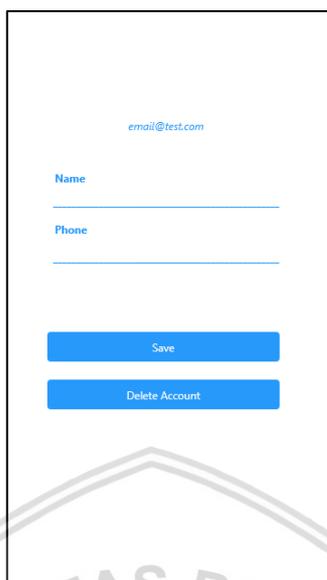


Gambar 5.36 Perancangan Antarmuka Halaman Akun

Pada Gambar 5.35 terdapat perancangan antarmuka halaman akun. Disini pengguna dapat melihat alamat email, nama dan nomor telepon yang terdaftar pada database. Pengguna juga dapat memilih pilihan *edit profil* dan *logout* untuk keluar dari aplikasi.

7. Halaman *Edit Profil*





email@test.com

Name

Phone

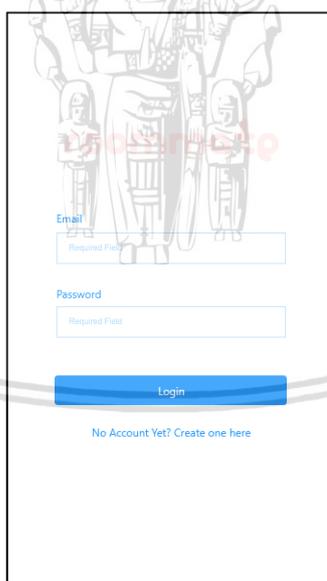
Save

Delete Account

Gambar 5.37 Perancangan Antarmuka Halaman *Edit Profil*

Pada Gambar 5.36 terdapat halaman *edit profil* untuk mengubah nama dan nomor telepon, sampai menghapus akun pada *database*.

8. Halaman *Login*



Email

Required Field

Password

Required Field

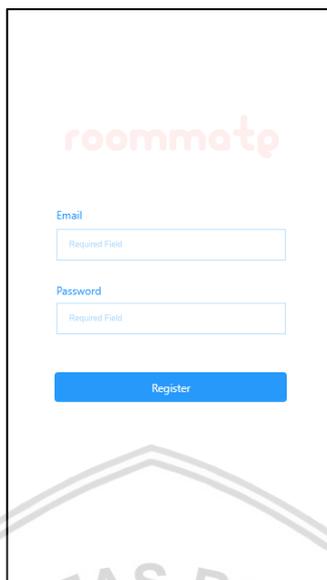
Login

No Account Yet? Create one here

Gambar 5.38 Perancangan Antarmuka Halaman *Login*

Pada Gambar 5.37 terdapat perancangan antarmuka halaman *login*. Yang berisi *field* untuk mengisi alamat *email* dan *password* untuk proses *authentication*. Serta ada pilihan untuk *register* jika pengguna belum memiliki akun pada *database*.

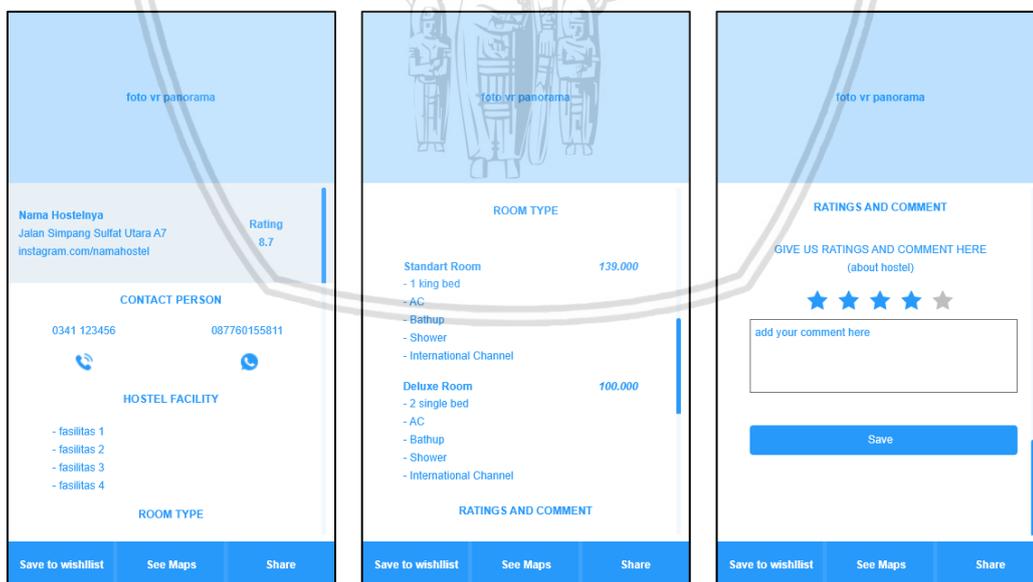
9. Halaman *Register*



Gambar 5.39 Perancangan Antarmuka Halaman Register

Pada Gambar 5.38 terdapat perancangan antarmuka halaman *register*. Pada halaman ini pengguna mengisi alamat *email* dan *password* untuk mendaftarkan akun ke dalam *database*.

10. Halaman Detail Informasi Hostel



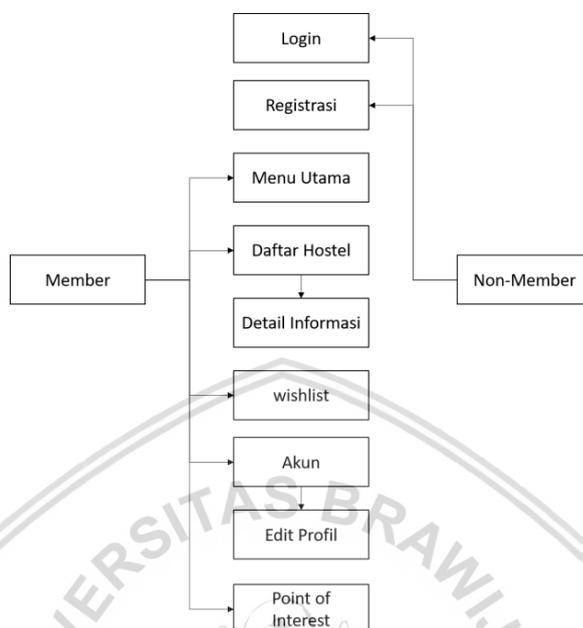
Gambar 5.40 Perancangan Antarmuka Halaman Detail Informasi Hostel

Pada gambar 5.39 terdapat perancangan antarmuka halaman detail informasi hostel. Dilengkapi dengan melihat gambar, melihat informasi, menghubungi pihak hostel, simpan ke *wishlist*, membagikan melalui media sosial, memberikan komentar dan rating.



5.1.2.10 Perancangan Screenflow Antarmuka Iterasi 2

Pada tahap ini diberikan alur dari antarmuka aplikasi. Bagaimana menuju ke halaman lain ketika memberikan aksi pada tombol.



Gambar 5.41 Perancangan Screenflow Antarmuka Iterasi Kedua

5.1.2.11 Hasil Evaluasi Iterasi 2

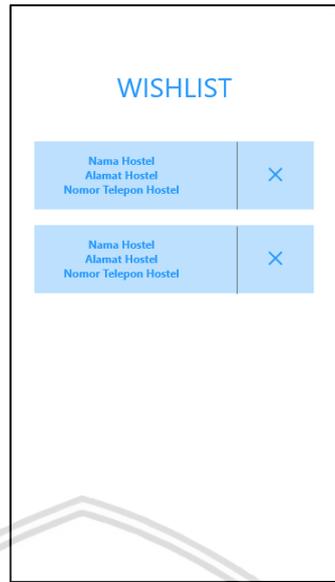
Pada evaluasi iterasi kedua ini mendapatkan hasil yang sudah dilakukan pada pengujian. Dari hasil evaluasi terdapat beberapa halaman yang perlu perbaikan, berikut penjelasannya:

1. Pada halaman *list* hostel, tampilan kurang menarik bagi pengguna.
2. Ditambahkan fitur untuk menghapus daftar *wishlist*.

5.1.2.12 Perancangan Antarmuka Iterasi 3

Perubahan perancangan antarmuka iterasi ketiga hanya pada halaman *wishlist* dan penambahan halaman galeri untuk melihat beberapa foto panorama 360° dari hostel.

1. Halaman *Wishlist*

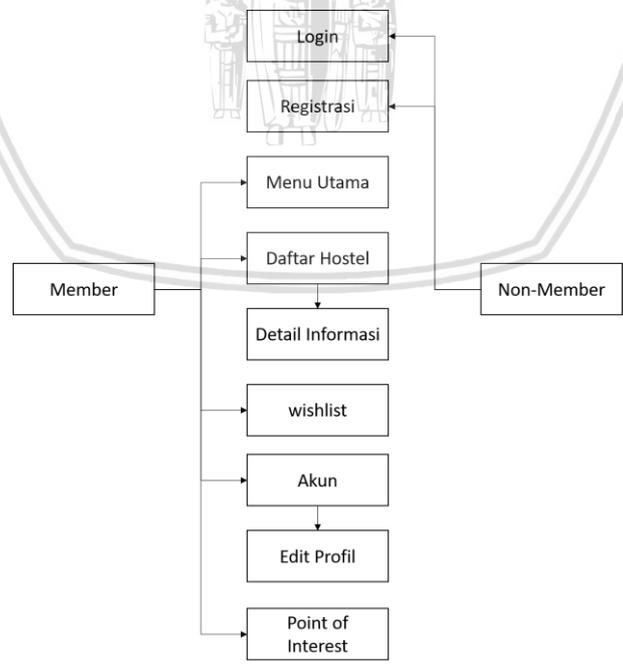


Gambar 5.42 Perancangan Antarmuka Halaman *Wishlist*

Pada gambar 5.41 terdapat perancangan antarmuka halaman *wishlist* yang telah dievaluasi. Adanya penambahan fitur hapus daftar yang berada pada kanan daftar.

5.1.2.13 Perancangan *Screenflow* Antarmuka Iterasi 3

Pada tahap ini diberikan alur dari antarmuka aplikasi. Bagaimana menuju ke halaman lain ketika memberikan aksi pada tombol.



Gambar 5.43 Perancangan *Screenflow* Antarmuka Iterasi Kedua



5.1.2.14 Hasil Evaluasi Iterasi 3

Hasil dari evaluasi iterasi ketiga tidak mendapatkan kritik dan saran lagi dari calon pengguna, maka perancangan antarmuka akan langsung diimplementasikan ke dalam Android Studio.

5.2 Implementasi

Tahap selanjutnya adalah membuat implementasi dari hasil perancangan yang sudah dibuat sebelumnya. Implementasi ini hanya berupa penambahan dan perubahan yang terkait dari hasil evaluasi pada sub bab diatas.

5.2.1 Spesifikasi Sistem

Dalam pembuatan sistem informasi hostel ini, dikembangkan dalam lingkungan implementasi yang terdiri dari dua perangkat yaitu perangkat keras dan perangkat lunak.

5.2.1.1 Spesifikasi Perangkat Keras

Spesifikasi perangkat keras yang digunakan dalam pembuatan aplikasi ini ditunjukkan pada tabel 5.5 dibawah:

Tabel 5.5 Spesifikasi Perangkat Keras Komputer

HP Pavilion X360 13-a013d	
<i>Processor</i>	Intel® Celeron® CPU N2840
<i>Memory(RAM)</i>	4.00 GB
<i>Harddisk</i>	1000GB/TB
<i>Graphic Card</i>	VGA Intel HD Graphics 4400
<i>Monitor</i>	11"

5.2.1.2 Spesifikasi Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan dalam pembuatan aplikasi ini ditunjukkan pada tabel 5.6 dibawah:

Tabel 5.6 Spesifikasi Perangkat Lunak Komputer

HP Pavilion X360 13-a013d	
<i>Operating System</i>	Microsoft Windows 10 Pro 64bit
<i>Programming Language</i>	Java
<i>Programming Tool</i>	Android Studio 2.3.3
<i>Android SDK Tools</i>	26.0.1
<i>Gradle version</i>	26

Database Management System	Firestore realtime database SDK
----------------------------	---------------------------------

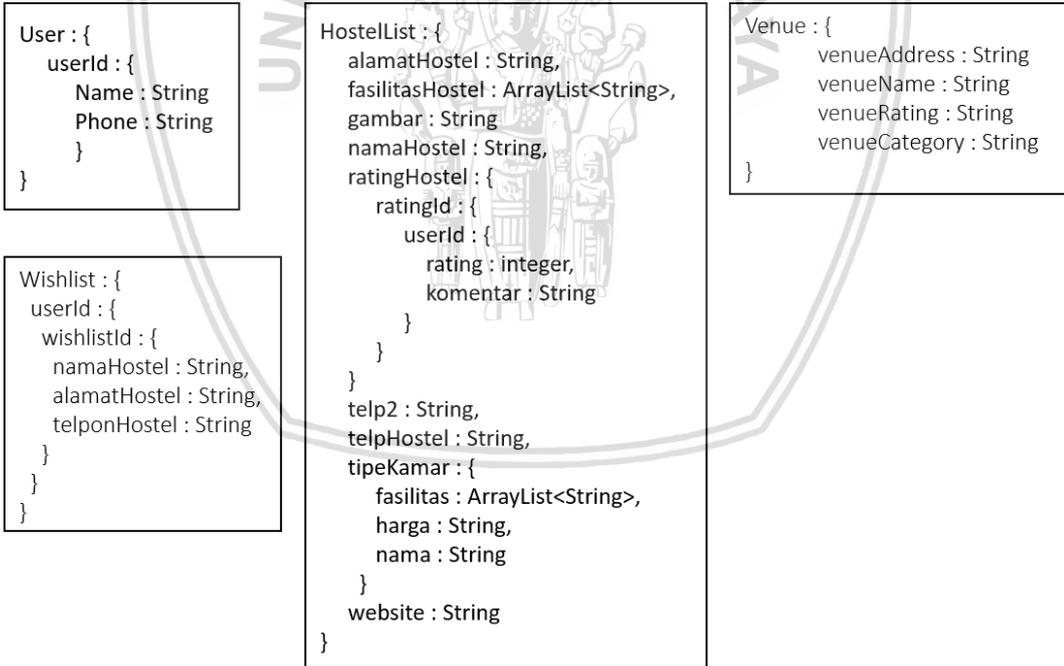
5.2.2 Batasan Implementasi

Batasan implementasi dalam pembuatan sistem informasi hostel adalah sebagai berikut:

1. Sistem informasi hostel ini dirancang dan diimplementasikan dengan menggunakan Android Studio.
2. Bahasa pemrograman yang digunakan adalah Java.
3. Manajemen basis data yang digunakan adalah *Firestore Realtime Database*.
4. Adanya sensor gyroscope pada perangkat ponsel milik pengguna.

5.2.3 Implementasi Database

Manajemen basis data yang digunakan pada pengembangan sistem informasi hostel di Kota Malang ini menggunakan *Firestore Realtime Database*. Implementasi *database* akan digambarkan dengan bentuk *tree* seperti pada gambar di bawah :



Gambar 5.44 Implementasi Database

5.2.4 Implementasi Kode Program

Pada sub bab ini akan menjelaskan tentang implementasi kode program pada method utama sistem informasi hostel di Kota Malang menggunakan Bahasa program Java. Serta diberikan penjelasan dari setiap baris kode program.

5.2.4.1 Implementasi Method whatsappInstalledOrNot

Method whatsappInstalledOrNot digunakan oleh sistem untuk menampilkan list hostel dari API Foursquare. Method ini ditunjukkan oleh Gambar dibawah ini:

```
1. private boolean whatsappInstalledOrNot(String uri) {
2.     PackageManager pm = DetailsActivity.this.getPackageManager();
3.     boolean app_installed = false;
4.     try {
5.         pm.getPackageInfo(uri, PackageManager.GET_ACTIVITIES);
6.         app_installed = true;
7.     } catch (PackageManager.NameNotFoundException e) {
8.         app_installed = false;
9.     }
10.    return app_installed;
11. }
```

Gambar 5.45 Implementasi Method whatsappInstalledOrNot

Penjelasan dari kode program method whatsappInstalledOrNot ditunjukkan pada tabel dibawah ini:

Tabel 5.7 Tabel Penjelasan Method whatsappInstalledOrNot

1.	Inisialisasi method whatsappInstalledOrNot dengan parameter String uri
2.	Deklarasi class PackageManager
4-6	Dilakukan pencarian, jika pada packageManager terdapat aplikasi whatsapp, maka mendeklarasikan nilai true pada app_installed.
7-8	Jika tidak ada aplikasi whatsapp, maka akan mendeklarasikan nilai false pada app_installed.
10	Mengembalikan nilai app installed

5.2.4.2 Implementasi Method InitComponent

```
1. String smsNumber = hostelPhone2.getText().toString();
2.     boolean isWhatsappInstalled =
3.     whatsappInstalledOrNot("com.whatsapp");
4.     if (isWhatsappInstalled) {
5.         try {
6.             Intent sendIntent = new
7.             Intent("android.intent.action.MAIN");
8.             sendIntent.setAction(Intent.ACTION_SEND);
9.             sendIntent.setType("text/plain");
10.            sendIntent.putExtra("jid", smsNumber +
11.            "@s.whatsapp.net"); //phone number without "+" prefix
12.            sendIntent.setPackage("com.whatsapp");
13.            startActivity(sendIntent);
14.        } catch (Exception e) {
15.            Toast.makeText(DetailsActivity.this, "Error/n" +
16.            e.toString(), Toast.LENGTH_SHORT).show();
17.        }
18.    } else {
19.        Uri uri =
20.        Uri.parse("market://details?id=com.whatsapp");
21.        Intent goToMarket = new Intent(Intent.ACTION_VIEW,
22.        uri);
23.        Toast.makeText(DetailsActivity.this, "WhatsApp not
24.        Installed",
25.            Toast.LENGTH_SHORT).show();
26.        DetailsActivity.this.startActivity(goToMarket);
```



```

27.     }
28.     }
29. }
30.
31.
32.
33.
34.
    
```

Gambar 5.46 Implementasi Method initComponents

Penjelasan dari kode program method initComponents ditunjukkan pada tabel dibawah ini:

Tabel 5.8 Tabel Penjelasan Method initComponents

1	Mendeklarasikan variable smsNumber yang diambil dari hostelPhone2 data
2-3	Menjalankan method whatsappInstalledOrNot yaitu melakukan pengecekan apakah aplikasi whatsapp sudah terinstall atau belum
4-17	Seleksi kondisi jika aplikasi sudah terinstall maka: Melakukan Intent ke aplikasi tersebut dan langsung disambungkan dengan nomorSms yang sudah dideklarasikan
19-26	Jika aplikasi whatsapp belum terinstall, maka mengambil nilai uri yaitu membuka aplikasi market jika ingin melakukan unduh aplikasi whatsapp

5.2.4.3 Implementasi Method InitData

```

1. private void InitData() {
2.     databaseReference =
3.     FirebaseDatabase.getInstance().getReference();
4.
5.     databaseReference.child("HostelList").addListenerForSingleValueEvent
6.     nt(new ValueEventListener() {
7.
8.         @Override
9.         public void onDataChange(DataSnapshot dataSnapshot) {
10.
11.
12.             for (DataSnapshot user :
13. dataSnapshot.getChildren()) {
14.                 String alamatHostel =
15. user.child("alamatHostel").getValue() + "";
16.                 Log.i("tes", (String)
17. user.child("alamatHostel").getValue());
18.                 ArrayList<String> fasilitasHostel =
19. (ArrayList<String>) user.child("fasilitasHostel").getValue();
20.                 list = new ArrayList<>();
21.                 list.clear();
22.                 for (DataSnapshot tes :
23. user.child("tipeKamar").getChildren()){
24.                     RoomType roomType =
25. tes.getValue(RoomType.class);
26.                     list.add(roomType);
27.                 }
28.                 String gambar =
29. user.child("gambar").getValue() + "";
30.                 String komentar =
31. user.child("komentar").getValue() + "";
32.
    
```

```

33.         String nameHostel =
34. user.child("namaHostel").getValue() + "";
35.         String ratingHostel =
36. user.child("ratingHostel").getValue() + "";
37.         String telp2 = user.child("telp2").getValue()
38. + "";
39.         String telpHostel =
40. user.child("telpHostel").getValue() + "";
41.         String website =
42. user.child("website").getValue() + "";
43.
44.         HostelData data = new HostelData(alamatHostel,
45. fasilitasHostel, gambar, komentar, nameHostel, ratingHostel,
46. telp2, telpHostel, website);
47.         if (namaHostel.equals(data.getNamaHostel())) {
48.
49.             hostelName.setText(data.getNamaHostel());
50.
51. hostelAddress.setText(data.getAlamatHostel());
52.             recyclerView.setAdapter(new
53. FacilityAdapter(fasilitasHostel));
54.             hostelWebsite.setText(data.getWebsite());
55.             recycleViewType.setAdapter(new
56. RoomTypeAdapter(getApplicationContext(), list));
57.             hostelPhone.setText(data.getTelpHostel());
58.             hostelPhone2.setText(data.getTelp2());
59.             break;
60.         }
61.     }
62. }
63.
64. @Override
65. public void onCancelled(DatabaseError databaseError) {
66.
67. }
68. });
}

```

Gambar 5.47 Implementasi Method initData

Penjelasan dari kode program method `initData` ditunjukkan pada tabel dibawah ini:

Tabel 5.9 Tabel Penjelasan Method initData

1	Mendeklarasikan DatabaseReference
2-6	Memanggil method <code>addListenerForSingleValueEvent</code> pada data yang ada di <code>child Hostel</code>
8-29	Method untuk mengambil nilai <code>alamatHostel</code> , <code>fasilitasHostel</code> , <code>gambar</code> , <code>harga</code> , <code>komentar</code> , <code>nameHostel</code> , <code>ratingHostel</code> , <code>telp2</code> , <code>telpHostel</code> , <code>tipeKamar</code> dan <code>website</code>
31-34	Mendeklarasikan method <code>HostelData</code> dengan nama data yang berisi parameter <code>alamatHostel</code> , <code>fasilitasHostel</code> , <code>gambar</code> , <code>harga</code> , <code>komentar</code> , <code>nameHostel</code> , <code>ratingHostel</code> , <code>telp2</code> , <code>telpHostel</code> , <code>tipeKamar</code> dan <code>website</code>
35-42	Seleksi kondisi jika <code>namaHostel</code> yang dipilih sama dengan <code>nameHostel</code> pada data, maka mengambil variable data dan menyimpannya pada variable <code>hostelName</code> , <code>hostelAddress</code> , <code>hostelWebsite</code> , <code>hostelPhone</code> , <code>hostelPrice</code> dan <code>hostelPhone2</code> untuk ditampilkan pada halaman <code>detailsActivity</code>

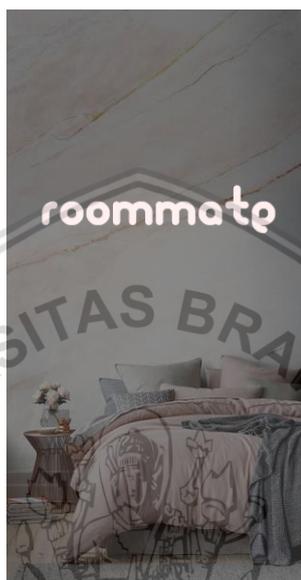


5.2.5 Implementasi Antarmuka

Pada sub bab ini merupakan bagian yang menjelaskan hasil dari implementasi antarmuka yang sudah dibuat sesuai dengan perancangan.

1. Implementasi Halaman *Splashscreen*

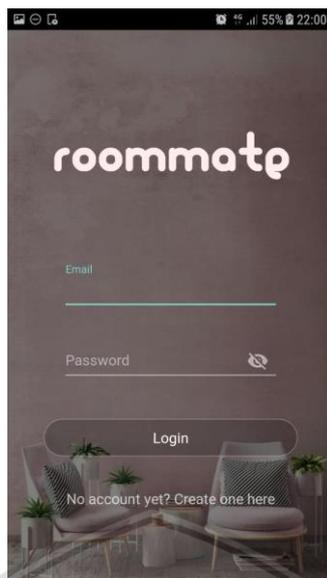
Implementasi dari halaman *splashscreen* bisa dilihat pada Gambar 5.47. ketika pertama kali pengguna membuka aplikasi, halaman inilah yang pertama kali muncul.



Gambar 5.48 Implementasi Halaman Splashscreen

2. Implementasi Halaman *Login*

Implementasi dari halaman *login* bisa dilihat pada Gambar 5.48. Setelah halaman *splashscreen*, pengguna langsung masuk pada halaman *Login*. Pengguna harus memasukkan alamat *email* dan *password* jika telah melakukan registrasi. Ketika pengguna belum melakukan registrasi, maka pengguna harus memilih untuk membuat akun baru pada bagian bawah halaman.



Gambar 5.49 Implementasi Halaman *Login*

3. Implementasi Halaman *Register*

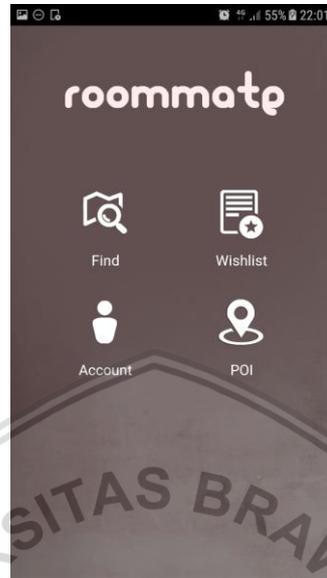
Implementasi dari halaman *register* bisa dilihat pada Gambar 5.49. Ketika pengguna memilih untuk membuat akun baru, maka pengguna akan masuk pada halaman registrasi. Pengguna harus memasukkan alamat *email* dan *password* untuk proses *autentifikasi*.



Gambar 5.50 Implementasi Halaman *Register*

4. Implementasi Halaman Menu Utama

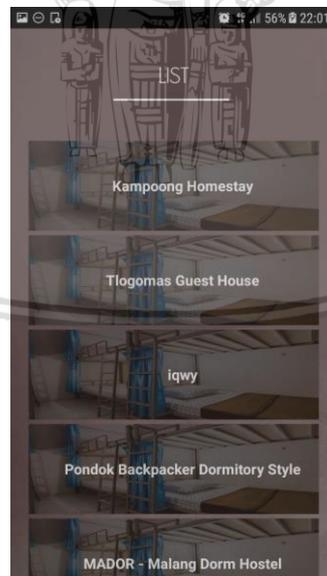
Implementasi dari halaman menu utama bisa dilihat pada Gambar 5.50. Pengguna dapat memilih menu yang ingin ditampilkan.



Gambar 5.51 Implementasi Halaman Menu Utama

5. Implementasi Halaman *List* Hostel

Implementasi halaman *list* hostel bisa dilihat pada Gambar 5.51. Desain *List* sudah terbentuk dengan isi nama hostel.

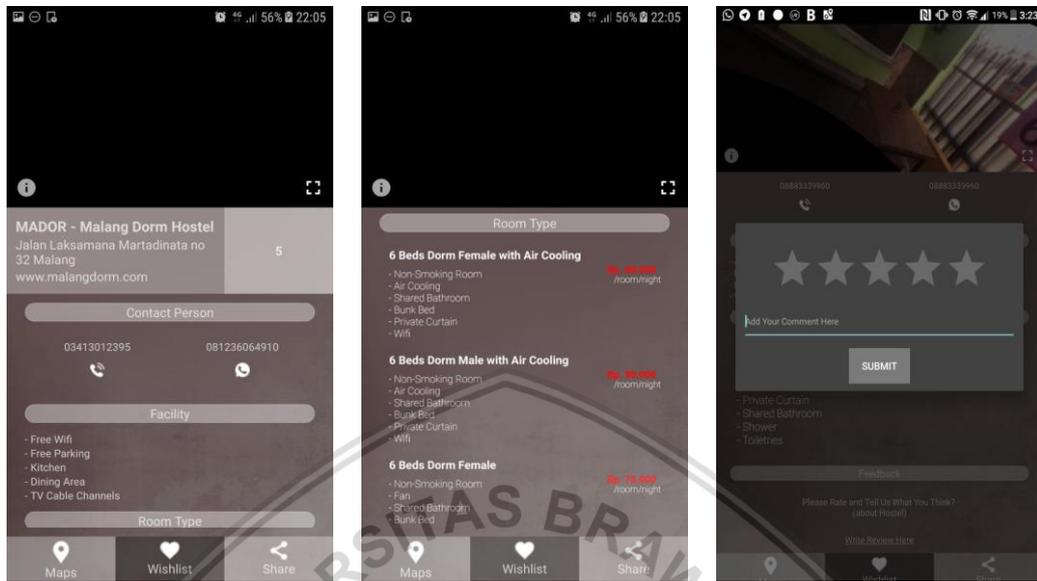


Gambar 5.52 Implementasi Halaman List Hostel

6. Implementasi Halaman Detail Hostel

Implementasi halaman detail hostel bisa dilihat pada gambar 5.52. Ketika pengguna telah memilih hostel pada list, pengguna akan masuk pada halaman

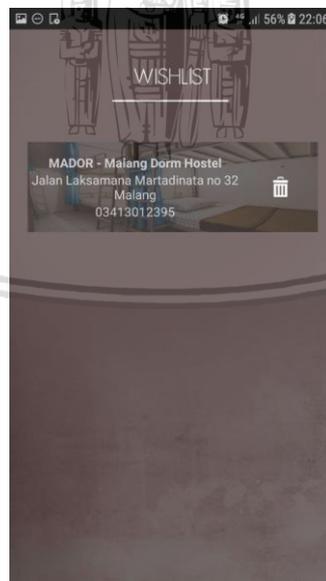
informasi dari hostel. Mencakup nama, alamat dan telepon. Pengguna juga dapat memberikan rating dan komentar untuk perkembangan hostel, dapat membagikan aplikasi tersebut ke media social.



Gambar 5.53 Implementasi Halaman Detail Hostel

7. Implementasi Halaman *Wishlist*

Implementasi halaman *wishlist* bisa dilihat pada gambar 5.53. Pada halaman ini berisi *list* nama hostel yang sudah disimpan pengguna ke dalam *wishlist*.

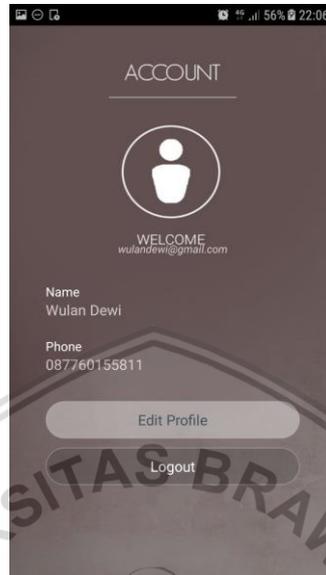


Gambar 5.54 Implementasi Halaman Wishlist

8. Implementasi Halaman Akun



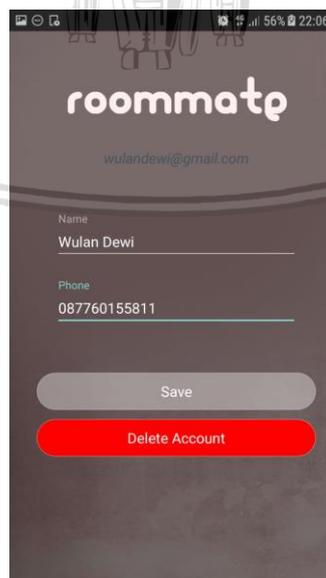
Implementasi halaman akun bisa dilihat pada gambar 5.54. Disini pengguna dapat melihat alamat email, nama dan nomor telepon yang sudah terdaftar pada *database*. Pengguna juga dapat melakukan *edit profile* dan *logout* pada halaman ini.



Gambar 5.55 Implementasi Halaman Akun

9. Implementasi Halaman *Edit Profile*

Implementasi halaman *edit profile* dapat dilihat pada gambar 5.55. Berisi *form* untuk mengisi nama dan nomor telepon, serta menghapus akun yang ada.



Gambar 5.56 Implementasi Halaman *Edit Profile*



10. Implementasi Halaman POI

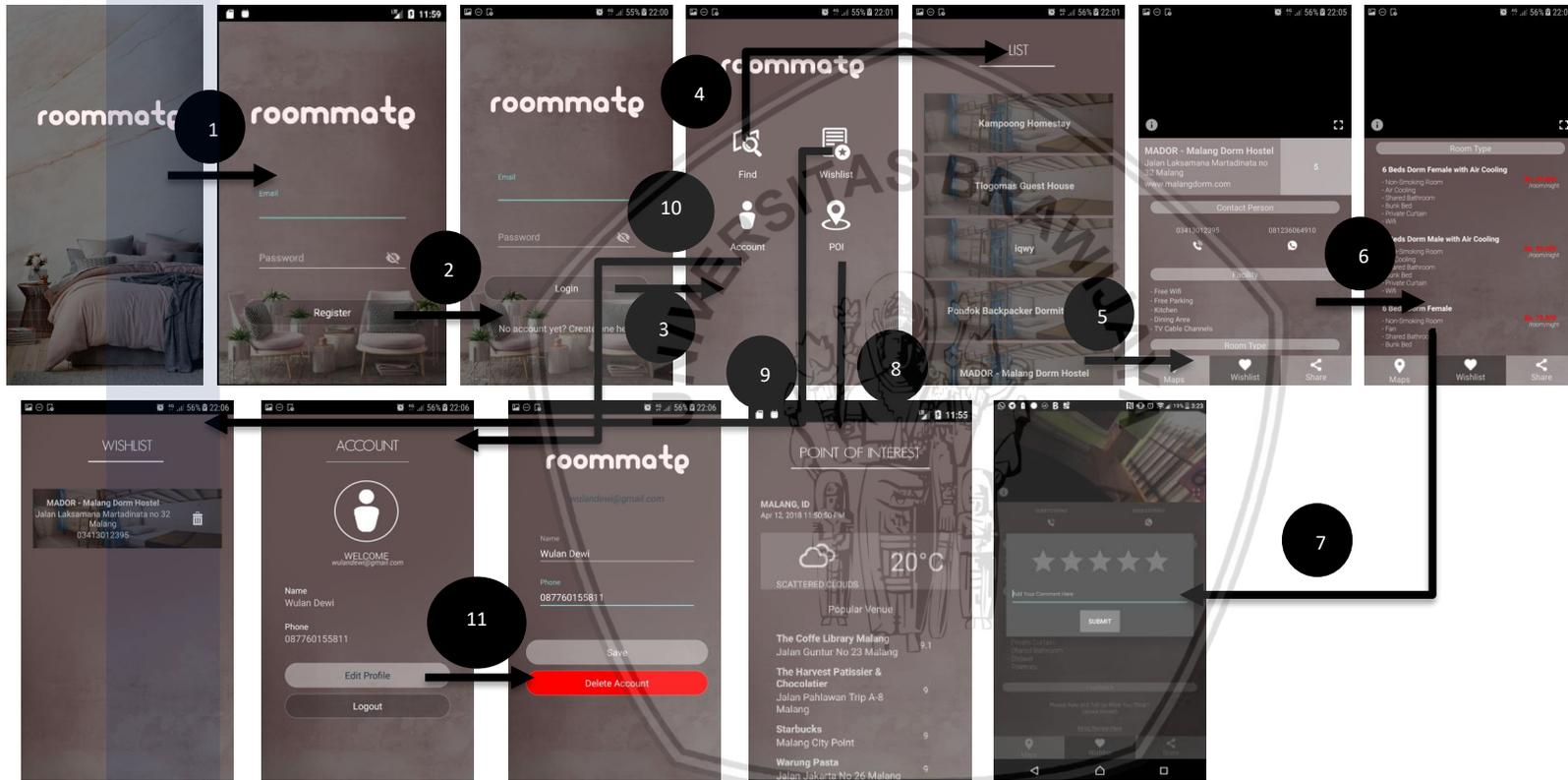
Implementasi halaman POI dapat dilihat pada Gambar 5.56. Berisi informasi cuaca saat ini sampai temperatur, beserta rekomendasi *popular venue* menurut rating sehingga pengguna bisa menentukan tempat untuk berkunjung.



Gambar 5.57 Implementasi Halaman POI



5.2.6 Implementasi Screenflow Antarmuka



Gambar 5.58 Implementasi Screenflow Aplikasi

Pada gambar 5.57 terdapat implementasi *screenflow* dari aplikasi Roommate. Dibawah ini akan dijelaskan bagaimana proses alur *screenflow* aplikasi Roommate:

1. Halaman *splashscreen* menuju ke halaman *register*.
2. Pada halaman *register* ketika pengguna telah mengisi *form* data yaitu alamat *email* dan *password* maka pengguna menekan tombol *register* untuk menuju halaman *login*.
3. Pada halaman *login* , pengguna mengisi *form* alamat *email* dan *password* yang telah didaftarkan sebelumnya lalu menekan tombol *login*. Ketika autentikasi data berhasil maka pengguna akan masuk pada halaman menu utama. Pada halaman ini terdapat empat menu yaitu *Find*, *Poi*, *Wishlist* dan *Account*.
4. Pengguna memilih menu *Find* dan langsung ditampilkan daftar dari semua hostel yang ada pada kota malang.
5. Pengguna memilih salah satu hostel yang ada dan menuju halaman details dari hostel yang dipilih.
6. Pengguna menggeserkan halaman kearah bawah untuk melihat fitur fitur yang ada pada halaman detail hostel.
7. Pengguna memilih pilihan untuk menulis komentar dan memberikan rating pada hostel, maka akan ditampilkan *action dialog* yang berisi *form* bagi pengguna untuk mengisikan rating dan komentar.
8. Ketika pengguna memilih menu *POI*, maka akan ditampilkan halaman baru yang berisi cuaca sekarang dan rekomendasi tempat populer.
9. Ketika pengguna memilih menu *Wishlist* maka akan ditampilkan halaman daftar *wishlist* hostel yang sudah disimpan oleh pengguna sebelumnya.
10. Sedangkan untuk pilihan menu *Account*, akan ditampilkan halaman dengan informasi profil pengguna dan dua tombol pilihan untuk melakukan edit profil dan *logout*.
11. Ketika pengguna memilih tombol *Edit Profil* maka akan ditampilkan halaman *edit profil* untuk mengubah nama dan nomor telepon pengguna. Serta terdapat dua pilihan tombol yaitu untuk melakukan simpan data dan menghapus akun.

BAB 6 PENGUJIAN

Pada bab ini dilakukan pengujian terhadap aplikasi yang telah dibuat berdasarkan perancangan dan implementasi yang mengacu pada Bab 5. Pengujian yang dilakukan adalah pengujian unit, pengujian validasi pengujian *usability* dan pengujian *compatibility*. Untuk hasil dari kuisioner pengujian telah terlampir pada halaman lampiran.

6.1 Pengujian Unit

Pengujian unit merupakan suatu teknik uji menggunakan *basis path testing* untuk menguji kode program berdasarkan algoritma pada setiap metode yang ada di klas. Proses pengujiannya adalah memodelkan algoritme menjadi *flow graph* dan mencari *cyclomatic complexity* untuk menentukan kompleksitas logika dari sebuah program, menentukan jalur independen dan memberikan kasus uji.

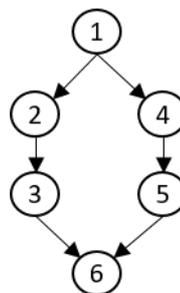
6.1.1 Pengujian Method *whatsappInstalledOrNot*

<pre> Private Boolean whatsappInstalledOrNot (String uri) { PackageManager pm = DetailsActivity.this.getPackageManager(); boolean app_installed = false; try { pm.getPackageInfo(uri, PackageManager.GET_ACTIVITIES); app_installed = true; } catch (PackageManager.NameNotFoundException e) { app_installed = false; } return app_installed; } </pre>	<div style="display: flex; flex-direction: column; align-items: center;"> <div style="display: flex; align-items: center; margin-bottom: 5px;"> } ① </div> <div style="display: flex; align-items: center; margin-bottom: 5px;"> — ② </div> <div style="display: flex; align-items: center; margin-bottom: 5px;"> } ③ </div> <div style="display: flex; align-items: center; margin-bottom: 5px;"> — ④ </div> <div style="display: flex; align-items: center; margin-bottom: 5px;"> — ⑤ </div> <div style="display: flex; align-items: center;"> — ⑥ </div> </div>
--	--

Gambar 6.1 Kode Method *whatsappInstalledOrNot*

Method *whatsappInstalledOrNot* merupakan suatu *method* yang digunakan dalam sistem untuk mengetahui apakah pada *package manager* terdapat aplikasi whatsapp. Gambar 6.1 menunjukkan kode program dari *method whatsappInstalledOrNot*.

Berdasarkan proses pengujian tersebut, Gambar 6.2 menunjukkan *flow graph* dari *method whatsappInstalledOrNot* yang berfungsi menggambarkan alur dari *method* tersebut.



Gambar 6.2 Flow Graph Method *whatsappInstalledOrNot*

Tahapan selanjutnya adalah menghitung *cyclomatic complexity*. Berdasarkan *flow graph method* diatas didapatkan perhitungan dengan menggunakan persamaan $V(G) = E - N + 2$, dimana $V(G)$ merupakan jumlah kompleksitas siklomatis, lalu E (*edge*) merupakan garis penghubung antar node dan N merupakan jumlah simpul (*node*).

$$\begin{aligned} V(G) &= 6 - 6 + 2 \\ &= 2 \end{aligned}$$

Setelah mendapatkan nilai dari *cyclomatic complexity*, maka didapatkan dua buah basis set jalur independent, yaitu:

Jalur 1 : 1 – 2 – 3 – 6

Jalur 2 : 1 – 4 – 5 – 6

Pengujian pada tiap jalur dilakukan dengan kasus uji yang ditunjukkan pada Tabel 6.1 dibawah ini :

Tabel 6.1 Kasus Uji Pengujian Unit *Method whatsappInstalledOrNot*

Jalur	Kasus Uji	Hasil yang Diharapkan	Hasil yang Didapatkan
1	Ketika pada packagemanager terdapat aplikasi whatsapp.	<i>Method</i> menginisialisasi nilai true pada Boolean <code>app_installed</code>	<i>Method</i> menginisialisasi nilai true pada Boolean <code>app_installed</code>
2	Ketika pada packagemanager tidak ada aplikasi whatsapp.	<i>Method</i> menginisialisasi nilai false pada <code>app_installed</code>	<i>Method</i> menginisialisasi nilai false pada <code>app_installed</code>

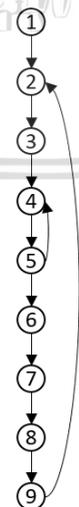
6.1.2 Pengujian Method InitData

<pre>private void InitData() { databaseReference = FirebaseDatabase.getInstance().getReference(); databaseReference.child("HostelList").addListenerForSingleValueEvent(new ValueEventListener() { @Override public void onDataChange(DataSnapshot dataSnapshot) { for (DataSnapshot user : dataSnapshot.getChildren()) { String alamatHostel = user.child("alamatHostel").getValue() + ""; Log.i("tes", (String) user.child("alamatHostel").getValue()); ArrayList<String> fasilitasHostel = (ArrayList<String>) user.child("fasilitasHostel").getValue(); list = new ArrayList<>(); list.clear(); for (DataSnapshot tes : user.child("tipeKamar").getChildren()){ RoomType roomType = tes.getValue(RoomType.class); list.add(roomType); } String gambar = user.child("gambar").getValue() + ""; String komentar = user.child("komentar").getValue() + ""; String namaHostel = user.child("namaHostel").getValue() + ""; String ratingHostel = user.child("ratingHostel").getValue() + ""; String telp2 = user.child("telp2").getValue() + ""; String telpHostel = user.child("telpHostel").getValue() + ""; String website = user.child("website").getValue() + ""; HostelData data = new HostelData(alamatHostel, fasilitasHostel, gambar, komentar, nameHostel, ratingHostel, telp2, telpHostel, website); if (namaHostel.equals(data.getNamaHostel())) { hostelName.setText(data.getNamaHostel()); hostelAddress.setText(data.getAlamatHostel()); recyclerView.setAdapter(new FacilityAdapter(fasilitasHostel)); hostelWebsite.setText(data.getWebsite()); hostelRating.setText(data.getRatingHostel()); recycleViewType.setAdapter(new RoomTypeAdapter(getApplicationContext(), list)); hostelPhone.setText(data.getTelpHostel()); hostelPhone2.setText(data.getTelp2()); break; } } } } }</pre>	
---	--

Gambar 6.3 Kode Method InitData

Method InitData merupakan suatu *method* yang digunakan dalam sistem untuk mengetahui apakah pada *package manager* terdapat aplikasi whatsapp. Gambar 6.3 menunjukkan kode program dari *method* InitData.

Berdasarkan proses pengujian tersebut, Gambar 6.4 menunjukkan *flow graph* dari *method* InitData yang berfungsi menggambarkan alur dari *method* tersebut.



Gambar 6.4 Flow Graph Method InitData



Tahapan selanjutnya adalah menghitung *cyclomatic complexity*. Berdasarkan *flow graph method* diatas didapatkan perhitungan dengan menggunakan persamaan $V(G) = E - N + 2$, dimana $V(G)$ merupakan jumlah kompleksitas siklomatis, lalu E (*edge*) merupakan garis penghubung antar node dan N merupakan jumlah simpul (*node*).

$$V(G) = 10 - 9 + 2$$

$$= 3$$

Setelah mendapatkan nilai dari *cyclomatic complexity*, maka didapatkan tiga buah basis set jalur independent, yaitu:

Jalur 1 : 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9

Jalur 2 : 1 – 2 – 3 – 4 – 5 – 4 – 5 – 6 – 7 – 8 – 9

Jalur 3 : 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9

Pengujian pada tiap jalur dilakukan dengan kasus uji yang ditunjukkan pada Tabel 6.2 dibawah ini :

Tabel 6.2 Kasus Uji Pengujian Unit *Method InitData*

Jalur	Kasus Uji	Hasil yang Diharapkan	Hasil yang Didapatkan
1	Ketika variable namaHostel sama dengan getNamaHosel	Sistem menampilkan semua data menurut nama yang dipilih.	Sistem menampilkan semua data menurut nama yang dipilih.
2	Ketika data tipe kamar ada lebih dari satu	Sistem menampilkan semua daftar tipe kamar yang ada berdasarkan hostel yang dipilih.	Sistem menampilkan semua daftar tipe kamar yang ada berdasarkan hostel yang dipilih.
3	Ketika data hostel dan informasi detail lebih dari satu	Sistem memanggil dan menampilkan semua informasi detail berdasarkan hostel yang dipilih.	Sistem memanggil dan menampilkan semua informasi detail berdasarkan hostel yang dipilih.

6.2 Pengujian *Validasi*

Pengujian ini berfungsi untuk mengetahui apakah sistem yang telah dibangun sudah sesuai dengan kebutuhan fungsional yang sudah dirancang. Metode pengujian menggunakan *black box*.



a. Kasus Uji Registrasi Akun

Nama Kasus Uji	Registrasi Akun
Tujuan Pengujian	Untuk membuktikan bahwa sistem telah menyimpan <i>email</i> dan <i>password</i> kedalam <i>database</i> .
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Memasukkan <i>email</i> dan <i>password</i>. 2. Menekan tombol register.
Hasil yang Diharapkan	<i>Email</i> dan <i>password</i> telah terdaftar pada <i>database</i> .

b. Kasus Uji Login Akun

Nama Kasus Uji	Login Akun
Tujuan Pengujian	Untuk membuktikan bahwa sistem telah melakukan autentikasi pada <i>email</i> dan <i>password</i> yang telah dimasukkan oleh pengguna.
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Memasukkan <i>email</i> dan <i>password</i>. 2. Menekan tombol <i>login</i>.
Hasil yang Diharapkan	<i>Email</i> dan <i>password</i> telah berhasil di autentikasi oleh sistem dan pengguna masuk pada halaman utama.

c. Kasus Uji Menambah Daftar Keinginan

Nama Kasus Uji	Menambah Daftar Keinginan
Tujuan Pengujian	Untuk membuktikan bahwa sistem telah menyimpan daftar keinginan pada <i>database</i> menurut <i>userId</i> .
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Memilih tombol <i>wishlist</i> pada halaman informasi hostel. 2. Melihat halaman <i>wishlist</i>.
Hasil yang Diharapkan	Daftar keinginan yang telah dipilih oleh pengguna muncul pada halaman <i>wishlist</i> .

d. Kasus Uji Melihat Daftar Hostel

Nama Kasus Uji	Melihat Daftar Hostel
Tujuan Pengujian	Untuk membuktikan bahwa sistem dapat menampilkan daftar hostel pada halaman list.
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Memilih menu List pada halaman menu utama. 2. Melihat daftar hostel.

Hasil yang Diharapkan	Daftar hostel muncul pada halaman list.
------------------------------	---

e. Kasus Uji Melihat Informasi Hostel

Nama Kasus Uji	Melihat Informasi Hostel
Tujuan Pengujian	Untuk membuktikan bahwa sistem dapat menampilkan informasi hostel ketika pengguna selesai memilih hostel yang diinginkan.
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Memilih salah satu hostel pada daftar yang ada. 2. Melihat informasi lengkap hostel tersebut pada halaman baru.
Hasil yang Diharapkan	Informasi lengkap hostel muncul pada halaman baru sesuai dengan nama hostel yang dipilih.

f. Kasus Uji Melihat Rute

Nama Kasus Uji	Melihat Rute
Tujuan Pengujian	Untuk membuktikan bahwa sistem dapat menampilkan halaman <i>Maps</i> dengan rute tujuan alamat hostel yang dipilih.
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Menekan tombol <i>maps</i> yang ada pada bawah halaman. 2. Melihat <i>Google Maps</i> dengan rute tujuan alamat hostel.
Hasil yang Diharapkan	Membuka <i>Google Maps</i> dengan rute tujuan alamat hostel.

g. Kasus Uji Membagikan ke Media Sosial

Nama Kasus Uji	Membagikan ke Media Sosial
Tujuan Pengujian	Untuk membuktikan bahwa pengguna dapat membagikan informasi tentang hostel yang dipilih ke media sosial.
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Pengguna memilih tombol <i>share</i> pada bagian bawah halaman. 2. Pengguna memilih media sosial yang dituju. 3. Pengguna membagikan informasi hostel kepada pengguna lain melalui media sosial yang dipilih.

Hasil yang Diharapkan	Informasi hostel yang dipilih oleh pengguna telah dibagikan pada pengguna lain melalui media sosial yang dipilih.
------------------------------	---

h. Kasus Uji Menulis Komentar

Nama Kasus Uji	Menulis Komentar
Tujuan Pengujian	Untuk membuktikan bahwa komentar yang diisikan oleh pengguna telah tersimpan pada database.
Prosedur Pengujian	1. Pengguna menulis komentar pada <i>form</i> yang telah disediakan. 2. Sistem menyimpan data komentar pada <i>database</i> .
Hasil yang Diharapkan	Komentar telah tersimpan pada <i>database</i> .

i. Kasus Uji Memberikan Rating

Nama Kasus Uji	Memberikan Rating
Tujuan Pengujian	Untuk membuktikan bahwa rating yang telah diisikan oleh pengguna telah tersimpan pada <i>database</i> .
Prosedur Pengujian	1. Pengguna memilih rating yang ingin diberikan. 2. Sistem menyimpan data rating pada <i>database</i> .
Hasil yang Diharapkan	Rating telah tersimpan pada database.

j. Kasus Uji Melihat *Point Of Interest*

Nama Kasus Uji	Melihat <i>Point Of Interest</i>
Tujuan Pengujian	Untuk membuktikan bahwa sistem dapat menampilkan <i>popular venue</i> berdasarkan cuaca pada saat ini.
Prosedur Pengujian	1. Pengguna memilih halaman POI pada halaman utama.
Hasil yang Diharapkan	Pengguna melihat tampilan waktu sekarang, cuaca, suhu serta <i>popular venue</i> yang sudah divilter berdasarkan cuaca pada waktu sekarang.

k. Kasus Uji Melihat Akun

Nama Kasus Uji	Melihat Akun
Tujuan Pengujian	Untuk membuktikan bahwa sistem dapat menampilkan informasi akun berdasarkan <i>email</i> dan <i>password</i> yang telah melakukan <i>login</i> .
Prosedur Pengujian	1. Pengguna memilih halaman <i>account</i> pada halaman utama.
Hasil yang Diharapkan	Pengguna melihat informasi akun yaitu nama dan nomor telepon pengguna.

l. Kasus Uji Melakukan Ubah Profil

Nama Kasus Uji	Melakukan Ubah Profil
Tujuan Pengujian	Untuk membuktikan bahwa pengguna dapat melakukan pembaharuan akun.
Prosedur Pengujian	1. Memilih tombol <i>account</i> pada halaman utama. 2. Memilih tombol <i>edit profile</i> . 3. Menuliskan nama pengguna. 4. Menuliskan nomor telepon pengguna.
Hasil yang Diharapkan	Sistem menyimpan data baru pengguna dan menampilkan halaman akun

m. Kasus Uji Melihat Rating

Nama Kasus Uji	Melihat Rating
Tujuan Pengujian	Untuk membuktikan bahwa pengguna dapat melihat rata-rata rating hostel.
Prosedur Pengujian	1. Memilih hostel yang diinginkan pada halaman <i>list</i> .
Hasil yang Diharapkan	Sistem menampilkan rata-rata rating pada halaman informasi hostel.

n. Kasus Uji Menghapus Akun

Nama Kasus Uji	Menghapus Akun
Tujuan Pengujian	Untuk membuktikan bahwa akun sudah terhapus dari <i>database</i> .
Prosedur Pengujian	1. Memilih tombol <i>account</i> pada halaman utama. 2. Memilih tombol <i>edit profile</i> .

	3. Memilih tombol <i>delete account</i> .
Hasil yang Diharapkan	Akun telah terhapus dari <i>database</i> .

o. Kasus Uji Menghapus Daftar Keinginan

Nama Kasus Uji	Menghapus Daftar Keinginan
Tujuan Pengujian	Untuk membuktikan bahwa daftar keinginan telah terhapus dari <i>database</i> .
Prosedur Pengujian	1. Memilih tombol <i>wishlist</i> pada halaman utama. 2. Memilih tombol hapus pada hostel yang diinginkan.
Hasil yang Diharapkan	Daftar keinginan telah terhapus dari <i>database</i> .

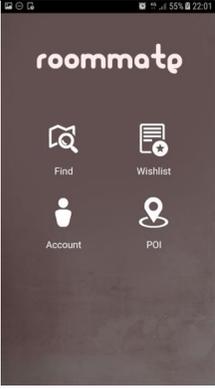
p. Kasus Uji Melihat Foto Panorama

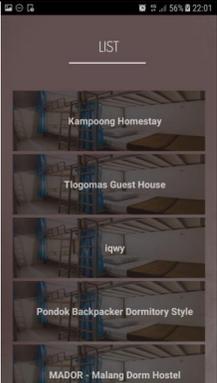
Nama Kasus Uji	Melihat Foto Panorama
Tujuan Pengujian	Untuk membuktikan bahwa pengguna dapat melihat foto panorama 360°.
Prosedur Pengujian	1. Pengguna memilih hostel yang diinginkan pada halaman <i>list</i> .
Hasil yang Diharapkan	Pengguna melihat halaman galeri pada halaman informasi hostel.

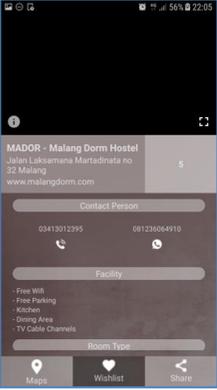
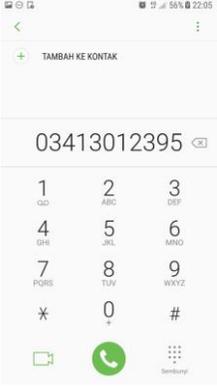
q. Kasus Uji *Logout* Akun

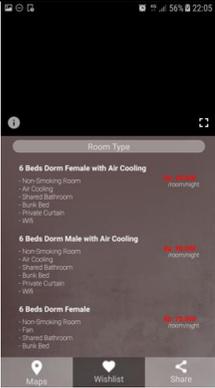
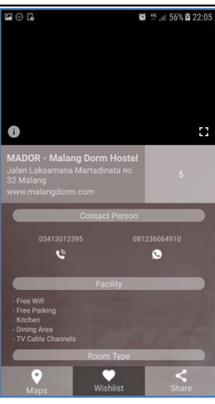
Nama Kasus Uji	<i>Logout</i> Akun
Tujuan Pengujian	Untuk membuktikan bahwa pengguna dapat keluar dari sistem.
Prosedur Pengujian	1. Menekan tombol <i>logout</i> pada halaman akun.
Hasil yang Diharapkan	Sistem menampilkan halaman <i>login screen</i> .

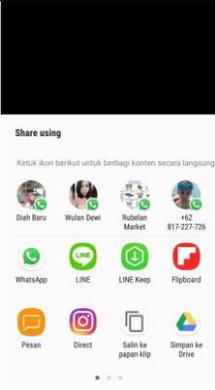
Tabel 6.3 Tabel Pengujian Validasi

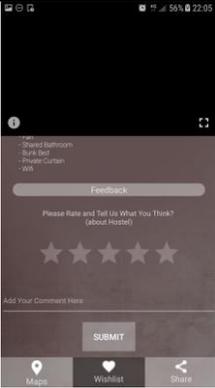
No	Nama Kasus Uji	Hasil yang diharapkan	Hasil yang didapatkan	Status validitas
1	Pengguna melakukan registrasi akun	Proses autentifikasi berhasil dan masuk ke halaman <i>login</i> .		Valid
2	Pengguna melakukan <i>login</i> akun	Autentikasi pengguna berhasil dan pengguna masuk kedalam halaman utama aplikasi.		Valid

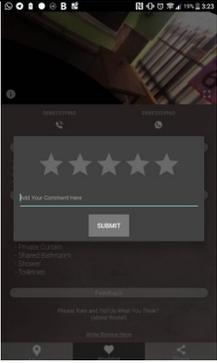
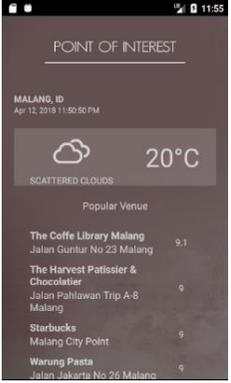
3	Pengguna melihat daftar hostel	Pengguna dapat melihat daftar hostel.		Valid
4	Pengguna melihat informasi dari hostel yang dipilih	Pengguna dapat melihat informasi lengkap tentang hostel yang dipilih.		Valid

<p>5</p>	<p>Pengguna melihat rating</p>	<p>Pengguna dapat melihat rating hostel.</p>		<p>Valid</p>
<p>6</p>	<p>Pengguna menghubungi pihak hostel</p>	<p>Pengguna dapat menghubungi pihak hostel melalui aplikasi whatsapp atau aplikasi telepon.</p>		<p>Valid</p>

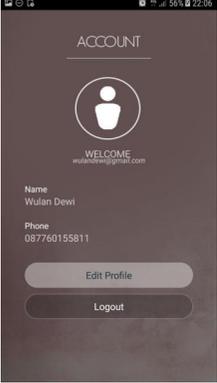
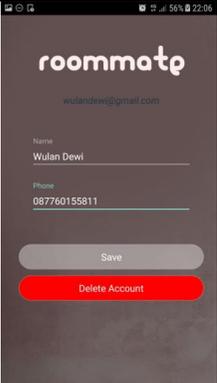
7	Pengguna melihat tipe kamar	Pengguna dapat melihat tipe kamar pada hostel beserta fasilitas dan harga kamar.		Valid
8	Pengguna melihat gambar dengan mode panoramaview	Pengguna dapat melihat gambar hostel dalam mode panorama 360 derajat.		Valid

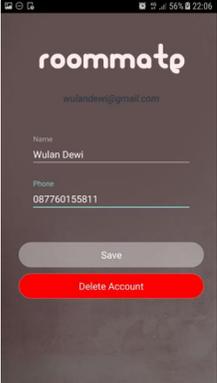
<p>9</p>	<p>Pengguna melihat rute dengan tujuan hostel</p>	<p>Pengguna dapat langsung mengetahui rute arah hostel yang dipilih dengan aplikasi google maps.</p>	 <p>MADOR - Malang Dorm... X</p> <p>Matahari Department Store Pasar Besar Malang</p> <p>Bank Syariah Mandiri KCP Malang Pasar Besar</p> <p>Toko Kain Nelly</p> <p>Toko Dewa - Dewi</p> <p>log.malang</p> <p>Tahu Lontong Lonceng Panca Buthi</p> <p>Toko H Santir</p> <p>MADOR Malang Dorm Hostel</p> <p>4.5 ★★★★★ (221)</p> <p>INFO SELENG... PETUNJUK ARAH</p>	<p>Valid</p>
<p>10</p>	<p>Pengguna membagikan ke media sosial</p>	<p>Pengguna dapat membagikan informasi hostel dan aplikasi melalui media sosial yang ada.</p>	 <p>Share using</p> <p>Ketuk ikon berikut untuk berbagi konten secara langsung.</p> <p>Dial Bero Wutan Dewi Robelen Market +62 817-227-726</p> <p>WhatsApp LINE LINE Keep Flipboard</p> <p>Pesan Direct Salin ke papan klip Simpan ke Drive</p>	<p>Valid</p>

11	Pengguna menulis komentar	Pengguna dapat menuliskan komentar yang ditujukan untuk hostel yang dipilih.		Valid
12	Pengguna menambah daftar keinginan	Pengguna dapat menambah hostel yang dipilih pada daftar keinginan.		Valid

13	Pengguna memberikan rating	Pengguna dapat memberikan rating yang ditujukan untuk hostel yang dipilih.		Valid										
14	Pengguna melihat <i>point of interest</i>	Pengguna dapat melihat tempat populer yang ada berdasarkan cuaca pada saat ini.	 <table border="1"> <thead> <tr> <th>Popular Venue</th> <th>Rating</th> </tr> </thead> <tbody> <tr> <td>The Coffe Library Malang Jalan Guntur No 23 Malang</td> <td>9.1</td> </tr> <tr> <td>The Harvest Patisserie & Chocolatier Jalan Pahlawan Trip A-8 Malang</td> <td>9</td> </tr> <tr> <td>Starbucks Malang City Point</td> <td>9</td> </tr> <tr> <td>Warung Pasta Jalan Jakarta No 26 Malang</td> <td>9</td> </tr> </tbody> </table>	Popular Venue	Rating	The Coffe Library Malang Jalan Guntur No 23 Malang	9.1	The Harvest Patisserie & Chocolatier Jalan Pahlawan Trip A-8 Malang	9	Starbucks Malang City Point	9	Warung Pasta Jalan Jakarta No 26 Malang	9	Valid
Popular Venue	Rating													
The Coffe Library Malang Jalan Guntur No 23 Malang	9.1													
The Harvest Patisserie & Chocolatier Jalan Pahlawan Trip A-8 Malang	9													
Starbucks Malang City Point	9													
Warung Pasta Jalan Jakarta No 26 Malang	9													

15	Pengguna melihat daftar keinginan	Pengguna dapat melihat daftar keinginan yang telah disimpan.		Valid
16	Pengguna menghapus daftar keinginan	Pengguna dapat menghapus daftar keinginan dari halaman wishlist.		Valid

17	Pengguna melihat akun	Pengguna dapat melihat akun.		Valid
18	Pengguna melakukan edit profil	Pengguna dapat melakukan edit profil yaitu dengan menambah atau mengubah nama dan nomor telepon.		Valid

<p>19</p>	<p>Pengguna menghapus akun</p>	<p>Pengguna dapat menghapus akun ketika sudah tidak digunakan lagi.</p> 	 	<p>Valid</p>
-----------	--------------------------------	---	--	--------------

20	Pengguna melakukan logout aplikasi	Pengguna keluar dari aplikasi dan mengakhiri session aplikasi.	 A screenshot of a mobile application login screen. At the top, the word "roommate" is displayed in a white, lowercase, sans-serif font. Below it are two input fields: "Email" and "Password". The "Password" field has a small eye icon to its right. A "Login" button is centered below the fields. At the bottom, a message reads "Authentication successful" with a green checkmark icon to its right. The background of the app is dark with a faint image of a living room.	Valid
----	------------------------------------	--	---	-------

6.3 Pengujian Usability

Tabel 6.4 Tabel Hasil Efektivitas

No	Elemen Efektivitas	Nama Persona					Hasil Usability (%)
		Ulul Albab K.	Yudha Sesarahditya	Sarah Diba N.	Farhesa Kamil	Ghifari D.	
Skenario melihat daftar hostel							
1	Pengguna berhasil menemukan list hostel	√	√	√	√	√	100%
2	Pengguna berhasil membuka list hostel dan melihat informasi hostel	√	√	√	√	√	100%
3	Pengguna berpindah ke menu utama dengan mudah	√	√	√	√	√	100%
Skenario edit profil							
4	Pengguna berhasil membuka menu edit profil	√	√	√	√	√	100%
5	Pengguna berhasil mengisikan nama dan nomor telepon baru	√	√	√	√	√	100%

6	Pengguna berpindah ke menu utama dengan mudah	√	√	-	-	√	60%
Skenario melihat <i>point of interest</i>							
7	Pengguna berhasil melihat cuaca saat ini	√	√	√	√	√	100%
8	Pengguna berhasil melihat rekomendasi <i>popular venue</i>	√	√	√	√	√	100%
9	Pengguna berpindah ke menu utama dengan mudah	√	√	-	-	√	60%
Scenario melihat wishlist							
10	Pengguna memilih icon wishlist pada hostel yang diinginkan	√	√	-	-	-	40%
11	Pengguna melihat list hostel yang dipilih di dalam halaman wishlist	√	√	√	√	√	100%
12	Pengguna berpindah ke menu utama dengan mudah	√	√	√	√	√	100%
Scenario memberikan rating dan komentar							
13	Pengguna berhasil mengisi rating	√	√	√	√	√	100%

14	Pengguna berhasil mengisikan komentar	√	√	√	√	√	100%
15	Pengguna berpindah ke menu utama dengan mudah	√	√	√	√	√	100%
Scenario membagikan ke sosial media							
16	Pengguna berhasil membagikan ke media social	√	√	√	√	√	100%
17	Pengguna berpindah ke menu utama dengan mudah	√	√	√	√	√	100%
Total Hasil							91.7%

Tabel 6.5 Tabel Hasil Efisiensi

No	Elemen Efisiensi	Nama Persona					Hasil Usability (%)
		Ulul Albab Kholili	Yudha Sesarahditya	Sarah Diba Nahdi	Farhesa Kamil	Ghifari Desomsoni	
Skenario melihat daftar hostel							
1	Frekuensi pengguna bertanya pada observator rendah	√	√	√	√	√	100%

2	Frekuensi bantuan dari observatory rendah	√	√	√	√	√	100%
Skenario edit profil							
1	Frekuensi pengguna bertanya pada observator rendah	√	√	√	√	√	100%
2	Frekuensi bantuan dari observatory rendah	√	√	√	√	√	100%
Skenario melihat <i>point of interest</i>							
1	Frekuensi pengguna bertanya pada observator rendah	√	√	√	√	√	100%
2	Frekuensi bantuan dari observatory rendah	√	√	√	√	√	100%
Skenario melihat wishlist							
1	Frekuensi pengguna bertanya pada observator rendah	√	√	√	√	√	100%
2	Frekuensi bantuan dari observatory rendah	√	√	√	√	√	100%
Skenario memberi rating dan komentar							

1	Frekuensi pengguna bertanya pada observator rendah	√	√	√	√	√	100%
2	Frekuensi bantuan dari observatory rendah	√	√	√	√	√	100%
Skenario membagikan ke media sosial							
1	Frekuensi pengguna bertanya pada observator rendah	√	√	√	√	√	100%
2	Frekuensi bantuan dari observatory rendah	√	√	√	√	√	100%
Total Hasil							100%

Tabel 6.6 Tabel Kepuasan Pengguna

No	Kriteria	Jawaban				
		SS	S	N	TS	STS
1	Saya merasa puas dengan betapa mudahnya menggunakan aplikasi ini	3	1	1	0	0
2	Aplikasi ini mudah untuk digunakan	3	1	1	0	0
3	Saya bisa menyelesaikan tugas dan scenario pada aplikasi ini	0	3	2	0	0
4	Saya merasa nyaman menggunakan aplikasi ini	5	0	0	0	0

5	Saya merasa aplikasi ini mudah untuk dipelajari	4	1	0	0	0
6	Saya bisa menjadi produktif dengan cepat menggunakan aplikasi ini	0	4	1	0	0
7	Aplikasi ini memberi pesan kesalahan yang jelas	0	2	0	1	0
8	Setiap kali membuat kesalahan, saya memperbaiki dengan cepat dan mudah	1	4	0	0	0
9	Informasi yang tersedia pada aplikasi ini sudah jelas	0	3	2	0	0
10	Saya mudah untuk menemukan informasi yang saya butuhkan	5	0	0	0	0
11	Informasi yang efektif membantu saya menyelesaikan tugas dan skenario	0	4	1	0	0
12	Organisasi informasi pada aplikasi ini jelas	5	0	0	0	0
13	Antarmuka pada aplikasi ini menyenangkan	5	0	0	0	0
14	Saya suka menggunakan antarmuka aplikasi ini	5	0	0	0	0
15	Aplikasi ini memiliki semua fungsi dan kemampuan yang saya harapkan	0	2	2	1	0
16	Secara keseluruhan saya merasa puas dengan aplikasi ini	0	5	0	0	0
Hasil		36	30	10	2	0

6.4 Pengujian *Compatibility*

Tabel 6.7 Tabel Hasil Uji Tes *Compatibility*

No	Perangkat	Screenshot Validasi	Status Validasi
1	Sony Xperia XA Ultra Dual <ul style="list-style-type: none"> - LCD : 6.0" (1080 x 1920) - OS : Android 7.0, API 24 - Sensor gyroscope : ada 		Valid
2	Samsung Galaxy S8 <ul style="list-style-type: none"> - LCD : 5.8" (1440 x 2960) - OS : Android 8.0, API 26 - Sensor gyroscope : ada 		Valid
3	Samsung Galaxy Note 3 <ul style="list-style-type: none"> - LCD : 5.7" (1080 x 1920) - OS : Android 4.4.2, API 17 - Sensor gyroscope : ada 		Valid

4	Xiaomi Redmi note 3 <ul style="list-style-type: none"> - LCD : 5.5" (720 x 1280) - OS : Android 5.0.2, API 21 - Sensor gyroscope : ada 		Valid
---	---	--	-------

Berdasarkan tabel di atas, terdapat empat macam perangkat yang digunakan untuk pengujian. Hasil dari pengujian tersebut selanjutnya dihitung persentase *compatibility*. Adapun perhitungannya dapat dilihat pada tabel berikut:

Pengujian	Nilai Maksimum	Berjalan	Gagal
Menjalankan Aplikasi pada perangkat android	4	4	0
Melihat gambar panorama 360°	4	4	0
Nilai Total		8	0

Rumus untuk menghitung aspek *compatibility* adalah sebagai berikut:

$$Compatibility = \frac{Nilai\ Total}{Nilai\ Maksimum} \times 100\%$$

Berdasarkan hasil pengujian pada tabel, dengan nilai total sebesar 8 dan nilai maksimum sebesar 8, maka dari rumus tersebut didapatkan persentase dari aspek *compatibility* sebagai berikut:

$$Compatibility = \frac{8}{8} \times 100\% = 100\%$$

6.5 Analisis Hasil Pengujian

Pada tahap ini menjelaskan hasil analisis dari beberapa pengujian yang dilakukan pada aplikasi Roommate yaitu analisis pengujian unit, analisis pengujian validasi, analisis pengujian *usability* dan analisis pengujian *compatibility*.

6.5.1 Analisis Hasil Pengujian Unit

Pengujian unit dilakukan dengan cara menggunakan metode *whitebox testing* dengan *basis path tesing*. Berdasarkan hal tersebut, hasil pengujian yang telah dilakukan maka dapat disimpulkan bahwa unit dari aplikasi Roommate sudah memenuhi kebutuhan fungsional yang telah dirancang.

6.5.2 Analisis Hasil Pengujian Validasi

Pengujian validasi digunakan dengan cara menggunakan metode *blackbox testing* dengan membandingkan hasil uji dan daftar kebutuhan yang ada. Jika telah sesuai, maka hasil tersebut mendapatkan nilai *valid* karena telah berhasil melakukan implementasi sesuai dengan perancangan kebutuhan yang diinginkan. Berdasarkan pernyataan tersebut, dapat disimpulkan bahwa hasil dari pengujian validasi aplikasi Roommate telah memenuhi kebutuhan fungsional.

6.5.3 Analisis Hasil Pengujian Usability

Hasil pengujian *usability* diambil dari rata-rata hasil efektivitas, efisiensi dan kepuasan pengguna. Dihitung menggunakan rumus yang mengacu pada bab 2 dengan nomor rumus (2.3), hasil yang didapatkan yaitu 92.06% sehingga dapat disimpulkan bahwa aplikasi Roommate ini mempunyai tingkat *usability* yang tinggi dan menjadi sebuah aplikasi yang *user-friendly*.

6.5.3.1 Analisis Hasil Pengujian Usability menurut Efektivitas

Mengacu pada rumus efektivitas di bab 2 dengan nomor rumus (2.1), hasil dari pengujian *usability* menurut efektivitas pengguna yaitu mendapatkan mayoritas hasil 100%. Sehingga mendapatkan hasil rata-rata dengan total 91.7%. Maka dapat disimpulkan bahwa performansi pengguna dalam menggunakan aplikasi Roommate sangat baik. Hal ini menunjukkan bahwa aplikasi Roommate dapat membantu sesuai kebutuhan pengguna dengan baik.

6.5.3.2 Analisis Hasil Pengujian Usability menurut Efisiensi

Mengacu pada rumus efisiensi di bab 2 dengan nomor rumus (2.1), Hasil dari pengujian *usability* menurut efisiensi pengguna memiliki nilai dengan rata 100%. Untuk tingkat efisiensi dapat disimpulkan bahwa aplikasi Roommate sangat efisien untuk melakukan tindakan sesuai kebutuhan pengguna dengan baik.

6.5.3.3 Analisis Hasil Pengujian Usability menurut Kepuasan Pengguna

Pengujian ini menggunakan rumus Skala Linkert yang mengacu pada bab 2 dengan nomor rumus (2.2). Presentasi hasil dari pengujian ini dapat dilihat pada tabel 6.8

Tabel 6.8 Presentasi Hasil Pengujian Kepuasan Pengguna

No	Kriteria	Hasil Presentase

1	Saya merasa puas dengan betapa mudahnya menggunakan aplikasi ini	88%
2	Aplikasi ini mudah untuk digunakan	88%
3	Saya bisa menyelesaikan tugas dan scenario pada aplikasi ini	72%
4	Saya merasa nyaman menggunakan aplikasi ini	100%
5	Saya merasa aplikasi ini mudah untuk dipelajari	96%
6	Saya bisa menjadi produktif dengan cepat menggunakan aplikasi ini	76%
7	Aplikasi ini memberi pesan kesalahan yang jelas	56%
8	Setiap kali membuat kesalahan, saya memperbaiki dengan cepat dan mudah	84%
9	Informasi yang tersedia pada aplikasi ini sudah jelas	72%
10	Saya mudah untuk menemukan informasi yang saya butuhkan	100%
11	Informasi yang efektif membantu saya menyelesaikan tugas dan skenario	76%
12	Organisasi informasi pada aplikasi ini jelas	100%
13	Antarmuka pada aplikasi ini menyenangkan	100%
14	Saya suka menggunakan antarmuka aplikasi ini	100%
15	Aplikasi ini memiliki semua fungsi dan kemampuan yang saya harapkan	64%
16	Secara keseluruhan saya merasa puas dengan aplikasi ini	80%
Hasil Rata-rata		84,5%

Menurut kriteria interpretasi skor pada tabel 2.3, hasil rata-rata pada angka 84.5% dikategorikan sebagai Sangat Kuat. Maka disimpulkan bahwa pengguna sangat puas pada aplikasi Roomate tersebut.

6.5.4 Analisis Hasil Pengujian *Compatibility*

Berdasarkan hasil pengujian, persentase aspek *compatibility* aplikasi Roomate dapat diketahui yaitu 100%. Sehingga dapat disimpulkan bahwa aplikasi Roomate telah memenuhi standar dalam aspek *compatibility*.

BAB 7 PENUTUP

Pada bab ini berisi kesimpulan yang akan menjawab rumusan masalah yang telah dijelaskan pada bab sebelumnya yaitu bab pendahuluan, serta saran dari penulis yang dapat digunakan untuk pengembangan aplikasi ini lebih lanjut.

7.1 Kesimpulan

Berdasarkan pengujian dan analisis yang dilakukan pada penelitian ini, maka dapat disimpulkan bahwa diantaranya yaitu:

1. Proses pembangunan aplikasi sistem informasi berbasis android dengan metode *User Centered Design* ini dibagi menjadi empat alur utama. Dua alur pertama yang dilakukan pada bab analisis kebutuhan yaitu *specify the context of use* yang dilakukan adalah identifikasi aktor, *specify user and organizational requirement* yang dilakukan adalah mengidentifikasi kebutuhan calon pengguna. Untuk dua alur terakhir dilakukan pada bab perancangan yaitu *product design solution* yang dilakukan dengan cara membuat perancangan antarmuka dari hasil analisis kebutuhan sebelumnya, sedangkan untuk alur terakhir yaitu *evaluate design against requirement* yang dilakukan oleh calon pengguna untuk memperbaiki sistem dengan dilakukannya iterasi.
2. Untuk pembangunan aplikasi sistem informasi hostel ini menggunakan data yang didapatkan melalui Foursquare API. Data yang didapatkan mencakup nama-nama hostel yang terdaftar di Kota Malang serta tempat populer berdasarkan rating pengguna Foursquare yang digunakan untuk membuat rekomendasi tempat populer berdasarkan cuaca saat ini pada aplikasi.
3. Pembangunan aplikasi sistem informasi roommate dilakukan pengujian *usability* dengan tiga kriteria yaitu efektivitas, efisiensi dan tingkat kepuasan pengguna. Untuk kriteria tingkat kepuasan digunakan kuisisioner PSSUQ sebagai acuan. Dari analisis hasil pengujian yang dilakukan, didapatkan hasil pada kriteria efektivitas sebesar 91.7%, kriteria efisiensi sebesar 100% dan tingkat kepuasan pengguna sebesar 84.5%. Jika dihitung rata-rata, maka nilai akhir hasil pengujian *usability* yaitu 92.6% yang dapat disimpulkan bahwa aplikasi sistem informasi Roommate mempunyai tingkat *usability* yang tinggi dan menjadi sebuah aplikasi *user-friendly*.

7.2 Saran

Berdasarkan hasil penelitian yang dilakukan, penulis dapat menyampaikan saran untuk penelitian selanjutnya diantaranya sebagai berikut:

1. Informasi hostel yang diberikan lebih lengkap lagi agar memudahkan calon pengguna mencari apa yang diinginkan.

2. Diberikan galeri foto untuk fitur panorama 360° agar pengguna lebih leluasa melihat interior dan eksterior hostel yang ada.
3. Dilakukan penyaringan pada data *output* dari Foursquare API agar sesuai dengan konteks bahasan utama.



DAFTAR PUSTAKA

Amadyah, A., 2008. Rancangan Sistem Pameran Online Menggunakan Metode UCD (User Centered Learning). *SNTS AKPRIND*.

Anon., 2018. *Developers*. [Online] Available at: <https://developer.android.com/guide/practices/compatibility> [Accessed 12 7 2018].

Barnum, C. M., 2011. *Usability Testing Essentials: Ready, Set Test!*. s.l.:s.n.

Budiaji, W., 2013. SKALA PENGUKURAN DAN JUMLAH RESPON SKALA LIKERT. *Jurnal Ilmu Pertanian dan Perikanan*, Volume 2, pp. 127-133.

Devi, K. R., Sen, A. & Hemachandran, K., 2012. A working Framework for the User-Centered Design Approach and a Survey of the available Methods. *International Journal of Scientific and Research Publications*, 2(4).

Dumas, J. S. & Fox, J. E., 2008. USABILITY TESTING: CURRENT PRACTICE AND FUTURE DIRECTIONS. In: *The Human-Computer Interaction Handbook (2nd Edition)*. USA: s.n., pp. 1129-1149.

Likert, R., 1932. Technique for the measurement of attitudes. *Archives of Psychology*, Volume 140, pp. 1-55.

Liu, Z., Hu, Y. & Cai, L., 2014. Research on Software Security and Compatibility Test for Mobile Application. *Shanghai Key Laboratory of Computer Software Testing and Evaluation*.

Mulia, A. H., Priharsari, D. & Wardani, N. H., 2016. PENGEMBANGAN ANTARMUKA APLIKASI ASTOFIT DENGAN PENDEKATAN USER CENTERED DESIGN. *jurnal ptiik ub*, Volume 7.

Nielsen, J., 2012. *Nielsen Norman Group Evidence-Based User Experience Research, Training and Consulting*. [Online] Available at: <https://www.nngroup.com/articles/how-many-test-users/> [Accessed 12 July 2018].

Publik, B. I., 2017. *Kota Malang Optimis Jumlah Kunjungan Wisman Meningkat*. [Online] Available at: <https://malangkota.go.id/2017/08/07/kota-malang-optimis-jumlah-kunjungan-wisman-meningkat/> [Accessed 7 3 2018].

Saputri, I. S. Y., Fadhli, M. & Surya, I., 2017. Penerapan Metode UCD (User Centered Design) pada E-Commerce Putri Intan Shop Berbasis Web. *Jurnal Teknologi dan Sistem Informasi*.

Sauro, J. & Lewis, J. R., 2014. When Designing Usability Questionnaires, Does It Hurt to Be Positive?.

Sharma, J., Singh, S. P. & Singh, P., 2011. Location Based Information Delivery in Tourism. *IEEE International Conference on Advanced Computing & Communication Technologies*.

Singhal, M. & Shukla, A., 2012. Implementation of Location based Services in Android using GPS and Web Services. *IJCSI International Journal of Computer Science Issues*, 9(1).

STAPIC, Z., 2013. A PROPOSAL OF AN ONTOLOGY-BASED METHODOLOGICAL FRAMEWORK FOR MULTI-PLATFORM MOBILE APPLICATIONS DEVELOPMENT.

Triadi, D., 2013. *Bedah Tuntas Fitur Android*. 1 ed. Yogyakarta: Jogja Great! Publisher.

Zahara, 2013. PERANCANGAN APLIKASI E-COMMERCE PENJUALAN SPARE PART FORKLIFT DENGAN METODE USER CENTERED DESIGN (UCD). *Pelita Informatika Budi Darma*, V(2).

