



**RANCANG BANGUN APLIKASI PENCARIAN LOKASI INDEKOS DI  
KOTA JEMBER MENGGUNAKAN *LOCATION BASED SERVICE* (LBS)  
DAN ALGORITMA *BEE COLONY OPTIMIZATION* (BCO)**

**SKRIPSI**

Oleh

**Mohammad A'la Maududy**

**NIM 122410101045**

**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS JEMBER**

**2020**



**RANCANG BANGUN APLIKASI PENCARIAN LOKASI INDEKOS DI  
KOTA JEMBER MENGGUNAKAN *LOCATION BASED SERVICE* (LBS)  
DAN ALGORITMA *BEE COLONY OPTIMIZATION* (BCO)**

**SKRIPSI**

diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat untuk menyelesaikan Pendidikan Sarjana (S1) Fakultas Ilmu Komputer Universitas Jember dan mencapai gelar Sarjana Komputer

Oleh

**Mohammad A'la Maududy**

**NIM 122410101045**

**FAKULTAS ILMU KOMPUTER**

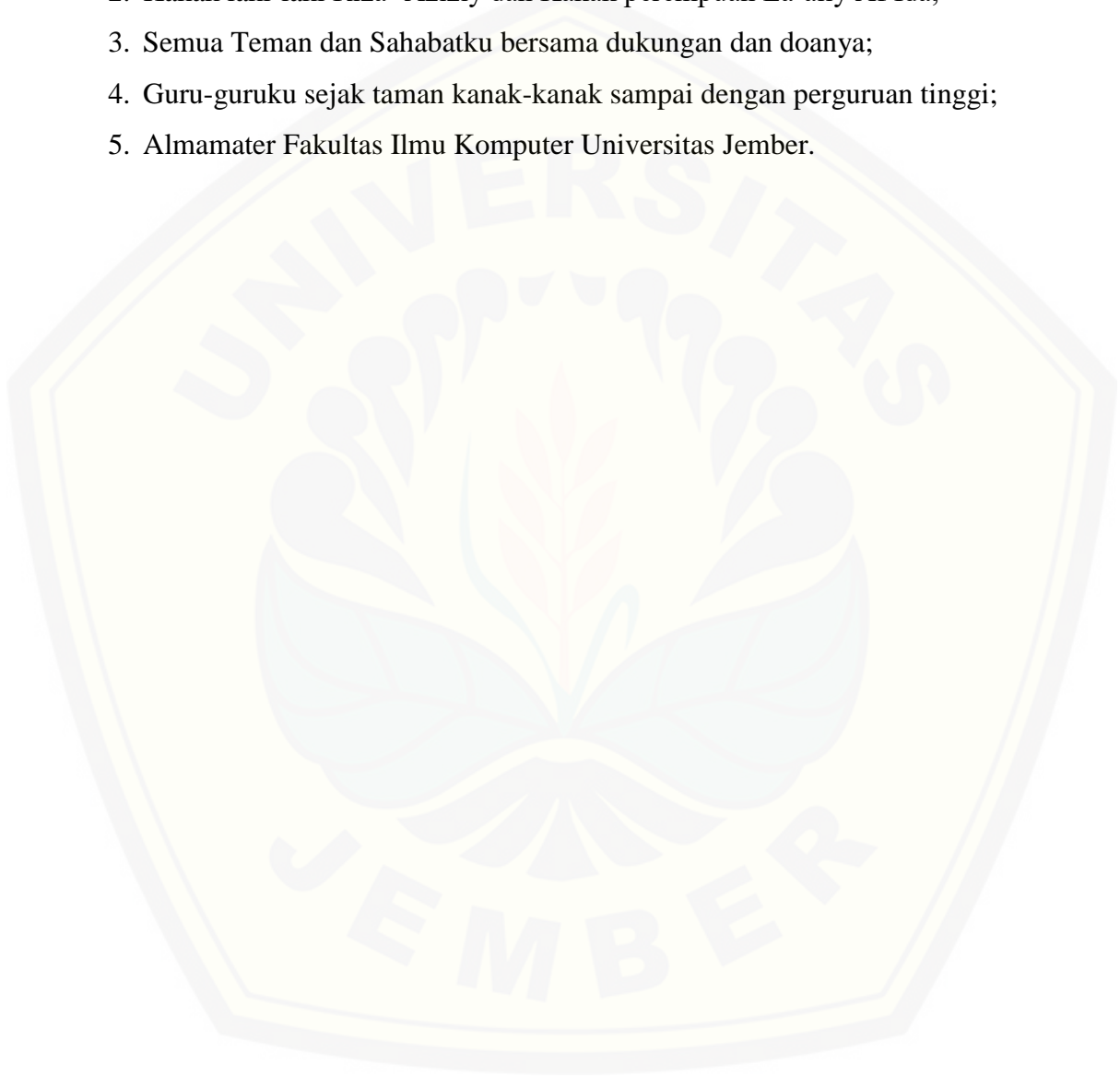
**UNIVERSITAS JEMBER**

**2020**

## PERSEMBAHAN

Skripsi ini saya persembahkan untuk:

1. Ayahanda Mohammad Syamsul Hadi dan Ibunda Ulfiatul Hayati tercinta;
2. Kakak laki-laki Riza 'Aziziy dan Kakak perempuan La'aliy Af Ida;
3. Semua Teman dan Sahabatku bersama dukungan dan doanya;
4. Guru-guruku sejak taman kanak-kanak sampai dengan perguruan tinggi;
5. Almamater Fakultas Ilmu Komputer Universitas Jember.



**MOTO**

“DZIKIRLAH AKAN DAKU, SUPAYA AKU DZIKIR PULA AKAN DIKAU;  
BERSYUKURLAH AKAN DAKU DAN JANGAN INKARI NIKMAT-KU”.

*(QS. Al-Baqarah : 152)*



**PERNYATAAN**

Saya yang bertanda tangan di bawah ini:

Nama : Mohammad A'la Maududy

NIM : 122410101045

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Rancang Bangun Aplikasi Pencarian Lokasi Indekos di Kota Jember Menggunakan *Location Based Service* (LBS) dan Algoritma *Bee Colony Optimization* (BCO)”, adalah benar-benar hasil karya sendiri, kecuali jika dalam pengutipan substansi disebutkan sumbernya, belum pernah diajukan pada institusi mana pun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa adanya tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika di kemudian hari pernyataan ini tidak benar.

Jember, 10 Januari 2020

Yang menyatakan,

Mohammad A'la Maududy

NIM 122410101045

**SKRIPSI**

**RANCANG BANGUN APLIKASI PENCARIAN LOKASI INDEKOS DI  
KOTA JEMBER MENGGUNAKAN *LOCATION BASED SERVICE* (LBS)  
DAN ALGORITMA *BEE COLONY OPTIMIZATION* (BCO)**

Oleh

Mohammad A'la Maududy

NIM 122410101045

Pembimbing :

Dosen Pembimbing Utama : Prof. Dr.Saiful Bukhori, ST., M.Kom

Dosen Pembimbing Pendamping : Oktalia Juwita S.Kom., M.MT

**PENGESAHAN PEMBIMBING**

Skripsi berjudul “Rancang Bangun Aplikasi Pencarian Lokasi Indekos di Kota Jember Menggunakan *Location Based Service* (LBS) dan Algoritma *Bee Colony Optimization* (BCO)”, telah diuji dan disahkan pada:

Hari, tanggal : Jumat, 10 Januari 2020

Tempat : Fakultas Ilmu Komputer Universitas Jember

Disetujui oleh:

Pembimbing I,

Pembimbing II,

Prof. Dr.Saiful Bukhori, ST., M.Kom

NIP 196811131994121001

Oktalia Juwita S.Kom., M.MT

NIP 198110202014042001

**PENGESAHAN PENGUJI**

Skripsi berjudul “Rancang Bangun Aplikasi Pencarian Lokasi Indekos di Kota Jember Menggunakan *Location Based Service* (LBS) dan Algoritma *Bee Colony Optimization* (BCO)”, telah diuji dan disahkan pada:

Hari, tanggal : Jumat, 10 Januari 2020

Tempat : Fakultas Ilmu Komputer Universitas Jember

Tim Penguji,

Penguji I,

Penguji II,

Achmad Maududie, ST., M.Sc

NIP 197004221995121001

Priza Pandunata, S.Kom., M.Sc

NIP 19830131201504001

Mengesahkan

a.n Dekan Fakultas

Dekan Fakultas Ilmu Komputer

Prof. Dr.Saiful Bukhori, ST., M.Kom

NIP 196811131994121001



## RINGKASAN

### **Rancang Bangun Aplikasi Pencarian Lokasi Indekos di Kota Jember Menggunakan *Location Based Service* (LBS) dan Algoritma *Bee Colony Optimization* (BCO)**

Tempat tinggal merupakan salah satu kebutuhan dasar manusia sebagai tempat berteduh dan istirahat. Bagi orang-orang yang jauh dari daerah asalnya, menyewa tempat tinggal adalah solusi. Salah satu tempat tinggal sementara yang paling banyak dicari saat ini yaitu jenis Indekos. Menurut Kamus Besar Bahasa Indonesia Indekos adalah tinggal di rumah orang lain dengan atau tanpa makan.

Kota Jember merupakan salah satu kota pelajar yang mempunyai beberapa perguruan tinggi baik negeri maupun swasta, disamping itu Kota Jember juga menjadi salah satu kota besar di Jawa Timur sehingga kebutuhan akan indekos sangat meningkat dari tahun ke tahun.

Keberadaan indekos yang tersebar luas membutuhkan waktu yang lama untuk mencari yang tepat dan terdekat dengan lokasi kita berada. Selain itu minimnya informasi yang didapat membuat penyewa tidak mendapatkan tempat tinggal dengan fasilitas yang diinginkan. Begitu juga untuk para pemilik indekos juga kesulitan untuk mengiklankan penyewaan tempat tinggal miliknya.

Untuk mengatasi permasalahan itu, maka dibangun aplikasi pencarian indekos dengan menggunakan teknologi *Location-Based Service* (LBS) untuk mencari indekos terdekat dan penerapan Algoritma *Bee Colony Optimization* (BCO) untuk mencari indekos rekomendasi. Nantinya aplikasi ini dapat memudahkan para pemilik indekos untuk mempromosikan indekos miliknya dan juga para pencari indekos untuk mendapatkan indekos sesuai dengan yang mereka inginkan dan tentunya dekat dengan lokasi pilihan mereka.

Penelitian ini menggunakan *Location-Based Service* (LBS) yang meliputi penggunaan fungsi GPS dan pemanfaatan *Leaflet Maps API* untuk menentukan dan menampilkan lokasi pencari indekos atau titik awal dan lokasi indekos sebagai titik akhir, serta penentuan rute perjalanannya. Algoritma *Bee Colony Optimization* (BCO) diimplementasikan pada aplikasi pencari lokasi indekos jember ini, yaitu untuk menentukan *Indekos Rekomendasi*.

Pada algoritma BCO atribut yang dibutuhkan adalah jarak dari user menuju setiap indekos yang ada dan juga bobot tiap indekos. Hasil pembangunan Aplikasi Pencarian Lokasi Indekos di Kota Jember menggunakan *Location Based Service* (LBS) dan Algoritma *Bee Colony Optimization* (BCO) dapat diakses oleh 3 aktor, yaitu admin, pemilik indekos, dan pencari indekos. Untuk fitur utama aplikasi yaitu terletak pada fitur pencarian indekos oleh aktor pencari indekos.

## PRAKATA

Puji syukur kepada Tuhan Yang Maha Esa yang telah melimpahkan rahmat, karunia-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “Rancang Bangun Aplikasi Pencarian Lokasi Indekos di Kota Jember Menggunakan *Location Based Service (LBS)* dan Algoritma *Bee Colony Optimization (BCO)*”. Skripsi ini disusun untuk melengkapi salah satu syarat menyelesaikan pendidikan Strata Satu (S1) pada Program Studi Sisitem Informasi Fakultas Ilmu Komputer Universitas Jember.

Dalam proses penyusunan skripsi ini tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis menyampaikan terima kasih kepada:

1. Prof. Dr. Saiful Bukhori, S.T., M.Kom selaku Dekan Fakultas Ilmu Komputer Universitas Jember;
2. Prof. Dr. Saiful Bukhori, S.T., M.Kom selaku Dosen Pembimbing Utama dan Oktalia Juwita S.Kom., M.M.T. selaku Dosen Pembimbing Anggota yang telah meluangkan waktu, pikiran, dan perhatian dalam penulisan skripsi;
3. Anang Andrianto, S.T., M.T., sebagai dosen pembimbing akademik, yang telah mendampingi penulis sebagai mahasiswa;
4. Seluruh Bapak dan Ibu dosen beserta staf karyawan di Fakultas Ilmu Komputer Universitas Jember;
5. Mohammad Syamsul Hadi dan Ulfiatul Hayati selaku orang tua dari penulis yang telah merawat, mendidik dan membesarkan penulis dengan sepenuh hati;
6. Kakak perempuan La’aliy Af Ida beserta keluarga;
7. Kakak laki-laki Riza ‘Aziziy beserta keluarga;

8. Alm. Dewi Sri Wulandari selaku teman terdekat saya yang selalu menemani, membantu, dan mendukung saya dalam suka dan duka selama masa perkuliahan;
9. Wardhatun Nafisah selaku teman terdekat saya yang selalu menemani, membantu, dan mendukung saya dalam proses pengerjaan skripsi;
10. Teman-teman FORMATION dan semua mahasiswa Fakultas Ilmu Komputer Universitas Jember yang telah menjadi keluarga kecil bagi penulis selama menempuh pendidikan S1;
11. Semua pihak yang tidak dapat disebutkan satu-persatu.

Penulis menyadari penulisan skripsi ini masih jauh dari kesempurnaan, oleh karena itu penulis mengharapkan kritik dan saran dari semua pihak. Akhir kata, penulis berharap semoga skripsi ini dapat memberikan manfaat bagi semua pihak.

Jember, 10 Januari 2020

Penulis

**DAFTAR ISI**

HALAMAN SAMPUL .....	i
PERSEMBAHAN .....	ii
MOTO .....	iii
PERNYATAAN .....	iv
SKRIPSI .....	v
PENGESAHAN PEMBIMBING .....	vi
PENGESAHAN PENGUJI .....	vii
RINGKASAN .....	viii
PRAKATA .....	x
DAFTAR ISI .....	xii
DAFTAR GAMBAR .....	xviii
DAFTAR TABEL .....	xx
BAB 1. PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	3
1.3 Tujuan .....	3
1.4 Batasan Masalah .....	3
1.5 Sistematika Penulisan .....	4
BAB 2. TINJAUAN PUSTAKA .....	5
2.1 Penelitian Terdahulu .....	5
2.2 Indekos .....	6
2.3 <i>Location-Based Service (LBS)</i> .....	7
2.4 <i>Global Positioning System (GPS)</i> .....	8

2.5	<i>Algoritma Bee Colony Optimization</i> .....	9
BAB 3. METODOLOGI PENELITIAN.....		12
3.1	Alur Penelitian.....	12
3.2	Jenis Penelitian .....	13
3.3	Pengumpulan Data .....	13
3.4	Analisis Data .....	14
3.5	Tahapan Penelitian .....	16
3.5.1	Analisis Kebutuhan .....	16
3.5.2	Desain Sistem.....	17
3.5.3	Implementasi .....	18
3.5.4	Pengujian.....	18
3.5.5	Pemeliharaan .....	19
BAB 4. PERANCANGAN DAN PENGEMBANGAN SISTEM.....		20
4.1	<i>Statement Of Purpose</i> .....	20
4.2	Analisis Kebutuhan Sistem .....	20
4.2.1.	Kebutuhan Fungsional.....	21
4.2.2.	Kebutuhan Non-Fungsional .....	21
4.3	Desain Sistem .....	21
4.3.1.	Komponen Sistem .....	22
4.3.2	<i>Usecase Diagram</i> .....	23
4.3.3	<i>Usecase Scenario</i> .....	27
4.3.4	<i>Activity Diagram</i> .....	30
4.3.5	<i>Sequence Diagram</i> .....	32
4.3.6	<i>Class Diagram</i> .....	35
4.3.7	<i>Entity Relationship Diagram (ERD)</i> .....	37



4.4	Penulisan Kode Program .....	37
4.5	Pengujian Sistem .....	40
4.5.1	Pengujian <i>White Box</i> .....	41
4.5.2	Pengujian <i>Black Box</i> .....	42
BAB 5. HASIL DAN PEMBAHASAN.....		44
5.1	Hasil Pembangunan Aplikasi .....	44
5.1.1	Halaman <i>Dashboard</i> .....	44
5.1.2	Halaman Daftar Pemilik Indekos .....	45
5.1.3	Halaman Masuk Pemilik Indekos atau Admin.....	45
5.1.4	Halaman <i>Dashboard</i> Pemilik Indekos .....	45
5.1.5	Halaman Manajemen Profil Pemilik Indekos .....	46
5.1.6	Halaman Daftar Pencari Indekos.....	47
5.1.7	Halaman Masuk Pencari Indekos .....	48
5.1.8	Halaman <i>Dashboard</i> Pencari Indekos .....	48
5.1.9	Halaman Fitur Pencarian Indekos .....	49
5.1.10	Halaman Hasil Pencarian Indekos.....	49
5.1.11	Halaman Penunjuk Arah Indekos.....	51
5.1.12	Halaman Manajemen Profil Pencari Indekos.....	51
5.1.13	Halaman <i>Dashboard</i> Admin .....	52
5.1.14	Halaman Manajemen Pemilik Indekos .....	52
5.1.15	Halaman Informasi Indekos Admin .....	53
5.1.16	Halaman Manajemen Pencari Indekos .....	54
5.1.17	Halaman Manajemen Profil Admin .....	55
5.2	Pembahasan .....	55
5.2.1.	Implementasi Algoritma <i>Bee Colony Optimization</i> (BCO) .....	56

5.2.2. Pembahasan Hasil Pengujian Sistem.....	70
BAB 6. PENUTUP .....	71
6.1 Kesimpulan.....	71
6.2 Saran.....	71
DAFTAR PUSTAKA .....	73
LAMPIRAN.....	74
A. <i>Use Case Scenario</i> .....	74
A.1 Skenario <i>Usecase</i> Daftar <i>Owner</i> .....	74
A.2 Skenario <i>Usecase</i> Daftar .....	75
A.3 Skenario <i>Usecase</i> Masuk <i>Owner</i> .....	75
A.4 Skenario <i>Usecase</i> Masuk .....	77
A.5 Skenario <i>Usecase</i> Manajemen Data Indekos .....	78
A.6 Skenario <i>Usecase</i> Cari Indekos.....	79
A.7 Skenario <i>Usecase</i> Lihat Indekos .....	80
A.8 Skenario <i>Usecase</i> Manajemen Profil <i>Owner</i> .....	81
A.9 Skenario <i>Usecase</i> Manajemen Profil .....	82
A.10 Skenario <i>Usecase</i> Manajemen Pemilik Indekos .....	83
A.11 Skenario <i>Usecase</i> Manajemen Pencari Indekos.....	84
A.12 Skenario <i>Usecase</i> Keluar <i>Owner</i> .....	85
A.13 Skenario <i>Usecase</i> Keluar .....	85
B. <i>Activity Diagram</i> .....	87
B.1 <i>Activity Diagram</i> Daftar <i>Owner</i> .....	87
B.2 <i>Activity Diagram</i> Daftar .....	88
B.3 <i>Activity Diagram</i> Masuk <i>Owner</i> .....	89
B.4 <i>Activity Diagram</i> Masuk .....	89



B.5	<i>Activity Diagram</i> Manejemen Data Indekos .....	90
B.6	<i>Activity Diagram</i> Cari Indekos.....	91
B.7	<i>Activity Diagram</i> Lihat Indekos .....	91
B.8	<i>Activity Diagram</i> Manajemen Profil <i>Owner</i> .....	92
B.9	<i>Activity Diagram</i> Manajemen Profil .....	92
B.10	<i>Activity Diagram</i> Manajemen Pemilik Indekos .....	93
B.11	<i>Activity Diagram</i> Manajemen Pencari Indekos.....	94
B.12	<i>Activity Diagram</i> Keluar <i>Owner</i> .....	95
B.13	<i>Activity Diagram</i> Keluar .....	96
C.	<i>Sequence Diagram</i> .....	97
C.1	<i>Sequence Diagram</i> Daftar <i>Owner</i> .....	97
C.2	<i>Sequence Diagram</i> Daftar .....	98
C.3	<i>Sequence Diagram</i> Masuk <i>Owner</i> .....	99
C.4	<i>Sequence Diagram</i> Masuk .....	100
C.5	<i>Sequence Diagram</i> Manejemen Data Indekos .....	101
C.6	<i>Sequence Diagram</i> Cari Indekos.....	101
C.7	<i>Sequence Diagram</i> Lihat Indekos .....	102
C.8	<i>Sequence Diagram</i> Manajemen Profil <i>Owner</i> .....	103
C.9	<i>Sequence Diagram</i> Manajemen Profil .....	104
C.10	<i>Sequence Diagram</i> Manajemen Pemilik Indekos .....	105
C.11	<i>Sequence Diagram</i> Manajemen Pencari Indekos.....	106
C.12	<i>Sequence Diagram</i> Keluar <i>Owner</i> .....	107
C.13	<i>Sequence Diagram</i> Keluar .....	108
D.	Pengujian Black Box .....	109
D.1	User Admin .....	109

D.2	User Pemilik Indekos .....	111
D.3	User Pencari Indekos.....	111



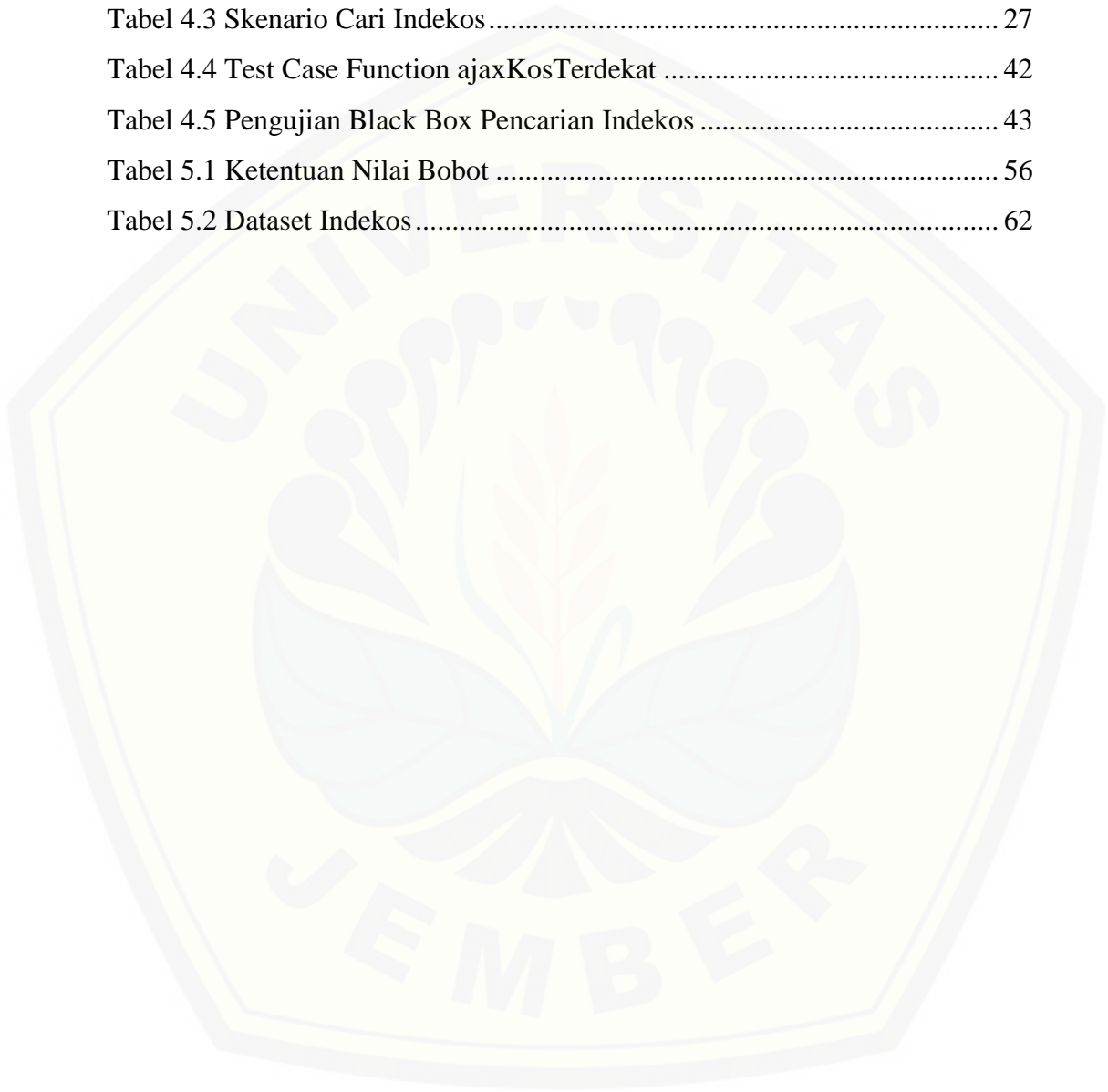
**DAFTAR GAMBAR**

Gambar 3.1 Alur Penelitian.....	12
Gambar 3.2 Flowchart penerapan algoritma BCO pada aplikasi.....	15
Gambar 3.3 Diagram SDLC Waterfall (Pressman, 1997).....	16
Gambar 4.1 Komponen Sistem aplikasi INDEKOS .....	22
Gambar 4.2 Use Case Diagram sistem pencari indekos jember .....	24
Gambar 4.3 Activity Diagram Cari Indekos .....	31
Gambar 4.4 Sequence Diagram Cari Indekos .....	34
Gambar 4.5 Class Diagram Aplikasi INDEKOS .....	36
Gambar 4.6 Entity Relationship Diagram.....	37
Gambar 4.7 Kode Program function probabilitas .....	37
Gambar 4.8 Kode Program function distance .....	38
Gambar 4.9 Kode Program function ajaxKosTerdekat.....	40
Gambar 4.10 Kode Program function getBobotAttribute .....	40
Gambar 4.11 Diagram Alir Function ajaxKosTerdekat.....	41
Gambar 5.1 Halaman Dashboard .....	44
Gambar 5.2 Halaman Daftar Pemilik Indekos .....	45
Gambar 5.3 Halaman Masuk Pemilik Indekos atau Admin.....	45
Gambar 5.4 Halaman Dashboard Pemilik Indekos .....	46
Gambar 5.5 Halaman Manajemen Profil Pemilik Indekos .....	47
Gambar 5.6 Halaman Daftar Pencari Indekos.....	47
Gambar 5.7 Halaman Masuk Pencari Indekos .....	48
Gambar 5.8 Menu Halaman Dashboard Pencari Indekos .....	48
Gambar 5.9 Halaman Fitur Pencarian Indekos .....	49
Gambar 5.10 Halaman Hasil Pencarian Indekos.....	50
Gambar 5.11 Halaman Penunjuk Arah Indekos.....	51
Gambar 5.12 Halaman Manajemen Profil Pencari Indekos.....	52
Gambar 5.13 Halaman Dashboard Admin .....	52
Gambar 5.14 Halaman Manajemen Pemilik Indekos .....	53
Gambar 5.15 Halaman Informasi Indekos Admin .....	54

Gambar 5.16 Halaman Manajemen Pencari Indeks.....	55
Gambar 5.17 Halaman Manajemen Profil Admin .....	55
Gambar 5.18 Rute menuju KOS BRANTAS 7.....	57
Gambar 5.19 Rute menuju KOS SAKINAH PUTRI MUSLIMAH.....	58
Gambar 5.20 Rute menuju KOS RIAU 21 .....	58
Gambar 5.21 Rute menuju KOS JAWA 6 .....	59
Gambar 5.22 Rute menuju CENTRAL KOS .....	59
Gambar 5.23 Rute menuju WIWASHA.....	60
Gambar 5.24 Rute menuju KOS SANJAYA .....	60
Gambar 5.25 Rute menuju KOS BANGKA 4 .....	61
Gambar 5.26 Rute menuju KOS BU NARDI.....	61
Gambar 5.27 Rute menuju KOS CATERINA SYARIAH .....	62
Gambar 5.28 Dataset dalam perhitungan aplikasi.....	69
Gambar 5.29 Hasil perhitungan Algoritma BCO dengan Aplikasi.....	69

**DAFTAR TABEL**

Tabel 4.1 Definsi Aktor.....	25
Tabel 4.2 Definisi Usecase.....	25
Tabel 4.3 Skenario Cari Indekos.....	27
Tabel 4.4 Test Case Function ajaxKosTerdekat .....	42
Tabel 4.5 Pengujian Black Box Pencarian Indekos .....	43
Tabel 5.1 Ketentuan Nilai Bobot .....	56
Tabel 5.2 Dataset Indekos.....	62



## BAB 1. PENDAHULUAN

Bab ini merupakan langkah awal dari penulisan tugas akhir ini. Bab ini berisi latar belakang, rumusan masalah, tujuan dan manfaat, batasan masalah, metodologi penelitian, dan sistematika penulisan.

### 1.1 Latar Belakang

Tempat tinggal merupakan salah satu kebutuhan dasar manusia sebagai tempat berteduh dan istirahat. Bagi orang-orang yang jauh dari daerah asalnya, menyewa tempat tinggal adalah salah satu cara mendapatkan tempat beristirahat. Namun, minimnya informasi yang didapat membuat penyewa tidak mendapatkan tempat tinggal dengan fasilitas yang diinginkan. Salah satu tempat tinggal sementara yang paling banyak dicari saat ini yaitu jenis Indekos. Menurut Kamus Besar Bahasa Indonesia Indekos adalah tinggal di rumah orang lain dengan atau tanpa makan.

Kota Jember merupakan salah satu kota pelajar yang mempunyai beberapa perguruan tinggi baik negeri maupun swasta, disamping itu Kota Jember juga menjadi salah satu kota besar di Jawa Timur sehingga kebutuhan akan indekos sangat meningkat dari tahun ke tahun. Indekos yang tersebar terdiri dari indekos harian hingga indekos bulanan dan kotrak satu tahunan. Keberadaan indekos yang tersebar luas itu membutuhkan waktu yang lama untuk mencari yang tepat dan terdekat dengan lokasi kita berada.

Indekos dengan fasilitas sesuai kebutuhan dan terdekat dengan tempat beraktivitas akan lebih menguntungkan untuk para konsumen. Konsumen yang rata-rata berusia dua puluh tahunan lebih tertarik dengan indekos yang mempunyai tempat strategis serta dekat dengan kampus atau kantor dengan berbagai kemudahan fasilitas yang ditawarkan. Konsumen juga menginginkan agar privasinya terjaga serta nyaman dan aman dari gangguan kejahatan. Harga yang ditawarkan juga menjadi salah satu daya saing untuk memikat hati para calon penghuni indekos. Oleh karena itu, baik konsumen ataupun pebisnis indekos



membutuhkan wadah untuk saling bertukar informasi mengenai kebutuhan masing-masing pihak.

Informasi merupakan kebutuhan utama bagi sebagian besar manusia. Sebagian besar orang telah bergantung pada internet untuk memperoleh informasi. Seiring dengan teknologi yang terus berkembang seakan tidak ada titik akhir, hal itu menjadi sebuah peluang bagi para pengembang Information Technology (IT) untuk menciptakan sebuah sistem berbasis website guna bertukar informasi. Sistem berbasis website yaitu sistem yang diakses melalui web browser dengan menggunakan jaringan sebagai media transmisi. Sistem website juga merupakan sebuah perangkat lunak atau software yang dikodekan dengan bahasa pemrograman seperti *html*, *Javascript*, *php*, dan bahasa pemrograman lainnya. Penggunaan internet membuat informasi bisa didapatkan dimanapun kita berada dalam waktu singkat. Website tersebut memungkinkan pengguna untuk menghubungkan informasi tentang lokasi indekos yang terdekat dengan posisi kita saat itu, lokasi yang terdekat bisa diperoleh dari titik koordinat dimana kita berada kemudian sistem akan mencari kos yang terdekat dengan titik tersebut, sehingga meminimalisir waktu dalam mencari indekos yang diinginkan.

Masalah setiap calon pengguna yang ingin melakukan survei tempat indekos adalah minimnya informasi tentang rute perjalanan. Pencari indekos biasanya membutuhkan waktu yang lama untuk menuju ke lokasi karena tidak semua indekos memberikan informasi yang jelas mengenai tempat indekos. Informasi rute perjalanan tersebut dapat mempermudah pengguna memilih indekos yang diinginkan. Oleh karena itu, dibutuhkan suatu sistem untuk mengoptimasi pencarian rute jalur perjalanan dari posisi awal menuju posisi tujuan pada suatu peta lokasi. Hal ini dapat dimanfaatkan oleh para pencari indekos dalam memilih jalur yang akan ditempuh agar perjalanan menjadi lebih nyaman dan efisien dari segi jarak tempuh dan biaya.

*Location Based Service* (LBS) adalah layanan atau service yang dapat menentukan letak atau lokasi kita sesuai dengan perubahan entitas posisi kita. Di dalam penelitian ini, LBS digunakan untuk menentukan lokasi pencari dan lokasi indekos. Penerapan LBS direpresentasikan dengan menggunakan *Leaflet Maps*.

*Leaflet Maps* tersebut bisa menangkap dan menampilkan lokasi pencari dan indekos serta rute perjalanannya. Dalam penelitian ini, peneliti menggunakan Algoritma *Bee Colony Optimization* (BCO) untuk menentukan indekos rekomendasi. BCO menggunakan bobot dari masing masing indekos dan juga jarak dari lokasi pencari indekos menuju ke masing masing lokasi indekos sebagai parameter penentu indekos rekomendasi.

## 1.2 Rumusan Masalah

Berdasarkan uraian yang telah disampaikan dalam latar belakang mendefinisikan permasalahan yang harus diselesaikan dalam penulisan ini, yaitu:

1. Bagaimana menerapkan metode *Location-Based Service* (LBS) dan Algoritma *Bee Colony Optimization* (BCO) untuk mencari lokasi indekos?
2. Bagaimana membangun aplikasi pencarian indekos dengan menggunakan metode *Location-Based Service* (LBS) untuk mencari indekos terdekat dan penerapan Algoritma *Bee Colony Optimization* (BCO) untuk mencari indekos rekomendasi?

## 1.3 Tujuan

Tujuan dalam penulisan ini merupakan jawaban dari perumusan masalah yang telah disebutkan. Tujuan yang ingin dicapai dalam penelitian ini adalah:

1. Menerapkan metode *Location-Based Service* (LBS) dan Algoritma *Bee Colony Optimization* (BCO) untuk mencari lokasi indekos.
2. Membangun aplikasi pencarian indekos dengan menggunakan metode *Location-Based Service* (LBS) untuk mencari indekos terdekat dan penerapan Algoritma *Bee Colony Optimization* (BCO) untuk mencari indekos rekomendasi.

## 1.4 Batasan Masalah

Penulis memberikan batasan masalah untuk objek dan tema yang dibahas sehingga tidak terjadi penyimpangan dalam proses penulisan dan pembuatan aplikasi. Berikut adalah batasan masalah yang dicantumkan:



1. Aplikasi yang dibangun yaitu aplikasi pencarian indekos dengan menggunakan metode *Location-Based Service* (LBS) dan Algoritma *Bee Colony Optimization* (BCO) berbasis *website*.
2. Data indekos yang digunakan dalam penelitian ini merupakan indekos yang berada dekat dengan daerah Universitas Jember (radius 5 KM)

## 1.5 Sistematika Penulisan

Sistematika penulisan dan keruntutan skripsi ini disusun sebagai berikut:

1. **Pendahuluan**  
Bab ini menjelaskan tentang latar belakang, perumusan masalah, tujuan, dan manfaat, ruang lingkup studi dan sistematika penulisan.
2. **Tinjauan Pustaka**  
Bab ini menjelaskan tentang materi, informasi, tinjauan pustaka, dan studi terdahulu yang menjadi kerangka pemikiran dalam penelitian.
3. **Metodologi Penelitian**  
Bab ini menjelaskan tentang metode penelitian yang digunakan dalam penelitian.
4. **Pengembangan Sistem**  
Bab ini menjelaskan tentang pengembangan sistem yang dikembangkan.
5. **Hasil dan Pembahasan**  
Bab ini menjelaskan tentang hasil dan pembahasan dari penelitian yang dilakukan.
6. **Penutup**  
Bab ini berisi tentang kesimpulan dari penelitian dan saran untuk penelitian selanjutnya.

## BAB 2. TINJAUAN PUSTAKA

Bab ini memaparkan tinjauan yang berkaitan dengan masalah yang dibahas, kajian teori yang berkaitan dengan masalah, kerangka pemikiran yang merupakan sintesis dari kajian teori yang dikaitkan dengan permasalahan yang dihadapi. Teori-teori ini diambil dari buku, literature, jurnal, dan internet.

### 2.1 Penelitian Terdahulu

Penelitian terdahulu yang dilakukan oleh Ayu Septyaningsih (2015) dengan judul “Pencarian Lokasi Bengkel Mobil Di Kota Jember Menggunakan Location Based Service (LBS) dan Algoritma Bee Colony Optimization Berbasis Android”, dalam penelitian ini, peneliti menggunakan bantuan GPS dari *Mobile Device* dan *Google Maps API*. Algoritma Bee digunakan oleh peneliti untuk mendapatkan jalur terpendek dengan melakukan penelusuran total jarak keseluruhan yang dibutuhkan dalam menempuh rute perjalanan dari posisi asal menuju semua objek yang diinginkan. Sistem pencarian lokasi bengkel mobil di Jember ini berbasis client server dimana aplikasi Bengkel Mobil Jember sebagai client yang menjalankan data dari web server. Hasil dari penelitian ini yaitu Aplikasi Bengkel Mobil Jember yang diterapkan pada android versi 4.1 keatas.

Penelitian kedua dilakukan oleh Akbar, Toufan dan Kurniawan (2013) yang berjudul “Aplikasi Wisata Kota Bandung Menggunakan Metode Location-Based Services (LBS) pada Android”. Menggunakan metode LBS sebagai penyedia layanan personalisasi kepada pengguna perangkat bergerak (*mobile device*) yang disesuaikan dengan lokasi mereka saat ini selanjutnya digunakan untuk menentukan lokasi wisata di kota Bandung. Dalam pengambilan koordinat, sistem memanfaatkan GPS. Kemudian untuk mendapatkan peta dan rute, sistem menggunakan Google Map APIs dengan memberikan parameter koordinat bumi. Setelah mengirim parameter tersebut ke GoogleMap, maka GoogleMap server akan membalas berupa peta statik (gambar). Dalam pencarian rute, sistem mengirimkan dua koordinat bumi sebagai alamat awal dan alamat tujuan,

kemudian GoogleMapDirection server akan membalas berupa data rute dalam bentuk JSON yang selanjutnya sistem akan memparsing data tersebut dan ditampilkan kepada pengguna.

Penelitian ketiga dilakukan oleh Muhammad, Nurul, dan Budi (2017) yang berjudul “Sistem Optimasi Rute Tempat Wisata Kuliner Di Malang Menggunakan Algoritma Bee Colony”. Pencarian lokasi wisata kuliner dipilih dalam pencarian rute optimum dari posisi awal menuju posisi tujuan. Dipilihnya algoritma bee colony dikarenakan algoritma ini dirasa memiliki kemampuan untuk keluar dari local minimum dan dapat secara efisien digunakan untuk optimasi. Algoritma bee colony dirasa mampu menyelesaikan permasalahan Traveling Salesman Problem lebih baik dibandingkan dengan algoritma lain yang juga didasarkan pada kecerdasan berkelompok. Pada pengujian, didapatkan hasil bahwa penggunaan algoritma bee colony telah mengalami konvergensi dalam pencarian solusi terbaiknya yang dapat dilihat dari fitness yang dihasilkan. Salah satu yang terbaik telah mengalami konvergensi pada jumlah bee colony sebanyak 20 dari 50 jumlah bee colony. Selain itu konvergensi juga dapat dilihat pada jumlah iterasi 20 dari jumlah maksimum iterasi 50.

## 2.2 Indekos

Indekos dirancang untuk memenuhi kebutuhan hunian yang bersifat sementara dengan sasaran orang rantau yang belum memiliki tempat tinggal di dalam daerah dan berasal dari luar kota. Indekos biasanya ditempati oleh orang yang menginginkan tempat tinggal yang berdekatan dengan lokasi mobilitas. Oleh karena itu, fungsi dari indekos dapat dijabarkan sebagai berikut:

- 1) Sebagai sarana tempat tinggal sementara mahasiswa atau pelajar selama masa studi.
- 2) Sebagai sarana tempat tinggal sementara para pekerja yang tidak memiliki rumah tinggal di area mobilitasnya.
- 3) Sebagai sarana untuk mahasiswa menjadi lebih mandiri, disiplin, dan bertanggung jawab.

- 4) Sebagai sarana sosial antar penghuni indekos dan juga lingkungan sekitarnya.

### 2.3 *Location-Based Service (LBS)*

*Location-Based Service (LBS)* memberikan layanan personalisasi kepada pengguna perangkat bergerak (mobile device) yang disesuaikan dengan lokasi mereka saat ini. LBS membuka pasar baru bagi pengembang, operator jaringan selular, dan penyedia layanan untuk mengembangkan dan memberikan nilai tambah layanan. Memberikan informasi kondisi lalu lintas saat ini, menambahkan informasi rute perjalanan, membantu menemukan lokasi wisata terdekat, dan banyak lagi. Dua unsur utama LBS adalah :

1. *Location Manager (API Maps)*

Menyediakan tool atau *source* untuk LBS, *Application Program Interface (API)*. Maps menyediakan fasilitas untuk menampilkan, memanipulasi maps atau peta beserta feature-feature lainnya seperti tampilan satelit, street (jalan), maupun gabungannya. Paket ini berada `com.google.android.map`.

2. *Location Provider (API Location)*

Menyediakan teknologi pencarian lokasi yang digunakan oleh device atau perangkat. API Location berhubungan dengan data GPS (*Global Positioning System*) dan data lokasi real-time. API Location berada pada paket android yaitu dalam paket `android.location`. Dengan Location Manager, kita dapat menentukan lokasi kita saat ini, Track gerakan atau perpindahan, serta kedekatan dengan lokasi tertentu dengan mendeteksi perpindahan.

*Location Based Service* memiliki komponen untuk melakukan layanan berbasis lokasi. Dalam menggunakan layanan berbasis lokasi elemen yang diperlukan antara lain:

- a. *Mobile Devices* yaitu sebuah alat yang digunakan untuk meminta informasi yang dibutuhkan. Biasanya perangkat yang memungkinkan yaitu PDA,

*Mobile Phones*, Laptop dan perangkat lainnya yang mempunyai fasilitas navigasi.

- b. *Communication Network* adalah jaringan selular yang mengirimkan data pengguna dan permintaan layanan.
- c. *Positioning Component* untuk pengolahan layanan biasanya posisi pengguna harus ditentukan dengan menggunakan GPS.
- d. *Service and Application Provider* adalah penyedia layanan pengguna selular yang bertanggung jawab untuk memproses layanan.
- e. *Data and Content Provider* yaitu penyedia layanan informasi data yang dapat diminta oleh pengguna.

#### **2.4 Global Positioning System (GPS)**

*Global Positioning System* (GPS) merupakan sebuah alat yang dirancang diantaranya untuk mengetahui posisi lintang dan bujur suatu daerah dengan bantuan satelit. Selain itu, GPS juga dapat berfungsi untuk menentukan ketinggian, kompas, posisi matahari dan bulan terbenam, peta, navigator dan masih banyak lagi. GPS (*Global Positioning System*) adalah sistem navigasi yang berbasis satelit yang saling berhubungan yang berada di orbitnya. GPS adalah satu-satunya sistem satelit navigasi global untuk penentuan lokasi, kecepatan, arah, dan waktu

Dengan teknologi GPS dapat digunakan untuk beberapa keperluan sesuai dengan tujuannya. GPS dapat digunakan oleh peneliti, olahragawan, petani, tentara, pilot, petualang, pendaki, pengantar barang, pelaut, kurir, penebang pohon, pemadam kebakaran dan orang dengan berbagai kepentingan untuk meningkatkan produktivitas, keamanan, dan untuk kemudahan. Dari beberapa pemakai di atas dikategorikan menjadi:

1. Lokasi digunakan untuk menentukan dimana lokasi suatu titik dipermukaan bumi berada.
2. Navigasi membantu mencari lokasi suatu titik di bumi
3. *Tracking* membantu untuk memonitoring pergerakan obyek dan membantu memetakan posisi tertentu, dan perhitungan jaringan terdekat



4. *Timing* dapat dijadikan dasar penentuan jam seluruh dunia, karena memakai jam atom yang jauh lebih presisi di banding dengan jam biasa.

Tidak peduli posisi Anda, di tengah laut, di tengah hutan, di atas gunung, ataupun di pusat kota. Selama GPS dapat menerima sinyal dari satelit secara langsung tanpa halangan, maka GPS akan selalu memberikan informasi koordinat posisi Anda. GPS membutuhkan area pandang yang bebas langsung ke langit. Halangan-halangan seperti pohon, gedung, bahkan kaca film kelas V-Kool, bisa mengurangi akurasi sinyal yang diterima oleh GPS. Bahkan bukan tidak mungkin GPS tidak bisa menerima sinyal sama sekali dari satelit. GPS juga memiliki *feature* tambahan yang mampu memberikan informasi selama anda di perjalanan, seperti kecepatan, lama perjalanan, jarak yang telah ditempuh, waktu, dan masih banyak.

Sebuah GPS juga memiliki *firmware* yang bisa di-upgrade. Upgrade *firmware* ini bisaanya disediakan pada site produsen GPS tersebut. Upgrade *firmware* bisaanya menggunakan kabel yang dibundel atau-pun tersedia sebagai aksesoris. Kabel ini juga ternyata bisa digunakan untuk menghubungkan GPS ke komputer (baik itu notebook, PC, maupun PDA dengan sedikit bantuan konverter). *Software* GPS yang tersedia untuk berbagai *platform* tersebut juga cukup banyak. Dengan *software* tersebut, Anda dapat dengan mudah mendownload informasi dari GPS. Memori sebuah GPS memang relatif terbatas, sehingga kemampuan ekstra untuk menyimpan informasi yang pernah Anda tempuh ke PC/PDA (yang bisaanya memiliki memori lebih besar) tentu akan sangat menyenangkan. Untuk media komunikasi GPS dengan *hardware* lain.

## 2.5 Algoritma Bee Colony Optimization

Lebah merupakan serangga sosial yang sangat terorganisir. Koloni lebah buatan bersama-sama mencari solusi optimal dari masalah yang diberikan. Setiap lebah buatan menghasilkan satu solusi untuk masalah ini. Ada dua fase dalam satu langkah algoritma BCO yaitu fase maju (forward pass) dan fase mundur (backward pass).

Lebah menggunakan aturan transisi dalam membuat keputusan untuk memilih titik yang dikunjungi berikutnya. Probabilitas transisi ( $P_{ij,n}$ ) mengukur kemungkinan perpindahan dari titik  $i$  ke titik  $j$  pada transisi  $n$ . Probabilitas transisi fungsi jarak dari 2 titik dan *arc fitness* pada jalur yang dilalui. Fungsi ini diformulasikan pada persamaan 1:

$$P_{ij,n} = \frac{[\rho_{ij,n}]^\alpha \cdot \left[\frac{1}{d_{ij}}\right]^\beta}{\sum_{j \in A_{i,n}} \left([\rho_{ij,n}]^\alpha \cdot \left[\frac{1}{d_{ij}}\right]^\beta\right)} \dots\dots\dots(1)$$

dimana,

$P_{ij,n}$  = Probabilitas transisi.

$i$  = Posisi asal ( titik ke  $i$ ).

$j$  = Posisi tujuan (titik ke  $j$ ) yang bisa ditempuh dari posisi asal.

$n$  = Transisi.

$\rho$  = Arc fitness suatu jalur.

$d$  = Jarak.

$\alpha$  = Variabel biner yang menonaktifkan pengaruh arc fitness dalam model.

$\beta$  = Parameter yang mengontrol tingkat signifikan jarak.

Pembobotan nilai arc fitness ( $\rho$ ) dilakukan menggunakan Persamaan 2 :

$$\rho_{ij,n} = \begin{cases} \lambda & , j \in F_{i,n}, |A_{i,n}| > 1 \\ \frac{1-\lambda|A_{i,n} \cap F_{i,n}|}{|A_{i,n} - F_{i,n}|} & , j \notin F_{i,n}, |A_{i,n}| > 1 \\ 1 & , |A_{i,n}| = 1 \end{cases} \dots\dots\dots(2)$$

dimana,

$A_{i,n}$  = Suatu set titik yang bertetangga dengan posisi asal  $i$  pada transisi  $n$ .

$F_{i,n}$  = Satu titik yang merupakan bagian dari  $A_{i,n}$  yang dipilih oleh lebah pada transisi  $n$ .

$\lambda$  = Nilai arc fitness ( $\rho$ ) untuk jalur yang dipilih oleh lebah.

Sekembalinya lebah ke sarang setelah membangun tur lengkap, waggle dance akan dilakukan untuk diperlihatkan bagi lebah lainya yang ada disarang. Kebijakan yang diterapkan dalam memungkinkan waggle dance adalah lebah yang berhasil menemukan sumber makanan yang diperbolehkan untuk menari. Tarian seekor lebah selain memberikan informasi jalan yang lebih pendek juga mengandung informasi durasi waktu. Proses perhitungan waggle dance menggunakan persamaan 3 namun dimulai dengan melakukan perhitungan profitabilitas tiap lebah menggunakan persamaan 4, kemudian menghitung profitabilitas koloni lebah dengan persamaan 5.

$$D_i = K \cdot \frac{P f_i}{P f_{colony}} \dots\dots\dots(3)$$

$$P f_i = \frac{1}{L_i} \dots\dots\dots(4)$$

$$P f_{colony} = \frac{1}{N_{Bee}} \sum_{i=1}^{N_{Bee}} P f_i \dots\dots\dots(5)$$

dimana,

$D_i$  = durasi waggle dance lebah ke i

K = skala faktor waggle dance

$P f_i$  = Profitabilitas lebah ke i

$L_i$  = Panjang jarak yang ditempuh lebah ke i

$P f_{colony}$  = Profitabilitas koloni lebah

$N_{Bee}$  = Banyaknya lebah yang dilepas dan berhasil menemukan tujuan

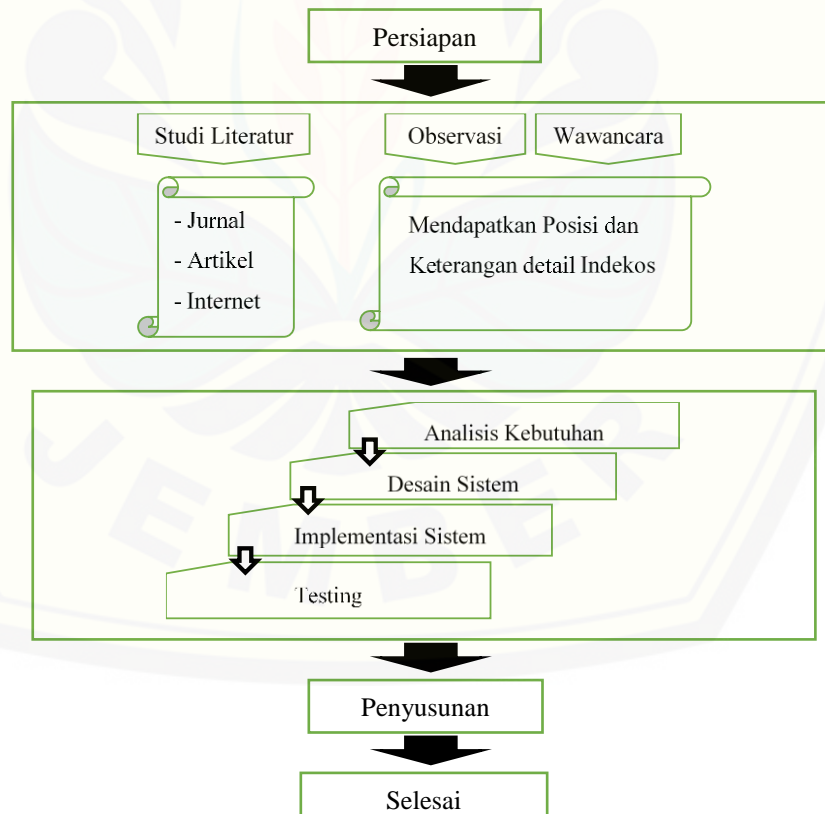


### BAB 3. METODOLOGI PENELITIAN

Bab ini menjelaskan tentang gambaran tahapan yang sistematis yang dilakukan untuk menganalisa data untuk menjawab perumusan masalah sehingga dapat mencapai tujuan sebenarnya dari penelitian. Metodologi penelitian akan menjelaskan tentang jenis penelitian, tempat dan waktu penelitian serta tahapan dari penelitian.

#### 3.1 Alur Penelitian

Alur penelitian merupakan urutan langkah penelitian yang dilakukan mulai dari studi literatur, pengumpulan data, pengolahan data, perancangan sistem, dan pengimplementasian rancangan sistem. diagram alur pada penelitian ini dapat dilihat pada Gambar 3.1 berikut.



Gambar 3.1 Alur Penelitian

### 3.2 Jenis Penelitian

Penelitian ini menggunakan jenis penelitian kualitatif dan kuantitatif. Menggunakan penelitian jenis kualitatif karena pada penelitian ini menganalisa studi literatur yang berhubungan dengan indikator untuk menentukan indeks dengan bobot yang perlu digunakan dalam aplikasi. Menggunakan penelitian jenis kuantitatif karena data yang diolah dalam bentuk angka.

### 3.3 Pengumpulan Data

Pengumpulan data merupakan langkah awal untuk menentukan data apa saja yang dibutuhkan dan bagaimana mendapatkan maupun mengumpulkannya. Pengumpulan data yang dilakukan pada penelitian ini dengan cara mengumpulkan data dari beberapa sumber dokumen, observasi lokasi indeks dan penangkapan titik koordinat lokasi dengan metode Location Based Service. Penangkapan titik koordinat dilakukan dengan mengunjungi sample indeks yang akan digunakan dalam penelitian ini, kemudian membuka aplikasi maps untuk menangkap posisi Longitude dan Latitude. Pada tahap pengumpulan data awal ini, yang digunakan adalah aplikasi Google Maps. Pada tahap ini data, fakta, dan informasi dicari dan diidentifikasi.

Pengamatan merupakan hal terpenting karena merupakan proses penyesuaian data yang didapat dengan hal yang sebenarnya ada dilapangan. Pengamatan dilakukan langsung pada objek dan melakukan pencatatan lokasi yang ditemukan dan penilaian tentang kriteria indeks. Pengamatan indeks kemudian diberi penilaian pada setiap objek.

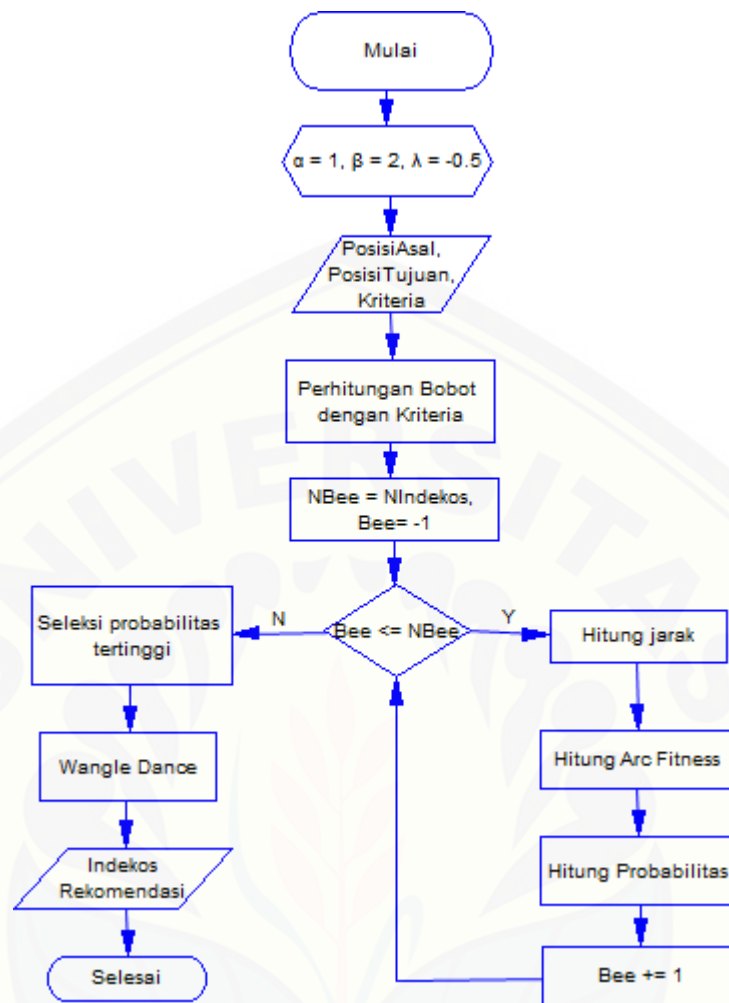
Jenis pengumpulan data yang digunakan dalam pembuatan aplikasi pencari lokasi indeks yaitu menggunakan teknik observasi dan instrumen penelitian yang digunakan meliputi studi literatur. Pengumpulan data dengan observasi langsung atau dengan pengamatan langsung adalah cara pengambilan data dengan menggunakan mata tanpa ada pertolongan alat standar lain untuk keperluan tersebut. Observasi dilakukan pada lokasi indeks di sekitar Universitas Jember (radius 5 KM) dan mengetahui bagaimana informasi indeks dengan lengkap sehingga dapat menentukan bobot dari indeks. Informasi yang dapat digunakan

meliputi; a) harga, b) kamar kosong, c) jenis indekos (putra/putri), d) perabot kamar. e) dapur, f) penjaga. g) wifi, h) kamar mandi dalam, dan i) jam malam. Bobot indekos tersebut digunakan sebagai parameter perhitungan dalam algoritma Bee Colony Optimization. Dalam penelitian ini. Bobot dihitung dari rata-rata hasil kuesioner. Bobot tersebut digunakan sebagai parameter dalam menghitung BCO. Jarak yang didapat berdasarkan koordinat yang dicatat juga menjadi parameter untuk perhitungan optimasi indekos. Parameter yang digunakan digunakan adalah jarak dan bobot yang telah dihitung.

### 3.4 Analisis Data

Tahap analisis data dimulai dengan menganalisis data yang telah dikumpulkan. Data yang telah terkumpul akan dimasukan sebagai parameter dalam perhitungan algoritma Bee Colony Optimization. Perhitungan didapat dari pengaruh bobot yang telah ditentukan dengan jarak pencari dan indekos. Langkah selanjutnya memasukkan data yang telah didapat kedalam perhitungan algoritma *Bee Colony Optimization*.

Fitur utama dalam aplikasi INDEKOS yaitu Pencarian Indekos, pada fitur ini hasilnya yaitu data-data indekos yang posisi nya terdekat dengan posisi user saat itu atau posisi yang telah dipilih oleh user, dan juga terdapat salah satu indekos yang bertanda “Rekomendasi” yang didapat dari perhitungan Bee Colony Optimization berdasarkan parameter jarak dan bobot tiap indekos. Berdasarkan sistem yang nantinya akan dibuat maka Flowchart penerapan algoritma *Bee Colony Optimization* pada aplikasi dapat dilihat pada Gambar 3.2

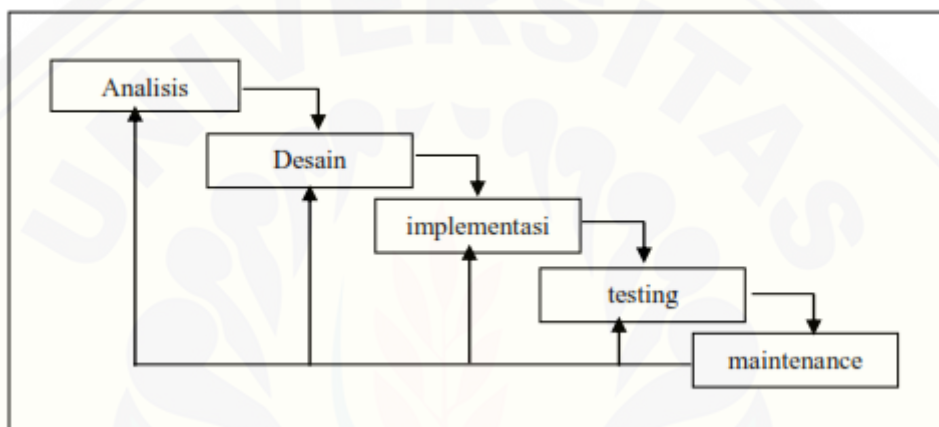


Gambar 3.2 Flowchart penerapan algoritma Bee Colony Optimization pada aplikasi

Pada flowchart diatas dapat dijelaskan bahwa penerapan *Bee Colony Optimization* dimulai dari deklarasi nilai  $\alpha$ ,  $\beta$ , dan  $\lambda$ . Kemudian input posisiAsal yang didapat dari penentuan titik oleh pencari indekos, posisi Tujuan dan Kriteria didapat dari database yang sebelumnya telah ditentukan oleh masing-masing pemilik indekos. Selanjutnya dilakukan perhitungan Bobot berdasarkan Kriteria. Lalu penetapan nilai NBee dan Bee yang kemudian digunakan dalam proses perulangan untuk masing-masing indekos yang didalamnya terdapat proses perhitungan jarak, arc fitness, dan probabilitas. Setelah selesai menghitung sesuai jumlah NIndekos kemudian dilakukan seleksi probabilitas yang memiliki nilai tertinggi. Lalu dilakukan penetapan indekos rekomendasi (wangle dance) yang kemudian ditampilkan sebagai hasil output.

### 3.5 Tahapan Penelitian

Penelitian ini akan dilakukan dalam beberapa tahap yang disesuaikan dengan metode *Software Development Life Cycle (SDLC) waterfall* yang dibagi menjadi beberapa tahapan, yaitu analisis kebutuhan, desain sistem, implementasi, pengujian dan pemeliharaan. Menurut (Pressman, 1997) model *waterfall* melakukan pendekatan pengembangan perangkat lunak yang sistematis dan sekuensial dimulai pada tingkat kemajuan sistem sampai analisis, desain, kode, pengujian, dan pemeliharaan. Sebagaimana ditunjukkan pada gambar Gambar 3.3.



Gambar 3.3 Diagram SDLC Waterfall (Pressman, 1997)

#### 3.5.1 Analisis Kebutuhan

Tahap pertama pada proses perancangan sistem adalah tahap analisis kebutuhan. Pada tahap ini dilakukan pencarian data indekos yang dibutuhkan dalam pembuatan sistem, dalam proses ini dilakukan dengan cara melakukan wawancara terhadap beberapa pemilik indekos. Dalam penelitian ini, analisis data dilakukan selama dan setelah pengumpulan data. Pada bagian analisis data, penjelasan pemilik indekos melalui wawancara digunakan sebagai bahan peneliti untuk menentukan macam kriteria indekos. Kemudian berdasarkan kriteria indekos yang telah ditentukan, disebarakan kuisisioner pembobotan untuk masing-masing kriteria oleh beberapa pencari indekos, hasil kuisisioner yang didapat lalu di rata-rata untuk menentukan pembobotan

Data koordinat posisi user didapat dari metode *Location Based Service* yang bekerja menggunakan GPS dan direpresentasikan menggunakan *Leaflet Maps API*.



Data koordinat ini digunakan untuk menghitung jarak dari titik awal yang ditentukan pencari indekos menuju masing-masing indekos. Hasil perhitungan jarak ini kemudian digunakan untuk menampilkan hasil pencarian indekos terdekat. Untuk hasil indekos rekomendasi didapat dari perhitungan jarak tersebut dengan bobot masing-masing indekos menggunakan algoritma Bee Colony Optimization.

### 3.5.2 Desain Sistem

Tahap yang selanjutnya yaitu desain system menggunakan *Unified Modeling Language* (UML), UML merupakan sekumpulan pemodelan konvensi yang digunakan untuk menentukan atau menggambarkan sebuah system perangkat lunak dalam kaitannya dengan objek (Whitten, 2004). Diagram-diagram yang digunakan adalah sebagai berikut :

a. *Business Process*

*Business process* digunakan untuk menggambarkan masukan dan keluaran data yang dibutuhkan dan dihasilkan oleh sistem, media dari sistem, dan tujuan dari pembuatan sistem.

b. *Use Case Diagram*

*Use case diagram* merupakan dokumentasi yang menggambarkan fitur dan aktor yang dapat mengakses fitur tersebut pada sistem yang akan dibangun.

c. *Use Case Scenario*

*Use case scenario* digunakan untuk menjelaskan alur sistem sesuai dengan yang ada pada *Use case diagram* dan menjelaskan keadaan yang akan terjadi pada suatu *event* tertentu.

d. *Sequence Diagram*

*Sequence diagram* adalah diagram yang digunakan untuk menggambarkan interaksi yang terjadi antar objek di dalam sistem yang disusun pada sebuah urutan dan rangkaian waktu. Interaksi antarobjek tersebut meliputi, aktor yang mengakses fitur tersebut sesuai dengan yang ada pada *Use case diagram*, tampilan sistem, controller, model, dan pesan yang disampaikan jika terjadi suatu *event* tertentu.

e. *Activity Diagram*

*Activity diagram* menggambarkan alur aktivitas dalam sistem yang sedang dirancang, meliputi awal alur terjadi, *decision* yang terjadi, dan bagaimana alur berakhir.

f. *Class Diagram*

*Class diagram* adalah sebuah spesifikasi yang menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek.

g. *Entity Relationship Diagram* (ERD)

*Entity relationship diagram* merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi.

### 3.5.3 Implementasi

Implementasi sistem merupakan tahap untuk mengimplementasikan atau mengubah desain sistem yang telah dibuat kedalam kode program. Pembuatan sistem meliputi pembuatan desain *interface*, *coding*, dan *database*. Menggunakan tools *Sublime Text* sebagai editor dengan bahasa pemrograman *PHP* dengan bantuan *framework* *Laravel* dan tools *Xampp for windows* untuk *database* manajemen yang menggunakan DBMS *MariaDB*.

### 3.5.4 Pengujian

Pengujian aplikasi dilakukan dengan 2 cara yaitu:

- a. *White Box Testing* merupakan cara pengujian dengan melihat modul yang telah dibuat dengan program-program yang ada oleh *developer*. Modul yang menghasilkan output yang tidak sesuai maka baris-baris program, variabel dan parameter yang terlibat pada unit tersebut satu persatu akan dicek dan diperbaiki (Pressman, 2001).
- b. *Black Box Testing* melibatkan pengguna/*user*, dimana hanya memperhatikan fungsionalitas yang berkaitan dengan masukan/keluaran (I/O) apakah sesuai dengan sistem yang dijalankan (Pressman, 2001).

### 3.5.5 Pemeliharaan

Pemeliharaan merupakan proses yang diperlukan ketika sistem telah digunakan oleh pengguna. Tahap ini berguna agar peneliti bisa memperbaiki kesalahan yang tidak ditemukan pada tahapan-tahapan sebelumnya serta untuk peningkatan kinerja sistem di masa mendatang.





## BAB 4. PERANCANGAN DAN PENGEMBANGAN SISTEM

Bab ini akan membahas tentang perancangan dan pengembangan aplikasi dengan menggunakan penerapan algoritma *Bee Colony Optimization* dalam pencarian lokasi indekos di Kota Jember berbasis web. Proses desain dan perancangan akan dimulai dengan analisis kebutuhan fungsional dan nonfungsional dan dilanjutkan dengan desain sistem berdasarkan usecase diagram, skenario, activity diagram, sequence diagram, class diagram dan entity relation diagram (ERD), kemudian penulisan kode program, lalu pengujian sistem.

### 4.1 *Statement Of Purpose*

Aplikasi yang dirancang merupakan aplikasi untuk mencari lokasi indekos di kota Jember khususnya di sekitar Universitas Jember. Sistem ini bertujuan untuk mempermudah para pencari indekos menemukan indekos sesuai kebutuhan. Dalam perancangan sistem ini terdapat batasan-batasan. Sistem yang penulis buat ini berhubungan dengan pemetaan lokasi indekos dan rute untuk menuju lokasi indekos tersebut.

Aplikasi INDEKOS merupakan aplikasi untuk mencari lokasi indekos di kota jember yang berbasis web menggunakan metode *Location Based Service* (LBS) dan algoritma *Bee Colony Optimization* (BCO). Teknologi LBS digunakan untuk mendapatkan posisi user saat itu dan algoritma BCO digunakan untuk mendapatkan lokasi indekos paling rekomendasi. Aplikasi ini dapat digunakan para pencari indekos yang sedang mencari lokasi indekos terdekat dengan posisi user saat itu. Aplikasi ini memiliki fitur untuk dapat melihat detail indekos dan mendapatkan rute menuju ke lokasi indekos yang ingin dituju.

### 4.2 **Analisis Kebutuhan Sistem**

Kebutuhan sistem adalah kemampuan yang harus dimiliki oleh sistem untuk memenuhi apa yang diinginkan oleh pengguna. Tujuan dari proses tersebut adalah, Untuk mempermudah menganalisis sebuah sistem dibutuhkan dua jenis kebutuhan.

Kebutuhan fungsional dan kebutuhan nonfungsional ini ditentukan oleh penulis dengan refrensi wawancara.

#### 4.2.1. Kebutuhan Fungsional

Kebutuhan fungsional berisi proses-proses yang akan dilakukan oleh sistem. Kebutuhan fungsional dari aplikasi ini adalah sebagai berikut:

1. Sistem dapat menentukan keberadaan user saat menggunakan aplikasi berdasarkan Location Based Service.
2. Sistem dapat menampilkan data spasial berupa maps.
3. Sistem dapat menentukan lokasi indekos dari posisi user saat itu menggunakan implementasi algoritma *Bee Colony Optimization*.
4. Sistem dapat menentukan rute dan memunculkan informasi indekos.
5. Sistem dapat mengelola data menggunakan website sebagai server dan client.

#### 4.2.2. Kebutuhan Non-Fungsional

Kebutuhan non-fungsional merupakan hal yang dibutuhkan oleh sistem untuk mendukung aktivitas sistem sesuai dengan kebutuhan fungsional yang telah disusun. Kebutuhan non fungsional dari aplikasi ini adalah sebagai berikut:

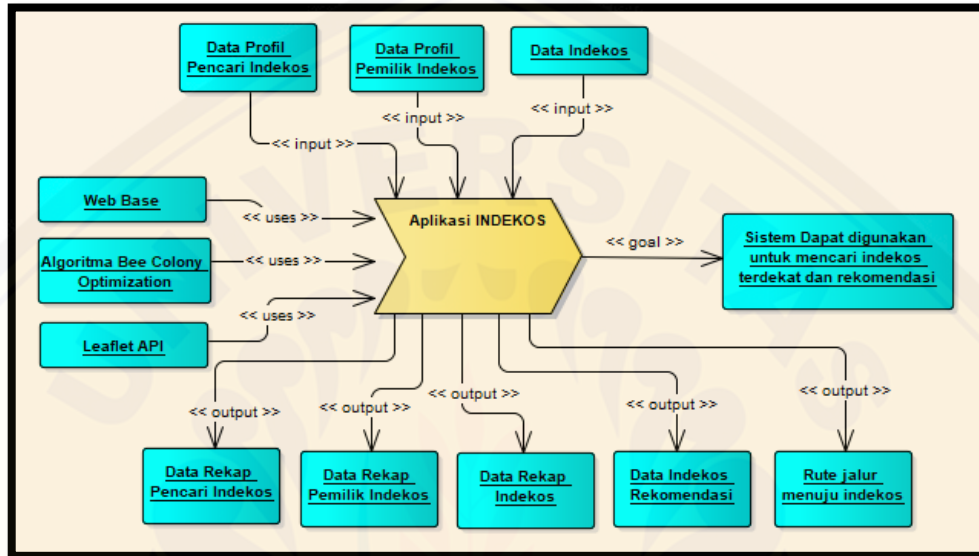
1. Sistem dirancang dengan tampilan *user-friendly* untuk memudahkan dalam penggunaan aplikasi.
2. Sistem dapat dijalankan oleh beberapa web browser diantaranya Internet Explore, Google Chrome dan Mozilla Firefox.
3. Aplikasi dapat diakses setiap saat selama 24 jam dalam 7 hari.

### 4.3 Desain Sistem

Tahapan yang dilakukan setelah melakukan analisis kebutuhan sistem yaitu tahap perencanaan pembangunan sistem yang dapat digambarkan dengan desain sistem. Desain sistem informasi penjualan beras dengan fitur peramalan penjualan menggunakan metode double moving average ini meliputi *use case diagram*, *use case skenario*, *activity diagram*, *sequence diagram*, *class diagram*, dan *entity relationship diagram* (ERD).

#### 4.3.1. Komponen Sistem

Komponen Sistem adalah sekumpulan elemen atau komponen yang berinteraksi untuk mencapai suatu tujuan tertentu. Komponen Sistem aplikasi INDEKOS dapat dilihat pada gambar 4.1.



Gambar 4.1 Komponen Sistem aplikasi INDEKOS

Penjelasan tentang komponen *input*, *uses*, *output*, dan *goal* akan dijelaskan di bawah ini:

##### 1. *Input*

###### a) Data Profil Pencari Indekos

Data Profil Pencari Indekos di inputkan sendiri oleh pencari indekos, data ini dibutuhkan untuk mengkonfirmasi kebenaran pemilik akun pencari indekos.

###### b) Data Profil Pemilik Indekos.

Data Profil Pemilik Indekos di inputkan sendiri oleh pemilik indekos, selain untuk mengkonfirmasi kebenaran pemilik indekos, juga berguna untuk dilampirkan kepada pencari indekos termasuk nomer telepon yang nantinya bisa dihubungi.

###### c) Data Indekos

Data Indekos mencakup juga bobot dari masing-masing kriteria yang dimiliki indekos dan juga koordinat lokasinya. Data ini digunakan sebagai parameter perhitungan algoritma *Bee Colony Optimization*.

##### 2. *Uses*

###### a) *Web Base*

Aplikasi pencari lokasi indekos jember dibangun berbasis website, sehingga dapat diakses melalui *mobilephone* maupun komputer.

b) Leaflet *API*

*Location Based Service* pada penelitian ini digunakan dan direpresentasikan dengan Aplikasi *Maps* yakni Leaflet *Maps*, dengan Leaflet *API Maps* kita bisa menentukan posisi indekos maupun posisi pencari indekos, menentukan jaraknya, serta menunjukkan rute perjalanan.

c) Algoritma *Bee Colony Optimization*

Algoritma *Bee Colony Optimization* digunakan dalam aplikasi ini untuk menentukan indekos rekomendasi.

3. *Output*

a) Data Rekap Pencari Indekos

Aplikasi ini dapat menampilkan rekap pencari indekos yang dapat diakses oleh admin.

b) Data Rekap Pemilik Indekos

Aplikasi ini dapat menampilkan rekap pemilik indekos yang dapat diakses oleh admin.

c) Data Rekap Indekos

Aplikasi ini dapat menampilkan rekap indekos yang sesuai dengan *filter* pilihan pencari indekos.

d) Data Indekos Rekomendasi

Aplikasi ini dapat menampilkan Indekos rekomendasi dari hasil pencarian indekos.

e) Rute Jalur Menuju Indekos

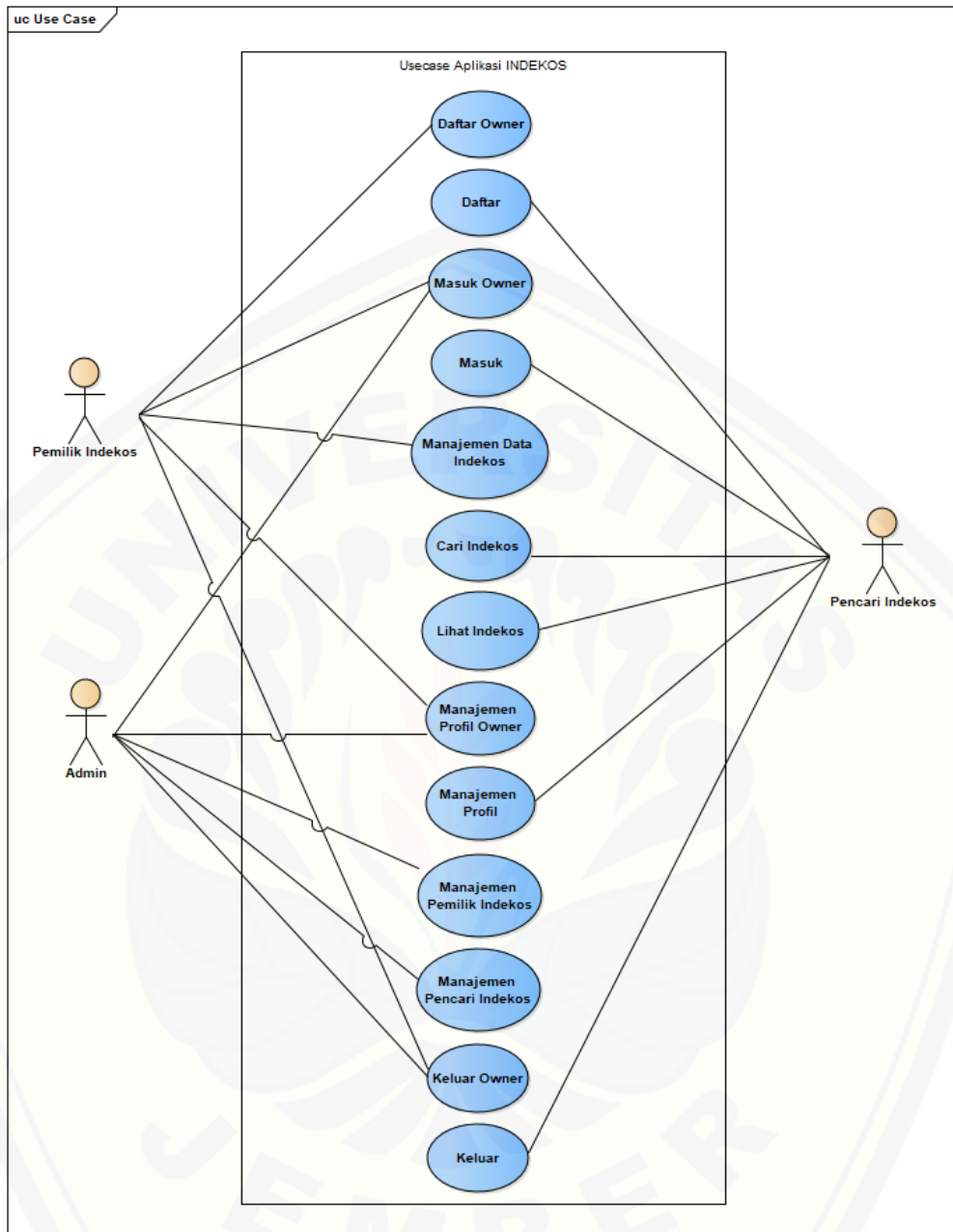
Aplikasi ini dapat menampilkan rute jalur perjalanan dari lokasi pilihan pencari indekos menuju lokasi indekos yang dipilih.

4. *Goal*

Sistem dapat digunakan untuk mencari indekos terdekat dan rekomendasi sesuai dengan kebutuhan dan pilihan pencari indekos.

4.3.2 *Usecase Diagram*

*Usecase Diagram* adalah dokumentasi untuk menggambarkan fitur dan aktor yang terdapat pada sistem yang akan dibuat. *Usecase* diagram sistem pencari indekos jember dapat dilihat seperti yang dijelaskan pada Gambar 4.2.



Gambar 4.2 Use Case Diagram sistem pencari indekos jember

Penjelasan tentang definisi aktor dan definisi *use case* dalam *use case diagram* pada Gambar 4.2 akan dijelaskan di bawah ini.

#### 1. Definisi Aktor

Tahap ini menjelaskan tentang aktor-aktor pengguna dari Aplikasi Pencarian Lokasi Indekos di Kota Jember Menggunakan *Location Based Service* (LBS) dan Algoritma *Bee Colony Optimization* yang akan dibangun. Terdapat 3 aktor seperti yang dijelaskan pada Tabel 4.1.



Tabel 4.1 Definsi Aktor

No.	Aktor	Deskripsi
1.	Admin	Aktor Admin memiliki hak akses yang dapat melakukan proses login, manajemen data profil, manajemen data pemilik indekos, dan manajemen data pencari indekos.
2.	Pemilik Indekos	Aktor Pemilik Indekos dapat melakukan proses daftar untuk kemudian memiliki hak akses yang dapat melakukan proses login, manajemen data profil, dan manajemen data indekos.
3.	Pencari Indekos	Aktor Pencari Indekos dapat melakukan proses daftar untuk kemudian memiliki hak akses yang dapat melakukan proses login, manajemen data profil, mencari data indekos, melihat detail indekos, dan melihat penunjuk arah menuju lokasi indekos.

## 2. Definisi *Usecase*

Definisi *Usecase* merupakan penjelasan dari masing-masing fitur yang terdapat di Aplikasi Pencarian Lokasi Indekos di Kota Jember Menggunakan *Location Based Service (LBS)* dan Algoritma *Bee Colony Optimization (BCO)* yang akan dibangun. Terdapat 3 aktor seperti yang dijelaskan pada Tabel 4.2.

Tabel 4.2 Definisi *Usecase*

No.	<i>Usecase</i>	Deskripsi
1.	Daftar <i>Owner</i>	Menggambarkan proses pendaftaran user pemilik indekos.
2.	Daftar	Menggambarkan proses pendaftaran user pencari indekos.



3.	Masuk <i>Owner</i>	Menggambarkan proses autentifikasi user admin dan pemilik indekos untuk masuk ke sistem.
4.	Masuk	Menggambarkan proses autentifikasi user pencari indekos untuk masuk ke sistem.
5.	Manajemen data indekos	Menggambarkan proses manajemen data indekos meliputi melihat dan menyunting.
6.	Cari Indekos	Menggambarkan proses pencarian dan melihat rekap data indekos sesuai <i>filter</i> dan algoritma bee colony optimization.
7.	Lihat Indekos	Menggambarkan proses melihat data detail indekos beserta rute perjalanan dari titik awal menuju titik lokasi indekos.
8.	Manajemen Profil <i>Owner</i>	Menggambarkan proses manajemen profil user pemilik indekos yang meliputi melihat dan menyunting.
9.	Manajemen Profil	Menggambarkan proses manajemen profil user pencari indekos yang meliputi melihat dan menyunting.
10.	Manajemen Pemilik Indekos	Menggambarkan proses manajemen data pemilik indekos baik melihat data profil maupun detail indekos, serta menghapus data tersebut.
11.	Manajemen Pencari Indekos	Menggambarkan proses manajemen data pencari indekos yang meliputi melihat dan menghapus data pencari indekos.
12.	Keluar <i>Owner</i>	Menggambarkan proses keluar dari system oleh user admin dan pemilik indekos.
13.	Keluar	Menggambarkan proses keluar dari sistem oleh user pencari indekos.

#### 4.3.3 *Usecase Scenario*

*Usecase scenario* digunakan untuk menjelaskan alur sistem sesuai dengan yang ada pada *Usecase diagram* seperti pada Gambar 4.2.

##### 1. Skenario *Usecase* Daftar *Owner*

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif skenario *usecase* Daftar *Owner* dijelaskan pada Lampiran A (*Usecase Scenario*).

##### 2. Skenario *Usecase* Daftar

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif skenario *usecase* Daftar dijelaskan pada Lampiran A (*Usecase Scenario*).

##### 3. Skenario *Usecase* Masuk *Owner*

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif skenario *usecase* Masuk *Owner* dijelaskan pada Lampiran A (*Usecase Scenario*).

##### 4. Skenario *Usecase* Masuk

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif skenario *usecase* Masuk dijelaskan pada Lampiran A (*Usecase Scenario*).

##### 5. Skenario *Usecase* Manajemen Data Indekos

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif skenario *usecase* Manajemen Data Indekos dijelaskan pada Lampiran A (*Usecase Scenario*).

##### 6. Skenario *Usecase* Cari Indekos

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal. Skenario *usecase* Cari Indekos dapat dilihat pada tabel 4.3.

Tabel 4.3 Skenario Cari Indekos

Nama	Cari Indekos
Aktor	Pencari Indekos
Pre-Kondisi	Pencari Indekos berhasil login

Post-Kondisi	Pencari Indekos berhasil Melihat Data Indekos
<b>SKENARIO NORMAL</b> “Cari Indekos”	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
1. Memilih Menu “Cari Kos”	
	<p>2. Menampilkan <i>Map</i> beserta titik Posisi user dan pilihan dropdown untuk filter dan prediksi perhitungan Algoritma <i>Bee Colony Optimization</i> yang berisi :</p> <ol style="list-style-type: none"> <li>a. Kamar Kosong</li> <li>b. Jenis Indekos</li> <li>c. Harga</li> <li>d. Perabot Kamar</li> <li>e. Dapur</li> <li>f. Penjaga</li> <li>g. Wifi</li> <li>h. Kamar mandi dalam</li> <li>i. Jam Malam</li> </ol> <p>Titik posisi user pada <i>Map</i> juga bisa diganti untuk memastikan posisi yang diinginkan user.</p>
3. Memilih jenis filter yang diinginkan	
4. Klik tombol “Cari Kos”	
	5. Ambil data dari database berdasarkan filter pilihan.
	6. Perhitungan probabilitas dan penerapan Algoritma <i>Bee Colony Optimization</i>
	<p>7. Menampilkan tabel indekos terdekat dan rekomendasi berdasarkan filter pilihan yang berisi :</p> <ol style="list-style-type: none"> <li>a. Nama Indekos</li> <li>b. Jarak</li> <li>c. Rekomendasi</li> <li>d. Bobot</li> <li>e. Probabilitas</li> <li>f. Probabilitas dibagi rata-rata</li> </ol> <p>Serta terdapat tombol “Tunjukkan” untuk masing-masing indekos yang tampil untuk melihat detail</p>

	indekos beserta penunjuk arah dari posisi user ke posisi indekos pilihan.
--	---

7. Skenario *Usecase* Lihat Indekos

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal skenario *usecase* Lihat Indekos dijelaskan pada Lampiran A (*Usecase Scenario*).

8. Skenario *Usecase* Manajemen Profil *Owner*

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif skenario *usecase* Manajemen Profil *Owner* dijelaskan pada Lampiran A (*Usecase Scenario*).

9. Skenario *Usecase* Manajemen Profil

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif skenario *usecase* Manajemen Profil dijelaskan pada Lampiran A (*Usecase Scenario*).

10. Skenario *Usecase* Manajemen Pemilik Indekos

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal skenario *usecase* Manajemen Pemilik Indekos dijelaskan pada Lampiran A (*Usecase Scenario*).

11. Skenario *Usecase* Manajemen Pencari Indekos

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal skenario *usecase* Manajemen Pencari Indekos dijelaskan pada Lampiran A (*Usecase Scenario*).

12. Skenario *Usecase* Keluar *Owner*

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal skenario *usecase* Keluar *Owner* dijelaskan pada Lampiran A (*Usecase Scenario*).

13. Skenario *Usecase* Keluar

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal skenario *usecase* Keluar dijelaskan pada Lampiran A (*Usecase Scenario*).

#### 4.3.4 *Activity Diagram*

*Activity diagram* menggambarkan alur aktivitas pada Aplikasi Pencarian Lokasi Indekos di Kota Jember Menggunakan *Location Based Service* (LBS) dan Algoritma *Bee Colony Optimization*.

##### 1. *Activity Diagram* Daftar Owner

*Activity diagram* Daftar Owner menggambarkan aksi aktor dan reaksi sistem pada fitur pendaftaran user pemilik Indekos pada aplikasi, yang dapat dilihat pada Lampiran B (*Activity Diagram*).

##### 2. *Activity Diagram* Daftar

*Activity diagram* Daftar dapat digunakan oleh pencari indekos. *Activity diagram* ini digunakan untuk pendaftaran user pencari indekos pada aplikasi, yang dapat dilihat pada Lampiran B (*Activity Diagram*).

##### 3. *Activity diagram* Masuk Owner

*Activity diagram* Masuk Owner menggambarkan aksi aktor dan reaksi sistem pada saat *login* ke aplikasi, aktor yang bisa *login* disini adalah admin dan pemilik indekos. *Activity diagram* Masuk Owner dapat dilihat pada Lampiran B (*Activity Diagram*).

##### 4. *Activity Diagram* Masuk

*Activity diagram* Masuk menggambarkan aksi aktor dan reaksi sistem pada saat *login* ke aplikasi, aktor yang bisa *login* disini adalah pencari indekos. *Activity diagram* Masuk dapat dilihat pada Lampiran B (*Activity Diagram*).

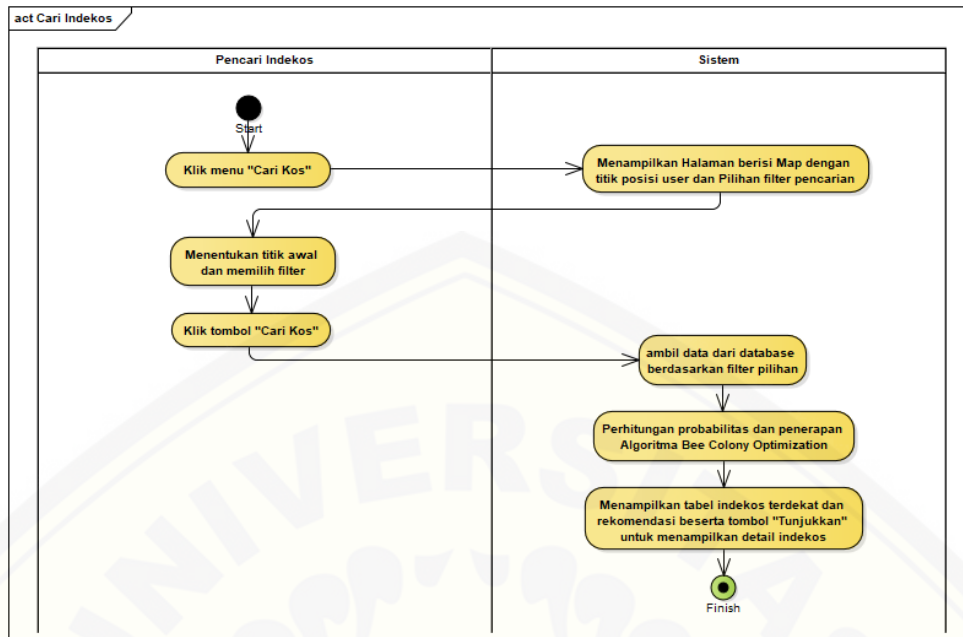
##### 5. *Activity Diagram* Manajemen Data Indekos

*Activity diagram* pada Manajemen Data Indekos dapat digunakan oleh pemilik indekos. *Activity diagram* ini digunakan untuk Melihat sekaligus *Edit* data Indekos, yang dapat dilihat pada Lampiran B (*Activity Diagram*).

##### 6. *Activity Diagram* Cari Indekos

*Activity diagram* Cari Indekos menggambarkan aksi aktor dan reaksi sistem pada fitur pencarian indekos, aktor yang bisa menjalankan fitur ini adalah pencari indekos. *Activity Diagram* mencari data indekos dapat dilihat pada Gambar 4.3.





Gambar 4.3 Activity Diagram Cari Indeks

#### 7. Activity Diagram Lihat Indeks

Activity diagram Lihat Indeks dapat digunakan oleh pencari indeks. Activity diagram ini digunakan untuk melihat detail indeks beserta Rute dari lokasi pencari menuju lokasi indeks, yang dapat dilihat pada Lampiran B (Activity Diagram).

#### 8. Activity Diagram Manajemen Profil Owner

Activity diagram Manajemen Profil Owner menggambarkan aksi aktor dan reaksi sistem pada fitur Manajemen Profil yang dapat digunakan oleh pemilik indeks dan admin untuk Melihat sekaligus Edit data profil, yang dapat dilihat pada Lampiran B (Activity Diagram).

#### 9. Activity Diagram Manajemen Profil

Activity diagram Manajemen Profil menggambarkan aksi aktor dan reaksi sistem pada fitur Manajemen Profil yang dapat digunakan oleh pencari indeks untuk Melihat sekaligus Edit data profil, yang dapat dilihat pada Lampiran B (Activity Diagram).

#### 10. Activity Diagram Manajemen Pemilik Indeks

Activity diagram Manajemen Pemilik Indeks dapat digunakan oleh admin. Activity Diagram ini menggambarkan aksi aktor dan reaksi sistem pada fitur



Manajemen data pemilik indekos, seperti melihat informasi detail indekos maupun menghapus seluruh data dari pemilik indekos, yang dapat dilihat pada Lampiran B (*Activity Diagram*).

#### 11. *Activity Diagram* Manajemen Pencari Indekos

*Activity diagram* Manajemen Pencari Indekos dapat digunakan oleh admin. *Activity Diagram* ini menggambarkan aksi aktor dan reaksi sistem pada fitur Manajemen data pencari indekos, seperti melihat maupun menghapus data pemilik indekos tertentu, yang dapat dilihat pada Lampiran B (*Activity Diagram*).

#### 12. *Activity Diagram* Keluar Owner

*Activity diagram* Keluar Owner menggambarkan aksi aktor dan reaksi sistem pada saat *logout* dari aplikasi, aktor yang bisa *logout* disini adalah admin dan pemilik indekos. *Activity diagram* Keluar Owner dapat dilihat pada Lampiran B (*Activity Diagram*).

#### 13. *Activity Diagram* Keluar

*Activity diagram* Keluar menggambarkan aksi aktor dan reaksi sistem pada saat *logout* dari aplikasi, aktor yang bisa *logout* disini adalah pencari indekos. *Activity diagram* Keluar dapat dilihat pada Lampiran B (*Activity Diagram*).

### 4.3.5 *Sequence Diagram*

*Sequence diagram* adalah diagram yang digunakan untuk menggambarkan interaksi yang terjadi antar objek di dalam sistem yang disusun pada sebuah urutan dan rangkaian waktu pada Aplikasi Pencarian Lokasi Indekos di Kota Jember Menggunakan *Location Based Service* (LBS) dan Algoritma *Bee Colony Optimization* (BCO).

#### 1. *Sequence Diagram* Daftar Owner

Penggambaran *sequence diagram* Daftar Owner digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada Lampiran C (*Sequence Diagram*).

2. *Sequence Diagram* Daftar

Penggambaran *sequence diagram* Daftar digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada Lampiran C (*Sequence Diagram*).

3. *Sequence diagram* Masuk Owner

Penggambaran *sequence diagram* Masuk Owner digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada Lampiran C (*Sequence Diagram*).

4. *Sequence Diagram* Masuk

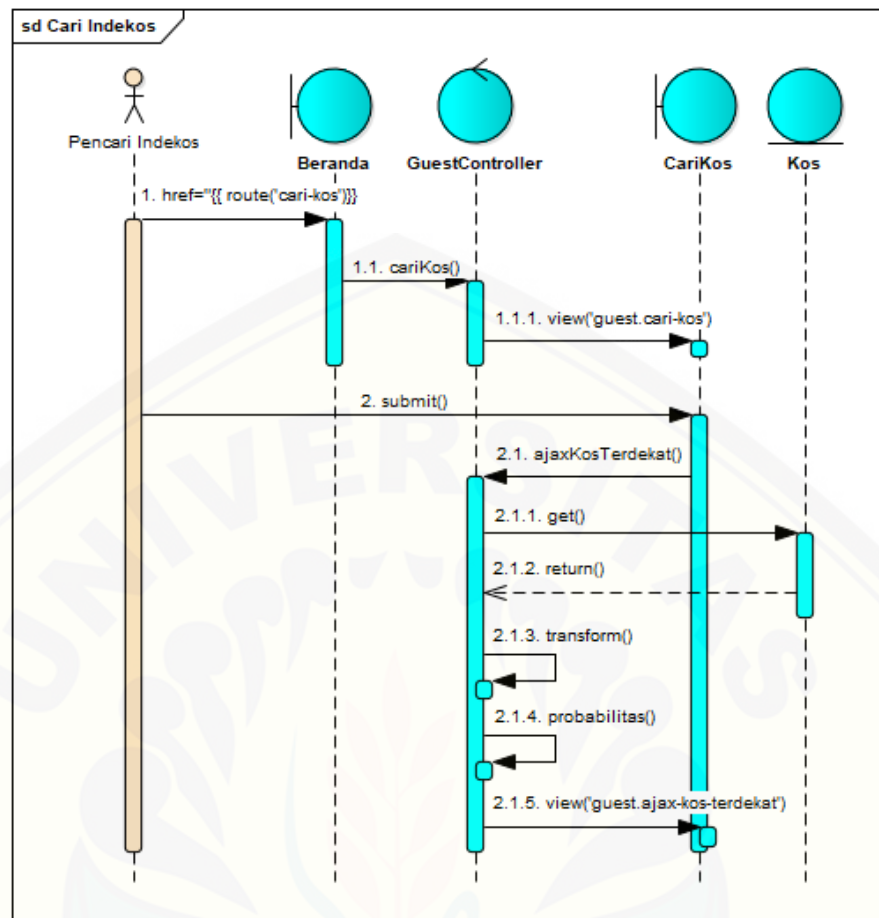
Penggambaran *sequence diagram* Masuk digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada Lampiran C (*Sequence Diagram*).

5. *Sequence Diagram* Manajemen Data Indekos

Penggambaran *sequence diagram* Manajemen Data Indekos digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada Lampiran C (*Sequence Diagram*).

6. *Sequence Diagram* Cari Indekos

Penggambaran *sequence diagram* Cari Indekos digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada Gambar 4.4.

Gambar 4.4 *Sequence Diagram* Cari Indeks

#### 7. *Sequence Diagram* Lihat Indeks

Penggambaran *sequence diagram* Lihat Indeks digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada Lampiran C (*Sequence Diagram*).

#### 8. *Sequence Diagram* Manajemen Profil Owner

Penggambaran *sequence diagram* Manajemen Profil *Owner* digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada Lampiran C (*Sequence Diagram*).

#### 9. *Sequence Diagram* Manajemen Profil

Penggambaran *sequence diagram* Manajemen Profil digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada Lampiran C (*Sequence Diagram*).

10. *Sequence Diagram* Manajemen Pemilik Indekos

Penggambaran *sequence diagram* Manajemen Pemilik Indekos digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada Lampiran C (*Sequence Diagram*).

11. *Sequence Diagram* Manajemen Pencari Indekos

Penggambaran *sequence diagram* Manajemen Pencari Indekos digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada Lampiran C (*Sequence Diagram*).

12. *Sequence Diagram* Keluar Owner

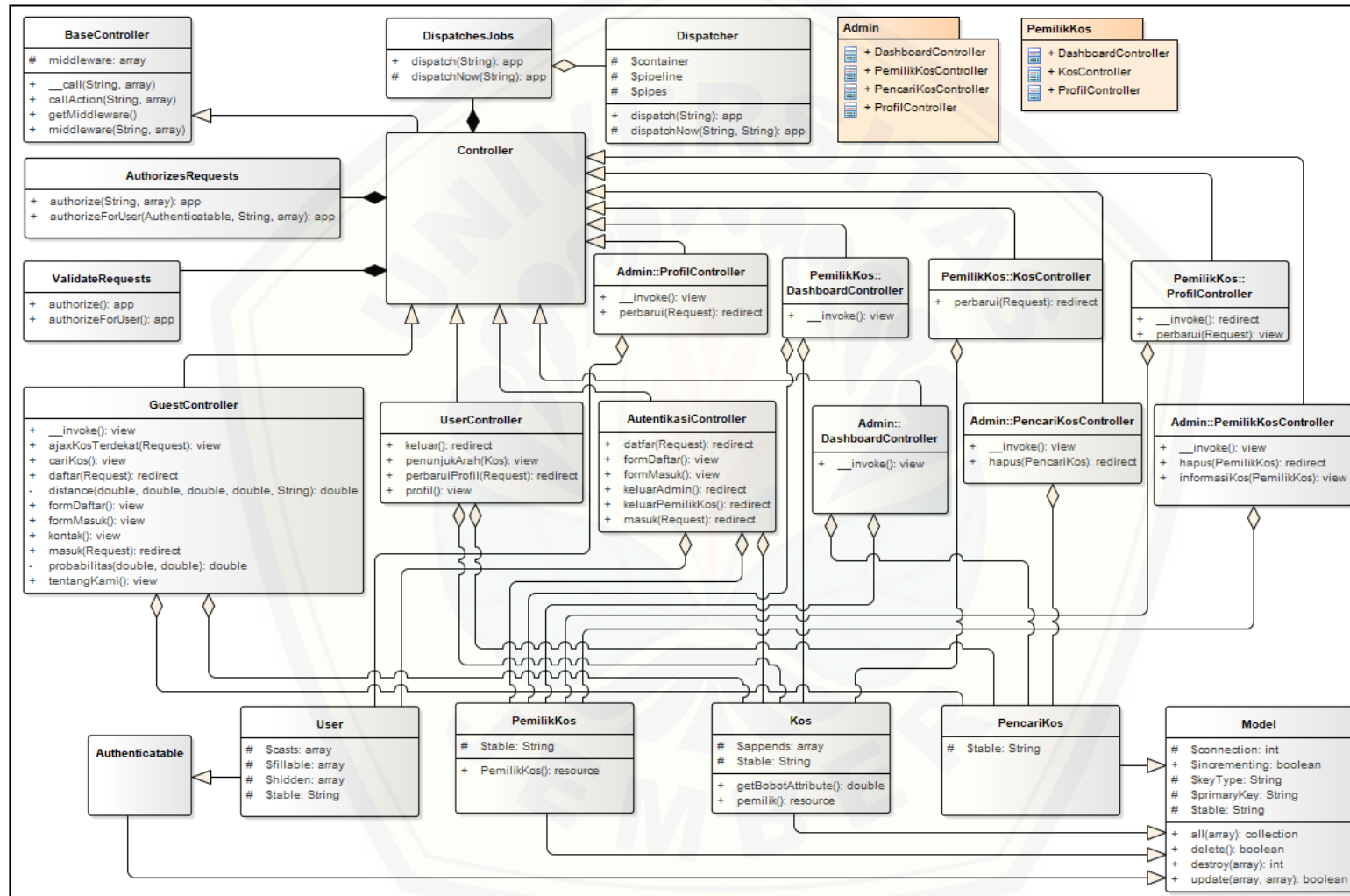
Penggambaran *sequence diagram* Keluar Owner digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada Lampiran C (*Sequence Diagram*).

13. *Sequence Diagram* Keluar

Penggambaran *sequence diagram* Keluar digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada Lampiran C (*Sequence Diagram*).

#### 4.3.6 *Class Diagram*

*Class Diagram* menggambarkan hubungan antar kelas yang digunakan untuk membangun suatu sistem. *Class Diagram* Aplikasi Pencarian Lokasi Indekos di Kota Jember Menggunakan *Location Based Service* (LBS) dan Algoritma *Bee Colony Optimization* (BCO) dapat dilihat pada Gambar 4.5.

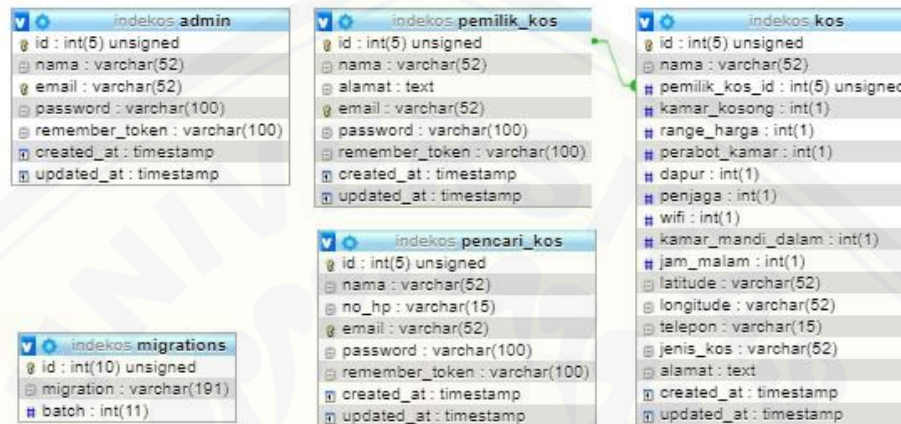


Gambar 4.5 Class Diagram Aplikasi Pencarian Lokasi Indekos di Kota Jember Menggunakan Location Based Service (LBS) dan Algoritma Bee Colony Optimization (BCO)



## 4.3.7 Entity Relationship Diagram (ERD)

ERD merupakan gambaran komponen dan struktur database yang digunakan dalam pembangunan sistem. ERD Aplikasi Pencarian Lokasi Indekos di Kota Jember Menggunakan *Location Based Service* (LBS) dan Algoritma *Bee Colony Optimization* (BCO) dapat dilihat pada Gambar 4.6.



Gambar 4.6 Entity Relationship Diagram

## 4.4 Penulisan Kode Program

Desain yang telah dibuat akan diimplementasikan ke dalam kode program. Bahasa pemrograman yang digunakan dalam sistem ini adalah bahasa Pemrograman PHP dengan framework Laravel sedangkan DBMS yang digunakan adalah MariaDB. Penulisan kode program pada fitur Pencarian Lokasi Indekos menggunakan *Location Based Service* (LBS) dan Algoritma *Bee Colony Optimization* (BCO) dapat dilihat pada Gambar 4.7 sampai 4.9.

```

95     private function probabilitas($arcFitness, $jarak)
96     {
97         $alpha      = 1;
98         $beta       = 2;
99         $diJ        = 1/$jarak;
100        $transisi    = pow($arcFitness, $alpha);
101        $jarakTransisi = pow($diJ, $beta);
102        $hasil      = $transisi * $jarakTransisi;
103        return $hasil;
104    }

```

Gambar 4.7 Kode Program *function* probabilitas

Gambar 4.7 merupakan *function* yang terletak di kelas *GuestController* yang digunakan untuk menghitung probabilitas tiap indekos.



```

201 private function distance($lat1, $lon1, $lat2, $lon2, $unit) {
202     if (($lat1 == $lat2) && ($lon1 == $lon2)) {
203         return 0;
204     }
205     else {
206         $theta = $lon1 - $lon2;
207         $dist = sin(deg2rad($lat1)) * sin(deg2rad($lat2)) + cos(deg2rad($lat1
208             )) * cos(deg2rad($lat2)) * cos(deg2rad($theta));
209         $dist = acos($dist);
210         $dist = rad2deg($dist);
211         $miles = $dist * 60 * 1.1515;
212         $unit = strtoupper($unit);
213
214         if ($unit == "K") {
215             return ($miles * 1.609344);
216         } else if ($unit == "N") {
217             return ($miles * 0.8684);
218         } else {
219             return $miles;
220         }
221     }
}

```

Gambar 4.8 Kode Program *function* distance

Gambar 4.8 merupakan *function* yang terletak di kelas *GuestController* untuk menghitung jarak titik yang nantinya akan digunakan dalam perhitungan probabilitas.

```

106 public function ajaxKosTerdekat(Request $request)
107 {
108     // \DB::enableQueryLog();
109     $kosTerdekat = Kos::where('id', '>', 0);
110     $kamar_kosong = $request->kamar_kosong;
111     $jenis_kos = $request->jenis_kos;
112     $harga = $request->harga;
113     $perabot_kamar = $request->perabot_kamar;
114     $dapur = $request->dapur;
115     $penjaga = $request->penjaga;
116     $wifi = $request->>wifi;
117     $kamar_mandi_dalam = $request->kamar_mandi_dalam;
118     $jam_malam = $request->jam_malam;
119     if($kamar_kosong != 'Semua'){
120         if($kamar_kosong == 'Ya')
121             $kosTerdekat = $kosTerdekat->where('kamar_kosong', '9');
122         else
123             $kosTerdekat = $kosTerdekat->where('kamar_kosong', '5');
124     }
125     if($jenis_kos != 'Bebas')
126         $kosTerdekat = $kosTerdekat->where('jenis_kos', $jenis_kos);
127     if($harga != 'Semua'){
128         if($harga == '<= 300 ribu')
129             $kosTerdekat = $kosTerdekat->where('range_harga', '9');
130         else
131             $kosTerdekat = $kosTerdekat->where('range_harga', '5');
132     }
}

```

```
133     if($perabot_kamar != 'Semua'){
134         if($perabot_kamar == 'Ya')
135             $kosTerdekat = $kosTerdekat->where('perabot_kamar', '9');
136         else
137             $kosTerdekat = $kosTerdekat->where('perabot_kamar', '5');
138     }
139     if($dapur != 'Semua'){
140         if($dapur == 'Ya')
141             $kosTerdekat = $kosTerdekat->where('dapur', '9');
142         else
143             $kosTerdekat = $kosTerdekat->where('dapur', '5');
144     }
145     if($penjaga != 'Semua'){
146         if($penjaga == 'Ya')
147             $kosTerdekat = $kosTerdekat->where('penjaga', '9');
148         else
149             $kosTerdekat = $kosTerdekat->where('penjaga', '5');
150     }
151     if($wifi != 'Semua'){
152         if($wifi == 'Ada')
153             $kosTerdekat = $kosTerdekat->where('wifi', '9');
154         else
155             $kosTerdekat = $kosTerdekat->where('wifi', '5');
156     }
157     if($kamar_mandi_dalam != 'Semua'){
158         if($kamar_mandi_dalam == 'Ada')
159             $kosTerdekat = $kosTerdekat->where('kamar_mandi_dalam', '9');
160         else
161             $kosTerdekat = $kosTerdekat->where('kamar_mandi_dalam', '5');
162     }
163     if($jam_malam != 'Semua'){
164         if($jam_malam == 'Ya')
165             $kosTerdekat = $kosTerdekat->where('jam_malam', '9');
166         else
167             $kosTerdekat = $kosTerdekat->where('jam_malam', '5');
168     }
169     $kosTerdekat = $kosTerdekat->get();
170     // return \DB::getQueryLog();
171     $kosTerdekat->transform(function($item) use ($request){
172         $lat = $request->lat;
173         $lng = $request->lng;
174         $jarak = round($this->distance($item->latitude, $item->longitude, $lat
175             , $lng, 'K'), 2);
176         $item->jarak = $jarak;
177         $item->probabilitas_sementara = $this->probabilitas($item->bobot, $
178             jarak);
179         return $item;
180     });
181     $totalProbabilitasSementara = $kosTerdekat->sum('probabilitas_sementara');
182     $kosTerdekat->transform(function($item) use ($totalProbabilitasSementara){
183         $item->probabilitas = $item->probabilitas_sementara / $
184             totalProbabilitasSementara;
185         return $item;
186     });
187     $sorted = collect($kosTerdekat->sortByDesc('probabilitas')->values()->all(
188         ));
189     $sorted->transform(function($item, $i){
190         if($i == 0){
191             $item->rekomendasi = '<font color="green">Rekomendasi</font>';
192         }else{
193             $item->rekomendasi = '';
194         }
195         return $item;
196     });
197     });
```

```
193     $sorted = $sorted->sortBy('jarak')->values()->all();
194     return view('guest.ajax-kos-terdekat', [
195         'kosTerdekat' => $sorted,
196         'lat'         => $request->lat,
197         'lng'         => $request->lng,
198     ]);
199 }
```

Gambar 4.9 Kode Program *function* ajaxKosTerdekat

Gambar 4.9 merupakan *function* yang terletak di kelas *GuestController* yang digunakan untuk menjalankan perhitungan jarak dan probabilitas sampai dengan memperoleh hasil akhir untuk digunakan untuk menampilkan rekomendasi indekos sesuai dengan pilihan *filter* yang dipilih.

Pada aplikasi ini kita juga memanfaatkan salahsatu fitur bawaan Laravel yaitu “Eloquent: Mutators” untuk menambahkan kolom *virtual* “bobot” pada tabel “Kos” di dalam model Kos yang juga digunakan untuk perhitungan probabilitas yang terdapat pada Gambar 4.9. Penulisan kode program pengisian kolom “bobot” dapat dilihat pada Gambar 4.10.

```
15     public function getBobotAttribute()
16     {
17         $jumlah = $this->range_harga*5
18         + $this->perabot_kamar * 3
19         + $this->dapur * 2
20         + $this->penjaga * 2
21         + $this->wifi * 2
22         + $this->jam_malam * 1
23         + $this->kamar_mandi_dalam * 1;
24         return $jumlah/100;
25     }
```

Gambar 4.10 Kode Program *function* getBobotAtribute

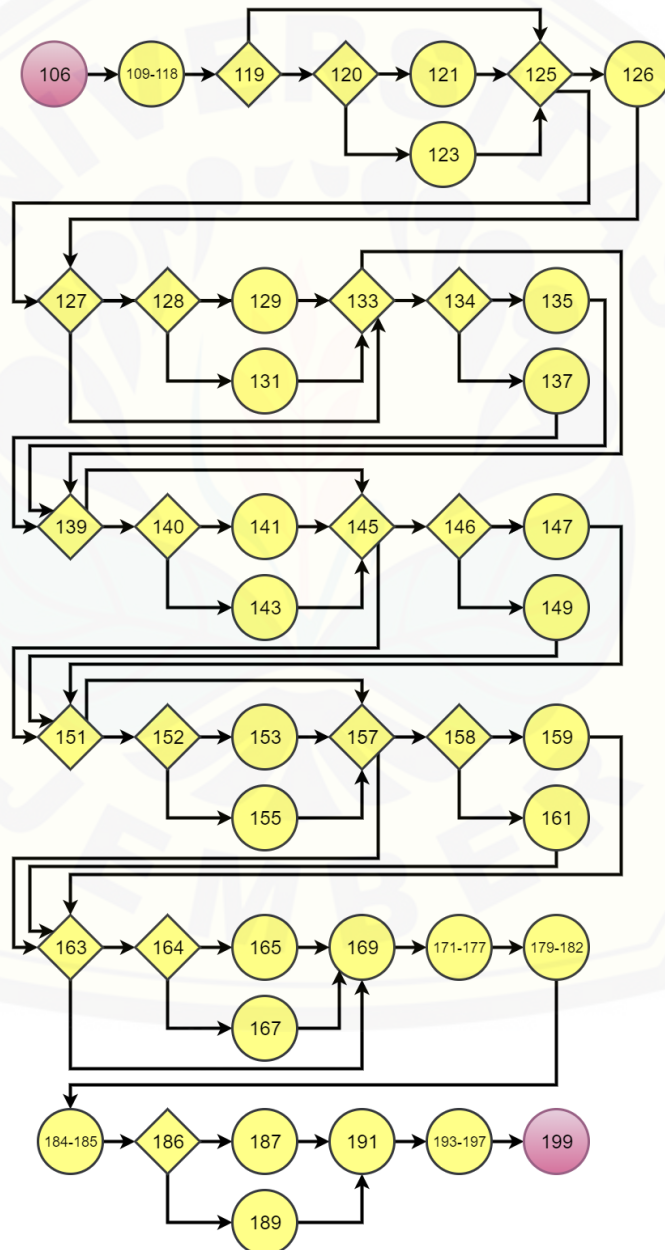
#### 4.5 Pengujian Sistem

Pengujian dilakukan untuk mengevaluasi kinerja dari aplikasi yang telah dibuat. Proses pengujian dilakukan dengan pengujian *whitebox* terlebih dahulu kemudian akan dilanjutkan dengan pengujian *blackbox*. Pengujian *whitebox* adalah pengujian dengan metode *test case* yang digambarkan berdasarkan notasi diagram alir. Pengujian *blackbox* adalah pengujian aspek fundamental sistem tanpa memperhatikan struktur logika internal perangkat lunak. Metode ini digunakan untuk mengetahui apakah perangkat lunak berfungsi dengan benar.



4.5.1 Pengujian *White Box*

Pengujian *white box* pada Aplikasi Pencarian Lokasi Indekos di Kota Jember Menggunakan *Location Based Service (LBS)* dan Algoritma *Bee Colony Optimization (BCO)* ini dengan cara menggambar diagram alir, menghitung *cyclomatic complexity (CC)*, dan membuat tabel pengujian *test case*. Pengujian kompleksitas siklomatik pada fitur penentuan indekos rekomendasi dapat dilihat pada Gambar 4.11 dan tabel pengujian *test case* dapat dilihat pada Tabel 4.6.



Gambar 4.11 Diagram Alir *Function ajaxKosTerdekat*

$$CC = Edge - Node + 2 = 63 - 46 + 2 = 19$$

Tabel 4.6 *Test Case Function* ajaxKosTerdekat

Test case	<b>Jika berhasil melakukan pencarian indeks terdekat dan rekomendasi sesuai pilihan pencari indeks.</b>
Target yang Diharapkan	Berhasil menampilkan data indeks terdekat dan rekomendasi sesuai pilihan pencari indeks.
Hasil pengujian	Benar
Path/jalur	<p>Path 1 = 106 – (109-118) – 119 – 125 – 127 – 133 – 139 – 145 – 151 – 157 – 163 – 169 – (171-177) – (179-182) – (184-185) – 186 – 187 – 191 – (193-197) – 199</p> <p>Path 2 = 106 – (109-118) – 119 – 125 – 127 – 133 – 139 – 145 – 151 – 157 – 163 – 169 – (171-177) – (179-182) – (184-185) – 186 – 189 – 191 – (193-197) – 199</p>

#### 4.5.2 Pengujian *Black Box*

Pengujian *black box* merupakan pengujian yang dilakukan oleh *user* dengan menjalankan program secara langsung dan menganalisis *input* dan *output* yang dihasilkan aplikasi. Pengujian sistem menggunakan *Black Box* untuk aktor pencari indeks dengan fitur pencarian indeks dapat dilihat pada Tabel 4.8. Untuk aktor admin dengan fitur manajemen user dan aktor pencari indeks dengan fitur manajemen indeks dapat dilihat pada lampiran D.

Tabel 4.8 Pengujian *Black Box* Pencarian Indekos

No.	Fitur	Aksi	Hasil	Kesimpulan
1.	Pencarian Indekos (Aktor Pencari Indekos)	Memilih menu “Cari Kos”	<p>a. Menampilkan Leaflet <i>Map</i> dengan titik posisi user saat mengakses aplikasi yang bias digeser.</p> <p>b. Menampilkan pilihan pencarian dalam bentuk <i>dropdown</i>.</p>	Berhasil
		Memilih Tombol “Cari Kos”	Menampilkan Tabel Indekos hasil pencarianurut berdasarkan Lokasi terdekat dan sebuah baris indekos dengan tanda Indekos Rekomendasi, serta tombol “Tunjukkan” di masing-masing indekos.	Berhasil
		Memilih tombol “Tunjukkan” di salah satu indekos	<p>a. Menampilkan Leaflet <i>Map</i> yang menunjukkan posisi awal pilihan dan posisi indekos pilihan beserta dengan rute perjalanannya.</p> <p>b. Menampilkan detail indekos beserta nomor telepon indekos yang bisa ditekan untuk langsung menelepon.</p>	Berhasil



## BAB 6. PENUTUP

Bab ini berisi mengenai kesimpulan dan saran dari peneliti tentang penelitian yang telah dilakukan. Kesimpulan dan saran tersebut diharapkan dapat digunakan sebagai acuan pada penelitian selanjutnya.

### 6.1 Kesimpulan

1. Algoritma *Bee Colony Optimization* diterapkan dalam perhitungan untuk menemukan indeks rekomendasi dengan prioritas yang didapat dari bobot (arc fitness) dan jarak antara user dengan setiap indeks. Di dalam penelitian ini, LBS digunakan untuk menentukan lokasi pencari dan lokasi indeks. Penerapan LBS direpresentasikan dengan menggunakan *Leaflet Maps*. *Leaflet Maps* tersebut bisa menangkap dan menampilkan lokasi pencari dan indeks serta rute perjalanannya.
2. Aplikasi Pencari Lokasi Indeks ini dibangun sesuai dengan kebutuhan dari penelitian dan dapat membantu para pencari indeks untuk mendapatkan informasi indeks lebih cepat dan efisien. Hasil dari pencarian berupa posisi indeks terdekat dan rekomendasi beserta rute menuju indeks pilihan menggunakan *Leaflet Maps*.

### 6.2 Saran

Beberapa saran dan masukan berikut diharapkan dapat memberikan perbaikan sistem dalam penelitian selanjutnya, antara lain:

1. Pada penelitian ini Aplikasi Pencari Lokasi Indeks penentuan rutenya menggunakan *plugin Leaflet-routing*. Penulis menyarankan agar dapat menentukan rute terdekat dengan Algoritma *Bee Colony Optimization* beserta penjelasan persimpangan yang dilewati pada *Leaflet Maps*.
2. Penilaian kriteria pada Penelitian ini hanya terbagi menjadi dua. Diharapkan penelitian selanjutnya menggunakan lebih banyak penilaian kriteria agar dapat menentukan indeks rekomendasi dengan lebih optimal.

3. Penelitian lebih lanjut di harapkan penelitian menggunakan algoritma perhitungan yang berbeda lalu melakukan perbandingan dengan penelitian ini dan menentukan mana algoritma yg paling tepat untuk menentukan indekos rekomendasi.
4. Pengembangan lebih lanjut untuk penelitian ini dapat dilakukan dengan membangun Aplikasi Pencari Lokasi Indekos pada *platform mainstream* Android ataupun juga *platform* lain seperti iOS, Windows Phone dan perangkat lainnya.



## DAFTAR PUSTAKA

Akbar, T., & Kurniawan. (2013). *Aplikasi Wisata Kota Bandung Menggunakan Metode Location Based Serviced pada Android*. Politeknik Telkom Bandung.

Amri, S. (2014). *Tutorial Dasar Pemograman google Maps API*. Depok.

Jeffery L. Whitten, L. D. 2004. *Metode Desain & Analisis Sistem*. Yogyakarta: Andi

Muhammad, Nurul, dan Budi (2017). *Sistem Optimasi Rute Tempat Wisata Kuliner Di Malang Menggunakan Algoritma Bee Colony*. Jurnal Pengembangan Ilmu Teknologi Informasi dan Ilmu Komputer. Vol 1 No 3.

Pawenang, P., & M. I. Pulungan, R. (2012). *Implementasi Algoritma Bee Colony Optimization pada Prototipe Intelligent logistics System*. *Symposium in Industrial technology*, 51-57.

Pressman, Roger.S. (1997), *Software Engineering: A Practioner's Approach 4th*, McGrawHill.

Safaat, N. (2013). *Aplikasi Berbasis Android*. Bandung: Informatika.

Samsudin.(2016),Oktober. *Pengertian GPS*. From [www.samsudinpunya.blogspot.com/2011/10/gps.html](http://www.samsudinpunya.blogspot.com/2011/10/gps.html). diakses pada Agustus 2019.

Septyaningsih, A. (2015). *Pencarian Lokasi Bengkel Mobil Di Kota Jember Menggunakan Location Based Service (LBS) dan Algoritma Bee Colony Optimization Berbasis Android*. Universitas Jember.

Suryo, S. S. (2013). *Perancangan Aplikasi GIS Pencarian Rute Terpendek Peta*. 1-15.

.

.

## LAMPIRAN

A. *Use Case Scenario*A.1 Skenario *Usecase* Daftar *Owner*

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif skenario *usecase* Daftar *Owner* dijelaskan pada tabel 1.

Tabel 1 Skenario Daftar *Owner*

Nama	Daftar <i>Owner</i>
Aktor	Pemilik Indekos
Pre-Kondisi	Pemilik Indekos berhasil membuka Aplikasi
Post-Kondisi	Pemilik Indekos berhasil mendaftar pada aplikasi
<b>SKENARIO NORMAL</b> "Daftar <i>Owner</i> "	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
1. Klik Menu "Daftar"	
	2. Menampilkan form pendaftaran pencari indekos.
3. Klik tombol "Daftar sebagai pemilik Indekos"	
	4. Menampilkan form pendaftaran pemilik indekos.
5. Mengisi form	
6. Klik tombol "Daftar"	
	7. Validasi form
	8. Memasukkan ke Database
	9. Membuat Session
	10. Load data indekos dari database
	11. Menampilkan form data indekos.
<b>SKENARIO ALTERNATIF</b> "Data form tidak valid"	
	8a. Menampilkan informasi error

A.2 Skenario *Usecase* Daftar

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif skenario *usecase* Daftar dijelaskan pada tabel 2.

Tabel 2 Skenario Daftar

Nama	Daftar
Aktor	Pencari Indekos
Pre-Kondisi	Pencari Indekos berhasil membuka Aplikasi
Post-Kondisi	Pencari Indekos berhasil mendaftar pada aplikasi
<b>SKENARIO NORMAL</b> "Daftar"	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
1. Klik Menu "Daftar"	
	2. Menampilkan form pendaftaran pencari indekos.
3. Mengisi form	
4. Klik tombol "Daftar"	
	5. Validasi form
	6. Memasukkan ke Database
	7. Membuat Session
	8. Menampilkan view beranda.
<b>SKENARIO ALTERNATIF</b> "Data form tidak valid"	
	6a. Menampilkan informasi error

A.3 Skenario *Usecase* Masuk *Owner*

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif skenario *usecase* Masuk *Owner* dijelaskan pada tabel 3.

Tabel 3 Skenario Masuk *Owner*

Nama	Masuk <i>Owner</i>
Aktor	Admin, Pemilik Indekos

Pre-Kondisi	Berhasil membuka Aplikasi
Post-Kondisi	Berhasil mendapatkan session dan masuk ke aplikasi.
<b>SKENARIO NORMAL</b> “Masuk Owner” untuk Admin	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
1. Klik menu “Masuk”	
	2. Menampilkan form login pencari indekos.
3. Klik tombol “Masuk sebagai pemilik indekos”	
	4. Menampilkan form login pemilik indekos.
5. Mengisi form	
6. Klik tombol “Masuk”	
	7. Validasi form
	8. Cek di database
	9. Pengecekan jenis user
	10. Membuat Session
	11. Menampilkan view admin dashboard
<b>SKENARIO ALTERNATIF</b> “Data form tidak valid”	
	8a. Menampilkan informasi error
<b>SKENARIO ALTERNATIF</b> “Data di database tidak ada”	
	9a. Menampilkan informasi error
<b>SKENARIO NORMAL</b> “Masuk Owner” untuk Pemilik Indekos	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
1. Klik menu “Masuk”	
	2. Menampilkan form login pencari indekos.



3. Klik tombol “Masuk sebagai pemilik indekos”	
	4. Menampilkan form login pemilik indekos.
5. Mengisi form	
6. Klik tombol “Masuk”	
	7. Validasi form
	8. Cek di database
	9. Pengecekan jenis user
	10. Membuat Session
	11. Load data indekos dari database
	12. Menampilkan form data indekos
<b>SKENARIO ALTERNATIF</b> “Data form tidak valid”	
	8a. Menampilkan informasi error
<b>SKENARIO ALTERNATIF</b> “Data di database tidak ada”	
	9a. Menampilkan informasi error

A.4 Skenario *Usecase* Masuk

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif skenario *usecase* Masuk dijelaskan pada tabel 4.

Tabel 4 Skenario Masuk

Nama	Masuk
Aktor	Pencari Indekos
Pre-Kondisi	Berhasil membuka Aplikasi
Post-Kondisi	Berhasil mendapatkan session dan masuk ke aplikasi.
<b>SKENARIO NORMAL</b> “Masuk”	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>

1. Klik menu “Masuk”	
	2. Menampilkan form login pencari indekos.
3. Mengisi form	
4. Klik tombol “Masuk”	
	5. Validasi form
	6. Cek di database
	7. Membuat Session
	8. Menampilkan view beranda
<b>SKENARIO ALTERNATIF</b> “Data form tidak valid”	
	6a. Menampilkan informasi error
<b>SKENARIO ALTERNATIF</b> “Data di database tidak ada”	
	7a. Menampilkan informasi error

A.5 Skenario *Usecase* Manajemen Data Indekos

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif skenario *usecase* Manajemen Data Indekos dijelaskan pada tabel 5.

Tabel 5 Skenario Manajemen Data Indekos

Nama	Manajemen Data Indekos
Aktor	Pemilik Indekos
Pre-Kondisi	Berhasil <i>login</i> ke aplikasi
Post-Kondisi	Berhasil menyunting data indekos.
<b>SKENARIO NORMAL</b> “Manajemen Data Indekos”	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
1. Mengisi form	
2. Klik tombol “Perbarui”	
	3. Validasi form

	4. Update data ke database
	5. Menampilkan form data indeks dengan informasi sukses.
<b>SKENARIO ALTERNATIF</b> “Data form tidak valid”	
	4a. Menampilkan informasi error

A.6 Skenario *Usecase* Cari Indeks

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal *scenario usecase* Cari Indeks dapat dilihat pada tabel 6.

Tabel 6 Skenario Cari Indeks

Nama	Cari Indeks
Aktor	Pencari Indeks
Pre-Kondisi	Pencari Indeks berhasil login
Post-Kondisi	Pencari Indeks berhasil Melihat Data Indeks
<b>SKENARIO NORMAL</b> “Cari Indeks”	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
1. Memilih Menu “Cari Kos”	
	<p>2. Menampilkan <i>Map</i> beserta titik Posisi user dan pilihan dropdown untuk filter dan prediksi perhitungan Algoritma <i>Bee Colony Optimization</i> yang berisi :</p> <ul style="list-style-type: none"> <li>j. Kamar Kosong</li> <li>k. Jenis Indeks</li> <li>l. Harga</li> <li>m. Perabot Kamar</li> <li>n. Dapur</li> <li>o. Penjaga</li> <li>p. Wifi</li> <li>q. Kamar mandi dalam</li> <li>r. Jam Malam</li> </ul> <p>Titik posisi user pada Map juga bisa diganti untuk memastikan posisi yang diinginkan user.</p>

3. Memilih jenis filter yang diinginkan	
4. Klik tombol “Cari Kos”	
	5. Ambil data dari database berdasarkan filter pilihan.
	6. Perhitungan probabilitas dan penerapan Algoritma Bee Colony Optimization
	7. Menampilkan tabel indeks terdekat dan rekomendasi berdasarkan filter pilihan yang berisi : g. Nama Indeks h. Jarak i. Rekomendasi j. Bobot k. Probabilitas l. Probabilitas dibagi rata-rata  Serta terdapat tombol “Tunjukkan” untuk masing-masing indeks yang tampil untuk melihat detail indeks beserta penunjuk arah dari posisi user ke posisi indeks pilihan.

A.7 Skenario *Usecase* Lihat Indeks

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal skenario *usecase* Lihat Indeks dijelaskan pada tabel 7.

Tabel 7 Skenario Lihat Indeks

Nama	Lihat Indeks
Aktor	Pencari Indeks
Pre-Kondisi	Pencari Indeks berhasil mencari indeks
Post-Kondisi	Berhasil melihat detail indeks beserta map dengan rute menuju indeks.
<b>SKENARIO NORMAL</b> “Lihat Indeks”	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
1. Klik tombol “Tunjukkan” di salah	

satu indekos pilihan pada tabel.	
	2. Ambil data dari database berdasarkan data yang dipilih.
	3. Menampilkan peta dengan penunjuk arahnya dan Detail indekos pilihan beserta tombol telepon.

A.8 Skenario *Usecase* Manajemen Profil *Owner*

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif skenario *usecase* Manajemen Profil *Owner* dijelaskan pada tabel 8.

Tabel 8 Skenario Manajemen Profil *Owner*

Nama	Manajemen Profil <i>Owner</i>
Aktor	Admin, Pemilik Indekos
Pre-Kondisi	Berhasil <i>login</i> ke aplikasi
Post-Kondisi	Berhasil menyunting data profil.
<b>SKENARIO NORMAL</b> “Manajemen Profil <i>Owner</i> ”	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
1. Klik dropdown Profil	
2. Klik menu “Profil”	
	3. Mengambil data profil dari database
	4. Menampilkan data profil pada form
5. Mengisi form	
6. Klik tombol “Perbarui”	
	7. Validasi form
	8. Update data ke database.
	9. Mengambil data profil dari database.
	10. Menampilkan data profil pada form dengan informasi sukses.

<b>SKENARIO ALTERNATIF</b> “Data form tidak valid”	
	8a. Menampilkan informasi error

#### A.9 Skenario *Usecase* Manajemen Profil

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif skenario *usecase* Manajemen Profil dijelaskan pada tabel 9.

Tabel 9 Skenario Manajemen Profil

Nama	Manajemen Profil
Aktor	Pencari Indekos
Pre-Kondisi	Berhasil <i>login</i> ke aplikasi
Post-Kondisi	Berhasil menyunting data profil.
<b>SKENARIO NORMAL</b> “Manajemen Profil”	
Aksi Aktor	Reaksi Sistem
1. Klik menu “Profil”	
	2. Mengambil data profil dari database
	3. Menampilkan data profil pada form
4. Mengisi form	
5. Klik tombol “Perbarui”	
	6. Validasi form
	7. Update data ke database.
	8. Mengambil data profil dari database.
	9. Menampilkan data profil pada form dengan informasi sukses.
<b>SKENARIO ALTERNATIF</b> “Data form tidak valid”	
	7a. Menampilkan informasi error



A.10 Skenario *Usecase* Manajemen Pemilik Indekos

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal skenario *usecase* Manajemen Pemilik Indekos dijelaskan pada tabel 10.

Tabel 10 Skenario Manajemen Pemilik Indekos

Nama	Manajemen Pemilik Indekos
Aktor	Admin
Pre-Kondisi	Berhasil <i>login</i> ke aplikasi
Post-Kondisi	Berhasil manajemen data pemilik indekos.
<b>SKENARIO NORMAL</b> “Manajemen Pemilik Indekos”	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
1. Klik menu “Pemilik Kos”	
	2. Mengambil data Pemilik Indekos dari database.
	3. Menampilkan tabel Pemilik Indekos beserta tombol "Informasi Kos" dan "Hapus"
<b>SKENARIO NORMAL</b> “Manajemen Pemilik Indekos: Lihat detail informasi indekos”	
4. Klik tombol "Informasi Kos"	
	5. Mengambil data indekos pilihan dari database
	6. Menampilkan Map sesuai titik dari data database dan Data Informasi Indekos
<b>SKENARIO NORMAL</b> “Manajemen Pemilik Indekos: Hapus pemilik indekos”	
4. Klik tombol "Hapus"	
	5. Konfirmasi hapus.
6. Klik “Ok”	
	7. Menghapus data dari database
	8. Mengambil data Pemilik Indekos dari database
	9. Menampilkan tabel pemilik indekos dan informasi berhasil menghapus

<b>SKENARIO ALTERNATIF</b> "Batal hapus"	
6a. Klik "Cancel"	
	7a. Kembali menampilkan tabel Pemilik Indekos beserta tombol "Informasi Kos" dan "Hapus"

A.11 Skenario *Usecase* Manajemen Pencari Indekos

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal skenario *usecase* Manajemen Pencari Indekos dijelaskan pada tabel 11.

Tabel 11 Skenario Manajemen Pencari Indekos

Nama	Manajemen Pencari Indekos
Aktor	Admin
Pre-Kondisi	Berhasil <i>login</i> ke aplikasi
Post-Kondisi	Berhasil manajemen data pencari indekos.
<b>SKENARIO NORMAL</b> "Manajemen Pencari Indekos"	
Aksi Aktor	Reaksi Sistem
1. Klik menu "Pencari Kos"	
	2. Mengambil data Pencari Indekos dari database.
	3. Menampilkan tabel Pencari Indekos beserta tombol "Hapus"
<b>SKENARIO NORMAL</b> "Manajemen Pemilik Indekos: Hapus pencari indekos"	
4. Klik tombol "Hapus"	
	5. Konfirmasi hapus.
6. Klik "Ok"	
	7. Menghapus data dari database
	8. Mengambil data Pencari Indekos dari database
	9. Menampilkan tabel pencari indekos dan informasi berhasil menghapus

<b>SKENARIO ALTERNATIF</b> “Batal hapus”	
6a. Klik “Cancel”	
	7a. Kembali menampilkan tabel Pencari Indekos beserta tombol "Hapus"

#### A.12 Skenario *Usecase* Keluar *Owner*

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal skenario *usecase* Keluar *Owner* dijelaskan pada tabel 12.

Tabel 12 Skenario Keluar *Owner*

Nama	Keluar <i>Owner</i>
Aktor	Admin, Pemilik Indekos
Pre-Kondisi	Sudah mendapatkan session dan <i>login</i> ke aplikasi
Post-Kondisi	Berhasil menghapus session dan keluar aplikasi.
<b>SKENARIO NORMAL</b> “Keluar <i>Owner</i> ”	
Aksi Aktor	Reaksi Sistem
1. Klik dropdown Profil	
2. Klik tombol “Keluar”	
	3. Menghapus session
	4. Menampilkan halaman masuk <i>owner</i>

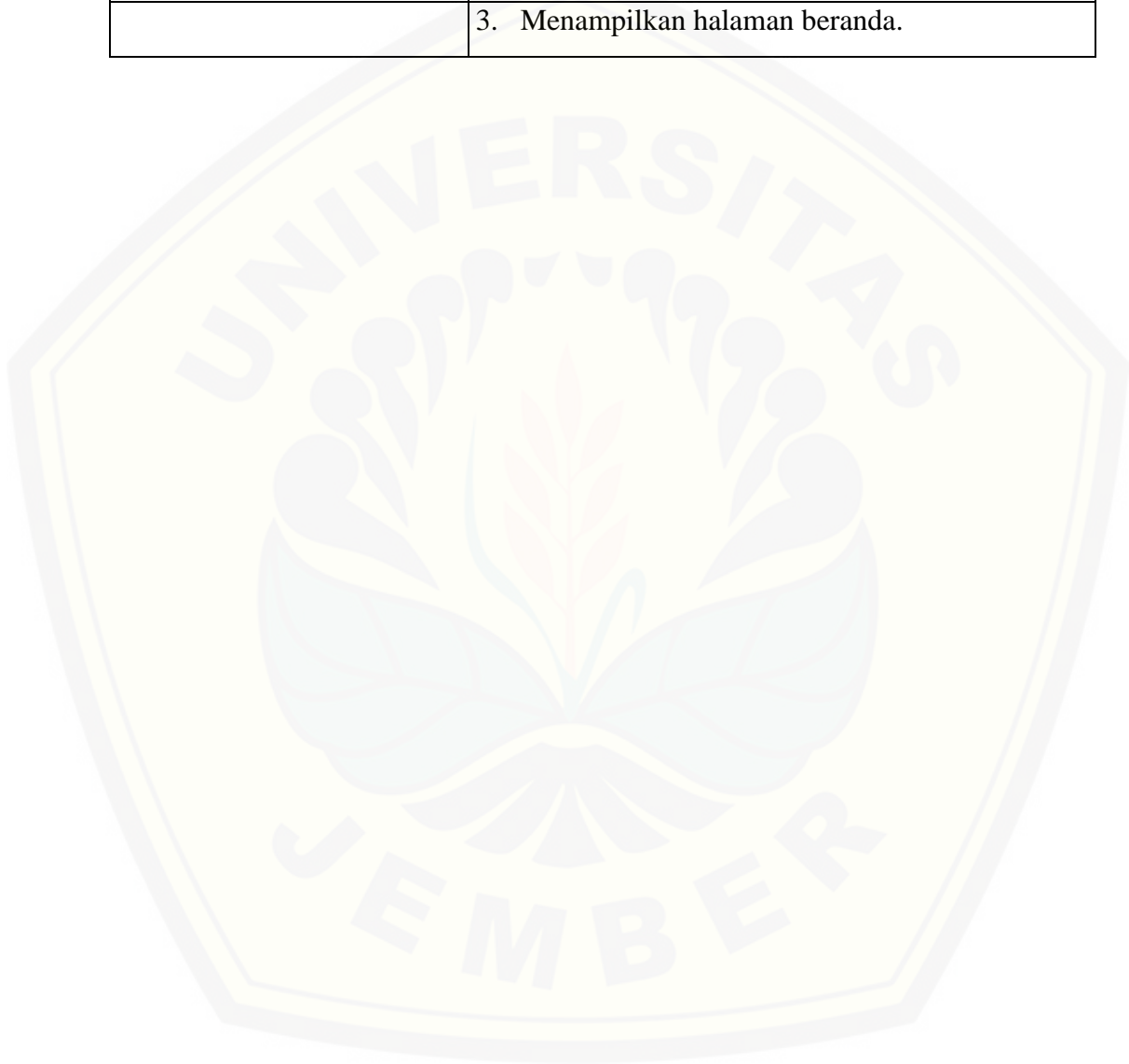
#### A.13 Skenario *Usecase* Keluar

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal skenario *usecase* Keluar dijelaskan pada tabel 13.

Tabel 13 Skenario Keluar

Nama	Keluar
Aktor	Pencari Indekos
Pre-Kondisi	Sudah mendapatkan session dan <i>login</i> ke aplikasi
Post-Kondisi	Berhasil menghapus session dan keluar aplikasi.

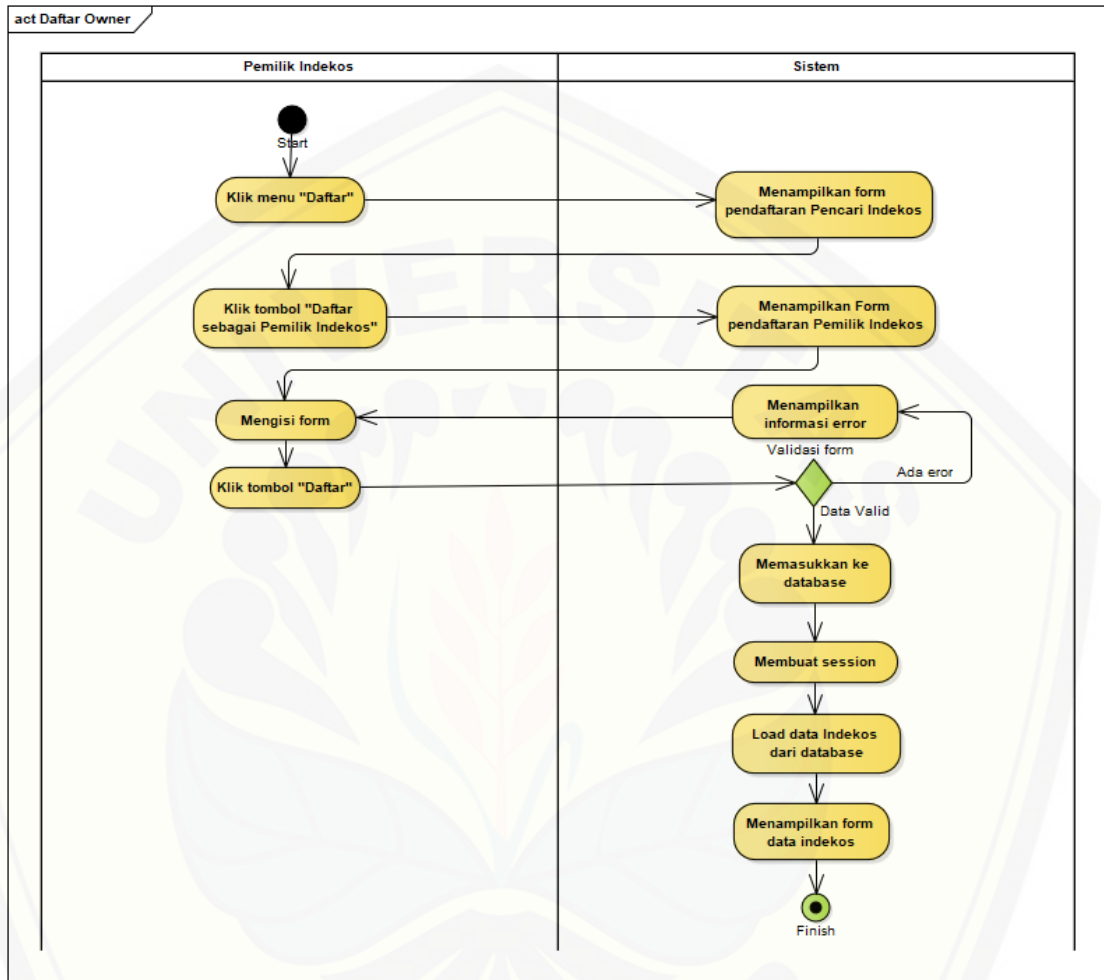
<b>SKENARIO NORMAL “Keluar”</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
1. Klik menu “Keluar”	
	2. Menghapus session
	3. Menampilkan halaman beranda.



**B. Activity Diagram**

**B.1 Activity Diagram Daftar Owner**

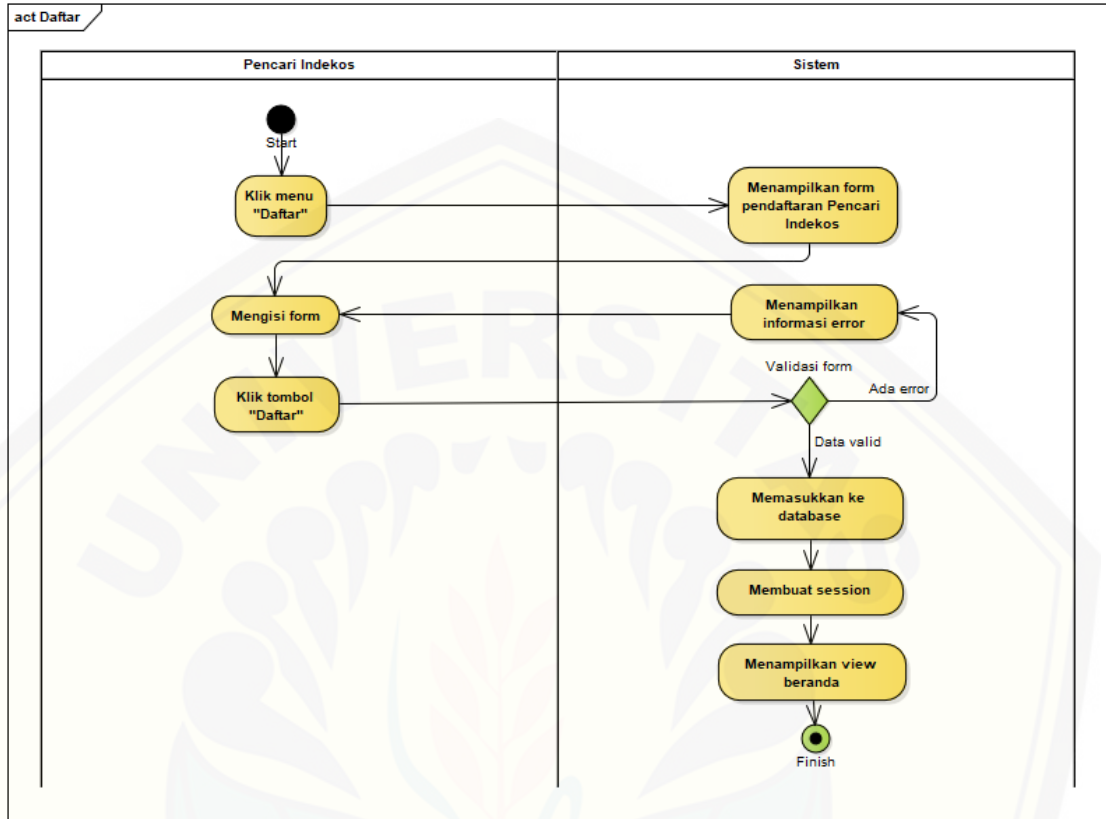
Activity diagram Daftar Owner dapat dilihat pada Gambar 1.



Gambar 1 Activity Diagram Daftar Owner

B.2 Activity Diagram Daftar

Activity diagram daftar pencari indekos dapat dilihat pada Gambar 2.

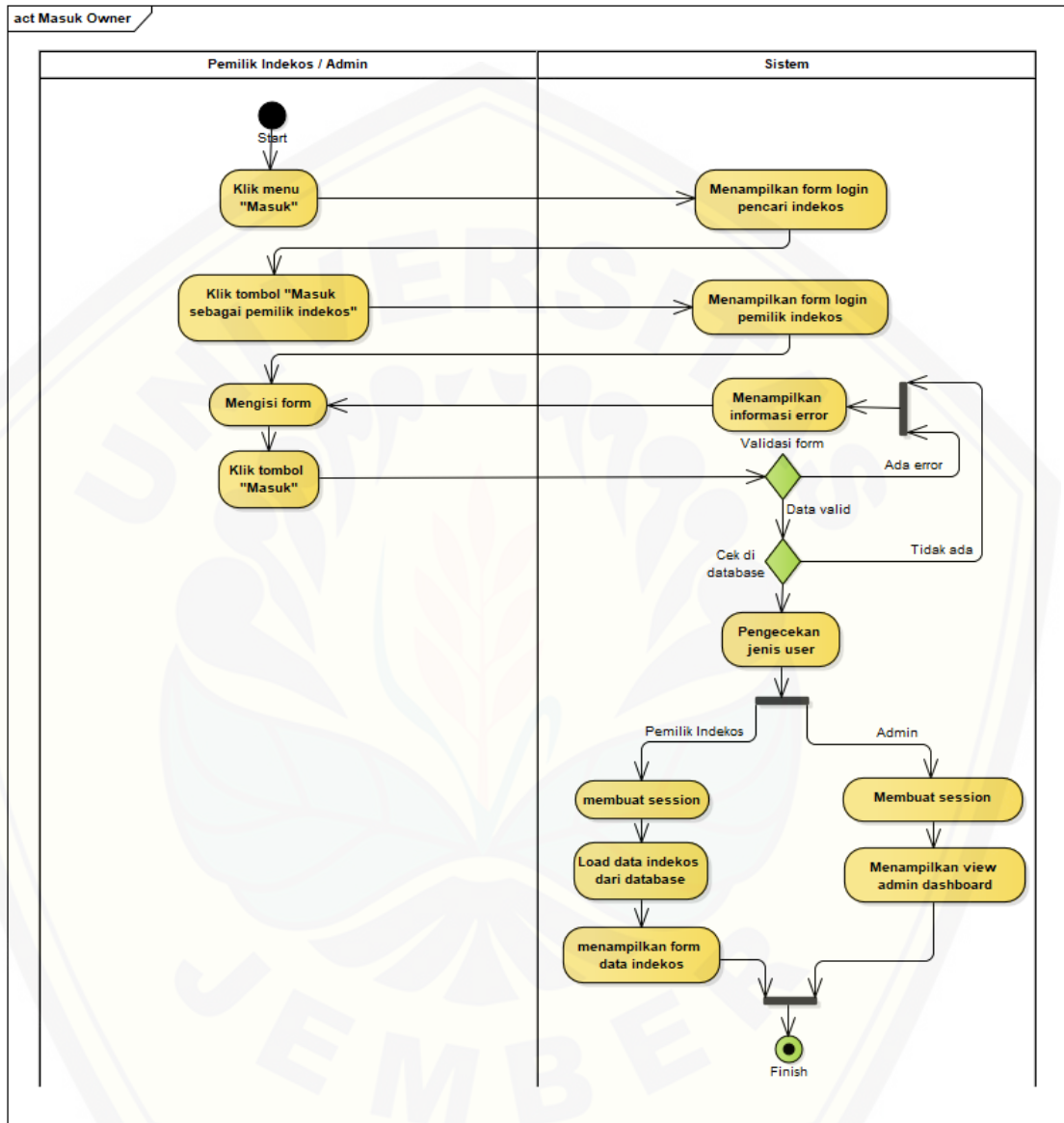


Gambar 2 Activity Diagram Daftar



B.3 Activity Diagram Masuk Owner

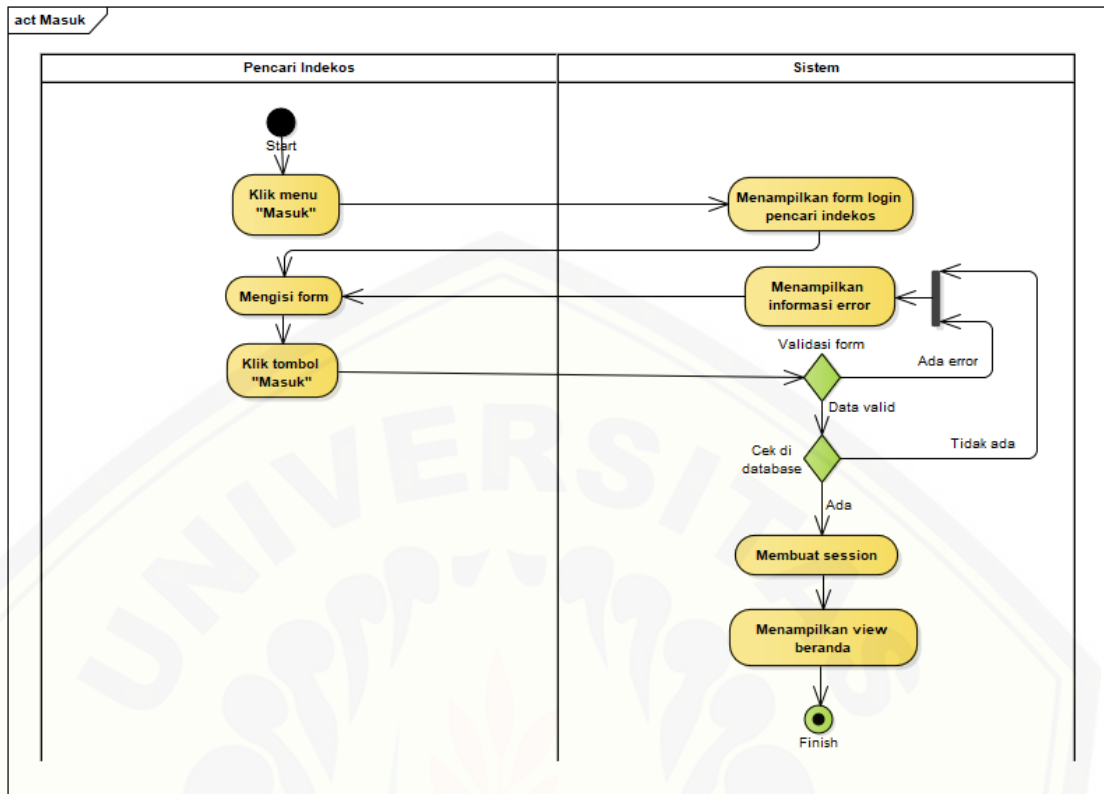
Activity diagram Masuk Owner dapat dilihat pada Gambar 3.



Gambar 3 Activity Diagram Masuk Owner

B.4 Activity Diagram Masuk

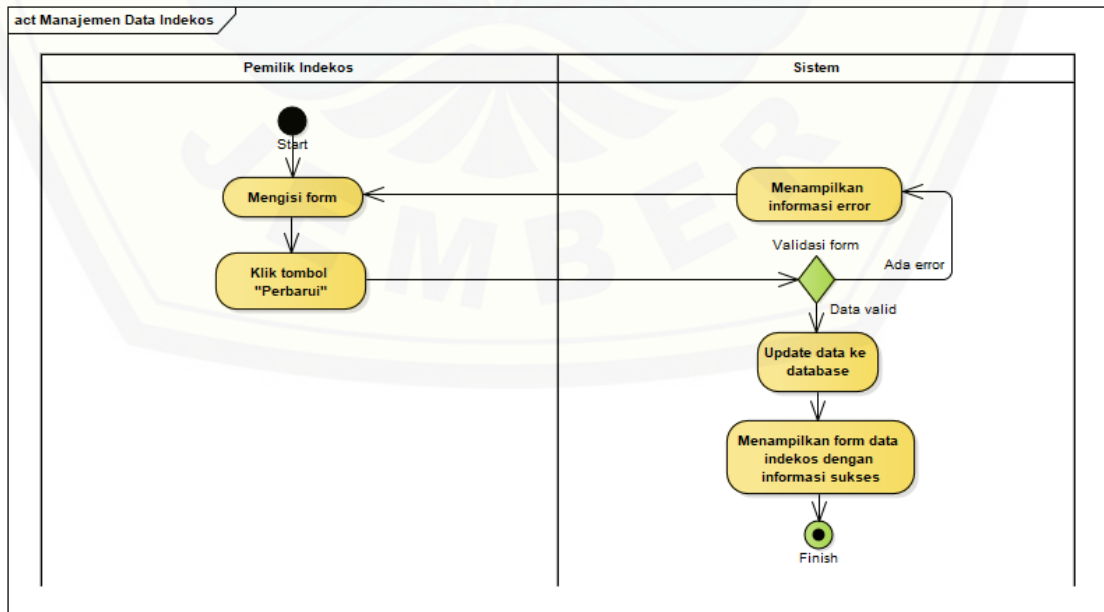
Activity Diagram Masuk dapat di lihat pada Gambar 4.



Gambar 4 Activity Diagram Masuk

B.5 Activity Diagram Manajemen Data Indekos

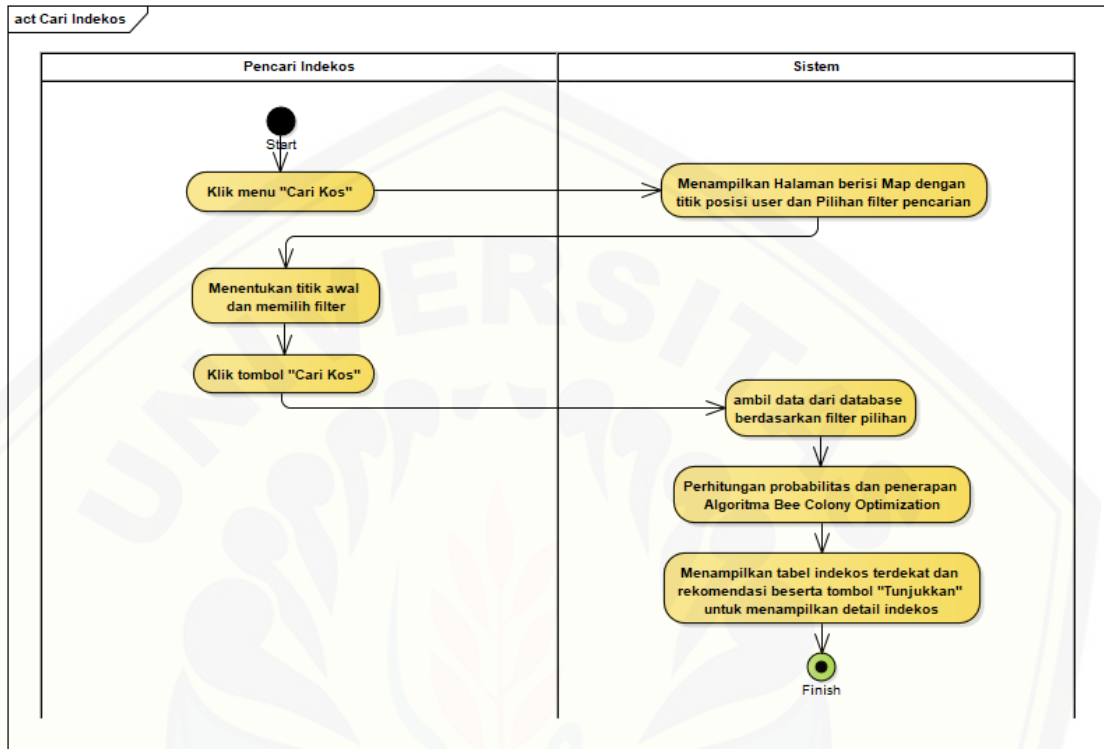
Activity Diagram Manajemen Data Indekos dapat di lihat pada Gambar 5.



Gambar 5 Activity Diagram Manajemen Data Indekos

B.6 *Activity Diagram* Cari Indekos

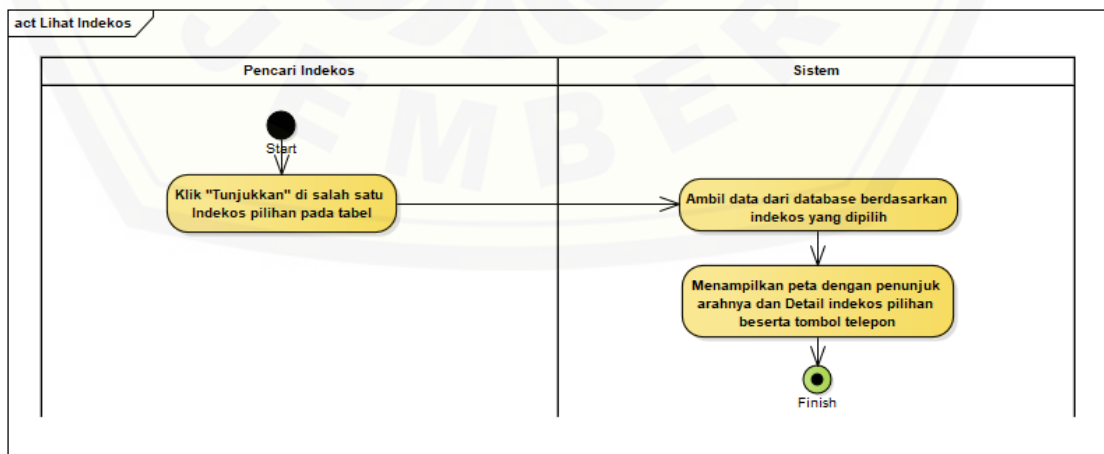
*Activity Diagram* Cari Indekos dapat di lihat pada Gambar 6.



Gambar 6 *Activity Diagram* Cari Indekos

B.7 *Activity Diagram* Lihat Indekos

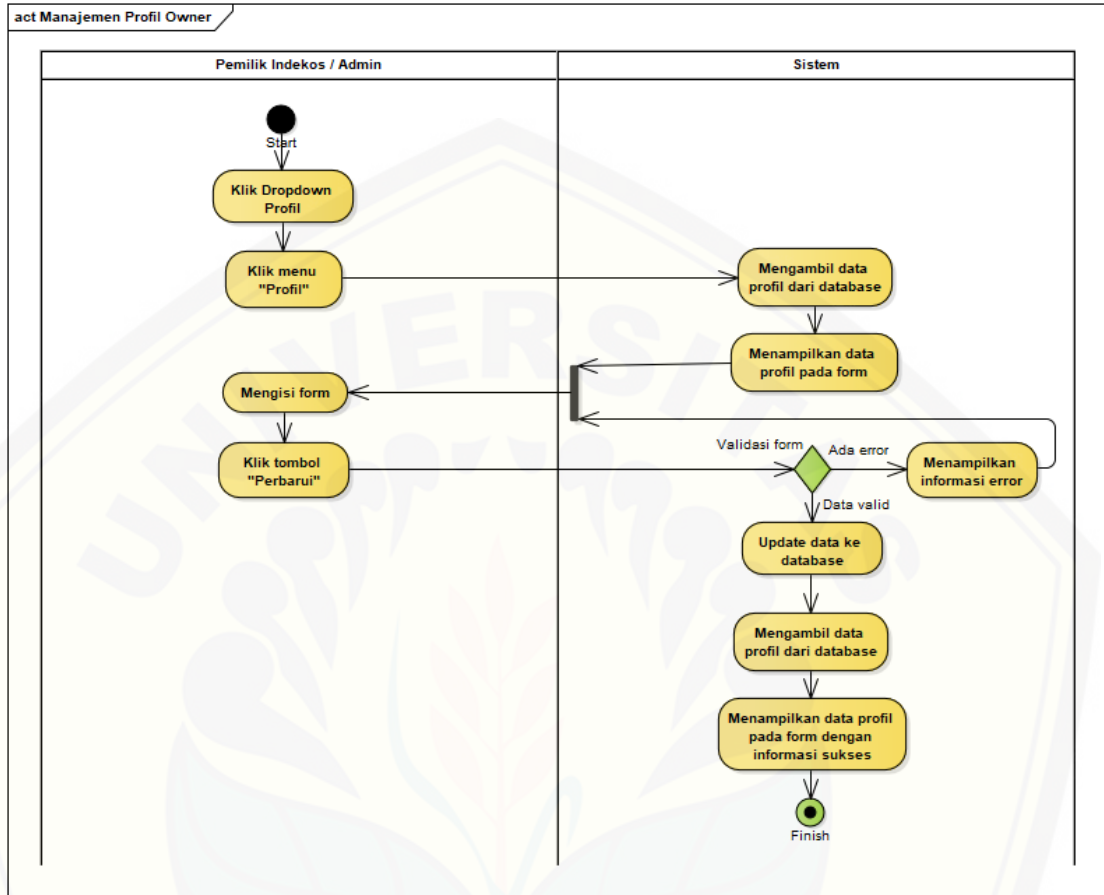
*Activity Diagram* Lihat Indekos dapat di lihat pada Gambar 7.



Gambar 7 *Activity Diagram* Lihat Indekos

B.8 *Activity Diagram Manajemen Profil Owner*

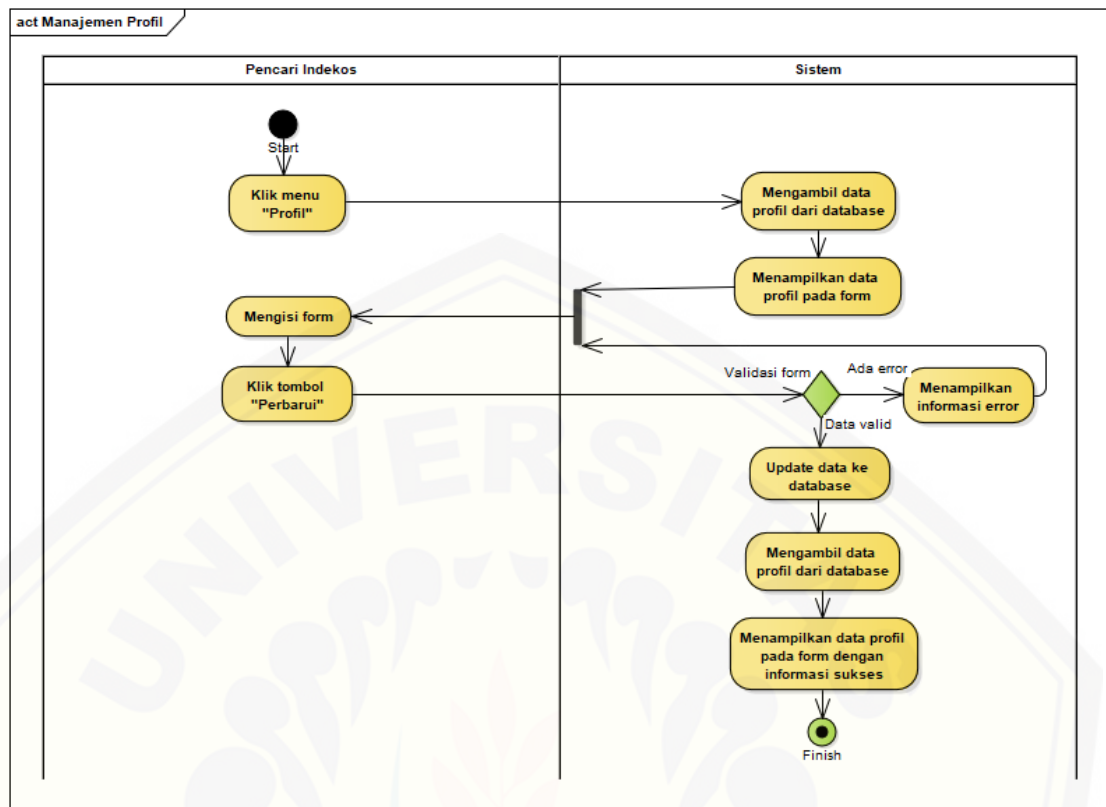
*Activity Diagram Manajemen Profil Owner* dapat di lihat pada Gambar 8.



Gambar 8 *Activity Diagram Manajemen Profil Owner*

B.9 *Activity Diagram Manajemen Profil*

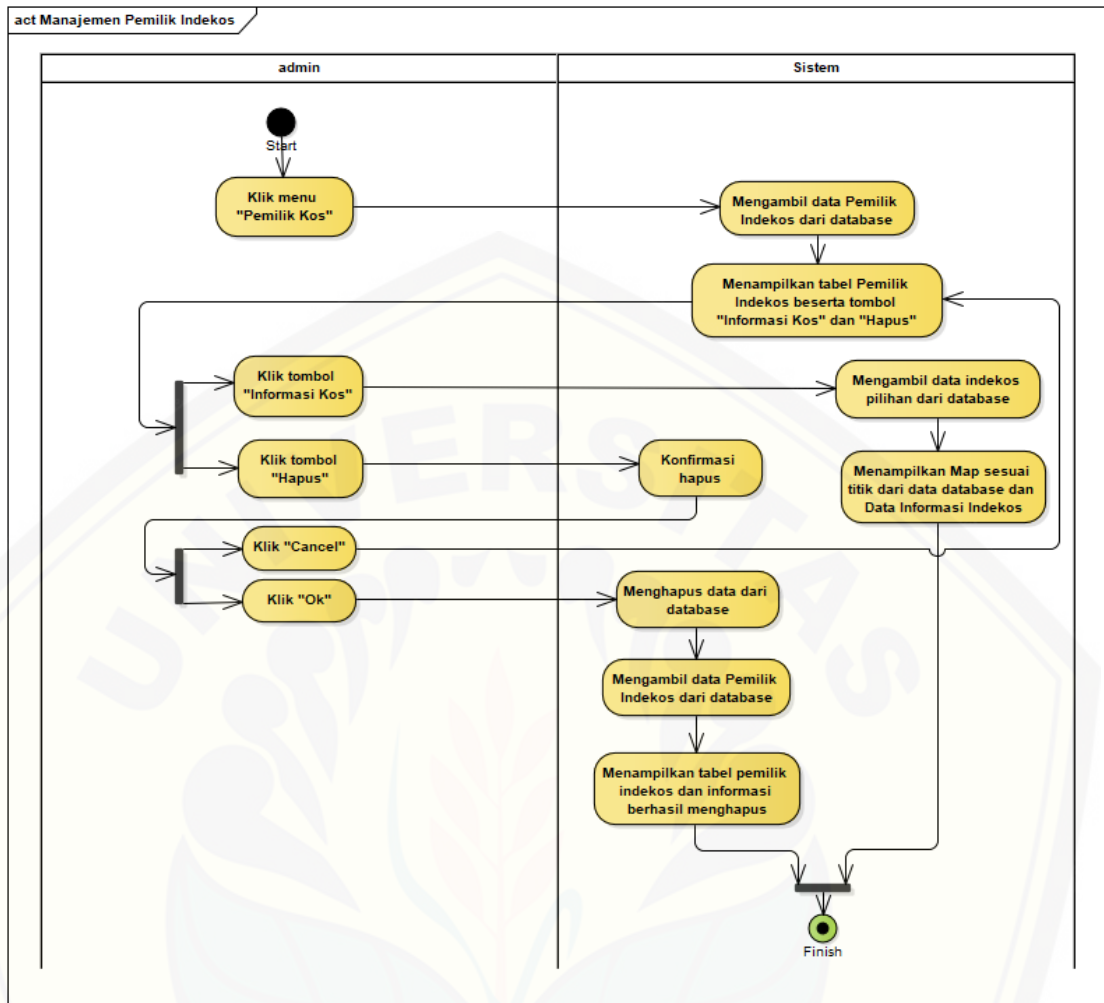
*Activity Diagram Manajemen Profil* dapat di lihat pada Gambar 9.



Gambar 9 Activity Diagram Manajemen Profil

B.10 Activity Diagram Manajemen Pemilik Indekos

Activity Diagram Manajemen Pemilik Indekos dapat di lihat pada Gambar 10.

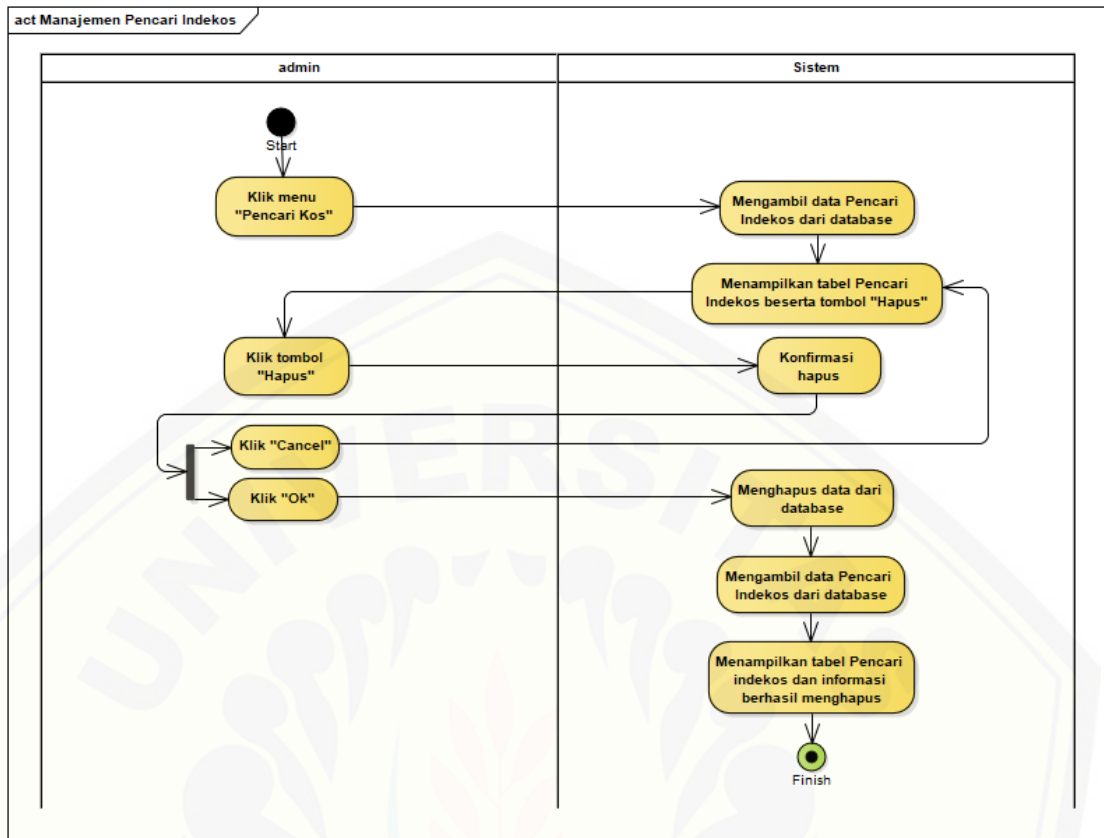


Gambar 10 Activity Diagram Manajemen Pemilik Indeks

B.11 Activity Diagram Manajemen Pencari Indeks

Activity Diagram Manajemen Pencari Indeks dapat di lihat pada Gambar 11.

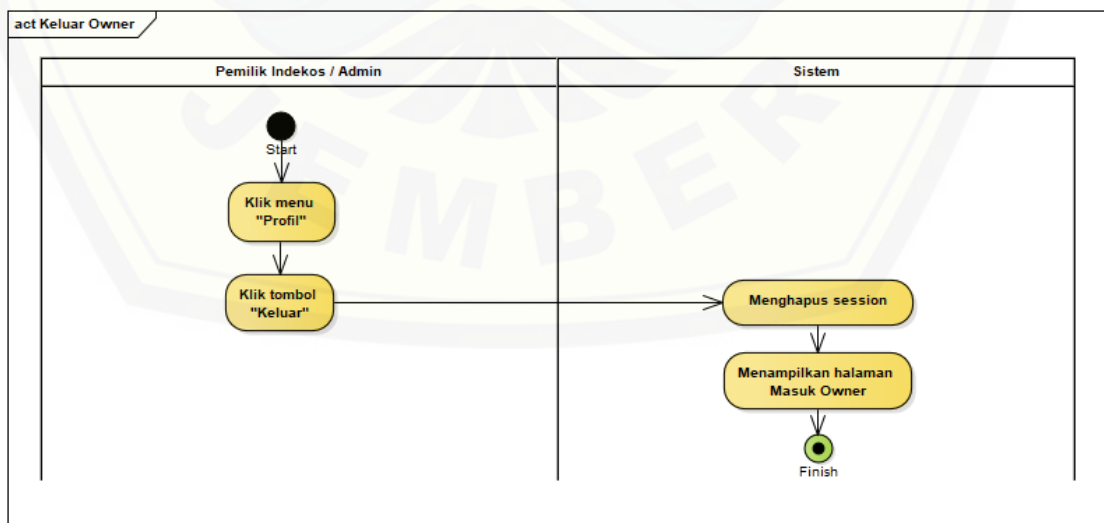




Gambar 11 Activity Diagram Manajemen Pencari Indeks

B.12 Activity Diagram Keluar Owner

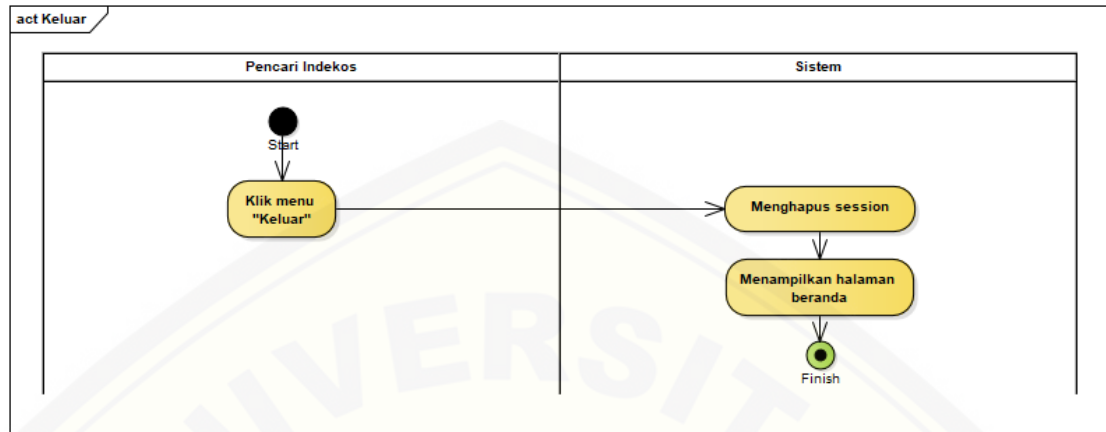
Activity Diagram Keluar Owner dapat di lihat pada Gambar 12.



Gambar 12 Activity Diagram Keluar Owner

B.13 *Activity Diagram* Keluar

*Activity Diagram* Keluar dapat di lihat pada Gambar 13.

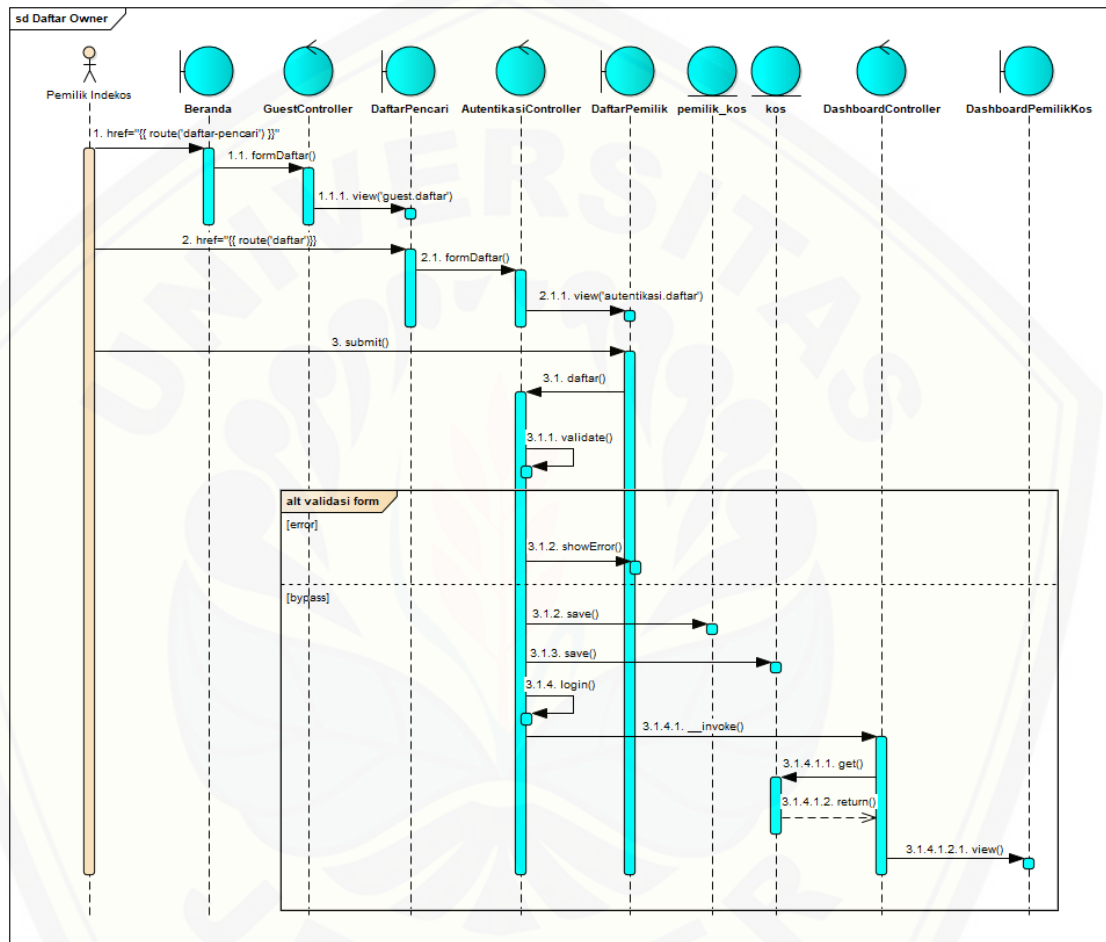


Gambar 13 *Activity Diagram* Keluar

C. *Sequence Diagram*

C.1 *Sequence Diagram Daftar Owner*

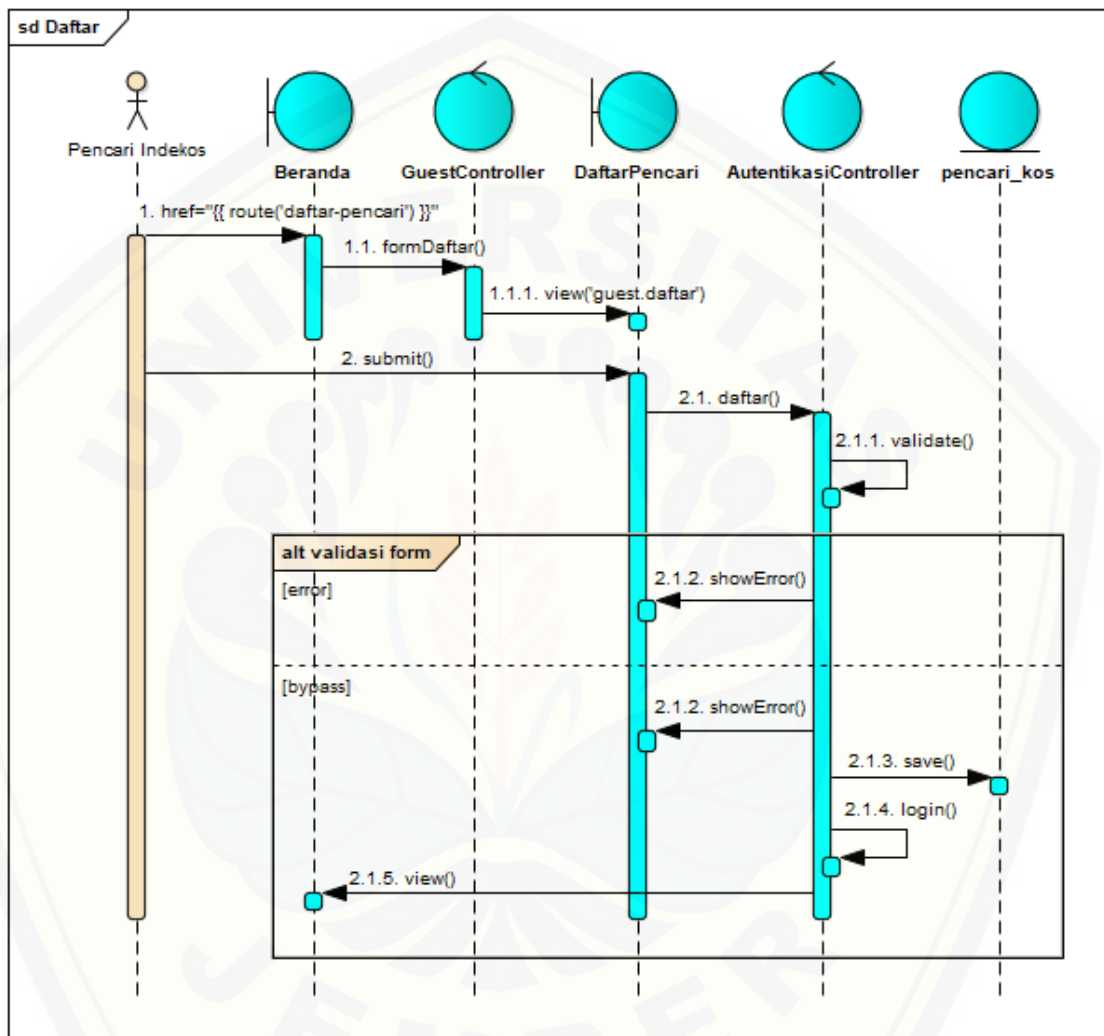
Penggambaran *sequence diagram* Daftar Owner digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada Gambar 14.



Gambar 14 *Sequence Diagram* Daftar Owner

C.2 Sequence Diagram Daftar

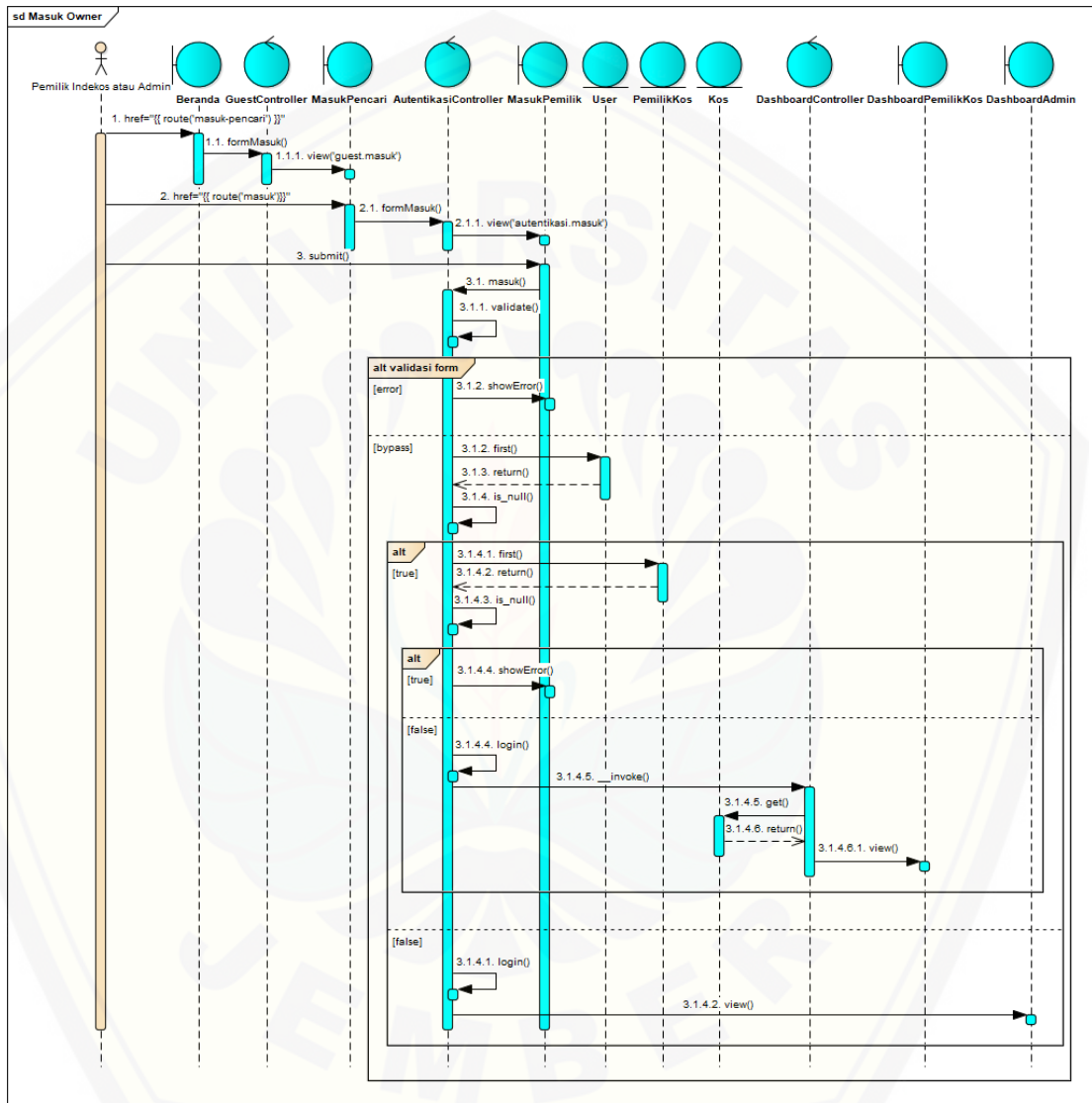
Penggambaran *sequence diagram* Daftar digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada Gambar 15.



Gambar 15 Sequence Diagram Daftar

C.3 Sequence Diagram Masuk Owner

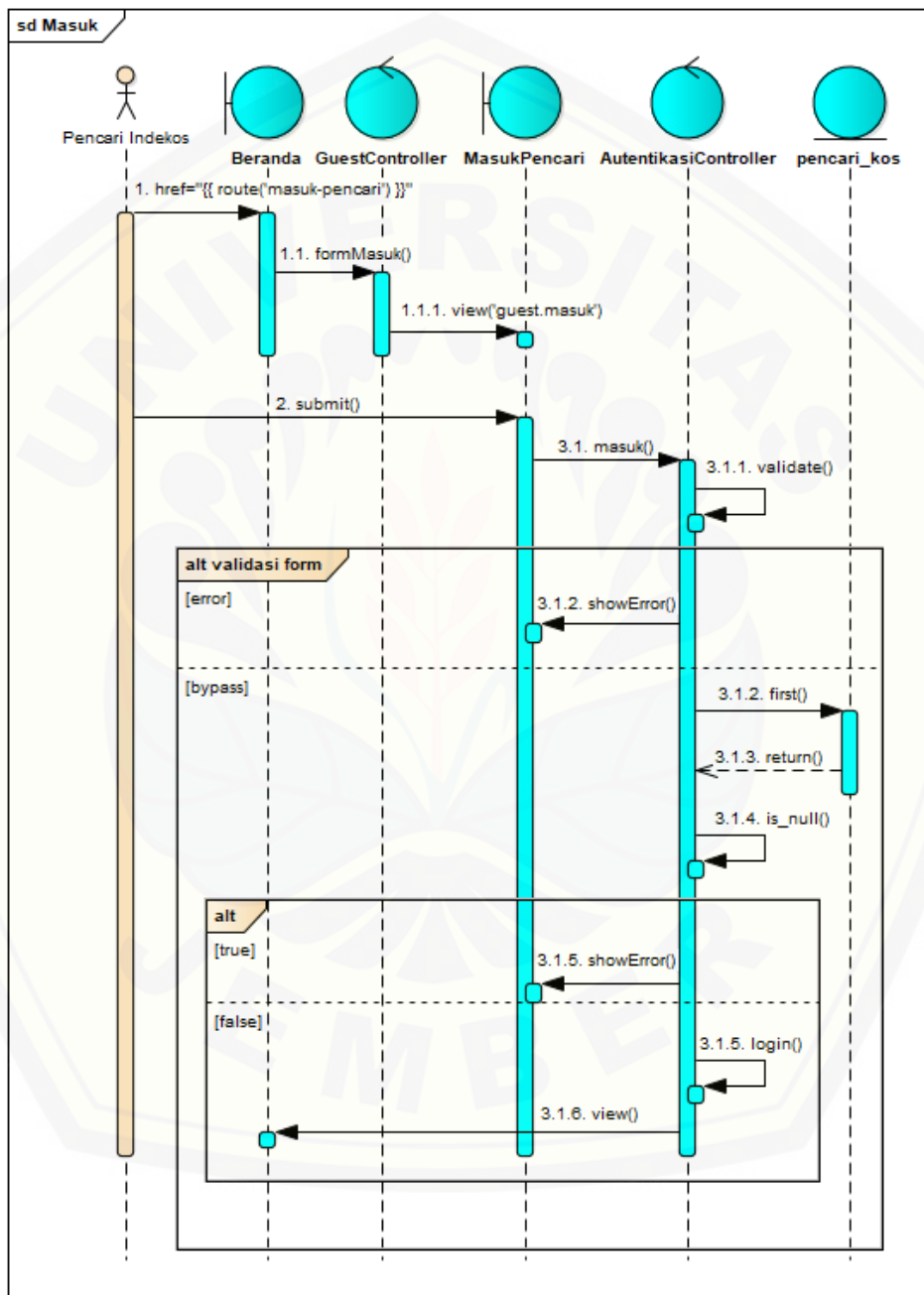
Penggambaran *sequence diagram* Masuk Owner digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada Gambar 16.



Gambar 16 Sequence Diagram Masuk Owner

C.4 Sequence Diagram Masuk

Penggambaran *sequence diagram* Masuk digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada Gambar 17.

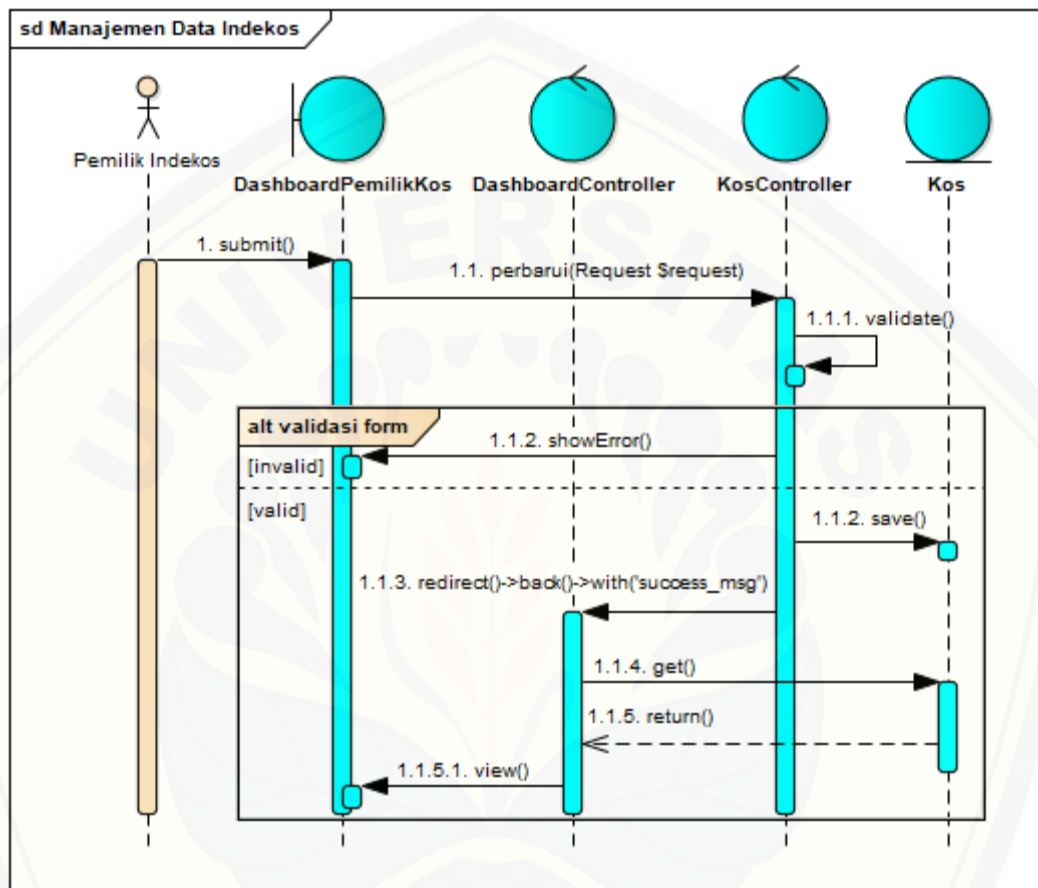


Gambar 17 Sequence Diagram Masuk



### C.5 Sequence Diagram Manejemen Data Indekos

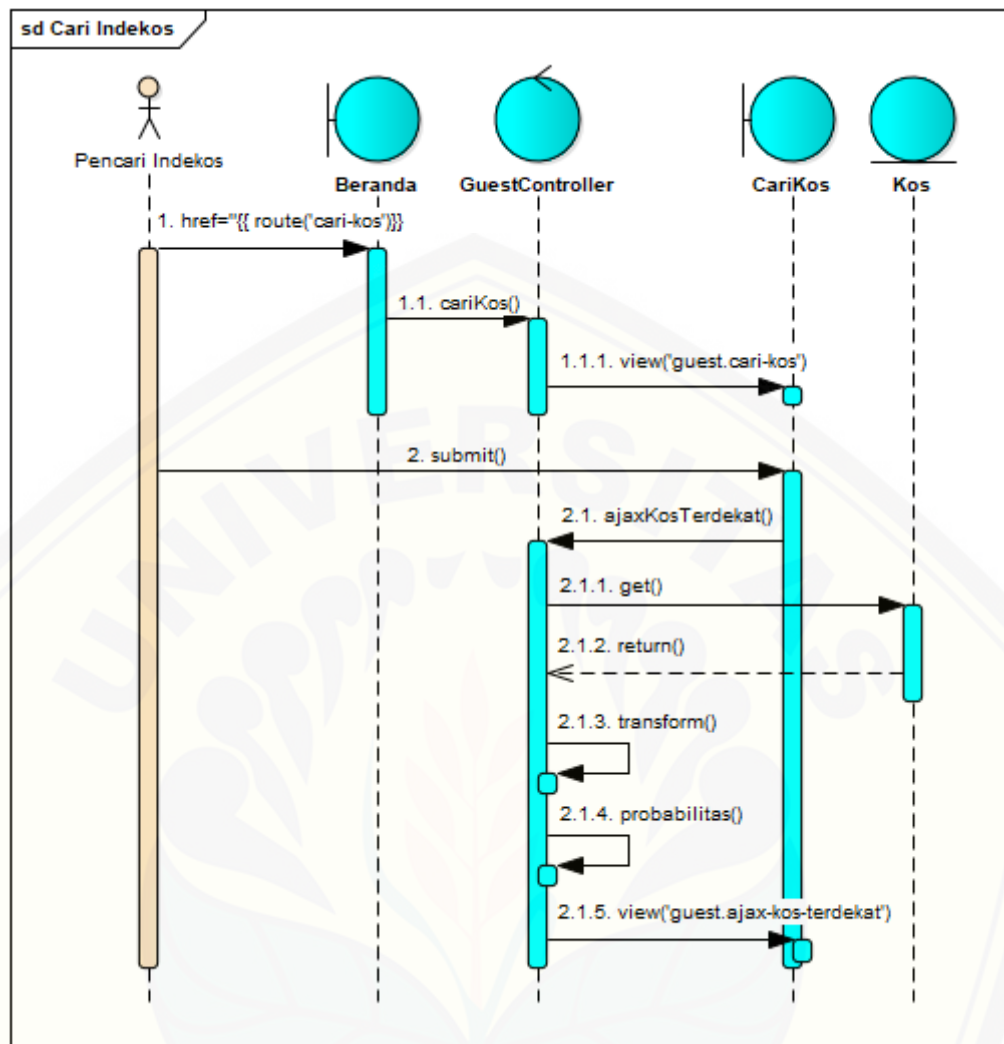
Penggambaran *sequence diagram* Manejemen Data Indekos digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada Gambar 18.



Gambar 18 Sequence Diagram Manejemen Data Indekos

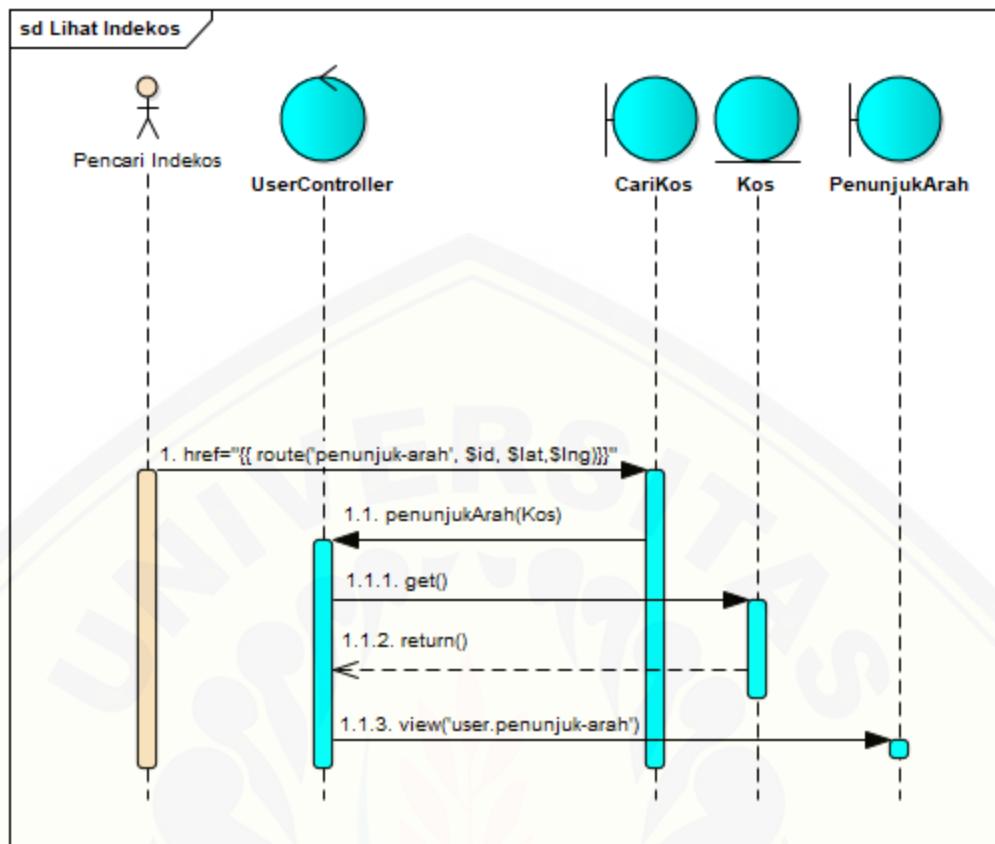
### C.6 Sequence Diagram Cari Indekos

Penggambaran *sequence diagram* Cari Indekos digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada Gambar 19.

Gambar 19 *Sequence Diagram* Cari Indekos

### C.7 *Sequence Diagram* Lihat Indekos

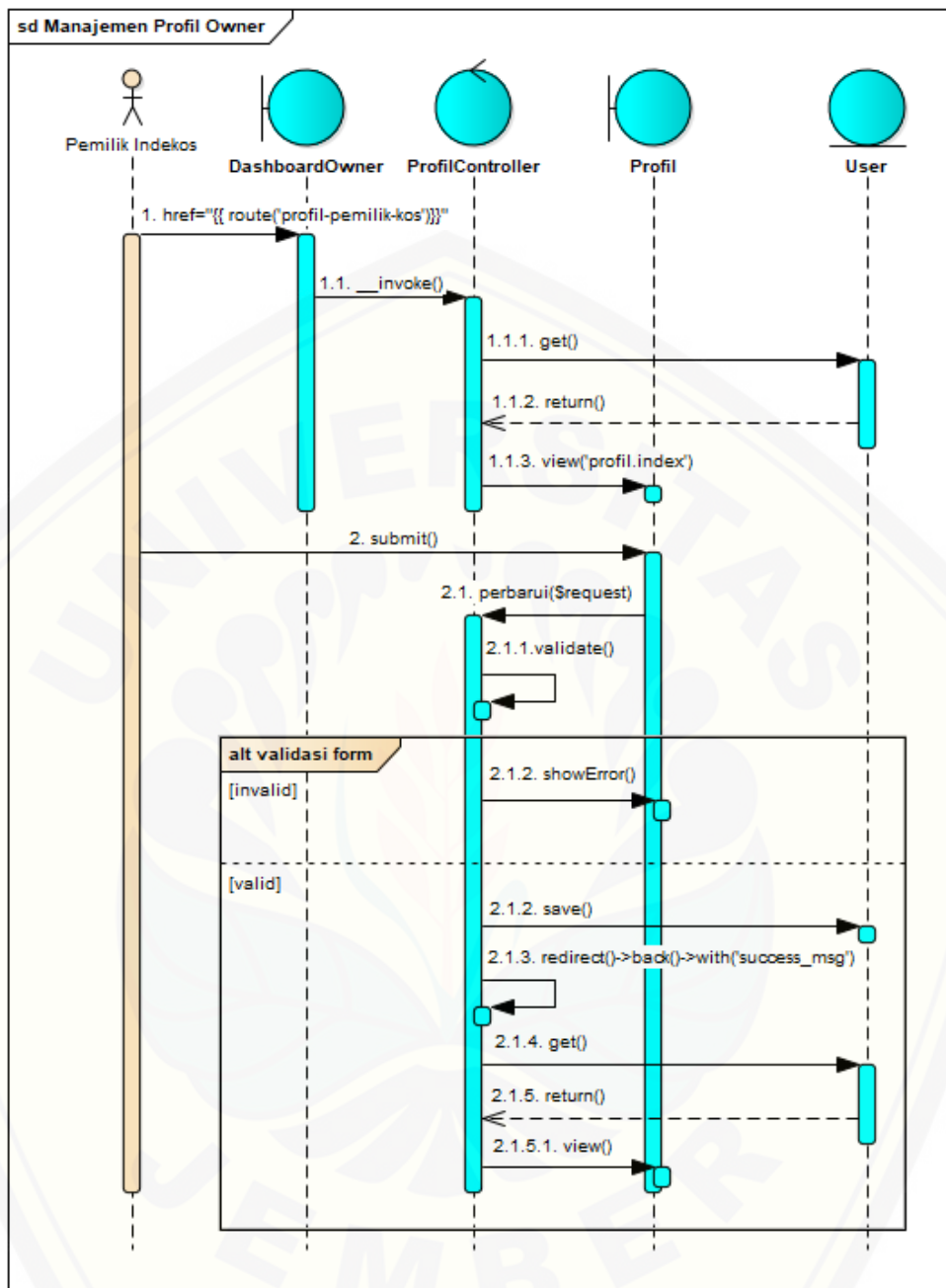
Penggambaran *sequence diagram* Lihat Indekos digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada Gambar 20.



Gambar 20 *Sequence Diagram* Lihat Indekos

#### C.8 *Sequence Diagram* Manajemen Profil Owner

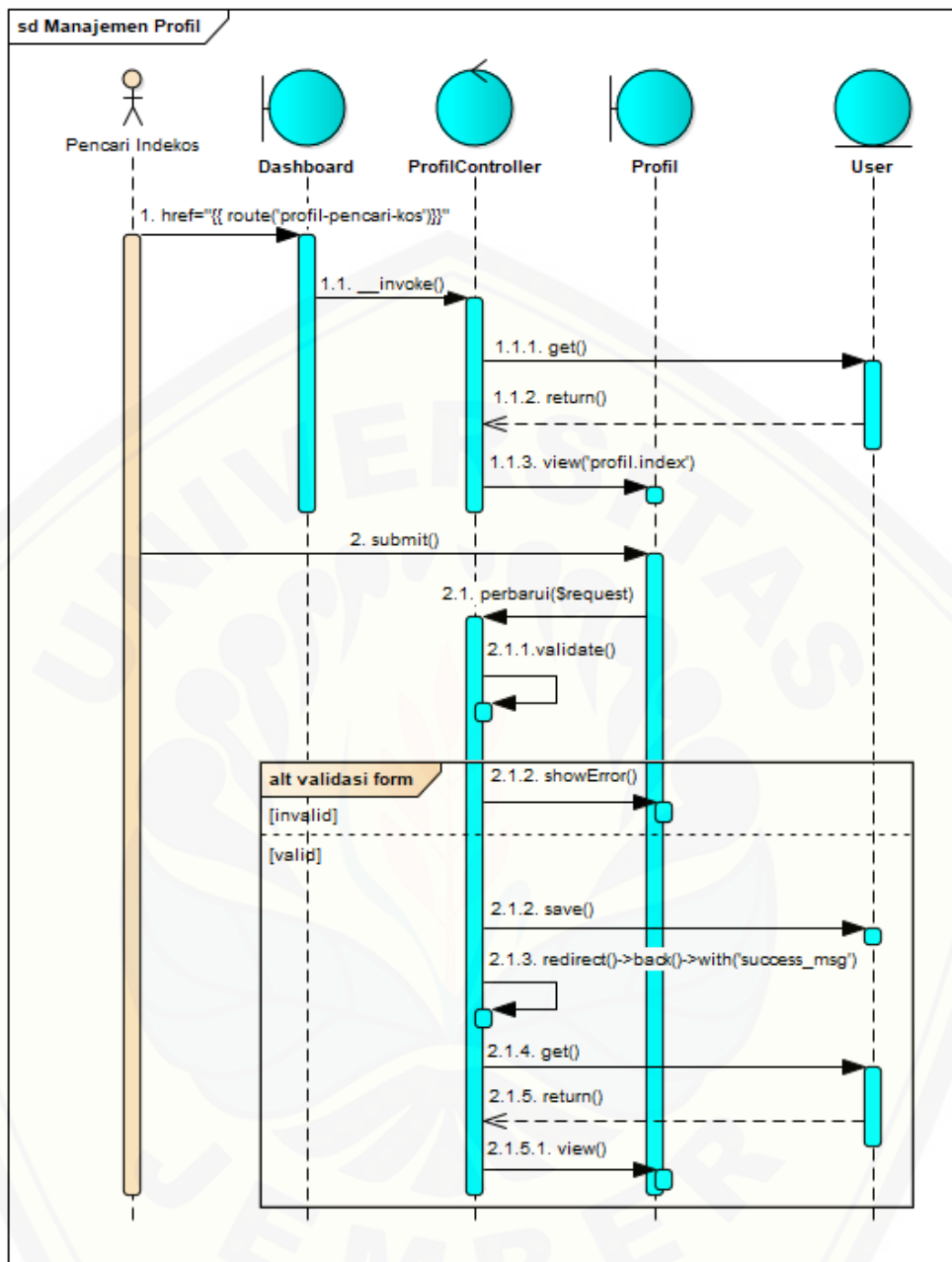
Penggambaran *sequence diagram* Manajemen Profil Owner digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada Gambar 21.



Gambar 21 *Sequence Diagram* Manajemen Profil Owner

C.9 *Sequence Diagram* Manajemen Profil

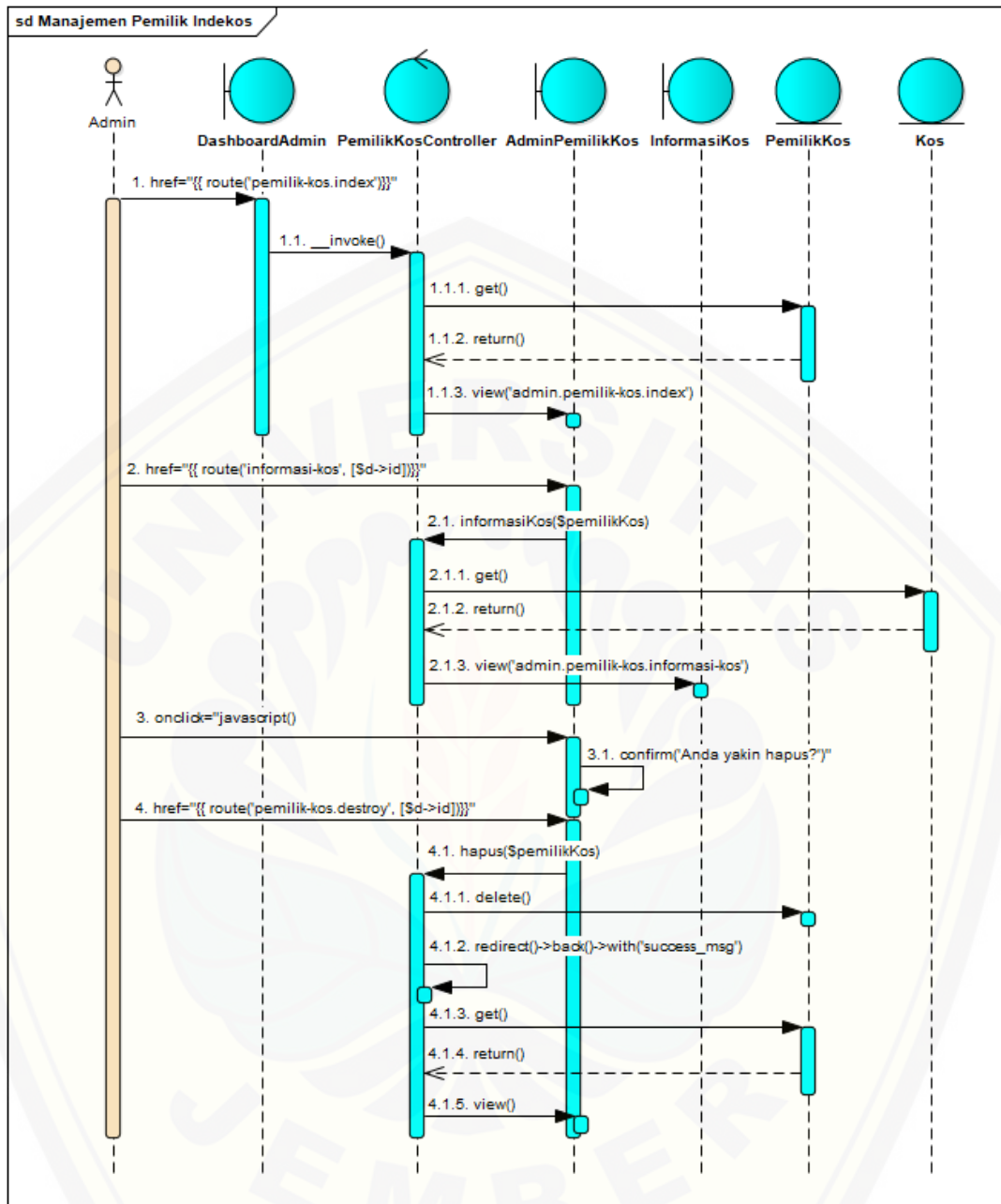
Penggambaran *sequence diagram* Manajemen Profil digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada Gambar 22.



Gambar 22 Sequence Diagram Manajemen Profil

C.10 Sequence Diagram Manajemen Pemilik Indekos

Penggambaran *sequence diagram* Manajemen Pemilik Indekos digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada Gambar 23.

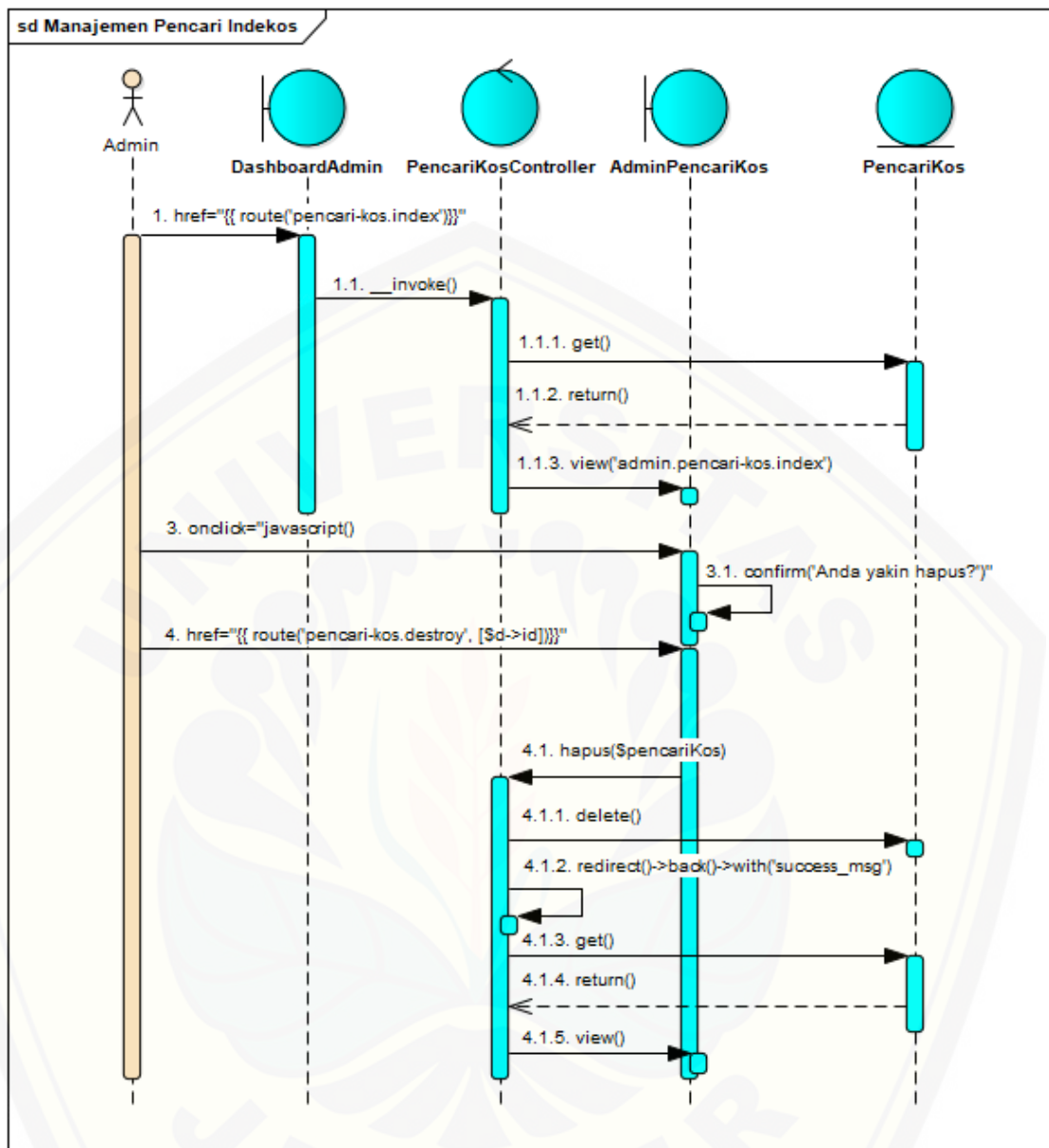


Gambar 23 Sequence Diagram Manajemen Pemilik Indekos

C.11 Sequence Diagram Manajemen Pencari Indekos

Penggambaran *sequence diagram* Manajemen Pencari Indekos digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada Gambar 24.

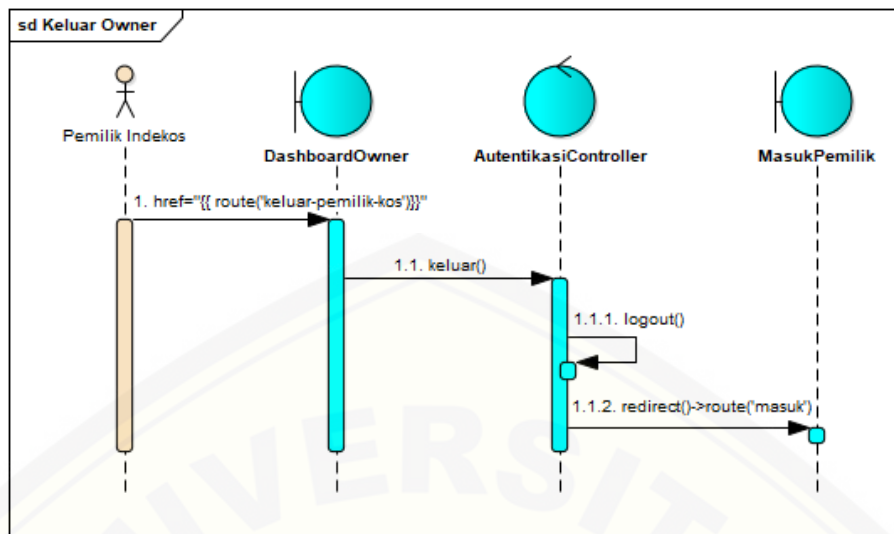




Gambar 24 *Sequence Diagram* Manajemen Pencari Indeks

C.12 *Sequence Diagram* Keluar Owner

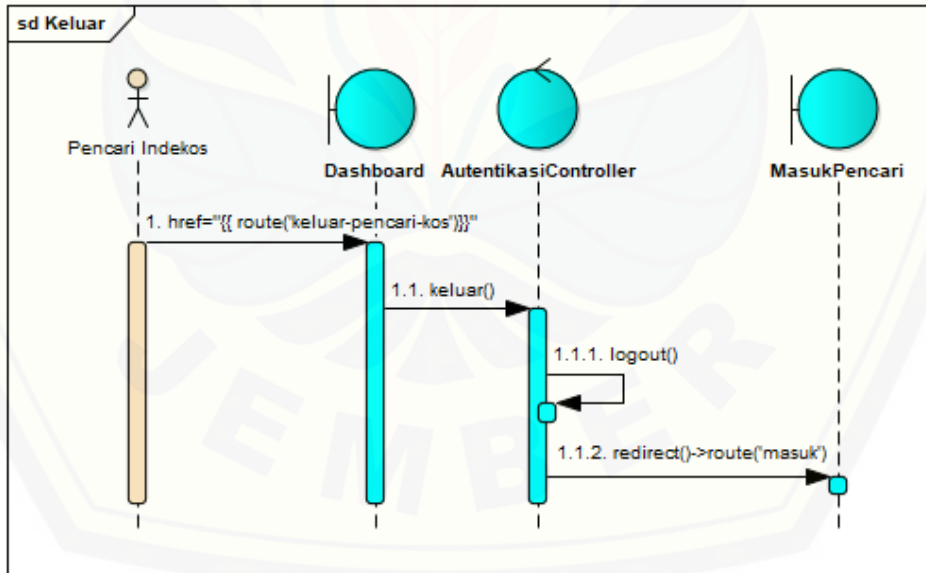
Penggambaran *sequence diagram* Keluar Owner digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada Gambar 25.



Gambar 25 Sequence Diagram Keluar Owner

C.13 Sequence Diagram Keluar

Penggambaran *sequence diagram* Keluar digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada Gambar 26.



Gambar 26 Sequence Diagram Keluar

**D. Pengujian Black Box****D.1 User Admin**

Pengujian sistem menggunakan *Black Box* untuk aktor admin dengan fitur Manajemen Pemilik Indekos dan Manajemen Pencari Indekos dapat dilihat pada Tabel 14.

Tabel 14 Pengujian *Black Box* Manajemen Pencari Indekos dan Pemilik Indekos

No.	Fitur	Aksi	Hasil	Kesimpulan
1.	Manajemen Pemilik Indekos	Memilih menu “Pemilik Kos”	Menampilkan tabel daftar pemilik indekos beserta tombol “informasi kos” dan tombol “hapus”.	Berhasil
		Memilih tombol “Hapus” di salah satu pemilik indekos	Menampilkan konfirmasi <i>alert</i> penghapusan data pemilik indekos.	Berhasil
		Memilih tombol “Cancel” pada alert	Menutup alert.	Berhasil
		Memilih tombol “Ok” pada alert	a. Menghapus data pemilik indekos beserta data indekos miliknya. b. Menampilkan tabel daftar pemilik indekos beserta tombol “informasi kos” dan tombol “hapus”.	Berhasil

		Memilih Tombol “Informasi Kos” di salah satu pemilik indekos	<ul style="list-style-type: none"> <li>a. Menampilkan detail pemilik indekos.</li> <li>b. Menampilkan Leaflet <i>Map</i> yang menunjukkan posisi indekos.</li> <li>c. Menampilkan detail indekos</li> </ul>	Berhasil
2.	Manajemen Pencari Indekos	Memilih menu “Pencari Kos”	Menampilkan tabel daftar pencari indekos beserta tombol “hapus”.	Berhasil
		Memilih tombol “Hapus” di salah satu pencari indekos	Menampilkan konfirmasi <i>alert</i> penghapusan data pencari indekos.	Berhasil
		Memilih tombol “Cancel” pada alert	Menutup alert.	Berhasil
		Memilih tombol “Ok” pada alert	<ul style="list-style-type: none"> <li>a. Menghapus data pencari indekos.</li> <li>b. Menampilkan tabel daftar pencari indekos beserta tombol “hapus”.</li> </ul>	Berhasil

## D.2 User Pemilik Indekos

Pengujian sistem menggunakan *Black Box* untuk aktor pemilik indekos dengan fitur manajemen data indekos dapat dilihat pada Tabel 15.

Tabel 15 Pengujian *Black Box* Manajemen Data Indekos

No.	Fitur	Aksi	Hasil	Kesimpulan
1.	Manajemen Data Indekos	Memindah titik lokasi indekos pada Leaflet <i>Maps</i> lalu klik tombol “Perbarui”	a. Menampilkan notifikasi sukses. b. Menampilkan Leaflet <i>Map</i> dengan titik lokasi indekos. c. Menampilkan informasi indekos pada form.	Berhasil
		Mengubah data informasi indekos pada form lalu klik tombol “Perbarui”	a. Menampilkan notifikasi sukses. b. Menampilkan Leaflet <i>Map</i> dengan titik lokasi indekos. c. Menampilkan informasi indekos pada form.	Berhasil

## D.3 User Pencari Indekos

Pengujian sistem menggunakan *Black Box* untuk aktor pencari indekos dengan fitur pencarian indekos dapat dilihat pada Tabel 16

Tabel 16 Pengujian *Black Box* Pencarian Indekos

No.	Fitur	Aksi	Hasil	Kesimpulan
1.	Pencarian Indekos	Memilih menu “Cari Kos”	a. Menampilkan Leaflet <i>Map</i> dengan titik posisi user saat mengakses aplikasi yang bias digeser.	Berhasil

			b. Menampilkan pilihan pencarian dalam bentuk <i>dropdown</i> .	
		Memilih Tombol “Cari Kos”	Menampilkan Tabel Indekos hasil pencarianurut berdasarkan Lokasi terdekat dan sebuah baris indekos dengan tanda Indekos Rekomendasi, serta tombol “Tunjukkan” di masing-masing indekos.	Berhasil
		Memilih tombol “Tunjukkan” di salah satu indekos	a. Menampilkan Leaflet <i>Map</i> yang menunjukkan posisi awal pilihan dan posisi indekos pilihan beserta dengan rute perjalanannya. b. Menampilkan detail indekos beserta nomor telepon indekos yang bisa ditekan untuk langsung menelepon.	Berhasil