



**SISTEM INFORMASI DAN REKOMENDASI PEMILIHAN *HOMESTAY*
BAGI WISATAWAN DI KAWASAN DESA WISATA MENGGUNAKAN
METODE *WEIGHTED PRODUCT* (STUDI KASUS : DESA WISATA
TANOKER KECAMATAN LEDOKOMBO KABUPATEN JEMBER)**

SKRIPSI

Oleh

Refanda Zulkarnain

NIM 112410101035

**PROGRAM STUDI SISTEM INFORMASI
UNIVERSITAS JEMBER**

2015



**SISTEM INFORMASI DAN REKOMENDASI PEMILIHAN
HOMESTAY BAGI WISATAWAN DI KAWASAN DESA WISATA
MENGUNAKAN METODE *WEIGHTED PRODUCT* (STUDI KASUS : DESA
WISATA TANOKER KECAMATAN LEDOKOMBO KABUPATEN
JEMBER)**

SKRIPSI

diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat
untuk menyelesaikan Program Studi Sistem Informasi (S1)
dan mencapai gelar Sarjana Sistem Informasi

Oleh :
Refanda Zulkarnain

112410101035

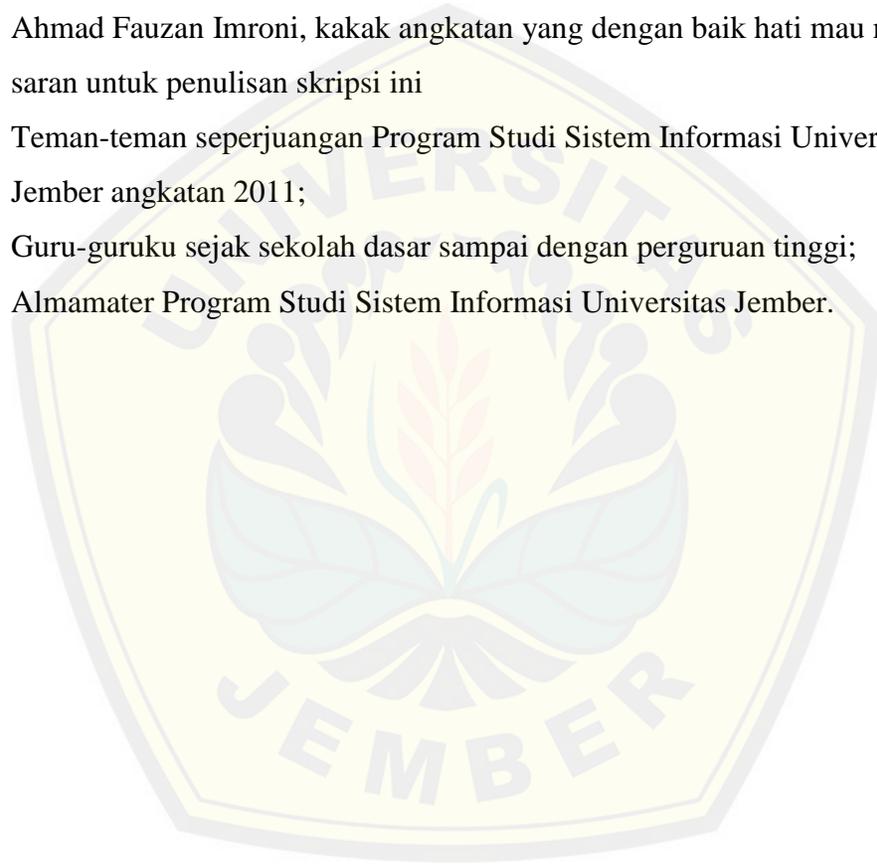
**PROGRAM STUDI SISTEM INFORMASI
UNIVERSITAS JEMBER**

2015

PERSEMBAHAN

Skripsi ini saya persembahkan untuk :

1. Ibunda Retno Ernayanti dan Ayahanda Fadjar Zulkarnain yang tercinta;
2. Kakakku tercinta Oryza Filial Zulkarnain, S.KM;
3. Ahmad Fauzan Imroni, kakak angkatan yang dengan baik hati mau memberi saran untuk penulisan skripsi ini
4. Teman-teman seperjuangan Program Studi Sistem Informasi Universitas Jember angkatan 2011;
5. Guru-guruku sejak sekolah dasar sampai dengan perguruan tinggi;
6. Almamater Program Studi Sistem Informasi Universitas Jember.



MOTTO



PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama: Refanda Zulkarnain

NIM : 112410101035

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Sistem Informasi dan Rekomendasi Pemilihan *Homestay* Bagi Wisatawan di Kawasan Desa Wisata Menggunakan Metode *Weighted Product*(WP) (Studi Kasus: Desa Wisata Tanoker Kecamatan Ledokombo Kabupaten Jember”, adalah benar-benar hasil karya sendiri, kecuali jika dalam pengutipan substansi disebutkan sumbernya, dan belum pernah diajukan pada institusi manapun, serta 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, 2015

Yang menyatakan,

Refanda Zulkarnain

NIM 112410101035

SKRIPSI

**SISTEM INFORMASI DAN REKOMENDASI PEMILIHAN *HOMESTAY* BAGI
WISATAWAN DI KAWASAN DESA WISATA MENGGUNAKAN METODE
WEIGHTED PRODUCT (STUDI KASUS : DESA WISATA TANOKER
KECAMATAN LEDOKOMBO KABUPATEN JEMBER)**

Oleh

Refanda Zulkarnain

NIM 112410101035

Pembimbing:

Pembimbing Utama : Prof. Drs. Slamini, M.Comp.Sc.,Ph.D.

NIP 19780330 200312 1 003

Pembimbing Pendamping : Muhamad Arief Hidayat, S.Kom, M.Kom.

NIP 198101232010121003

PENGESAHAN

Skripsi berjudul Sistem Informasi dan Rekomendasi Pemilihan *Homestay* Bagi Wisatawan Di Kawasan Desa Wisata Menggunakan Metode *Weighted Product* (WP) (Studi Kasus: Desa Wisata Tanoker Kecamatan Ledokombo Kabupaten Jember) telah diuji dan disahkan pada:

Hari, tanggal : Senin, 29 Juni 2015

Tempat : Program Studi Sistem Informasi Jember

Tim Penguji:

Penguji I,

Penguji II,

Anang Andrianto S.T., M.T
NIP19690615 199702 1 002

Windi Eka Yulia Retnani, S.Kom., MT.
NIP 198403052010122002

Mengesahkan
Ketua Program Studi,

Prof. Drs. Slamin, M.Comp. Sc., Ph.D.
NIP 196704201992011001

PENGESAHAN PEMBIMBING

Skripsi berjudul Sistem Informasi dan Rekomendasi Pemilihan *Homestay* Bagi Wisatawan Di Kawasan Desa Wisata Menggunakan Metode *Weighted Product* (WP) (Studi Kasus: Desa Wisata Tanoker Kecamatan Ledokombo Kabupaten Jember) telah diuji dan disahkan pada:

Hari, tanggal : Senin, 29 Juni 2015

Tempat : Program Studi Sistem Informasi Jember

Disetujui oleh:

Pembimbing Utama,

Pembimbing Pendamping,

Prof. Drs. Slamim, M.Comp.Sc.,Ph.D.

Muhamad Arief Hidayat, S.Kom, M.Kom.

NIP196704201992011001

NIP 198101232010121003

RINGKASAN

Sistem Informasi dan Rekomendasi Pemilihan *Homestay* Bagi Wisatawan Di Kawasan Desa Wisata Menggunakan Metode *Weighted Product* (WP) (Studi Kasus: Desa Wisata Tanoker Kecamatan Ledokombo Kabupaten Jember); Refanda Zulkarnain, 112410101035; 2015: 128 halaman; Program Studi Sistem Informasi Universitas Jember.

Berwisata adalah suatu aktivitas berpergian menuju berbagai daerah tertentu atau yang biasa disebut *traveling* dengan membawa barang dan akomodasi yang diperlukan. Wisatawan yang meningkat dengan cepat sayangnya tidak diikuti dengan kemudahan para wisatawan dalam mencari tempat menginap yang sesuai. Sedangkan orang yang ingin menawarkan tempat tinggalnya tidak menemukan media yang tepat untuk menginformasikan tempat tinggalnya. Penelitian ini bermaksud untuk memberikan fasilitas kemudahan informasi bagi kedua belah pihak melalui media yang mudah diakses. Media tersebut adalah *website* yang dapat diakses dengan mudah dimana saja dengan hanya menggunakan koneksi internet. Selain itu, untuk memudahkan calon pengunjung dalam menentukan *homestay* yang sesuai dengan kriteria yang diinginkan, penulis membuat sistem pendukung keputusan dengan metode seleksi pengambilan keputusan *Weighted Product* (WP) dengan bahasa pemrograman PHP dan MySQL sebagai database. Dengan sistem pendukung keputusan diharapkan dapat membantu para pengunjung dalam memilih *homestay* yang sesuai dengan kriteria yang diinginkan. Teknik penyimpanan dilakukan menggunakan media penyimpanan secara terpusat yaitu server sehingga perputaran informasi secara dinamis dapat terjadi. Dari hasil penelitian yang dilakukan, aplikasi dapat menjadi sebuah solusi sebagai media penghubung antara wisatawan dan orang yang menawarkan ruangan tempat tinggalnya.

DAFTAR ISI

	Halaman
SKRIPSI.....	ii
PERNYATAAN	v
PENGESAHAN.....	vii
PENGESAHAN PEMBIMBING	viii
RINGKASAN	ix
DAFTAR ISI.....	x
DAFTAR TABEL.....	Error! Bookmark not defined.
DAFTAR GAMBAR.....	Error! Bookmark not defined.
DAFTAR LAMPIRAN.....	13
BAB 1. PENDAHULUAN	14
Latar Belakang	14
Rumusan Masalah.....	16
Tujuan dan Manfaat	16
Batasan Masalah	17
Sistematika Penulisan	18
BAB 2. TINJAUAN PUSTAKA	19
2.1 Penelitian Terdahulu	19
2.2 Desa Wisata	20
2.3 Pengertian Sistem Pendukung Keputusan.....	21
2.4 Tujuan Sistem Pendukung Keputusan	21
2.5 Langkah-langkah Pemodelan Dalam Sistem Pendukung Keputusan	22
2.6 Karakteristik Sistem Pendukung Keputusan.....	23
2.7 Komponen Sistem Pendukung Keputusan.....	24
2.7.1 Subsistem Manajemen Data.....	24
2.7.2 Subsistem Manajemen Model.....	25
2.7.3 Subsistem Antarmuka Pengguna	25
2.7.4 Subsistem Basis Pengetahuan	26
2.8 Metode Pengambilan Keputusan	26
2.9 Metode <i>Weighted Product</i> (WP).....	27

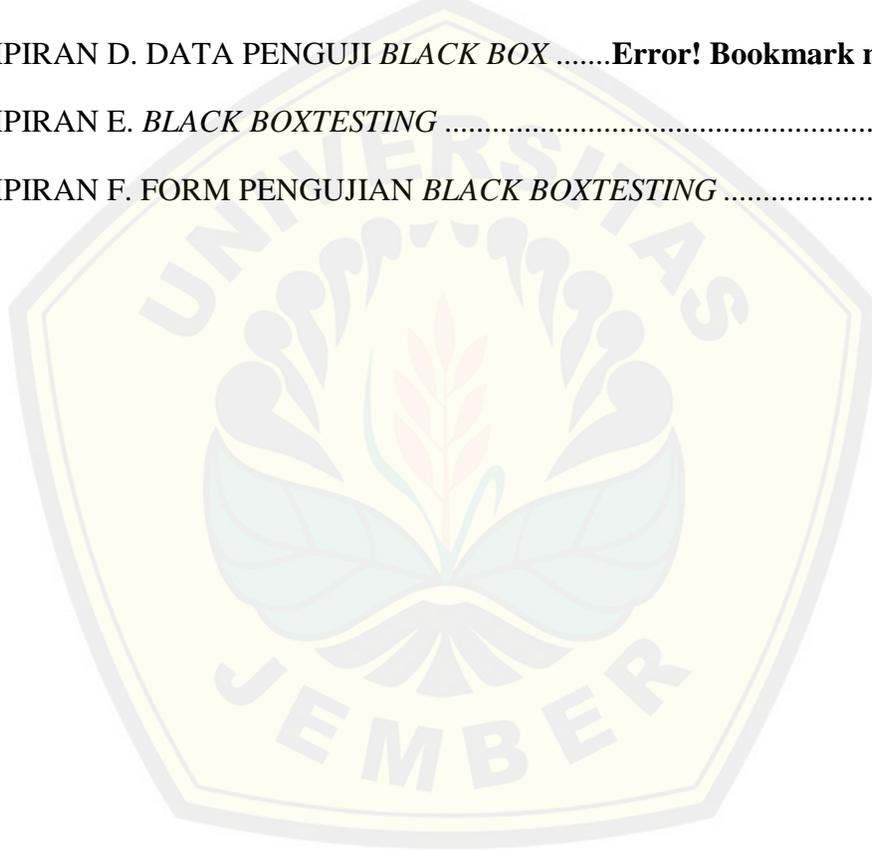
Jenis Penelitian.....	30
Objek Penelitian.....	30
Metode Penelitian	30
3.4 Metode Pengumpulan Data.....	30
3.5 Metode Analisis	31
3.6 Tahap Pengembangan Sistem	34
5.1.3 Fitur <i>Login</i>	69
5.1.4 Fitur Registrasi.....	72
5.1.5 Fitur Manajemen <i>User Aplikasi</i>	73
5.1.6 Fitur Manajemen Pemilik <i>Homestay</i>	75
5.1.7 Fitur Manajemen Booking <i>Homestay</i>	76
5.1.8 Fitur Manajemen Daftar <i>Homestay</i>	78
5.2 Hasil Pengembangan WP.....	81
LAMPIRAN B. ACTIVITY DIAGRAM	120
LAMPIRAN D. KODE PROGRAM.....	134
LAMPIRAN E. DATA PENGUJI <i>BLACK BOX</i>	143
LAMPIRAN F. <i>BLACK BOX TESTING</i>	144
LAMPIRAN G. FORM PENGUJIAN <i>BLACK BOX TESTING</i>	148



DAFTAR LAMPIRAN

Halaman

LAMPIRAN A. DATA LOKASI REKLAME INSIDENTIL KELAS JALAN SATU KAB. JEMBER.....	Error! Bookmark not defined.
LAMPIRAN B. <i>USE CASE</i> SKENARIO	106
LAMPIRAN C. KODE PROGRAM	Error! Bookmark not defined.
LAMPIRAN D. DATA PENGUJI <i>BLACK BOX</i>	Error! Bookmark not defined.
LAMPIRAN E. <i>BLACK BOXTESTING</i>	144
LAMPIRAN F. FORM PENGUJIAN <i>BLACK BOXTESTING</i>	148



BAB 1. PENDAHULUAN

Latar Belakang

Seiring dengan kemajuan zaman, kemajuan teknologi informasi juga semakin hari semakin berkembang dengan pesat. Perkembangan teknologi informasi ini, berdampak besar pada berbagai bidang kehidupan masyarakat baik dari segi sosial, ekonomi, pendidikan, maupun pariwisata. Oleh karena itu, sudah seharusnya suatu fasilitas umum memiliki sebuah sistem informasi yang baik. Dengan adanya sistem informasi yang baik maka akan memberikan kemudahan dalam mengakses berbagai hal yang berhubungan dengan perusahaan tersebut.

Saat ini, Kabupaten Jember merupakan salah satu kabupaten yang telah dilengkapi dengan berbagai fasilitas rekreasi, sarana olahraga dan wisata yang sangat menarik untuk dikunjungi. Kabupaten Jember sendiri memiliki potensi alam untuk dikembangkan menjadi kota pariwisata. Salah satu tempat wisata yang memiliki daya tarik bagi wisatawan domestik maupun mancanegara yaitu desa wisata. Desa wisata dengan segala keindahan potensi alam dan kearifan lokal yang masih terjaga menjadikan suatu pesona tersendiri bagi para wisatawan. Beragam pertunjukan seni dan budaya dapat menjadi paket wisata komplit yang sangat menarik. *Homestay* merupakan salah satu tempat yang dibutuhkan sebagai fasilitas penginapan di wilayah desa wisata. Namun sayangnya, *homestay* di Desa Wisata yang ada di Kabupaten Jember tidak terdata secara lengkap dan *up to date*. Hal ini tentu saja akan menyebabkan kesulitan bagi wisatawan untuk memilih *homestay* yang sesuai dengan keinginannya.

Desa wisata yang ada di Kabupaten Jember yaitu Desa Wisata Tanoker yang terletak di Kecamatan Ledokombo. Pihak dari Desa Wisata Tanoker juga masih melakukan pendataan reservasi *homestay* secara manual dengan mencatat dan mendatangi satu persatu rumah penduduk yang dapat digunakan sebagai *homestay*. Sehingga selain mengalami kesulitan dalam memperoleh informasi

mengenai *homestay* di Desa Wisata Tanoker yang ada di Kabupaten Jember, calon pengunjung juga akan mengalami kesulitan dalam memilih *homestay* yang sesuai dengan kriteria yang diinginkan. Oleh karena itu, data mengenai *homestay* di Desa Wisata Tanoker tersebut perlu dilengkapi. Salah satunya dengan memberikan fasilitas sistem informasi tersendiri di website resmi pihak Desa Wisata Tanoker. Sehingga setiap *homestay* dapat melakukan *input* dan update kelengkapan data-data *homestay* mereka ke dalam sebuah form yang telah disediakan. Selain data-data *homestay* akan menjadi lebih lengkap, data-data tersebut juga akan menjadi tersimpan ke dalam database dengan rapi dan teratur. Dengan demikian, calon pengunjung dapat memperoleh informasi yang lengkap mengenai *homestay-homestay* yang ada di Desa Wisata Tanoker. Dan untuk mempermudah calon pengunjung dalam menentukan *homestay* yang sesuai dengan kriteria yang diinginkan, maka dibutuhkan suatu sistem pendukung keputusan pemilihan *homestay*.

Sistem Pendukung Keputusan (SPK) digunakan sebagai alat bantu bagi para pengambil keputusan untuk memperluas kapabilitas para pengambil keputusan, namun tidak untuk menggantikan penilaian para pengambil keputusan (Turban, Aronso, & Liang, 2005). Dengan adanya sistem pendukung keputusan pemilihan *homestay* di Desa Wisata Tanoker dapat membantu para calon pengunjung dalam melakukan proses pemilihan *homestay* dengan cepat dan tepat, serta mampu memberikan rekomendasi keputusan *homestay* terpilih secara lebih objektif. Dengan adanya sistem tersebut diharapkan *homestay* yang terpilih benar-benar sesuai dengan yang diinginkan oleh calon pengunjung.

Dalam penelitian ini, penulis menggunakan metode *Weighted Product* (WP). Metode yang dipilih adalah metode *Wighted Product* karena pemilihan *homestay* digolongkan ke dalam masalah yang bersifat multicriteria (ada banyak kriteria untuk mencapai tujuan). Melalui metode *Weighted Product* (WP) dapat menentukan sendiri bobot kepentingan dari masing-masing kriteria. Metode *Weighted Product* (WP) ini menentukan nilai bobot untuk setiap atribut,

kemudian dilanjutkan dengan proses perankingan yang akan menyeleksi alternative terbaik dari sejumlah alternatif, dalam hal ini alternatif yang dimaksud adalah *homestay* yang memiliki kriteria sesuai dengan yang diinginkan calon pengunjung. Dengan metode perankingan tersebut, diharapkan penilaian akan lebih tepat karena didasarkan pada nilai kriteria dan bobot yang sudah ditentukan sehingga akan mendapatkan hasil yang lebih akurat dan optimal terhadap *homestay* terpilih yang akan dipertimbangkan oleh pengambil keputusan. Berdasarkan latar belakang diatas maka dalam tugas akhir ini penulis mengambil judul "Sistem Informasi Dan Rekomendasi Pemilihan *Homestay* bagi Wisatawan di Kawasan Desa Wisata Menggunakan Metode *Weighted Product* (Studi Kasus : Desa Wisata Tanoker Kecamatan Ledokombo Kabupaten Jember)".

Rumusan Masalah

Berdasarkan latar belakang masalah di atas maka rumusan masalah dalam penelitian ini adalah :

- a. Bagaimana membangun sistem yang dapat memberi informasi tentang *homestay* di Desa Wisata Tanoker Kabupaten Jember
- b. Bagaimana menghasilkan rekomendasi untuk membantu wisatawan dalam menentukan pilihan *homestay* sesuai keinginan menggunakan metode WP (*Weighted Product*)

Tujuan dan Manfaat

Tujuan dan manfaat berisi tentang tujuan dari pembuatan sistem informasi dan rekomendasi pemilihan *homestay* untuk wisatawan di Desa Wisata Tanoker Kabupaten Jember. Sedangkan pada bagian manfaat berisi tentang manfaat apa yang akan diperoleh pada pembuatan sistem informasi ini, baik bagi peneliti sendiri maupun bagi objek pada penelitian ini.

1.1.1 Tujuan

Adapun tujuan yang ingin dicapai dalam penelitian ini adalah sebagai berikut :

- a. Membangun sistem yang dapat memberikan informasi tentang *homestay* di Desa Wisata Tanoker
- b. Menghasilkan sistem yang dapat memberikan rekomendasi, sehingga mempermudah wisatawan dalam menentukan *homestay* sesuai keinginan dengan menggunakan metode WP (*Weighted Product*).

1.1.2 Manfaat

Manfaat penelitian ini adalah :

a. Manfaat Akademis

Hasil penelitian ini diharapkan dapat memberikan kontribusi dan masukan bagi siapa saja yang membutuhkan informasi yang berhubungan dengan judul penelitian ini.

b. Manfaat bagi peniliti

- 1) Mengetahui bagaimana proses penerapan metode *Weighted Product* pada aplikasi rekomendasi pemilihan *homestay*.
- 2) Membantu pengembangan fasilitas pariwisata di Kabupaten Jember khususnya wisatawan agar lebih mudah dalam memilih *homestay* yang sesuai keinginan.

c. Manfaat bagi objek penelitian

- 1) Membantu pihak Desa Wisata Tanoker dalam pendataan *homestay*.
- 2) Memudahkan wisatawan dalam memilih *homestay* sesuai keinginan.

Batasan Masalah

Agar permasalahan yang dirumuskan dapat lebih terfokus, maka batasan masalah dalam penelitian ini antara lain:

- a. Metode yang digunakan adalah metode *Weighted Product* (WP) untuk penerapan perhitungan.
- b. Penyelesaian hanya membahas mengenai teknis perhitungan pemilihan *homestay* bagi wisatawan.

- c. *User Admin* pada sistem ini tidak dapat menambahkan kriteria atau atribut.

Sistematika Penulisan

Sistematika penulisan ini merupakan gambaran umum mengenai isi dari keseluruhan pembahasan, yang bertujuan untuk memudahkan pembaca dalam mengikuti alur pembahasan yang terdapat dalam penulisan skripsi ini. Adapun sistematika penulisan adalah sebagai berikut :

- a. Pendahuluan

Bab ini terdiri atas latar belakang, rumusan masalah, tujuan dan manfaat, batasan masalah dan sistematika penulisan.

- b. Tinjauan Pustaka

Bab ini berisi tentang kajian materi, penelitian terdahulu dan informasi apa saja yang digunakan dalam penelitian ini. Dimulai dari kajian pustaka mengenai pengertian dari sistem informasi hingga metode *Weighted Product*.

- c. Metodologi Penelitian

Bab ini menguraikan tentang metode apa yang dilakukan selama penelitian. Dimulai dari tahap pencarian permasalahan hingga pengujian aplikasi yang akan dibuat.

- d. Hasil dan Pembahasan

Bab ini menjelaskan tentang hasil dan pembahasan dari penelitian yang telah dilakukan. Dengan menggambarkan dampak apa yang terjadi pada saat sebelum penggunaan sistem dan sesudah penggunaan sistem.

- e. Penutup

Bab ini berisi kesimpulan dari penelitian yang telah dilakukan dan saran untuk penelitian selanjutnya.

BAB 2. TINJAUAN PUSTAKA

Dalam penelitian pengimplementasian *Weighted Product* dalam sistem informasi dan rekomendasi pemilihan *homestay* bagi wisatawan di kawasan desa wisata, dibutuhkan beberapa landasan teori yang digunakan untuk memperkuat dan mengarahkan penelitian agar tidak keluar dari kaidah keilmuan yang ada.

2.1 Penelitian Terdahulu

Pada sebuah penelitian, metode *Weighted Product* WP digunakan untuk melakukan perankingan calon penerima beasiswa Peningkatan Prestasi Akademik (Ningrum, 2011). Penelitian ini menjelaskan proses perankingan calon penerima beasiswa PPA tahun 2011 di FMIPA UNS dengan menggunakan metode *Weighted Product*. *Weighted Product* adalah suatu metode yang digunakan untuk mencari alternatif optimal dengan kriteria tertentu. Inti dari metode *Weighted Product* adalah menentukan nilai bobot untuk setiap kriteria, kemudian dilanjutkan dengan proses perankingan yang akan menyeleksi alternatif yang sudah diberikan. Hasil perankingan diperoleh dengan mengurutkan nilai preferensi dari yang terbesar sampai yang terkecil. Kriteria yang digunakan pada penelitian ini antara lain nilai rata-rata rapor SMA kelas 12, penghasilan orang tua, jumlah tanggungan keluarga, dan pekerjaan orang tua.

Pada penelitian lainnya metode WP digunakan untuk mendukung keputusan pemberian bonus pegawai pada Hotel Alamanda di Klaten (Yuke Permatasari, 2013). Penelitian ini membahas tentang sistem pendukung keputusan pemberian bonus pada pegawai untuk setiap periodenya. Sistem penentuan pemberian bonus pegawai pada Hotel Alamanda masih dilakukan secara manual dengan mengecek satu-persatu kriteria yang dijadikan pedoman dasar dalam pengambilan keputusan siapakah pegawai yang memperoleh bonus pada suatu periode sehingga membuat banyak waktu yang tersita. Maka dari itu dibutuhkanlah suatu sistem yang terkomputerisasi

agar dapat mengatasi dan meringankan tugas manajer yaitu dengan sistem pendukung keputusan menggunakan metode *Weighted Product*. Data yang dikelola oleh sistem terkomputerisasi diharapkan dapat menyajikan informasi yang cepat, tepat, dan jelas.

Dari dua penelitian di atas dapat menjadi referensi pada penelitian saya, namun agar penggunaan metode *Weighted Product* (WP) bisa lebih tepat untuk kasus yang saya teliti maka perlu menambah jenis kriteria yang digunakan agar proses pencarian *homestay* lebih spesifik dan sesuai dengan keinginan calon wisatawan. Selain itu, pada penelitian saya, nilai kepentingan bobot preferensi ditentukan sendiri oleh calon wisatawan karena setiap calon wisatawan memiliki keinginan yang berbeda dalam memilih *homestay*.

2.2 Desa Wisata

Desa wisata adalah sebuah kawasan pedesaan yang memiliki beberapa karakteristik khusus untuk menjadi daerah tujuan wisata. Di kawasan ini, penduduknya masih memiliki tradisi dan budaya yang relatif masih asli. Selain itu, beberapa faktor pendukung seperti makanan khas, sistem pertanian dan sistem sosial turut mewarnai sebuah kawasan desa wisata. Di luar faktor-faktor tersebut, alam dan lingkungan yang masih asli dan terjaga merupakan salah satu faktor terpenting dari sebuah kawasan tujuan wisata.

Selain berbagai keunikan, kawasan desa wisata juga harus memiliki berbagai fasilitas untuk menunjangnya sebagai kawasan tujuan wisata. Berbagai fasilitas ini akan memudahkan para pengunjung desa wisata dalam melakukan kegiatan wisata. Fasilitas-fasilitas yang sebaiknya dimiliki oleh kawasan desa wisata antara lain adalah sarana transportasi, telekomunikasi, kesehatan, dan juga akomodasi. Khusus untuk sarana akomodasi, desa wisata menyediakan sarana penginapan berupa pondok-pondok wisata (*homestay*) sehingga para pengunjung pun turut merasakan suasana pedesaan yang masih asli.

2.3 Pengertian Sistem Pendukung Keputusan

Sistem Pendukung Keputusan (SPK) atau *Decision Support Systems* (DSS) pada awalnya diciptakan oleh dua professor di MIT (Anthony Gorry dan Michael S.Morton) pada tahun 70-an. Menurut mereka DSS harus diarahkan untuk mendukung manajemen pada masalah-masalah yang *semi-structured* (semiterstruktur), yaitu masalah yang memiliki informasi kurang lengkap sehingga para manajer ragu dalam mengambil keputusan. DSS akan memberi dukungan atau alternatif penyelesaian sehingga para manajer dapat menguji alternatif ini untuk memilih mana yang terbaik. (Akid, 2009).

Hal yang sama juga di ungkapkan Kusrini (2007), bahwa DSS merupakan sistem informasi interaktif yang menyediakan informasi, pemodelan dan pemanipulasian data. Sistem ini digunakan untuk membantu pengambilan keputusan dalam situasi yang semi terstruktur dan situasi yang tidak terstruktur, dimana tak seorang pun tahu secara pasti bagaimana keputusan seharusnya dibuat.

2.4 Tujuan Sistem Pendukung Keputusan

Tujuan dari Sistem Pendukung Keputusan adalah sebagai berikut (Turban dkk, 2005):

- a. Membantu manajer dalam pengambilan keputusan atas masalah semi terstruktur.
- b. Memberikan dukungan atas pertimbangan manajer dan bukannya di maksudkan untuk menggantikan fungsi manajer.
- c. Meningkatkan efektivitas keputusan yang di ambil manajer lebih daripada perbaikan efisiensinya.
- d. Kecepatan komputasi. Komputer memungkinkan para pengambil keputusan untuk melakukan banyak *komputasi* secara cepat dengan biaya yang rendah.
- e. Peningkatan produktivitas. Membangun suatu kelompok pengambil keputusan, terutama para pakar, bisa sangat mahal. Pendukung terkomputerisasi bisa mengurangi ukuran kelompok dan memungkinkan para anggotanya untuk

berada di berbagai lokasi yang berbeda-beda (menghemat biaya perjalanan). Selain itu, produktivitas staf pendukung (misalnya analisis keuangan dan hukum) bisa di tingkatkan. Produktivitas juga bisa di tingkatkan menggunakan peralatan optimasi yang menentukan cara terbaik untuk menjalankan sebuah bisnis.

- f. Dukungan kualitas. Komputer bisa meningkatkan kualitas keputusan yang di buat. Sebagai contoh, semakin banyak data yang di akses, makin banyak juga alternatif yang bisa di evaluasi. Analisis resiko bisa di lakukan dengan cepat dan pandangan dari para pakar (beberapa dari mereka berada di lokasi yang jauh) bisa dikumpulkan dengan cepat dan dengan biaya yang lebih rendah. Keahlian bahkan bisa di ambil langsung dari sebuah sistem computer melalui metode kecerdasan tiruan. Dengan *computer*, para pengambil keputusan bisa melakukan simulasi yang kompleks, memeriksa banyak scenario yang memungkinkan, dan menilai berbagai pengaruh secara cepat dan ekonomis. Semua kapabilitas tersebut mengarah kepada keputusan yang lebih baik.

2.5 Langkah-langkah Pemodelan Dalam Sistem Pendukung Keputusan

Saat melakukan pemodelan dalam pembangunan DSS dilakukan langkah-langkah (Kusrini 2007), yaitu sebagai berikut:

- a. Studi Kelayakan (Intelegence)

Pada langkah ini, sasaran ditentukan dan dilakukan pencarian prosedur, pengumpulan data, identifikasi masalah, identifikasi kepemilikan masalah, klasifikasi masalah, hingga akhirnya terbentuk sebuah pernyataan masalah.

- b. Perancangan (Design)

Pada tahapan ini akan di formulasikan model yang akan digunakan dan kriteriakriteria yang ditentukan. Setelah itu, di cari alternatif model yang bisa menyelesaikan permasalahan tersebut. Langkah selanjutnya adalah memprediksi keluaran yang mungkin. Kemudian di tentukan variable-variabel model

c. Pemilihan (Choice)

Setelah pada tahap perancangan di tentukan berbagai alternative model beserta variable-variabelnya. Pada tahapan ini akan dilakukan pemilihan modelnya, termasuk solusi dari model tersebut. Selanjutnya, dilakukan analisis sensitivitas, yakni dengan mengganti variable.

d. Fase Implementasi (*implementation of solution*).

Pada hakikatnya implementasi suatu solusi yang diusulkan untuk suatu masalah adalah inisiasi terhadap hal baru, atau pengenalan terhadap perubahan. Dan perubahan harus dikelola. Harapan-harapan pengguna harus dikelola sebagai bagian dari manajemen perubahan.

2.6 Karakteristik Sistem Pendukung Keputusan

Beberapa karakteristik yang membedakan Sistem Pendukung Keputusan dengan sistem informasi lainnya (Turban dkk, 2005) yaitu:

- a. Sistem Pendukung Keputusan dirancang untuk membantu pengambilan keputusan dalam memecahkan masalah yang sifatnya semi terstruktur ataupun tidak terstruktur.
- b. Dalam proses pengolahannya, sistem pendukung keputusan mengkombinasikan model-model analisis dengan teknik pemasukan dan konvensional secara fungsifungsi pencarian informasi.
- c. Sistem Pendukung Keputusan dirancang sedemikian rupa sehingga dapat digunakan atau dioperasikan dengan mudah oleh orang-orang yang tidak memiliki dasar kemampuan pengoprasisan komputer yang tinggi. Oleh karena itu pendekatan yang digunakan biasanya model interaktif.
- d. Sistem Pendukung Keputusan dirancang dengan menekankan pada aspek fleksibilitas serta kemampuan adaptasi yang tinggi. Sehingga mudah disesuaikan dengan berbagai perubahan lingkungan yang terjadi pada kebutuhan pemakai.

Dari karakter yang telah disebutkan, maka sistem pendukung keputusan mampu memberikan manfaat bagi penggunanya. Kemampuan yang dimaksud diantaranya adalah :

- a. Dapat menangani masalah semi terstruktur dan tidak terstruktur.
- b. Dapat membantu pengambilan keputusan pada semua level manajerial.
- c. Sistem pendukung keputusan dapat menunjang pengambilan keputusan yang sifatnya kelompok ataupun perorangan.
- d. Memiliki kemampuan pemodelan dan analisis pembuatan keputusan.
- e. Dapat beradaptasi dan bersifat fleksibel terhadap kebutuhan.
- f. Mudah melakukan interaksi dan digunakan pada pemakai akhir.
- g. Dapat meningkatkan efektifitas dalam pembuatan keputusan, namun bukan efisiensi.
- h. Sistem pendukung keputusan mudah melakukan pengaksesan berbagai sumber dan format data.
- i. Dapat diimplementasikan sebagai aplikasi yang berdiri sendiri ataupun terdistribusi.
- j. Model secara umum digunakan untuk menganalisis situasi pengambilan keputusan.
- k. Pengambil keputusan memiliki kendali penuh pada proses pengambilan keputusan dalam memecahkan masalah.

2.7 Komponen Sistem Pendukung Keputusan

Sistem Pendukung Keputusan terdiri dari empat subsistem utama, yaitu Subsistem Manajemen Data, Subsistem Manajemen Model, Subsistem Antarmuka Pengguna, dan Subsistem Basis Pengetahuan.

2.7.1 Subsistem Manajemen Data

Pada subsistem manajemen data terdapat basis data yang berisikan data-data yang relevan dengan situasi yang ada dan dikelola menggunakan perangkat lunak yang disebut *Database Management System* (DBMS). Biasanya data disimpan dan

diakses melalui suatu *database web server*. Kemampuan yang dibutuhkan dari manajemen basis data dapat disimpulkan sebagai berikut:

- 1) Kemampuan untuk mengkombinasikan berbagai variasi data melalui pengambilan dan ekstraksi data.
- 2) Kemampuan untuk menambahkan sumber data secara cepat dan mudah.
- 3) Kemampuan untuk menggambarkan struktur data logikal sesuai dengan pengertian pemakai. Kemampuan menangani data secara personal, sehingga pemakai dapat mencoba berbagai alternatif penanganan data.
- 4) Kemampuan mengelola berbagai variasi data.

2.7.2 Subsistem Manajemen Model

Salah satu keunggulan SPK adalah kemampuan mengintegrasikan akses data dan model keputusan. Hal ini dapat dilakukan dengan menambahkan model keputusan ke dalam sistem informasi yang menggunakan database sebagai mekanisme integrasi dan komunikasi antar model. Kemampuan yang dibutuhkan pada subsistem manajemen model meliputi:

- 1) Kemampuan untuk menciptakan model-model baru secara cepat dan mudah.
- 2) Kemampuan untuk mengakses dan mengintegrasikan model-model keputusan.
- 3) Kemampuan untuk mengelola basis model dengan fungsi manajemen yang analog dan manajemen database.

2.7.3 Subsistem Antarmuka Pengguna

Fleksibilitas dan kekuatan karakteristik SPK ialah adanya kemampuan berinteraksi antara sistem dan pemakai, yang dinamakan subsistem *user interface* (antarmuka pengguna). Subsistem ini dapat dibagi menjadi tiga bagian yaitu:

- 1) Bahasa aksi, meliputi apa yang dapat digunakan oleh pemakai dalam berkomunikasi dengan sistem.
- 2) Bahasa tampilan dan presentasi, meliputi apa yang harus diketahui oleh pemakai.

- 3) Basis pengetahuan, meliputi apa yang harus diketahui pemakai agar penggunaan sistem pendukung keputusan bisa efektif.

Kemampuan yang harus dimiliki oleh subsistem pendukung keputusan ini meliputi:

- 1) Kemampuan menangani versi dialog, sesuai kondisi pemakai.
- 2) Kemampuan mengakomodasi tindakan pemakai dengan berbagai alat masukan.
- 3) Kemampuan menampilkan data dengan berbagai variasi format dan alat keluaran.
- 4) Kemampuan untuk mendukung dan mengetahui basis pengetahuan pemakai.

2.7.4 Subsistem Basis Pengetahuan

Subsistem basis pengetahuan adalah subsistem yang sifatnya opsional, namun akan sangat menguntungkan apabila digunakan untuk menunjang tiga subsistem utama. Subsistem ini menggunakan kecerdasan buatan sehingga sistem dapat mengambil tindakan secara otomatis sesuai dengan keinginan pengguna.

2.8 Metode Pengambilan Keputusan

Multi Attribute Decision Making (MADM) merupakan metode pengambilan keputusan dalam menetapkan alternatif terbaik dari sejumlah alternatif yang telah ada berdasarkan beberapa kriteria tertentu (Kusumadewi, 2006). Kriteria tersebut merupakan ukuran-ukuran, aturan-aturan serta standar yang digunakan dalam pengambilan keputusan. Secara umum MADM menyeleksi alternatif terbaik dari sejumlah alternatif. Menurut Janko dalam Kusumadewi (2006), ada beberapa fitur yang digunakan dalam MADM, yakni :

1. Alternatif, merupakan beberapa obyek berbeda yang memiliki kesempatan sama untuk dipilih oleh pengambil keputusan.
2. Atribut, sering disebut juga kriteria atau karakteristik.
3. Konflik antar kriteria, antar kriteria satu dengan yang lainnya biasanya muncul konflik. Seperti antara keuntungan dengan biaya.

4. Bobot keputusan, pada MADM dicari bobot kepentingan dari setiap kriteria guna menunjukkan kepentingan relatif dari setiap kriteria. $W=(w_1, w_2, w_3, \dots)$.
5. Matriks keputusan, suatu matriks keputusan X yang berukuran $m \times n$, berisi elemen-elemen x_{ij} , yang merepresentasikan rating dari alternatif A_i ($i=1, 2, \dots, m$) terhadap kriteria C_j ($j=1, 2, \dots, n$).

MADM dapat diselesaikan dengan beberapa metode, yakni *Simple Additive Weighting Methode* (SAW), *Weighted Product Methode* (WP), ELECTRE, *Technique for Order Preference by Similarity to Ideal Solution* (TOPSIS), *Analytic Hierarchy Process* (AHP).

2.9 Metode *Weighted Product* (WP)

Metode WP merupakan metode yang menggunakan perkalian guna menghubungkan rating atribut, dimana rating setiap atribut harus dipangkatkan dahulu dengan bobot atribut yang bersangkutan (Tan, Fery 2013). Menentukan kriteria yang akan digunakan, kemudian memberi nilai dari setiap kriteria tersebut. Tabel 2.1 menjelaskan contoh dari beberapa kriteria yang digunakan berdasarkan nilai dari setiap kriteria.

Tabel 2. 1 Contoh Tabel Kriteria dan Penilaiannya

No	Kriteria	Nilai	Bobot
1	A_1	W_{01}	W_1
2	A_2	W_{02}	W_2
3	A_3	W_{03}	W_3
4	A_4	W_{04}	W_4
5	A_5	W_{05}	W_5
6	A_n	W_{0n}	W_n

(Sumber : Riza Alfita, 2010)

Pada tabel 2.1 dijelaskan bahwa kriteria yang digunakan diasumsikan dengan kriteria A_1 , sampai A_n . Kemudian rating setiap kriteria diasumsikan dengan nilai W_{01}

sampai W_{0n} dan bobot dari masing-masing kriteria diasumsikan dengan W_1 sampai W_n . Menurut Kusumadewi (2006:75) bobot yang diberikan pada setiap nilai kriteria ditentukan berdasarkan tingkat kepentingan setiap kriteria. Tingkat kepentingan yang diberikan kepada setiap kriteria dapat dilihat pada Tabel 2.2.

Tabel 2. 2 Tabel Pembobotan Berdasarkan Tingkat Kepentingan

No	Tingkat Kepentingan	Bobot
1	Sangat Rendah	1
2	Rendah	2
3	Cukup	3
4	Tinggi	4
5	Sangat Tinggi	5

(Sumber: Kusumadewi, 2006)

Dalam proses perhitungan metode *Weighted Product* (WP) memiliki tiga tahapan, yaitu:

1. Menghitung nilai perbaikan bobot menggunakan rumus 1.

$$W_j = \frac{w_o}{\sum w_o} \quad (2.1)$$

Keterangan :

W_j = Nilai perbaikan bobot *user*

w_o = Nilai Bobot dari *user*

$\sum w_o$ = Jumlah nilai bobot dari *user*

(Sumber: Kusumadewi dkk, 2006)

2. Menghitung nilai preferensi untuk alternatif A_n dengan vektor S menggunakan rumus 2.

$$S_i = \prod_{j=1}^n (X_{ij})^{W_j} \quad (2.2)$$

Keterangan :

S_i = Nilai prefensi alternatif A_n (Vektor S)

W_j = Nilai perbaikan bobot *user* (Positif jika *benefit*/ Negatif jika *cost*)

X_{ij} = Nilai rating kinerja atribut ke-i terhadap subkriteria ke-j

(Sumber: Kusumadewi dkk, 2006)

- Menghitung nilai preferensi relatif dari setiap alternatif dengan menggunakan rumus 3.

$$V_i = \frac{\prod_{j=1}^n (X_{ij})^{W_j}}{\prod_{j=1}^n (X_j^*)^{W_j}} \quad (2.3)$$

Keterangan :

V_i = Nilai preferensi relatif dari setiap alternatif (Vektor V)

W_j = Nilai perbaikan bobot *user* (Positif jika *benefit*/ Negatif jika *cost*)

X_{ij} = Nilai rating kinerja atribut ke-i terhadap subkriteria ke-j

X_j^* = Jumlah nilai preferensi alternatif (Vektor S)

(Sumber: Kusumadewi dkk, 2006)

Hasil dari nilai vektor V merupakan acuan yang digunakan untuk perankingan dari alternatif yang dihitung. Nilai alternatif terbaik yang dipilih merupakan nilai vektor V terbesar dari setiap alternatif. Tahapan metode *Weighted Product* (WP) dapat dilihat pada Gambar 2.1.



Gambar 2. 1 Tahapan Metode Weighted Product (WP)

(Sumber: Kusumadewi, 2006)

Tahapan metode *Weighted Product* (WP) pada Gambar 2.4, tahap awal yaitu menentukan nilai benefit dan cost semua kriteria. Kemudian menghitung perbaikan bobot kriteria, dimana apabila dijumlahkan harus sama dengan 1. Tahap selanjutnya, yaitu memangkatkan semua nilai atribut alternatif dengan nilai perbaikan bobot kriteria yang saling berhubungan. Pada tahap tersebut menghasilkan nilai vektor S. Seluruh nilai vektor S semua alternatif dijumlahkan. Lalu, membagi nilai vektor S setiap alternatif dengan hasil penjumlahan vektor S semua alternatif. Setelah itu akan mendapatkan nilai vektor V setiap alternatif. Tahap terakhir yaitu meranking vektor V dari semua alternatif.

BAB 3. METODOLOGI PENELITIAN

Jenis Penelitian

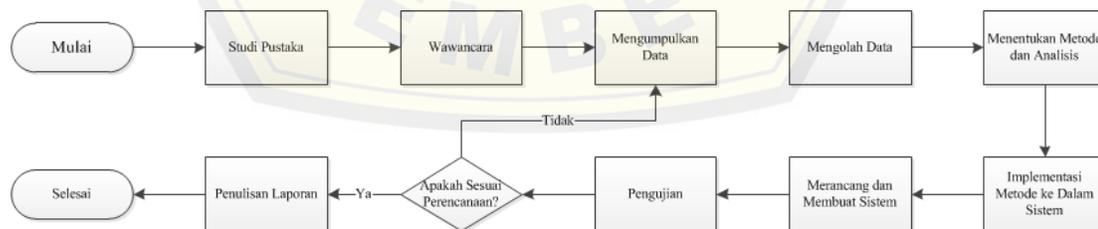
Jenis penelitian yang digunakan dalam penelitian ini merupakan penelitian kualitatif. Penelitian kualitatif merupakan penelitian tentang riset yang bersifat deskriptif dan cenderung menggunakan analisis karena penelitiannya dilakukan pada kondisi sebenarnya.

Objek Penelitian

Penelitian dilakukan di wilayah Jember. Objek pada penelitian ini adalah *homestay* yang ada di Desa Wisata Tanoker Kecamatan Ledokombo Kabupaten Jember.

Metode Penelitian

Penelitian ini akan dilakukan dengan beberapa tahap agar penelitian jelas, berjalan dengan baik dan teratur. Tahap-tahap tersebut diantara tahap pengumpulan data, tahap analisis, tahap pengembangan, dan tahap penulisan laporan. *Detail* tahap penelitian ini dapat dilihat pada Gambar 3.1.



Gambar 3.1 Tahapan Penelitian

3.4 Metode Pengumpulan Data

Teknik pengumpulan data pada penelitian ini meliputi :

- a. Studi Literatur

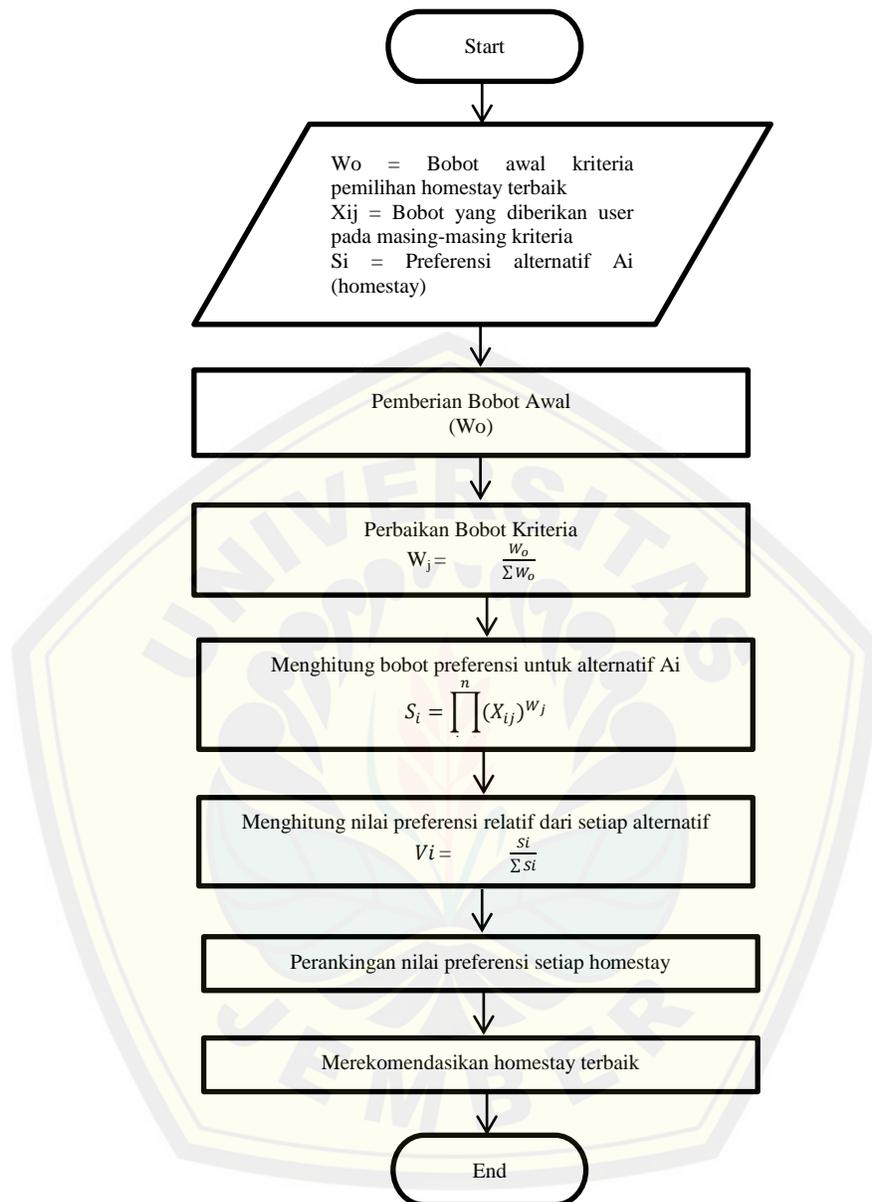
Studi literatur bertujuan untuk menyusun dasar teori yang digunakan dalam penelitian. Sumber yang dapat digunakan sebagai studi literatur antara lain buku, jurnal, karya ilmiah, dan situs web.

b. Wawancara

Salah satu cara untuk mengumpulkan data adalah melakukan wawancara dengan menanyakan langsung kepada narasumber. Pada penelitian ini penulis melakukan wawancara dengan beberapa pihak terkait untuk memperoleh data yang dibutuhkan.

3.5 Metode Analisis

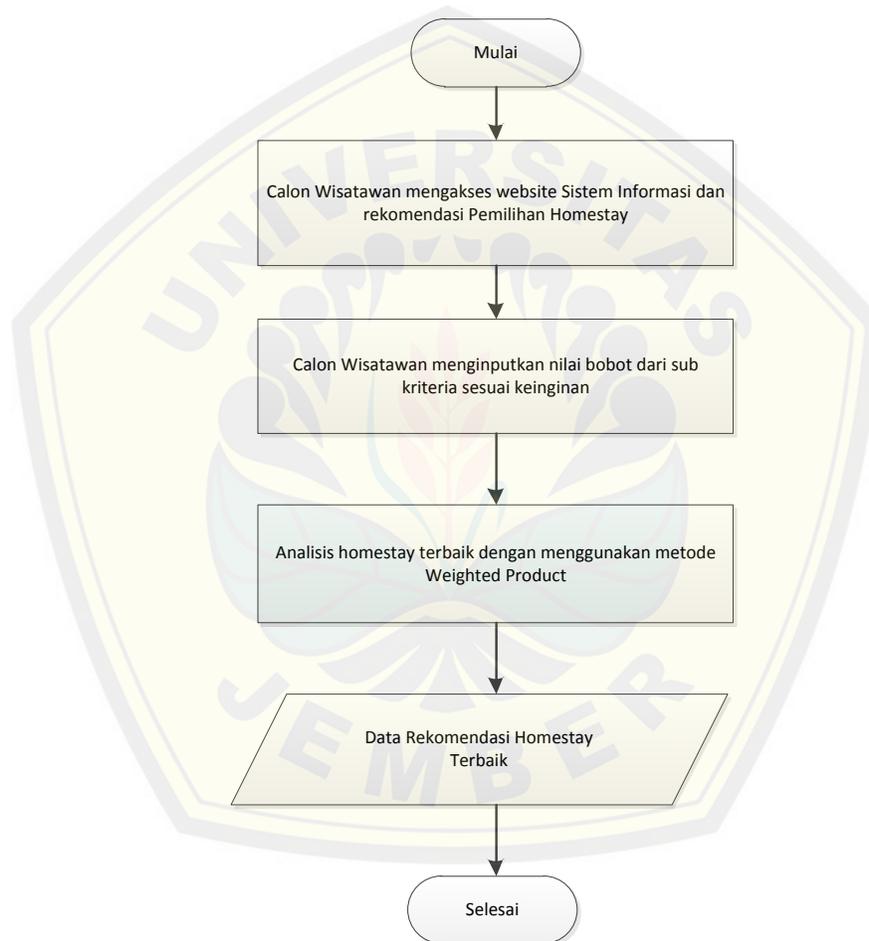
Tahapan analisis data dimulai dengan mengolah data-data yang telah dikumpulkan dari studi literatur, wawancara dan sumber data yang lain. Tahapan berikutnya adalah menganalisa data yang telah diolah menggunakan metode *Weighted Product* (WP) seperti pada Gambar 3.2. Metode WP dimulai dengan pemberian bobot awal kriteria yang dimasukkan oleh *user*. Setelah *user* memberikan nilai bobot selanjutnya dilakukan perbaikan bobot dimana akan bernilai negatif apabila termasuk *cost* dan bernilai positif apabila termasuk *benefit*. Perbaikan bobot tersebut apabila dijumlah harus berjumlah 1. Kemudian menghitung nilai S pada setiap alternatif, yaitu dengan memangkatkan nilai atribut dengan nilai bobot. Setelah didapatkan nilai S, selanjutnya mencari nilai V dari setiap alternatif yaitu dengan membagi nilai S setiap atribut dengan jumlah seluruh nilai S. Selanjutnya tahap perankingan dari nilai V yang sudah didapat untuk menghasilkan nilai tertinggi yang akan menjadi rekomendasi *homestay* terbaik sesuai dengan keinginan *user*.



Gambar 3.2 Tahapan Penerapan Metode *Weighted Product* (WP)

Data yang digunakan dalam pemberian bobot awal adalah data bobot dari masing-masing kriteria yang meliputi jumlah kamar, harga, fasilitas, jarak dengan pusat kegiatan, dan kepadatan penduduk. Selanjutnya data bobot kriteria yang diberikan *user* akan diproses menggunakan metode *Weighted Product* (WP) dengan

tahapan awal perbaikan bobot *user*. Data hasil perbaikan bobot *user* diproses kembali untuk menentukan nilai vektor S setiap lokasi alternatif. Selanjutnya nilai vektor S akan digunakan untuk menentukan nilai vektor V tiap lokasi alternatif yang nantinya akan ditampilkan kepada *user* dalam bentuk perankingan. *Detail* proses analisis pencarian lokasi terbaik menggunakan metode *Weighted Product* (WP) dapat dilihat pada Gambar 3.3.



Gambar 3.3 Diagram Alir Pencarian *Homestay*

Berdasarkan Gambar 3.2 metode *Weighted Product* (WP) digunakan untuk mengatasi masalah pemilihan *homestay* yang kerap dialami calon wisatawan. Wisatawan yang selama ini sulit untuk menentukan *homestay* sesuai dengan

keinginan diharapkan terbantu dengan adanya sistem ini. Karena dengan menggunakan analisis metode *Weighted Product* (WP) wisatawan akan mencari *homestay* berdasarkan tingkat kebutuhan atau kepentingan mereka. Sehingga informasi *homestay* yang akan ditempati nanti dapat sesuai keinginan.

3.6 Tahap Pengembangan Sistem

Dalam penelitian ini model pengembangan sistem yang digunakan oleh penulis adalah model *prototype*. Berikut ini merupakan tahapan dari model *prototype*:

1. Pengumpulan Data

Pengumpulan data dilakukan dengan beberapa cara diantaranya adalah wawancara, studi literature, observasi langsung ke lapangan. Data yang diperlukan dalam pembangunan sistem informasi dan rekomendasi pemilihan *homestay* adalah data *homestay* di Desa Wisata Tanoker.

2. Analisis Kebutuhan

Tahapan ini digunakan untuk menganalisis kebutuhan yang dibutuhkan *user* dalam pengembangan sistem. Kebutuhan yang dimaksud meliputi data kebutuhan fungsional dan data kebutuhan non fungsional. Data yang dibutuhkan merupakan data *homestay* di Desa Wisata Tanoker Kecamatan Ledokombo Kabupaten Jember yang nantinya akan dianalisis dengan metode WP dalam pengambilan keputusan.

3. Membangun Prototype

Setelah kebutuhan fungsional dan non fungsional ditentukan, tahapan selanjutnya adalah membangun *prototype*, yaitu dengan memberikan perancangan sementara kepada *user* sesuai dengan kebutuhan fungsional. Setelah semua kebutuhan telah ditetapkan maka tahapan selanjutnya adalah pembuatan desain sistem. Desain sistem yang digunakan adalah bahasa pemodelan *Unified Modeling Language* (UML) dan menggunakan konsep OOP. Dalam UML terdapat beberapa diagram yang harus digambarkan untuk mendukung pembuatan program. Diagram-diagram tersebut adalah *business process*, *Use case diagram*, *Use case skenario*, *Sequence diagram*, *activity diagram*, *class diagram*, dan *entity relationship diagram*.

a. *Bussiness Process*

Diagram *bussiness process* menjelaskan gambaran umum dari sistem yang akan dikembangkan. Seperti data-data yang digunakan dalam mengembangkan sistem, *input* dan *output* dari sistem serta goal dari sistem yang akan dikembangkan.

b. *Use Case*

Use case mendefinisikan fitur-fitur yang akan dibangun dalam sistem beserta *user* yang memiliki hak akses terhadap fitur tersebut. Pada diagram ini semua fitur yang menyusun sistem harus dijelaskan secara jelas karena *use case diagram* merupakan gambaran dari kebutuhan *user*.

c. *Use Case Skenario*

Fitur-fitur yang telah dirancang akan dideskripsikan menjadi sebuah skenario program yang menjadi landasan dari proses yang terjadi di dalam sistem. Pada diagram ini skenario alur sistem dideskripsikan menjadi normal *flow* dan *alternative flow*.

d. *Activity Diagram*

Activity diagram menerjemahkan skenario sistem yang telah dibuat menjadi sebuah diagram alur untuk mempermudah programmer dalam melakukan *coding program*. Semua alur sistem yang digambarkan dalam *activity diagram* harus sama dengan skenario yang berada pada *use case skenario*.

e. *Sequence Diagram*

Sequence diagram menggambarkan proses yang terjadi di dalam sistem, yaitu interaksi antara *user* ketika mengakses fitur sistem dan sistem meresponnya melalui *model*, *view* dan *controller*. Dalam diagram ini dijelaskan *function-function* yang digunakan ketika *user* mengakses fitur-fitur sistem.

f. *Class Diagram*;

Class diagram mendefinisikan *class-class* yang menyusun sistem beserta relasi logika yang dimiliki oleh *class-class* tersebut. Dalam penelitian ini peneliti menggunakan *Object Oriented Programming* sehingga *class-class* yang menyusun *class diagram* nanti terdiri dari *model*, *view*, dan *controller*.

g. *Entity Relationship Diagram (ERD)*

Dalam *entity relationship diagram* digambarkan *database* yang digunakan sistem dalam melakukan manajemen data sistem. Pada diagram ini akan dijelaskan tabel-tabel data yang menyusun sistem beserta dengan relasi logikanya.

4. Penulisan Kode Program

Tahapan implementasi desain kedalam bahasa pemrograman dilakukan berdasarkan desain hasil analisis kebutuhan yang telah dibuat pada tahapan sebelumnya. Beberapa hal yang dilakukan dalam tahapan implementasi penulisan program yakni :

- a. Penulisan *code program (coding)* menggunakan bahasa pemrograman *Hyper Text Pre-Processor (PHP)*, *Cascade Style Sheet (CSS)* dan *Javascript*.
- b. Manajemen data menggunakan DBMS *MySql*.

5. Pengujian Program

Pada pengujian program dilakukan dua metode umum, yakni :

a. *White-box Testing*

Pengujian *white box* dilakukan pada kode program yang membangun sistem. Pengujian program yang dilakukan menggunakan *Cyclomatic Complexity (CC)* yang ditujukan untuk mengukur tingkat kompleksitas dari program yang telah dirancang.

b. *Black-Box Testing*

Black-box testing merupakan pengujian perogram yang dilakukan tanpa mengetahui kinerja internal dari program tersebut. Jadi pengujian ini hanya melihat kinerja program dari sisi spesifikasi dan kebutuhan yang telah dianalisis pada saat awal perancangan. Pengujian ini dilakukan oleh *developer* dan *user* yaitu orang yang ingin melakukan pemilihan *homestay* di Desa Wisata Tanoker Kecamatan Ledokombo Kabupaten Jember.

BAB 4. DESAIN DAN PERANCANGAN SISTEM

Bab 4 ini menjelaskan penerapan desain dan perancangan sistem, menggunakan model pengembangan prototype, meliputi identifikasi kebutuhan, membangun prototype, mengkodekan sistem, menguji sistem, implementasi program, dan pemeliharaan.

4.1 Pengumpulan Data

Pengumpulan data dilakukan dengan beberapa cara diantaranya adalah wawancara, studi literature, observasi langsung ke lapangan. Data yang diperlukan dalam pembangunan sistem informasi dan rekomendasi pemilihan *homestay* adalah data *homestay* di Desa Wisata Tanoker.

Seperti dijelaskan pada bab sebelumnya, penulis melakukan observasi dan wawancara kepada pemilik Desa Wisata Tanoker, disana penulis mencari informasi tentang bagaimana sistem pembukuan dan pencatatan data serta reservasi *homestay* di Desa Wisata tanoker dan bagaimana sistem pemilihan *homestay* bagi wisatawan di desa wisata tersebut.

4.2 Analisis Kebutuhan

Setelah melakukan observasi dan pengumpulan data, tahapan selanjutnya adalah analisa kebutuhan, yaitu tahap pengumpulan kebutuhan yang dimiliki oleh sistem guna memenuhi keinginan pengguna yang dideskripsikan secara lengkap melalui kebutuhan fungsional dan non-fungsional.

1. Kebutuhan Fungsional

Tabel 4.1 Kebutuhan Fungsional

SRS_ID	Identifikasi
SRS_01	Sistem mampu mengelola data pengguna yang akan menggunakan sistem, ada proses <i>edit</i> dan hapus.
SRS_02	Sistem mampu mencatat data-data <i>homestay</i> di Desa Wisata Tanoker, ada proses <i>edit</i> dan hapus.
SRS_03	Sistem mampu mengelola reservasi (pemesanan) dari <i>homestay</i> .
SRS_04	Sistem mampu merubah <i>profile</i> dan <i>password</i> super admin, admin, dan pemilik <i>homestay</i>
SRS_05	Sistem mampu mencatat data pemilik <i>homestay</i> baru di Desa Wisata Tanoker, ada proses <i>edit</i> dan hapus.
SRS_06	Sistem mampu melakukan pemilihan <i>homestay</i> sesuai keinginan calon pengunjung (wisatawan) dengan menggunakan Sistem Pendukung Keputusan (SPK) metode <i>Weighted Product</i>

2. Kebutuhan Non Fungsional

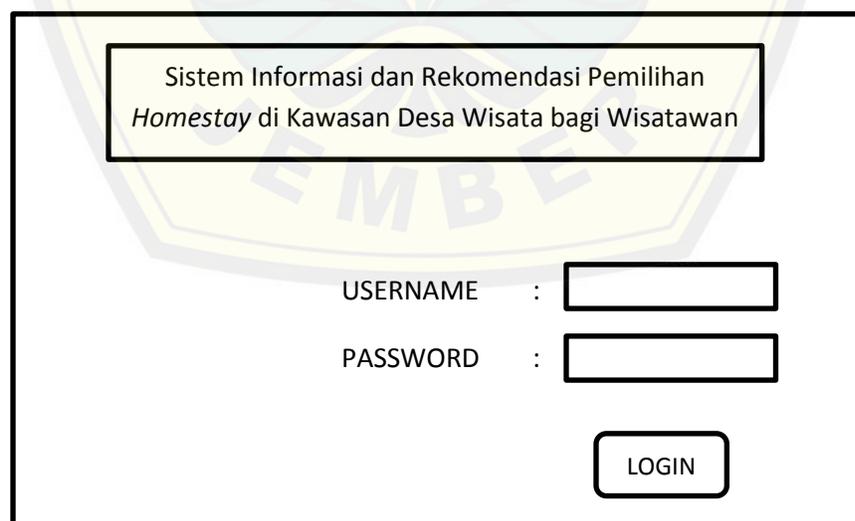
Tabel 4.2 Kebutuhan non-fungsional

SRS_ID	Identifikasi
SRS_07	Sistem dapat diakses secara bersamaan oleh <i>user</i> yang berbeda
SRS_08	Sistem dapat berjalan pada <i>platform</i> atau sistem operasi apa saja yang mendukung aplikasi berbasis web
SRS_09	Sistem menggunakan <i>enkripsi password</i> pengguna untuk keamanan saat <i>autentifikasi login</i>
SRS_10	Sistem pendukung keputusan berbasis web ini harus mudah digunakan (<i>user friendly</i>)

4.3 Membangun Prototype

Setelah kebutuhan fungsional dan non fungsional ditentukan, tahapan selanjutnya adalah membangun prototype, yaitu dengan memberikan perancangan sementara kepada *user* sesuai dengan kebutuhan fungsional.

1. Halaman Login, merupakan halaman yang digunakan oleh admin untuk dapat masuk dalam sistem yang akan digunakan. Untuk mendapat akses masuk ke dalam sistem, *user* (super admin, admin, dan pemilik *homestay*) harus menginputkan login *username* dan *password* dengan benar pada textbox yang telah disediakan. Pada saat tombol login ditekan, sistem akan mengecek *username* dan *password* yang diinputkan dengan data *username* dan *password* yang telah tersimpan di database. Jika data *username* dan *password* benar atau cocok maka *user* akan masuk ke dalam sistem sesuai dengan *role* yang dimilikinya. Sebaliknya jika *username* dan *password* salah atau tidak cocok maka akan diberikan pesan peringatan. Halaman login dapat dilihat pada gambar 4.1



The image shows a login form titled "Sistem Informasi dan Rekomendasi Pemilihan Homestay di Kawasan Desa Wisata bagi Wisatawan". The form contains two input fields: "USERNAME" and "PASSWORD", each followed by a colon and a rectangular text box. Below these fields is a rounded rectangular button labeled "LOGIN".

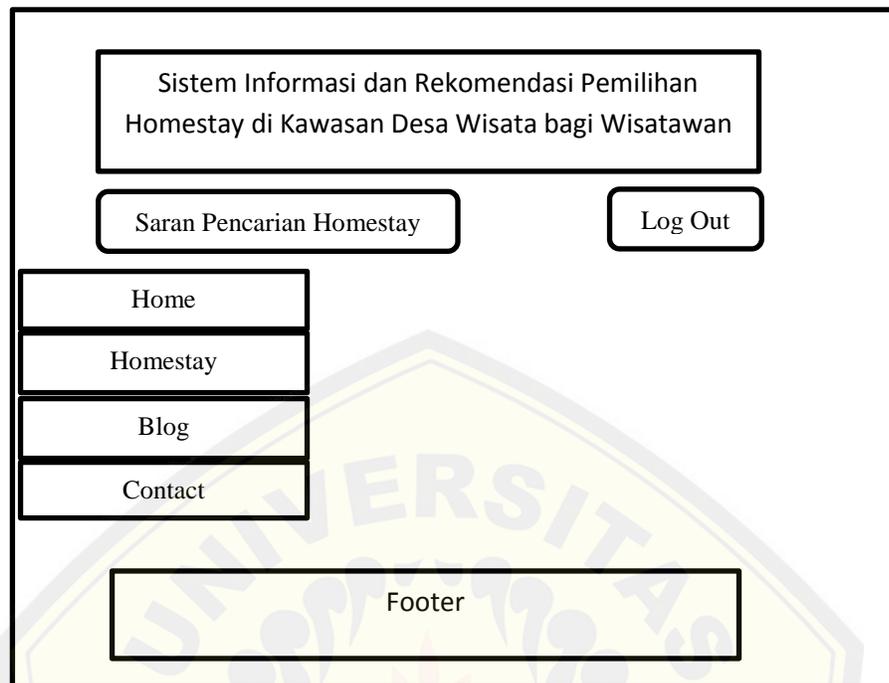
Gambar 4.1 Rancangan Antarmuka Login

- Halaman Utama Super Admin dan Admin, rancangan antarmuka gambar 4.2 ini merupakan halaman utama dari pengguna administrator setelah pengguna sukses melakukan login. Pilihan menu yang akan ditampilkan pada bagian sidebar sesuai dengan hak akses masing-masing *role*. Fitur yang disediakan antara lain *accept booking* baru, ubah *password*, dan ubah *profile*.



Gambar 4.2 Rancangan Antarmuka Halaman Utama Super Admin dan Admin

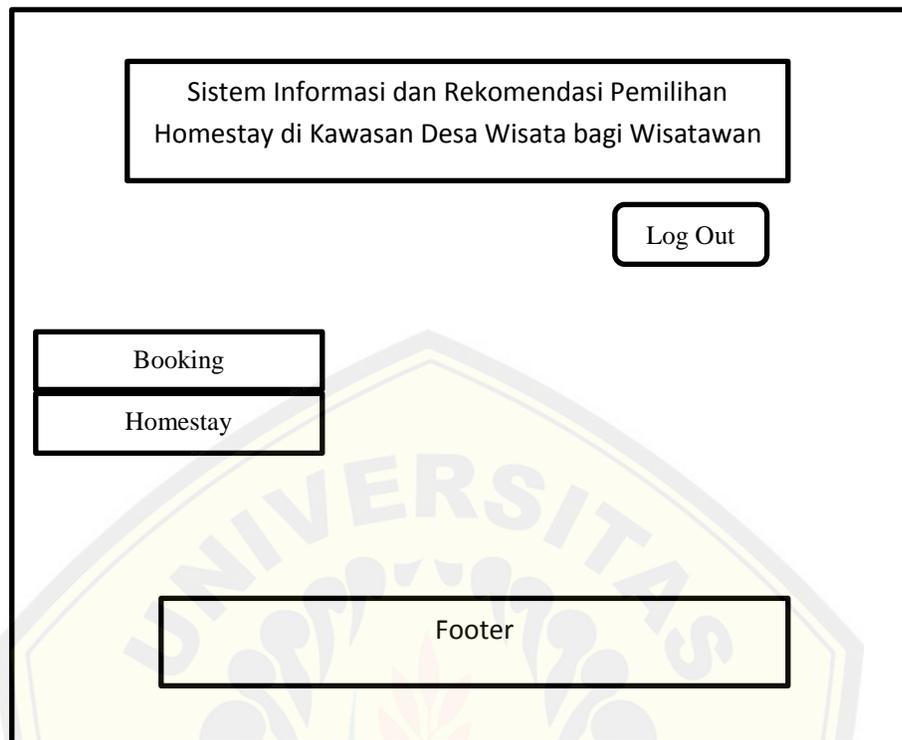
- Halaman Wisatawan, halaman ini digunakan oleh *user* dengan *role* wisatawan untuk melakukan reservasi *homestay* dan melakukan pemilihan *homestay* sesuai dengan keinginan wisatawan. Pada halaman ini, wisatawan dapat melakukan pemesanan terhadap *homestay* dengan mengisi *form booking*. Fitur saran pencarian *homestay* dapat dimanfaatkan oleh wisatawan untuk mendapatkan rekomendasi *homestay* terbaik sesuai dengan nilai bobot dari tiap kriteria yang diinputkan oleh *user*.



Gambar 4.3 Rancangan Antarmuka Halaman Utama Wisatawan

4. Halaman Pemilik *Homestay*

Halaman ini digunakan oleh *user* dengan *role* pemilik *homestay* admin untuk mendaftarkan *homestay* yang dimiliki oleh warga (pemilik *homestay*), antarmuka ini dapat mencatat data registrasi *homestay* yang diinputkan oleh pemilik *homestay*. Setelah mendaftarkan *homestay* yang dimiliki, pemilik *homestay* menunggu untuk mengetahui apakah *homestay* yang didaftarkan telah sesuai dengan persyaratan pihak desa wisata atau tidak. Apabila sudah sesuai maka super admin akan menerima *homestay* tersebut sehingga *homestay* yang didaftarkan oleh pemilik *homestay* dapat tampil di sistem informasi dan rekomendasi pemilihan *homestay*. Sebaliknya, apabila super admin tidak menerima *homestay* tersebut maka *homestay* tidak dapat tampil di sistem informasi dan rekomendasi pemilihan *homestay*.



Gambar 4.4 Rancangan Antarmuka Halaman Utama Pemilik *Homestay*

4.3.1 Evaluasi Prototype

Evaluasi ini dilakukan oleh *user*, jika pembangunan prototype sudah sesuai dengan keinginan *user* maka dapat dilanjutkan ke tahapan selanjutnya, atau pengkodean sistem, jika tidak sesuai maka harus mengulangi langkah 1 dan 2.

4.3.2 Mengkodekan Sistem

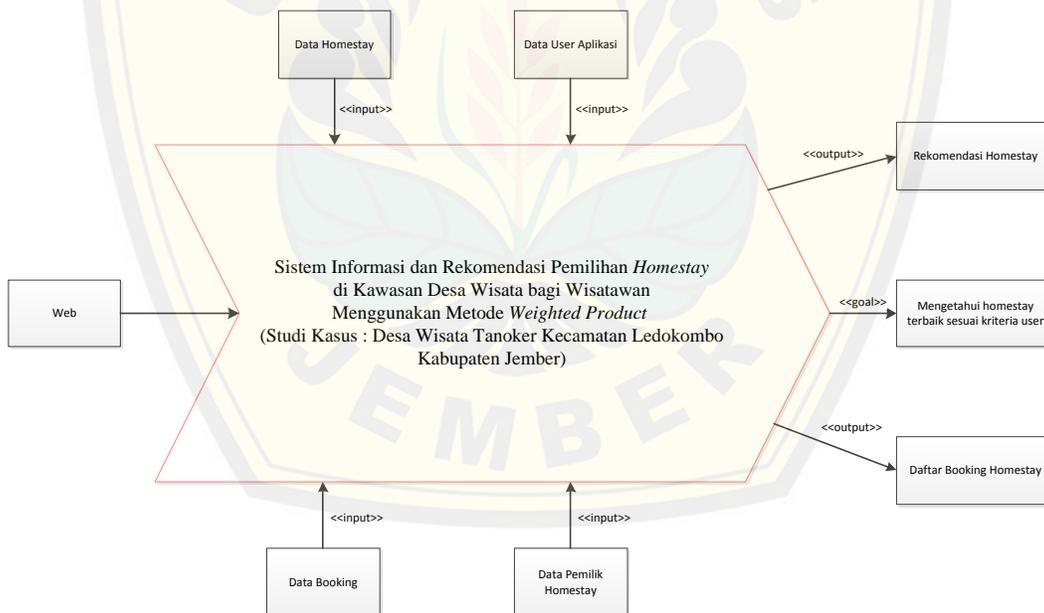
Mengkodekan sistem adalah tahap penerapan ke dalam perancangan sistem dan kedalam bahasa pemrograman.

Setelah semua kebutuhan telah ditetapkan maka tahapan selanjutnya adalah pembuatan desain sistem. Desain sistem yang digunakan adalah bahasa pemodelan *Unified Modeling Language (UML)* dan menggunakan konsep OOP. Dalam UML terdapat beberapa diagram yang harus digambarkan untuk mendukung pembuatan program. Diagram-diagram tersebut adalah *business process*, *Use case diagram*, *Use*

case skenario, Sequence diagram, activity diagram, class diagram, dan entity relationship diagram.

4.3.2.1 Bussiness Process

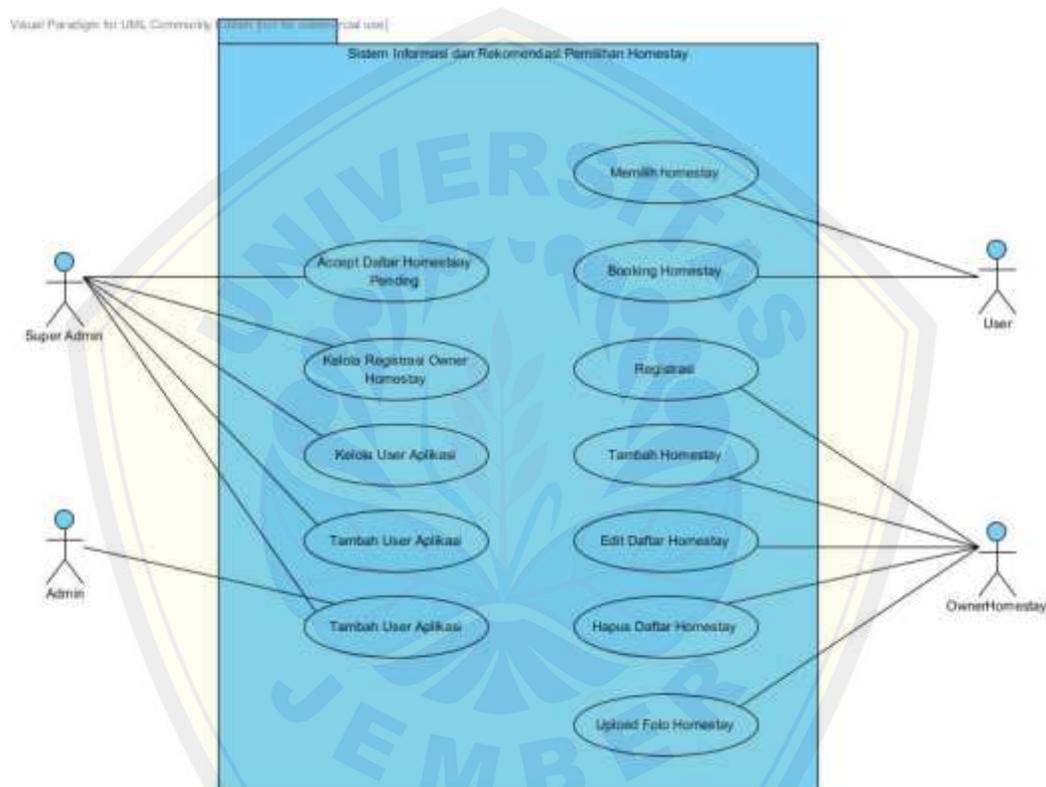
Bussiness process merupakan sebuah model diagram yang menggambarkan proses keseluruhan pada sistem yang terdiri dari *trigger*, *uses*, *input*, *supply*, *output*, dan *goal*. *Input* pada sistem informasi dan rekomendasi pemilihan *homestay* yaitu data alternatif, data kriteria, dan data nilai alternatif. Data *homestay* sebagai data pendukung atau *supply*. Keluaran dari sistem berupa daftar rekomendasi *homestay* sesuai dengan keinginan dan kriteria yang dimasukkan oleh wisatawan. Tujuannya adalah untuk mengetahui daftar *homestay* yang sesuai dengan keinginan wisatawan. Sistem informasi dan rekomendasi ini akan dibangun berbasis web. Gambaran *bussiness process* sistem dapat dilihat pada Gambar 4.5



Gambar 4.5 Bussiness Process

4.3.2.2 Use case Diagram

Use case diagram merupakan identifikasi fungsi atau fitur yang ada pada sistem digambarkan berinteraksi dengan *user* sebagai akses fitur yang bisa digunakan oleh *user* tersebut. Fitur-fitur pada sistem ini digambarkan dengan *elips* dan terdapat 3 *tipe user*. *Use case diagram* pada sistem informasi dan rekomendasi ini digambarkan pada Gambar 4.6



Gambar 4.6 Use Case Diagram

1) Definisi Aktor

Definisi aktor menjelaskan tentang aktor yang berinteraksi dengan sistem dan menjelaskan tentang aktor yang berinteraksi dengan sistem dan menjelaskan hak akses fitur yang dapat digunakan oleh aktor tersebut. Penjelasan definisi aktor dapat dilihat pada tabel 4.3

Tabel 4.3 Definisi Aktor

No.	Aktor	Deskripsi
1.	Super Admin	Aktor yang memiliki hak akses penuh atas fitur yang ada pada sistem di antaranya dapat menggunakan fitur mengelola data <i>homestay</i> dan mengelola data pengguna sistem.
2.	Admin	Aktor ini memiliki hak untuk menerima daftar booking baru dan mengelola data reservasi (pemesanan) <i>homestay</i> . Selain itu juga dapat mengelola akun yang dimiliki yaitu berupa edit <i>profile</i> dan edit <i>password</i> .
3.	User	Aktor yang memiliki akses terbatas pada sistem. Aktor ini hanya dapat melakukan reservasi (pemesanan) <i>homestay</i> , dan melakukan pemilihan <i>homestay</i> sesuai dengan kriteria yang diinginkan.
4.	Pemilik <i>Homestay</i>	Aktor yang dapat melihat info pemesanan dan daftar <i>homestay</i> yang dimiliki serta melakukan registrasi (pendaftaran) <i>homestay</i> yang dimiliki.

2) Definisi *Use case*

Definisi *Use case* merupakan penjelasan dari setiap *Use case* diagram yang merupakan fitur-fitur dari sistem. Penjelasan definisi *Use case* dapat dilihat pada Tabel 4.

Tabel 4.4 Definisi *Use case*

No.	<i>Use case</i>	Deskripsi
1.	Accept Daftar <i>Homestay Pending</i>	<i>Use case</i> yang menggambarkan proses konfirmasi daftar <i>homestay pending</i>
2.	Kelola Registrasi	<i>Use case</i> yang menggambarkan proses

dilanjutkan

lanjutan

	Pemilik <i>Homestay</i>	pengelolaan data registrasi Pemilik <i>Homestay</i>
3.	Kelola <i>User</i> Aplikasi	<i>Use case</i> yang menggambarkan proses menentukan status <i>user</i> aktif atau non aktif
4.	Tambah <i>User</i> Aplikasi	<i>Use case</i> yang menggambarkan proses <i>input</i> data <i>user</i> aplikasi
5.	Accept Daftar Booking Baru	<i>Use case</i> yang menggambarkan proses konfirmasi pemesanan <i>homestay</i>
6.	Memilih <i>Homestay</i>	<i>Use case</i> yang menggambarkan proses untuk memilih <i>homestay</i> sesuai dengan bobot kriteria yang di <i>inputkan</i> oleh <i>user</i>
7.	Booking <i>Homestay</i>	<i>Use case</i> yang menggambarkan proses pemesanan <i>homestay</i>
8.	Registrasi	<i>Use case</i> yang menggambarkan proses pendaftaran akun Pemilik <i>Homestay</i>
9.	Tambah <i>Homestay</i>	<i>Use case</i> yang menggambarkan proses <i>input</i> data <i>homestay</i> baru
10.	Edit Daftar <i>Homestay</i>	<i>Use case</i> yang menggambarkan proses edit data <i>homestay</i>
11.	Hapus Daftar <i>Homestay</i>	<i>Use case</i> yang menggambarkan proses hapus data <i>homestay</i>
12.	Upload Foto <i>Homestay</i>	<i>Use case</i> yang menggambarkan proses upload foto <i>homestay</i>

4.3.2.3 *Use case* Skenario

Use case skenario merupakan penjelasan alur skenario dari sistem yang akan dibuat berdasarkan masing-masing fitur. Setiap *Use case* diagram pada Gambar akan dibuat skenario baik itu skenario normal maupun skenario alternatif.

1) *Use case* Skenario Accept Daftar *Homestay Pending*

Use case skenario ini menjelaskan alur untuk konfirmasi daftar *homestay pending*. Aktor yang bekerja pada *use case* skenario accept daftar *homestay pending* yaitu Super Admin. Pemilik *homestay* mendaftarkan *homestay* yang dimiliki kemudian Super Admin melihat detail dari *homestay* tersebut. Apabila telah sesuai, Super Admin dapat menerima *homestay* tersebut, sehingga *homestay* dapat dipilih oleh calon wisatawan. Namun apabila *homestay* tersebut ditolak oleh Super Admin maka *homestay* tidak dapat dipilih oleh calon wisatawan. Detail penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif dapat dilihat pada Lampiran A.1.

2) *Use case* Skenario Kelola Registrasi Pemilik *Homestay*

Use case skenario ini menjelaskan alur untuk mengelola data registrasi Pemilik *Homestay* yaitu melakukan konfirmasi registrasi Pemilik *Homestay* baru. Pemilik *homestay* yang belum terdaftar wajib untuk mendaftarkan diri terlebih dahulu agar memiliki hak akses pada halaman pemilik *homestay*. Setelah melakukan registrasi, menunggu Super Admin untuk memberikan konfirmasi, apabila diterima maka pemilik *homestay* dapat memiliki hak akses ke halaman pemilik *homestay* sehingga pemilik *homestay* dapat mendaftarkan dan mengelola *homestay* yang dimiliki. Detail penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif dapat dilihat pada Lampiran A.2.

3) *Use case* Skenario Kelola *User* Aplikasi

Use case skenario ini menjelaskan alur untuk proses menentukan status *user* aktif atau non aktif. *Use case* skenario ini juga dimiliki oleh Super Admin. Apabila ada *user* atau admin yang mengundurkan diri maka Super Admin dapat memilih status non aktifkan. Jika ingin mengaktifkan status admin kembali maka Super Admin harus mengubah status menjadi aktifkan. Hal tersebut digunakan agar admin yang non aktif tidak memiliki hak akses terhadap sistem. Detail penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif dapat dilihat pada Lampiran A.3.

4) *Use case* Skenario Tambah *User* Aplikasi

Use case skenario ini menjelaskan alur untuk proses *input* data *user* aplikasi. Super admin dapat menambahkan admin yang bisa mengakses sistem. Super admin harus mengisi Form Tambah *User* Aplikasi kemudian menyimpannya, sehingga admin baru dapat terdaftar. Detail penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif dapat dilihat pada Lampiran A.4.

5) *Use case* Skenario Accept Daftar Booking Baru

Use case skenario ini menjelaskan alur untuk proses konfirmasi pemesanan *homestay* yaitu melakukan konfirmasi apabila pembayaran pemesanan *homestay* telah lunas. Aktor yang berperan yaitu admin dan super admin. Ketika telah di accept, maka email tanda *homestay* yang dipesan telah lunas akan terkirim ke wisatawan. Detail penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif dapat dilihat pada Lampiran A.5.

6) *Use case* Skenario Memilih *Homestay*

Use case skenario ini menjelaskan alur untuk proses memilih *homestay* sesuai dengan bobot kriteria yang di *inputkan* oleh *user*. Calon wisatawan dapat memilih berdasarkan sub kriteria yang paling prioritas kemudian memberikan nilai kepentingan pada setiap kriteria. Ada 6 kriteria, yaitu harga, jumlah kamar, jarak dengan pusat kegiatan, kepadatan penduduk, fasilitas wifi, dan fasilitas hewan peliharaan. Detail penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif dapat dilihat pada Lampiran A.6.

7) *Use case* Skenario Booking *Homestay*

Use case skenario ini menjelaskan alur untuk proses proses pemesanan *homestay*. Wisatawan memilih *homestay* yang akan dipesan kemudian mengisi form booking. Data booking tersebut masuk ke shopping cart. Setelah itu apabila telah menyelesaikan data booking, wisatawan mengisi form order. Kemudian email pemesanan *homestay* akan dikirim ke email wisatawan tersebut. Detail penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif dapat dilihat pada Lampiran A.7.

8) *Use case* Skenario Registrasi

Use case skenario ini menjelaskan alur untuk proses pendaftaran akun Pemilik *Homestay*. Pemilik *homestay* mengisi form registrasi secara lengkap. Kemudian setelah mendaftar, menunggu konfirmasi dari super admin. Apabila telah dikonfirmasi maka pemilik *homestay* dapat log in dengan email dan *password* yang dimiliki. Detail penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif dapat dilihat pada Lampiran A.8.

9) *Use case* Skenario Tambah *Homestay*

Use case skenario ini menjelaskan alur untuk proses *input* data *homestay* baru. Pemilik *homestay* mengisi form tambah *homestay* dengan lengkap beserta gambar dari *homestay* tersebut. Apabila *homestay* telah mendapat konfirmasi dari super admin, maka *homestay* tersebut dapat dipilih oleh calon wisatawan untuk dipesan. Detail penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif dapat dilihat pada Lampiran A.9.

10) *Use case* Skenario Edit Daftar *Homestay*

Use case skenario ini menjelaskan alur untuk proses edit data *homestay*. Ketika *homestay* yang telah didaftarkan mengalami perubahan data, maka pemilik *homestay* dapat mengubah isi dari *homestay* tersebut. Kemudian menunggu kembali untuk mendapat konfirmasi dari super admin. Apabila belum mendapat konfirmasi maka *homestay* tersebut tidak dapat dipilih oleh wisatawan. Detail penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif dapat dilihat pada Lampiran A.10.

11) *Use case* Skenario Hapus Daftar *Homestay*

Use case skenario ini menjelaskan alur untuk proses hapus data *homestay*. *Homestay* yang telah dihapus tidak akan tampil sehingga tidak dapat dipilih oleh wisatawan untuk dipesan. Detail penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif dapat dilihat pada Lampiran A.11.

12) *Use case* Skenario Upload Foto *Homestay*

Use case skenario ini menjelaskan alur untuk proses upload foto *homestay*. Foto dari *homestay* ini merupakan info yang sangat penting. Setelah melihat kondisi *homestay* melalui foto, super admin dapat mengambil keputusan untuk menerima atau menolak *homestay* tersebut. Detail penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif dapat dilihat pada Lampiran A.12.

4.3.2.4 *Activity Diagram*

Activity diagram merupakan gambaran dari alur aktivitas dalam sistem yang sedang dirancang, bagaimana alir tersebut berawal, decision yang mungkin terjadi dan bagaimana alir tersebut berakhir (Febriyanna, dkk, 2015). *Activity diagram* merupakan urutan aktivitas dalam sistem yang bertujuan untuk memperlihatkan aktivitas yang dilakukan serta membantu memahami tahapan proses yang terjadi di dalam suatu fungsi fitur sistem. Pada sistem informasi *homestay* ini terdapat 12 *use case diagram* yang akan digambarkan pada *activity diagram*.

1) *Activity Diagram* Accept Daftar *Homestay Pending*

Activity diagram ini menggambarkan alur aktivitas dari proses *accept* daftar *homestay pending* yang terjadi dalam sistem. Aktor yang menjalankan adalah super admin. Aktor akan memilih pilihan konfirmasi yaitu setuju atau tolak. Apabila disetujui, maka *homestay* dapat muncul di halaman web dan dapat dipesan oleh calon wisatawan. Detail penjelasan alur aktivitas dari proses *accept* daftar *homestay pending* dapat dilihat pada Lampiran B.1.

2) *Activity Diagram* Kelola Registrasi Pemilik *Homestay*

Activity diagram ini menggambarkan alur aktivitas dari proses kelola registrasi Pemilik *Homestay* yang terjadi dalam sistem. Aktor yang menjalankan adalah super admin. Apabila admin menyetujui registrasi yang telah dilakukan oleh pemilik *homestay*, maka pemilik *homestay* dapat login dengan email dan *password*

yang telah didaftarkan. Detail penjelasan alur aktivitas dari proses kelola registrasi Pemilik *Homestay* dapat dilihat pada Lampiran B.2.

3) Activity Diagram Kelola *User* Aplikasi

Activity diagram ini menggambarkan alur aktivitas dari proses kelola *user* aplikasi yang terjadi dalam sistem. Kelola *user* aplikasi ini merupakan hak super admin untuk dapat mengubah status *user* atau admin. Apabila ada admin yang mengundurkan diri maka status admin tersebut diubah menjadi non aktif. Demikian pula jika admin tersebut telah aktif kembali maka statusnya diubah menjadi aktif. Detail penjelasan alur aktivitas dari proses kelola *user* aplikasi dapat dilihat pada Lampiran B.3.

4) Activity Diagram Tambah *User* Aplikasi

Activity diagram ini menggambarkan alur aktivitas dari proses tambah *user* aplikasi yang terjadi dalam sistem. Super admin memiliki hak untuk menambah admin yang dapat mengakses sistem. Super admin mengisi data pada form tambah *user* aplikasi kemudian memilih Simpan. Data admin baru tersebut akan tersimpan di database. Detail penjelasan alur aktivitas dari proses tambah *user* aplikasi dapat dilihat pada Lampiran B.4.

5) Activity Diagram Accept Daftar Booking Baru

Activity diagram ini menggambarkan alur aktivitas dari proses *accept* daftar booking baru yang terjadi dalam sistem. Super admin dan admin dapat memberikan konfirmasi apabila *homestay* yang telah dipesan oleh wisatawan telah lunas. Super admin dan admin dapat menggunakan fitur cari booking yang ada pada sistem untuk memudahkan dalam pencarian ID Order. Detail penjelasan alur aktivitas dari proses *accept* daftar booking baru dapat dilihat pada Lampiran B.5.

6) Activity Diagram Memilih *Homestay*

Activity diagram ini menggambarkan alur aktivitas dari proses memilih *homestay* sesuai keinginan calon wisatawan yang terjadi dalam sistem. Calon wisatawan dapat memilih sesuai prioritas kriteria yang diinginkan. Pada fitur ini terdapat dua halaman. Halaman pertama digunakan calon wisatawan untuk mengisi

atau memilih prioritas dari masing-masing kriteria. Kemudian ketika klik selanjutnya, calon wisatawan dapat mengisi bobot kepentingan dari masing-masing kriteria tersebut. Detail penjelasan alur aktivitas dari proses memilih *homestay* dapat dilihat pada Lampiran B.6.

7) Activity Diagram Booking *Homestay*

Activity diagram ini menggambarkan alur aktivitas dari proses *booking homestay* yang terjadi dalam sistem. Calon wisatawan memilih *homestay* yang akan dipesan. Kemudian pilih *booking homestay* ini untuk bisa mengisi form booking. Pada form booking, calon wisatawan mengisi jumlah kamar yang dibutuhkan beserta tanggal check in dan check out. Setelah di klik simpan, data tersebut akan masuk ke Shopping cart. Lalu pilih lanjut order untuk bisa mengisi form order. Pada form order, calon wisatawan mengisi data email, nama, no telepon yang bisa dihubungi. Konfirmasi pemesanan akan masuk ke email calon wisatawan tersebut. Detail penjelasan alur aktivitas dari proses *booking homestay* dapat dilihat pada Lampiran B.7.

8) Activity Diagram Registrasi

Activity diagram ini menggambarkan alur aktivitas dari proses registrasi pemilik *homestay* yang terjadi dalam sistem. Pemilik *homestay* yang belum memiliki akun, wajib mendaftar pada form registrasi. Pada form tersebut, pemilik *homestay* mengisi data email, *password*, no ktp, dll. Form tersebut harus diisi secara lengkap agar bisa di simpan. Detail penjelasan alur aktivitas dari proses registrasi pemilik *homestay* dapat dilihat pada Lampiran B.8.

9) Activity Diagram Tambah *Homestay*

Activity diagram ini menggambarkan alur aktivitas dari proses tambah *homestay* yang terjadi dalam sistem. Pemilik *homestay* yang telah memiliki hak akses pada sistem, dapat mendaftarkan *homestay* yang dimiliki. Pemilik *homestay* mengisi form tambah *homestay* dan menyertakan foto *homestay*. Apabila telah terdaftar, pemilik *homestay* menunggu konfirmasi dari super admin apakah menerima atau

menolak *homestay* yang didaftarkan tersebut. Detail penjelasan alur aktivitas dari proses tambah *homestay* dapat dilihat pada Lampiran B.9

10) Activity Diagram Edit Daftar *Homestay*

Activity diagram ini menggambarkan alur aktivitas dari proses edit daftar *homestay* yang terjadi dalam sistem. Setelah mendaftarkan *homestay* yang dimiliki, dan telah disetujui oleh super admin, apabila ada perubahan, maka bisa mengubah data *homestay* tersebut. Setelah mengisi form edit *homestay* dan menyimpannya, pemilik *homestay* kembali menunggu konfirmasi dari super admin untuk menerima atau menolak *homestay* tersebut. Detail penjelasan alur aktivitas dari proses edit daftar *homestay* dapat dilihat pada Lampiran B.10.

11) Activity Diagram Hapus Daftar *Homestay*

Activity diagram ini menggambarkan alur aktivitas dari proses hapus daftar *homestay* yang terjadi dalam sistem. *Homestay* yang telah dihapus tidak dapat muncul di halaman pemesanan *homestay*, sehingga calon wisatawan tidak dapat memilih *homestay* tersebut untuk dipesan. Detail penjelasan alur aktivitas dari proses hapus daftar *homestay* dapat dilihat pada Lampiran B.11.

12) Activity Diagram Upload Foto *Homestay*

Activity diagram ini menggambarkan alur aktivitas dari proses upload foto *homestay* yang terjadi dalam sistem. Untuk mengunggah foto, pemilik *homestay* memiliki fitur upload pada kolom aksi. Setelah selesai di-upload foto tersebut dapat muncul di halaman pemesanan *homestay*. Detail penjelasan alur aktivitas dari proses upload foto *homestay* dapat dilihat pada Lampiran B.12.

4.3.2.5 Sequence diagram

Sequence diagram merupakan interaksi antar objek di dalam sistem dengan objek di sekitar sistem (Febriyanna, dkk, 2015). Interaksi yang terjadi di dalam sistem berupa interaksi antara *view*, *controller*, dan *model*. Sedangkan interaksi yang terjadi di sekitar sistem adalah interaksi antara *user* dengan sistem. *Sequence*

diagram menggambarkan perilaku objek terhadap *use case* dengan mendeskripsikan waktu hidup objek dan message yang dikirim dan diterima antar objek.

1) Sequence Diagram Accept Daftar *Homestay Pending*

Sequence diagram accept daftar *homestay pending* menggambarkan interaksi antara aktor yaitu *Super Admin* dengan sistem yang berupa interaksi dengan *view*, *controller*, dan *model*. Super admin memilih daftar *homestay pending*, yang berjalan yaitu *function pending()* untuk menampilkan daftar *homestay pending*. Kemudian klik tombol konfirmasi pada field aksi. Setelah itu super admin memilih setuju atau tolak. Apabila memilih setuju maka *homestay* tersebut dapat muncul di halaman pemesanan *homestay*. Namun apabila memilih tolak, maka *homestay* tersebut tidak dapat muncul di halaman pemesanan *homestay*. Interaksi yang terjadi digambarkan dengan jelas pada Lampiran C.1.

2) Sequence Diagram Kelola Registrasi Pemilik *Homestay*

Sequence diagram kelola registrasi pemilik *homestay* menggambarkan interaksi antara aktor yaitu *Super Admin* dengan sistem yang berupa interaksi dengan *view*, *controller*, dan *model*. Super admin memilih menu owner, kemudian memilih setuju pada field aksi. Setelah itu, akan muncul pilihan setuju dan cancel. Ketika super admin memilih setuju maka akan berjalan *function confirm()* yang berguna untuk menerima registrasi pemilik *homestay* tersebut, sehingga pemilik *homestay* dapat login menggunakan *password* dan email yang telah didaftarkan tersebut. Namun apabila cancel, sistem akan menutup pilhan konfirmasi dan kembali ke halaman Daftar Owner. Interaksi yang terjadi digambarkan dengan jelas pada Lampiran C.2.

3) Sequence Diagram Kelola *User Aplikasi*

Sequence diagram kelola *user* aplikasi menggambarkan interaksi antara aktor yaitu *Super Admin* dengan sistem yang berupa interaksi dengan *view*, *controller*, dan *model*. Super admin memilih menu Daftar *User Aplikasi*. Kemudian untuk mengubah status *user* yaitu terletak pada *controller set_aktif()* kemudian pada model akan dijalankan *update()* untuk mengubah status *user*. Status *user* ada dua,

yaitu *user* aktif dan *user* non aktif. Interaksi yang terjadi digambarkan dengan jelas pada Lampiran C.3.

4) Sequence Diagram Tambah *User* Aplikasi

Sequence diagram tambah *user* aplikasi menggambarkan interaksi antara aktor yaitu *Super Admin* dengan sistem yang berupa interaksi dengan *view*, *controller*, dan *model*. *Super admin* mengisi form tambah *user* aplikasi, kemudian mengklik Simpan. *Controller* yang bekerja yaitu *controller admin_userapp* pada *function add()* setelah itu *model* akan melakukan *insert* data tersebut ke database. Kemudian *view* akan menampilkan notifikasi Sukses Simpan *User*. Interaksi yang terjadi digambarkan dengan jelas pada Lampiran C.4.

5) Sequence Diagram Accept Daftar Booking Baru

Sequence diagram accept daftar booking baru menggambarkan interaksi antara aktor yaitu *Super Admin* dan *Admin* dengan sistem yang berupa interaksi dengan *view*, *controller*, dan *model*. *Controller* yang bekerja yaitu *admin_booking* sedangkan *model* yang bekerja yaitu *booking_model*. Ketika *super admin* mengklik konfirmasi maka *function* yang berjalan pada *controller admin_booking* yaitu *confirm_accept()*. Setelah memilih pilihan konfirmasi, *view* akan menampilkan notifikasi sukses simpan konfirmasi. Interaksi yang terjadi digambarkan dengan jelas pada Lampiran C.5.

6) Sequence Diagram Memilih *Homestay*

Sequence diagram memilih *homestay* menggambarkan interaksi antara aktor yaitu *user* (calon wisatawan) dengan sistem yang berupa interaksi dengan *view*, *controller*, dan *model*. Saat *user* memilih *homestay* sesuai dengan kriteria maka *controller* yang berjalan yaitu *controller web* pada *function suggestion* dan *model web_model*. *Function suggestion* akan mengambil kriteria dengan perintah *GetCriterias()* pada *web_model*. Interaksi yang terjadi digambarkan dengan jelas pada Lampiran C.6.

7) Sequence Diagram Booking *Homestay*

Sequence diagram booking homestay menggambarkan interaksi antara aktor yaitu *user* (calon wisatawan) dengan sistem yang berupa interaksi dengan *view*, *controller*, dan *model*. Ketika *user* memilih *homestay* yang akan dipesan maka yang berjalan adalah *controller* web pada *function homestay()*. Kemudian *web_model* akan menjalankan *function GetHomestay()*. Lalu ketika *user* mulai mengisi data reservasi yang berjalan adalah *function reservation()* pada *controller* web. Interaksi yang terjadi digambarkan dengan jelas pada Lampiran C.7.

8) Sequence Diagram Registrasi

Sequence diagram registrasi menggambarkan interaksi antara aktor yaitu Pemilik *Homestay* dengan sistem yang berupa interaksi dengan *view*, *controller*, dan *model*. Saat pemilik *homestay* mendaftarkan email dan *password* agar mendapat hak akses ke sistem, *controller* yang berjalan adalah *owner_app function register()*. Ketika sukses menyimpan data registrasi maka akan menampilkan notifikasi Sukses simpan data. Interaksi yang terjadi digambarkan dengan jelas pada Lampiran C.8.

9) Sequence Diagram Tambah *Homestay*

Sequence diagram tambah *homestay* menggambarkan interaksi antara aktor yaitu Pemilik *Homestay* dengan sistem yang berupa interaksi dengan *view*, *controller*, dan *model*. Pemilik *homestay* mendaftarkan *homestay* yang dimiliki dengan mengisi form tambah *homestay*. Kemudian form yang telah terisi tersebut disimpan dengan *controller owner_homestay* pada *function add()*. Ketika berhasil menyimpan form maka pada halaman Tambah *Homestay* akan muncul notifikasi Sukses Simpan *Homestay*. Interaksi yang terjadi digambarkan dengan jelas pada Lampiran C.9.

10) Sequence Diagram Edit Daftar *Homestay*

Sequence diagram edit daftar *homestay* menggambarkan interaksi antara aktor yaitu Pemilik *Homestay* dengan sistem yang berupa interaksi dengan *view*, *controller*, dan *model*. Pemilik *homestay* mengubah data *homestay* yang dimiliki dengan mengisi form ubah *homestay*. Kemudian form yang telah terisi tersebut

disimpan dengan *controller* *owner_homestay* pada *function* *edit(\$kode)*. Ketika berhasil menyimpan form maka pada halaman Daftar *Homestay* akan muncul notifikasi Sukses Ubah *Homestay*. Interaksi yang terjadi digambarkan dengan jelas pada Lampiran C.10.

11) Sequence Diagram Hapus Daftar *Homestay*

Sequence diagram hapus daftar *homestay* menggambarkan interaksi antara aktor yaitu Pemilik *Homestay* dengan sistem yang berupa interaksi dengan *view*, *controller*, dan *model*. Pemilik *homestay* dapat menghapus *homestay* yang dimiliki dengan klik hapus pada field Aksi. Kemudian *controller* yang berjalan yaitu *controller* *owner_homestay* pada *function* *delete(\$kode)*. Ketika berhasil menghapus *homestay* pada halaman Tambah *Homestay* akan muncul notifikasi Sukses Hapus *Homestay*. *Homestay* yang terhapus secara otomatis akan terhapus pula pada halaman pemesanan *homestay*. Interaksi yang terjadi digambarkan dengan jelas pada Lampiran C.11.

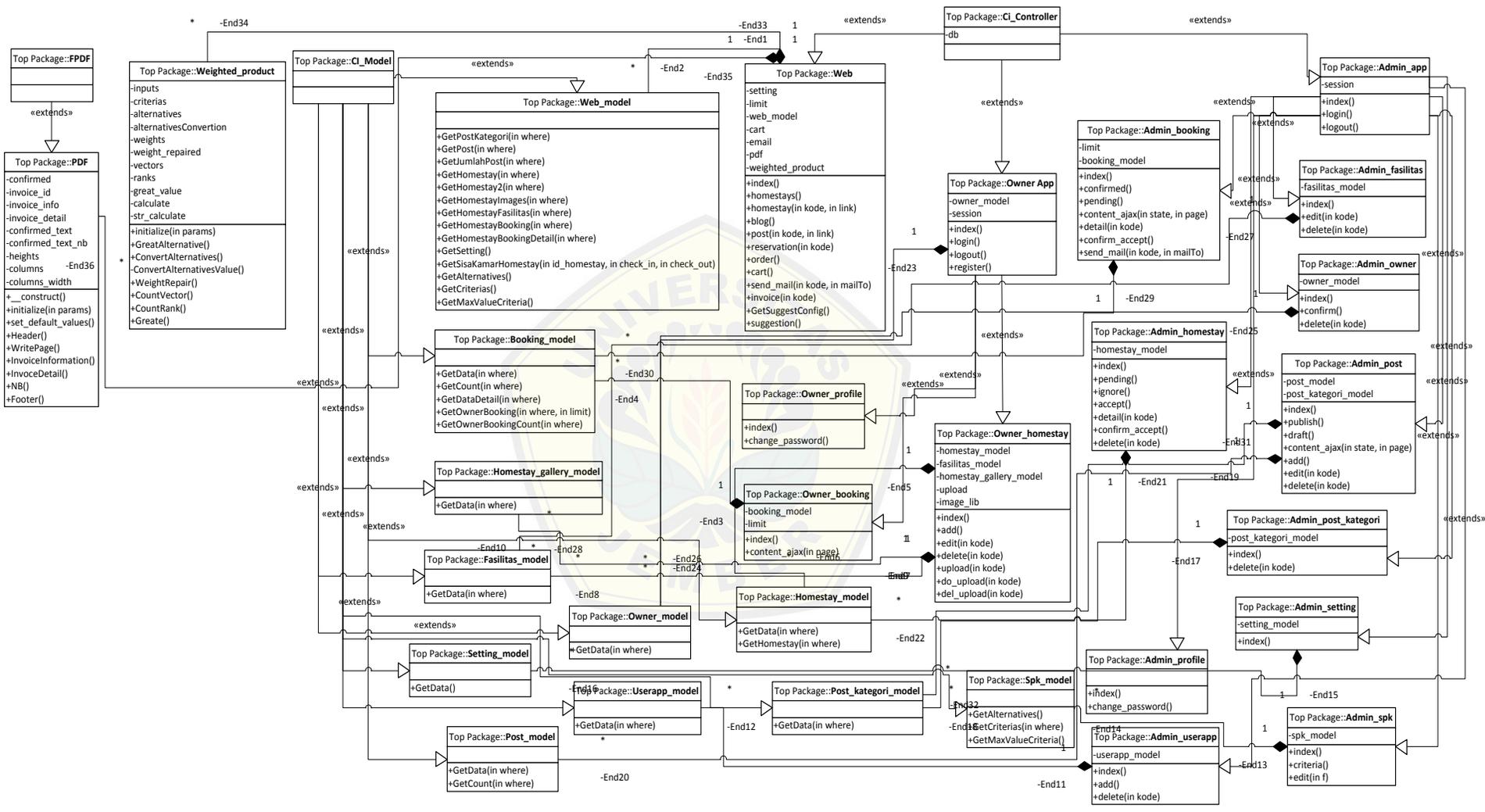
12) Sequence Diagram Upload Foto *Homestay*

Sequence diagram upload foto *homestay* menggambarkan interaksi antara aktor yaitu *Admin* dan *member* dengan sistem yang berupa interaksi dengan *view*, *controller*, dan *model*. Pemilik *homestay* dapat mengunggah foto *homestay* yang dimiliki dengan mengisi form upload foto *homestay*. Kemudian form yang telah terisi tersebut disimpan dengan *controller* *owner_homestay* pada *function* *du_upload(\$kode)*. Ketika berhasil menyimpan foto maka foto akan muncul pada data *homestay* tersebut. Interaksi yang terjadi digambarkan dengan jelas pada Lampiran C.12.

5.1.2.1 Class Diagram

Class diagram merupakan gambaran sistem yang akan dibuat secara statis. Dalam *class diagram* dijelaskan *class-class* yang akan dibangun beserta relasi yang dimiliki (Febriana, 2014:2). Pada *class diagram* ini terdapat beberapa *class* yang mengextend CI *controller* dan CI *model*. Untuk *controller* terdapat *class* *admin_app*,

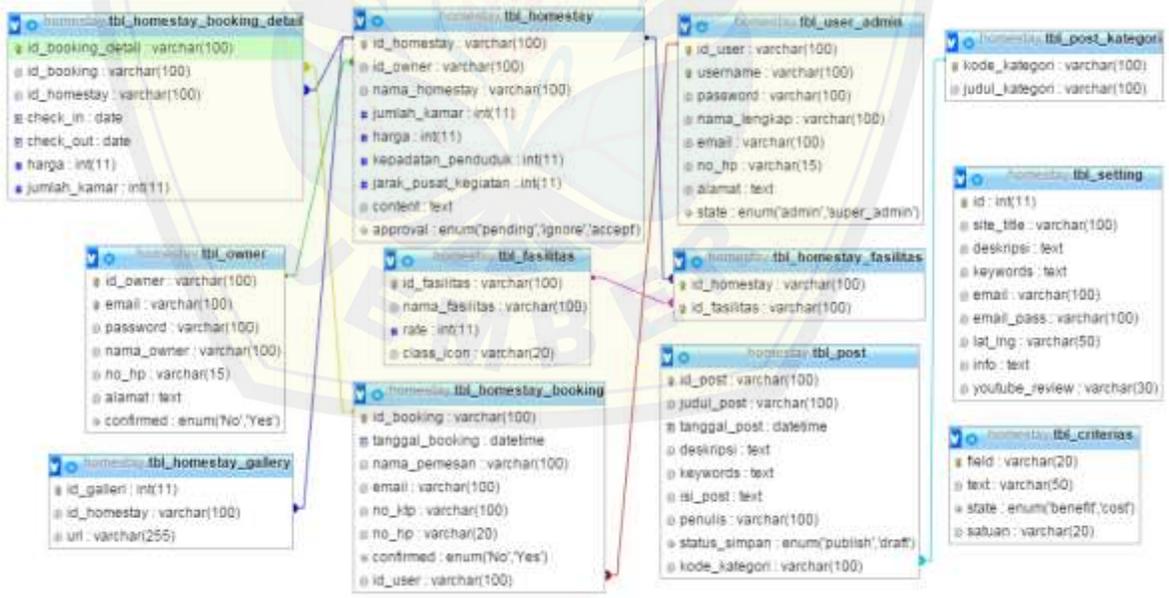
admin_booking, admin_fasilitas, admin_homestay, admin_owner, admin_post, admin_post_kategori, admin_profile, admin_setting, admin_spk, admin_userapp, owner_app, owner_booking, owner_homestay, owner_profile, dan web. Class yang mengextend CI Model antara lain class booking_model, fasilitas_model, homestay_gallery_model, homestay_model, owner_model, post_kategori_model, post_model, setting_model, spk_model, userapp_model, dan web_model. Fitur rekomendasi *homestay* dapat dilihat pada *controller* web dan model *web_model*. Controller web pada function *suggestion()* akan memanggil library *weighted_product*. Library *weighted_product* bertugas untuk melakukan perhitungan dengan menggunakan metode *Weighted Product* (WP). Library tersebut berfungsi ketika calon wisatawan menggunakan fitur Saran Pencarian *Homestay* yang ada pada halaman utama *website* sistem informasi dan rekomendasi pemilihan *homestay*. Pada library *weighted_product*, tahap pertama yaitu mengubah nilai alternatif dari nilai semula menjadi nilai bobot kepentingan sesuai *input* dari calon wisatawan. Setelah itu, melakukan perbaikan bobot, dimana perbaikan bobot tersebut akan menentukan pangkat dari tiap nilai kriteria. Apabila kriteria termasuk *cost*, pangkat akan bernilai negatif, sedangkan *benefit* pangkat akan bernilai positif. Setelah didapatkan nilai pangkat, selanjutnya menghitung nilai S tiap alternatif dengan memangkatkan tiap nilai kriteria dengan pangkat yang didapatkan dari perbaikan bobot. Setelah nilai S didapat, kemudian mencari nilai V, yaitu dengan membagi nilai S pada tiap nilai alternatif dengan jumlah seluruh nilai S. Tahap selanjutnya yaitu perangkaian untuk mengurutkan nilai V dari yang terbesar hingga terkecil. Alternatif yang memiliki nilai V terbesar itulah yang akan menjadi pilihan *homestay* terbaik dengan menggunakan metode *Weighted Product* (WP). Detail *class diagram* dari sistem informasi dan rekomendasi pemilihan *homestay* dapat dilihat pada Gambar 4.7.



Gambar 4.7 Class Diagram

5.1.2.2 Entity Relationship Diagram

Entity Relationship Diagram menjelaskan gambaran basisdata yang ada di dalam sistem berdasarkan semua fitur yang ada di dalam use case diagram. Pada sistem ini terdapat 12 tabel yang tersimpan dalam database. Tabel tersebut antara lain tbl_homestay yang berisi data homestay yang telah terdaftar, tbl_homestay_booking_detail yang berisi data pemesanan homestay, tbl_user_admin yaitu tabel yang berisi data admin beserta username dan password, tbl_post_kategori yaitu berisi kategori dari posting yang terdapat pada blog, tbl_owner berisi data owner yang terdaftar pada sistem informasi dan rekomendasi pemilihan homestay, tbl_fasilitas yaitu tabel yang berisi fasilitas-fasilitas yang ada pada homestay, tbl_homestay_fasilitas yaitu tabel yang merupakan gabungan dari primary key pada tbl_homestay dan tbl_fasilitas dengan hubungan one to many karena pada satu homestay terdapat beberapa fasilitas, tbl_homestay_booking yang berisi data lengkap dari pemesan homestay. Dalam gambar 4.8 dijelaskan Entity Relationship Diagram yang akan dibangun pada sistem informasi dan rekomendasi pemilihan homestay.



Gambar 4.8 Entity Relationship Diagram

4.4 Penulisan Kode Program

Tahap ini merupakan tahap implementasi terhadap desain-desain yang telah dibuat berdasarkan pemodelan UML. Implementasi penulisan kode program ditulis dengan bahas pemrograman *Page Hypertext Pre-Processor* (PHP) dengan bantuan *framework Code Igniter*. Penulisan kode program ditulis berdasarkan fungsi-fungsi yang ada pada sistem. Detail penulisan kode program dapat dilihat pada Lampiran D.

4.5 Pengujian Sistem

Pengujian sistem akan dilakukan dengan menggunakan dua metode pengujian yaitu *White Box Testing* dan *Black Box Testing*

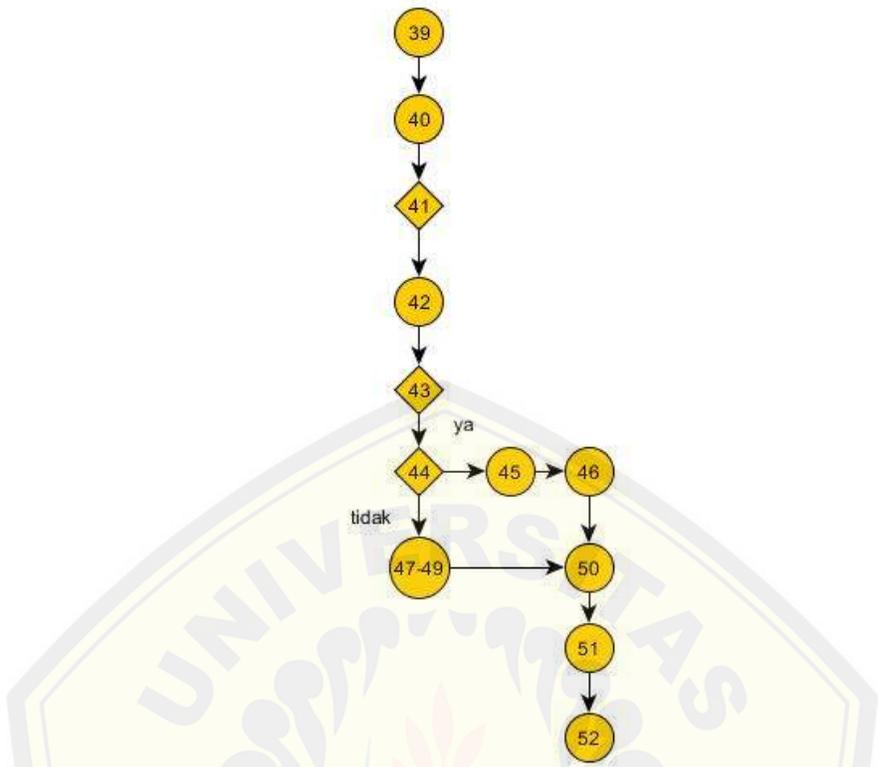
a. *White Box Testing*

White box testing merupakan pengujian dari setiap modul program meliputi listing program, diagram alir, cyclomatic complexity, jalur program independen dan test case. Pada tahap pengujian ini, fungsi yang diuji adalah sebagai berikut.

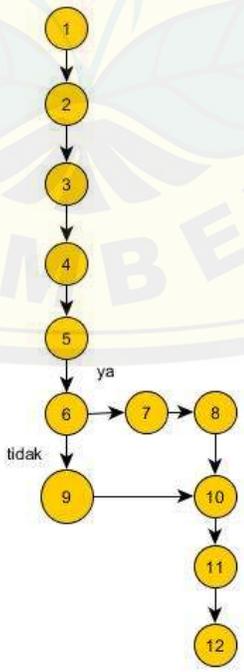
1. Pengujian Listing Program ConvertAlternatives()

```
38 public function ConvertAlternatives(){
39     $this->alternativesConversion = array();
40     $criteria = array_column($this->criteria, 'key');
41     foreach($this->alternatives as $al){
42         $row = array();
43         foreach($al as $key => $val){
44             if(in_array($key, $criteria)){
45                 $row[$key] = $this->ConvertAlternativesValue($key, $val);
46             }
47             else{
48                 $row[$key] = $val;
49             }
50         }
51         array_push($this->alternativesConversion, $row);
52     }
}
```

Gambar 4.9 Listing Program ConvertAlternatives()



Gambar 4.10 Diagram Alir ConvertAlternatives()



Gambar 4.11 Grafik Alir ConvertAlternatives()

a. Kompleksitas Siklometik

Perhitungan kompleksitas siklometik untuk grafik alir G menggunakan rumus

$V(G) = E - N + 2$. E adalah jumlah *edge* dan N adalah jumlah *node* pada grafik alir.

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 12 - 12 + 2 \\ &= 2 \end{aligned}$$

b. Basis Set

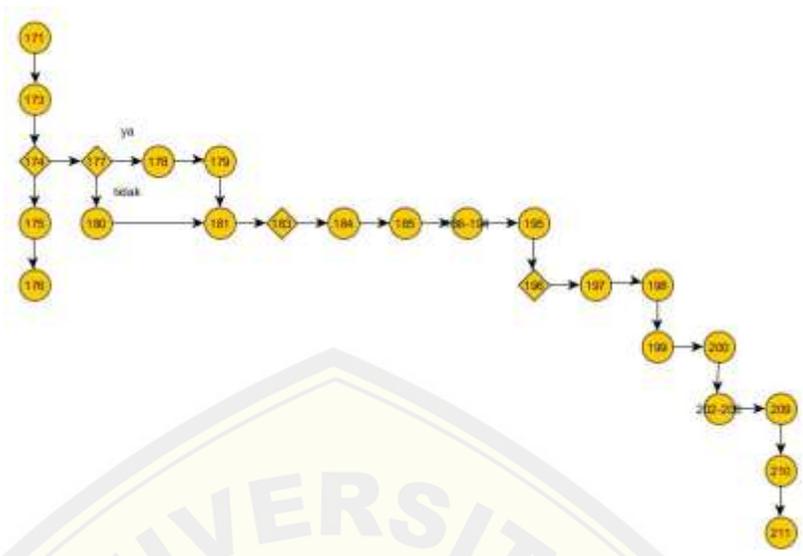
Jalur 1 : 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 10 – 11 – 12

Jalur 2 : 1 – 2 – 3 – 4 – 5 – 6 – 9 – 10 – 11 – 12

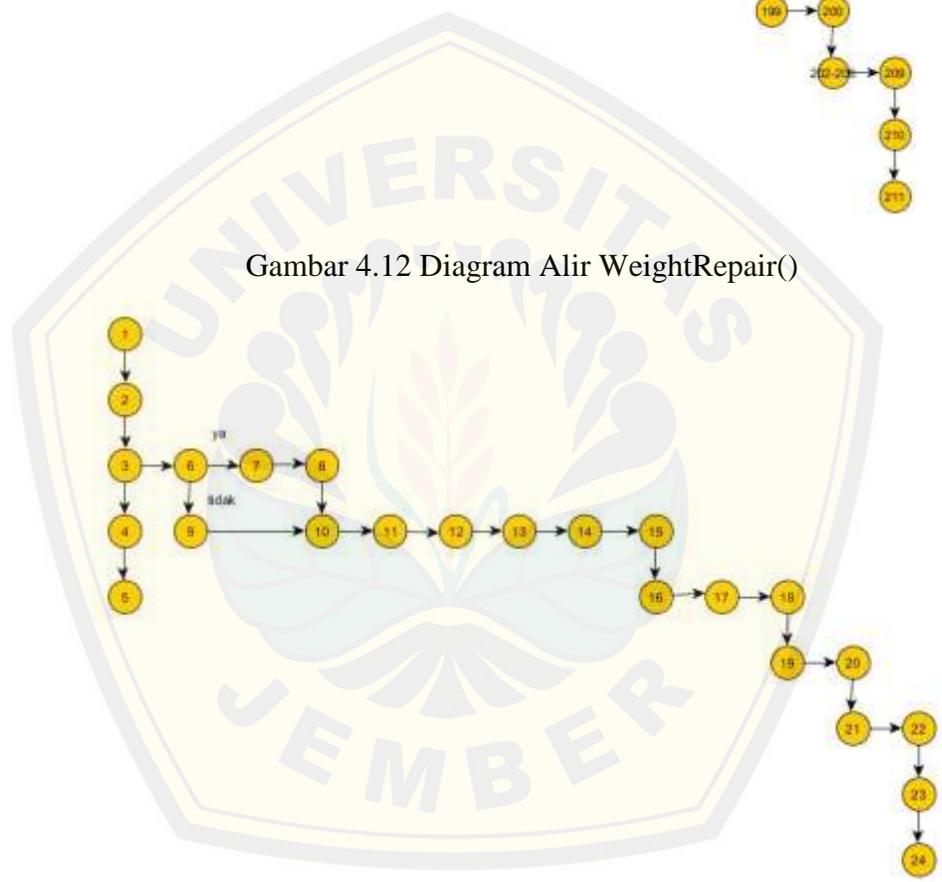
c. *Test Case*

<i>Test case function subval_sort()</i>	
Jalur 1	
<i>Test case</i>	Pengecekan nilai kriteria dari alternatif
Target yang diharapkan	Mengubah isi nilai dari setiap kolom pada Tabel Alternative sesuai dengan <i>input</i> dari <i>user</i>
Hasil pengujian	Benar
Path/Jalur	1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 10 – 11 – 12
Jalur 2	
<i>Test case</i>	Pengecekan nilai setiap kriteria agar nilai sesuai dengan data <i>homestay</i> yang ada di database
Target yang diharapkan	Nilai pada setiap kolom sesuai dengan yang ada di database
Hasil pengujian	Benar
Path/Jalur	1 – 2 – 3 – 4 – 5 – 6 – 9 – 10 – 11 – 12

2. Pengujian Listing Program WeightRepair()



Gambar 4.12 Diagram Alir WeightRepair()



Gambar 4.13 Grafik Alir WeightRepair()

a. Kompleksitas Siklometik

Perhitungan kompleksitas siklometik untuk grafik alir G menggunakan rumus $V(G) = E - N + 2$. E adalah jumlah *edge* dan N adalah jumlah *node* pada grafik alir.

$V(G) = E - N + 2$

$$= 25 - 24 + 2$$

$$= 3$$

b. Basis Set

Jalur 1 : 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 10

Jalur 2 : 1 – 2 – 3 – 4 – 5 – 9 – 10

Jalur 3 : 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 10 – 11 – 12 – 13 – 14 – 15 – 16 – 17 –
18 – 19 – 20 – 21 – 22 – 23 – 24

c. Test Case

<i>Test case function subval_sort()</i>	
Jalur 1	
<i>Test case</i>	Pengecekan status dari setiap kriteria
Target yang diharapkan	Mengubah isi nilai dari setiap kriteria apabila <i>cost</i> bernilai negatif
Hasil pengujian	Benar
Path/Jalur	1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 10
Jalur 2	
<i>Test case</i>	Pengecekan status dari setiap kriteria
Target yang diharapkan	Mengubah isi nilai dari setiap kriteria apabila <i>benefit</i> bernilai positif
Hasil pengujian	Benar
Path/Jalur	1 – 2 – 3 – 4 – 5 – 9 – 10
Jalur 3	
<i>Test case</i>	Penghitungan perbaikan bobot
Target yang diharapkan	Hasil perhitungan perbaikan bobot sesuai dengan nilai <i>cost</i> dan <i>benefit</i>
Hasil pengujian	Benar
Path/Jalur	1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 10 – 11 – 12 – 13 – 14 – 15 – 16 – 17 – 18 – 19 – 20 – 21 – 22 – 23 – 24

b. Black Box Testing

Black box testing merupakan pengujian yang menitik beratkan pada uji fungsionalitas dari program yang dibuat. Hal yang perlu dilakukan dalam pengujian ini adalah menguji *interface* dari program untuk memastikan suatu masukan di proses

oleh sistem dengan benar dan menghasilkan keluaran yang sesuai dengan perancangan. Pengujian dilakukan oleh *user* pada tiap fitur yang ada pada sistem. Dokumen hasil pengujian sistem dapat dilihat pada Lampiran F.

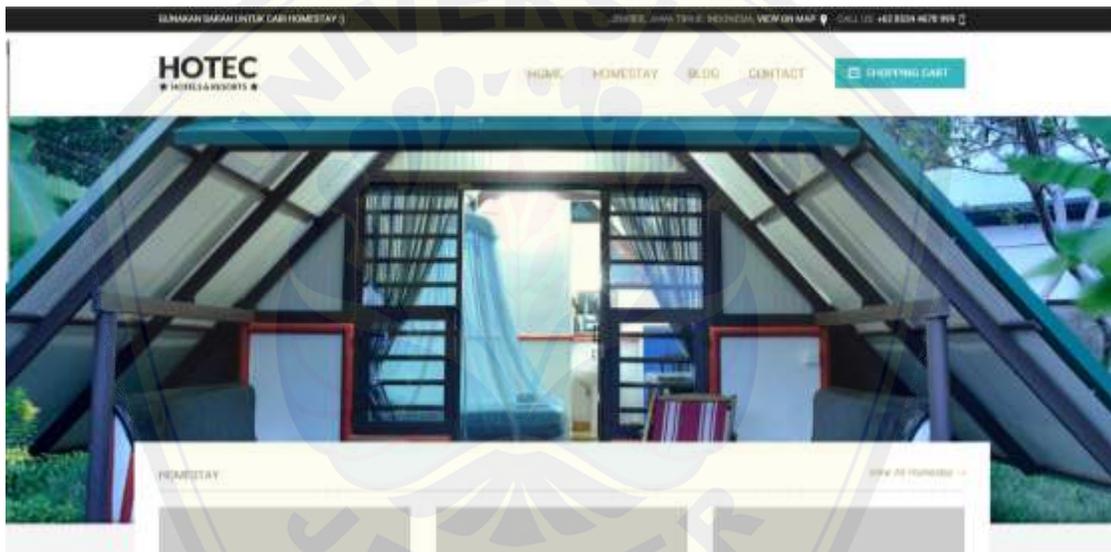
4.6 Evaluasi sistem

Pada tahap terakhir dari pengembangan sistem menggunakan metode prototype adalah evaluasi sistem. Evaluasi sistem dilakukan sesudah sistem selesai dibuat dan telah melalui tahap pengujian terlebih dahulu. Sistem kemudian diserahkan ke pengguna dan sistem masih belum dilakukan evaluasi karena dari pengujian sistem, sistem tidak mengalami kesalahan dan sudah memenuhi kebutuhan. Pihak desa wisata (*user*) setuju dengan sistem informasi dan rekomendasi *homestay* karena sesuai dengan kebutuhan mereka yang membutuhkan media dalam mencatat data *homestay*, melakukan reservasi, dan memberikan kesempatan pada calon wisatawan untuk memilih *homestay* sesuai dengan keinginan masing-masing. Setiap calon wisatawan memiliki keinginan yang berbeda tentang tipe *homestay* sehingga sistem rekomendasi pemilihan *homestay* ini dapat membantu calon wisatawan dalam memperoleh keputusan *homestay* sesuai keinginan mereka. Selain itu, pihak pemilik *homestay* juga dapat mendaftarkan *homestay* yang mereka miliki dengan mudah tanpa harus secara langsung mendatangi pihak desa wisata. Tahapan evaluasi ini akan terus dilakukan karena sewaktu-waktu *user* bisa saja menginginkan penambahan atau perubahan fitur. Seiring dengan penggunaan sistem yang dilakukan oleh *user*, sistem sewaktu-waktu bisa saja menimbulkan bugs sehingga perlu perbaikan.

BAB 5. HASIL DAN PEMBAHASAN

5.1 Hasil Pembuatan Sistem

Sistem informasi dan rekomendasi pemilihan *homestay* memiliki fitur utama yaitu pencarian *homestay* terbaik berdasarkan beberapa kriteria yang telah ditentukan menggunakan analisis metode *Weighted Product* (WP). Sistem ini juga memiliki beberapa fitur pendukung yaitu login, logout, register, manajemen *user* aplikasi, manajemen pemilik *homestay*, manajemen booking *homestay*, dan manajemen *homestay*. Tampilan halaman utama Sistem Informasi dan Rekomendasi Pemilihan *Homestay* dapat dilihat gambar 5.1.



Gambar 5.1. Tampilan Halaman Utama Sistem Informasi dan Rekomendasi Pemilihan *Homestay*

5.1.1 Fitur Rekomendasi *Homestay* Terbaik

Pada fitur ini terdapat proses rekomendasi *homestay* yaitu mencari *homestay* terbaik yang direkomendasikan untuk dipesan berdasarkan kriteria yang sudah dipilih oleh *user*. Proses mencari rekomendasi *homestay* terbaik dilakukan oleh sistem dengan mengeksekusi *input* nilai yang diinginkan dari beberapa kriteria. Tahap *input*

nilai yang diinginkan dari *user* berdasarkan kriteria yang telah ditentukan dapat dilihat pada Gambar 5.2.

The screenshot shows a web application interface for selecting homestay recommendation criteria. The interface is divided into two sections: "Input Tingkat Kepentingan" (Input Importance Level) and "Input Bobot Kepentingan" (Input Weight Importance). Both sections list various criteria with corresponding input fields and priority levels.

Input Tingkat Kepentingan

Kriteria	Nilai	Prioritas
Harga	10000 - 20000	Prioritas 1
	> 10000	Prioritas 2
	< 10000	Prioritas 1
Jarak Pusat Kegiatan	10 - 100	Prioritas 1
	< 10	Prioritas 2
	> 100	Prioritas 1
Jumlah Kamar	3 - 5	Prioritas 1
	< 3	Prioritas 2
	> 5	Prioritas 1
Kapasitas Perakun	20 - 30	Prioritas 1
	< 20	Prioritas 1
	> 30	Prioritas 2
Penyedia ada	Ada	Prioritas 1
	Tidak Ada	Prioritas 2
Penyedia Homestay Profesional	Tidak Ada	Prioritas 1
	Ada	Prioritas 2

Input Bobot Kepentingan

Kriteria	Bobot
Harga	5
Jarak Pusat Kegiatan	5
Jumlah Kamar	5
Kapasitas Perakun	5
Penyedia ada	5
Penyedia Homestay Profesional	5

Gambar 5.2 Tampilan Pemilihan Kriteria Rekomendasi *Homestay*

Tahap selanjutnya setelah *user* menginputkan nilai yang diinginkan dari beberapa kriteria tersebut, sistem menganalisisnya menggunakan metode *Weighted*

Product (WP) untuk memberikan rekomendasi *homestay* terbaik. *Screenshot* rekomendasi *homestay* terbaik yang diberikan sistem dapat dilihat pada Gambar 5.3.



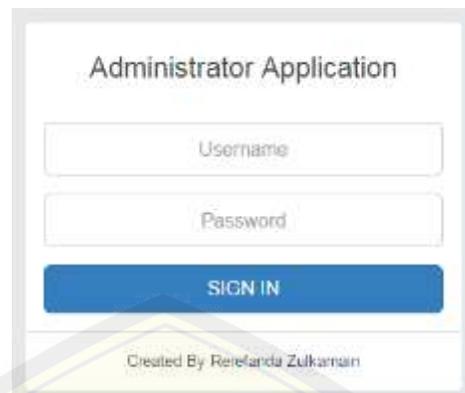
Gambar 5.3 Tampilan Rekomendasi *Homestay* dari Sistem

Pada Gambar 5.3 sistem menampilkan rekomendasi *homestay* terbaik. Sistem memberikan rekomendasi *homestay* yang berada di urutan satu, namun *user* bebas menentukan *homestay* mana yang akan dipilih saat melakukan pemesanan *homestay* nanti.

5.1.3 Fitur *Login*

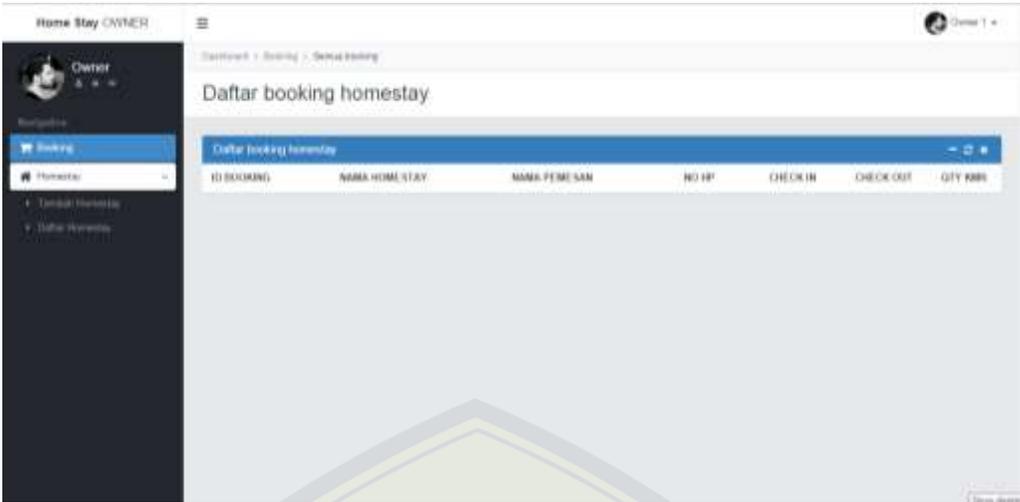
Fitur ini merupakan fitur autentifikasi yang berguna untuk *owner* berikan hak akses kepada *user* terhadap fitur-fitur yang berada di dalam sistem. Dalam sistem ini terdapat empat *user* yaitu *super admin*, *owner*, *user*, dan *admin*. Fitur *login*

digunakan *super admin* dan *admin* serta *owner* yang telah melakukan registrasi terhadap sistem. Tampilan *login* dapat dilihat pada Gambar 5.4

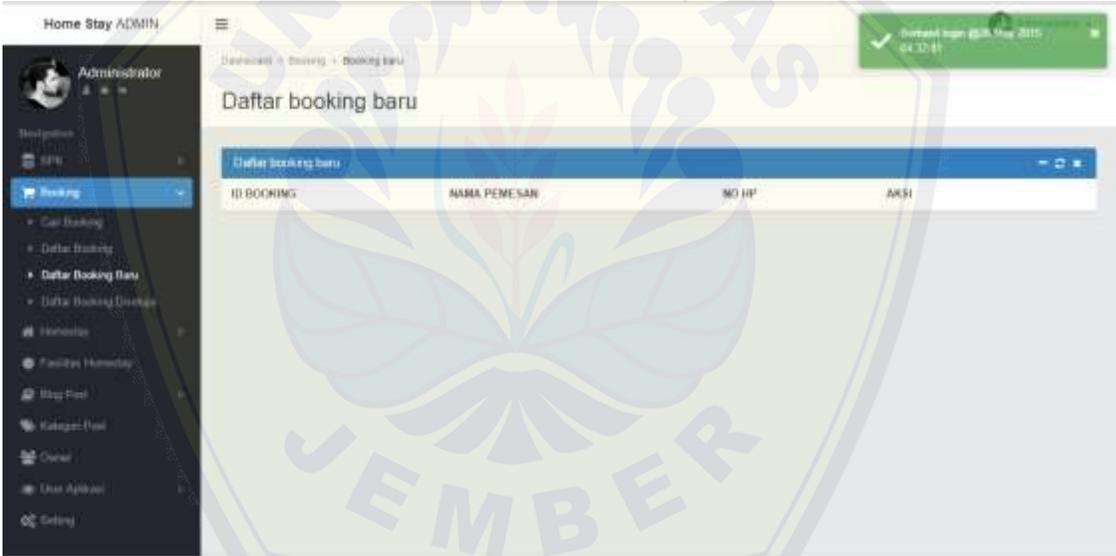


Gambar 5.4. Tampilan *Login*

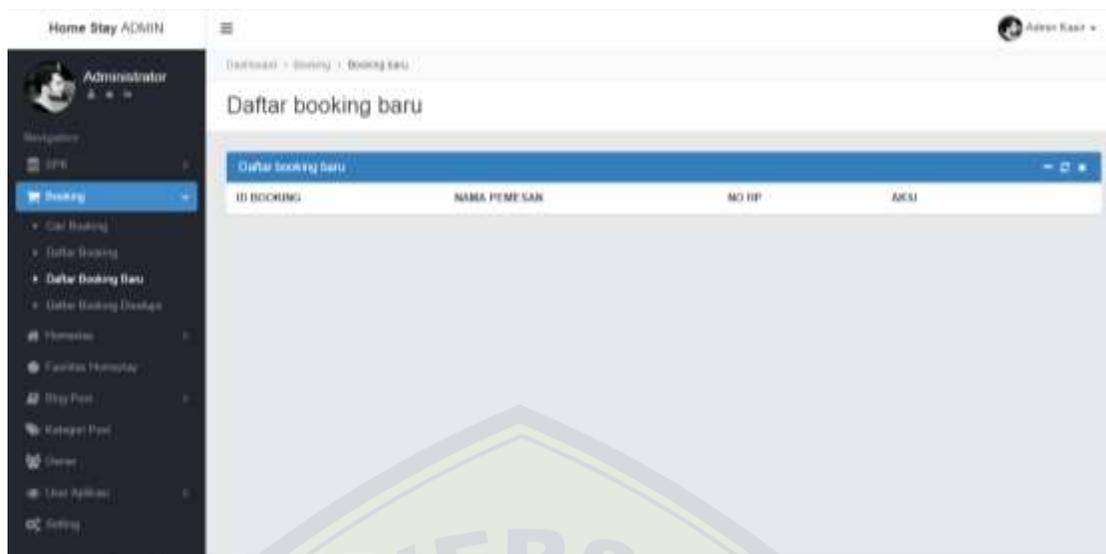
Pada Gambar 5.4 *user* harus menginputkan *username* dan *password* untuk bisa mengakses fitur-fitur tambahan yang dimiliki oleh sistem informasi reklame insidental. *User* yang melakukan *login* yaitu *owner*, *super admin*, dan *admin* akan di *redirect* ke *homepage* masing-masing. Jika *owner* yang melakukan *login* maka akan di *redirect* ke halaman *homepage owner*, sedangkan jika *super admin* yang melakukan *login* maka akan di *redirect* ke *homepage super admin*, begitu pun jika *admin* yang melakukan *login* maka akan di *redirect* ke *homepage admin*. Tampilan *homepage owner*, *homepage super admin* dan *homepage admin* dapat dilihat pada Gambar 5.5, Gambar 5.6. dan Gambar 5.7. Apabila *admin*, *super admin*, dan *owner* salah dalam *input username* dan *password* maka tidak dapat *login* ke dalam sistem informasi da rekomendasi pemilihan *homestay*. Setiap *user* memiliki fitur yang berbeda sesuai dengan kebutuhan dalam sistem tersebut. Menu *logout* pada masing-masing *homepage admin*, *super admin*, dan *owner* berfungsi untuk *user* keluar dari sistem dan *user* tidak dapat mengakses fitur-fitur sistem seperti pada waktu *login*.



Gambar 5.5 Tampilan Homepage Owner



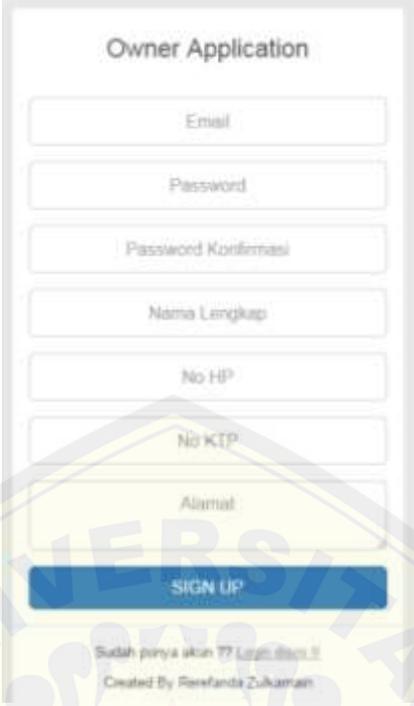
Gambar 5.6 Tampilan Homepage Super Admin



Gambar 5.7 Tampilan *Homepage Admin*

5.1.4 Fitur Registrasi

Fitur ini digunakan oleh *owner homestay* untuk membuat akun pribadi agar dapat melakukan *login* ke dalam sistem ini. Pemilik homestay harus mengisi secara lengkap form Owner Application. Field yang harus diisi antara lain alamat *email*, *password*, nama lengkap, nomor *handphone*, nomor KTP, dan alamat pemilik homestay tersebut tinggal. Pemilik homestay memilih *Sign Up* untuk menyimpan data tersebut ke dalam database. Setelah mendaftar, pemilik *homestay* menunggu super admin untuk menerima pendaftaran tersebut. Ketika super admin telah mengkonfirmasi, pemilik *homestay* dapat login ke akun pribadinya. Pada akun tersebut, pemilik *homestay* dapat mengelola data *homestay* yang dimiliki seperti tambah, ubah, hapus dan unggah foto *homestay* yang dimiliki. Namun apabila super admin belum menerima permintaan registrasi dari pemilik *homestay*, maka pemilik *homestay* belum bisa *login* ke sistem informasi dan rekomendasi pemilihan *homestay*. Tampilan halaman registrasi pemilik *homestay* dapat dilihat pada Gambar 5.8.



Owner Application

Email

Password

Password Konfirmasi

Nama Lengkap

No HP

No KTP

Alamat

SIGN UP

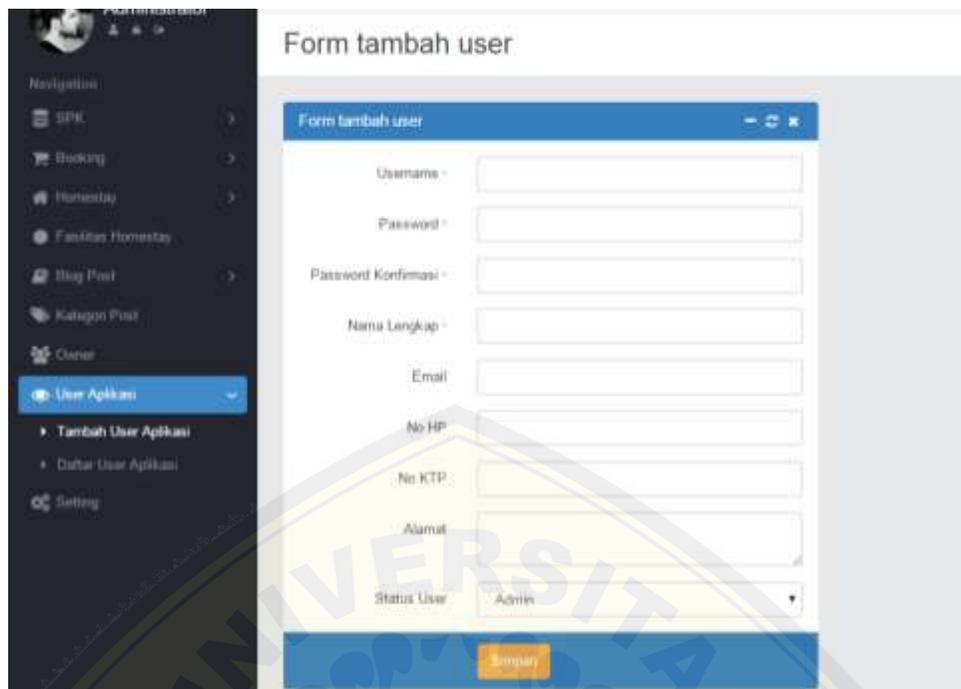
Sudah punya akun?? Login disini!!
Created By Rinfanda Zukaman

Gambar 5.8 Tampilan Halaman Registrasi Owner *Homestay*

5.1.5 Fitur Manajemen *User Aplikasi*

5.1.5.1 Tambah *User Aplikasi*

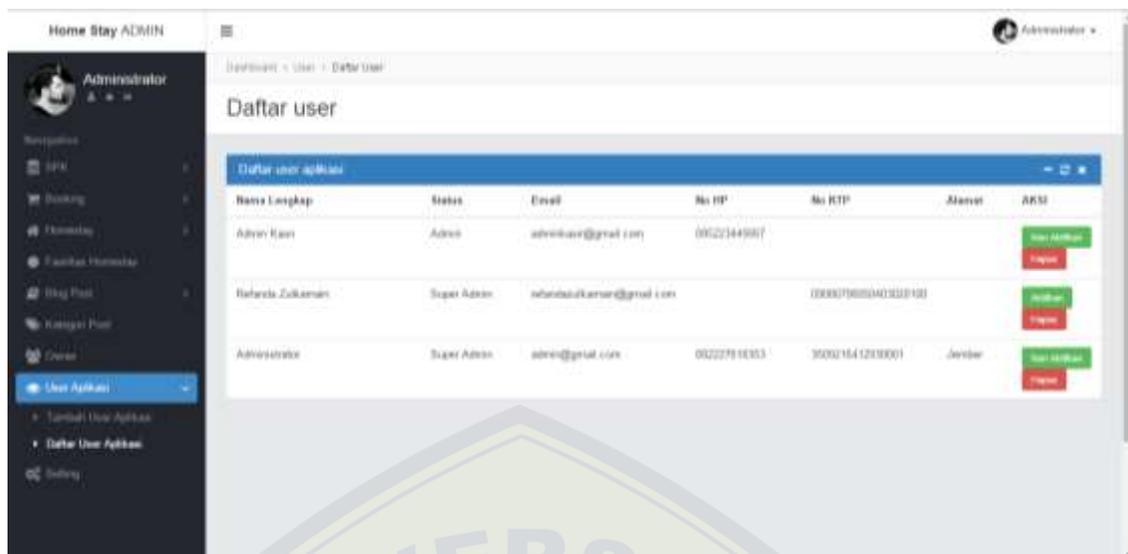
Pada fitur ini *user* yaitu *super admin* dapat menambah data *user* aplikasi baru. Fitur ini digunakan agar *user* yang telah terdaftar dapat memiliki hak akses terhadap sistem informasi dan rekomendasi pemilihan *homestay*. Tahap mengisi Form tambah *user* yaitu super admin mengisi data user secara lengkap antara lain username, *password*, nama lengkap, alamat *email*, nomor *handphone*, nomor KTP, alamat dan yang terakhir super admin memilih status dari *user* tersebut. Setelah Form tambah user telah terisi secara lengkap, super admin memilih Simpan untuk menyimpan data tersebut ke dalam database. Tampilan fitur tambah *user* aplikasi dapat dilihat Gambar 5.9.

The image shows a mobile application interface for adding a user. On the left is a dark navigation menu with options like 'SPK', 'Booking', 'Homestay', 'Facilities Homestay', 'Blog Post', 'Kategori Post', 'User', 'User Aplikasi', 'Tambah User Aplikasi', 'Daftar User Aplikasi', and 'Setting'. The 'User Aplikasi' option is selected. The main content area is titled 'Form tambah user' and contains a form with the following fields: Username, Password, Password Konfirmasi, Nama Lengkap, Email, No HP, No KTP, and Alamat. Below these fields is a dropdown menu for 'Status User' with 'Admin' selected. At the bottom of the form is an orange 'Simpan' button.

Gambar 5.9 Tampilan Halaman Tambah *User* Aplikasi

5.1.5.2 Daftar *User* Aplikasi

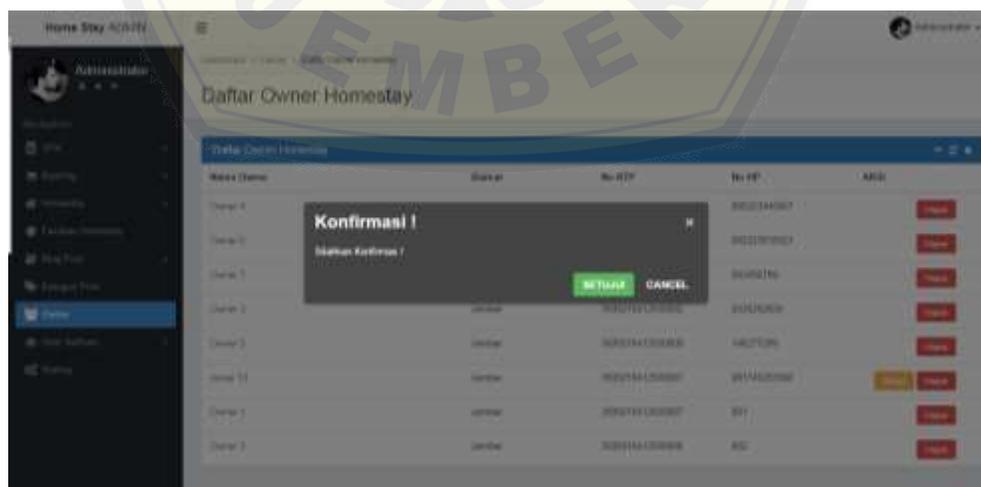
Pada fitur ini *user* yaitu *super admin* dapat melihat daftar *user* aplikasi sekaligus dapat mengubah status dari *user* yaitu aktif atau non-aktif. *User* yang telah tidak aktif sebagai admin maka *super admin* memilih pilihan non aktifkan. Apabila *user* tersebut kembali aktif menjadi admin maka status akan diubah oleh *super admin* dengan cara memilih pilihan aktifkan. Fitur ini memberikan fasilitas kepada *user* admin untuk lebih mudah dalam melihat daftar *user* sistem informasi dan rekomendasi pemilihan *homestay*. Data lengkap *user* berupa nama, status, alamat *email*, nomor *handphone*, nomor KTP, dan alamat tercatat secara lengkap di Tabel Daftar *user*. Selain itu, *super admin* juga dapat menghapus data dari *user* aplikasi. Tampilan fitur daftar *user* aplikasi dapat dilihat Gambar 5.10.



Gambar 5.10 Tampilan Halaman Daftar *User* Aplikasi

5.1.6 Fitur Manajemen Pemilik *Homestay*

Pada fitur ini *user* yaitu *super admin* dapat melihat daftar pemilik *homestay* sekaligus dapat mengubah status dari *user* setelah super admin memilih konfirmasi maka pemilik *homestay* yang baru daftar mendapat persetujuan untuk masuk ke sistem pemilik *homestay*. Setelah muncul pilihan konfirmasi, super admin dapat memilih setuju untuk menyetujui owner baru dan memilih cancel untuk menutup pilihan konfirmasi. Selain itu, super admin juga dapat menghapus data dari owner. Tampilan fitur manajemen pemilik *homestay* dapat dilihat Gambar 5.11.

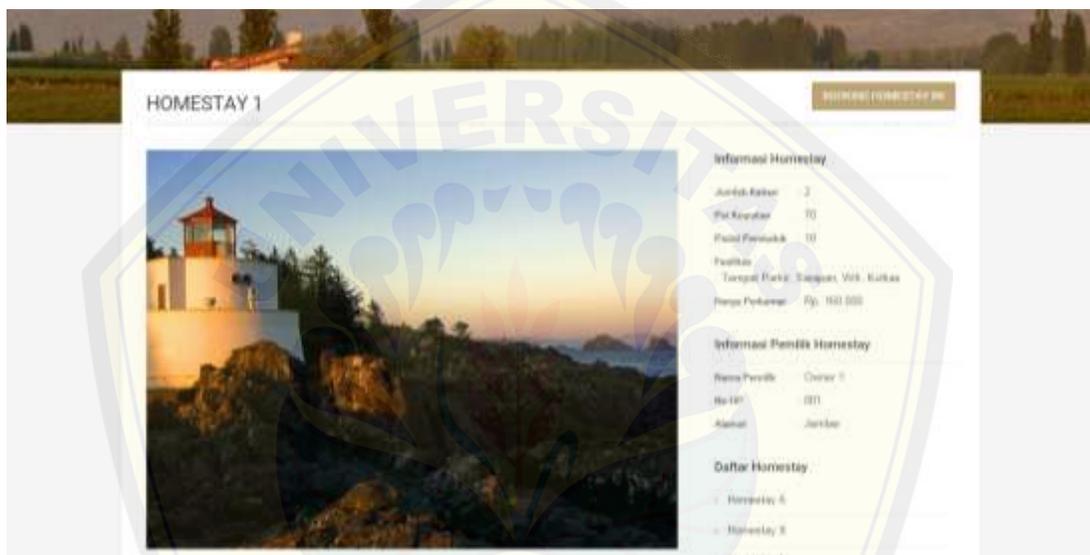


Gambar 5.11 Tampilan Halaman Owner

5.1.7 Fitur Manajemen Booking *Homestay*

5.1.7.1 Booking *Homestay*

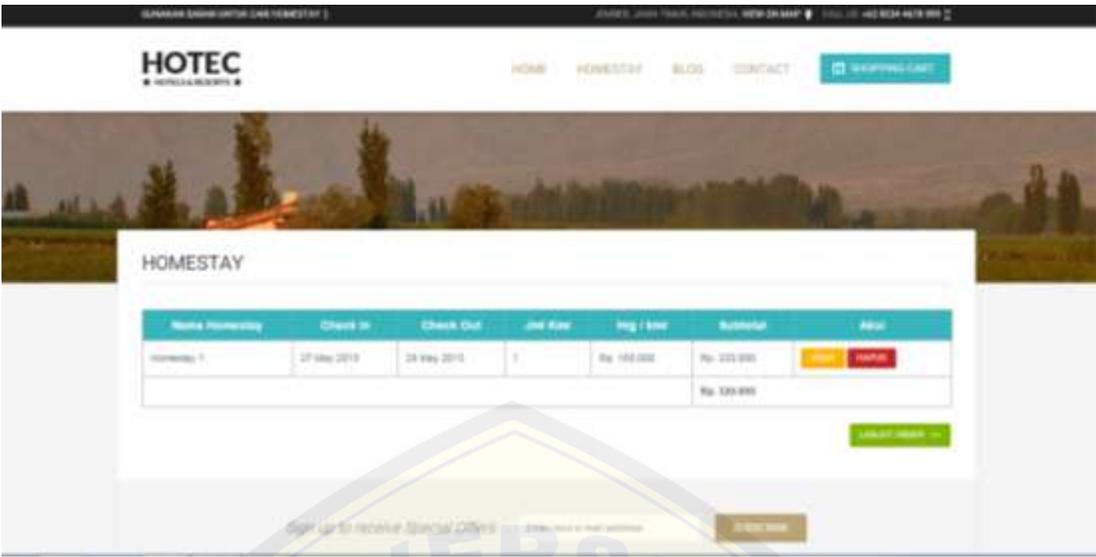
Pada fitur ini *user* yaitu calon wisatawan yang akan memesan *homestay* dapat melihat daftar *homestay* yang tersedia sekaligus dapat melihat detail dari *homestay* yang dipilih. Setelah itu, *user* dapat memesan *homestay* yang dipilih tersebut dengan mengisi form booking dan form order. Tampilan fitur booking *homestay* dapat dilihat Gambar 5.12.



Gambar 5.12 Tampilan Halaman Owner

5.1.7.2 Ubah Booking *Homestay*

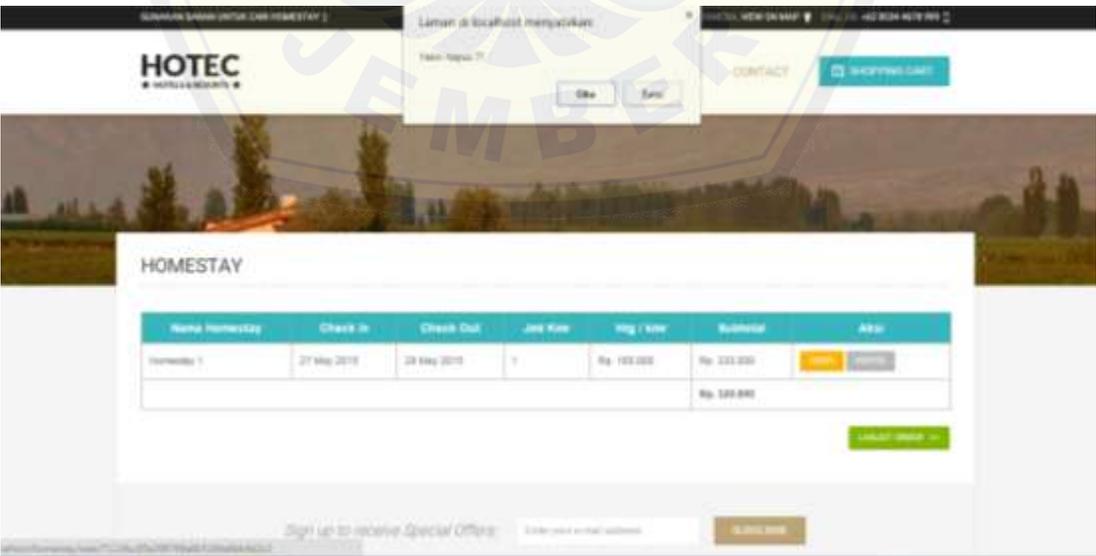
Pada fitur ini *user* yaitu calon wisatawan yang telah memesan *homestay* dapat merubah data *homestay* yang telah dipesan yaitu dengan klik pilihan ubah pada field aksi kemudian *user* mengedit isi dari form booking. Tampilan fitur ubah booking *homestay* dapat dilihat Gambar 5.13.



Gambar 5.13 Tampilan Halaman Owner

5.1.7.3 Hapus Booking Homestay

Pada fitur ini *user* yaitu calon wisatawan yang telah memesan *homestay* dapat menghapus data *homestay* yang telah dipesan yaitu dengan klik pilihan hapus pada field aksi kemudian sistem akan memunculkan pilihan konfirmasi untuk menghapus. Jika memilih Oke, maka data *homestay* akan terhapus. Sebaliknya jika memilih batal maka akan batal menghapus. Tampilan fitur hapus booking *homestay* dapat dilihat Gambar 5.14.

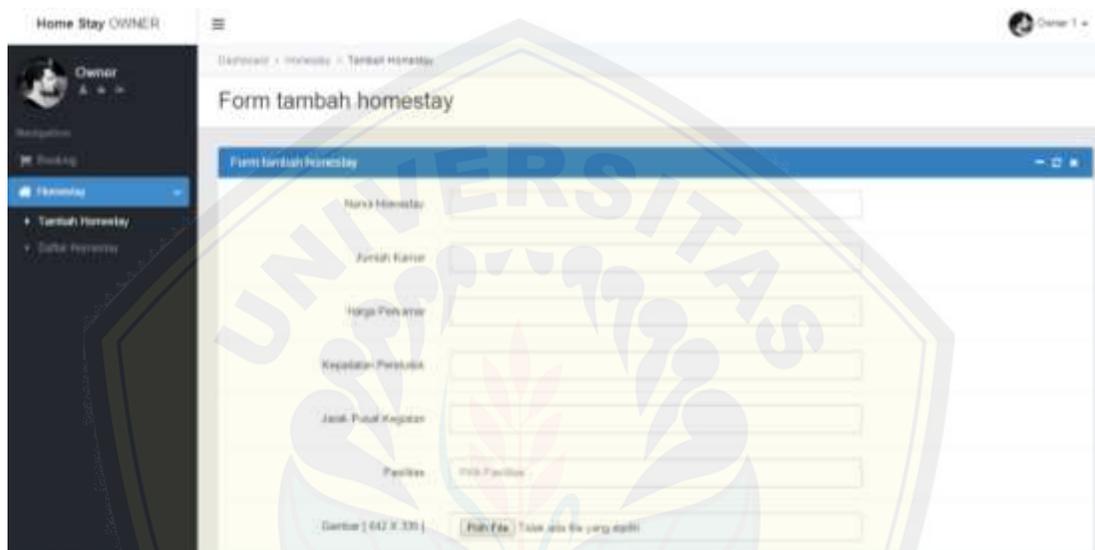


Gambar 5.14 Tampilan Halaman Booking Homestay

5.1.8 Fitur Manajemen Daftar *Homestay*

5.1.8.1 Tambah *Homestay*

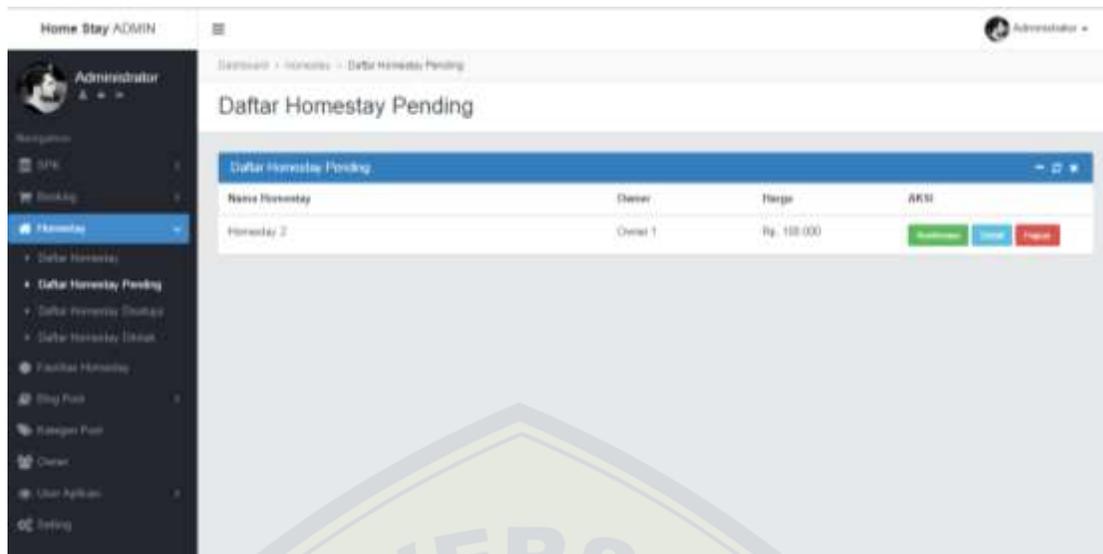
Pada fitur ini *user* yaitu *owner* dapat menambah data *homestay* baru. Setelah mendaftar, *owner* menunggu persetujuan dari super admin untuk mengkonfirmasi *homestay* yang telah didaftarkan. Tampilan fitur tambah *homestay* dapat dilihat Gambar 5.15.



Gambar 5.15 Tampilan Halaman Tambah *Homestay*

5.1.8.2 Accept Daftar *Homestay* Pending

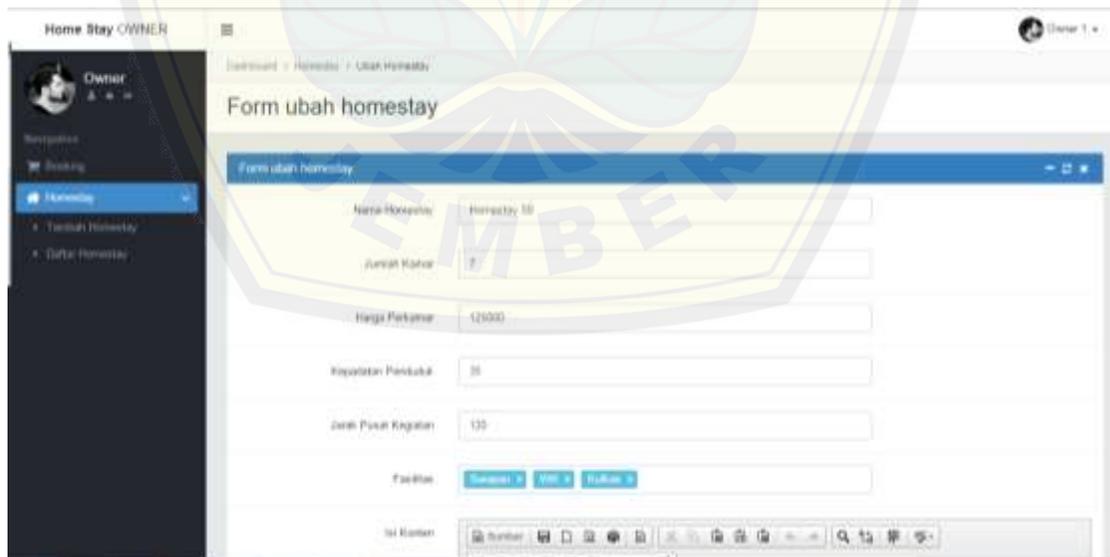
Pada fitur ini *user* yaitu *super admin* dapat melakukan konfirmasi *homestay*. Terdapat pilihan konfirmasi di field aksi. Jika admin menerima maka *homestay* tersebut tampil di daftar *homestay* yang dapat dipesan oleh calon wisatawan. Namun apabila masih berstatus *pending*, maka *homestay* tersebut belum dapat tampil di halaman sistem informasi dan rekomendasi *homestay* serta belum dapat dipesan oleh calon wisatawan. Tampilan fitur accept daftar *homestay* pending dapat dilihat Gambar 5.16.



Gambar 5.16 Tampilan Halaman Daftar *Homestay* Pending

5.1.8.3 Ubah Daftar *Homestay*

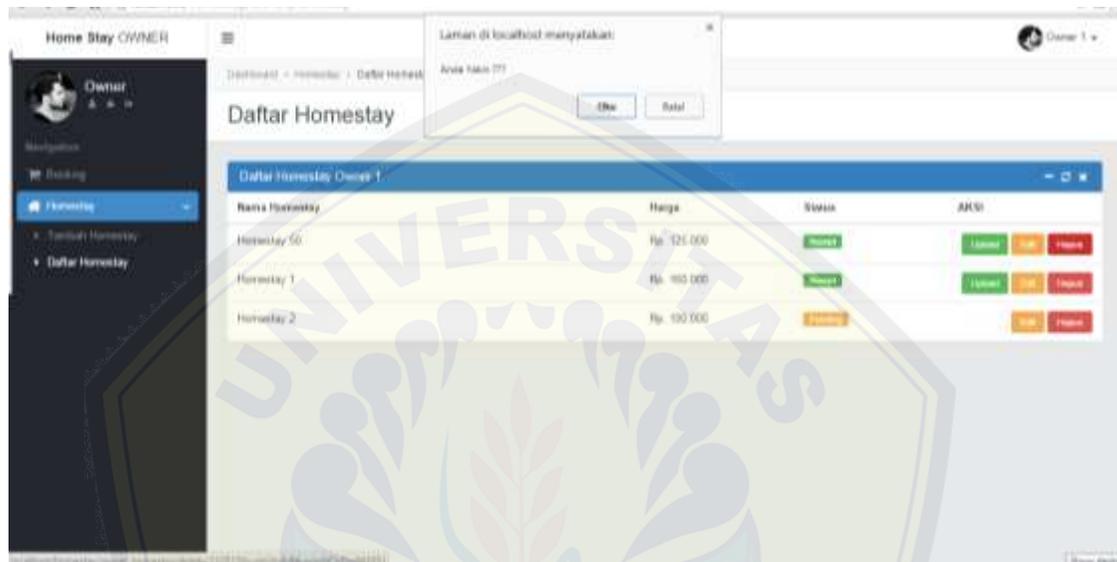
Pada fitur ini *user* yaitu owner yang telah mendaftarkan *homestay* yang dimiliki dapat merubah data *homestay* yaitu dengan klik pilihan edit pada field aksi kemudian *user* mengedit isi dari form ubah *homestay*. Tampilan fitur ubah *homestay* dapat dilihat Gambar 5.17.



Gambar 5.17 Tampilan Halaman Edit Daftar *Homestay*

5.1.8.4 Hapus Daftar *Homestay*

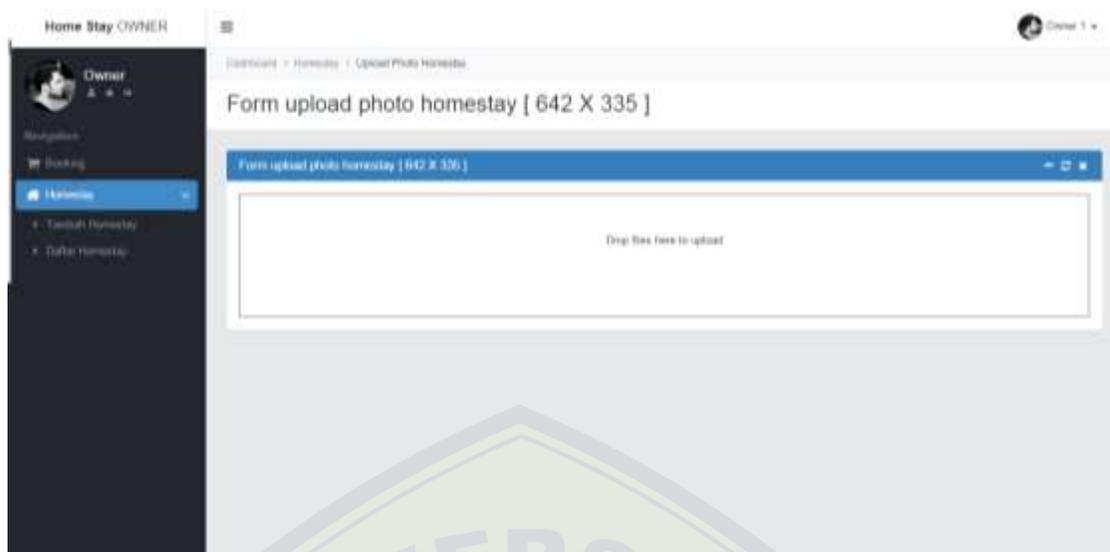
Pada fitur ini *user* yaitu owner dapat menghapus data *homestay* yang telah didaftarkan yaitu dengan klik pilihan hapus pada field aksi kemudian sistem akan memunculkan pilihan konfirmasi untuk menghapus. Jika memilih Oke, maka data *homestay* akan terhapus. Sebaliknya jika memilih batal maka akan batal menghapus. Tampilan fitur hapus *homestay* dapat dilihat Gambar 5.18.



Gambar 5.18 Tampilan Halaman Hapus Daftar *Homestay*

5.1.8.5 Upload Photo *Homestay*

Pada fitur ini *user* yaitu owner dapat mengunggah foto *homestay* yang telah didaftarkan yaitu dengan klik pilihan upload pada field aksi kemudian sistem akan memunculkan form upload photo *homestay*. Setelah itu, pemilik *homestay* dapat mendrag atau menginputkan foto *homestay* ke dalam *form* tersebut. Foto yang telah terupload masih bisa diubah dengan cara menghapusnya terlebih dahulu. Tampilan fitur hapus *homestay* dapat dilihat Gambar 5.19.

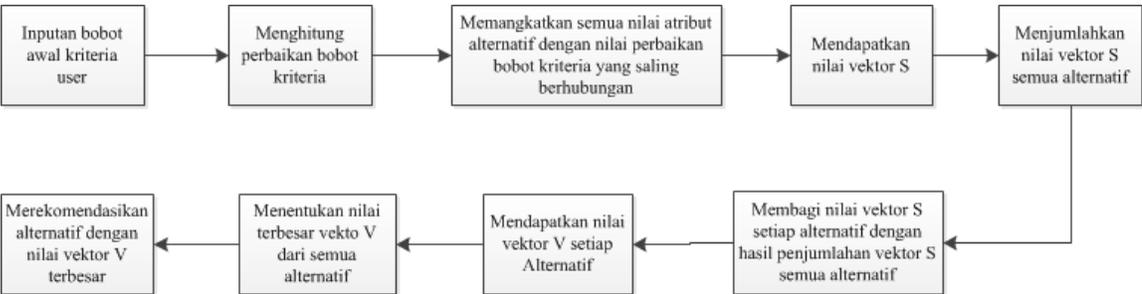


Gambar 5.19 Tampilan Halaman Upload Photo *Homestay*

5.2 Hasil Pengembangan WP

Pada sub bab ini akan dibahas hasil dari pengembangan Sistem Informasi dan Rekomendasi Pemilihan *Homestay*, apakah penelitian sudah sesuai dengan tujuan penelitian atau belum. Dalam hal ini, yang akan dibahas yaitu pencarian *homestay* terbaik menggunakan metode *Weighted Product* (WP).

Pencarian *homestay* terbaik menggunakan analisis metode *Weighted Product* (WP) dilakukan dengan menentukan kriteria bernilai *benefit* dan kriteria *cost* dahulu kemudian menghitung nilai perbaikan bobot kriteria yang diinputkan oleh *user* dan memangkatkan nilai rating setiap atribut alternatif dengan nilai perbaikan bobot tersebut guna mendapatkan nilai vektor *S* dari setiap alternatif. Selanjutnya membagi nilai vektor *S* setiap alternatif dengan hasil penjumlahan nilai vektor *S* semua alternatif guna mendapatkan nilai vektor *V*. Nilai vektor *V* terbesar dari semua alternatif merupakan dasar yang digunakan untuk merekomendasikan alternatif terbaik oleh sistem kepada *user*. Penjelasan dalam diagram proses pencarian *homestay* terbaik menggunakan analisis metode *Weighted Product* (WP) dapat dilihat pada Gambar 5.20.



Gambar 5.20 Proses Pencarian *Homestay* Terbaik metode *Weighted Product* (WP)

Pencarian *homestay* terbaik pada sistem dilakukan dengan menginputkan kriteria yang diinginkan oleh *user* pada halaman saran pencarian *homestay*. *User* harus mengisi form kriteria yang telah disediakan seperti pada Gambar 5.21.

Kategori	Nilai	Pondasi
Harga	10000 - 15000	Pondasi 1
	15000 - 20000	Pondasi 2
	20000 - 25000	Pondasi 3
Jarak Pusat Kegiatan	0-100	Pondasi 1
	100-200	Pondasi 2
	200-300	Pondasi 3
Jumlah Kamar	1-2	Pondasi 1
	3-4	Pondasi 2
	5-6	Pondasi 3
Kapasitas Pembantu	20-30	Pondasi 1
	30-40	Pondasi 2
	40-50	Pondasi 3
Fasilitas Wifi	Ada	Pondasi 1
	Tidak Ada	Pondasi 2
	Tidak Ada	Pondasi 3
Fasilitas Hewan Peliharaan	Tidak Ada	Pondasi 1
	Ada	Pondasi 2

Input Bobot Kepentingan (Ubah nilai bobot jika perlu)

Harga: 1

Jarak Pusat Kegiatan: 1

Jumlah Kamar: 1

Kapasitas Penduduk: 1

Fasilitas WiFi: 1

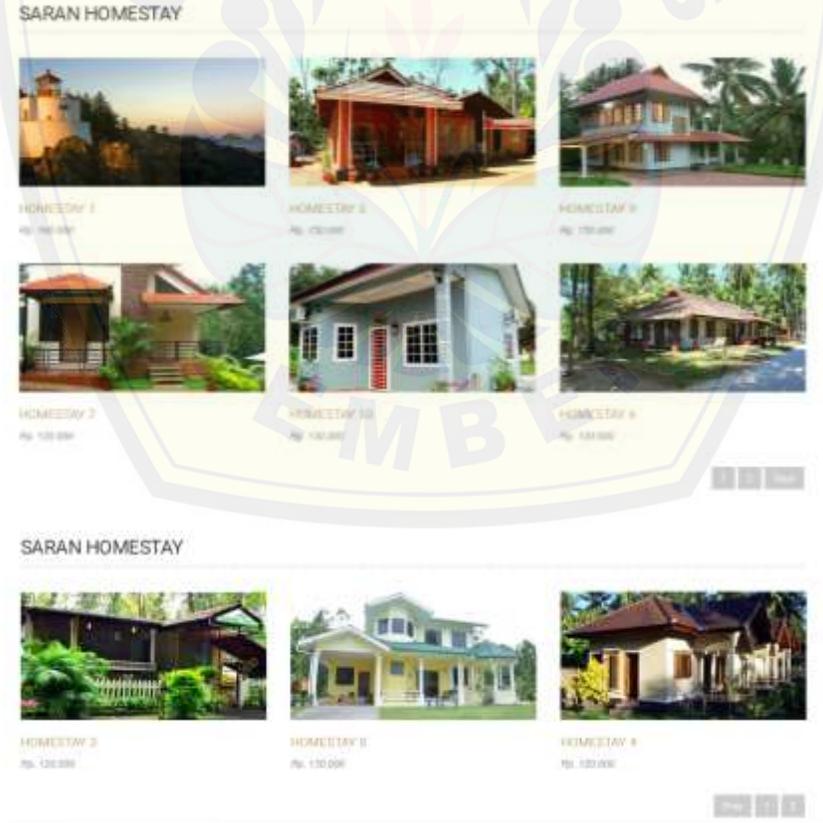
Fasilitas Hewan Peliharaan: 1

HANYA NALA

SUBMIT

Gambar 5.21 Tampilan Menginputkan Nilai Bobot Kriteria pada Rekomendasi *Homestay*

Pada Gambar 5.21 dapat dilihat kriteria telah ditentukan dan semua *form* sudah diisi, maka *user* akan melakukan pencarian *homestay* terbaik dengan melakukan klik submit now. Pencarian lokasi terbaik akan menghasilkan rekomendasi *homestay* terbaik sesuai dengan kriteria *user* seperti pada Gambar 5.21.



Gambar 5.22 Tampilan Hasil Rekomendasi *Homestay* Menggunakan Metode *Weighted Product*

Perhitungan pencarian *homestay* sehingga sistem memberikan rekomendasi *homestay* seperti pada Gambar 5.22 dilakukan dengan menggunakan beberapa *function*, karena adanya beberapa tahapan yang harus sesuai dengan analisis metode *Weighted Product* (WP). *Function* yang digunakan yaitu *function* *GreatAlternative()*, *function* *ConvertAlternatives()*, *function* *ConvertAlternativesValue(\$key,\$oriValue)*, *function* *WeightRepair()*, *function* *CountVector()*, *function* *CountRank()*, *function* *Greate()*. *Listing* program dari beberapa *function* yang digunakan dapat dilihat di Lampiran D.

Hitungan metode *Weighted Product* (WP) pada sistem dapat dilihat pada Gambar 5.23.

Data Alternatif

ID Homestay	Nama Homestay	Harga	Jarak Pusat Kegiatan	Jumlah Kamar	Kapasitas Peselubuh	Fasilitas Wifi	Fasilitas Pets
260413Fdx22d11C%aBRy5lwWE129829	Homestay 1	100000	70	2	50	Ada	Tidak Ada
260413VXU4u1U4GfPn0lBryJ7081137	Homestay 2	120000	100	6	50	Tidak Ada	Tidak Ada
060513E0169wEMaI061dDTWk7G051100	Homestay 4	120000	90	4	50	Tidak Ada	Tidak Ada
060513dFFt4wPwBQvW7HDnS2031232	Homestay 5	120000	100	8	50	Ada	Tidak Ada
0605132EAskbry6LJmOdkKerw53325	Homestay 6	120000	800	4	50	Ada	Tidak Ada
0605131zZonV5ZwWdLWwPAA1D051502	Homestay 7	120000	170	4	50	Tidak Ada	Tidak Ada
060513aYDQpB7aXwbcnduak05337	Homestay 8	150000	170	4	50	Ada	Tidak Ada
060513k1mdF7yWbcT7U0eL0Pg0051742	Homestay 9	150000	400	6	50	Ada	Tidak Ada
060513Vn0JHTP4fco9EaWxOr051819	Homestay 10	150000	600	4	50	Ada	Tidak Ada

Atas Kriteria

Range	Nilai
100000 - 120000	3
= 100000	2
= 120000	1

Range	Nilai
= 100	3
50 - 100	2
= 50	1

Range	Nilai
= 3	3
= 2	2
1 > 5	1

Range	Nilai
20 - 50	3
< 20	2
= 50	1

Range	Nilai
Ada	3
Tidak Ada	1

Range	Nilai
Tidak Ada	3
Ada	1

Hasil Koneksi Data Alternatif

ID Homestay	Nama Homestay	Harga	Jarak Pusat Kegiatan	Jumlah Kamar	Kapasitas Peselubuh	Fasilitas Wifi	Fasilitas Pets
260413Fdx22d11C%aBRy5lwWE129829	Homestay 1	1	3	5	2	5	5
260413VXU4u1U4GfPn0lBryJ7081137	Homestay 2	3	3	3	5	1	5
060513E0169wEMaI061dDTWk7G051100	Homestay 4	3	3	1	3	1	5
060513dFFt4wPwBQvW7HDnS2031232	Homestay 5	3	3	3	3	3	3
0605132EAskbry6LJmOdkKerw53325	Homestay 6	3	3	1	3	3	3
0605131zZonV5ZwWdLWwPAA1D051502	Homestay 7	3	3	1	1	1	3
060513aYDQpB7aXwbcnduak05337	Homestay 8	3	3	1	5	3	3
060513k1mdF7yWbcT7U0eL0Pg0051742	Homestay 9	3	3	3	3	3	3
060513Vn0JHTP4fco9EaWxOr051819	Homestay 10	3	3	1	3	3	3

Kriteria	Perbaikan Bobot	Hasil Perbaikan Bobot	Status	Frugkat
Fasilitas Hewan Peliharaan	$3 / (3 + 2 + 5 + 2 + 3 + 3)$	0.1666666666667	Bersifat	0.1666666666667
Fasilitas Wifi	$2 / (3 + 2 + 5 + 2 + 3 + 3)$	0.1111111111111	Bersifat	0.1111111111111
Harga	$3 / (3 + 2 + 3 + 2 + 3 + 3)$	0.2777777777778	Cost	-0.2777777777778
Jarak Pusat Kegiatan	$2 / (3 + 2 + 3 + 2 + 3 + 3)$	0.1111111111111	Cost	-0.1111111111111
Jumlah Kamar	$3 / (3 + 2 + 5 + 2 + 3 + 3)$	0.1666666666667	Bersifat	0.1666666666667
Kepribada Pemilik	$3 / (3 + 2 + 3 + 2 + 3 + 3)$	0.1666666666667	Cost	-0.1666666666667

Menghitung Vector S

Si	Perhitungan	Hasil
S1	$(3^0 \cdot 0.1666666666667) \cdot (5^0 \cdot 0.1111111111111) \cdot (3^0 \cdot 0.2777777777778) \cdot (3^0 \cdot 0.1111111111111) \cdot (5^0 \cdot 0.1666666666667) \cdot (3^0 \cdot 0.1666666666667)$	1.50702
S2	$(3^0 \cdot 0.1666666666667) \cdot (1^0 \cdot 0.1111111111111) \cdot (3^0 \cdot 0.2777777777778) \cdot (3^0 \cdot 0.1111111111111) \cdot (3^0 \cdot 0.1666666666667) \cdot (3^0 \cdot 0.1666666666667)$	0.67977
S3	$(3^0 \cdot 0.1666666666667) \cdot (1^0 \cdot 0.1111111111111) \cdot (3^0 \cdot 0.2777777777778) \cdot (3^0 \cdot 0.1111111111111) \cdot (1^0 \cdot 0.1666666666667) \cdot (3^0 \cdot 0.1666666666667)$	0.61632
S4	$(3^0 \cdot 0.1666666666667) \cdot (5^0 \cdot 0.1111111111111) \cdot (3^0 \cdot 0.2777777777778) \cdot (3^0 \cdot 0.1111111111111) \cdot (5^0 \cdot 0.1666666666667) \cdot (2^0 \cdot 0.1666666666667)$	0.38709
S5	$(3^0 \cdot 0.1666666666667) \cdot (5^0 \cdot 0.1111111111111) \cdot (3^0 \cdot 0.2777777777778) \cdot (3^0 \cdot 0.1111111111111) \cdot (1^0 \cdot 0.1666666666667) \cdot (3^0 \cdot 0.1666666666667)$	0.09653
S6	$(3^0 \cdot 0.1666666666667) \cdot (1^0 \cdot 0.1111111111111) \cdot (3^0 \cdot 0.2777777777778) \cdot (3^0 \cdot 0.1111111111111) \cdot (1^0 \cdot 0.1666666666667) \cdot (1^0 \cdot 0.1666666666667)$	0.60932
S7	$(3^0 \cdot 0.1666666666667) \cdot (5^0 \cdot 0.1111111111111) \cdot (3^0 \cdot 0.2777777777778) \cdot (3^0 \cdot 0.1111111111111) \cdot (1^0 \cdot 0.1666666666667) \cdot (5^0 \cdot 0.1666666666667)$	0.5375
S8	$(3^0 \cdot 0.1666666666667) \cdot (5^0 \cdot 0.1111111111111) \cdot (3^0 \cdot 0.2777777777778) \cdot (3^0 \cdot 0.1111111111111) \cdot (3^0 \cdot 0.1666666666667) \cdot (3^0 \cdot 0.1666666666667)$	0.31625
S9	$(3^0 \cdot 0.1666666666667) \cdot (5^0 \cdot 0.1111111111111) \cdot (3^0 \cdot 0.2777777777778) \cdot (3^0 \cdot 0.1111111111111) \cdot (1^0 \cdot 0.1666666666667) \cdot (3^0 \cdot 0.1666666666667)$	0.89633

Preferensi (Vi) dengan id key sorting descending

Vi	Nilai V	ID KEY	Nama Homestay
V1	0.5077	290415F4b254b1c1b2d8f5f0a5e12829	Homestay 1
V4	0.12198	060513d4fFdaPobQbWdH0aNs203123	Homestay 8
V6	0.11525	060513b3JmdV5N8d3TU0e10p051142	Homestay 9
V6	0.09638	060513a2ZmV32bWdLWW9Aa1D05102	Homestay 7
V9	0.09597	0605135a0JHTF4ao9EaWNe0q53E19	Homestay 10
V1	0.09597	0605131EAsb3v4LJmd0dKce291333	Homestay 6
V2	0.09368	2604135XU7a4L4G27NcDbr32983137	Homestay 3
V7	0.08814	060513aYDQgHT0XoJm0a5f051337	Homestay 5
V3	0.08404	060513E01m0dEkaK0EaDTWK7G051180	Homestay 4

Gambar 5.23 Tampilan Hasil Hitung pada Sistem

Pencarian lokasi yang dilakukan oleh sistem sesuai dengan kriteria yang diinputkan oleh user pada Gambar 5.21 akan menghasilkan rekomendasi homestay seperti pada Gambar 5.22. Hasil pencarian homestay terbaik yang dilakukan oleh sistem menggunakan metode *Weighted Product* (WP) juga dihitung secara manual guna benar tidaknya perhitungan yang telah dilakukan oleh sistem. Perhitungan secara manual menggunakan beberapa rumus berikut ini:

1. Menghitung nilai perbaikan bobot user menggunakan rumus 5.1.

$$W_j = \frac{W_o}{\sum W_o} \tag{5.1}$$

Keterangan :

W_j = Nilai perbaikan bobot user

W_o = Nilai Bobot dari user

$\sum W_o$ = Jumlah nilai bobot dari user

(Sumber: Kusumadewi dkk, 2006)

2. Menghitung nilai vektor S setiap alternatif menggunakan rumus 5.2.

$$S_i = \prod_{j=1}^n (X_{ij})^{W_j} \tag{5.2}$$

Keterangan :

S_i = Nilai prefensi alternatif A_n (Vektor S)

W_j = Nilai perbaikan bobot *user* (Positif jika *benefit*/ Negatif jika *cost*)

X_{ij} = Nilai rating kinerja atribut ke-i terhadap subkriteria ke-j

(Sumber: Kusumadewi dkk, 2006)

3. Menghitung nilai vektor V dari setiap Alternatif dengan menggunakan rumus 5.3.

$$V_i = \frac{\prod_{j=1}^n (X_{ij})^{W_j}}{\prod_{j=1}^n (X_j^*)^{W_j}} \quad (5.3)$$

Keterangan :

V_i = Nilai preferensi relatif dari setiap alternatif (Vektor V)

W_j = Nilai perbaikan bobot *user* (Positif jika *benefit*/ Negatif jika *cost*)

X_{ij} = Nilai rating kinerja atribut ke-i terhadap subkriteria ke-j

X_j^* = Jumlah nilai preferensi alternatif (Vektor S)

(Sumber: Kusumadewi dkk, 2006)

Hasil Perhitungan manual rekomendasi *homestay* menggunakan metode *Weighted Product* (WP) dapat dilihat pada penjelasan di bawah ini. Pada tabel 5.1 ini merupakan tabel yang berisi nilai tingkat kepentingan, dimana hal ini akan ditentukan oleh calon wisatawan yang menggunakan fitur saran pencarian *homestay*.

Tabel 5.1 Nilai Tingkat Kepentingan

Tingkat Kepentingan	Nilai
Tinggi	5
Cukup	3
Rendah	1

Tabel 5.2 Range Nilai Tiap Sub Kriteria

HARGA (C1)	KEPADATAN PENDUDUK (C3)	WIFI (C5)
RANGE	RANGE	RANGE
<100000	<20	ADA
100000 > 150000	20 > 30	TIDAK ADA
>150000	>30	
JUMLAH KAMAR (C2)	JARAK DENGAN PUSAT (C4)	PELIHARAAN (C6)
RANGE	RANGE	RANGE
<3	<50	ADA
3 > 5	50 > 100	TIDAK ADA
>5	>100	

Tabel 5.3 Input Sub Kriteria Sesuai Keinginan User

USER
A

HARGA (C1)	KEPADATAN PENDUDUK (C3)	WIFI (C5)
RANGE NILAI	RANGE NILAI	RANGE NILAI
100000 > 150000 5	20>30 5	ADA 5
<100000 3	<20 3	TIDAK ADA 1
>150000 1	>30 1	
JUMLAH KAMAR (C2)	JARAK DENGAN PUSAT KEGIATAN (C4)	PELIHARAAN (C6)
RANGE NILAI	RANGE NILAI	RANGE NILAI
<3 5	>100 5	ADA 1
>5 3	50 > 100 3	TIDAK ADA 5
3 > 5 1	<50 1	

Tabel 5.4 Nilai Awal Alternatif Homestay

Alternatif	Jumlah Kamar	Harga/malam	Jarak dengan Pusat Kegiatan	Kepadatan Penduduk	WIFI	PELIHARAAN
HOMESTAY 7	4	120000	150	50	TIDAK ADA	TIDAK ADA
HOMESTAY	4	120000	300	10	ADA	TIDAK ADA

6						
HOMESTAY 8	4	130000	170	30	ADA	TIDAK ADA
lanjutan	6	150000	100	10	ADA	TIDAK ADA
HOMESTAY 4	4	120000	50	10	TIDAK ADA	dilanjutkan
HOMESTAY 9	6	150000	400	10	ADA	
HOMESTAY 10	4	130000	600	10	ADA	TIDAK ADA
HOMESTAY 1	2	160000	70	10	ADA	TIDAK ADA
HOMESTAY 3	6	120000	100	20	TIDAK ADA	TIDAK ADA

User menginputkan bobot masing-masing kriteria (W_0) sebesar 3, 5, 2, 3, 2, 3 sehingga jumlah bobot user adalah 18. Tahap selanjutnya adalah sebagai Berikut:

1. Menghitung Nilai Perbaikan bobot kriteria user dengan rumus 1.

$$W_j = \frac{W_0}{\sum W_0} \quad (5.1)$$

Jumlah semua perbaikan bobot kriteria harus = 1

Tabel 5.5 Perbaikan Bobot

Cost/Benefit	benefit	cost	Cost	cost	benefit	benefit	Jumlah Bobot
Kebutuhan	3	5	2	3	2	3	18
Perbaikan Bobot	0,166666667	0,277777778	0,111111111	0,166666667	0,111111111	0,166666667	1
Pangkat	0,166666667	0,277777778	0,111111111	0,166666667	0,111111111	0,166666667	

2. Menghitung Nilai Vektor S

Tahap pertama yaitu menentukan kriteria *benefit* dan *cost* dari semua kriteria yang digunakan. Berikut ini tahapannya:

$$W_1 = benefit, W_2 = cost, W_3 = cost, W_4 = cost, W_5 = benefit, W_6 = benefit.$$

Sehingga:

$$W_1 = 0,166666667 \times 1 = 0,166666667$$

$$W_2 = 0,277777778 \times -1 = -0,277777778$$

$$W_3 = 0,111111111 \times -1 = -0,111111111$$

$$W_4 = 0,166666667 \times -1 = -0,166666667$$

$$W_5 = 0,111111111 \times 1 = 0,111111111$$

$$W_6 = 0,166666667 \times 1 = 0,166666667$$

Tahap kedua adalah mengalikan hasil nilai pemangkatan atribut kriteria C_i dengan nilai W_i pada alternatif $i (A_i)$ seperti pada Tabel 5.2 menggunakan rumus 2. Perhitungan terus dilakukan sampai nilai vektor S semua alternatif diperoleh.

$$S_i = \prod_{j=1}^n (X_{ij})^{W_j} \tag{5.2}$$

Tabel 5.6 Tabel Konversi Alternatif *Homestay*

Alternatif	Jumlah Kamar	Harga/malam	Jarak dengan Pusat Kegiatan	Kepadatan Penduduk	Wifi	Peliharaan
HOMESTAY 7	1	5	5	1	1	5
HOMESTAY 6	1	5	5	3	5	5
HOMESTAY 8	1	5	5	5	5	5
HOMESTAY 5	3	5	3	3	5	5
HOMESTAY 4	1	5	3	3	1	5
HOMESTAY 9	3	5	5	3	5	5
HOMESTAY 10	1	5	5	3	5	5
HOMESTAY 1	5	1	3	3	5	5
HOMESTAY 3	3	5	3	5	1	5

$$S_1 = A_1C_1^{W_1} \times A_1C_2^{W_2} \times A_1C_3^{W_3} \times A_1C_4^{W_4} \times A_1C_5^{W_5} \times A_1C_6^{W_6}$$

$$= 1^{0,166666667} \times 5^{-0,277777778} \times 5^{-0,111111111} \times 1^{-0,166666667} \times 1^{0,111111111} \times 5^{0,166666667}$$

$$= 0,699315787$$

$$S_2 = A_2C_1^{W_1} \times A_2C_2^{W_2} \times A_2C_3^{W_3} \times A_2C_4^{W_4} \times A_2C_5^{W_5} \times A_2C_6^{W_6}$$

$$= 1^{0,166666667} \times 5^{-0,277777778} \times 5^{-0,111111111} \times 3^{-0,166666667} \times 5^{0,111111111} \times 5^{0,166666667}$$

$$\begin{aligned}
&= 0,696332166 \\
S_3 &= A_3C_1^{W1} \times A_3C_2^{W2} \times A_3C_3^{W3} \times A_3C_4^{W4} \times A_3C_5^{W5} \times A_3C_6^{W6} \\
&= 1^{0,166666667} \times 5^{-0,277777778} \times 5^{-0,111111111} \times 1^{-0,166666667} \times 1^{0,111111111} \times 5^{0,166666667} \\
&= 0,639501644
\end{aligned}$$

Tabel 5.7 Tabel Nilai Vektor S

Alternatif	Nilai Vektor S
HOMESTAY 7	0,699315787
HOMESTAY 6	0,696332166
HOMESTAY 8	0,639501644
HOMESTAY 5	0,885088152
HOMESTAY 4	0,61631535
HOMESTAY 9	0,836251031
HOMESTAY 10	0,696332166
HOMESTAY 1	1,507022212
HOMESTAY 3	0,679748725
TOTAL VEKTOR S	7,255907233

3. Menghitung Nilai Vektor V

Setelah diperoleh nilai vektor S dari semua alternatif, maka tahap selanjutnya menjumlahkan semua nilai vektor S alternatif. Kemudian menggunakan rumus 3, nilai vektor S setiap alternatif dibagi dengan jumlah vektor S semua alternatif untuk mendapatkan nilai vektor V dari setiap alternatif. Berikut ini tahap perhitungan vektor V:

$$V_i = \frac{\prod_{j=1}^n (X_{ij})^{W_j}}{\prod_{j=1}^n (X_j^*)^{W_j}} \tag{5.3}$$

Jumlah vektor S semua alternatif sebesar 7,255907233, sehingga nilai vektor v dari setiap alternatif yaitu:

$$V_1 = \text{Vektor } S_1 / \sum \text{Vektor } S = 0,699315787 / 7,255907233 = 0,096378821$$

Setelah nilai vektor V dari semua alternatif diperoleh, nilai terbesar dari semua vektor V alternatif tersebut digunakan sebagai dasar untuk merekomendasikan alternatif kepada *user*. Jadi alternatif dengan nilai vektor V terbesar merupakan alternatif yang direkomendasikan kepada *user*.

Tabel 5.8 Tabel Nilai Vektor V

	Nilai Vektor V
	0,096378821
	0,095967622
	0,088135312
	0,121981735
	0,084939806
	0,115251064
	0,095967622
	0,207695904
	0,093682114
TERTINGGI	0,207695904

Pada Tabel 5.8 dapat dilihat bahwa hasil perhitungan secara manual juga memberikan rekomendasi *homestay* yang sama dengan perhitungan yang dilakukan oleh sistem yaitu *homestay* 1 dengan nilai vektor V sebesar 0,2076959.

5.3 Hasil Evaluasi Sistem

Sistem informasi dan rekomendasi pemilihan *homestay* ini telah diuji secara *white box testing* dan *black box testing*. Pada *white box testing* telah dilakukan pengujian terhadap kode program sistem yaitu dengan membuat diagram alir kode program, grafik alir kode program, menghitung kompleksitas siklometrik, dan membuat *test case* untuk menguji kebenaran dari kode program yang telah dibuat.

Pada listing program *function* ConvertAlternative() menghasilkan perhitungan kompleksitas siklometrik dua sehingga terdapat dua *path* (jalur) dalam *test case*

listing program tersebut. Hasil *test case* tersebut dinyatakan benar untuk dua jalur yang telah diuji. Pada pengujian *black box* yang dilakukan oleh *user* ada beberapa fitur yang diuji beserta dengan *alternative flow* pada setiap fitur. Setiap fitur yang diuji dinyatakan benar sesuai dengan hasil yang diharapkan. *User* yaitu pihak desa wisata menyetujui dengan *design* dan alur sistem karena termasuk mudah dalam menjalankannya. *User interface* juga dibuat sesuai dengan keinginan pihak desa wisata sehingga mudah dipahami.

5.4 Pembahasan

Sistem informasi dan rekomendasi pemilihan *homestay* ini menggunakan metode Weighted Product (WP) dalam melakukan pencarian *homestay* yang sesuai dengan kriteria calon wisatawan. Calon wisatawan memilih nilai prioritas dari masing-masing kriteria sesuai dengan pilihan yang ada, kemudian mengklik tombol selanjutnya untuk menginputkan bobot kriteria. Pada menu drop down telah disediakan angka default yaitu 5. Calon wisatawan bisa mengganti bobot tersebut sesuai dengan nilai kepentingan tiap kriteria yang diinginkan. Apabila harga yang menurut calon wisatawan tersebut penting maka bisa menambahkan nilai tersebut. Namun apabila calon wisatawan tidak mempermasalahkan harga maka bisa membuat angka kepentingan tersebut kurang dari 5 atau tetap pada angka 5. Nilai bobot kepentingan yang diinputkan oleh user sangat berpengaruh pada hasil rekomendasi *homestay* nantinya karena nilai bobot kepentingan tersebut akan menjadi dasar dalam perankingan *homestay*. Pada gambar di bawah ini akan menjelaskan apabila input setiap nilai bobot kepentingan kriteria dibuat berbeda.

SARAN PENCARIAN HOMESTAY

Input Tingkat Kepentingan

Harga : Prioritas 1

Prioritas 2

Prioritas 3

Jarak Pusat Kegiatan : Prioritas 1

Prioritas 2

Prioritas 3

Jumlah Kamar : Prioritas 1

Prioritas 2

Prioritas 3

Kapasitas Pembuka : Prioritas 1

Prioritas 2

Prioritas 3

Fasilitas Wifi : Prioritas 1

Prioritas 2

Fasilitas Homestay Perikanan : Prioritas 1

Prioritas 2

SARAN PENCARIAN HOMESTAY

Input Bobot Kepentingan (Ubah nilai bobot jika perlu)

Harga : ▼

Jarak Pusat Kegiatan : ▼

Jumlah Kamar : ▼

Kapasitas Pembuka : ▼

Fasilitas Wifi : ▼

Fasilitas Homestay Perikanan : ▼

Gambar 5.24 Tampilan Saran Pencarian *Homestay* yang telah diisi oleh user



Gambar 5.25 Tampilan Saran *Homestay* setelah dilakukan perhitungan dengan metode Weighted Product

Tampak ketika user menginputkan nilai semua bobot kriteria dengan nilai 5 maka menghasilkan *homestay* 10 sebagai peringkat pertama dalam pencarian *homestay*. Ketika dicoba dengan menginputkan angka lain pada nilai bobot kepentingan masing-masing kriteria akan menampilkan hasil perankingan *homestay* yang berbeda.

Input Tingkat Kepentingan

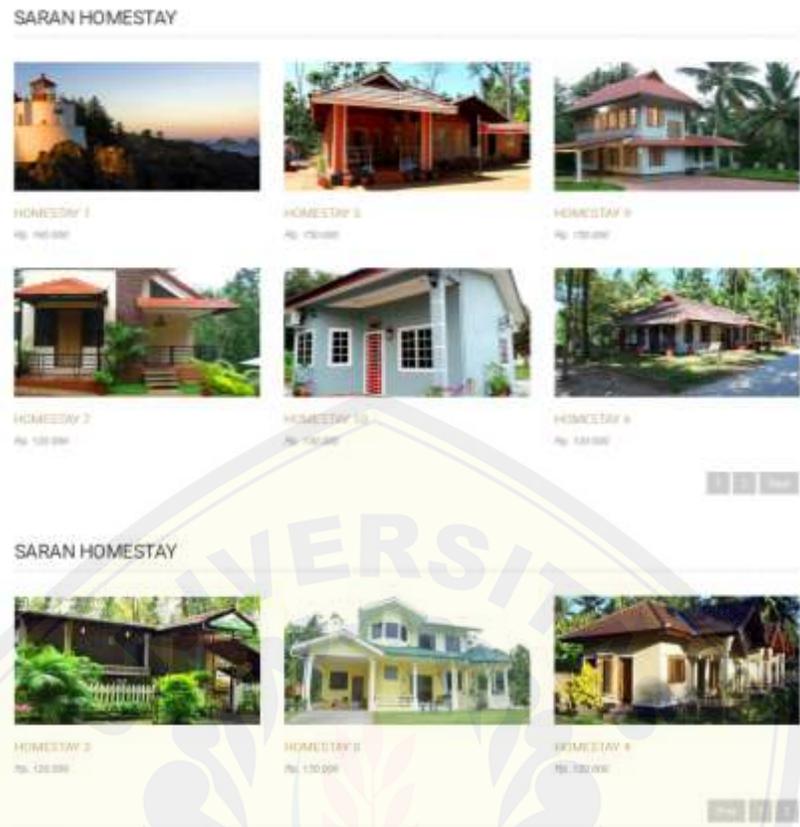
Harga:	10000 - 15000	Prioritas 1
	< 10000	Prioritas 2
	> 15000	Prioritas 3
Jarak Pusat Kegiatan:	< 1 KM	Prioritas 1
	1 - 2 KM	Prioritas 2
	> 2 KM	Prioritas 3
Jumlah Kamar:	< 2	Prioritas 1
	> 2	Prioritas 2
	2 - 5	Prioritas 3
Kapasitas Pemondok:	20 - 30	Prioritas 1
	> 30	Prioritas 2
	< 20	Prioritas 3
Facilities Wifi:	Ada	Prioritas 1
	Tidak Ada	Prioritas 2
Facilities Hewan Peliharaan:	Tidak Ada	Prioritas 1
	Ada	Prioritas 2

Input Bobot Kepentingan (Ubah nilai bobot jika perlu)

Harga:	1
Jarak Pusat Kegiatan:	2
Jumlah Kamar:	1
Kapasitas Pemondok:	1
Facilities Wifi:	2
Facilities Hewan Peliharaan:	1

Simulasi

Gambar 5.25 Tampilan Saran Pencarian *Homestay* yang telah diisi oleh user



Gambar 5.26 Tampilan Saran *Homestay* setelah dilakukan perhitungan dengan metode Weighted Product

Tampak ketika user menginputkan nilai semua bobot kriteria dengan nilai yang berbeda dengan nilai sebelumnya, maka menghasilkan *homestay* 1 sebagai peringkat pertama dalam pencarian *homestay*. Urutan *homestay* pun berubah dari awal hingga akhir. Ketika dicoba kembali dengan menginputkan angka lain pada nilai bobot kepentingan masing-masing kriteria akan menampilkan hasil perangkingan *homestay* yang berbeda pula.

BAB 6. PENUTUP

6.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan dari tahap analisis, perancangan sistem, pengkodean, dan pengujian sistem, maka dapat ditarik beberapa kesimpulan yaitu:

- 1) Sistem Informasi dan Rekomendasi Pemilihan *Homestay* bagi Wisatawan di Kawasan Desa Wisata Menggunakan Metode *Weighted Product* (Studi Kasus : Desa Wisata Tanoker Kecamatan Ledokombo Kabupaten Jember) telah berhasil dibangun.
- 2) Metode *Weighted Product* (WP) untuk memperoleh alternatif rekomendasi *homestay* sebagai alat bantu pengambilan keputusan pemilihan *homestay* berhasil diterapkan dan dapat membantu memberikan acuan bagi calon wisatawan dalam memilih *homestay* di desa wisata khususnya Desa Wisata Tanoker.
- 3) Pada sistem pendukung keputusan ini didapatkan hasil perankingan *homestay* dengan menerapkan metode *Weighted Product* (WP). *User* (calon wisatawan) memiliki nilai bobot kepentingan yang berbeda, sehingga hasil rekomendasi *homestay* setiap user akan berbeda pula.

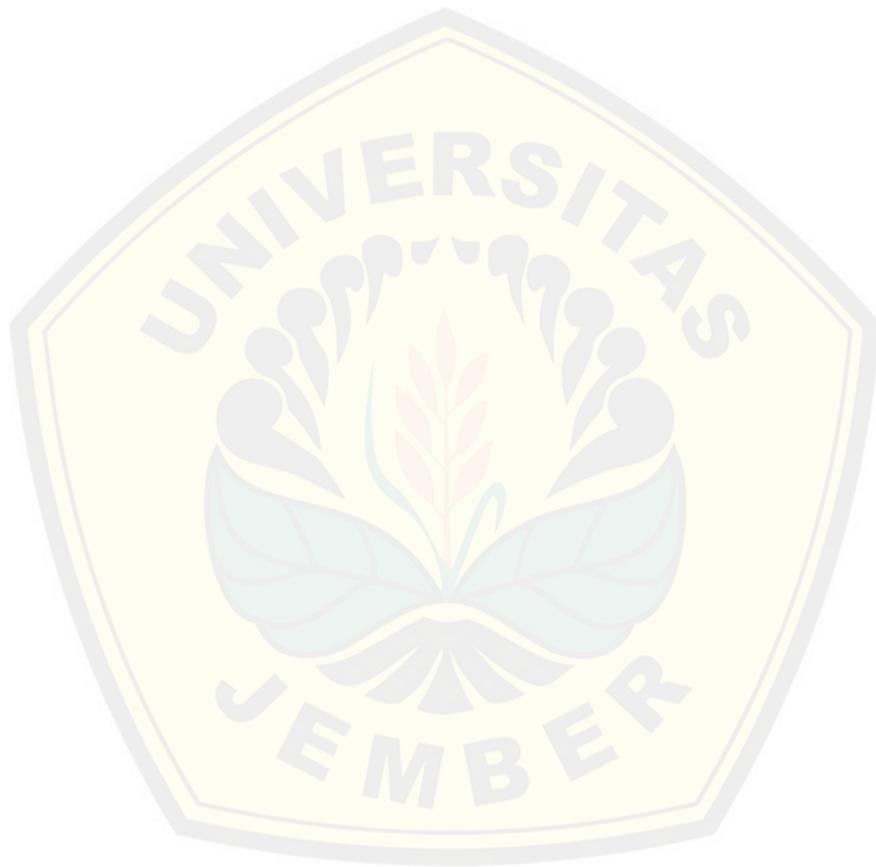
6.2 Saran

Beberapa hal yang dapat disarankan untuk pengembangan lebih lanjut adalah sebagai berikut:

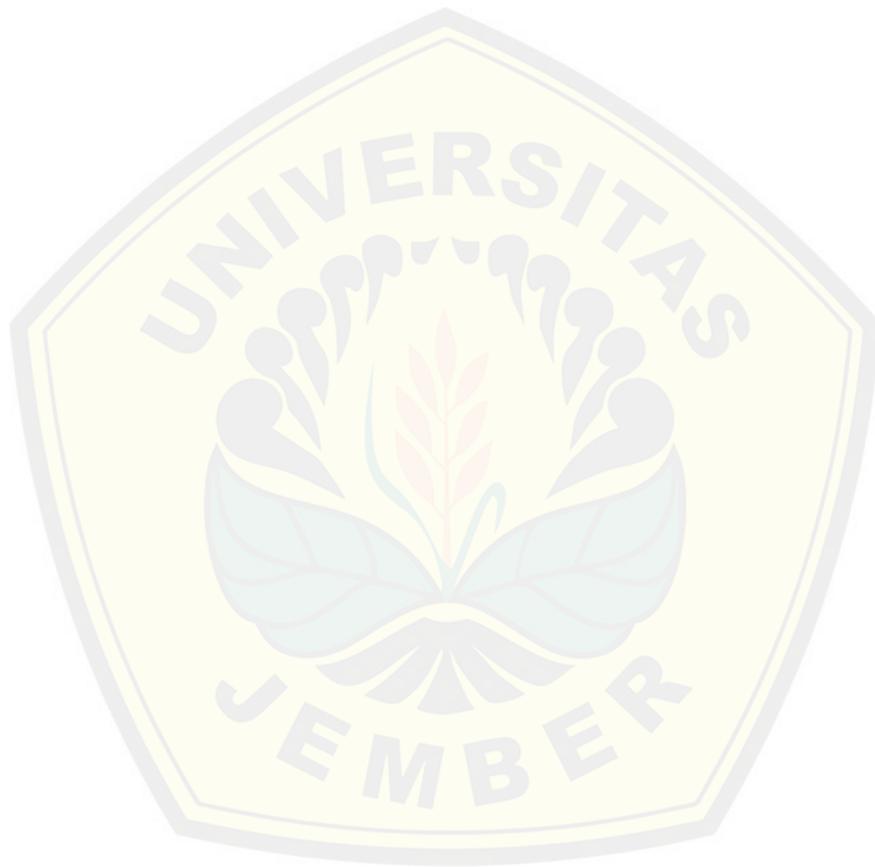
- 1) Membuat aplikasi ini tersedia dalam banyak bahasa agar berguna untuk turis mancanegara.
- 2) Menerapkan aplikasi ini pada desa wisata lain selain Desa Wisata Tanoker yang ada di luar Kabupaten Jember.

DAFTAR PUSTAKA

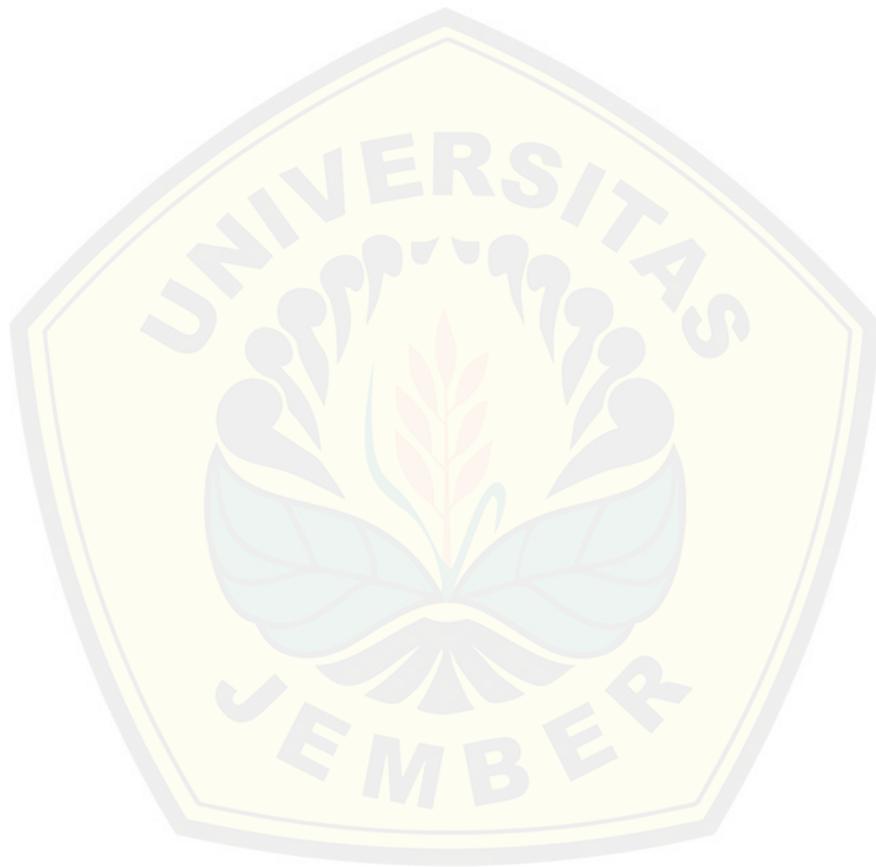
- Yeh, Chun-Shing. (2002). "A Problem-based Selection of Multi-Attribute Decision Making Methods", dalam Fuzzy Multi-Attribute Decision Making (FUZZY MADM). Yogyakarta : Graha Ilmu.
- Nugroho, Bunafit. (2004). Aplikasi Pemrograman Web Dinamis dengan PHP dan MySQL : Studi Kasus Membuat Sistem Informasi Pengolahan. Yogyakarta : Gaya Media.
- J. L. Whitten. (2004). Metode Desain dan Analisis Sistem. Yogyakarta : Andi.
- Turban. Aronso. and Liang. (2005) Decision Support Systems and Intelligent Systems. Yogyakarta : Andi.
- Jogiyanto. (2005). Analisis dan Desain Sistem Informasi. Yogyakarta : Andi.
- S. Kusumadewi, S. Hartati, A. Harjoko and R. Wardoyo. (2006). Fuzzy Multi-Attribute Decision Making (FUZZY MADM). Yogyakarta : Graha Ilmu.
- Kusrini. (2007). Konsep dan Aplikasi Sistem Pendukung Keputusan. Yogyakarta : Andi.
- F. Nugraha. (2011). Sistem Pendukung Keputusan Dengan Metode Simple Additive Weighting (SAW) Dalam Manajemen Aset. Semarang : Program Pascasarjana Universitas Diponegoro.
- Tan, Fery. (2013). Metode Prototyping Dalam Pembuatan Sebuah Software. blogger-sharearea.blogspot.com/2013















LAMPIRAN A. USE CASE SKENARIO

A.1 Use case Skenario Accept Daftar *Homestay* Pending

ID	: 01
Name	: Accept Daftar <i>Homestay</i> Pending
Participating Actor	: Super Admin
Pre Condition	: Mengakses data <i>Homestay pending</i> SI <i>Homestay</i>
Post Condition	: Menerima data <i>Homestay pending</i> SI <i>Homestay</i>

Normal Flow

Actor	System
1. Klik Daftar <i>Homestay</i> Pending	
	2. Menampilkan halaman Daftar <i>Homestay</i> Pending
3. Pilih detail pada field aksi	
	4. Menampilkan data <i>Homestay</i> yang baru didaftarkan secara rinci
5. Klik Setujui	
	6. Menampilkan Sukses Simpan Konfirmasi dan menampilkan status Accept di halaman Daftar <i>Homestay</i> Owner

Alternative Flow Tolak Daftar *Homestay* Pending

Actor	System
1. Klik Tolak	
	2. Menampilkan Sukses Simpan Konfirmasi dan menampilkan status Ignore di halaman Daftar <i>Homestay</i> Owner

dilanjutkan

lanjutan

A.2 *Use case* Skenario Kelola Registrasi Owner *Homestay*

ID	: 02
Name	: Kelola Registrasi Owner <i>Homestay</i>
Participating Actor	: Super Admin
Pre Condition	: Mengakses data registrasi owner <i>Homestay SI Homestay</i>
Post Condition	: Menyetujui registrasi owner <i>Homestay</i> <i>SI Homestay</i>

Normal Flow

Actor	System
1. Klik Owner	
	2. Menampilkan halaman Daftar Owner <i>Homestay</i>
3. Pilih Setujui pada field aksi	
	4. Menampilkan pilihan Konfirmasi
5. Klik Setujui	
	6. Menampilkan Sukses Konfirmasi Owner dan owner bisa log in dengan email dan <i>password</i> yang dimiliki

Alternative Flow Batal Accept Registrasi Owner

Actor	System
5. Klik Cancel	
	6. Menutup pilihan Konfirmasi
	7. Menampilkan halaman Daftar Owner <i>Homestay</i>

dilanjutkan

lanjutan

A.3 Use case Skenario Kelola *User* Aplikasi

ID	: 03
Name	: Kelola <i>User</i> Aplikasi
Participating Actor	: Super Admin
Pre Condition	: Mengakses halaman Daftar <i>User</i>
Post Condition	: Memilih status dari <i>user</i>

Normal Flow

Actor	System
1. Klik Daftar <i>User</i> Aplikasi	
	2. Menampilkan halaman Daftar <i>User</i>
3. Pilih Non Aktifkan untuk menonaktifkan admin dan pilih aktifkan untuk mengaktifkan kembali admin SI <i>Homestay</i>	
	4. Menampilkan alert Ubah Status
5. Klik non aktifkan atau aktifkan	
	6. Menampilkan alert Sukses ubah status <i>user</i>

A.4 Use case Skenario Tambah *User* Aplikasi

ID	: 04
Name	: Tambah <i>User</i> Aplikasi
Participating Actor	: Super Admin
Pre Condition	: Mengakses form tambah <i>user</i>
Post Condition	: Mengisi field pada form tambah <i>user</i>

dilanjutkan

lanjutan

Normal Flow	
Actor	System
1. Pilih Tambah <i>User</i> Aplikasi	
	2. Menampilkan halaman Form tambah <i>user</i>
3. Mengisi semua field pada Form tambah <i>user</i>	
4. Klik Simpan	5. Menyimpan data <i>user</i> ke database dan menampilkan pesan Sukses Simpan <i>User</i>
	6. Menampilkan Halaman Daftar <i>User</i> Aplikasi
Alternative Flow Gagal Simpan Data	
4. Klik Simpan	5. Menampilkan pesan Gagal Simpan <i>User</i>
	6. Menampilkan Halaman Daftar <i>User</i> Aplikasi
Alternative Flow Form Belum Terisi Semua	
Actor	System
4. Klik Simpan	5. Menampilkan peringatan This field is required pada field yang belum terisi

A.5 Use case Skenario Accept Daftar Booking Baru

ID	: 05
Name	: Accept Daftar Booking Baru

dilanjutkan

lanjutan

Participating Actor	: - Primary actor : Super Admin - Secondary actor : Admin
Pre Condition	: Mengakses daftar booking baru SI <i>Homestay</i>
Post Condition	: Menerima daftar booking baru SI <i>Homestay</i>

Normal Flow

Actor	System
3. Klik Konfirmasi pada field Aksi	
	4. Menampilkan dialog box untuk konfirmasi booking baru
5. Klik Konfirmasi	
	6. Menampilkan Sukses Simpan Konfirmasi

A.6 *Use case* Skenario Memilih *Homestay*

ID	: 06
Name	: Memilih <i>homestay</i>
Participating Actor	: <i>User</i>
Pre Condition	: Mengakses Halaman <i>Homestay</i>
Post Condition	: Memilih <i>Homestay</i>

Normal Flow

Actor	System
1. Klik Gunakan saran untuk cari <i>homestay</i>	
	2. Menampilkan halaman form saran pencarian <i>homestay</i>
3. Mengisi nilai bobot sub kriteria dan nilai bobot kriteria sesuai dengan kriteria <i>homestay</i> yang diinginkan	

dilanjutkan

lanjutan

	4. Menampilkan halaman form saran pencarian <i>homestay</i> yang sudah terisi
5. Klik Submit Now	
	6. Menampilkan halaman rekomendasi <i>homestay</i> sesuai dengan kriteria yang diinginkan
Alternative Flow <i>Input</i> Nilai Bobot Lebih Dari 100	
Actor	System
5. Klik Submit Now	
	6. Menampilkan alert Nilai Bobot lebih dari 100
Alternative Flow Form Belum Terisi Semua	
Actor	System
5. Klik Submit Now	
	6. Menampilkan alert Maaf, field tidak boleh kosong
Alternative Flow <i>Input</i> Nilai Bobot Kurang Dari 100	
Actor	System
5. Klik Submit Now	
	6. Menampilkan alert Nilai Bobot lebih dari 100
Alternative Flow <i>Input</i> Nilai Bobot Bukan Angka	
Actor	System
5. Klik Submit Now	
	6. Menampilkan alert <i>Input</i> harus angka

A.7 Use case Skenario Booking *Homestay*

ID	: 07
Name	: Booking <i>homestay</i>
Participating Actor	: <i>User</i>
Pre Condition	: Mengakses Halaman <i>Homestay</i>
Post Condition	: Memesan <i>Homestay</i>

Normal Flow

Actor	System
1. Klik menu <i>Homestay</i> pada Halaman Utama	
	2. Menampilkan halaman berisi daftar <i>homestay</i> yang tersedia
3. Memilih <i>homestay</i> yang akan dipesan	
	4. Menampilkan halaman detail <i>homestay</i>
5. Klik Booking <i>Homestay</i> Ini	
6. Mengisi Form Booking	
	7. Menampilkan Form Booking yang sudah terisi
8. Klik Simpan	
	9. Menampilkan alert Sukses Simpan Booking Dikeranjang
10. Klik Shopping Cart	
11. Klik Lanjut Order	
	12. Menampilkan Form Order
13. Mengisi Form Order	
14. Klik Simpan	

	15. Menampilkan alert Kami telah mengirim pesan ke email anda. Mohon cek inbox email
Alternative Flow Tanggal Check Out Kurang Dari Tanggal Check In	
Actor	System
7. Klik Simpan	
	8. Menampilkan alert mohon cek kembali tanggal check in dan check out nya
Alternative Flow Form Belum Terisi Semua	
Actor	System
14. Klik Simpan	
	15. Field akan berwarna merah menunjukkan field harus diisi
Alternative Flow Menggunakan Email yang Sama Tetapi Order Sebelumnya Belum Terlunasi	
Actor	System
14. Klik Simpan	
	15. Menampilkan alert Alamat Email yang sebelumnya belum dikonfirmasi
Alternative Flow Seluruh Kamar Sudah Dipesan di Tanggal yang Sama	
Actor	System
8. Klik Simpan	
	9. Menampilkan alert Quota Kamar <i>Homestay</i> tidak mencukupi pada tanggal tersebut

A.8 Use case Skenario Registrasi

ID	: 08
Name	: Registrasi Owner <i>Homestay</i>
Participating Actor	: Pemilik <i>Homestay</i>
Pre Condition	: Mengakses Form Registrasi Owner <i>Homestay</i>
Post Condition	: Mengisi Form Registrasi Owner <i>Homestay</i>

Normal Flow

Actor	System
1. Klik Daftar Disini	
	2. Menampilkan halaman register
3. Mengisi Form Owner Application	
	4. Menampilkan form yang sudah terisi
5. Klik Sign Up	
	6. Menampilkan alert Sukses registrasi owner mohon tunggu konfirmasi

Alternative Flow Form Belum Terisi Semua

Actor	System
5. Klik Sign Up	
	6. Menampilkan peringatan This field is required pada field yang belum terisi

Alternative Flow Registrasi Owner Gagal

Actor	System
-------	--------

dilanjutkan

lanjutan

5. Klik Sign Up

6. Menampilkan alert Gagal

Simpan Data

7. Kembali ke halaman Registrasi

Owner *Homestay*

A.9 Use case Skenario Tambah *Homestay*

ID : 09

Name : Tambah *Homestay*

Participating Actor : Pemilik *Homestay*

Pre Condition : Mengakses Form Tambah *Homestay*

SI *Homestay*

Post Condition : Mengisi Form Tambah *Homestay* SI

Homestay

Normal Flow

Actor

System

1. Klik Tambah *Homestay*

2. Menampilkan halaman Form

Tambah *Homestay*

3. Mengisi Form Tambah

Homestay

4. Menampilkan Form Tambah

Homestay yang sudah terisi

5. Klik Simpan

6. Menampilkan alert Sukses

registrasi *homestay* mohon

tunggu konfirmasi

dilanjutkan

lanjutan

Alternative Flow Batal Tambah *Homestay*

Actor	System
5. Klik Cancel	
	6. Kembali ke halaman Tambah <i>Homestay</i>

Alternative Flow Form Belum Terisi Semua

Actor	System
5. Klik Simpan	
	6. Menampilkan peringatan This field is required pada field yang belum terisi

Alternative Flow Tambah *Homestay* Gagal

Actor	System
7. Klik Save	
	8. Menampilkan alert Data Tidak Berhasil Ditambah
	9. Kembali ke halaman Tambah <i>Homestay</i>

A.10 Use case Skenario Edit Daftar *Homestay*

ID	: 10
Name	: Edit Daftar <i>Homestay</i> Owner
Participating Actor	: Pemilik <i>Homestay</i>
Pre Condition	: Mengakses Form Ubah <i>Homestay</i>
Post Condition	: Mengisi Form Ubah <i>Homestay</i>

Normal Flow

Actor	System
1. Klik Daftar <i>Homestay</i>	
2. Pilih Edit pada field aksi	

dilanjutkan

lanjutan

	3. Menampilkan halaman Form Ubah <i>Homestay</i>
4. Mengisi Form Ubah <i>Homestay</i>	
	5. Menampilkan Form Ubah <i>Homestay</i> yang sudah terisi
6. Klik Simpan	
	7. Menampilkan alert Sukses Ubah <i>Homestay</i>
Alternative Flow Batal Tambah <i>Homestay</i>	
Actor	System
7. Klik Cancel	
	8. Kembali ke halaman Tambah <i>Homestay</i>
Alternative Flow Form Belum Terisi Semua	
Actor	System
5. Klik Simpan	
	6. Menampilkan peringatan This field is required pada field yang belum terisi
Alternative Flow Tambah <i>Homestay</i> Gagal	
Actor	System
8. Klik Save	
	9. Menampilkan alert Gagal Ubah <i>Homestay</i>
	10. Kembali ke halaman Tambah <i>Homestay</i>

A.11 Use case Skenario Hapus Daftar *Homestay*

ID	: 11
Name	: Hapus Daftar <i>Homestay</i> Owner
Participating Actor	: Pemilik <i>Homestay</i>
Pre Condition	: Mengakses Hapus Daftar <i>Homestay</i> Owner
Post Condition	: Menghapus Daftar <i>Homestay</i> Owner

Normal Flow

Actor	System
1. Klik Daftar <i>Homestay</i>	
	2. Menampilkan halaman Daftar <i>Homestay</i> Owner
3. Pilih Hapus pada field aksi	
	4. Menampilkan konfirmasi hapus <i>homestay</i>
5. Klik Oke	
	6. Menampilkan alert Sukses Hapus <i>Homestay</i>

Alternative Flow Batal Hapus *Homestay*

Actor	System
7. Klik Batal	
	8. Menampilkan halaman Daftar <i>Homestay</i> Owner

A.12 Use case Skenario Upload Foto *Homestay*

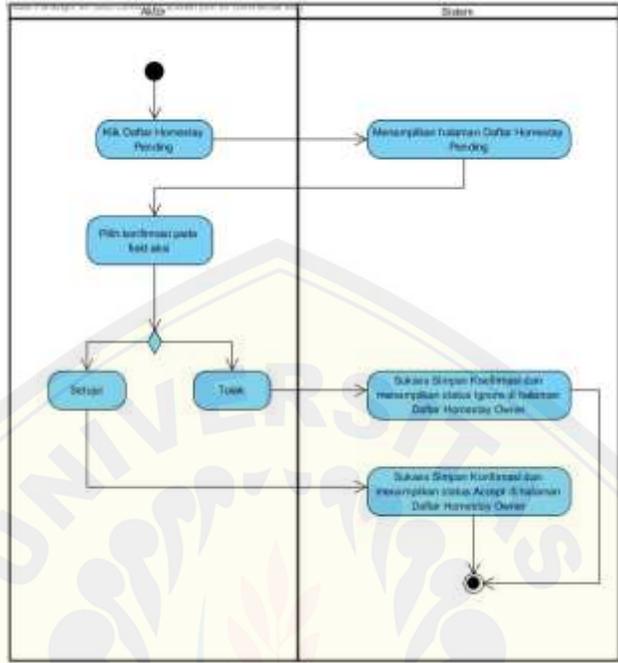
ID	: 12
Name	: Upload Foto <i>Homestay</i>
Participating Actor	: Pemilik <i>Homestay</i>
Pre Condition	: Mengakses Upload Foto <i>Homestay</i>
Post Condition	: Menambahkan Foto <i>Homestay</i>

Normal Flow

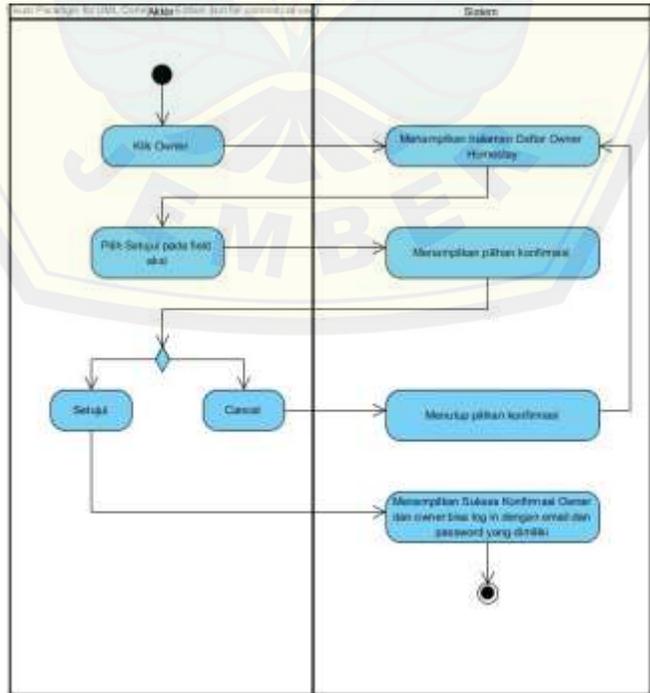
Actor	System
1. Klik Daftar <i>Homestay</i>	
	2. Menampilkan halaman Daftar <i>Homestay</i> Owner
3. Pilih Upload pada field aksi	
	4. Menampilkan Form upload photo <i>homestay</i>
5. Mengisi Form Upload Photo dengan ukuran foto 642x335	
	6. Menampilkan Form Upload Photo yang sudah terisi foto <i>homestay</i> dan alert Sukses Upload

LAMPIRAN B. ACTIVITY DIAGRAM

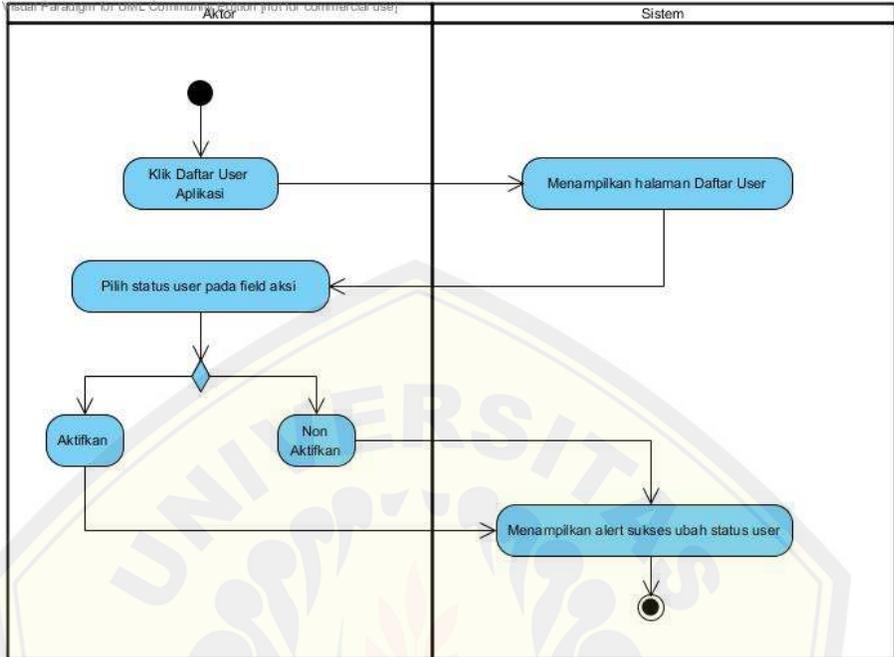
B.1 Activity Diagram Accept Daftar *Homestay Pending*



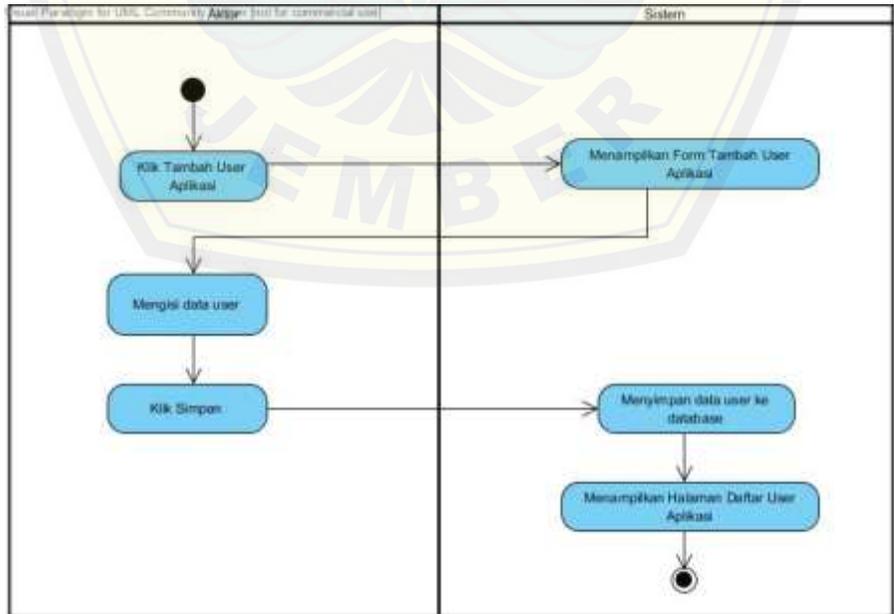
B.2 Activity Diagram Kelola Registrasi Pemilik *Homestay*



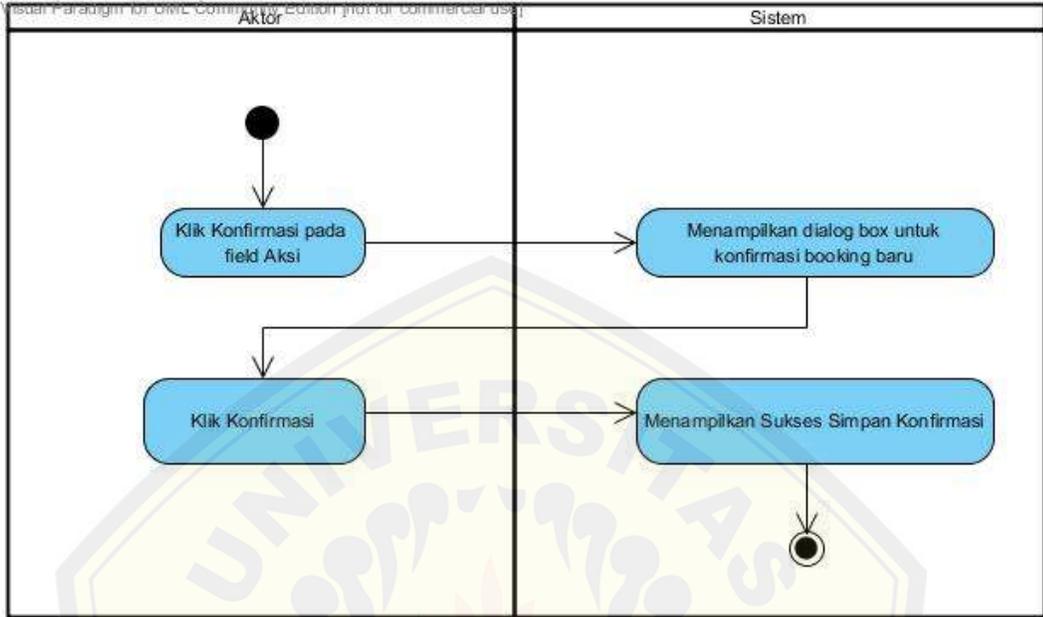
B.3 Activity Diagram Kelola User Aplikasi



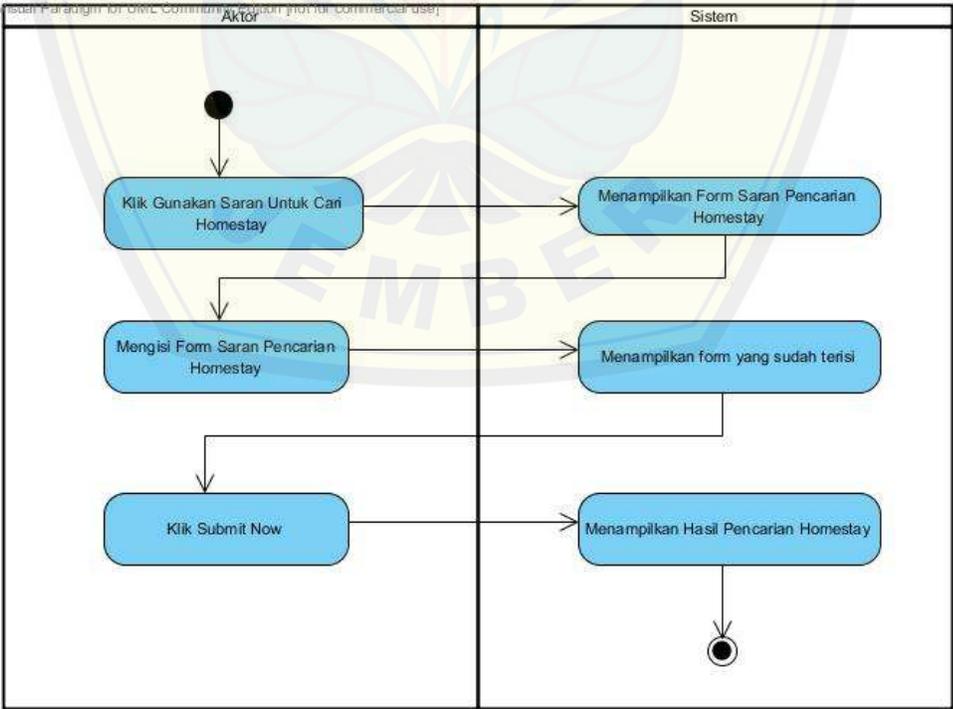
B.4 Activity Diagram Tambah User Aplikasi



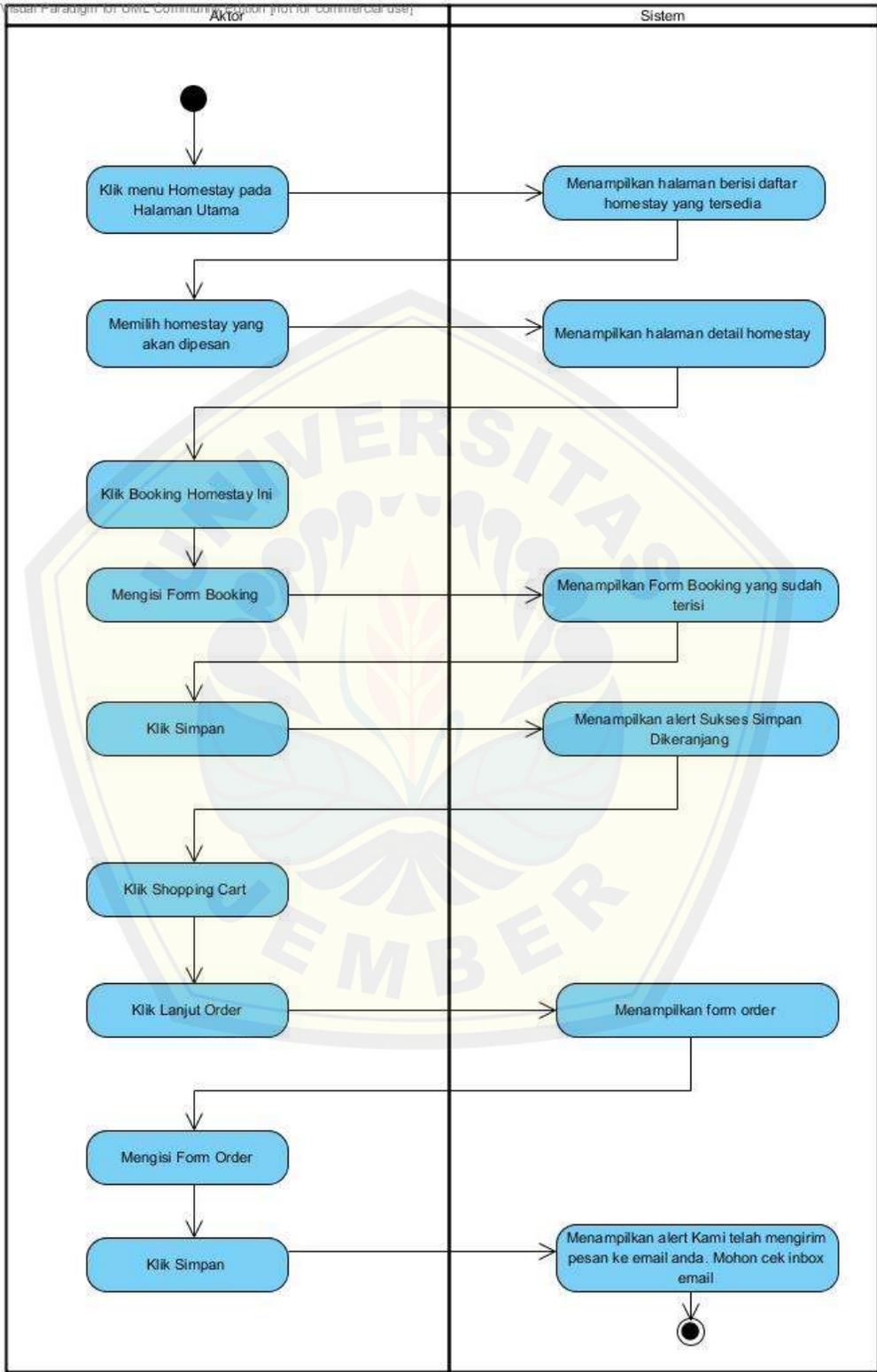
B.5 Activity Diagram Accept Daftar Booking Baru



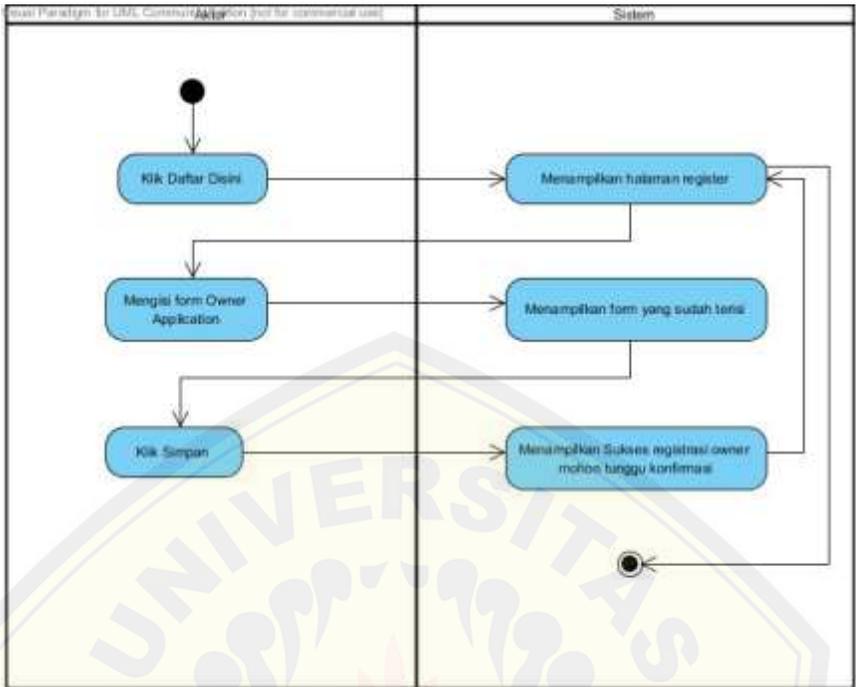
B.6 Activity Diagram Memilih Homestay



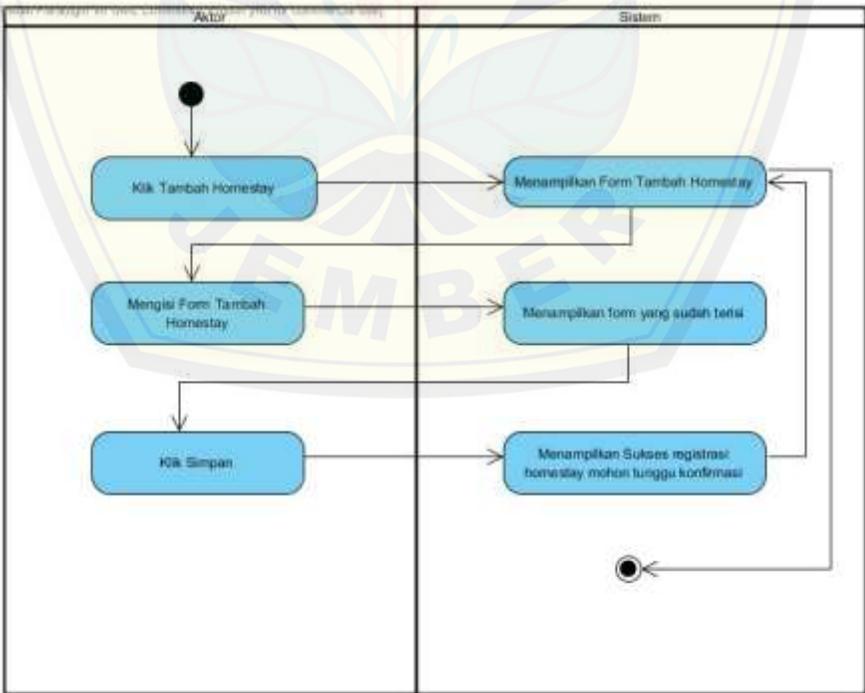
B.7 Activity Diagram Booking Homestay



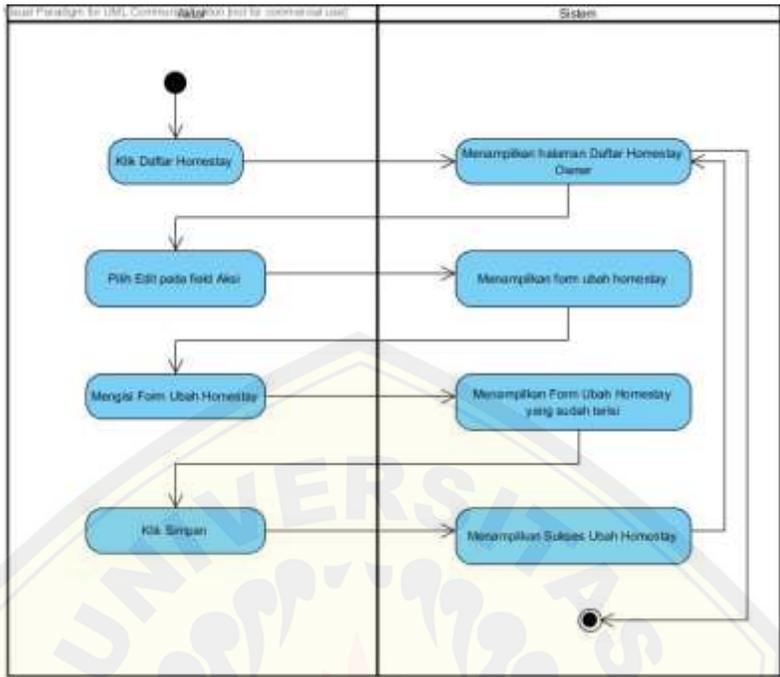
B.8 Activity Diagram Registrasi



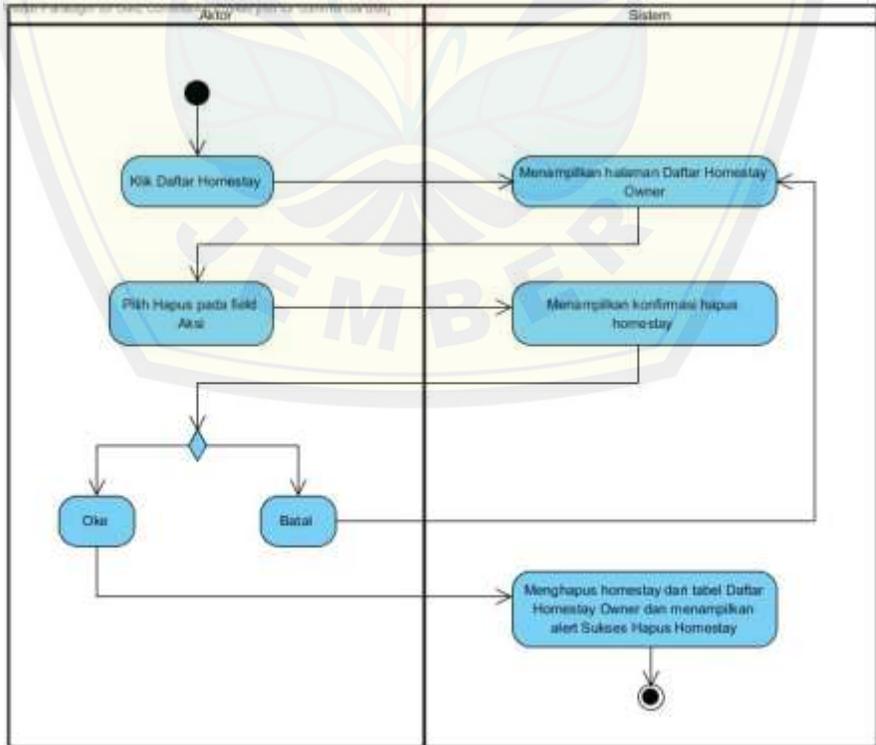
B.9 Activity Diagram Tambah Homestay



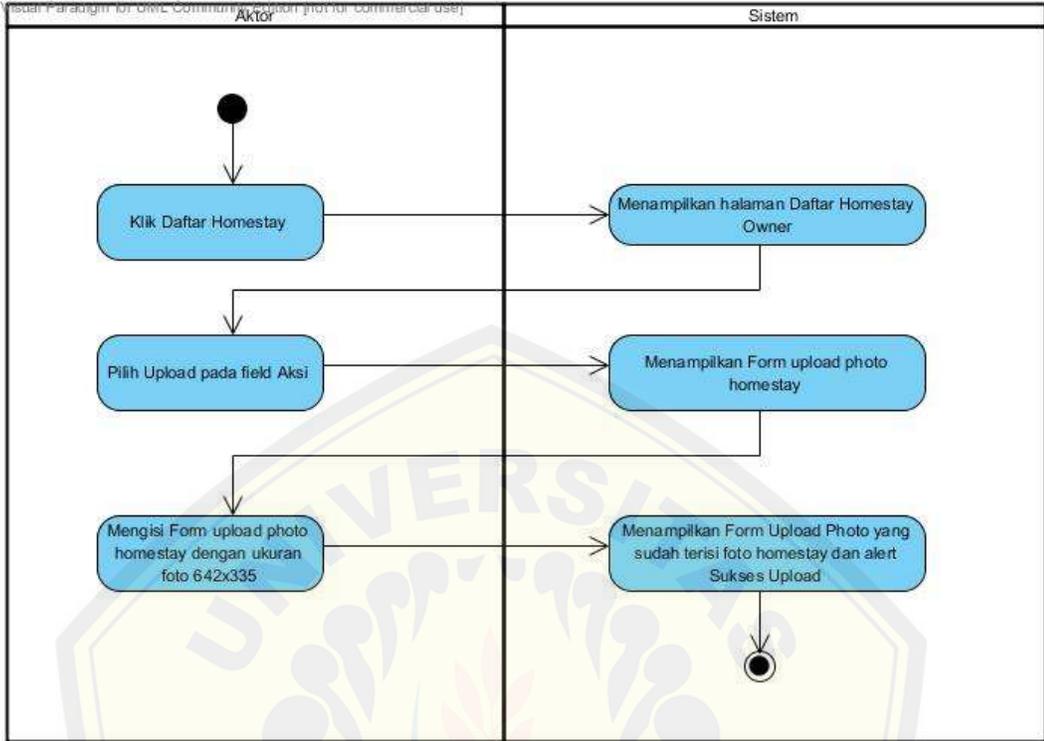
B.10 Activity Diagram Edit Daftar Homestay



B.11 Activity Diagram Hapus Daftar Homestay

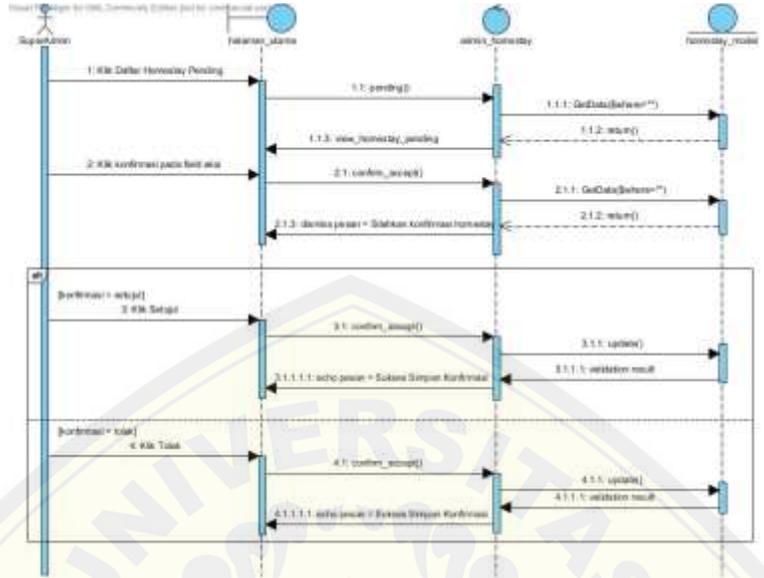


B.12 Activity Diagram Upload Foto *Homestay*

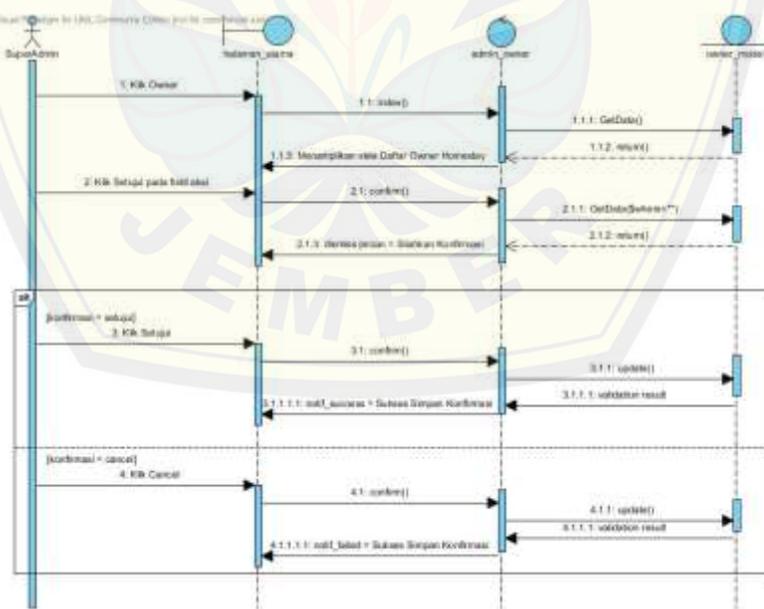


LAMPIRAN C. SEQUENCE DIAGRAM

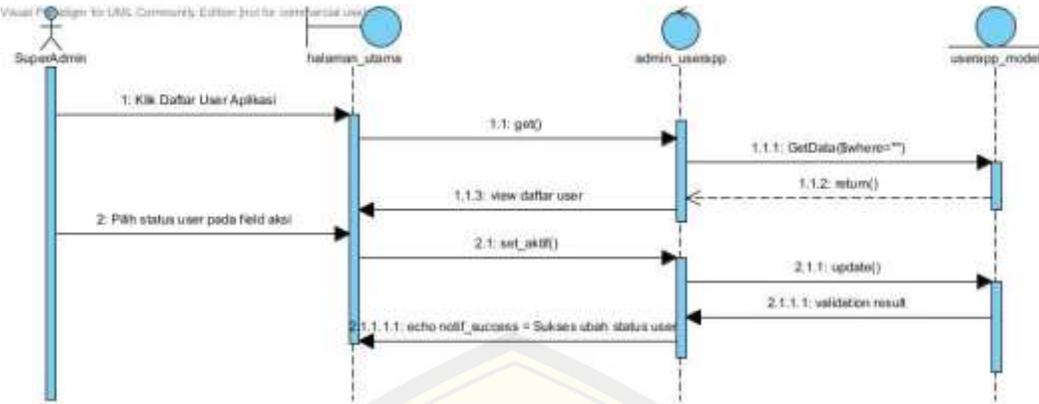
C.1 Sequence Diagram Accept Daftar Homestay Pending



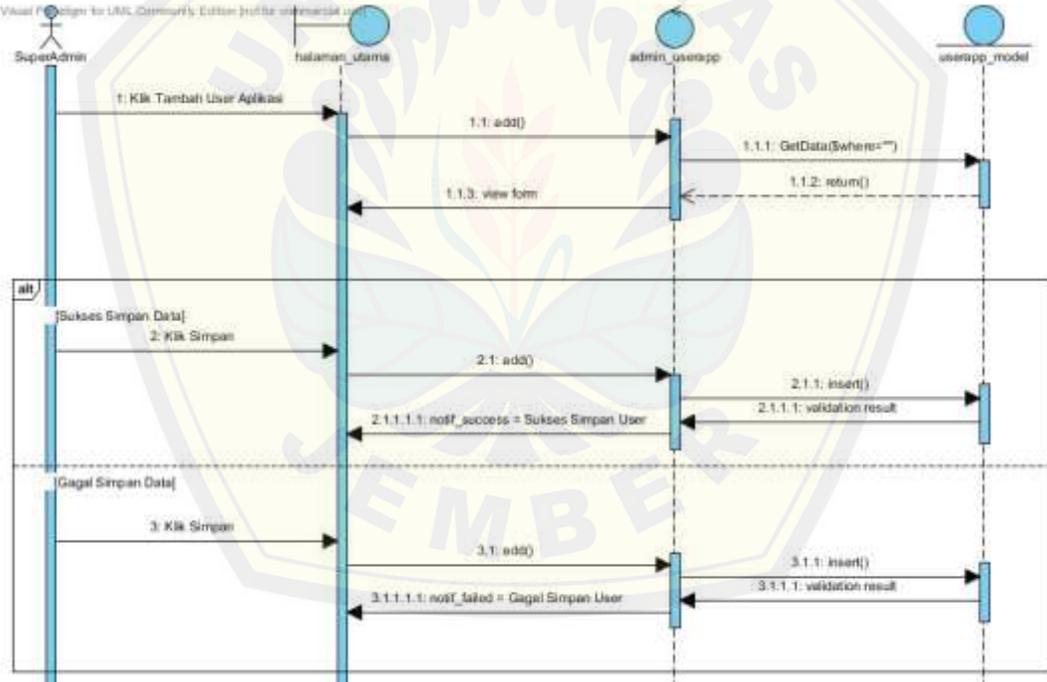
C.2 Sequence Diagram Kelola Registrasi Pemilik Homestay



C.3 Sequence Diagram Kelola User Aplikasi



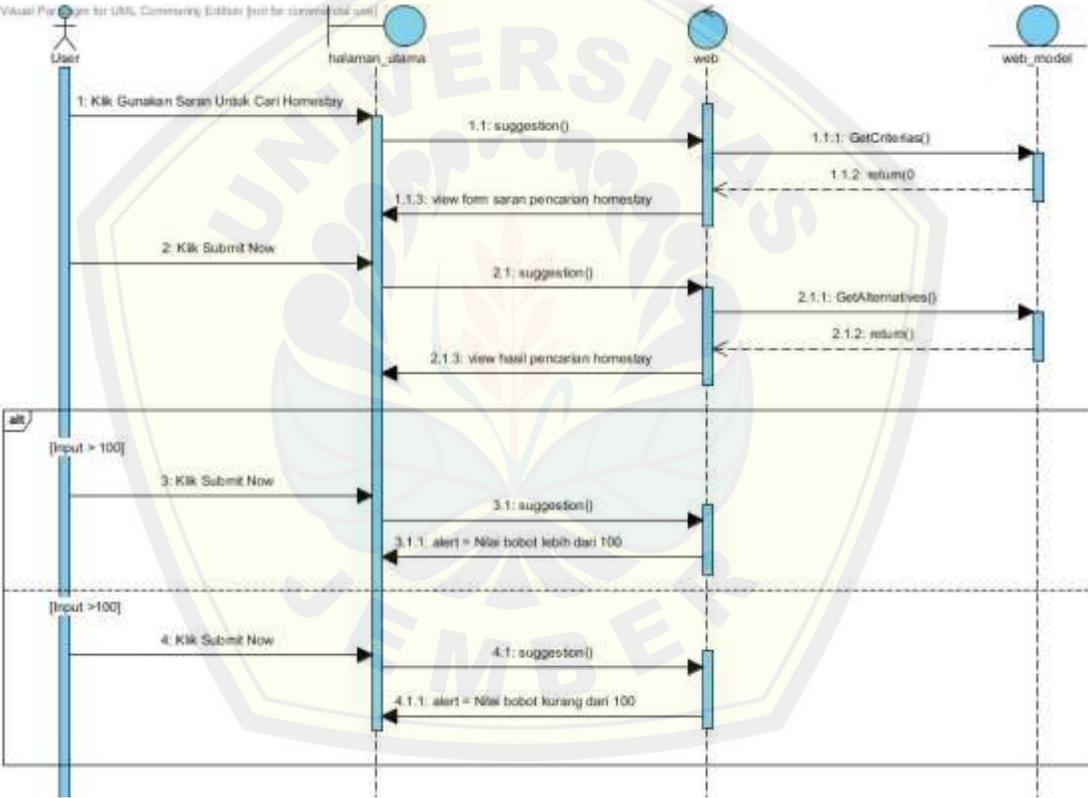
C.4 Sequence Diagram Tambah User Aplikasi



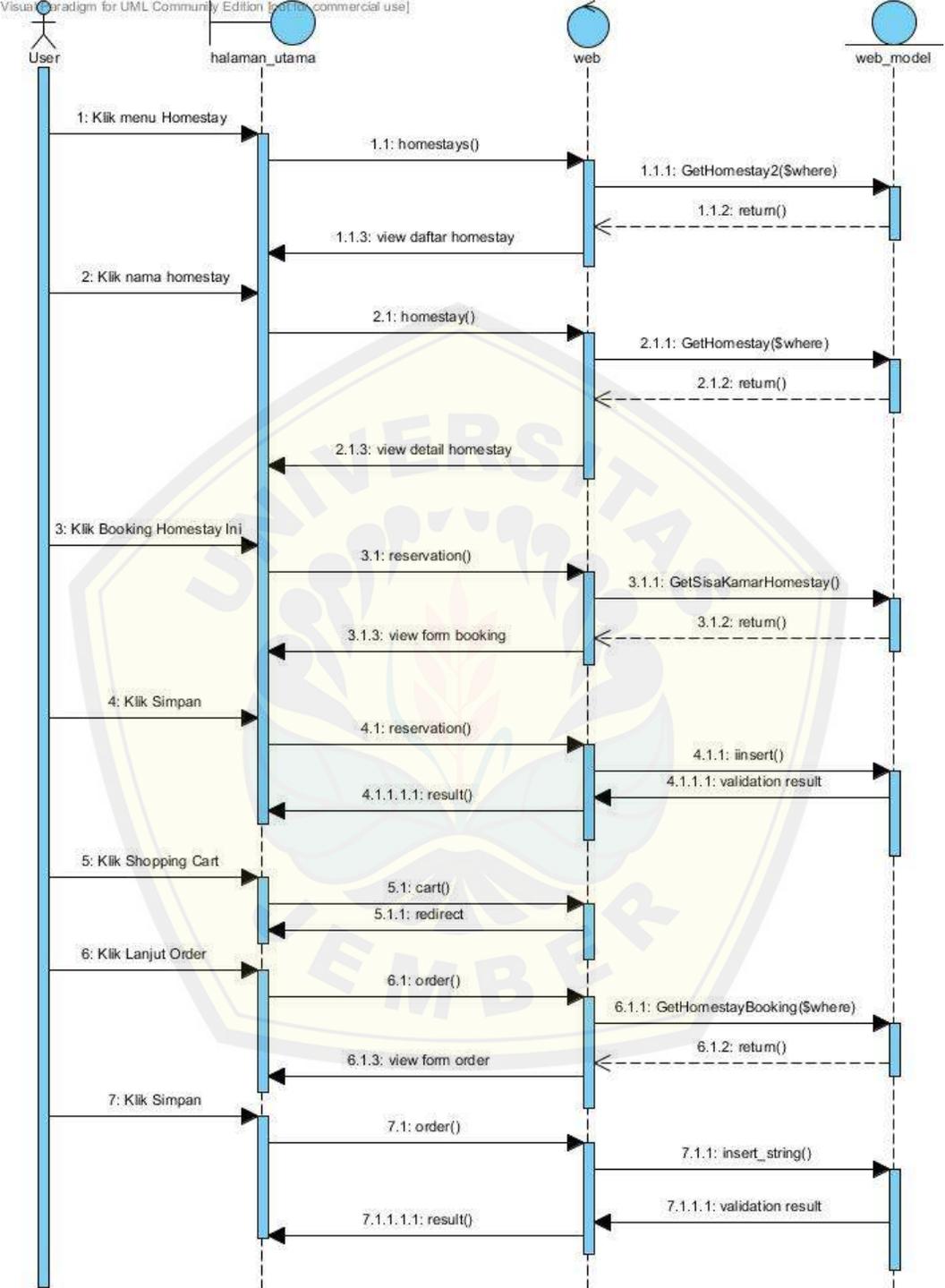
C.5 Sequence Diagram Accept Daftar Booking Baru



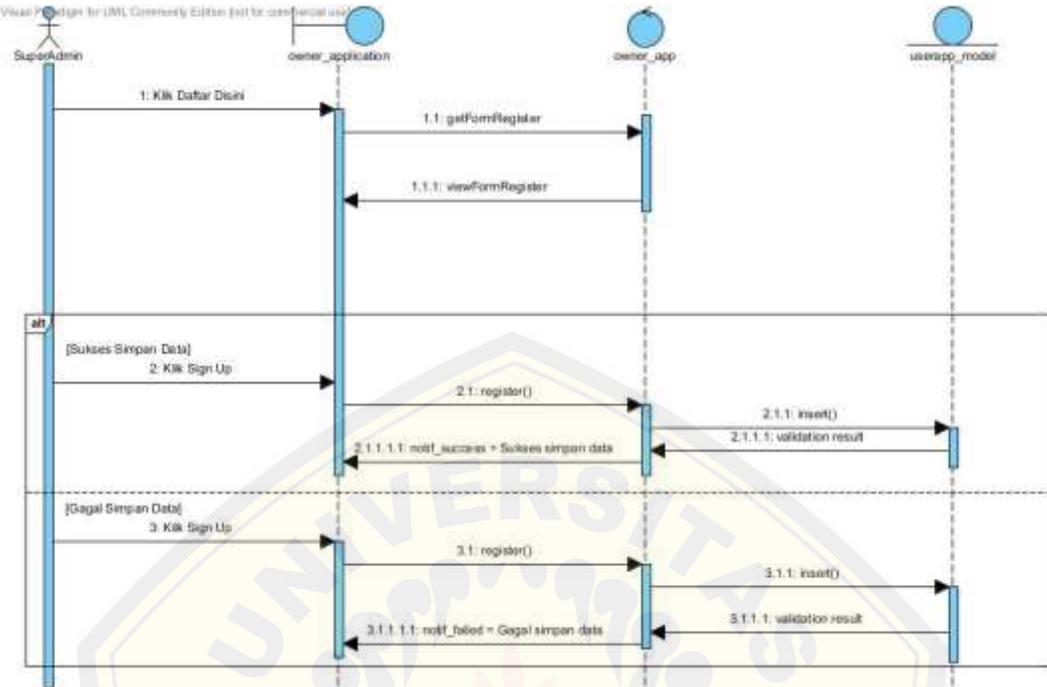
C.6 Sequence Diagram Memilih Homestay



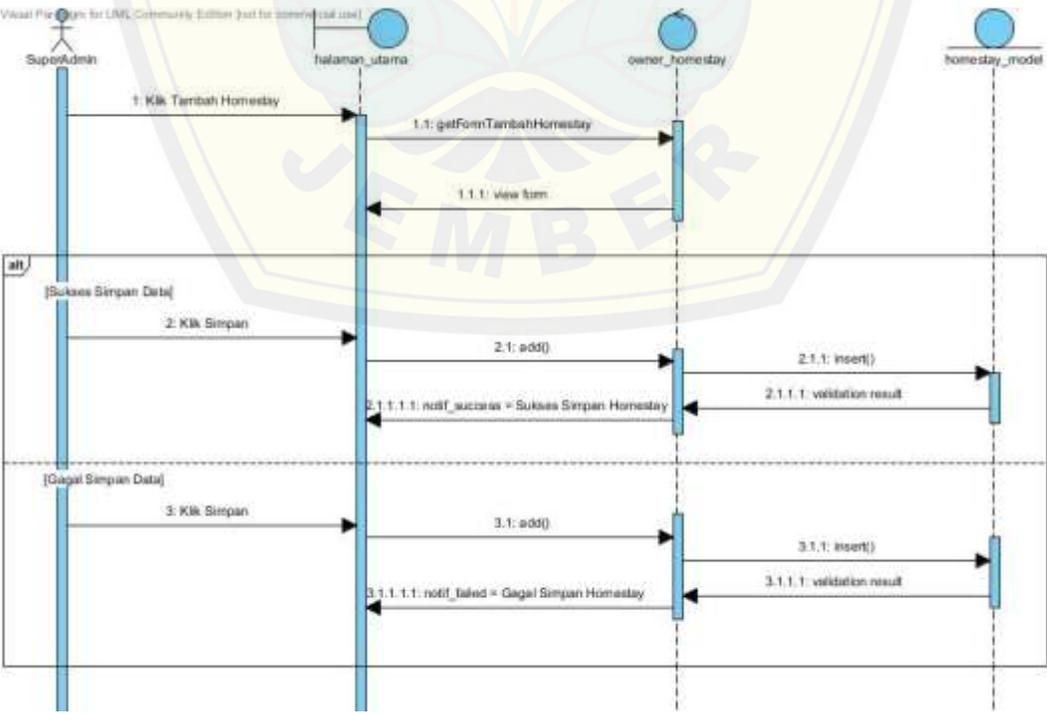
C.7 Sequence Diagram Booking Homestay



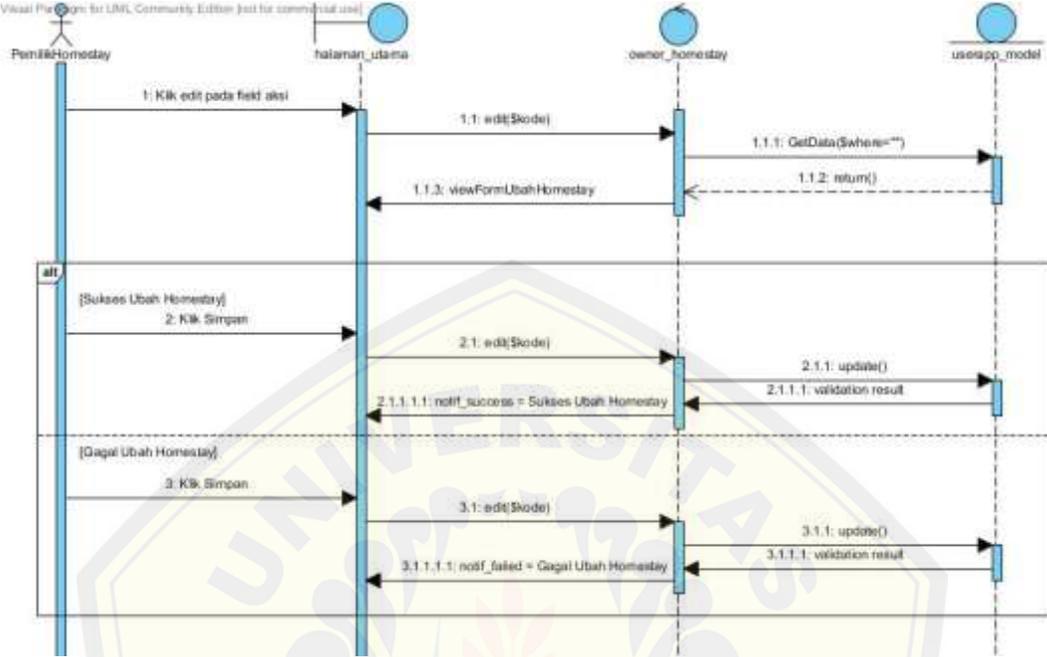
C.8 Sequence Diagram Registrasi



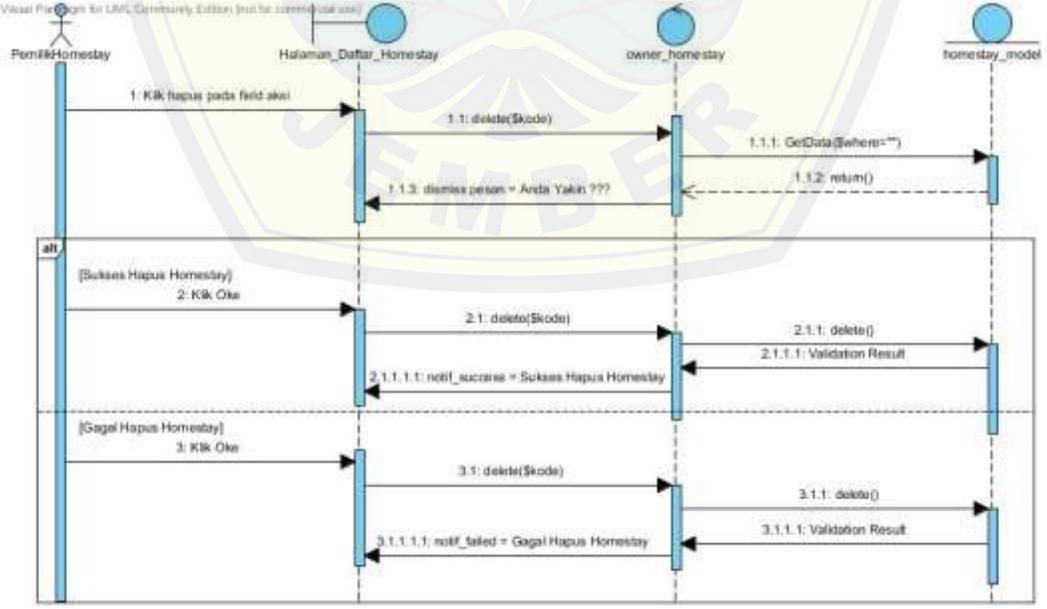
C.9 Sequence Diagram Tambah Homestay



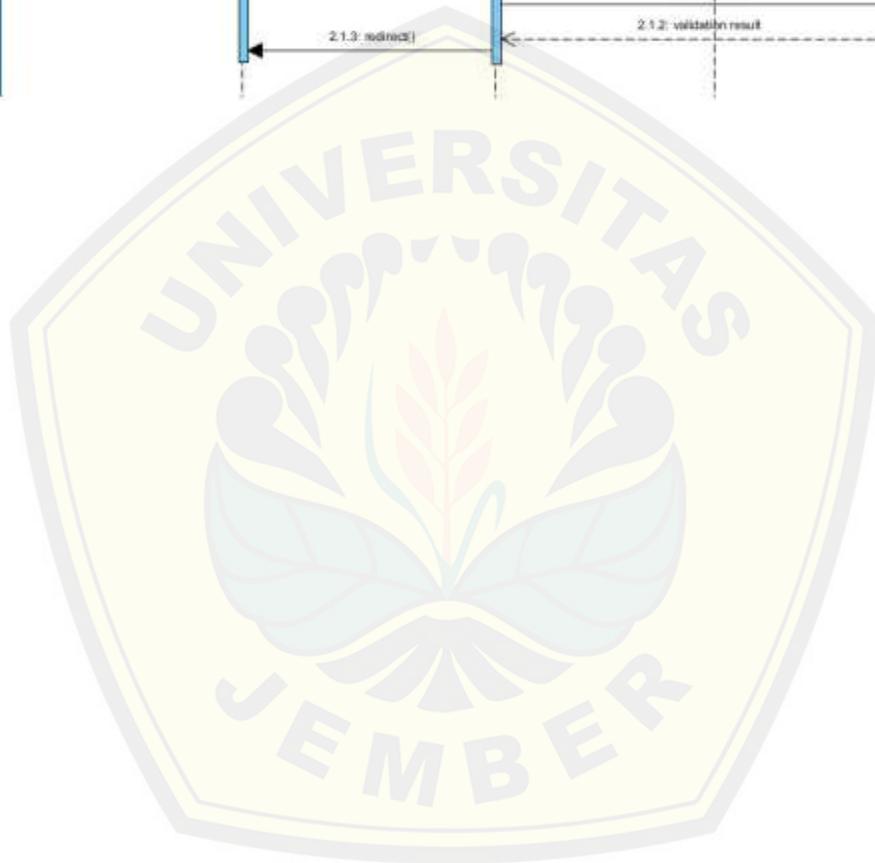
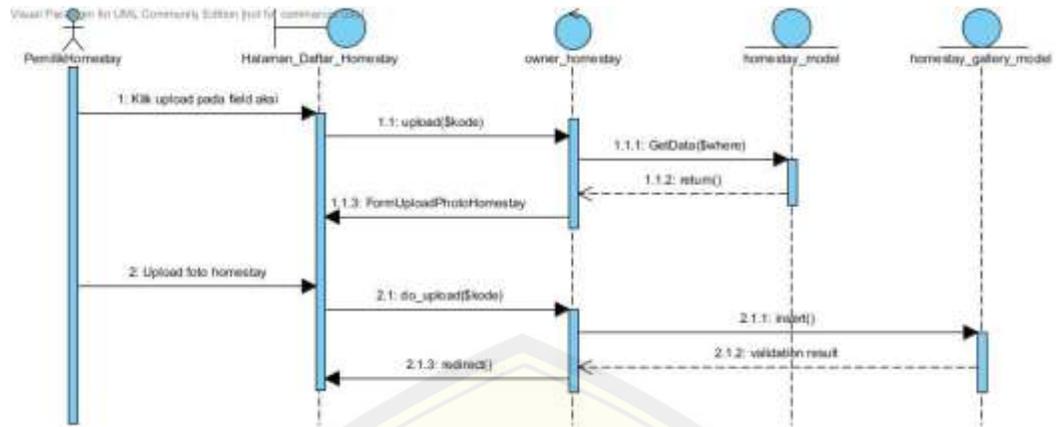
C.10 Sequence Diagram Edit Daftar Homestay



C.11 Sequence Diagram Hapus Daftar Homestay



C.12 Sequence Diagram Upload Foto *Homestay*



LAMPIRAN D. KODE PROGRAM

Kode Program Rekomendasi Pemilihan *Homestay*

```

<?php
class Weighted_product{
    public function __construct(){

        public function initialize($params = array()){
            if (count($params) > 0){
                foreach ($params as $key => $val){
                    if (isset($this->$key)){
                        $this->$key = $val;
                    }
                }
            }

            $this->ConvertAlternatives();
            $this->WeightRepair();
            $this->CountVector();
            $this->CountRank();
            return $this;
        }

        public function GreatAlternative(){
            return $this->great_value;
        }

        public $criterias = array(); // from config
        public $alternatives = array(); // from config
        public $alternativesConversion = array();
        public $inputs = array(); // from config
        public $weight_repaired = array();
        public $vectors = array();
        public $ranks = array();
        public $great_value = 0;
        public $calculate = false; // from config
        public $str_calculate = "";

        public function ConvertAlternatives(){
            $this->alternativesConversion = array();
            $criteria = array_column($this->criterias,'key');
            foreach($this->alternatives as $al){
                $row = array();
                foreach($al as $key => $val){
                    if(in_array($key,$criteria)){
                        $row[$key] = $this-
>ConvertAlternativesValue($key,$val);
                    }else{

```

```

        $row[$key] = $val;
    }
}
array_push($this->alternativesConversion,$row);
}

if($this->calculate){
    // original alternatives
    $this->str_calculate .= "<h4 style='margin-bottom:10px;'>Data
Alternative</h4>";
    $this->str_calculate .= "<table style='border-collapse:collapse;
border:1px solid; width:100%;' border='1'>";
    $this->str_calculate .= "<tr style='background:#CCC;'>";
    $this->str_calculate .= "<th style='text-align:left; padding-
left:5px;'>ID Homestay</th>";
    $this->str_calculate .= "<th style='width:200px; text-
align:left; padding-left:5px;'>Nama Homestay</th>";
    $this->str_calculate .= "<th style='width:100px; text-
align:left; padding-left:5px;'>Harga</th>";
    $this->str_calculate .= "<th style='text-align:left; padding-
left:5px;'>Jarak Pusat Kegiatan</th>";
    $this->str_calculate .= "<th style='text-align:left; padding-
left:5px;'>Jumlah Kamar</th>";
    $this->str_calculate .= "<th style='text-align:left; padding-
left:5px;'>Kepadatan Penduduk</th>";
    $this->str_calculate .= "<th style='text-align:left; padding-
left:5px;'>Fasilitas Wifi</th>";
    $this->str_calculate .= "<th style='text-align:left; padding-
left:5px;'>Fasilitas Pets</th>";
    $this->str_calculate .= "</tr>";
    foreach($this->alternatives as $al){
        $this->str_calculate .= "<tr>";
        $this->str_calculate .= "<td style='text-align:left;
padding-left:5px;'>".$al['key']."</td>";
        $this->str_calculate .= "<td style='text-align:left;
padding-left:5px;'>".$al['nama_homestay']."</td>";
        $this->str_calculate .= "<td style='text-align:left;
padding-left:5px;'>".$al['harga']."</td>";
        $this->str_calculate .= "<td style='text-align:left;
padding-left:5px;'>".$al['jarak_pusat_kegiatan']."</td>";
        $this->str_calculate .= "<td style='text-align:left;
padding-left:5px;'>".$al['jumlah_kamar']."</td>";
        $this->str_calculate .= "<td style='text-align:left;
padding-left:5px;'>".$al['kepadatan_penduduk']."</td>";
        $this->str_calculate .= "<td style='text-align:left;
padding-left:5px;'>".$al['fasilitas_wifi']."</td>";
        $this->str_calculate .= "<td style='text-align:left;
padding-left:5px;'>".$al['fasilitas_hewan_peliharaan']."</td>";
        $this->str_calculate .= "</tr>";
    }
    $this->str_calculate .= "</table>";
}

```

```

// rule conversion
$criteria_column = array_column($this->criteria,'key');

$this->str_calculate .= "<h4 style='margin-bottom:10px;'>Aturan
Konversi</h4>";
$this->str_calculate .= "<div style='width:100%'>";
foreach($this->inputs as $keyIn => $input){
    $idx =
array_search($keyIn,$criteria_column);
    $criteriaText = $this->criteria[$idx]['text'];

    $this->str_calculate .= "<div style='width:24%; float:left;
margin:20px 20px;'>";
        $this->str_calculate .= "<table style='border-
collapse:collapse; border:1px solid; width:90%;' border='1'>";
            $this->str_calculate .= "<tr
style='background:#CCC;'>";
                $this->str_calculate .= "<th
style='text-align:center; padding-left:5px;' colspan='2'>$criteriaText</th>";
                $this->str_calculate .= "</tr>";
                $this->str_calculate .= "<tr
style='background:#CCC;'>";
                    $this->str_calculate .= "<th
style='text-align:left; padding-left:5px;'>Range</th>";
                    $this->str_calculate .= "<th
style='text-align:left; padding-left:5px;'>Nilai</th>";
                    $this->str_calculate .= "</tr>";
                    foreach($input as $key => $val){
                        $this->str_calculate .= "<tr>";
                        $this->str_calculate .= "<td
style='text-align:left; padding-left:5px;'>$val</td>";
                        $this->str_calculate .= "<td
style='text-align:left; padding-left:5px;'>$key</td>";
                        $this->str_calculate .= "</tr>";
                    }
                $this->str_calculate .= "</table>";
            $this->str_calculate .= "</div>";
        }
    $this->str_calculate .= "</div>";
    $this->str_calculate .= "<div style='clear:both;'></div>";

// converted alternatives
$this->str_calculate .= "<h4 style='margin-bottom:10px;'>Hasil
Konversi Data Alternative</h4>";
$this->str_calculate .= "<table style='border-collapse:collapse;
border:1px solid; width:100%;' border='1'>";
    $this->str_calculate .= "<tr style='background:#CCC;'>";
        $this->str_calculate .= "<th style='text-align:left; padding-
left:5px;'>ID Homestay</th>";
        $this->str_calculate .= "<th style='width:200px; text-

```

```

align:left; padding-left:5px;'>Nama Homestay</th>";
        $this->str_calculate .= "<th style='width:100px; text-
align:left; padding-left:5px;'>Harga</th>";
        $this->str_calculate .= "<th style='text-align:left; padding-
left:5px;'>Jarak Pusat Kegiatan</th>";
        $this->str_calculate .= "<th style='text-align:left; padding-
left:5px;'>Jumlah Kamar</th>";
        $this->str_calculate .= "<th style='text-align:left; padding-
left:5px;'>Kepadatan Penduduk</th>";
        $this->str_calculate .= "<th style='text-align:left; padding-
left:5px;'>Fasilitas Wifi</th>";
        $this->str_calculate .= "<th style='text-align:left; padding-
left:5px;'>Fasilitas Pets</th>";
        $this->str_calculate .= "</tr>";
        foreach($this->alternativesConversion as $al){
            $this->str_calculate .= "<tr>";
            $this->str_calculate .= "<td style='text-align:left;
padding-left:5px;'>".$al['key']."</td>";
            $this->str_calculate .= "<td style='text-align:left;
padding-left:5px;'>".$al['nama_homestay']."</td>";
            $this->str_calculate .= "<td style='text-align:left;
padding-left:5px;'>".$al['harga']."</td>";
            $this->str_calculate .= "<td style='text-align:left;
padding-left:5px;'>".$al['jarak_pusat_kegiatan']."</td>";
            $this->str_calculate .= "<td style='text-align:left;
padding-left:5px;'>".$al['jumlah_kamar']."</td>";
            $this->str_calculate .= "<td style='text-align:left;
padding-left:5px;'>".$al['kepadatan_penduduk']."</td>";
            $this->str_calculate .= "<td style='text-align:left;
padding-left:5px;'>".$al['fasilitas_wifi']."</td>";
            $this->str_calculate .= "<td style='text-align:left;
padding-left:5px;'>".$al['fasilitas_hewan_peliharaan']."</td>";
            $this->str_calculate .= "</tr>";
        }
        $this->str_calculate .= "</table>";
    }
}

private function ConvertAlternativesValue($key,$oriValue){
    $criteria = $this->inputs[$key];
    foreach($criteria as $key => $val){
        //echo $val."<br />";
        if(preg_match("/(<|>)/", $val)){
            $split = explode(" ",$val);
            //echo "<pre>", print_r($split), "</pre>";
            if(count($split) == 2){
                if(eval("return ".$oriValue.$val.";")){
                    //echo $bilangan.", Bobot = $key<br />";
                    return $key;
                }
            }
        }else{

```

```

        if(strtoupper($val) == strtoupper($oriValue)){
            return $key;
        }
    }
}

return null;
}

private function WeightRepair() { // changed !!!
    $sum_criterias = array_sum(array_column($this->
    >criterias,'weight'));
    //echo "<pre>", print_r($weight_criterias), "</pre>";
    $this->weight_repaired = array();
    foreach($this->criterias as $c){
        $key = $c['key'];
        $temp_val = $c['weight']/$sum_criterias;
        if(strtolower($c['state'])!="benefit"){
            $temp_val = -abs($temp_val);
        }
        $this->weight_repaired[$key] = $temp_val;
    }
    //echo "<pre>", print_r(($this->criterias), "</pre>";
    if($this->calculate){
        $criteria_keys = array_column($this->criterias,'key');
        $f_pembagi = implode(" +
",array_column($this->criterias,'weight'));
        $this->str_calculate .= "<h4 style='margin-bottom:10px;'>Perbaikan
        Bobot</h4>";
        $this->str_calculate .= "<table style='border-collapse:collapse;
        border:1px solid; width:100%;' border='1'>";
        $this->str_calculate .= "<tr style='background:#CCC;'>";
        $this->str_calculate .= "<th style='width:300px; text-
        align:left; padding-left:5px;'>Kriteria</th>";
        $this->str_calculate .= "<th style='text-align:left; padding-
        left:5px;'>Perbaikan Bobot</th>";
        $this->str_calculate .= "<th style='text-align:left; padding-
        left:5px;'>Hasil Perbaikan Bobot</th>";
        $this->str_calculate .= "<th style='text-align:left; padding-
        left:5px;'>Status</th>";
        $this->str_calculate .= "<th style='text-align:left; padding-
        left:5px;'>Pangkat</th>";
        $this->str_calculate .= "</tr>";
        $i = -1;
        foreach($this->weight_repaired as $key => $val){
            $i++;
            $idx = array_search($key,$criteria_keys);
            $state = $this->criterias[$idx]['state'];
            $bobot = $this->criterias[$i]['weight'];

```

```

        $this->str_calculate .= "<tr>";
        $this->str_calculate .= "<td style='text-align:left;
padding-left:5px;'>".ucwords(str_replace("_","",$key))."</td>";
        $this->str_calculate .= "<td style='text-align:left;
padding-left:5px;'> $bobot / ( $f_pembagi ) </td>";
        $this->str_calculate .= "<td style='text-align:left;
padding-left:5px;'>".(abs($val))."</td>";
        $this->str_calculate .= "<td style='text-align:left;
padding-left:5px;'>".ucfirst($state)."</td>";
        $this->str_calculate .= "<td style='text-align:left;
padding-left:5px;'>".($val)."</td>";
        $this->str_calculate .= "</tr>";
    }
    $this->str_calculate .= "</table>";
}
}
private function CountVector(){
    $criterias = array_column($this->criterias,'key'); // changed
!!
    $this->vectors = array();
    foreach($this->alternativesConversion as $a){
        $key = $a['key'];
        $temp_val = array();
        foreach($criterias as $c){
            $temp_val[$c] = array(
                "text" => ($a[$c])." ^ ". $this-
>weight_repaired[$c],
                "value" => pow(($a[$c]),$this-
>weight_repaired[$c]),
            );
            $value =
array_product(array_column($temp_val,'value'));
            $this->vectors[$key] = array(
                'counter' => $temp_val,
                'value' => round($value,5),
            );
        }
    }
    if($this->calculate){
        $this->str_calculate .= "<h4 style='margin-
bottom:10px;'>Menghitung Vector S</h4>";
        $this->str_calculate .= "<table style='border-collapse:collapse;
border:1px solid; width:100%;' border='1'>";
        $this->str_calculate .= "<tr style='background:#CCC;'>";
        $this->str_calculate .= "<th style='text-align:left; padding-left:5px;'>Si</th>";
        $this->str_calculate .= "<th style='text-align:left; padding-
left:5px;'>Perhitungan</th>";
        $this->str_calculate .= "<th style='text-align:left; padding-left:5px;'>Hasil</th>";
    }
}

```

```

        $this->str_calculate .= "</tr>";
        $i = 0;
        foreach ($this->vectors as $v) {
            $i++;
            $txt = implode(" * (" ,array_column($v['counter'],'text'));

            $this->str_calculate      .= "<tr>";
            $this->str_calculate      .= "<td style='text-
align:left; padding-left:5px;'>S$i</td>";
            $this->str_calculate      .= "<td style='text-
align:left; padding-left:5px;'>($txt)</td>";
            $this->str_calculate      .= "<td style='text-
align:left; padding-left:5px;'>".round($v['value'],5)."</td>";
            $this->str_calculate      .= "</tr>";
        }
        $this->str_calculate      .= "</table>";
    }
}

private function CountRank(){
    $sum_normalisasi      = array_sum(array_column($this->vectors,'value'));
    $this->ranks            = array();
    foreach($this->vectors as $key => $v){
        $this->ranks[$key] = $v['value']/$sum_normalisasi;
    }
    //echo "<pre>", print_r($this->ranks), "</pre>";
    arsort($this->ranks);
    //echo "<pre>", print_r($this->ranks), "</pre>";
    $keys = array_keys($this->ranks);
    $this->great_value = array_shift($keys); // get first value
    if($this->calculate){
        $criteria_keys      = array_column($this->criteria,'key');
        $f_pembagi          = implode(" +
",array_column($this->vectors,'value'));
        $alternative_keys = array_column($this->alternatives,'key');
        $this->str_calculate .= "<h4 style='margin-bottom:10px;'>Menghitung Preferensi ( Vi )</h4>";
        $this->str_calculate .= "<table style='border-collapse:collapse;
border:1px solid; width:100%;' border='1'>";
        $this->str_calculate .= "<tr style='background:#CCC;'>";
        $this->str_calculate .= "<th style='text-align:left; padding-
left:5px;'>Vi</th>";
        $this->str_calculate .= "<th style='text-align:left; padding-
left:5px;'>Perhitungan</th>";
        $this->str_calculate .= "<th style='text-align:left; padding-
left:5px;'>Hasil</th>";
        $this->str_calculate .= "</tr>";
        $str_temp          = "<h4 style='margin-bottom:10px;'>Preferensi ( Vi ) dengan id key</h4>";
        $str_temp          .= "<table style='border-collapse:collapse;
border:1px solid; width:100%;' border='1'>";
        $str_temp          .= "<tr style='background:#CCC;'>";

```

```

$str_temp      .= "<th style='text-align:left; padding-left:5px;'>Vi</th>";
                $str_temp      .= "<th style='text-align:left; padding-
left:5px;'>Nilai V</th>";
                $str_temp      .= "<th style='text-align:left; padding-
left:5px;'>ID KEY</th>";
                $str_temp      .= "<th style='text-align:left; padding-
left:5px;'>Nama Homestay</th>";
                $str_temp      .= "</tr>";
$i = 0;
                $keyTemp = array();
                foreach($this->vectors as $key => $v){
                    $i++;
                    $this->str_calculate .= "<tr>";
                    $this->str_calculate .= "<td style='text-align:left;
padding-left:5px;'>V$i</td>";
                    $this->str_calculate .= "<td style='text-align:left;
padding-left:5px;'>".$v['value']." / ( $f_pembagi ) </td>";
                    $this->str_calculate .= "<td style='text-align:left;
padding-left:5px;'>".round($this->ranks[$key],5)."</td>";
                    $this->str_calculate .= "</tr>";
                    $idx      = array_search($key, $alternative_keys);
                    $data      = $this->alternatives[$idx];
                    $str_temp .= "<tr>";
                    $str_temp      .= "<td style='text-align:left;
padding-left:5px;'>V$i</td>";
                    $str_temp      .= "<td style='text-align:left;
padding-left:5px;'>".round($this->ranks[$key],5)."</td>";
                    $str_temp      .= "<td style='text-align:left;
padding-left:5px;'>$key</td>";
                    $str_temp      .= "<td style='text-align:left;
padding-left:5px;'>".$data['nama_homestay']."</td>";
                    $str_temp      .= "</tr>";
                    $keyTemp[$key] = "V$i";
                }
                $this->str_calculate .= "</table>";
                $str_temp      .= "</table>";

                $this->str_calculate      .= $str_temp;

                $this->str_calculate      .= "<h4 style='margin-bottom:10px;'>Preferensi
( Vi ) dengan id key sorting descending</h4>";
                $this->str_calculate      .= "<table style='border-collapse:collapse;
border:1px solid; width:100%;' border='1'>";
                $this->str_calculate      .= "<tr style='background:#CCC;'>";

```

```

$this->str_calculate .= "<th style='text-align:left; padding-left:5px;'>Vi</th>";
                $this->str_calculate .= "<th style='text-align:left; padding-
left:5px;'>Nilai V</th>";
                $this->str_calculate .= "<th style='text-align:left; padding-
left:5px;'>ID KEY</th>";
                $this->str_calculate .= "<th style='text-align:left; padding-
left:5px;'>Nama Homestay</th>";
                $this->str_calculate .= "</tr>";
                foreach($this->ranks as $key => $v){
                    $idx    = array_search($key, $alternative_keys);
                    $data    = $this->alternatives[$idx];

                    $this->str_calculate .= "<tr>";
                    $this->str_calculate .= "<td style='text-align:left;
padding-left:5px;'>".$keyTemp[$key]."</td>";
                    $this->str_calculate .= "<td style='text-align:left;
padding-left:5px;'>".round($this->ranks[$key],5)."</td>";
                    $this->str_calculate .= "<td style='text-align:left;
padding-left:5px;'> ".$key." </td>";
                    $this->str_calculate .= "<td style='text-align:left;
padding-left:5px;'> ".$data['nama_homestay']." </td>";
                    $this->str_calculate .= "</tr>";
                }
                $this->str_calculate    .= "</table><br /><br />";
            }
        }

    public function Greate(){
        $alternativesKeys = array_column($this->alternatives,'key');
        $idx = array_search($this->great_value,$alternativesKeys);
        echo $this->alternatives[$idx]['nama_homestay'];
    }
}

```

LAMPIRAN E. DATA PENGUJI *BLACK BOX*

No.	Nama	Pekerjaan	Jenis Kelamin
1.	Bapak Supo	Penggagas Komunitas Belajar Tanoker	L
2.	Ibu Farha Ciciek	Penggagas Komunitas Belajar Tanoker	P
3.	Ira	Pengajar di Komunitas Belajar Tanoker	P



LAMPIRAN F. *BLACK BOX TESTING*

No.	Fitur	Action	Hasil yang Diharapkan	Kesimpulan
1.	<i>Login</i>	○ <i>Input username dan password yang benar, kemudian klik login.</i>	○ <i>Login sukses dan masuk sesuai hak akses.</i>	
		○ <i>Login sebagai super admin Username : admin Password : admin</i>	○ <i>Masuk ke sistem sebagai super admin</i>	
		○ <i>Login sebagai admin Username : admin_kasir Password : admin</i>	○ <i>Masuk ke sistem sebagai admin</i>	[√] Berhasil [] Gagal
		○ <i>Login sebagai pemilik homestay Username : owner1@gmail.com Password : admin</i>	○ <i>Masuk ke sistem sebagai pemilik homestay</i>	
		○ <i>username atau password salah</i>	○ <i>Menampilkan pesan username atau password salah</i>	[√] Berhasil [] Gagal
2.	<i>Logout</i>	○ <i>Field username atau password kosong</i>	○ <i>Menampilkan notifikasi This field is required</i>	
		○ <i>Klik menu logout</i>	○ <i>User keluar dari sistem, keluar dari hak akses yang dimiliki dan kembali ke halaman login.</i>	[√] Berhasil [] Gagal
3.	<i>Register</i>	○ <i>Inputkan username, email, password dan confirm password dengan benar, kemudian klik tombol register.</i>	○ <i>Data masuk ke database dan menampilkan pesan sukses simpan data.</i>	[√] Berhasil [] Gagal
		○ <i>Inputkan data tidak valid (email atau confirm password)</i>	○ <i>Menampilkan notifikasi gagal simpan data</i>	
		○ <i>Salah satu form kosong</i>	○ <i>Menampilkan notifikasi This field is required</i>	[√] Berhasil [] Gagal
		○ <i>Email sudah terpakai sebelumnya</i>	○ <i>Menampilkan notifikasi email sudah dipakai</i>	
4.	<i>Ganti password</i>	○ <i>Inputkan password dan confirm new password yang benar kemudian klik simpan</i>	○ <i>Password di database terupdate dan menampilkan pesan sukses ubah password</i>	[√] Berhasil [] Gagal
		○ <i>Inputkan old password salah</i>	○ <i>Menampilkan pesan password lama tidak</i>	[√] Berhasil [] Gagal

lanjutan

		<ul style="list-style-type: none"> o <i>Inputkan confirm new password</i> tidak sama dengan <i>new password</i> o Salah satu field kosong 	<p>sama</p> <ul style="list-style-type: none"> o Menampilkan pesan <i>Please enter the same value again.</i> o Menampilkan notifikasi <i>This field is required</i> 	
5.	Edit Profile	<ul style="list-style-type: none"> o <i>Inputkan username, nama lengkap, email, no hp, no ktp, dan alamat</i> o Salah satu field kosong o <i>Input</i> data tidak berhasil 	<ul style="list-style-type: none"> o Data akun diupdate ke <i>database</i> dan menampilkan pesan sukses ubah profile o Menampilkan notifikasi <i>This field is required</i> o Menampilkan notifikasi gagal ubah profile 	<p>[√] Berhasil [] Gagal</p> <p>[√] Berhasil [] Gagal</p>
6.	Pencarian Lokasi	<ul style="list-style-type: none"> o Pilih subkriteria jalan, jenis reklame insidentil, perkiraan pajak, kepadatan kendaraan, radius perdagangan, radius pendidikan dan luas reklame yang benar kemudian klik submit o Klik <i>view</i> lokasi o Salah satu field yang harus diisi dikosongi o Klik tombol <i>reset</i> o Klik tombol kembali 	<ul style="list-style-type: none"> o Menampilkan lokasi reklame yang direkomendasikan beserta penilaiannya. o Menampilkan titik lokasi reklame yang dipilih dalam <i>Google Maps</i> o Menampilkan notifikasi field harus diisi o Mengosongkan semua field pencarian lokasi o Menampilkan kembali halaman home page <i>member</i> 	<p>[√] Berhasil [] Gagal</p> <p>[√] Berhasil [] Gagal</p>
7.	Accept Daftar <i>Homestay</i> Pending	<ul style="list-style-type: none"> o Klik konfirmasi kemudian pilih setuju atau tolak 	<ul style="list-style-type: none"> o Menampilkan notifikasi sukses simpan konfirmasi 	<p>[√] Berhasil [] Gagal</p>
8.	Kelola Registrasi Pemilik <i>Homestay</i>	<ul style="list-style-type: none"> o Klik konfirmasi kemudian pilih setuju o Klik konfirmasi kemudian pilih cancel 	<ul style="list-style-type: none"> o Menampilkan notifikasi sukses simpan konfirmasi o Menampilkan kembali halaman Daftar Owner <i>Homestay</i> 	<p>[√] Berhasil [] Gagal</p> <p>[√] Berhasil [] Gagal</p>
9.	Kelola <i>User</i> Aplikasi	<ul style="list-style-type: none"> o Klik aktifkan dan memilih aktifkan o Klik hapus dan memilih oke o Klik non aktifkan dan memilih non aktifkan o Klik hapus dan 	<ul style="list-style-type: none"> o Menampilkan notifikasi Sukses ubah status <i>user</i> o Menampilkan notifikasi Sukses hapus <i>user</i> o Menampilkan notifikasi Sukses ubah status <i>user</i> o Menampilkan kembali halaman Daftar Owner 	<p>[√] Berhasil [] Gagal</p> <p>[√] Berhasil [] Gagal</p>

dilanjutkan

lanjutan

	memilih batal	<i>Homestay</i>	
10. Tambah <i>User</i> Aplikasi	o Mengisi form tambah <i>user</i> dan klik simpan	o Menyimpan data <i>user</i> ke database dan menampilkan notifikasi Sukses Simpan <i>User</i>	[√] Berhasil [] Gagal
	o Salah satu field kosong o <i>Inputan username</i> sudah digunakan	o Menampilkan notifikasi <i>This field is required</i> o Menampilkan notifikasi <i>Username</i> sudah dipakai	[√] Berhasil [] Gagal
11. Accept Daftar Booking Baru	o Klik konfirmasi kemudian pilih konfirmasi	o Menampilkan notifikasi sukses simpan konfirmasi	[√] Berhasil [] Gagal
12. Booking <i>Homestay</i>	o <i>Input</i> form booking dan klik Simpan	o Menampilkan notifikasi Sukses Simpan Booking Dikeranjang	[√] Berhasil [] Gagal
	o <i>Input</i> form order dan klik Simpan	o Menampilkan notifikasi Kami telah mengirim pesan ke email anda. Mohon cek inbox email	
	o Gagal <i>input</i> form booking dan klik Simpan	o Menampilkan notifikasi Gagal Simpan Booking Dikeranjang	
	o <i>Input</i> tanggal check out kurang dari tanggal check in o Kuota kamar telah habis	o Menampilkan notifikasi Mohon cek kembali tanggal check in dan check out nya o Menampilkan notifikasi Quota Kamar <i>Homestay</i> tidak mencukupi pada tanggal (check in dan check out nya)	[√] Berhasil [] Gagal
	o Tidak ada koneksi dengan jaringan internet atau email tidak sesuai o Gagal saat melakukan booking <i>homestay</i>	o Menampilkan notifikasi Terjadi kesalahan ketika kami mengirim pesan ke email anda o Menampilkan notifikasi Gagal Simpan Booking <i>Homestay</i>	
13. Tambah <i>Homestay</i>	o Mengisi form tambah <i>user</i> dan klik simpan	o Menyimpan data form ke dalam database dan menampilkan notifikasi Sukses Simpan <i>Homestay</i>	[√] Berhasil [] Gagal
	o Menginputkan data form tambah <i>homestay</i> dengan data yang benar dan klik Simpan namun data gagal disimpan o Salah satu field	o Menampilkan notifikasi Gagal Simpan <i>Homestay</i> o Menampilkan notifikasi <i>This field is required</i>	[√] Berhasil [] Gagal

d lanj utkan

lanjutan

		yang harus diisi dikosongi		
14.	Edit Daftar <i>Homestay</i>	<ul style="list-style-type: none"> ○ Klik Edit pada field aksi di halaman Daftar <i>Homestay</i> ○ Edit data form ubah <i>homestay</i> dengan data yang benar dan klik Simpan 	<ul style="list-style-type: none"> ○ Menampilkan halaman Form ubah <i>homestay</i> ○ Menyimpan data form ubah <i>homestay</i> kedalam <i>database</i> dan menampilkan pesan Sukses Ubah <i>Homestay</i> 	<ul style="list-style-type: none"> [<input checked="" type="checkbox"/>] Berhasil [<input type="checkbox"/>] Gagal
		<ul style="list-style-type: none"> ○ Edit data form ubah <i>homestay</i> dengan data yang benar dan klik Simpan namun data gagal disimpan ○ Salah satu field yang harus diisi dikosongi 	<ul style="list-style-type: none"> ○ Menampilkan pesan data gagal disimpan ○ Menampilkan notifikasi <i>This field is required</i> 	<ul style="list-style-type: none"> [<input checked="" type="checkbox"/>] Berhasil [<input type="checkbox"/>] Gagal
15.	Hapus Daftar <i>Homestay</i>	<ul style="list-style-type: none"> ○ Klik Hapus pada field aksi di halaman Daftar <i>Homestay</i> dan klik Oke 	<ul style="list-style-type: none"> ○ Menghapus data <i>homestay</i> dan menampilkan notifikasi Sukses Hapus <i>Homestay</i> 	<ul style="list-style-type: none"> [<input checked="" type="checkbox"/>] Berhasil [<input type="checkbox"/>] Gagal
		<ul style="list-style-type: none"> ○ Klik Hapus pada field aksi di halaman Daftar <i>Homestay</i> dan klik Batal 	<ul style="list-style-type: none"> ○ Kembali ke halaman Daftar <i>Homestay</i> 	<ul style="list-style-type: none"> [<input checked="" type="checkbox"/>] Berhasil [<input type="checkbox"/>] Gagal
16.	Upload Foto <i>Homestay</i>	<ul style="list-style-type: none"> ○ Klik Upload pada field aksi di halaman Daftar <i>Homestay</i> dan klik Oke 	<ul style="list-style-type: none"> ○ Menampilkan halaman Form upload photo <i>homestay</i> 	<ul style="list-style-type: none"> [<input checked="" type="checkbox"/>] Berhasil [<input type="checkbox"/>] Gagal

LAMPIRAN G. FORM PENGUJIAN *BLACK BOX TESTING*

**SISTEM INFORMASI DAN REKOMENDASI
PEMILIHAN *HOMESTAY* DI KAWASAN DESA
WISATA BAGI WISATAWAN MENGGUNAKAN
METODE *WEIGHTED PRODUCT* (STUDI KASUS :
DESA WISATA TANOKER KECAMATAN
LEDOKOMBO KABUPATEN JEMBER)**

Kepada

Yth. Saudara Penguji

di tempat

Dengan Hormat,

Berkaitan dengan kegiatan penelitian yang saya lakukan dengan judul **SISTEM INFORMASI DAN REKOMENDASI PEMILIHAN *HOMESTAY* DI KAWASAN DESA WISATA BAGI WISATAWAN MENGGUNAKAN METODE *WEIGHTED PRODUCT* (STUDI KASUS : DESA WISATA TANOKER KECAMATAN LEDOKOMBO KABUPATEN JEMBER)**, sebagai salah satu syarat untuk memperoleh gelar Sarjana Sistem Informasi pada Program Studi Sistem Informasi Universitas Jember, maka dengan ini saya mengharapkan bantuan saudara untuk mengisi daftar pernyataan pengujian sistem yang saya sertakan di bawah ini.

Agar memperoleh masukan yang berarti, saya berharap pengujian ini diisi sesuai dengan keadaan yang sebenarnya. Atas perhatian dan bantuan saya ucapkan terima kasih.

Hormat Saya,

Peneliti

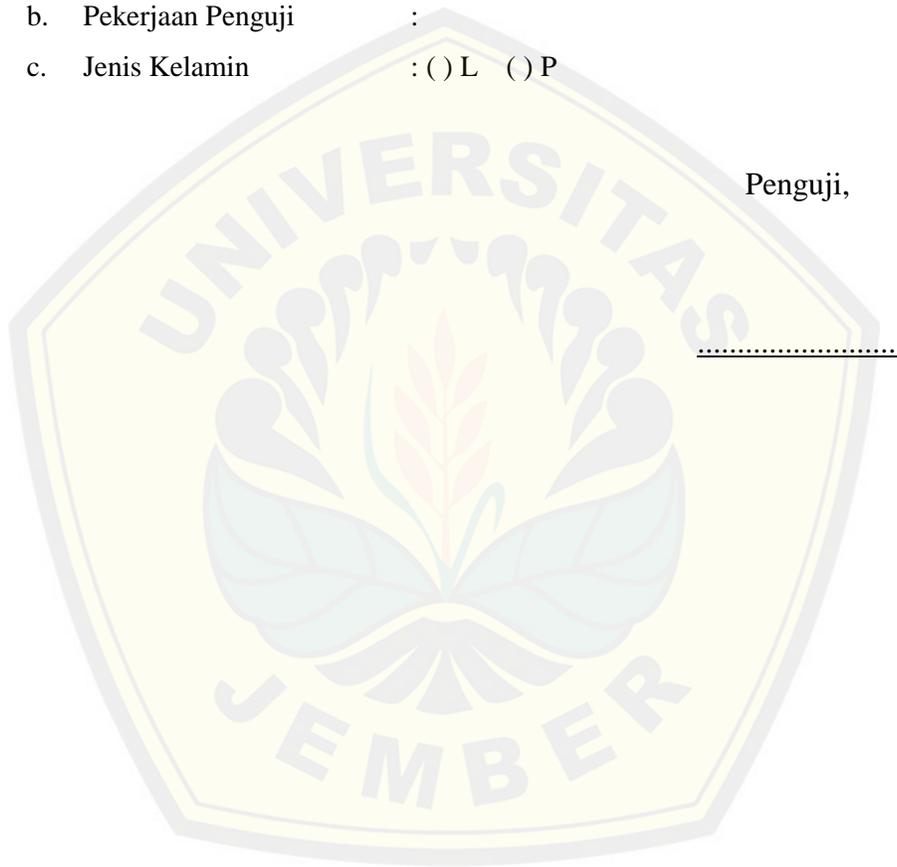
1. Petunjuk Pengisian

Berilah tanda cek (\checkmark) pada jawaban yang sesuai dengan keadaan yang sebenarnya (berhasil/gagal) dan beri keterangan bila ada sistem yang gagal dalam proses pengujian.

2. Identitas Penguji

- No. Penguji (diisi peneliti) :
a. Nama Penguji :
b. Pekerjaan Penguji :
c. Jenis Kelamin : () L () P

Penguji,



No.	Fitur	Action	Hasil yang Diharapkan	Kesimpulan
1.	Login	o <i>Input username dan password yang benar, kemudian klik login.</i>	o <i>Login sukses dan masuk sesuai hak akses.</i>	
		o <i>Login sebagai super admin Username : admin Password : admin</i>	o <i>Masuk ke sistem sebagai super admin</i>	
		o <i>Login sebagai admin Username : admin_kasir Password : admin</i>	o <i>Masuk ke sistem sebagai admin</i>	[] Berhasil [] Gagal
		o <i>Login sebagai pemilik homestay Username : owner1@gmail.com Password : admin</i>	o <i>Masuk ke sistem sebagai pemilik homestay</i>	
		o <i>username atau password salah</i>	o <i>Menampilkan pesan username atau password salah</i>	[] Berhasil [] Gagal
2.	Logout	o <i>Field username atau password kosong</i>	o <i>Menampilkan notifikasi This field is required</i>	
		o <i>Klik menu logout</i>	o <i>User keluar dari sistem, keluar dari hak akses yang dimiliki dan kembali ke halaman login.</i>	[] Berhasil [] Gagal
3.	Register	o <i>Inputkan username, email, password dan confirm password dengan benar, kemudian klik tombol register.</i>	o <i>Data masuk ke database dan menampilkan pesan sukses simpan data.</i>	[] Berhasil [] Gagal
		o <i>Inputkan data tidak valid (email atau confirm password)</i>	o <i>Menampilkan notifikasi gagal simpan data</i>	
		o <i>Salah satu form kosong</i>	o <i>Menampilkan notifikasi This field is required</i>	[] Berhasil [] Gagal
		o <i>Email sudah terpakai sebelumnya</i>	o <i>Menampilkan notifikasi email sudah dipakai</i>	
4.	Ganti password	o <i>Inputkan password dan confirm new password yang benar kemudian klik simpan</i>	o <i>Password di database terupdate dan menampilkan pesan sukses ubah password</i>	[✓] Berhasil [] Gagal
		o <i>Inputkan old password salah</i>	o <i>Menampilkan pesan password lama tidak sama</i>	[] Berhasil [] Gagal

dilanjutkan

lanjutan

		<ul style="list-style-type: none"> o <i>Inputkan confirm new password</i> tidak sama dengan <i>new password</i> o Salah satu field kosong 	<ul style="list-style-type: none"> o Menampilkan pesan <i>Please enter the same value again.</i> o Menampilkan notifikasi <i>This field is required</i> 	
5.	Edit Profile	<ul style="list-style-type: none"> o <i>Inputkan username, nama lengkap, email, no hp, no ktp, dan alamat</i> o Salah satu field kosong o <i>Input</i> data tidak berhasil 	<ul style="list-style-type: none"> o Data akun diupdate ke <i>database</i> dan menampilkan pesan sukses ubah profile o Menampilkan notifikasi <i>This field is required</i> o Menampilkan notifikasi gagal ubah profile 	<ul style="list-style-type: none"> [] Berhasil [] Gagal [] Berhasil [] Gagal
6.	Pencarian Lokasi	<ul style="list-style-type: none"> o Pilih subkriteria jalan, jenis reklame insidental, perkiraan pajak, kepadatan kendaraan, radius perdagangan, radius pendidikan dan luas reklame yang benar kemudian klik submit o Klik <i>view</i> lokasi o Salah satu field yang harus diisi dikosongi o Klik tombol <i>reset</i> o Klik tombol kembali 	<ul style="list-style-type: none"> o Menampilkan lokasi reklame yang direkomendasikan beserta penilaiannya. o Menampilkan titik lokasi reklame yang dipilih dalam <i>Google Maps</i> o Menampilkan notifikasi field harus diisi o Mengosongkan semua field pencarian lokasi o Menampilkan kembali halaman home page <i>member</i> 	<ul style="list-style-type: none"> [] Berhasil [] Gagal [] Berhasil [] Gagal
7.	Accept Daftar <i>Homestay</i> Pending	<ul style="list-style-type: none"> o Klik konfirmasi kemudian pilih setuju atau tolak 	<ul style="list-style-type: none"> o Menampilkan notifikasi sukses simpan konfirmasi 	<ul style="list-style-type: none"> [] Berhasil [] Gagal
8.	Kelola Registrasi Pemilik <i>Homestay</i>	<ul style="list-style-type: none"> o Klik konfirmasi kemudian pilih setuju o Klik konfirmasi kemudian pilih cancel 	<ul style="list-style-type: none"> o Menampilkan notifikasi sukses simpan konfirmasi o Menampilkan kembali halaman Daftar Owner <i>Homestay</i> 	<ul style="list-style-type: none"> [] Berhasil [] Gagal [] Berhasil [] Gagal
9.	Kelola <i>User</i> Aplikasi	<ul style="list-style-type: none"> o Klik aktifkan dan memilih aktifkan o Klik hapus dan memilih oke o Klik non aktifkan dan memilih non aktifkan o Klik hapus dan memilih batal 	<ul style="list-style-type: none"> o Menampilkan notifikasi Sukses ubah status <i>user</i> o Menampilkan notifikasi Sukses hapus <i>user</i> o Menampilkan notifikasi Sukses ubah status <i>user</i> o Menampilkan kembali halaman Daftar Owner <i>Homestay</i> 	<ul style="list-style-type: none"> [] Berhasil [] Gagal [] Berhasil [] Gagal

dilanjutkan

lanjutan

10.	Tambah <i>User</i> Aplikasi	<ul style="list-style-type: none"> ○ Mengisi form tambah <i>user</i> dan klik simpan 	<ul style="list-style-type: none"> ○ Menyimpan data <i>user</i> ke database dan menampilkan notifikasi Sukses Simpan <i>User</i> 	<ul style="list-style-type: none"> [] Berhasil [] Gagal
		<ul style="list-style-type: none"> ○ Salah satu field kosong ○ <i>Inputan username</i> sudah digunakan 	<ul style="list-style-type: none"> ○ Menampilkan notifikasi <i>This field is required</i> ○ Menampilkan notifikasi <i>Username</i> sudah dipakai 	<ul style="list-style-type: none"> [] Berhasil [] Gagal
11.	Accept Daftar Booking Baru	<ul style="list-style-type: none"> ○ Klik konfirmasi kemudian pilih konfirmasi 	<ul style="list-style-type: none"> ○ Menampilkan notifikasi sukses simpan konfirmasi 	<ul style="list-style-type: none"> [] Berhasil [] Gagal
12.	Booking <i>Homestay</i>	<ul style="list-style-type: none"> ○ <i>Input</i> form booking dan klik Simpan 	<ul style="list-style-type: none"> ○ Menampilkan notifikasi Sukses Simpan Booking Dikeranjang 	<ul style="list-style-type: none"> [] Berhasil
		<ul style="list-style-type: none"> ○ <i>Input</i> form order dan klik Simpan 	<ul style="list-style-type: none"> ○ Menampilkan notifikasi Kami telah mengirim pesan ke email anda. Mohon cek inbox email 	<ul style="list-style-type: none"> [] Gagal
		<ul style="list-style-type: none"> ○ Gagal <i>input</i> form booking dan klik Simpan ○ <i>Input</i> tanggal check out kurang dari tanggal check in ○ Kuota kamar telah habis 	<ul style="list-style-type: none"> ○ Menampilkan notifikasi Gagal Simpan Booking Dikeranjang ○ Menampilkan notifikasi Mohon cek kembali tanggal check in dan check out nya ○ Menampilkan notifikasi Quota Kamar <i>Homestay</i> tidak mencukupi pada tanggal (check in dan check out nya) 	<ul style="list-style-type: none"> [] Berhasil [] Gagal
		<ul style="list-style-type: none"> ○ Tidak ada koneksi dengan jaringan internet atau email tidak sesuai ○ Gagal saat melakukan booking <i>homestay</i> 	<ul style="list-style-type: none"> ○ Menampilkan notifikasi Terjadi kesalahan ketika kami mengirim pesan ke email anda ○ Menampilkan notifikasi Gagal Simpan Booking <i>Homestay</i> 	
13.	Tambah <i>Homestay</i>	<ul style="list-style-type: none"> ○ Mengisi form tambah <i>user</i> dan klik simpan 	<ul style="list-style-type: none"> ○ Menyimpan data form ke dalam database dan menampilkan notifikasi Sukses Simpan <i>Homestay</i> 	<ul style="list-style-type: none"> [] Berhasil [] Gagal
		<ul style="list-style-type: none"> ○ Menginputkan data form tambah <i>homestay</i> dengan data yang benar dan klik Simpan namun data gagal disimpan ○ Salah satu field yang harus diisi 	<ul style="list-style-type: none"> ○ Menampilkan notifikasi Gagal Simpan <i>Homestay</i> ○ Menampilkan notifikasi <i>This field is required</i> 	<ul style="list-style-type: none"> [] Berhasil [] Gagal

dilanjutkan

lanjutan

		dikosongi		
14.	Edit Daftar <i>Homestay</i>	<ul style="list-style-type: none"> ○ Klik Edit pada field aksi di halaman Daftar <i>Homestay</i> ○ Edit data form ubah <i>homestay</i> dengan data yang benar dan klik Simpan 	<ul style="list-style-type: none"> ○ Menampilkan halaman Form ubah <i>homestay</i> ○ Menyimpan data form ubah <i>homestay</i> kedalam <i>database</i> dan menampilkan pesan Sukses Ubah <i>Homestay</i> 	<ul style="list-style-type: none"> [] Berhasil [] Gagal
		<ul style="list-style-type: none"> ○ Edit data form ubah <i>homestay</i> dengan data yang benar dan klik Simpan namun data gagal disimpan ○ Salah satu field yang harus diisi dikosongi 	<ul style="list-style-type: none"> ○ Menampilkan pesan data gagal disimpan ○ Menampilkan notifikasi <i>This field is required</i> 	<ul style="list-style-type: none"> [] Berhasil [] Gagal
15.	Hapus Daftar <i>Homestay</i>	<ul style="list-style-type: none"> ○ Klik Hapus pada field aksi di halaman Daftar <i>Homestay</i> dan klik Oke 	<ul style="list-style-type: none"> ○ Menghapus data <i>homestay</i> 	<ul style="list-style-type: none"> [] Berhasil [] Gagal
		<ul style="list-style-type: none"> ○ Klik Hapus pada field aksi di halaman Daftar <i>Homestay</i> dan klik Batal 	<ul style="list-style-type: none"> ○ Kembali ke halaman Daftar <i>Homestay</i> 	<ul style="list-style-type: none"> [] Berhasil [] Gagal
16.	Upload Foto <i>Homestay</i>	<ul style="list-style-type: none"> ○ Klik Upload pada field aksi di halaman Daftar <i>Homestay</i> dan klik Oke 	<ul style="list-style-type: none"> ○ Menampilkan halaman Form upload photo <i>homestay</i> 	<ul style="list-style-type: none"> [] Berhasil [] Gagal