



**SISTEM INFORMASI REKOMENDASI *HOMESTAY* MENGGUNAKAN
METODE *SIMPLE MULTI-ATTRIBUTE RATING TECHNIQUE*
EXPLOITING RANKS (SMARTER)**

(Studi Kasus :Kawah Ijen Kecamatan Licin Kabupaten Banyuwangi)

SKRIPSI

Oleh :

ALFIYATUL HIMMAH

NIM 122410101079

**PROGRAM STUDI SISTEM INFORMASI
UNIVERSITAS JEMBER**

2017



**SISTEM INFORMASI REKOMENDASI *HOMESTAY* MENGGUNAKAN
METODE *SIMPLE MULTI-ATTRIBUTE RATING TECHNIQUE*
EXPLOITING RANKS (SMARTER)
(Studi Kasus :Kawah Ijen Kecamatan Licin Kabupaten Banyuwangi)**

SKRIPSI

Diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat untuk menyelesaikan pendidikan di Program Studi Sistem Informasi Universitas Jember dan mendapat gelar Sarjana Komputer

oleh :

Alfiyatul Himmah

NIM 122410101079

**PROGRAM STUDI SISTEM INFORMASI
UNIVERSITAS JEMBER**

2017

PERSEMBAHAN

Skripsi ini saya persembahkan untuk :

1. Allah SWT, yang telah memberikan kesabaran, kemudahan dan kelancaran dalam mengerjakan skripsi ini.
2. Abah tersayang dan tercinta A. Suhaili beserta Ummi tersayang dan tercinta Miftahur Rohmah.
3. Kakak M. Shahibusy Syafaat sekeluarga, kakak A. Zainu Ramadlan sekeluarga, dan adik M. Farih Mahbubi.
4. Saudara-saudaraku beserta keluarga besar.
5. Sahabat-sahabatku ciwi-ciwi rempong dan formation beserta doa dan dukungannya.
6. Keluarga besar UKMK Etalase.
7. Guru-guruku sejak taman kanak-kanak hingga perguruan tinggi.
8. Almamater Program Studi Sistem Informasi Universitas Jember.

MOTO

“Kalau mengejar makhluk, kita tidak akan meraihnya..
Kalau kita mengejar Allah, apapun yang kita butuhkan akan diberikan..”

“Miliki sifat Qona’ah, maka kamu akan selalu berbahagia”



PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Alfiyatul Himmah

NIM : 122410101079

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Sistem Informasi Rekomendasi *Homestay* Menggunakan Metode *Simple Multi-Attribute Rating Technique Exploiting Ranks* (SMARTER)”, adalah benar-benar hasil karya sendiri, kecuali jika dalam pengutipan substansi disebutkan sumbernya, belum pernah diajukan pada institusi mana pun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

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

Jember, 31 Maret 2017

Yang menyatakan,

Alfiyatul Himmah

NIM 122410101079

SKRIPSI

**Sistem Informasi Rekomendasi *Homestay* Menggunakan Metode *Simple Multi-Attribute Rating Technique Exploiting Ranks* (SMARTER)
(Study Kasus : Kawah Ijen Kecamatan Licin Kabupaten Banyuwangi)**

oleh :

Alfiyatul Himmah

NIM 122410101079

Pembimbing

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

Dosen Pembimbing Pendamping : M. Arif Hidayat S.Kom.,M.Kom.

PENGESAHAN PEMBIMBING

Skripsi berjudul “Sistem Informasi Rekomendasi *Homestay* Menggunakan Metode *Simple Multi-Attribute Rating Technique Exploiting Ranks* (SMARTER) (Study Kasus : Kawah Ijen Kecamatan Licin Kabupaten Banyuwangi)”, telah diuji dan disahkan pada :

Hari, tanggal : Jum’at, 31 Maret 2017

Tempat : Program Studi Sistem Informasi Universitas Jember

Disetujui oleh :

Pembimbing I,

Pembimbing II,

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

M. Arif Hidayat S.Kom.,M.Kom.

NIP. 19670420 1992011001

NIP. 198101232010121003

PENGESAHAN PENGUJI

Skripsi berjudul “Sistem Informasi Rekomendasi *Homestay* Menggunakan Metode *Simple Multi-Attribute Rating Technique Exploiting Ranks* (SMARTER) (Study Kasus : Kawah Ijen Kecamatan Licin Kabupaten Banyuwangi)”, telah diuji dan disahkan pada :

Hari, tanggal : 31 Maret 2017

Tempat : Program Studi Sistem Informasi Universitas Jember

Tim Penguji :

Penguji I,

Penguji II,

Dr. Saiful Bukhori, ST., M.Kom

Fahrobby Adnan, S.Kom., M.MSI

NIP. 196811131994121001

NIP. 198706192014041001

Mengesahkan
Ketua Program Ilmu Komputer

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

NIP. 19670420 1992011001

RINGKASAN

Sistem Informasi Rekomendasi *Homestay* Menggunakan Metode *Simple Multi-Attribute Rating Technique Exploiting Ranks* (SMARTER) ; Alfiyatul Himmah, 122410101079 2016, 131 HALAMAN; Program Studi Sistem Informasi Universitas Jember.

Kawah Ijen merupakan destinasi wisata yang menjadi primadona di Kabupaten Banyuwangi. Kawah Ijen merupakan wisata yang sering dikunjungi dan menjadi daya tarik oleh wisatawan domestik, maupun wisatawan asing karena pemandangan gunung yang indah dan penampakan *bluefire*, yaitu suatu fenomena alam yang terjadi karena lelehan lava pijar berwarna biru yang sangat langka, dan hanya ditemui di beberapa negara di dunia, salah satunya di Kawah Ijen. Wisatawan yang datang untuk berwisata di Kawah Ijen ini tentunya membutuhkan fasilitas *homestay* untuk bermalam sebelum mendaki. *Homestay* yang ada di daerah lereng gunung yaitu di kawasan Kecamatan Licin dikelola oleh Badan Usaha Milik Desa (BUMDES). Para wisatawan masih perlu bertanya kepada pihak BUMDES tentang informasi ketersediaan *homestay* yang ada disekitar lereng gunung. Terkadang pihak BUMDES juga kesulitan dalam memeriksa ketersediaan *homestay* yang ada, karena harus melakukan survei satu persatu *homestay* yang tersedia, sehingga hal tersebut menyulitkan pihak BUMDES dan memakan waktu yang banyak bagi wisatawan dalam memilih *homestay*. Berdasarkan masalah tersebut diperlukan sistem informasi rekomendasi yang ditujukan untuk membantu merekomendasikan informasi *homestay* yang ada di sekitar kawasan Licin. Pada penelitian ini dibangun sistem informasi rekomendasi *homestay* menggunakan kriteria-kriteria yang umum digunakan dalam pencarian *homestay*. Metode yang digunakan dalam sistem ini adalah *Simple Multi-Attribute Rating Technique Exploiting Ranks* (SMARTER). Metode ini digunakan untuk pembobotan kriteria, sub kriteria, dan sub-sub kriteria *homestay* yang ada di

sekitar kawasan Licin. Sistem ini menghasilkan perangkaian *homestay* dari kriteria-kriteria yang telah dipilih oleh wisatawan.



PRAKATA

Puji syukur kehadirat Allah SWT atas segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi dengan judul “Sistem Informasi Rekomendasi *Homestay* Menggunakan Metode *Simple Multi-Attribute Rating Technique Exploiting Ranks* (SMARTER)”. Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan Strata Satu (S1) pada Program Studi Sistem Informasi Universitas Jember.

Penyusunan skripsi ini tidak lepas dari dukungan berbagai pihak. Oleh karena itu penulis menyampaikan terima kasih kepada :

1. Prof. Drs. Slamain, M.Comp.Sc., Ph.D., selaku Ketua Program Studi Sistem Informasi Universitas Jember;
2. Prof. Drs. Slamain, M.Comp.Sc., Ph.D., selaku Dosen Pembimbing Utama dan Muhammad Arief Hidayat S.Kom., M.Kom., selaku Dosen Pembimbing
3. Anggota yang telah meluangkan waktu, pikiran, dan perhatian dalam penulisan skripsi;
4. Windi Eka Yulia Retnani, S.Kom., M.T. sebagai dosen pembimbing akademik, yang telah mendampingi penulis sebagai mahasiswa.
5. Seluruh Bapak dan Ibu dosen beserta staf karyawan di Program Studi Sistem Informasi Universitas Jember;
6. Abah tersayang dan tercinta Suhaili beserta Ummi tersayang dan tercinta Miftahur Rohmah yang selalu mendukung, mendoakan dan memperjuangkan penulis dalam mendapatkan gelar sarjana;
7. Kakak M.Shahibusy Syafaat sekeluarga, kakak A. Zainu Ramadlan sekeluarga, adik M. Farih Mahbubi;
8. Saudara-saudaraku beserta keluarga besar;
9. Sahabatku ciwi-rempong dan formation beserta doa dan dukungannya;

10. Guru-guruku sejak taman kanak-kanak hingga perguruan tinggi;
11. Almater Program Studi Sistem Informasi Universitas Jember.

Dengan harapan bahwa penelitian ini nantinya akan terus berlanjut dan berkembang kelak, penulis juga menerima segala kritik dan saran dari semua pihak demi kesempurnaan skripsi ini. Akhirnya penulis berharap, semoga skripsi ini dapat bermanfaat.

Jember, 31 Maret 2017

Penulis

DAFTAR ISI

HALAMAN JUDUL	
SKRIPSI.....	i
PERSEMBAHAN.....	ii
MOTO	iii
PERNYATAAN.....	iv
SKRIPSI.....	v
PENGESAHAN PEMBIMBING.....	vi
PENGESAHAN PENGUJI.....	vii
RINGKASAN	viii
PRAKATA.....	x
DAFTAR ISI.....	xii
DAFTAR TABEL.....	xvi
DAFTAR GAMBAR	xvii
BAB 1. PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	3
1.3. Tujuan.....	3
1.4. Batasan Masalah.....	4
1.5. Sistematika Penulisan.....	4
BAB 2. TINJAUAN PUSTAKA	6
2.1. Penelitian Terdahulu	6
2.2. Sistem Informasi	7
2.3. Sistem Pendukung Pengambilan Keputusan (<i>Decision Support System</i>)	7
2.4. Metode Simple Multi-Attribute Rating Technique Exploiting Rank (SMARTER)	8
2.4.1. Pembobotan Rank Order Centroid (ROC)	9

BAB 3. METODOLOGI PENELITIAN.....	12
3.1. Jenis Penelitian.....	12
3.2. Tempat dan Waktu Penelitian	12
3.3. Tahapan Penelitian	12
3.4. Tahap Pengembangan Sistem.....	13
3.4.1. Analisis Kebutuhan (<i>Requirements</i>).....	14
3.4.2. Desain Sistem (<i>Design</i>).....	18
3.4.3. Implementasi (<i>Implementation</i>)	19
3.4.4. Pengujian (<i>Testing</i>).....	20
BAB 4. PERANCANGAN SISTEM.....	22
4.1. Deskripsi Umum Sistem.....	22
4.1.1. <i>Statement of Purpose</i> (SOP).....	22
4.1.2. Fungsi Sistem	23
4.2. Analisis Kebutuhan Sistem	23
4.2.1. Kebutuhan Fungsional.....	24
4.2.2. Kebutuhan Non-Fungsional	24
4.3. Desain Sistem.....	25
4.3.1. <i>Business Process</i>	25
4.3.2. Usecase Diagram.....	26
4.3.3. Skenario Sistem.....	28
4.3.4. <i>Sequence Diagram</i>	33
4.3.5. Activity Diagram.....	36
4.3.6. <i>Class Diagram</i>	39
4.3.7. <i>Entity Relationship Diagram</i>	41
4.4. Implementasi	41

4.4.1. Kode Program <i>Login</i>	42
4.4.2. Kode Program Kelola Data <i>Homestay</i>	42
4.4.3. Kode Program Ubah Informasi	42
4.4.4. Kode Program Lihat Informasi.....	42
4.4.5. Kode Program Ubah Admin.....	42
4.4.6. Kode Program <i>Booking Homestay</i>	42
4.4.7. Kode Program Ubah Data Pemesan	43
4.4.8. Kode Program Hapus Data Pemesan.....	43
4.4.9. Kode Program Lihat Detail <i>Booking</i>	43
4.4.10. Kode Program Konfirmasi <i>Booking</i>	43
4.4.11. Kode Program Cek Detail Pemesanan	43
4.4.12. Kode Program Mencari <i>Homestay</i>	43
4.4.13. Kode Program Lihat Detail Perhitungan	49
4.5. Pengujian Sistem	49
4.5.1. <i>White Box Testing</i>	50
4.5.2. <i>Black Box Testing</i>	62
BAB 5. HASIL DAN PEMBAHASAN.....	64
5.1. Hasil Pembuatan Sistem Informasi Rekomendasi <i>Homestay</i>	64
5.1.1. Fitur <i>Login</i>	65
5.1.2. Halaman <i>Dashboard</i>	65
5.1.3. Fitur Mengelola <i>Homestay</i>	66
5.1.4. Fitur Mengelola <i>Booking</i>	68
5.1.5. Fitur Cari <i>Homestay</i>	71
5.1.6. Fitur Lihat Detail Perhitungan.....	73

5.1.7. Fitur <i>Booking Homestay</i>	74
5.2. Hasil Implementasi Metode <i>Simple Multi-Attribute Rating Technique Exploiting Ranks</i> (SMARTER).....	75
5.2.1. Data Kriteria, Sub Kriteria dan Sub-Sub Kriteria	76
5.2.2. Perhitungan Manual Metode SMARTER Pada Sistem Informasi Rekomendasi <i>Homestay</i>	77
5.3. Implementasi Metode SMARTER Pada Sistem Informasi Rekomendasi <i>Homestay</i>	84
5.4. Pembahasan Pada Sistem Informasi Rekomendasi <i>Homestay</i>	87
5.4.1. Fitur Sistem	87
5.4.2. Hasil Pegujian.....	88
BAB 6. PENUTUP.....	95
6.1. Kesimpulan.....	95
6.2. Saran.....	96
DAFTAR PUSTAKA	97
LAMPIRAN A	99
LAMPIRAN B	114
LAMPIRAN C	123
LAMPIRAN D.....	129

DAFTAR TABEL

Tabel 2.1 Pembobotan ROC	10
Tabel 3.1 Kriteria Pencarian <i>Homestay</i>	15
Tabel 4.1 Deskripsi Aktor	27
Tabel 4.2 Deskripsi <i>Usecase</i>	27
Tabel 4.3 Skenario Mencari <i>Homestay</i>	30
Tabel 4.4 Fungsi Menyimpan Prioritas Sub Kriteria	44
Tabel 4.5 Fungsi Menyimpan Prioritas Sub-Sub Kriteria.....	45
Tabel 4.6 Fungsi Perhitungan Metode <i>Simple Multi-Attribute Rating Technique Exploiting Ranks</i> (SMARTER).....	46
Tabel 4.7 Fungsi Menampilkan Rekomendasi <i>Homestay</i> Terbaik	49
Tabel 4.8 <i>Test Case Method</i> Hitung.....	53
Tabel 4.9 <i>Test Case Method</i> Simpanprioritassubkriteria	58
Tabel 4.10 <i>Test Case Method</i> Simpanprioritassubsubkriteria	60
Tabel 4.11 <i>Black Box Testing</i>	62
Tabel 5.1 Pembobotan Kriteria Rekomendasi <i>Homestay</i>	77
Tabel 5.2 Pembobotan Sub Kriteria Fasilitas Utama Rekomendasi <i>Homestay</i>	78
Tabel 5.3 Pembobotan Sub Kriteria Fasilitas Tambahan Rekomendasi <i>Homestay</i>	78
Tabel 5.4 Pembobotan Sub-Sub Kriteria Fasilitas Utama Rekomendasi <i>Homestay</i> ...	79
Tabel 5.5 Pembobotan Sub-Sub Kriteria Fasilitas Tambahan Rekomendasi <i>Homestay</i>	79
Tabel 5.6 Nilai <i>Homestay</i> Kriteria Fasilitas Utama	80
Tabel 5.7 Nilai <i>Homestay</i> Kriteria Fasilitas Tambahan	81
Tabel 5.8 Perhitungan Bobot Akhir Kriteria Fasilitas Utama.....	83
Tabel 5.9 Perhitungan Bobot Akhir Kriteria Fasilitas Tambahan.....	83
Tabel 5.10 Hasil Perhitungan Bobot Akhir.....	84

DAFTAR GAMBAR

Gambar 3.1 Diagram Alir Tahapan Penelitian.....	13
Gambar 3.2 Tahapan Model <i>Waterfall</i>	13
Gambar 3.3 Diagram Alir Penerapan Metode SMARTER.....	17
Gambar 3.4 Alur Sistem Rekomendasi <i>Homestay</i>	18
Gambar 4.1 <i>Business Process</i>	26
Gambar 4.2 <i>Usecase Diagram</i>	27
Gambar 4.3 <i>Sequence Diagram</i> Mencari <i>Homestay</i>	35
Gambar 4.4 <i>Activity Diagram</i> Mencari <i>Homestay</i>	38
Gambar 4.5 <i>Class Diagram</i>	40
Gambar 4.6 <i>Listing Program Method</i> Hitung.....	52
Gambar 4.7 Diagram Alir <i>Method</i> Hitung	52
Gambar 4.8 <i>Listing Program Method</i> Simpanprioritassubkriteria	57
Gambar 4.9 Diagram Alir <i>Method</i> Simpanprioritassubkriteria	58
Gambar 4.10 <i>Listing Program Method</i> Simpanprioritassubsubkriteria.....	59
Gambar 4.11 Diagram Alir <i>Method</i> Simpanprioritassubsubkriteria.....	60
Gambar 5.1 Halaman Utama Sistem Informasi Rekomendasi <i>Homestay</i>	64
Gambar 5.2 Halaman <i>Login</i>	65
Gambar 5.3 Halaman <i>Dashboard</i>	66
Gambar 5.4 Halaman <i>Homestay</i>	66
Gambar 5.5 Modal Tambah <i>Homestay</i>	67
Gambar 5.6 Halaman Ubah <i>Homestay</i>	67
Gambar 5.7 Modal Menghapus atau Menon-aktifkan <i>Homestay</i>	68
Gambar 5.8 Halaman Melihat Detail <i>Homestay</i>	68
Gambar 5.9 Halaman <i>Booking</i>	69
Gambar 5.10 Halaman Ubah Data Pemesan	69
Gambar 5.11 Halaman Detail <i>Booking</i>	70
Gambar 5.12 Modal Hapus Data Pemesan	70

Gambar 5.13 Modal Konfirmasi <i>Booking</i>	70
Gambar 5.14 Halaman Form Prioritas Sub Kriteria.....	71
Gambar 5.15 Halaman Form Prioritas Sub-Sub Kriteria.....	72
Gambar 5.16 Halaman Rekomendasi <i>Homestay</i> Terbaik.....	73
Gambar 5.17 Lihat Detail Perhitungan.....	74
Gambar 5.18 Halaman <i>Form Booking</i>	75
Gambar 5.19 Data Pembobotan Kriteria.....	76
Gambar 5.20 Sub-Sub Kriteria <i>Homestay</i>	80
Gambar 5.21 Perhitungan Nilai <i>Utility</i>	81
Gambar 5.22 Hasil Perhitungan Nilai <i>Utility</i>	82
Gambar 5.23 Kode Program Menyimpan Bobot Sub Kriteria.....	84
Gambar 5.24 Kode Program Menyimpan Bobot Sub-Sub Kriteria.....	85
Gambar 5.25 Kode Program Menghitung Nilai <i>Homestay</i>	86
Gambar 5.26 Kode Program Menghitung Nilai <i>Utility</i>	86
Gambar 5.27 Kode Program Menghitung Bobot Akhir.....	87
Gambar 5.28 Kode Program Perangkingan setiap <i>Homestay</i>	87
Gambar 5.29 Pengisian Prioritas Sub Kriteria Oleh <i>User</i>	90
Gambar 5.30 Pengisian Prioritas Sub-Sub Kriteria Oleh <i>User</i>	91
Gambar 5.31 Saran Pencarian <i>Homestay</i> Terbaik Menggunakan metode SMARTER	92
Gambar 5.32 Pengisian Prioritas Sub Kriteria Oleh <i>User</i> dengan Inputan yang Berbeda.....	92
Gambar 5.33 Pengisian Prioritas Sub-Sub Kriteria Oleh <i>User</i> dengan Inputan yang Berbeda.....	93
Gambar 5.34 Saran Pencarian <i>Homestay</i> Terbaik Menggunakan metode SMARTER dengan Inputan yang Berbeda.....	94

BAB 1. PENDAHULUAN

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

1.1. Latar Belakang

Perkembangan teknologi saat ini sudah sangat dibutuhkan. Teknologi tersebut bisa sangat membantu dalam proses pengembangan suatu daerah. Teknologi yang paling dibutuhkan yaitu teknologi informasi, dimana informasi tersebut memiliki dampak terbesar dalam mengembangkan informasi daerah tersebut baik dari segi ekonomi, sosial, pendidikan, budaya dan pariwisata. Informasi pariwisata yang sering dibutuhkan oleh wisatawan dari daerah yaitu potensi alam, budaya, makanan khas, dan juga wisata.

Kabupaten Banyuwangi merupakan daerah yang terletak di bagian timur pulau Jawa. Banyuwangi memiliki berbagai potensi alam yang banyak direkomendasikan untuk tujuan wisata. Salah satu wisata yang menjadi primadona di Banyuwangi ini yaitu Kawah Ijen. Kawah Ijen terletak diantara Kabupaten Banyuwangi dan Kabupaten Bondowoso. Kawah Ijen merupakan wisata yang sering dikunjungi dan menjadi daya tarik oleh wisatawan domestik, maupun wisatawan asing karena pemandangan gunung yang indah dan penampakan *bluefire*, yaitu suatu fenomena alam yang terjadi karena lelehan lava pijar berwarna biru. Fenomena ini merupakan hal yang sangat langka dan hanya ditemui di beberapa negara di dunia dan salah satunya di Kawah Ijen tersebut. Akibat dari fenomena alam yang langka ini yang membuat Kawah Ijen sangat sering dikunjungi oleh wisatawan asing dari berbagai belahan dunia.

Wisatawan yang datang untuk berwisata di Kawah Ijen ini tentunya membutuhkan fasilitas *homestay* untuk bermalam sebelum mendaki. Fasilitas

homestay yang ada di daerah lereng gunung yaitu di kawasan Kecamatan Licin dikelola oleh Badan Usaha Milik Desa (BUMDES). Para wisatawan masih perlu bertanya kepada pihak BUMDES tentang informasi ketersediaan *homestay* yang ada disekitar lereng gunung. Terkadang pihak BUMDES juga kesulitan dalam memeriksa ketersediaan *homestay* yang ada, karena harus melakukan survei satu persatu *homestay* yang tersedia, sehingga hal tersebut menyulitkan pihak BUMDES dan memakan waktu yang banyak bagi wisatawan dalam memilih *homestay*. Berdasarkan masalah tersebut diperlukan sistem informasi rekomendasi yang ditujukan untuk membantu merekomendasikan informasi *homestay* yang ada di sekitar kawasan Licin. Berdasarkan masalah tersebut diharapkan adanya suatu sistem informasi rekomendasi yang ditujukan untuk membantu merekomendasikan informasi *homestay* yang ada di sekitar kawasan Licin dan juga fasilitas yang diberikan pada setiap *homestay* sesuai kriteria yang diinginkan oleh wisatawan.

Sistem rekomendasi ini bertujuan dalam membantu para wisatawan dalam memilih *homestay* sesuai kriteria yang diinginkan dengan cepat dan tepat. 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). Oleh karena itu sistem pendukung keputusan sangat dibutuhkan dalam penelitian ini untuk mendukung dalam merekomendasikan *homestay* yang sesuai, cepat dan tepat.

Metode yang digunakan dalam penelitian ini yaitu metode *Simple Multi-Attribute Rating Technique Exploiting Ranks* (SMARTER). Metode SMARTER dipilih oleh peneliti karena dalam pemilihan *homestay* terdapat banyak sekali kriteria yang akan dipilih. Menurut Edwards and Baron (dalam Alfita, 2012) ada 2 hal yang mendasari dan menjadikan kelebihan metode SMARTER yaitu :

1. Teknik yang sederhana, sehingga dapat digunakan oleh pembuat keputusan.
2. Teknik yang mudah guna mendapatkan keputusan yang dapat diandalkan. Metode ini digunakan untuk menentukan bobot dari masing-masing kriteria yang sudah

ditentukan. Kriteria tersebut didapat dari pengumpulan data dari observasi pada setiap *homestay* di daerah Kawasan Licin Kabupaten Banyuwangi. Pembobotan pada metode SMARTER menggunakan range antara 0 sampai 1, sehingga mempermudah perhitungan dan perbandingan nilai terbaik pada masing-masing alternatif yang ada, dalam hal ini alternatif tersebut merupakan kriteria *homestay* yang diinginkan oleh wisatawan. Oleh karena itu diharapkan pengambilan keputusan melalui pembobotan kriteria pada metode SMARTER ini dapat memberikan rekomendasi kepada wisatawan dalam mengambil keputusan.

1.2. Rumusan Masalah

Berdasarkan latar belakang di atas, dapat dirumuskan permasalahan sebagai berikut :

1. Bagaimana membangun sistem yang dapat memberikan informasi tentang *homestay* bagi wisatawan di kawasan Kawah Ijen Kecamatan Licin Kabupaten Banyuwangi ?
2. Bagaimana cara merekomendasikan *homestay* bagi wisatawan menggunakan metode *Simple Multi-Attribute Rating Technique Exploiting Ranks* (SMARTER)?

1.3. Tujuan

Adapun tujuan dari penelitian ini adalah sebagai berikut:

1. Membangun sistem yang dapat memberikan informasi tentang *homestay* bagi wisatawan di kawasan Kawah Ijen Kecamatan Licin Kabupaten Banyuwangi.
2. Merekomendasikan *homestay* bagi wisatawan sesuai keinginan dengan menerapkan metode *Simple Multi-Attribute Rating Technique Exploiting Ranks* (SMARTER).

1.4. Batasan Masalah

Beberapa hal yang membatasi penelitian ini adalah sebagai berikut:

1. *Homestay* yang diteliti yaitu berada di kawasan Kawah Ijen Kecamatan Licin Kabupaten Banyuwangi.
2. Metode yang digunakan dalam penelitian ini yaitu *Simple Multi-Attribute Rating Technique Exploiting Ranks* (SMARTER) untuk penerapan perhitungan.
3. Penyelesaian hanya membahas teknik perhitungan rekomendasi *homestay* bagi wisatawan dan bersifat statis dalam penentuan kriteria.
4. Data kriteria bersifat statis.
5. Pemesanan dapat dilakukan hanya untuk satu *homestay*.
6. Konfirmasi pembayaran dilakukan secara *offline* dan lewat telepon.

1.5. Sistematika Penulisan

Sistematika penulisan dalam penyusunan tugas akhir ini sebagai berikut :

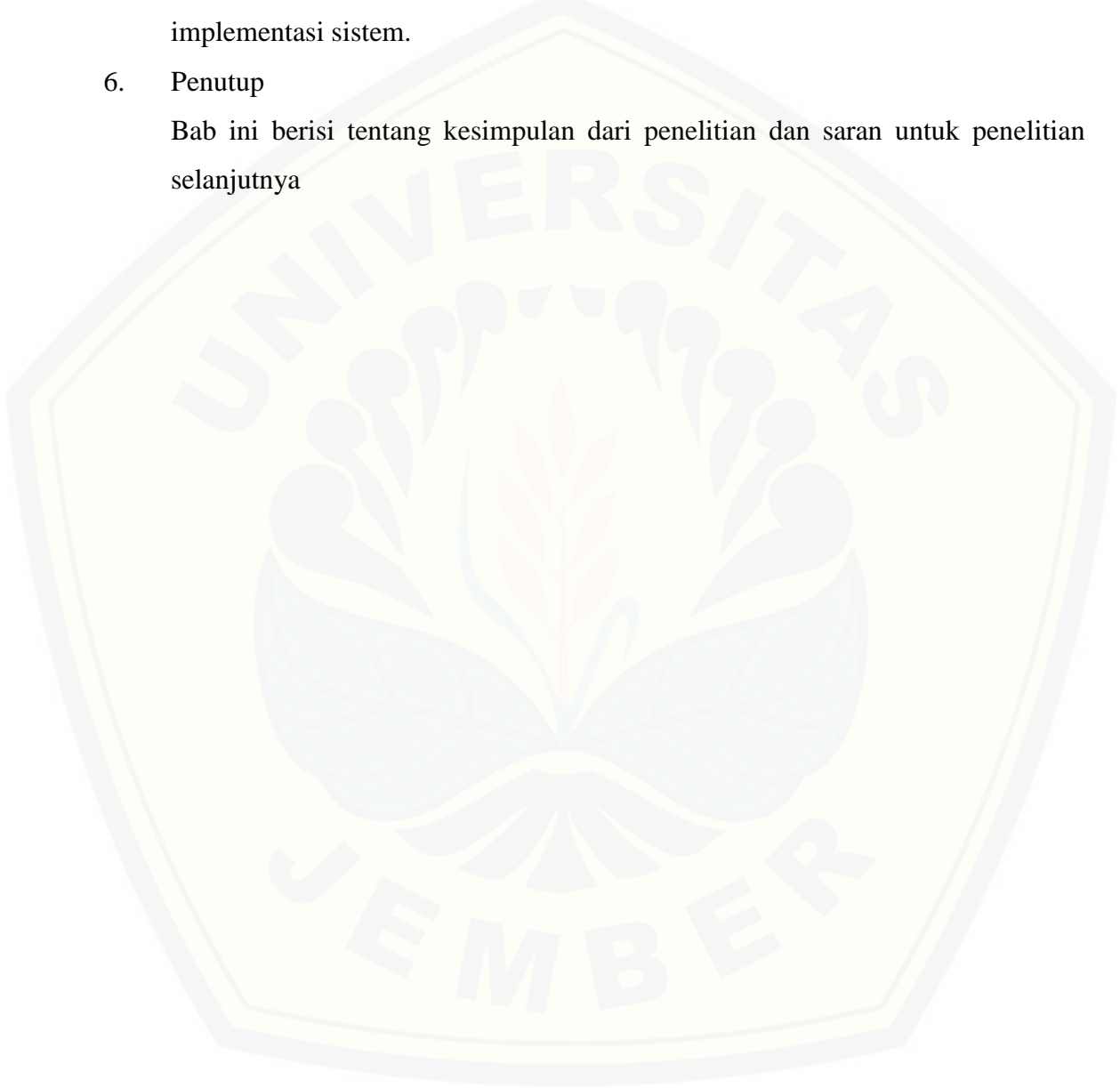
1. **Pendahuluan**
Bab ini menjelaskan tentang latar belakang, perumusan masalah, tujuan dan manfaat, ruang lingkup studi dan sistematika penulisan.
2. **Tinjauan Pustaka**
Bab ini menjelaskan tentang materi, informasi, tinjauan pustaka, dan studi terdahulu yang menjadi kerangka pemikiran dalam penelitian.
3. **Metodologi Penelitian**
Bab ini menjelaskan tentang metode penelitian yang digunakan dalam penelitian.
4. **Pengembangan Sistem**
Bab ini menjelaskan tentang pengembangan sistem yang dikembangkan.

5. Hasil dan Pembahasan

Bab ini menjelaskan tentang hasil dan pembahasan dari penelitian yang dilakukan. Dengan memaparkan hasil penelitian dan hasil percobaan implementasi sistem.

6. Penutup

Bab ini berisi tentang kesimpulan dari penelitian dan saran untuk penelitian selanjutnya



BAB 2. TINJAUAN PUSTAKA

Pada tinjauan pustaka ini berisi paparan teori-teori dan pustaka yang akan dipakai dalam penelitian. Teori-teori ini berupa teori dari buku literatur dan jurnal.

2.1. Penelitian Terdahulu

Penelitian sebelumnya yang pernah dilakukan dengan mengimplementasikan metode SMARTER yaitu dengan judul “*Decision Support System Of Reserve Building Cultural Revitalization Determination Using Simple Multiattribute Rating Technique Exploiting Ranks Method*” yang dilakukan oleh Riza Alfita dari Fakultas Teknik Universitas Trunojoyo pada tahun 2012. Penelitian ini membahas pembobotan penentuan revitalisasi cagar budaya dengan menggunakan metode SMARTER dimana kondisi fisik tiap bangunan yang ada diperhitungkan. Analisis tersebut berupa penilaian dan pembobotan terhadap tiap bangunan berdasarkan kriteria bangunan sebagai bangunan cagar budaya.

Penelitian lainnya dengan judul “Sistem Pendukung Keputusan Seleksi Penerimaan Mahasiswa Pengganti Beasiswa Penuh Bidikmisi Universitas Tanjungpura Dengan Menerapkan Metode SMARTER” oleh Dwi Haryanti, Helfi Nasution, Anggi Srimurdianti Sukanto dari jurusan Teknik Informatika Fakultas Teknik Universitas Tanjungpura tahun 2016. Penelitian ini membahas seleksi penerimaan mahasiswa pengganti beasiswa penuh Bidikmisi Universitas Tanjungpura dengan menerapkan metode SMARTER dan mengetahui tingkat persentase keakurasian sistem pendukung keputusan dengan metode SMARTER dalam menyeleksi mahasiswa pengganti beasiswa penuh Bidikmisi Universitas Tanjungpura terhadap nilai sesungguhnya. Metode SMARTER digunakan untuk proses perhitungan nilai setiap alternatif sehingga akan didapat urutan nilai setiap alternatif. Pembobotan pada metode SMARTER menggunakan rumus Rank Order Centroid (ROC). Hasil keluaran dari sistem ini adalah urutan nilai akhir calon mahasiswa

pengganti dari urutan nilai akhir terbesar hingga terkecil berdasarkan program studi dan angkatan.

Berdasarkan penelitian tersebut dapat disimpulkan bahwa dalam pengambilan keputusan perlu diadakannya kriteria-kriteria dan pembobotan dari alternatif-alternatif yang ada. Metode pengambilan keputusan menggunakan SMARTER juga dapat digunakan dalam penelitian ini. Dalam penelitian ini dibutuhkan kriteria tempat *homestay* mulai yang ditentukan oleh peneliti, mulai dari harga, jarak, jumlah kamar, dan fasilitas yang sesuai dengan keinginan wisatawan. Dalam penelitian ini penentuan prioritas kepentingan setiap kriteria dapat ditentukan sendiri oleh wisatawan, sehingga wisatawan dapat memilih kriteria sesuai keinginannya. Penerapan metode SMARTER dalam penelitian ini diharapkan dapat menghasilkan keluaran yaitu suatu pengambil keputusan dengan keakuratan yang tepat dari kriteria-kriteria yang diinginkan oleh wisatawan.

2.2. Sistem Informasi

Menurut Indah (2013), sistem informasi merupakan kegiatan atau aktifitas yang melibatkan serangkaian proses, berisi informasi-informasi yang digunakan untuk mencapai tujuan. Menurut Kadir (2014:8), sistem informasi merupakan suatu cakupan dari sejumlah komponen (manusia, komputer, teknologi informasi, dan prosedur kerja), ada sesuatu yang di proses (data menjadi informasi), dan dimaksudkan untuk mencapai suatu sasaran atau tujuan.

Beberapa definisi sistem informasi tersebut dapat disimpulkan bahwa sistem informasi merupakan serangkaian proses kegiatan atau aktifitas dari sejumlah komponen (manusia, komputer, teknologi informasi, dan prosedur kerja) yang memiliki suatu tujuan yang akan dicapai.

2.3. Sistem Pendukung Pengambilan Keputusan (*Decision Support System*)

Menurut Waghmode dan P.P. Jansandekar (dalam Fauziyah, 2015) mendefinisikan *Decision Support System* (DSS) “kumpulan dari aplikasi perangkat

lunak dan perangkat keras yang saling berintegrasi yang merupakan tulang punggung dari sebuah proses pengambilan keputusan”. Menurut Kusriani (2007), 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.

Kedua definisi tersebut dapat disimpulkan bahwa DSS merupakan sekumpulan dari aplikasi perangkat lunak dan perangkat keras yang saling berintegrasi untuk pemrosesan data dan penilaian guna membantu seseorang dalam mengambil keputusan.

Hasil dari DSS ini berbentuk sebuah perankingan dimana nilai dari perankingan itulah yang menjadikan suatu keputusan yang dapat diambil oleh seseorang. Banyak sekali metode perankingan yang dapat di terapkan untuk DSS seperti, *Metode Simple Additive Weighting (SAW)*, *Weight Product (WP)*, *Analytic Hierarchy Process (AHP)*, *Simple Multi Attribute Rating Technique (SMART)*, *Simple Multi Attribute Rating Technique Exploiting Ranks (SMARTER)*, dan masih banyak lagi.

2.4. Metode Simple Multi-Attribute Rating Technique Exploiting Rank (SMARTER)

Metode *Simple Multi-Attribute Rating Technique Exploiting Rank (SMARTER)* merupakan pengembangan dari metode *Simple Multi-Attribute Rating Technique (SMART)*. Metode SMART pertama kali diperkenalkan oleh Edward pada tahun 1971 dan baru diberi nama metode SMART pada tahun 1997. Metode ini telah dikembangkan menjadi *Simple Multi-Attribute Rating Technique Swing (SMARTS)* lalu pada tahun 1994 Edward dan Baron memodifikasi dan memperbaiki metode ini menjadi metode *Simple Multi-Attribute Rating Technique Exploiting Ranks (SMARTER)*.

Perbedaan antara metode SMARTER dengan metode SMART dan metode SMARTS terletak pada cara pembobotannya. Pembobotan kriteria pada ketiga metode tersebut tergantung pada urutan prioritas atribut dimana pada urutan pertama ditempati oleh atribut yang dianggap paling penting. Pada metode SMART dan SMARTER pembobotan diberikan langsung oleh pengambil keputusan. Tetapi prosedur pembobotan tersebut dianggap tidak profesional dimana setiap bobot yang diberikan harus mencerminkan jarak dan prioritas setiap kriteria dengan tepat. Untuk mengatasi hal tersebut, pada metode SMARTER digunakan rumus pembobotan *Rank Order Centroid* (ROC).

2.4.1. Pembobotan Rank Order Centroid (ROC)

ROC didasarkan pada tingkat kepentingan atau prioritas dari kriteria. Menurut Jeffreys dan Cockfield (dalam Rahmah, 2013), teknik ROC memberikan bobot pada setiap kriteria sesuai dengan ranking yang dinilai berdasarkan tingkat prioritas. Sehingga kriteria ke- n ditulis $Cr_1 \geq Cr_2 \geq Cr_3 \geq \dots \geq Cr_n$ yang berarti kriteria 1 lebih penting dari kriteria 2, dan kriteria 2 lebih penting dari kriteria 3 dan seterusnya. Penentuan bobot diberikan aturan yang sama yaitu $W_1 \geq W_2 \geq W_3 \geq \dots \geq W_n$ dimana W_1 merupakan bobot untuk kriteria Cr_1 . Dapat juga dijelaskan sebagai berikut :

Jika terdapat kriteria : $Cr_1 \geq Cr_2 \geq Cr_3 \geq \dots \geq Cr_n$

Maka bobotnya adalah : $W_1 \geq W_2 \geq W_3 \geq \dots \geq W_n$

Selanjutnya, jika k adalah banyaknya kriteria, maka

$$W_1 = \frac{\left(1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{k}\right)}{k}$$

$$W_2 = \frac{\left(0 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{k}\right)}{k}$$

$$W_3 = \frac{\left(0 + 0 + \frac{1}{3} + \dots + \frac{1}{k}\right)}{k}, \text{ dan}$$

$$W_k = \frac{\left(0 + 0 + \dots + 0 + \frac{1}{k}\right)}{k}$$

Keterangan :

W_1 = bobot dari kriteria prioritas ke-1

W_2 = bobot dari kriteria prioritas ke-2

W_3 = bobot dari kriteria prioritas ke-3

W_k = bobot dari kriteria prioritas ke-k

Menurut Rahmah (2013), secara umum pembobotan ROC dapat dilihat pada Persamaan 2.1,

$$W_k = \frac{1}{k} \sum_{i=1}^k \left(\frac{1}{i}\right) \quad [2.1]$$

Keterangan :

W_k = bobot dari kriteria prioritas ke-k

k = Kriteria prioritas ke-k

i = Kriteria Prioritas ke-i

Nilai persamaan sebelumnya untuk pembobotan 2 sampai 9 kriteria dapat dilihat pada Tabel 2.1 Pembobotan ROC.

Tabel 2.1 Pembobotan ROC

Ranking	Kriteria							
	2	3	4	5	6	7	8	9
1	0.75	0.611	0.521	0.457	0.408	0.370	0.340	0.314
2	0.25	0.278	0.271	0.257	0.242	0.228	0.215	0.203
3		0.111	0.146	0.157	0.158	0.156	0.152	0.148
4			0.063	0.090	0.103	0.109	0.111	0.111
5				0.040	0.061	0.073	0.079	0.083
6					0.028	0.044	0.054	0.061
7						0.028	0.034	0.042
8							0.016	0.026
9								0.012

Istra,dkk. (dalam Rahmah,2013) menuliskan rumus metode SMARTER secara umum pada Persamaan 2.2,

$$\text{maksimumkan } \sum_{j=1}^k W_j U_{ij}, \forall i = 1 \text{ sampai } n \quad [2.2]$$

Keterangan :

W_j = nilai skala pembobotan ke-j dari k kriteria

U_{ij} = nilai utilitas i pada kriteria j

n = banyak data.

Menurut Okfalisa dan Gunawan (2014), nilai utilitas dapat dilihat dari Persamaan 2.3,

$$v(x) = \sum_{j=1}^n w_j v_j(x) \quad [2.3]$$

Keterangan :

W_i = Bobot yang mempengaruhi dari dimensi ke i

V_i = Objek evaluasi pada dimensi ke i

n = Jumlah dimensi nilai yang berbeda.

BAB 3. METODOLOGI PENELITIAN

Bab ini menjelaskan tentang tahapan-tahapan penelitian yang sistematis untuk menggambarkan bagaimana langkah-langkah yang akan dilakukan sehingga dapat mencapai tujuan sebenarnya dari penelitian.

3.1. Jenis Penelitian

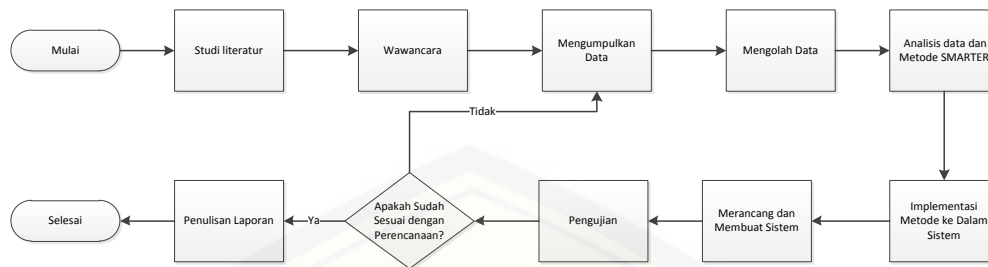
Jenis penelitian yang akan dilakukan merupakan penelitian pengembangan. Jenis penelitian pengembangan digunakan karena penelitian ini mengembangkan proses manual menjadi sistem informasi rekomendasi *homestay* menggunakan metode *Simple Multi-Attribute Rating Technique Exploiting Ranks* (SMARTER) di Kawasan Kawah Ijen Kecamatan Licin Kabupaten Banyuwangi yang terkomputerisasi.

3.2. Tempat dan Waktu Penelitian

Tempat yang digunakan dalam penelitian adalah kawasan Kawah Ijen Kecamatan Licin Kabupaten Banyuwangi. Waktu penelitian dilakukan selama 2 bulan yaitu bulan Mei 2016 sampai dengan bulan Juni 2016.

3.3. Tahapan Penelitian

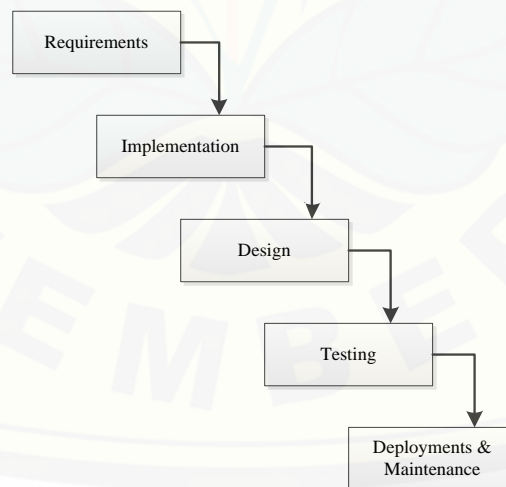
Penelitian ini akan dilakukan dalam beberapa tahapan, yaitu dimulai dengan tahap pengumpulan data meliputi studi literatur, wawancara, mengumpulkan data dan mengolah data. Selanjutnya tahap pengembangan system meliputi analisis data, implementasi metode ke sistem, merancang sistem, membuat sistem, dan pengujian sistem. Tahap terakhir yaitu tahap penulisan laporan. Tahapan dalam penelitian ini digambarkan pada Gambar 3.1 Diagram Alir Tahapan Penelitian.



Gambar 3.1 Diagram Alir Tahapan Penelitian

3.4. Tahap Pengembangan Sistem

Metode pengembangan yang digunakan pada penelitian ini adalah *System Development Life Cycle* (SDLC) dengan menggunakan model *waterfall*. Model *waterfall* digunakan karena kualitas sistem dari model ini cenderung baik yang disebabkan tahapan-tahapannya yang terstruktur. Model *waterfall* memiliki fase yang diselesaikan secara bertahap dan ketika satu fase sudah diselesaikan, maka fase lainnya baru bisa dilakukan. Fase-fase tersebut meliputi *requirements*, *design*, *implementation*, *testing* dan *maintenance* (Mishra dan Dubey, 2013). Gambaran model *waterfall* dapat dilihat pada Gambar 3.2 Tahapan Model *Waterfall*.

Gambar 3.2 Tahapan Model *Waterfall*

Sumber : (Mishra dan Dubey, 2013)

3.4.1. Analisis Kebutuhan (*Requirements*)

Tahap pertama pada proses pengembangan perangkat lunak ini adalah analisis kebutuhan. Kebutuhan yang digunakan yaitu dari hasil wawancara kepada pihak-pihak pemilik *homestay* dan pihak Badan Usaha Milik Desa (BUMDES). Data-data yang telah didapat kemudian dikelompokkan menjadi kebutuhan fungsional dan kebutuhan non-fungsional.

3.4.1.1. Pengumpulan Data

Teknik pengumpulan data yang akan dilakukan pada penelitian ini adalah sebagai berikut:

a. Studi Literatur

Studi literatur ini dilakukan dengan tujuan sebagai dasar pembahasan penyusunan dasar teori yang digunakan dalam penelitian. Sumber yang digunakan sebagai studi literatur berupa buku, jurnal, dan karya ilmiah dari penelitian yang sejenis sebelumnya.

b. Wawancara

Wawancara dilakukan dengan tujuan untuk mengetahui harga, jarak ke kawah ijen, jarak ke sarana umum, jumlah kamar, wifi, transportasi, katering dengan mengumpulkan data yang akan di gunakan dalam penelitian. Data yang terkumpul akan dijadikan bahan atau kriteria dalam pembuatan sistem.

Data yang terkumpul kemudian dianalisis untuk digunakan sebagai parameter yang akan diimplementasikan kedalam perhitungan metode *Simple Multi-Attribute Rating Technique Exploiting Ranks* (SMARTER) dan diterapkan kedalam pembuatan Sistem Informasi Rekomendasi *Homestay* untuk mendapatkan hasil berupa peringkatan *homestay* terbaik. Peringkatan *homestay* diperoleh dari parameter-parameter yang telah di tentukan oleh peneliti dan dijadikan sebagai kriteria-kriteria hasil dari survei pencarian *homestay* secara umum seperti pada berbagai aplikasi pencarian *homestay* dengan ditambahkan

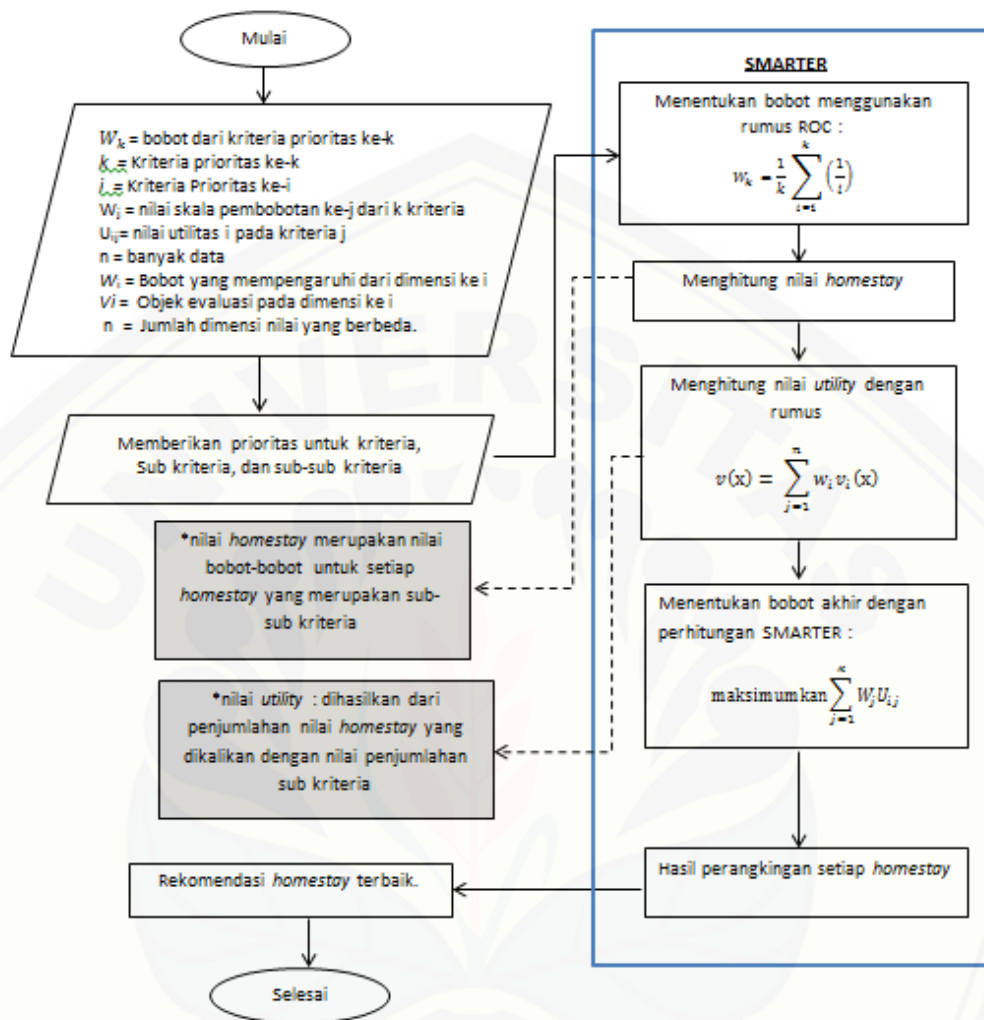
beberapa kriteria lainnya oleh peneliti agar lebih spesifik dalam melakukan pencariannya. Setiap kriteria memiliki sub kriteria dan sub-sub kriteria yang memiliki bobot masing-masing. Kriteria untuk rekomendasi *homestay* memiliki 2 kriteria fasilitas utama dan fasilitas tambahan, fasilitas utama merupakan fasilitas yang pasti dimiliki oleh setiap *homestay* dan fasilitas tambahan merupakan fasilitas yang hanya dimiliki oleh beberapa *homestay*. Penjelasan kriteria, sub kriteria dan sub-sub kriteria dapat dilihat pada Tabel 3.1 Kriteria Pencarian *Homestay*.

Tabel 3.1 Kriteria Pencarian *Homestay*

Kriteria	Sub Kriteria	Sub-Sub Kriteria
Fasilitas Utama	Harga	150.000
		200.000
		>200.000
	Jarak ke- Ijen	<10km
		10km-20km
		>20km
	Jumlah Kamar	>=3
		2
		1
	Jarak ke-sarana umum	<3km
		3km-7km
		>7km
Fasilitas Tambahan	Wifi	Ada
		Tidak Ada
	Transportasi	Ada
		Tidak Ada
	Katering	Ada
		Tidak Ada

3.4.1.2. Pengolahan Data

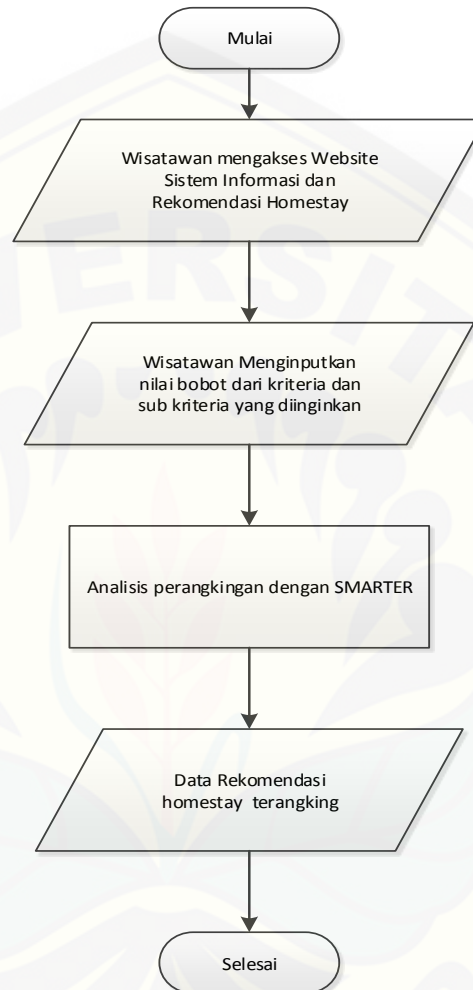
Tahap pengolahan data dengan menerapkan metode *Simple Multi-Attribute Rating Technique Exploiting Ranks* (SMARTER) pada Sistem Informasi Rekomendasi *Homestay* dapat dilihat pada Gambar 3.3 yaitu dimulai dengan menelaah data secara keseluruhan yang telah diperoleh dari tahap pengumpulan data yaitu dari studi literatur, wawancara maupun dari sumber lainnya. Langkah selanjutnya yaitu menentukan kriteria, sub kriteria dan sub-sub kriteria yaitu seperti Tabel 3.1 pada sub bab 3.1.2.1 . Kemudian memberikan prioritas untuk setiap kriteria , sub-dan sub-sub kriteria. Prioritas untuk kriteria sudah ditentukan oleh peneliti, dan untuk sub kriteria dan sub-sub kriteria ditentukan oleh *User*. Menentukan bobot menggunakan ROC setiap kriteria, sub kriteria dan sub-sub kriteria. Kemudian menghitung bobot nilai setiap *homestay*. Selanjutnya menghitung nilai *utility* yang dihasilkan dari penjumlahan nilai *homestay* berupa inputan untuk sub-sub kriteria yang dikalikan dengan nilai penjumlahan sub kriteria. Setelah itu, dilakukan proses perhitungan menggunakan metode SMARTER untuk menentukan bobot akhir. Dari perhitungan tersebut akan muncul hasil perankingan. Terakhir, merekomendasikan sesuai kriteria terbaik.



Gambar 3.3 Diagram Alir Penerapan Metode SMARTER

Alur proses analisis pencarian *homestay* terbaik menggunakan metode *Simple Multi-Attribute Rating Technique Exploiting Ranks* (SMARTER) dimulai dari *user* (wisatawan) mengakses website sistem informasi rekomendasi *homestay*, selanjutnya *user* (wisatawan) menginputkan nilai bobot kriteria, sub kriteria dan sub-sub kriteria yang diambil dari prioritas pilihan *user*. Kemudian sistem melakukan analisis perankingan menggunakan metode SMARTER. Hasilnya adalah rekomendasi

homestay terbaik. Detail proses analisis pencarian *homestay terbaik* dapat dilihat pada Gambar 3.4 Alur Sistem Rekomendasi *Homestay*.



Gambar 3.4 Alur Sistem Rekomendasi *Homestay*

3.4.2. Desain Sistem (*Design*)

Tahap yang selanjutnya yaitu desain sistem menggunakan *Unified Modeling Language* (UML) yang dirancang menggunakan konsep *Object-Oriented Programming* (OOP). Pemodelan UML yang akan digunakan sebagai berikut:

1. *Business Process*

Business Process digunakan untuk menggambarkan data yang dibutuhkan sistem sebagai input, lalu diproses menjadi output serta tujuan dari pembuatan sistem.

2. *Usecase Diagram*

Usecase diagram menggambarkan fitur-fitur apa saja yang tersedia serta aktor yang mempunyai hak akses sistem.

3. *Scenario*

Scenario menjelaskan bagaimana alur sistem bekerja dan keadaan yang terjadi ketika terjadi aksi tertentu.

4. *Sequence Diagram*

Sequence Diagram merupakan gambaran interaksi antar objek di dalam sistem. Interaksi antar objek yaitu pengguna, display, dan sebagian berupa pesan.

5. *Activity Diagram*

Activity Diagram menggambarkan aliran aktivitas dalam sistem, mulai dari alir dimulai, decision yang terjadi dan akhir dari alir.

6. *Class Diagram*

Class Diagram menggambarkan struktur dan deskripsi class, package dan objek beserta hubungan satu sama lain seperti pewarisan, asosiasi, dependensi dan lain-lain.

7. *Entity Relationship Diagram*

Entity Relationship Diagram menggambarkan struktur database yang akan dibangun pada sistem.

3.4.3. Implementasi (*Implementation*)

Pada tahap ini desain yang telah dibuat diimplementasikan ke dalam kode program. Hal yang dilakukan dalam tahap implementasi antara lain:

1. Penulisan kode program (*coding*) menggunakan bahasa pemrograman *Page Hyper Text Pre-Processor* (PHP) dengan bantuan *framework Code Igniter* (CI).
2. Manajemen basisdata menggunakan MySQL.

3. Tampilan berbasis *website*.

3.4.4. Pengujian (*Testing*)

Testing atau pengujian sistem dilakukan dengan *white-box testing* dan *black-box testing*. *White Box Testing* merupakan cara pengujian dengan melihat modul yang telah dibuat dengan program yang ada. Pengujian *white-box* menggunakan metode pengujian berbasis *path*, yang merupakan teknik pengujian *white-box*. Metode basis ini memungkinkan desainer *test case* mengukur kompleksitas logis dari desain prosedural dan menggunakannya sebagai pedoman untuk menetapkan basis set dari jalur eksekusi program.

Langkah-langkah pembuatan *test case* adalah sebagai berikut:

1. Menggunakan perancangan prosedural atau program sumber sebagai dasar digambarkan diagram alirnya.
2. Menentukan kompleksitas siklomatis untuk diagram alir yang telah dibuat:

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

Keterangan :

$V(G)$ = Kompleksitas Siklomatis .

E = Jumlah *Edge*

N = Jumlah *Node*

3. Menentukan *path* independen pada diagram alir
4. Membuat *test case* yang akan mengerjakan masing-masing *path* pada basis set. Data yang dipilih harus tepat sehingga setiap kondisi dari *node* predikat dikerjakan semua.

Black-box testing merupakan pengujian program pada aplikasi langsung tanpa melihat desain dan kode program. Tujuan dari *black-box testing* adalah untuk mengetahui apakah *input*, *output*, dan fitur-fitur pada program yang telah dibangun sesuai dengan kebutuhan *user*.

3.1.1. Pemeliharaan (*Maintenance*)

Perangkat lunak yang sudah selesai atau siap digunakan akan mengalami perubahan. Perubahan biasanya berupa *error* sehingga diperlukan perbaikan dan pemeliharaan kepada sistem.



BAB 4. PERANCANGAN SISTEM

Bab ini menguraikan tentang perancangan desain sistem untuk Sistem Informasi Rekomendasi *Homestay* Menggunakan Metode *Simple Multi-Attribute Rating Technique Exploiting Ranks* (SMARTER). Perancangan sistem dimulai dari analisis kebutuhan fungsional dan non-fungsional sistem, selanjutnya merancang *business process*, *usecase diagram*, *scenario*, *activity diagram*, *sequence diagram*, *class diagram* dan *entity relationship diagram* (ERD).

4.1. Deskripsi Umum Sistem

Deskripsi umum mengenai Sistem Informasi Rekomendasi *Homestay* menggunakan metode *Simple Multi Attribute Rating Technique Exploiting Ranks* (SMARTER) yang dibangun dalam penelitian ini akan dijelaskan pada *Statement of Purpose* (SOP) dan fungsi sistem.

4.1.1. *Statement of Purpose* (SOP)

Sistem Informasi Rekomendasi *Homestay* menggunakan metode *Simple Multi Attribute Rating Technique Exploiting Ranks* (SMARTER) merupakan sistem yang dibangun dengan tujuan untuk mempermudah para wisatawan yang berkunjung ke Kawah Ijen dalam memilih *homestay* sesuai dengan kriteria-kriteria yang diinginkan. Adapun kriteria-kriteria tersebut adalah fasilitas Utama dan fasilitas Tambahan. Kriteria tersebut didasarkan dari pencarian *homestay* oleh wisatawan secara umum. Dari kedua kriteria tersebut memiliki sub kriteria yang nantinya akan dipilih oleh wisatawan. Sub kriteria untuk kriteria fasilitas utama yaitu harga, jarak ke ijen, jumlah kamar , dan jarak ke sarana umum. Sub kriteria untuk kriteria fasilitas tambahan yaitu wifi, transportasi, dan katering. Pemberian prioritas pada setiap kriteria, sub kriteria, dan sub-sub kriteria dilakukan oleh *user*, sehingga *user* dapat memilih prioritas sesuai dengan keinginannya. Sistem menghasilkan nilai tertinggi

sampai terendah dari *homestay* yang telah dihitung menggunakan metode SMARTER. Terdapat fitur detail perhitungan, sehingga *user* mengetahui nilai dari *homestay* yang dipilih, dan juga fitur booking bagi *user* yang akan memesan *homestay* yang telah dipilih

4.1.2. Fungsi Sistem

Fungsi Utama dari Sistem Informasi Rekomendasi *Homestay* menggunakan metode *Simple Multi Attribute Rating Technique Exploiting Ranks* (SMARTER) ini yaitu mencari *homestay* sesuai keinginan dari wisatawan. Sistem ini memiliki 2 *user* yaitu admin (pihak Badan Usaha Milik Desa (BUMDES) kecamatan Licin) dan *user* (wisatawan/umum).

a. Admin (Pihak BUMDES)

Admin memiliki hak akses untuk mengelola data *homestay*. Saat admin berhasil melakukan *login*, sistem akan menampilkan halaman *home* bagi admin. Admin dapat mengakses halaman *Homestay* untuk mengelola data *homestay*, halaman Informasi untuk mengelola data informasi paket wisata bagi wisatawan, dan halaman Booking untuk melakukan mengelola *booking* bagi wisatawan yang memesan *homestay*.

b. *User* (wisatawan/umum)

User dapat mencari *homestay* dengan cara memberikan prioritas kepada kriteria-sub kriteri da- sub-sub kriteria sesuai keinginannya. *User* juga dapat memesan *homestay* yang diinginkannya, dan melihat detail perhitungan dari setiap *homestay* yang dipilih. *User* juga dapat melihat detail pemesanan.

4.2. Analisis Kebutuhan Sistem

Tahap pertama pada proses analisis kebutuhan perancangan perangkat lunak ini adalah analisis kebutuhan. Kebutuhan yang digunakan yaitu dari hasil wawancara kepada pihak-pihak pemilik *homestay* dan pihak Badan Usaha Milik Desa

(BUMDES). Data-data yang telah didapat kemudian dikelompokkan menjadi kebutuhan fungsional dan kebutuhan non-fungsional.

4.2.1. Kebutuhan Fungsional

Kebutuhan fungsional ini berisi tentang fitur-fitur penting yang akan digunakan dalam sistem, sehingga fungsi sistem dapat di jalankan sesuai dengan kebutuhan pengguna. Kebutuhan fungsional Sistem Informasi Rekomendasi *Homestay* menggunakan metode *Simple Multi-Attribute Rating Technique Exploiting Ranks* (SMARTER) adalah sebagai berikut :

1. Sistem mampu mengelola data *homestay* (*view, insert, edit, delete*).
2. Sistem mampu mengelola data informasi paket wisata (*view, edit*).
3. Sistem mampu mengelola data *booking* (*view, insert, edit, delete, confirm*).
4. Sistem mampu mengubah data admin.
5. Sistem mampu menampilkan detail pemesanan.
6. Sistem mampu melakukan perhitungan rekomendasi *homestay* menggunakan metode *Simple Multi Attribute Rating Technique Exploiting Ranks* (SMARTER).
7. Sistem mampu menampilkan rekomendasi *homestay* terbaik.

4.2.2. Kebutuhan Non-Fungsional

Kebutuhan non-fungsional ini berisi tentang fitur-fitur penting yang akan digunakan dalam mendukung kebutuhan fungsionalitas, sehingga memenuhi kebutuhan pengguna. Kebutuhan non-fungsional Sistem Informasi Rekomendasi *Homestay* menggunakan metode *Simple Multi-Attribute Rating Technique Exploiting Ranks* (SMARTER) adalah sebagai berikut :

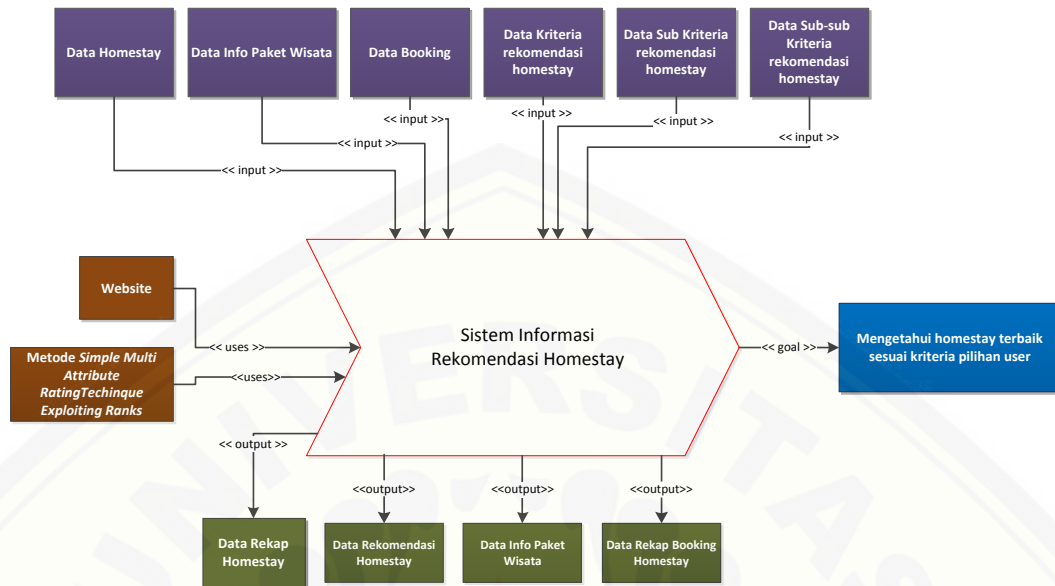
1. Sistem yang mudah digunakan (*userfriendly*)
2. Sistem dapat diakses pada *platform* yang berbeda dengan berbasis website.
3. Sistem memiliki *username* dan *password* untuk autentikasi dalam hak akses.

4.3. Desain Sistem

Tahap selanjutnya setelah tahap analisis kebutuhan adalah desain sistem. Desain sistem yang akan digunakan yaitu, *business process*, *usecase diagram*, *scenario*, *activity diagram*, *sequence diagram*, *class diagram* dan *entity relationship diagram (ERD)*.

4.3.1. Business Process

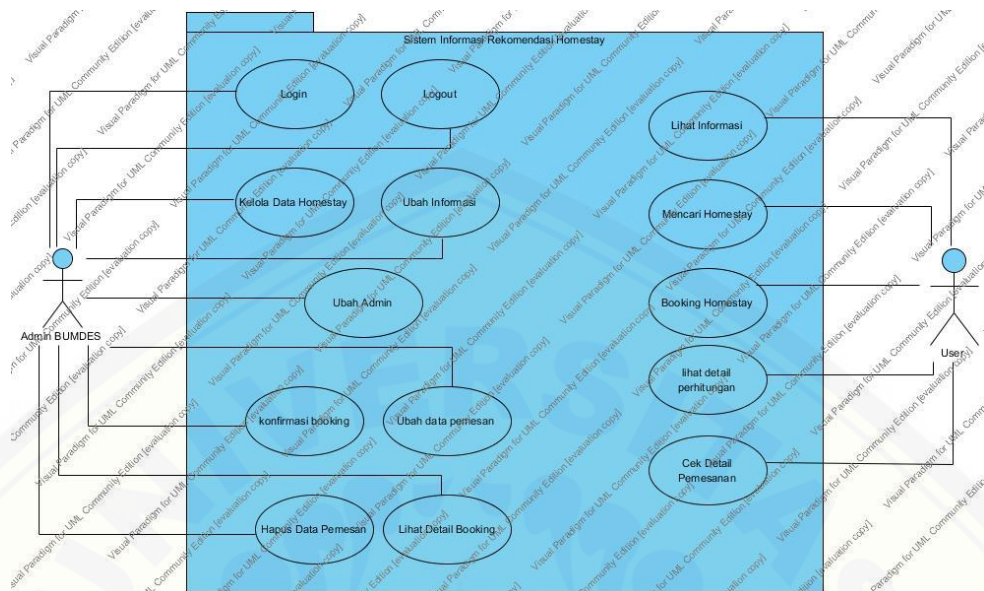
Business Process merupakan gambaran data yang dibutuhkan sistem sebagai input yang merupakan data masukan kedalam sistem, inputan dari sistem informasi rekomendasi *homestay* yaitu Data *Homestay*, Data Info Paket Wisata, Data *Booking*, Data Kriteria Rekomendasi *Homestay*, Data Sub Kriteria Rekomendasi *Homestay*, dan Data Sub-Sub Kriteria Rekomendasi *Homestay*. Selanjutnya inputan tersebut diproses menjadi output yang merupakan keluaran dari sistem, output dari sistem informasi rekomendasi *homestay* yaitu berupa Data Rekap *Homestay*, Data Rekomendasi *Homestay*, Data Info Paket Wisata, dan Data Rekap *Booking Homestay*. Inputan dan outputan tersebut sesuai dengan *goal* yang merupakan tujuan pembuatan sistem. *Business process* juga terdiri dari *used* dan *process*. *Uses* merupakan *platform* yang digunakan untuk basis sistem yaitu berbasis *website* serta menggunakan metode SMARTER, dan *process* adalah sistem yang bekerja. *Business process* pada Sistem Informasi Rekomendasi *Homestay* menggunakan metode *Simple Multi-Attribute Rating Technique Exploiting Ranks* (SMARTER) dapat dilihat pada Gambar 4.1 *Business Process*.



Gambar 4.1 Business Process

4.3.2. Usecase Diagram

Usecase diagram merupakan gambaran fitur-fitur apa saja yang tersedia serta aktor yang mempunyai hak akses kedalam sistem informasi rekomendasi *homestay*. Penjelasan deskripsi aktor pada sistem informasi rekomendasi *homestay* dapat dilihat pada Gambar 4.2 *Usecase Diagram*. Penjelasan deskripsi *usecase* pada sistem informasi rekomendasi *homestay* dapat dilihat pada Tabel 4.1 Deskripsi Aktor. Penjelasan *usecase diagram* pada sistem informasi rekomendasi *homestay* dapat dilihat pada Tabel 4.2 *Deskripsi Usecase*.



Gambar 4.2 Usecase Diagram

Tabel 4.1 Deskripsi Aktor

No.	Aktor	Deskripsi
1	Admin	Aktor admin memiliki hak akses untuk mengelola data <i>homestay</i> , mengubah informasi paket wisata, mengelola data <i>booking</i> seperti mengubah data pemesan, melihat detail <i>booking</i> , menghapus <i>booking</i> , dan memberikan persetujuan <i>booking</i> .
2.	User	Aktor <i>user</i> memiliki hak akses untuk mencari <i>homestay</i> , melakukan pemesanan, melihat detail perhitungan, cek pemesanan, dan melihat informasi paket wisata.

Tabel 4.2 Deskripsi Usecase

No.	Usecase	Deskripsi
1.	login	Usecase yang digunakan untuk mengelola data <i>login</i> untuk admin
2.	Kelola Data <i>Homestay</i>	Usecase yang digunakan untuk mengelola data <i>homestay</i> mulai dari menambah, mengedit, mnghapus, dan melihat.
3.	Ubah Informasi	Usecase yang digunakan untuk merubah data informasi paket wisata

4.	Lihat Informasi	<i>Usecase</i> yang digunakan untuk melihat informasi paket wisata
5	Ubah admin	<i>Usecase</i> yang digunakan untuk merubah data admin
6.	<i>Booking Homestay</i>	<i>Usecase</i> yang digunakan untuk memesan <i>homestay</i> yang dipilih.
7.	Ubah <i>Booking</i>	<i>Usecase</i> yang digunakan untuk merubah data pemesan.
8.	Hapus Data Pemesan	<i>Usecase</i> yang digunakan untuk menghapus data pemesan.
9.	Lihat Detail <i>Booking</i>	<i>Usecase</i> yang digunakan untuk melihat detail <i>booking</i>
10.	Konfirmasi <i>Booking</i>	<i>Usecase</i> yang digunakan untuk memberikan persetujuan pada <i>booking</i> .
11.	Cek Detail Pemesanan	<i>Usecase</i> yang digunakan untuk memeriksa detail pemesanan
12.	Mencari <i>Homestay</i>	<i>Usecase</i> yang digunakan untuk mencari <i>homestay</i> berdasarkan kriteria, sub kriteria, dan sub-sub kriteria yang sudah ditentukan.
13.	Lihat detail Perhitungan	<i>Usecase</i> yang digunakan untuk melihat detail perhitungan menggunakan metode SMARTER pada setiap <i>homestay</i>
14.	Logout	<i>Usecase</i> yang digunakan aktor admin untuk keluar dari sistem setelah selesai menggunakan sistem.

4.3.3. Skenario Sistem

Skenario sistem menjelaskan bagaimana alur sistem bekerja dan keadaan yang terjadi ketika terjadi aksi tertentu dari aktor dalam menjalankan sistem informasi rekomendasi *homestay*. Skenario sistem sesuai dengan *usecase diagram* pada Gambar 4.2, yaitu sebagai berikut :

1. Skenario *Login*

Penjelasan dan urutan aksi aktor dan reaksi sistem mulai dari skenario normal dan skenario alternatif *login* ditunjukkan pada lampiran A.

2. Skenario Kelola Data *Homestay*

Penjelasan dan urutan aksi aktor dan reaksi sistem mulai dari skenario normal dan skenario alternatif kelola data *homestay* ditunjukkan pada lampiran A.

3. Skenario Ubah Informasi

Penjelasan dan urutan aksi aktor dan reaksi sistem mulai dari skenario normal dan skenario alternatif ubah informasi ditunjukkan pada lampiran A.

4. Skenario Lihat Informasi

Penjelasan dan urutan aksi aktor dan reaksi sistem mulai dari skenario normal dan skenario alternatif lihat informasi ditunjukkan pada lampiran A.

5. Skenario Ubah Admin

Penjelasan dan urutan aksi aktor dan reaksi sistem mulai dari skenario normal dan skenario alternatif ubah admin ditunjukkan pada lampiran A.

6. Skenario *Booking Homestay*

Penjelasan dan urutan aksi aktor dan reaksi sistem mulai dari skenario normal dan skenario alternatif *booking homestay* ditunjukkan pada lampiran A.

7. Skenario Ubah Data Pemesan

Penjelasan dan urutan aksi aktor dan reaksi sistem mulai dari skenario normal dan skenario alternatif ubah data pemesan ditunjukkan pada lampiran A.

8. Skenario Hapus Data Pemesan

Penjelasan dan urutan aksi aktor dan reaksi sistem mulai dari skenario normal dan skenario alternatif hapus data pemesan ditunjukkan pada lampiran A.

9. Skenario Lihat Detail Booking

Penjelasan dan urutan aksi aktor dan reaksi sistem mulai dari skenario normal dan skenario alternatif lihat detail *booking* ditunjukkan pada lampiran A.

10. Skenario Konfirmasi *Booking*

Penjelasan dan urutan aksi aktor dan reaksi sistem mulai dari skenario normal dan skenario alternatif konfirmasi *booking* ditunjukkan pada lampiran A.

11. Skenario Cek Detail Pemesanan

Penjelasan dan urutan aksi aktor dan reaksi sistem mulai dari skenario normal dan skenario alternatif cek detail pemesanan ditunjukkan pada lampiran A.

12. Skenario Mencari *Homestay*

Penjelasan dan urutan aksi aktor dan reaksi sistem mulai dari skenario normal dan skenario alternatif mencari *homestay* ditunjukkan Tabel 4.3 Skenario Mencari *Homestay*.

Tabel 4.3 Skenario Mencari *Homestay*

Nama <i>Usecase</i>	Mencari <i>Homestay</i>
Partisipan Aktor	<i>User</i>
Pre-Kondisi	<i>User</i> ingin melakukan pencarian <i>homestay</i>
Pra-Kondisi	<i>User</i> telah melakukan pencarian <i>homestay</i>
<i>Event flow</i>	<i>User</i> membuka website <i>User</i> memilih menu Cari <i>Homestay</i>
SKENARIO Normal “Mencari <i>Homestay</i>”	

Aksi Aktor	Reaksi Sistem
1. Klik menu Cari <i>Homestay</i> .	
	2. Menampilkan form sub kriteria fasilitas utama dan fasilitas tambahan
3. Klik tombol Default untuk mengisi prioritas 1,2,3,4 sub kriteria fasilitas utama dan mengisi prioritas 1,2,3 sub kriteria fasilitas tambahan	
	4. Mengisi prioritas sub kriteria fasilitas utama dan fasilitas tambahan
5. Klik tombol Next	
	6. Menampilkan form sub-sub kriteria fasilitas utama dan fasilitas tambahan
7. Klik tombol Default untuk mengisi prioritas 1,2,3 sub-sub kriteria harga, jarak ke ijen, jumlah kamar, k, jarak ke sarana umum dan prioritas 1,2 sub-sub kriteria wifi, transportasi, katering.	
	8. Mengisi prioritas sub-sub kriteria fasilitas utama dan fasilitas tambahan
9. Klik tombol Cari	

10. Menampilkan rekomendasi *homestay* dengan 6 nilai tertinggi.

SKENARIO Alternatif
“Melihat Semua Homestay”

Aksi Aktor

Reaksi Sistem

11a. Klik tombol lihat semua

12a. Menampilkan semua *homestay*

SKENARIO Alternatif
“Klik Tombol Kembali”

Aksi Aktor

Reaksi Sistem

11b. Klik tombol kembali

12b. Menampilkan halaman utama

SKENARIO ALTERNATIF
“Data Tidak Lengkap”

Aksi Aktor

Reaksi Sistem

5c. Klik tombol Next

6c. Menampilkan message “*please select an item in this list*”

SKENARIO ALTERNATIF
“Data Tidak Lengkap”

9d. Klik tombol Cari

10d. Menampilkan message “*please select an item in this list*”

13. Skenario Lihat Detail Perhitungan

Penjelasan dan urutan aksi aktor dan reaksi sistem mulai dari skenario normal dan skenario alternatif lihat detail perhitungan ditunjukkan pada Lampiran A.

4.3.4. *Sequence Diagram*

Sequence Diagram merupakan gambaran interaksi antar objek di dalam sistem. Interaksi antar objek yaitu mulai dari aksi aktor kepada interface, kemudian sistem memberikan reaksi kepada aktor melalui proses internal pada sistem informasi rekomendasi *homestay*, sehingga sistem akan menampilkan keluaran sesuai dengan keinginan aktor. *Sequence diagram* pada sistem informasi rekomendasi *homestay* adalah sebagai berikut :

1. *Sequence Diagram Login*

Penggambaran *Sequence Diagram login* digunakan untuk pembuatan fungsi atau method yang ditunjukkan pada lampiran B.

2. *Sequence Diagram Kelola Data Homestay*

Penggambaran *Sequence Diagram kelola data homestay* digunakan untuk pembuatan fungsi atau method yang ditunjukkan pada lampiran B.

3. *Sequence Diagram Ubah Informasi*

Penggambaran *Sequence Diagram ubah informasi* digunakan untuk pembuatan fungsi atau method yang ditunjukkan pada lampiran B.

4. *Sequence Diagram Lihat Informasi*

Penggambaran *Sequence Diagram lihat informasi* digunakan untuk pembuatan fungsi atau method yang ditunjukkan pada lampiran B.

5. *Sequence Diagram* Ubah Admin

Penggambaran *Sequence Diagram* ubah admin digunakan untuk pembuatan fungsi atau method yang ditunjukkan pada lampiran B.

6. *Sequence Diagram* Booking Homestay

Penggambaran *Sequence Diagram* booking homestay digunakan untuk pembuatan fungsi atau method yang ditunjukkan pada lampiran B.

7. *Sequence Diagram* Ubah Booking

Penggambaran *Sequence Diagram* ubah booking digunakan untuk pembuatan fungsi atau method yang ditunjukkan pada lampiran B.

8. *Sequence Diagram* Hapus Data Pemesan

Penggambaran *Sequence Diagram* hapus data pemesan digunakan untuk pembuatan fungsi atau method yang ditunjukkan pada lampiran B.

9. *Sequence Diagram* Lihat Detail Booking

Penggambaran *Sequence Diagram* lihat detail booking digunakan untuk pembuatan fungsi atau method yang ditunjukkan pada lampiran B.

10. *Sequence Diagram* Konfirmasi Booking

Penggambaran *Sequence Diagram* konfirmasi booking digunakan untuk pembuatan fungsi atau method yang ditunjukkan pada lampiran B.

11. *Sequence Diagram* Cek Detail Pemesanan

Penggambaran *Sequence Diagram* cek detail pemesanan digunakan untuk pembuatan fungsi atau method yang ditunjukkan pada lampiran B.

13. *Sequence Diagram* Lihat Detail Perhitungan

Penggambaran *Sequence Diagram* lihat detail perhitungan digunakan untuk pembuatan fungsi atau method yang ditunjukkan pada Lampiran B.

4.3.5. Activity Diagram

Activity Diagram menggambarkan aliran aktivitas dalam sistem informasi rekomendasi *homestay*, mulai dari alir dimulai, decision yang terjadi dan akhir dari alir. *Activity Diagram* sistem informasi rekomendasi *homestay* sebagai berikut :

1. *Activity Diagram Login*

Alur *Activity Diagram login* ditunjukkan pada lampiran C.

2. *Activity Diagram Kelola Data Homestay*

Alur *Activity Diagram kelola data homestay* ditunjukkan pada lampiran C.

3. *Activity Diagram Ubah Informasi*

Alur *Activity Diagram ubah informasi* ditunjukkan pada lampiran C.

4. *Activity Diagram Lihat Informasi*

Alur *Activity Diagram lihat informasi* ditunjukkan pada lampiran C.

5. *Activity Diagram Ubah Admin*

Alur *Activity Diagram ubah admin* ditunjukkan pada lampiran C.

6. *Activity Diagram Booking Homestay*

Alur *Activity Diagram booking homestay* ditunjukkan pada lampiran C.

7. *Activity Diagram Ubah Data Pemesan*

Alur *Activity Diagram ubah data pemesan* ditunjukkan pada lampiran C.

8. *Activity Diagram* Hapus Data Pemesan

Alur *Activity Diagram* hapus data pemesanan ditunjukkan pada lampiran C.

9. *Activity Diagram* Lihat Detail Booking

Alur *Activity Diagram* lihat detail *booking* ditunjukkan pada lampiran C.

10. *Activity Diagram* Konfirmasi Booking

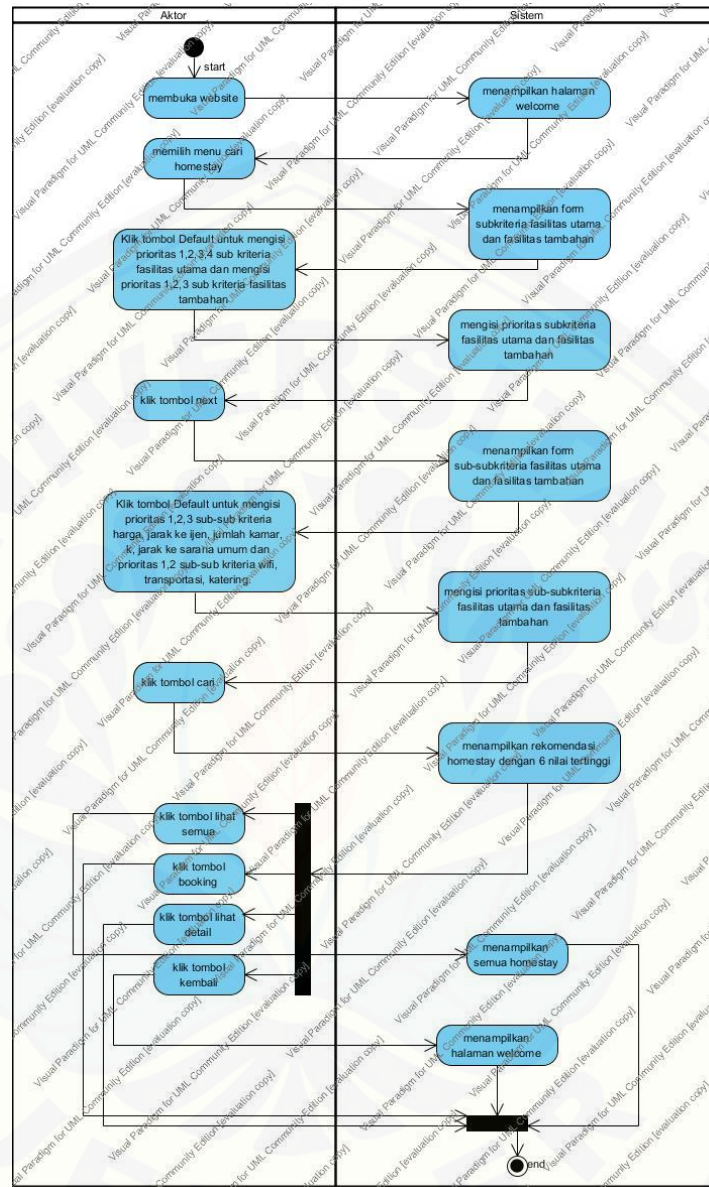
Alur *Activity Diagram* konfirmasi *booking* ditunjukkan pada lampiran C.

11. *Activity Diagram* Cek Detail Pemesanan

Alur *Activity Diagram* cek detail ditunjukkan pada lampiran C.

12. *Activity Diagram* Mencari Homestay

Alur *Activity Diagram* mencari *homestay* ditunjukkan pada Gambar 4.4 *Activity Diagram* Mencari Homestay.



Gambar 4.4 Activity Diagram Mencari Homestay

13. Activity Diagram Lihat Detail Perhitungan

Alur Activity Diagram lihat detail perhitungan ditunjukkan pada Lampiran C.

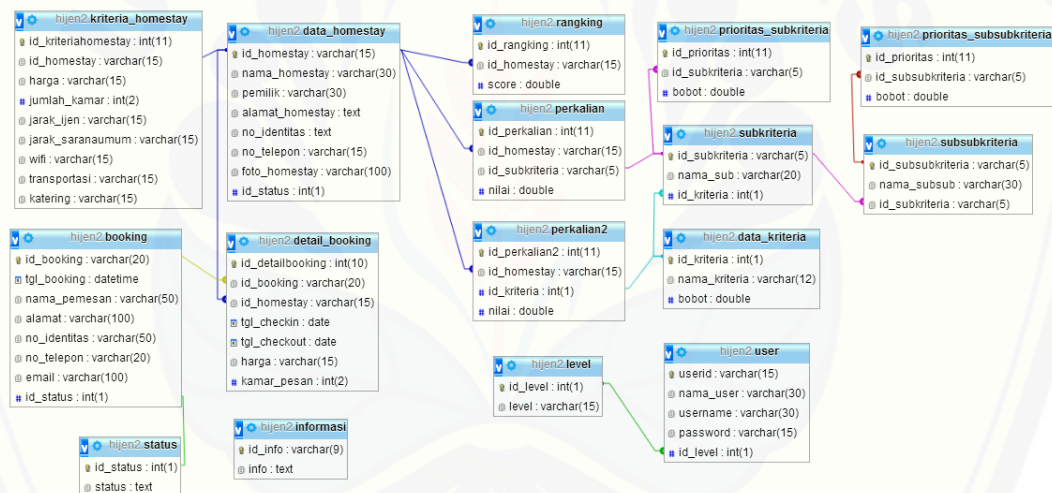
4.3.6. Class Diagram

Class Diagram merupakan gambaran dari struktur dan deskripsi dari class, package dan objek pada sistem informasi rekomendasi *homestay* yang berhubungan satu sama lain seperti pewarisan, asosiasi, dependensi dan lain-lain. *Class Diagram* pada sistem informasi rekomendasi *homestay* dapat dilihat pada Gambar 4.6 *Class Diagram*.



4.3.7. Entity Relationship Diagram

Entity Relationship Diagram merupakan gambaran struktur database yang akan dibangun pada sistem informasi rekomendasi *homestay*. Sebagai contoh pada ERD sistem informasi rekomendasi *homestay* ini memiliki tabel status yang berelasi dengan tabel *booking*. Relasi tersebut dapat dilihat dari *id_status* yang menjadi *primary key* pada tabel status, juga terdapat pada tabel *booking* sebagai *foreign key*, sehingga relasi pada kedua tabel tersebut memiliki arti bahwa satu status bisa digunakan pada banyak transaksi pada tabel *booking*. *Entity Relationship Diagram* pada sistem informasi rekomendasi *homestay* dapat dilihat pada Gambar 4.7 *Entity Relationship Diagram*.



Gambar 4.7 Entity Relationship Diagram

4.4. Implementasi

Pada tahap implementasi, desain yang telah dibuat diimplementasikan ke dalam kode program. Desain yang telah dibuat meliputi fitur-fitur yang dapat dilihat pada Gambar 4.2 *Usecase Diagram*. Kode program sistem informasi rekomendasi *homestay* ini menggunakan bahasa pemrograman *Page Hyper Text Pre-Processor*

(PHP) dengan bantuan *framework Code Igniter (CI)* untuk memudahkan dalam pengembangan dan pembuatan program.

4.4.1. Kode Program Login

Kode program *login* terletak pada kelas *view v_login, controller login, dan model m_login*. Penulisan kode program dapat dilihat pada lampiran D.

4.4.2. Kode Program Kelola Data Homestay

Kode program kelola data *homestay* terletak pada kelas *view a_homestay, controller halaman_admin dan form_homestay, serta model m_form_homestay*. Penulisan kode program dapat dilihat pada lampiran D.

4.4.3. Kode Program Ubah Informasi

Kode program ubah informasi terletak pada kelas *view a_info, controller halaman_admin dan form_informasi, serta model m_form_informasi*. Penulisan kode program dapat dilihat pada lampiran D.

4.4.4. Kode Program Lihat Informasi

Kode program lihat informasi terletak pada kelas *view welcome, controller login, dan model m_form_informasi*. Penulisan kode program dapat dilihat pada lampiran D.

4.4.5. Kode Program Ubah Admin

Kode program ubah admin terletak pada kelas *view a_ubah_admin, controller halaman_admin dan form_admin, serta model m_form_user*. Penulisan kode program dapat dilihat pada lampiran D.

4.4.6. Kode Program Booking Homestay

Kode program *booking homestay* terletak pada kelas *view w_formbooking, controller halaman_user dan form_booking, serta model m_form_booking*. Penulisan kode program dapat dilihat pada lampiran D.

4.4.7. Kode Program Ubah Data Pemesan

Kode program ubah data pemesan terletak pada kelas *view* *a_ubah_booking*, *controller* *halaman_admin* dan *form_booking*, serta *model* *m_form_booking*. Penulisan kode program dapat dilihat pada lampiran D.

4.4.8. Kode Program Hapus Data Pemesan

Kode program hapus data pemesan terletak pada kelas *view* *a_booking*, *controller* *halaman_admin* dan *form_booking*, serta *model* *m_form_booking*. Penulisan kode program dapat dilihat pada lampiran D.

4.4.9. Kode Program Lihat Detail *Booking*

Kode program lihat detail *booking* terletak pada kelas *view* *a_booking*, *controller* *halaman_admin* dan *form_booking*, serta *model* *m_form_booking*. Penulisan kode program dapat dilihat pada lampiran D.

4.4.10. Kode Program Konfirmasi *Booking*

Kode program konfirmasi *booking* terletak pada kelas *view* *a_booking*, *controller* *halaman_admin* dan *form_booking*, serta *model* *m_form_booking*. Penulisan kode program dapat dilihat pada lampiran D.

4.4.11. Kode Program Cek Detail Pemesanan

Kode program cek detail pemesanan terletak pada kelas *view* *w_formbooking*, *controller* *halaman_user* dan *model* *m_form_booking*. Penulisan kode program dapat dilihat pada lampiran D.

4.4.12. Kode Program Mencari *Homestay*

Kode program mencari *homestay* terletak pada kelas *view* *w_carihomestay*, *w_carihomestay2*, *w_rekomendasi*. *Controller* *halaman_user*. *Model* *m_form_carihomestay* dan *m_smarter*.

1. Model *m_form_carihomestay*

Pada model *m_form_carihomestay* terdapat beberapa fungsi yang digunakan dalam pencarian homestay yaitu method `simpanprioritassubkriteria()` dan method `simpanprioritassubsubkriteria()`.

Method `simpanprioritassubkriteria()` berfungsi untuk menyimpan prioritas sub kriteria yang terletak pada dan method `simpanprioritassubsubkriteria()` berfungsi untuk menyimpan prioritas sub-sub kriteria. Penulisan kode program *m_form_carihomestay* dapat dilihat pada Tabel 4.4 Fungsi Menyimpan Prioritas Sub Kriteria, Tabel 4.5 Fungsi Menyimpan Prioritas Sub-sub Kriteria.

Tabel 4.4 Fungsi Menyimpan Prioritas Sub Kriteria

No.	Kode Program
1.	<code>function simpanprioritassubkriteria(\$datau, \$datak){</code>
2.	<code> \$roc2 = [0.75, 0.25];</code>
3.	<code> \$roc3 = [0.611, 0.278, 0.111];</code>
4.	<code> \$roc4 = [0.521, 0.271, 0.146, 0.063];</code>
5.	<code> \$roc5 = [0.457, 0.257, 0.157, 0.09, 0.04];</code>
6.	<code> \$roc6 = [0.408, 0.242, 0.158, 0.103, 0.061, 0.028];</code>
7.	<code> \$roc7 = [0.37, 0.228, 0.156, 0.109, 0.073, 0.044, 0.028];</code>
8.	<code> \$roc8 = [0.34, 0.215, 0.152, 0.111, 0.079, 0.054, 0.034, 0.016];</code>
9.	<code> \$roc9 = [0.314, 0.203, 0.148, 0.111, 0.083, 0.061, 0.042, 0.026,</code>
10.	<code> 0.012];</code>
11.	<code> \$this->db->query("truncate prioritas_subkriteria");</code>
12.	<code> for (\$i=0; \$i < 4 ; \$i++) {</code>
13.	<code> \$this->db->query("insert into prioritas_subkriteria values (',</code>
14.	<code> '\$datau[\$i]', '\$roc4[\$i]')");</code>
15.	<code> }</code>
16.	<code> for (\$i=0; \$i < 3 ; \$i++) {</code>
17.	<code> \$this->db->query("insert into prioritas_subkriteria values (',</code>
18.	<code> '\$datak[\$i]', '\$roc3[\$i]')");</code>
19.	<code> }</code>
20.	<code>}</code>

Tabel 4.5 Fungsi Menyimpan Prioritas Sub-Sub Kriteria

No.	Kode Program
1.	function simpanprioritassubsubkriteria(\$harga, \$jarakijen, \$jaraksarana,
2.	\$jumlahkamar, \$wifi, \$transportasi, \$katering){
3.	\$roc2 = [0.75, 0.25];
4.	\$roc3 = [0.611, 0.278, 0.111];
5.	\$roc4 = [0.521, 0.271, 0.146, 0.063];
6.	\$roc5 = [0.457, 0.257, 0.157, 0.09, 0.04];
7.	\$roc6 = [0.408, 0.242, 0.158, 0.103, 0.061, 0.028];
8.	\$roc7 = [0.37, 0.228, 0.156, 0.109, 0.073, 0.044, 0.028];
9.	\$roc8 = [0.34, 0.215, 0.152, 0.111, 0.079, 0.054, 0.034, 0.016];
10.	\$roc9 = [0.314, 0.203, 0.148, 0.111, 0.083, 0.061, 0.042, 0.026, 0.012];
11.	\$this->db->query("truncate prioritas_subsubkriteria");
12.	for (\$i=0; \$i < 3 ; \$i++) {
13.	\$this->db->query("insert into prioritas_subsubkriteria values (",
14.	'\$harga[\$i]', '\$roc3[\$i]");
15.	}
16.	for (\$i=0; \$i < 3 ; \$i++) {
17.	\$this->db->query("insert into prioritas_subsubkriteria values (",
18.	'\$jarakijen[\$i]', '\$roc3[\$i]");
19.	}
20.	for (\$i=0; \$i < 3 ; \$i++) {
21.	\$this->db->query("insert into prioritas_subsubkriteria values (",
22.	'\$jaraksarana[\$i]', '\$roc3[\$i]");
23.	}
24.	for (\$i=0; \$i < 3 ; \$i++) {
25.	\$this->db->query("insert into prioritas_subsubkriteria values (",
26.	'\$jumlahkamar[\$i]', '\$roc3[\$i]");
27.	}
28.	for (\$i=0; \$i < 2 ; \$i++) {
29.	\$this->db->query("insert into prioritas_subsubkriteria values (",
30.	'\$wifi[\$i]', '\$roc2[\$i]");
31.	}
32.	for (\$i=0; \$i < 2 ; \$i++) {
33.	\$this->db->query("insert into prioritas_subsubkriteria values (",
34.	'\$transportasi[\$i]', '\$roc2[\$i]");
35.	}
36.	for (\$i=0; \$i < 2 ; \$i++) {
37.	\$this->db->query("insert into prioritas_subsubkriteria values (",
38.	'\$katering[\$i]', '\$roc2[\$i]");
39.	}
40.	}

2. Kelas *model m_smarter*

Model m_smarter memiliki satu *method* yang berfungsi melakukan perhitungan metode SMARTER yaitu *method* *hitung()*. Penulisan kode program *model m_smarter* dapat dilihat pada Tabel 4.6 Fungsi Perhitungan Metode *Simple Multi-Attribute Rating Technique Exploiting Ranks* (SMARTER), Tabel 4.7 Fungsi Menampilkan Rekomendasi *Homestay* Terbaik.

Tabel 4.6 Fungsi Perhitungan Metode *Simple Multi-Attribute Rating Technique Exploiting Ranks* (SMARTER)

No.	Kode Program
1.	<code>public function hitung(){</code>
2.	<code> \$this->db->query("truncate ranking");</code>
3.	<code> \$this->db->query("truncate perkalian");</code>
4.	<code> \$this->db->query("truncate perkalian2");</code>
5.	<code> \$query1 = \$this->db->query("select count(*) as jumlah from</code>
6.	<code> data_homestay ");</code>
7.	<code> foreach (\$query1->result_array() as \$s) {</code>
8.	<code> \$num = \$s['jumlah'];</code>
9.	<code> }</code>
10.	<code> \$subumum = ["sub01","sub02","sub03","sub04"];</code>
11.	<code> \$subkhusus = ["sub05","sub06","sub07"];</code>
12.	<code> \$sub1 = array();</code>
13.	<code> \$sub2 = array();</code>
14.	<code> \$queryu = \$this->db->query("select * from data_kriteria</code>
15.	<code> where nama_kriteria = 'Umum');</code>
16.	<code> foreach (\$queryu->result_array() as \$v) {</code>
17.	<code> \$bobotumum = \$v['bobot'];</code>
18.	<code> }</code>
19.	<code> \$queryk = \$this->db->query("select * from data_kriteria</code>
20.	<code> where nama_kriteria = 'Khusus');</code>
21.	<code> foreach (\$queryk->result_array() as \$va) {</code>
22.	<code> \$bobotkhusus = \$va['bobot'];</code>
23.	<code> }</code>
24.	<code> for (\$i=0; \$i < \$num ; \$i++) {</code>
25.	<code> //mengambil data home stay disesuaikan dengan</code>
26.	<code> prioritas user</code>
27.	<code> \$query2 = \$this->db->query("select * from</code>
28.	<code> data_homestay d join kriteria_homestay k on d.id_homestay =</code>


```
29. k.id_homestay limit $i, 1");
30.     foreach ($query2->result_array() as $value) {
31.         $id = $value['id_homestay'];
32.         $sub1[0] = 0;
33.         if ($value['harga'] == 150000) {
34.             $sub1[0] = "Rp 150.000";
35.         }else if ($value['harga'] == 200000) {
36.             $sub1[0] = "Rp 200.000";
37.         }else if ($value['harga'] > 200000) {
38.             $sub1[0] = "> Rp 200.000";
39.         }
40.         $sub1[1] = 0;
41.         if ($value['jarak_ijen'] < 10) {
42.             $sub1[1] = "< 10 km";
43.         }else if ($value['jarak_ijen'] >= 10 &&
44. $value['jarak_ijen'] <= 20) {
45.             $sub1[1] = "10 km - 20 km";
46.         }else if ($value['jarak_ijen'] > 20) {
47.             $sub1[1] = "> 20 km";
48.         }
49.         $sub1[2] = 0;
50.         if ($value['jumlah_kamar'] == 1) {
51.             $sub1[2] = "1 kamar";
52.         }else if ($value['jumlah_kamar'] == 2) {
53.             $sub1[2] = "2 kamar";
54.         }else if ($value['jumlah_kamar'] >= 3) {
55.             $sub1[2] = ">= 3 kamar";
56.         }
57.         $sub1[3] = 0;
58.         if ($value['jarak_saranaumum'] < 3) {
59.             $sub1[3] = "< 3 km";
60.         }else if ($value['jarak_saranaumum'] >= 3 &&
61. $value['jarak_saranaumum'] <= 7) {
62.             $sub1[3] = "3 km - 7 km";
63.         }else if ($value['jarak_saranaumum'] >7) {
64.             $sub1[3] = "> 7 km";
65.         }
66.         $sub2[0] = $value['wifi'];
67.         $sub2[1] = $value['transportasi'];
68.         $sub2[2] = $value['katering'];
69.     }
70.
```



```
71. //menghitung step 2 prioritas umum
72. $jumlahumum = 0; $bobotsubu = 0; $bobotsubsubu=0;
73. for ($j=0; $j < 4 ; $j++) {
74. $query3 = $this->db->query("select a.id_subkriteria, bobot from
75. subkriteria a join prioritas_subkriteria b on (a.id_subkriteria =
76. b.id_subkriteria) where a.id_subkriteria = '$subumum[$j]'");
77.     foreach ($query3->result_array() as $val) {
78.         $bobotsubu = $val['bobot'];
79.         $idsubu = $val['id_subkriteria'];
80.     }
81. $query4 = $this->db->query("select nama_subsub, bobot from
82. subsubkriteria a join prioritas_subsubkriteria b on
83. (a.id_subsubkriteria = b.id_subsubkriteria) where a.id_subkriteria
84. = '$subumum[$j]' and nama_subsub = '$sub1[$j]'");
85.     foreach ($query4->result_array() as $valu) {
86.         $bobotsubsubu = $valu['bobot'];
87.     }
88.     $hasilu = $bobotsubu * $bobotsubsubu;
89.     $jumlahumum = $jumlahumum + $hasilu;
90.     $this->db->query("insert into perkalian values ('', 'Sid',
91. 'Sidsubu', '$hasilu')");
92. }
93. //menghitung step 2 prioritas khusus
94. $jumlahkhusus = 0; $bobotsubk = 0; $bobotsubsubk=0;
95. for ($k=0; $k < 3 ; $k++) {
96. $query5 = $this->db->query("select a.id_subkriteria, bobot from
97. subkriteria a join prioritas_subkriteria b on (a.id_subkriteria =
98. b.id_subkriteria) where a.id_subkriteria = '$subkhusus[$k]'");
99.     foreach ($query5->result_array() as $values) {
100.         $bobotsubk = $values['bobot'];
101.         $idsubk = $values['id_subkriteria'];
102.     }
103. $query6 = $this->db->query("select nama_subsub, bobot from
104. subsubkriteria a join prioritas_subsubkriteria b on
105. (a.id_subsubkriteria = b.id_subsubkriteria) where a.id_subkriteria
106. = '$subkhusus[$k]' and nama_subsub = '$sub2[$k]'");
107.     foreach ($query6->result_array() as $valued) {
108.         $bobotsubsubk = $valued['bobot'];
109.     }
110.     $hasilk = $bobotsubk * $bobotsubsubk;
111.     $this->db->query("insert into perkalian values ('', 'Sid', 'Sidsubk',
112. '$hasilk')");
```

113.	\$jumlahkhusus = \$jumlahkhusus + \$hasilk;
114.	}
115.	//menghitung bobot akhir umum dan khusus
116.	\$bobotakhirumum = \$jumlahumum * \$bobotumum;
117.	\$bobotakhirkhusus = \$jumlahkhusus * \$bobotkhusus;
118.	//menghitung total
119.	\$total = \$bobotakhirumum + \$bobotakhirkhusus;
120.	\$this->db->query("insert into perkalian2 values ('', '\$id', '1',
121.	'\$bobotakhirumum')");
122.	\$this->db->query("insert into perkalian2 values ('', '\$id', '2',
123.	'\$bobotakhirkhusus')");
124.	\$this->db->query("insert into rangking values ('', '\$id', '\$total')");
125.	}
126.	}

Tabel 4.7 Fungsi Menampilkan Rekomendasi *Homestay* Terbaik

1.	function getrangking() {
2.	return \$this->db->query("SELECT * FROM data_homestay dh
3.	join kriteria_homestay kh on dh.id_homestay=kh.id_homestay
4.	join status s on dh.id_status=s.id_status join rangking r on
5.	dh.id_homestay = r.id_homestay where dh.id_status='6'
6.	GROUP BY dh.id_homestay order by score desc limit 6")-
7.	>result_array();
	}

4.4.13. Kode Program Lihat Detail Perhitungan

Kode program lihat detail perhitungan terletak pada kelas *view* *w_detail* hitung, *controller* *halaman_user*, dan *model* *m_smarter*. Penulisan kode program dapat dilihat pada lampiran D.

4.5. Pengujian Sistem

Testing atau pengujian sistem dilakukan dengan dua cara, yaitu dengan *white-box testing* dan *black-box testing*. *White Box Testing* merupakan cara pengujian dengan melihat modul yang telah dibuat dengan program yang ada. Sedangkan *Black*

Box Testing pengujian program pada aplikasi langsung tanpa melihat desain dan kode program.

4.5.1. White Box Testing

Tahapan pengujian *white box* yaitu *listing program*, *flowchart* (diagram alir), dan *test case*. Fungsi atau *method* yang akan diuji pada perhitungan metode SMARTER sistem informasi rekomendasi *homestay* ini terletak pada *method* hitung di *model* *m_smarter*, dan untuk menyimpan bobot dari prioritas yang dipilih terletak pada *method* *simpanprioritassubkriteria* serta *method* *simpanprioritassubsubkriteria* di *model* *m_form_carihometay*. *Listing program* pada *method* hitung dapat dilihat pada Gambar 4.7 *Listing Program Method Hitung*. *Flowchart* pada *method* hitung dapat dilihat pada Gambar 4.8 *Diagram Alir Method Hitung*. *Test Case* pada *method* hitung dapat dilihat pada Tabel 4.8 *Test Case Method Hitung*. *Listing program* pada *method* *simpanprioritassubsubkriteria* dapat dilihat pada Gambar 4.9 *Listing Program Method Simpanprioritassubkriteria*. *Flowchart* pada *method* *simpanprioritassubkriteria* dapat dilihat pada Gambar 4.10 *Digram Alir Method Simpanprioritassubkriteria*. *Test Case* pada *method* *simpanprioritassubkriteria* dapat dilihat pada Tabel 4.9 *Test Case Method Simpanprioritassubkriteria*. *Listing program* pada *method* *simpanprioritassubsubkriteria* dapat dilihat pada Gambar 4.11 *Listing Program Method Simpanprioritassubkriteria*. *Flowchart* pada *method* *simpanprioritassubkriteria* dapat dilihat pada Gambar 4.12 *Digram Alir Method Simpanprioritassubkriteria*. *Test Case* pada *method* *simpanprioritassubkriteria* dapat dilihat pada Tabel 4.10 *Test Case Method Simpanprioritassubkriteria*.

a. *Test Case* Metode SMARTER

```

4 public function hitung(){
5     $this->db->query("truncate ranking");
6     $this->db->query("truncate perkalian");
7     $this->db->query("truncate perkalian2");
8     $query1 = $this->db->query("select count(*) as jumlah from data_homestay ");
9     foreach ($query1->result_array() as $s) {
10         $num = $s['jumlah'];
11     }
12     $subumum = ["sub01","sub02","sub03","sub04"];
13     $subkhusus = ["sub05","sub06","sub07"];
14     $sub1 = array();
15     $sub2 = array();
16     $queryu = $this->db->query("select * from data_kriteria where nama_kriteria = 'Umum'");
17     foreach ($queryu->result_array() as $v) {
18         $bobotumum = $v['bobot'];
19     }
20     $queryk = $this->db->query("select * from data_kriteria where nama_kriteria = 'Khusus'");
21     foreach ($queryk->result_array() as $va) {
22         $bobotkhusus = $va['bobot'];
23     }
24     for ($i=0; $i < $num; $i++) {
25         //mengambil data home stay disesuaikan dengan prioritas user
26         $query2 = $this->db->query("select * from data_homestay d join kriteria_homestay k
27             on d.id_homestay = k.id_homestay limit $i, 1");
28         foreach ($query2->result_array() as $value) {
29             $id = $value['id_homestay'];
30             $sub1[0] = 0;
31             if ($value['harga'] == 150000) {
32                 $sub1[0] = "Rp 150.000";
33             }else if ($value['harga'] == 200000) {
34                 $sub1[0] = "Rp 200.000";
35             }else if ($value['harga'] > 200000) {
36                 $sub1[0] = "> Rp 200.000";
37             }
38             $sub1[1] = 0;
39             if ($value['jarak_ijen'] < 10) {
40                 $sub1[1] = "< 10 km";
41             }else if ($value['jarak_ijen'] >= 10 && $value['jarak_ijen'] <= 20) {
42                 $sub1[1] = "10 km - 20 km";
43             }else if ($value['jarak_ijen'] > 20) {
44                 $sub1[1] = "> 20 km";
45             }
46             $sub1[2] = 0;
47             if ($value['jumlah_kamar'] == 1) {
48                 $sub1[2] = "1 kamar";
49             }else if ($value['jumlah_kamar'] == 2) {
50                 $sub1[2] = "2 kamar";
51             }else if ($value['jumlah_kamar'] >= 3) {
52                 $sub1[2] = ">= 3 kamar";
53             }
54             $sub1[3] = 0;
55             if ($value['jarak_saranauus'] < 3) {
56                 $sub1[3] = "< 3 km";
57             }else if ($value['jarak_saranauus'] >= 3 && $value['jarak_saranauus'] <= 7) {
58                 $sub1[3] = "3 km - 7 km";
59             }else if ($value['jarak_saranauus'] >7) {
60                 $sub1[3] = "> 7 km";
61             }
62             $sub2[0] = $value['wifi'];
63             $sub2[1] = $value['transportasi'];
64             $sub2[2] = $value['katering'];
65         }

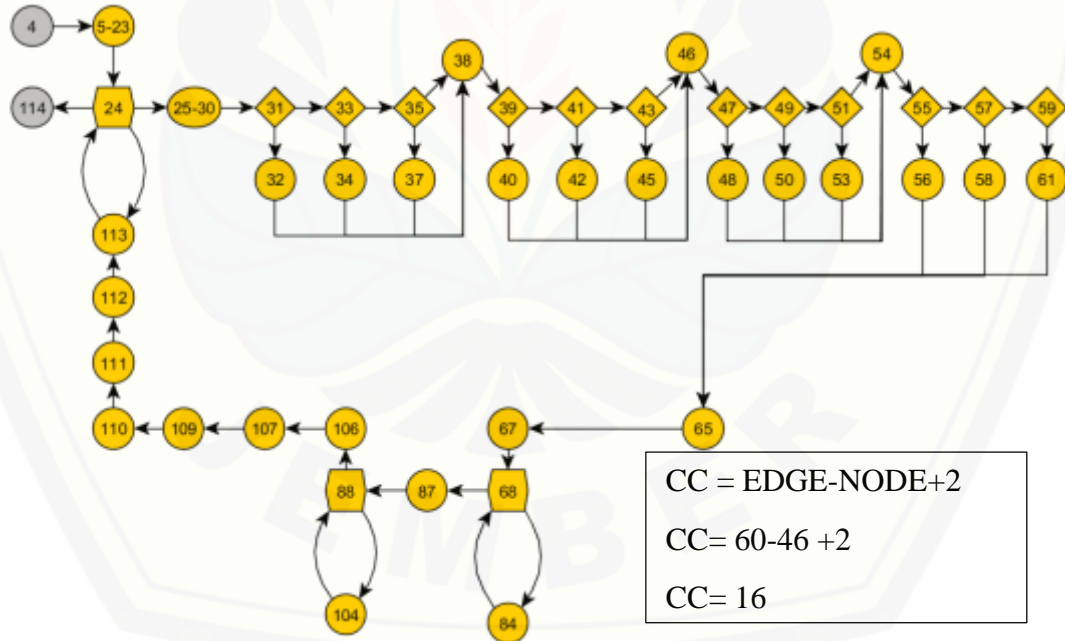
```

```

66 //menghitung step 2 prioritas umum
67 $jumlahumum = 0; $bobotsubu = 0; $bobotsubsubu=0;
68 for ($j=0; $j < 4; $j++) {
69     $query3 = $this->db->query("select a.id_subkriteria, bobot from subkriteria a join prioritas_subkriteria b
70     on(a.id_subkriteria = b.id_subkriteria) where a.id_subkriteria = '$subumum[$j]'");
71     foreach ($query3->result_array() as $val) {
72         $bobotsubu = $val['bobot'];
73         $idsubu = $val['id_subkriteria'];
74     }
75     $query4 = $this->db->query("select nama_subsub, bobot from subsubkriteria a join prioritas_subsubkriteria b
76     on (a.id_subsubkriteria = b.id_subsubkriteria) where a.id_subkriteria = '$subumum[$j]'
77     and nama_subsub = '$sub2[$j]'");
78     foreach ($query4->result_array() as $valu) {
79         $bobotsubsubu = $valu['bobot'];
80     }
81     $hasilu = $bobotsubu * $bobotsubsubu;
82     $jumlahumum = $jumlahumum + $hasilu;
83     $this->db->query("insert into perkalian values ('', '$id', '$idsubu', '$hasilu')");
84 }
85 //menghitung step 2 prioritas khusus
86 $jumlahkhusus = 0; $bobotsubk = 0; $bobotsubsubk=0;
87 for ($k=0; $k < 3; $k++) {
88     $query5 = $this->db->query("select a.id_subkriteria, bobot from subkriteria a join prioritas_subkriteria b
89     on (a.id_subkriteria = b.id_subkriteria) where a.id_subkriteria = '$subkhusus[$k]'");
90     foreach ($query5->result_array() as $values) {
91         $bobotsubk = $values['bobot'];
92         $idsubk = $values['id_subkriteria'];
93     }
94     $query6 = $this->db->query("select nama_subsub, bobot from subsubkriteria a join prioritas_subsubkriteria b
95     on (a.id_subsubkriteria = b.id_subsubkriteria) where a.id_subkriteria = '$subkhusus[$k]'
96     and nama_subsub = '$sub2[$k]'");
97     foreach ($query6->result_array() as $valued) {
98         $bobotsubsubk = $valued['bobot'];
99     }
100     $hasilk = $bobotsubk * $bobotsubsubk;
101     $this->db->query("insert into perkalian values ('', '$id', '$idsubk', '$hasilk')");
102     $jumlahkhusus = $jumlahkhusus + $hasilk;
103 }
104 //menghitung bobot akhir umum dan khusus
105 $bobotakhirumum = $jumlahumum * $bobotumum;
106 $bobotakhirkhusus = $jumlahkhusus * $bobotkhusus;
107 //menghitung total
108 $total = $bobotakhirumum + $bobotakhirkhusus;
109 $this->db->query("insert into perkalian2 values ('', '$id', '1', '$bobotakhirumum')");
110 $this->db->query("insert into perkalian2 values ('', '$id', '2', '$bobotakhirkhusus')");
111 $this->db->query("insert into ranking values ('', '$id', '$total')");
112 }
113 }
114 }

```

Gambar 4.6 Listing Program Method Hitung



Gambar 4.7 Diagram Alir Method Hitung

Tabel 4.8 *Test Case Method* Hitung

Test case method hitung()	
Jalur 1	
<i>Test Case</i>	Menghitung jumlah <i>homestay</i>
Target yang diharapkan	Menghasilkan jumlah <i>homestay</i>
Hasil pengujian	Benar
Path/Jalur	4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20- 21-22-23-24-25-30-32-34-35-37-38-40-42-45-46- 48-50-53-54-56-58-61-65-67-84-87-104-106-107- 109-110-111-112-113-114
Jalur 2	
<i>Test Case</i>	Mengambil nilai bobot pada kriteria fasilitas utama
Target yang diharapkan	Menghasilkan nilai bobot pada kriteria fasilitas utama
Hasil pengujian	Benar
Path/Jalur	4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20- 21-22-23-24-25-30-32-34-35-37-38-40-42-45-46- 48-50-53-54-56-58-61-65-67-84-87-104-106-107- 109-110-111-112-113-114
Jalur 3	
<i>Test Case</i>	Mengambil nilai bobot pada kriteria fasilitas tambahan
Target yang diharapkan	Menghasilkan nilai bobot pada kriteria fasilitas tambahan
Hasil pengujian	Benar
Path/Jalur	4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20- 21-22-23-24-25-30-32-34-35-37-38-40-42-45-46- 48-50-53-54-56-58-61-65-67-84-87-104-106-107- 109-110-111-112-113-114
Jalur 4	
<i>Test Case</i>	Mengambil nilai harga 200.000 pada setiap <i>homestay</i> berdasarkan prioritas yang dipilih <i>user</i>
Target yang diharapkan	Menampilkan nilai harga setiap <i>homestay</i> yang bernilai 200.000
Hasil pengujian	Benar
Path/Jalur	4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-

21-22-23-24-25-30-32-34-35-37-38-40-42-45-46-
48-50-53-54-56-58-61-65-67-84-87-104-106-107-
109-110-111-112-113-114

Jalur 5

<i>Test Case</i>	Mengambil nilai jarak ke ijen <10km pada setiap <i>homestay</i> berdasarkan prioritas yang dipilih <i>user</i>
Target yang diharapkan	Menampilkan nilai jarak ke ijen setiap <i>homestay</i> yang bernilai <10km
Hasil pengujian	Benar
Path/Jalur	4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20- 21-22-23-24-25-30-32-34-35-37-38-40-42-45-46- 48-50-53-54-56-58-61-65-67-84-87-104-106-107- 109-110-111-112-113-114

Jalur 6

<i>Test Case</i>	Mengambil nilai jumlah kamar 2 kamar pada setiap <i>homestay</i> berdasarkan prioritas yang dipilih <i>user</i>
Target yang diharapkan	Menampilkan nilai kamar setiap <i>homestay</i> yang bernilai 2 kamar
Hasil pengujian	Benar
Path/Jalur	4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20- 21-22-23-24-25-30-32-34-35-37-38-40-42-45-46- 48-50-53-54-56-58-61-65-67-84-87-104-106-107- 109-110-111-112-113-114

Jalur 7

<i>Test Case</i>	Mengambil nilai jarak ke sarana umum 3km-7km pada setiap <i>homestay</i> berdasarkan prioritas yang dipilih <i>user</i>
Target yang diharapkan	Menampilkan nilai jarak ke sarana umum setiap <i>homestay</i> yang bernilai 3km-7km
Hasil pengujian	Benar
Path/Jalur	4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20- 21-22-23-24-25-30-32-34-35-37-38-40-42-45-46- 48-50-53-54-56-58-61-65-67-84-87-104-106-107- 109-110-111-112-113-114

Jalur 8

<i>Test Case</i>	Mengambil nilai <i>wifi</i> dengan nilai Ada pada setiap <i>homestay</i> berdasarkan prioritas yang dipilih <i>user</i>
Target yang diharapkan	Menampilkan nilai <i>wifi</i> setiap <i>homestay</i> yang bernilai Ada
Hasil pengujian	Benar
Path/Jalur	4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-22-23-24-25-30-32-34-35-37-38-40-42-45-46-48-50-53-54-56-58-61-65-67-84-87-104-106-107-109-110-111-112-113-114

Jalur 9

<i>Test Case</i>	Mengambil nilai transportasi dengan nilai Ada pada setiap <i>homestay</i> berdasarkan prioritas yang dipilih <i>user</i>
Target yang diharapkan	Menampilkan nilai transportasi setiap <i>homestay</i> yang bernilai Ada
Hasil pengujian	Benar
Path/Jalur	4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-22-23-24-25-30-32-34-35-37-38-40-42-45-46-48-50-53-54-56-58-61-65-67-84-87-104-106-107-109-110-111-112-113-114

Jalur 10

<i>Test Case</i>	Mengambil nilai catering dengan nilai Ada pada setiap <i>homestay</i> berdasarkan prioritas yang dipilih <i>user</i>
Target yang diharapkan	Menampilkan nilai catering setiap <i>homestay</i> yang bernilai Ada
Hasil pengujian	Benar
Path/Jalur	4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-22-23-24-25-30-32-34-35-37-38-40-42-45-46-48-50-53-54-56-58-61-65-67-84-87-104-106-107-109-110-111-112-113-114

Jalur 11

<i>Test Case</i>	Mengambil nilai bobot sub kriteria fasilitas utama
Target yang diharapkan	Menghasilkan nilai bobot sub kriteria fasilitas

	utama
Hasil pengujian	Benar
Path/Jalur	4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20- 21-22-23-24-25-30-32-34-35-37-38-40-42-45-46- 48-50-53-54-56-58-61-65-67-84-87-104-106-107- 109-110-111-112-113-114
Jalur 12	
<i>Test Case</i>	Mengambil nilai bobot sub-sub kriteria fasilitas utama
Target yang diharapkan	Menghasilkan nilai bobot sub-sub kriteria fasilitas utama
Hasil pengujian	Benar
Path/Jalur	4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20- 21-22-23-24-25-30-32-34-35-37-38-40-42-45-46- 48-50-53-54-56-58-61-65-67-84-87-104-106-107- 109-110-111-112-113-114
Jalur 13	
<i>Test Case</i>	Menghitung jumlah dari perkalian sub kriteria dan sub-sub kriteria fasilitas utama
Target yang diharapkan	Menghasilkan jumlah sub kriteria dan sub-sub kriteria fasilitas utama
Hasil pengujian	Benar
Path/Jalur	4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20- 21-22-23-24-25-30-32-34-35-37-38-40-42-45-46- 48-50-53-54-56-58-61-65-67-84-87-104-106-107- 109-110-111-112-113-114
Jalur 14	
<i>Test Case</i>	Mengambil nilai bobot sub kriteria fasilitas tambahan
Target yang diharapkan	Menghasilkan nilai bobot sub kriteria fasilitas tambahan
Hasil pengujian	Benar
Path/Jalur	4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20- 21-22-23-24-25-30-32-34-35-37-38-40-42-45-46- 48-50-53-54-56-58-61-65-67-84-87-104-106-107-

109-110-111-112-113-114	
Jalur 15	
<i>Test Case</i>	Mengambil nilai bobot sub-sub kriteria fasilitas tambahan
Target yang diharapkan	Menghasilkan nilai bobot sub-sub kriteria fasilitas tambahan
Hasil pengujian	Benar
Path/Jalur	4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-22-23-24-25-30-32-34-35-37-38-40-42-45-46-48-50-53-54-56-58-61-65-67-84-87-104-106-107-109-110-111-112-113-114
Jalur 16	
<i>Test Case</i>	Menghitung jumlah dari perkalian sub kriteria dan sub-sub kriteria fasilitas tambahan
Target yang diharapkan	Menghasilkan jumlah sub kriteria dan sub-sub kriteria fasilitas utama
Hasil pengujian	Benar
Path/Jalur	4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-22-23-24-25-30-32-34-35-37-38-40-42-45-46-48-50-53-54-56-58-61-65-67-84-87-104-106-107-109-110-111-112-113-114

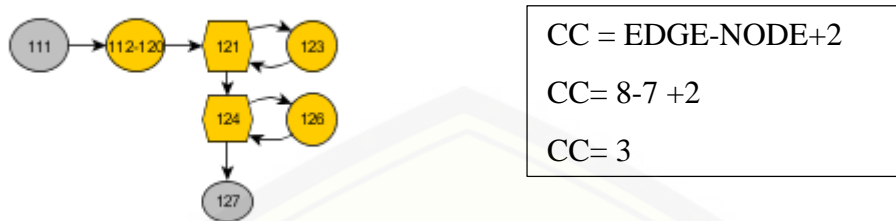
b. *Test Case* Simpan Bobot Prioritas

```

111 function simpanprioritassubkriteria($datau, $datak){
112     $roc2 = [0.75, 0.25];
113     $roc3 = [0.611, 0.278, 0.111];
114     $roc4 = [0.521, 0.271, 0.146, 0.063];
115     $roc5 = [0.457, 0.257, 0.157, 0.09, 0.04];
116     $roc6 = [0.408, 0.242, 0.158, 0.103, 0.061, 0.028];
117     $roc7 = [0.37, 0.228, 0.156, 0.109, 0.073, 0.044, 0.028];
118     $roc8 = [0.34, 0.215, 0.152, 0.111, 0.079, 0.054, 0.034, 0.016];
119     $roc9 = [0.314, 0.203, 0.148, 0.111, 0.083, 0.061, 0.042, 0.026, 0.012];
120     $this->db->query("truncate prioritas_subkriteria");
121     for ($i=0; $i < 4 ; $i++) {
122         $this->db->query("insert into prioritas_subkriteria values ('', '$datau[$i]', '$roc4[$i]')");
123     }
124     for ($i=0; $i < 3 ; $i++) {
125         $this->db->query("insert into prioritas_subkriteria values ('', '$datau[$i]', '$roc3[$i]')");
126     }
127 }

```

Gambar 4.8 Listing Program Method Simpanprioritassubkriteria



Gambar 4.9 Diagram Alir *Method* Simpanprioritassubkriteria

Tabel 4.9 *Test Case Method* Simpanprioritassubkriteria

Test case method simpanprioritassubkriteria()

Jalur 1	
<i>Test Case</i>	Pengecekan bobot ROC untuk sub kriteria
Target yang diharapkan	Berhasil menampung nilai bobot ROC untuk sub kriteria. dengan urutan harga prioritas1, jumlah kamar prioritas 2, jarak ke sarana umum prioritas 3, jarak ijen prioritas 4, katering prioritas1, wifi prioritas 2,transportasi 3
Hasil pengujian	Benar
Path/Jalur	111-112-113-114-115-116-117-118-119-120-122-123-126-127
Jalur 2	
<i>Test Case</i>	Menyimpan nilai bobot ROC untuk kriteria fasilitas utama dengan urutan harga prioritas1, jumlah kamar prioritas 2, jarak ke sarana umum prioritas 3, jarak ijen prioritas 4.
Target yang diharapkan	Berhasil menyimpan nilai bobot ROC untuk kriteria fasilitas utama dengan urutan harga prioritas1, jumlah kamar prioritas 2, jarak ke sarana umum prioritas 3, jarak ijen prioritas 4.
Hasil pengujian	Benar

Path/Jalur 111-112-113-114-115-116-117-118-119-120-122-123-126-127

Jalur 3

Test Case Menyimpan nilai bobot ROC untuk kriteria fasilitas tambahan dengan urutan katering prioritas1, wifi prioritas 2, transportasi 3

Target yang diharapkan Berhasil menyimpan nilai bobot ROC untuk kriteria fasilitas tambahan

Hasil pengujian Benar

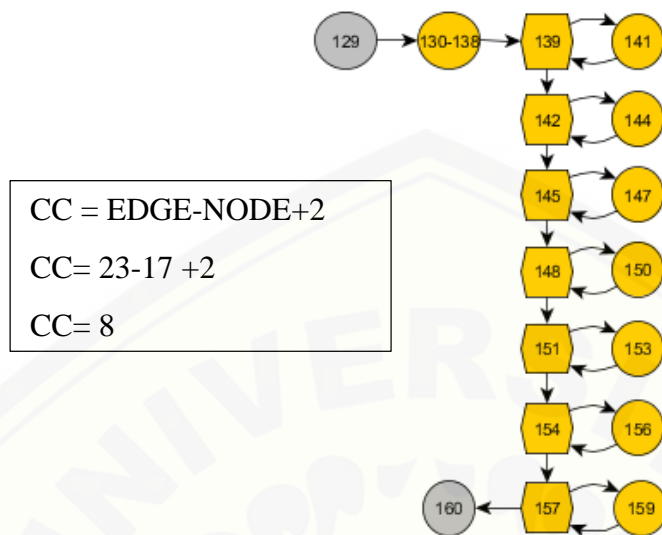
Path/Jalur 111-112-113-114-115-116-117-118-119-120-122-123-126-127

```

129 function simpanprioritassubsubkriteria($harga, $jarakijen, $jaraksarana, $jumlahkamar, $wifi, $transportasi, $katering){
130     $roc2 = [0.75, 0.25];
131     $roc3 = [0.611, 0.278, 0.111];
132     $roc4 = [0.521, 0.271, 0.146, 0.063];
133     $roc5 = [0.457, 0.257, 0.157, 0.09, 0.04];
134     $roc6 = [0.408, 0.242, 0.158, 0.103, 0.061, 0.028];
135     $roc7 = [0.37, 0.228, 0.156, 0.109, 0.073, 0.044, 0.028];
136     $roc8 = [0.34, 0.215, 0.152, 0.111, 0.079, 0.054, 0.034, 0.016];
137     $roc9 = [0.314, 0.203, 0.148, 0.111, 0.083, 0.061, 0.042, 0.026, 0.012];
138     $this->db->query("truncate prioritas_subsubkriteria");
139     for ($i=0; $i < 3 ; $i++) {
140         $this->db->query("insert into prioritas_subsubkriteria values ('', '$harga[$i]', '$roc3[$i]')");
141     }
142     for ($i=0; $i < 3 ; $i++) {
143         $this->db->query("insert into prioritas_subsubkriteria values ('', '$jarakijen[$i]', '$roc3[$i]')");
144     }
145     for ($i=0; $i < 3 ; $i++) {
146         $this->db->query("insert into prioritas_subsubkriteria values ('', '$jaraksarana[$i]', '$roc3[$i]')");
147     }
148     for ($i=0; $i < 3 ; $i++) {
149         $this->db->query("insert into prioritas_subsubkriteria values ('', '$jumlahkamar[$i]', '$roc3[$i]')");
150     }
151     for ($i=0; $i < 2 ; $i++) {
152         $this->db->query("insert into prioritas_subsubkriteria values ('', '$wifi[$i]', '$roc2[$i]')");
153     }
154     for ($i=0; $i < 2 ; $i++) {
155         $this->db->query("insert into prioritas_subsubkriteria values ('', '$transportasi[$i]', '$roc2[$i]')");
156     }
157     for ($i=0; $i < 2 ; $i++) {
158         $this->db->query("insert into prioritas_subsubkriteria values ('', '$katering[$i]', '$roc2[$i]')");
159     }
160 }

```

Gambar 4.10 Listing Program Method Simpanprioritassubsubkriteria



Gambar 4.11 Diagram Alir *Method* Simpanprioritassubsubkriteria

Tabel 4.10 *Test Case Method* Simpanprioritassubsubkriteria

Test case method simpanprioritassubsubkriteria()	
Jalur 1	
<i>Test Case</i>	Pengecekan bobot ROC untuk sub-sub kriteria
Target yang diharapkan	Berhasil menampung nilai bobot ROC untuk sub-sub kriteria harga 200.000 prioritas 1, jumlah kamar 2 prioritas 1, jarak ke sarana umum 3km-7km prioritas 1, jarak ijen <10km prioritas 1, katering Ada prioritas1, wifi prioritas Ada 1, transportasi Ada prioritas 1.
Hasil pengujian	Benar
Path/Jalur	129-130-131-132-133-134-135-136-137-138-141-144-147-150-153-156-159-160
Jalur 2	
<i>Test Case</i>	Menyimpan nilai bobot ROC sub kriteria harga untuk kriteria fasilitas utama
Target yang diharapkan	Berhasil menyimpan nilai bobot ROC sub kriteria harga untuk kriteria fasilitas utama
Hasil pengujian	Benar
Path/Jalur	129-130-131-132-133-134-135-136-137-138-141-

144-147-150-153-156-159-160

Jalur 3

<i>Test Case</i>	Menyimpan nilai bobot ROC sub kriteria jarak ke ijen untuk kriteria fasilitas utama
Target yang diharapkan	Berhasil menyimpan nilai bobot ROC sub kriteria jarak ke ijen untuk kriteria fasilitas utama
Hasil pengujian	Benar
Path/Jalur	129-130-131-132-133-134-135-136-137-138-141-144-147-150-153-156-159-160

Jalur 4

<i>Test Case</i>	Menyimpan nilai bobot ROC sub kriteria jumlah kamar untuk kriteria fasilitas utama
Target yang diharapkan	Berhasil menyimpan nilai bobot ROC sub kriteria jumlah kamar untuk kriteria fasilitas utama
Hasil pengujian	Benar
Path/Jalur	129-130-131-132-133-134-135-136-137-138-141-144-147-150-153-156-159-160

Jalur 5

<i>Test Case</i>	Menyimpan nilai bobot ROC sub jarak ke sarana umum untuk kriteria fasilitas utama
Target yang diharapkan	Berhasil menyimpan nilai bobot ROC sub kriteria jarak ke sarana umum untuk kriteria fasilitas utama
Hasil pengujian	Benar
Path/Jalur	129-130-131-132-133-134-135-136-137-138-141-144-147-150-153-156-159-160

Jalur 6

<i>Test Case</i>	Menyimpan nilai bobot ROC sub kriteria wifi untuk kriteria fasilitas tambahan
Target yang diharapkan	Berhasil menyimpan nilai bobot ROC sub kriteria wifi untuk kriteria fasilitas tambahan
Hasil pengujian	Benar

Path/Jalur	129-130-131-132-133-134-135-136-137-138-141-144-147-150-153-156-159-160
------------	---

Jalur 7

<i>Test Case</i>	Menyimpan nilai bobot ROC sub kriteria transportasi untuk kriteria fasilitas tambahan
Target yang diharapkan	Berhasil menyimpan nilai bobot ROC sub kriteria transportasi untuk kriteria fasilitas tambahan
Hasil pengujian	Benar
Path/Jalur	129-130-131-132-133-134-135-136-137-138-141-144-147-150-153-156-159-160

Jalur 8

<i>Test Case</i>	Menyimpan nilai bobot ROC sub kriteria katering untuk kriteria fasilitas tambahan
Target yang diharapkan	Berhasil menyimpan nilai bobot ROC sub kriteria katering untuk kriteria fasilitas tambahan
Hasil pengujian	Benar
Path/Jalur	129-130-131-132-133-134-135-136-137-138-141-144-147-150-153-156-159-160

4.5.2. Black Box Testing

Pengujian *black box* sistem informasi rekomendasi *homestay* pada fitur mencari *homestay* dan fitur lihat detail perhitungan dapat dilihat pada Tabel 4.11 *Black Box Testing* .

Tabel 4.11 *Black Box Testing*

No.	Fitur	Aksi	Hasil	Ket
1	Mencari <i>Homestay</i>	Memilih menu cari <i>homestay</i> .	Menampilkan form prioritas sub kriteria	Berhasil

		Memilih tombol <i>default</i>	Mengisi form prioritas sub kriteria	Berhasil
		Memilih tombol <i>next</i>	Menampilkan form prioritas subs-sub kriteria	Berhasil
		Memilih tombol <i>next</i> namun data tidak terisi lengkap	Menampilkan message “please fill out thhis field”	Berhasil
		Memilih tombol <i>default</i>	Mengisi form prioritas sub-sub kriteria	Berhasil
		Memilih tombol cari	Menampilkan rekomendasi <i>homestay</i> dengan 6 nilai tertinggi	Berhasil
		Memilih tombol cari namun data tidak terisi lengkap	Menampilkan message “please fill out thhis field”	Berhasil
		Memilih tombol kembali	Menampilkan Halaman Utama	Berhasil
2.	Lihat Detail Perhitungan	Memilih tombol lihat detail	Menampilkan detail perhitungan dari <i>homestay</i> yang dipilih	Berhasil
		Memilih tombol kembali	Menampilkan Halaman Utama	Berhasil

BAB 6. PENUTUP

Bab ini membahas tentang kesimpulan dan saran dari peneliti mengenai penelitian yang telah dilakukan. Kesimpulan dan saran tersebut diharapkan dapat digunakan sebagai acuan dalam penelitian selanjutnya untuk menjadikan sistem lebih baik dan benar.

6.1. Kesimpulan

1. Metode *Simple Multi-Attribute Rating Technique Exploiting Ranks* (SMARTER) digunakan dalam pencarian *homestay* berdasarkan kriteria-kriteria yang telah ditentukan dengan menghasilkan rekomendasi *homestay* terbaik sesuai dengan keinginan *user* untuk memudahkan dalam pengambilan keputusan.
2. Sistem Informasi Rekomendasi *Homestay* Menggunakan Metode *Simple Multi-Attribute Rating Technique Exploiting Ranks* (SMARTER) berbasis *website* yang memiliki 2 hak akses yaitu Admin dan *User* (calon wisatawan). Sistem ini dikembangkan menggunakan model *waterfall* yang pada penerapannya memiliki kelemahan yaitu terjadi proses pengulangan pada tahapan selanjutnya sehingga menyebabkan proses pengembangan kurang mengikuti urutan seperti pada teori.
3. Pemberian prioritas merupakan pembobotan kriteria, sub kriteria, dan sub-sub kriteria yang akan dijadikan langkah awal perhitungan metode *Simple Multi-Attribute Rating Technique Exploiting Ranks* (SMARTER). Pembobotannya menggunakan pembobotan *Ranks Of Centroid* (ROC) yang merupakan pembobotan berdasarkan tingkat kepentingan. Pada kriteria memiliki pembobotan 2 kriteria dari prioritas 1 sampai 2 dengan nilai 0,75 dan 0,25. Pada sub kriteria fasilitas utama memiliki pembobotan 4 kriteria dari prioritas 1 sampai 4 dengan nilai 0.521, 0.271, 0.146, dan 0.063. Pada sub kriteria fasilitas tambahan memiliki pembobotan 3 kriteria dari prioritas 1 sampai 3 dengan nilai 0.611, 0.278, dan 0.111. Pada sub-sub kriteria fasilitas utama masing-masing

memiliki pembobotan 3 kriteria dari prioritas 1 sampai 3 dengan nilai 0.611, 0.278, dan 0.111. Pada sub-sub kriteria fasilitas tambahan masing-masing memiliki pembobotan 2 kriteria dari prioritas 1 sampai 2 dengan nilai 0,75 dan 0,25.

4. Sistem Informasi Rekomendasi *Homestay* Menggunakan Metode *Simple Multi-Attribute Rating Technique Exploiting Ranks* (SMARTER) menunjukkan hasil rekomendasi *homestay* terbaik untuk setiap *user* dengan hasil yang berbeda-beda. Hal ini dikarenakan pemberian prioritas pada setiap kriteria, sub kriteria, dan sub-sub kriteria untuk setiap *user* berbeda-beda disesuaikan dengan keinginan *user*.

6.2. Saran

1. Pembobotan kriteria, sub kriteria dan sub-subkriteria yang dinamis.
2. Diharapkan adanya pengembangan sistem dengan bahasa yang bermacam-macam karena pengguna sistem ini berasal dari wisatawan mancanegara.
3. Pengembangan sistem ini dapat menambahkan fitur pemilihan paket wisata pada kawasan kawah ijen.
4. *Payment* atau pembayaran pada sistem *booking* dapat dilakukan secara online.

DAFTAR PUSTAKA

- Kadir, Abdul. (2014). *Pengenalan Sistem Informasi*. (Edisi Revisi). Yogyakarta : ANDI
- Indah, I. N. (2013). *Pembuatan Sistem Informasi Penjualan Pada Toko Sehat Jaya Elektronik Pacitan*.
- Turban. Aronso. and Liang. (2005). *Decision Support Systems and Intelligent Systems*. Yogyakarta : ANDI.
- Alfita, R. 2012. *Decision Support System Of Reserve Building Cultural Revitalization Determination Using Simple Multiattribute Rating Technique Exploiting Ranks Method*
- Haryanti, D., Nasution H., Sukamto A.S. (2016). *Sistem Pendukung Keputusan Seleksi Penerimaan Mahasiswa Pengganti Beasiswa Penuh Bidikmisi Universitas Tanjungpura*
- Mishra, A. & Dubey, D. (2013). *A Comparative Study of Different Software Development Life Cycle Models in Different Scenarios*. *International Journal of Advance Research in Computer Science and Management Studies*, 1(5).
Diambil kembali dari <http://www.ijarcsms.com>
- Kusrini. (2007). *Konsep dan Aplikasi Sistem Pendukung Keputusan*. Yogyakarta : ANDI.
- Fauziah. (2015). *Sistem Pendukung Keputusan Penentuan Pemberian Raskin Untuk Masyarakat Menggunakan Metode Smart (Simple Multi Attribute Rating Technique)*. Skripsi pada Program Studi Sistem Informasi, Universitas Jember, Jember.
- Rahmah, A. (2013). *Sistem Pendukung Keputusan Seleksi Masuk Mahasiswa Menggunakan Metode SMARTER*. Skripsi pada Universitas Pendidikan Indonesia, Jakarta.

Okfalisa dan Gunawan, A. (2014). *Sistem Pendukung Keputusan Untuk Menentukan Produk Asuransi Jiwa Bagi Nasabah Menggunakan Metode Smarter*. *Jurnal Sains, Teknologi dan Industri*, 12(1).



LAMPIRAN A

A.1. Skenario *Login*

Nama <i>Usecase</i>	<i>Login</i>
Partisipan Aktor	Admin
Pre-Kondisi	Admin ingin melakukan <i>login</i>
Pra-Kondisi	Admin telah melakukan <i>login</i>
<i>Event flow</i>	Admin membuka website Admin memilih menu <i>login</i>
SKENARIO Normal “login”	

Aksi Aktor	Reaksi Sistem
1. Klik menu <i>login</i> .	
	2. Menampilkan halaman <i>login</i> , tombol <i>login</i> dan tombol kembali.
3. Memasukkan <i>username</i> dan <i>password</i>	
4. Klik tombol <i>login</i>	
	5. Menampilkan halaman dashboard

SKENARIO Alternatif “Username atau password tidak lengkap”	
Aksi Aktor	Reaksi Sistem
4a. Klik tombol <i>login</i>	
	5a. Menampilkan message “ <i>Username is required!</i> ” atau “ <i>Password is required!</i> ”

SKENARIO Alternatif**“Username atau password salah”****Aksi Aktor****Reaksi Sistem**4b. Klik tombol *login*5b. Menampilkan message “*Login salah*”**SKENARIO Alternatif****“Klik Tombol Kembali”****Aksi Aktor****Reaksi Sistem**4c. Klik tombol *back*

4c. Menampilkan halaman utama

A.2. Skenario Kelola Data *Homestay*

Nama <i>Usecase</i>	Kelola Data <i>Homestay</i>
Partisipan Aktor	Admin
Pre-Kondisi	Admin ingin mengelola data <i>homestay</i>
Pra-Kondisi	Admin telah menambah data <i>homestay</i>
	Admin telah merubah data <i>homestay</i>
	Admin telah melihat data <i>homestay</i>
	Admin telah menghapus data <i>homestay</i>
<i>Event flow</i>	Admin membuka website
	Admin memilih menu <i>login</i>
	Admin memilih menu <i>homestay</i>
SKENARIO Normal	
“Tambah Data <i>Homestay</i>”	
Aksi Aktor	Reaksi Sistem

-
1. Klik menu *homestay*.
-
2. Menampilkan halaman *homestay*, tabel *homestay*, tombol tambah *homestay*, tombol *detail*, tombol ubah, dan tombol hapus.
-
3. Klik tombol tambah *homestay*
-
4. Menampilkan modal form tambah data *homestay*, tombol simpan, dan tombol kembali
-
5. Memasukkan id *homestay*, nama *homestay*, pemilik, alamat *homestay*, nomor identitas, nomor telepon, foto *homestay*, harga, jumlah kamar, jarak ke ijen, jarak ke sarana umum, wifi, transportasi, katering.
-
6. Klik tombol simpan
-
7. Menampilkan pesan “Data Berhasil Ditambahkan” dan tabel *homestay* yang telah diperbaharui.

SKENARIO Normal

“Ubah *Homestay*”

Aksi Aktor	Reaksi Sistem
3. Klik tombol ubah	4. Menampilkan form ubah <i>homestay</i> , tombol simpan dan tombol kembali
5. Merubah nama <i>homestay</i> , pemilik, alamat <i>homestay</i> , nomor identitas,	

nomor telepon, foto *homestay*,
 harga, jumlah kamar, jarak ke ijen,
 jarak ke sarana umum, wifi,
 transportasi, katering.

6. Klik tombol simpan

7. Menampilkan pesan “Data Berhasil Diubah” dan tabel *homestay* yang telah diperbaharui.

SKENARIO Normal

“Lihat Detail *Homestay*”

Aksi Aktor	Reaksi Sistem
3. Klik tombol detail	4. Menampilkan halaman detail <i>homestay</i> yang dipilih berisi id <i>homestay</i> , nama <i>homestay</i> , pemilik, alamat <i>homestay</i> , nomor identitas, nomor telepon, foto <i>homestay</i> , harga, jumlah kamar, jarak ke ijen, jarak ke sarana umum, wifi, transportasi, katering. dan tombol kembali

SKENARIO Normal

“Hapus *Homestay*”

Aksi Aktor	Reaksi Sistem
3. Klik tombol hapus	4. Menampilkan modal hapus <i>homestay</i> dengan pernyataan “Anda yakin akan menghapus

	data Homestay?”, tombol ya dan tombol tidak
5. Memilih tombol ya	
	6. Menampilkan pesan “Data Berhasil Dihapus” dan tabel <i>homestay</i> yang telah diperbaharui.

**SKENARIO Alternatif
“Data tidak lengkap”**

Aksi Aktor	Reaksi Sistem
4a. Klik tombol simpan	
	5a. Menampilkan message “ <i>please fill out this field</i> ”

**SKENARIO Alternatif
“Klik tombol tidak”**

Aksi Aktor	Reaksi Sistem
5b. Klik tombol <i>tidak</i>	
	6b. Menampilkan tabel <i>homestay</i> .

**SKENARIO Alternatif
“Klik Tombol Kembali”**

Aksi Aktor	Reaksi Sistem
4c. Klik tombol kembali	
	5c. Menampilkan tabel <i>homestay</i>

A.3. Skenario Ubah Informasi

Nama <i>Usecase</i>	Ubah Informasi
Partisipan Aktor	Admin
Pre-Kondisi	Admin ingin merubah data informasi

Pra-Kondisi	Admin telah merubah data informasi
<i>Event flow</i>	Admin membuka website
	Admin memilih menu <i>login</i>
	Admin memilih menu informasi
SKENARIO Normal	
“Ubah Informasi”	
Aksi Aktor	Reaksi Sistem
1. Klik menu informasi	
	2. Menampilkan form ubah informasi dan tombol ubah
3. Merubah data informasi	
4. Klik tombol ubah	
	5. Menampilkan data informasi yang telah diperbaharui

A.4. Skenario Lihat Informasi

Nama <i>Usecase</i>	Lihat Informasi
Partisipan Aktor	<i>User</i>
Pre-Kondisi	<i>User</i> ingin melihat informasi
Pra-Kondisi	<i>User</i> telah melihat informasi
<i>Event flow</i>	<i>User</i> membuka website
	<i>User</i> memilih menu info paket wisata
SKENARIO Normal	
“Lihat Informasi”	
Aksi Aktor	Reaksi Sistem
1. Klik menu info paket wisata	
	2. Menampilkan informasi paket wisata

A.5. Skenario Ubah Admin

Nama <i>Usecase</i>	Ubah Admin
Partisipan Aktor	Admin
Pre-Kondisi	Admin ingin merubah data admin
Pra-Kondisi	Admin telah merubah data admin
<i>Event flow</i>	Admin membuka website Admin memilih menu <i>login</i> Admin memilih tombol ubah <i>privacy</i>
SKENARIO Normal “Ubah Admin”	
Aksi Aktor	Reaksi Sistem
1. Klik sub menu ubah <i>privacy</i> pada <i>dropdown</i> admin	2. Menampilkan form ubah admin, tombol simpan dan tombol kembali
3. Merubah nama, username, password.	
4. Klik tombol simpan	5. Menampilkan halaman dashboard
SKENARIO Alternatif “Data tidak lengkap”	
Aksi Aktor	Reaksi Sistem
4. Klik tombol simpan	5. Menampilkan message “ <i>please fill out this field</i> ”
SKENARIO Alternatif “Klik Tombol Kembali”	

Aksi Aktor	Reaksi Sistem
4. Klik tombol kembali	
	5. Menampilkan halaman dashboard

A.6. Skenario *Booking Homestay*

Nama <i>Usecase</i>	<i>Booking Homestay</i>
Partisipan Aktor	<i>User</i>
Pre-Kondisi	<i>User ingin melakukan booking</i>
Pra-Kondisi	<i>User telah melakukan booking</i>
<i>Event flow</i>	<i>User membuka website</i> <i>User memilih menu cari homestay</i> <i>User memilih tombol booking</i>
SKENARIO Normal “<i>Booking Homestay</i>”	

Aksi Aktor	Reaksi Sistem
1. Klik tombol <i>booking</i>	
	2. Menampilkan form booking, tombol send, dan tombol kembali
3. Memasukkan nomor <i>booking</i> , nama <i>homestay</i> , <i>harga</i> , tanggal check-in, lama inap, jumlah kamar yang dipesan, pemesan, alamat, nomor identitas, no telepon, email.	
4. Klik tombol <i>send</i>	
	5. Menampilkan pop up “Terima Kasih.. Homestay Berhasil di Booking” dan menampilkan halaman

utama

SKENARIO Alternatif
“Klik Tombol Kembali”

Aksi Aktor

Reaksi Sistem

4a. Klik tombol kembali

5a. Menampilkan halaman utama

SKENARIO Alternatif
“Stok kamar tidak tersedia”

Aksi Aktor

Reaksi Sistem

4b. Klik tombol *send*

5b. Menampilkan *message* “Kamar Yang
Anda Pesan Melebihi Stok Yang
Tersedia”

SKENARIO Alternatif
“Data tidak lengkap”

Aksi Aktor

Reaksi Sistem

4c. Klik tombol *send*

5c. Menampilkan *message* “*please fill
out this field*”

A.7. Skenario Ubah Data Pemesan

Nama <i>Usecase</i>	Ubah Data Pemesan
Partisipan Aktor	Admin
Pre-Kondisi	Admin ingin merubah data <i>pemesan</i>
Pra-Kondisi	Admin telah merubah data <i>pemesan</i>
<i>Event flow</i>	Admin membuka website Admin memilih menu <i>login</i>

Admin memilih menu booking	
SKENARIO Normal	
“Ubah <i>Booking</i> ”	
Aksi Aktor	Reaksi Sistem
1. Klik menu booking	
	2. Menampilkan tabel <i>booking</i> , tombol ubah, detail, hapus, dan konfirmasi
3. Klik tombol ubah	
	4. Menampilkan form ubah data pemesan, tombol simpan, dan tombol kembali
5. Merubah nama pemesan, alamat, nomor identitas, nomor telepon, email.	
6. Klik tombol simpan	
	7. Menampilkan tabel <i>booking</i> yang telah diperbaharui dan pesan “Data berhasil diubah”
SKENARIO Alternatif	
“Data tidak lengkap”	
Aksi Aktor	Reaksi Sistem
6a. Klik tombol simpan	
	7a. Menampilkan message “ <i>please fill out this field</i> ”
SKENARIO Alternatif	
“Klik Tombol Kembali”	
Aksi Aktor	Reaksi Sistem

6b. Klik tombol kembali

7b. Menampilkan halaman dashboard

A.8. Skenario Hapus Data Pemesan

Nama <i>Usecase</i>	Hapus Data Pemesan
Partisipan Aktor	Admin
Pre-Kondisi	Admin ingin menghapus data pemesanan
Pra-Kondisi	Admin telah menghapus data pemesanan
<i>Event flow</i>	Admin membuka website Admin memilih menu <i>login</i> Admin memilih menu booking
SKENARIO Normal “Hapus Data Pemesan”	
Aksi Aktor	Reaksi Sistem
1. Klik menu booking	2. Menampilkan tabel <i>booking</i> , tombol ubah, detail, hapus, dan konfirmasi
3. Klik tombol hapus	4. Menampilkan modal hapus <i>booking</i> dengan pernyataan “Anda yakin akan menghapus data <i>Booking?</i> ”, tombol ya dan tombol tidak
5. Memilih tombol ya	6. Menampilkan pesan “Data Berhasil Dihapus” dan tabel <i>booking</i> yang telah diperbaharui.

SKENARIO Alternatif	
“Klik Tombol Tidak”	
Aksi Aktor	Reaksi Sistem
6a. Klik tombol tidak	7a. Menampilkan tabel <i>booking</i>

A.9. Skenario Lihat Detail *Booking*

Nama <i>Usecase</i>	Lihat Detail <i>Booking</i>
Partisipan Aktor	Admin
Pre-Kondisi	Admin ingin melihat detail <i>booking</i>
Pra-Kondisi	Admin telah melihat detail <i>booking</i>
<i>Event flow</i>	Admin membuka website Admin memilih menu <i>login</i> Admin memilih menu <i>booking</i>

SKENARIO Normal	
“Lihat Detail <i>Booking</i>”	
Aksi Aktor	Reaksi Sistem
1. Klik menu <i>booking</i>	2. Menampilkan tabel <i>booking</i> , tombol ubah, detail, hapus, dan konfirmasi
3. Klik tombol detail	4. Menampilkan detail pemesanan (nama pemesan, alamat, nomor identitas, nomor telepon, email.) dan detail <i>booking</i> (nomor <i>booking</i> , nama homestay, harga,

tanggal check-in, lama inap,
jumlah kamar yang
dipesan,)ombol kembali

5. Klik tombol kembali

6. Menampilkan tabel *booking*

A.10. Skenario Konfirmasi *Booking*

Nama <i>Usecase</i>	Lihat Konfirmasi <i>Booking</i>
Partisipan Aktor	Admin
Pre-Kondisi	Admin ingin mengkonfirmasi <i>booking</i>
Pra-Kondisi	Admin telah mengkonfirmasi <i>booking</i>
<i>Event flow</i>	Admin membuka website Admin memilih menu <i>login</i> Admin memilih menu <i>booking</i>
SKENARIO Normal “Konfirmasi <i>Booking</i>”	
Aksi Aktor	Reaksi Sistem
1. Klik menu <i>booking</i>	2. Menampilkan tabel <i>booking</i> , tombol ubah, detail, hapus, dan konfirmasi
3. Klik tombol konfirmasi	4. Menampilkan modal hapus <i>booking</i> dengan pernyataan “Konfirmasi <i>Booking?</i> ”, tombol ya dan tombol tidak
5. Memilih tombol ya	6. Menampilkan pesan “Data Berhasil Dihapus” dan tabel <i>booking</i> yang

telah diperbaharui.

SKENARIO Alternatif

“Klik Tombol Tidak”

Aksi Aktor	Reaksi Sistem
5a. Klik tombol tidak	
	6a. Menampilkan tabel <i>booking</i>

A.11. Skenario Cek Detail Pemesanan

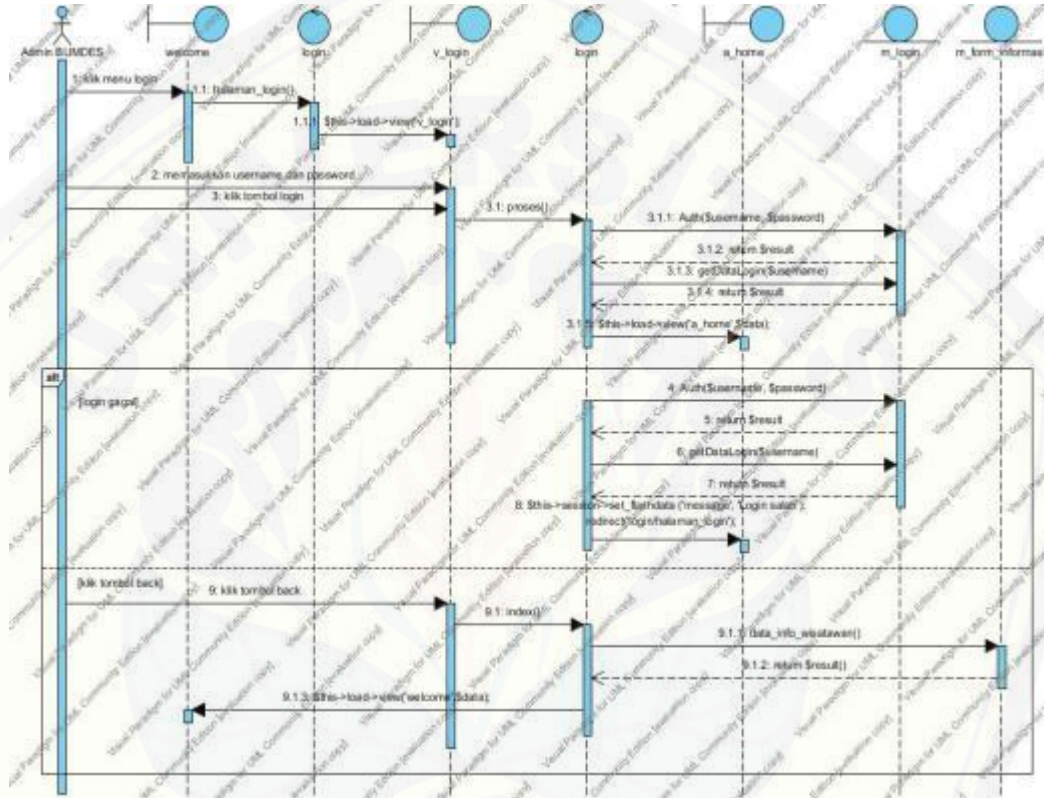
Nama <i>Usecase</i>	Cek Detail Pemesanan
Partisipan Aktor	<i>User</i>
Pre-Kondisi	<i>User</i> ingin melihat detail pemesanan
Pra-Kondisi	<i>User</i> telah melihat detail pemesanan
<i>Event flow</i>	<i>User</i> membuka website <i>User</i> memilih menu cek detail pemesanan
SKENARIO Normal	
“Cek Detail Pemesanan”	
Aksi Aktor	Reaksi Sistem
1. Klik menu cek detail pemesanan	
	2. Menampilkan form cek pemesanan
3. Memasukkan no. <i>booking</i>	
4. Klik tombol cek	
	5. Menampilkan detail pemesanan (nama pemesan, alamat, nomor identitas, nomor telepon, email.) dan detail <i>booking</i> (nomor <i>booking</i> , nama <i>homestay</i> , <i>harga</i> , tanggal check-in, lama inap, jumlah kamar)

A.12. Skenario Lihat Detail Perhitungan

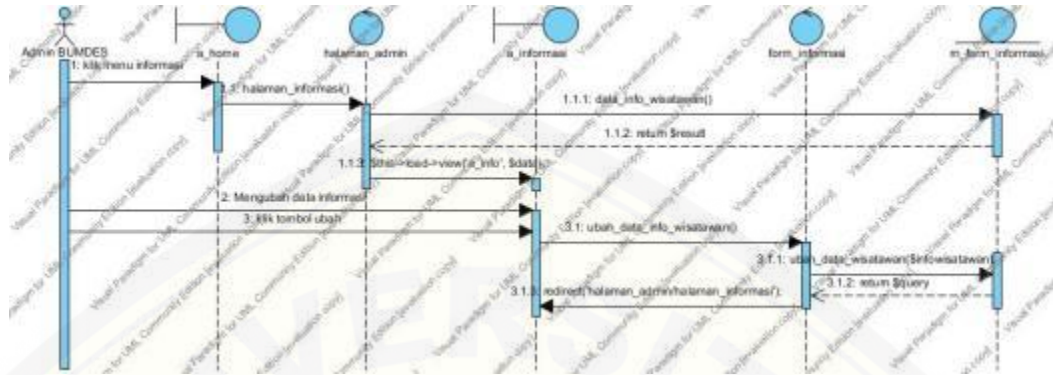
Nama <i>Usecase</i>	Lihat Detail Perhitungan
Partisipan Aktor	<i>User</i>
Pre-Kondisi	<i>User</i> ingin melihat detail perhitungan
Pra-Kondisi	<i>User</i> telah melihat detail perhitungan
<i>Event flow</i>	<p><i>User</i> membuka website</p> <p><i>User</i> memilih menu Cari Homestay</p> <p><i>User</i> mengisi form prioritas sub kriteria fasilitas utama dan fasilitas tambahan</p> <p><i>User</i> klik tombol next</p> <p><i>User</i> mengisi form prioritas sub-sub kriteria fasilitas utama dan fasilitas tambahan</p> <p><i>User</i> klik tombol cari</p> <p><i>User</i> klik tombol lihat detail pada salah satu <i>homestay</i></p>
SKENARIO Normal	
“Lihat Detail Perhitungan”	
Aksi Aktor	Reaksi Sistem
1. Klik tombol lihat detail pada salah satu <i>homestay</i>	2. Menampilkan detail perhitungan <i>homestay</i> yang dipilih
SKENARIO Normal	
“Klik Tombol Kembali”	
Aksi Aktor	Reaksi Sistem
1. Klik tombol kembali	2. Menampilkan halaman welcome

LAMPIRAN B

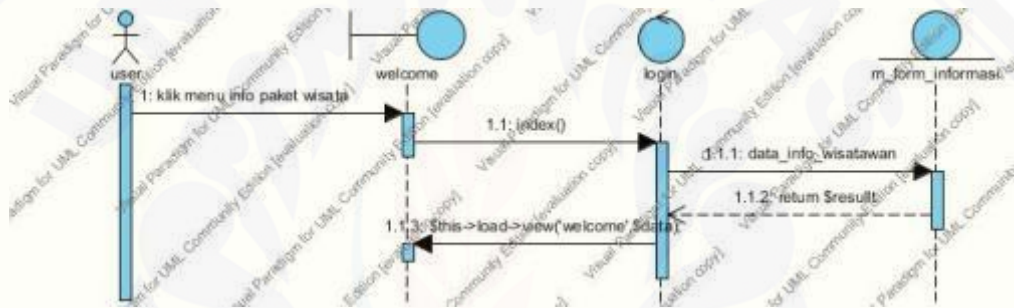
B.1. Sequence Diagram *Login*



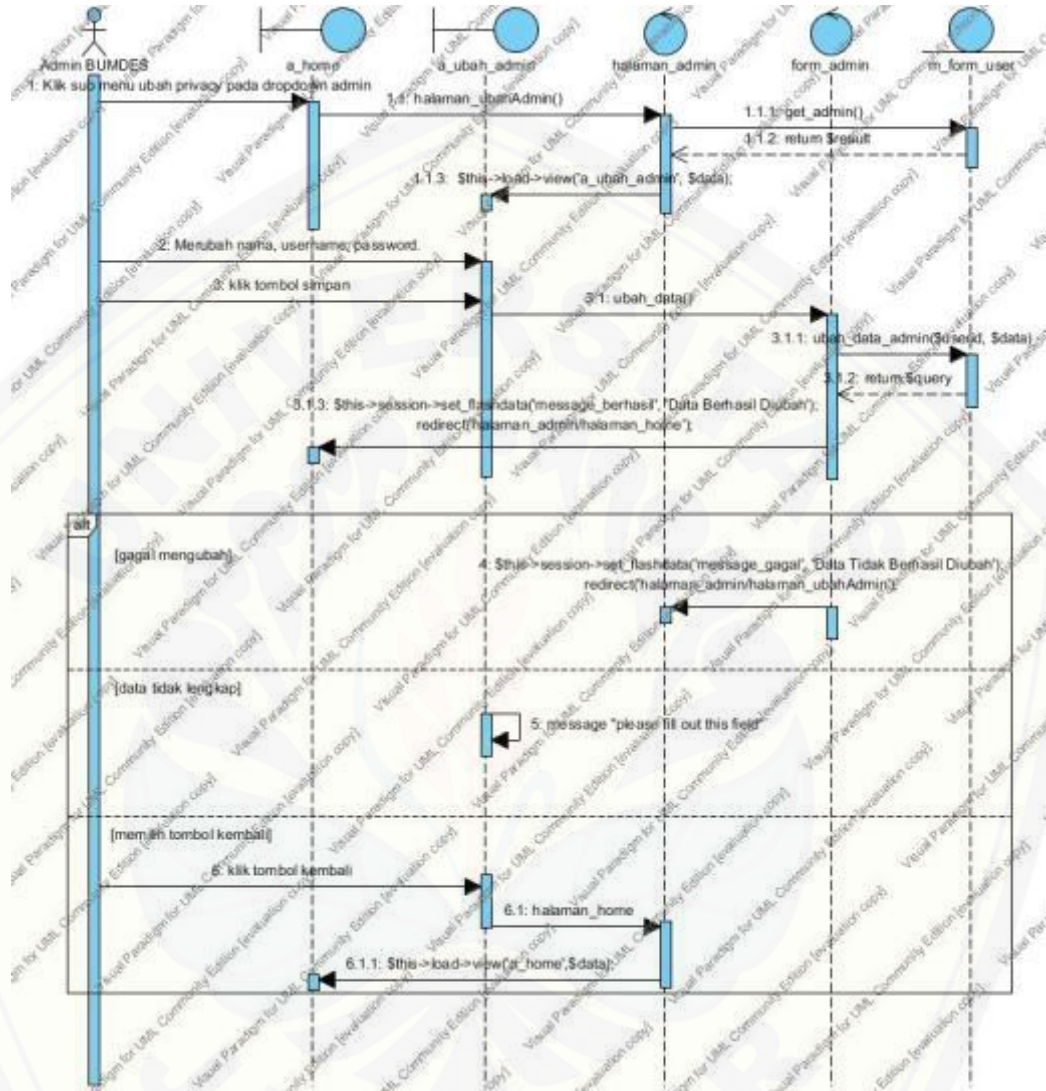
B.3. Sequence Diagram Ubah Informasi



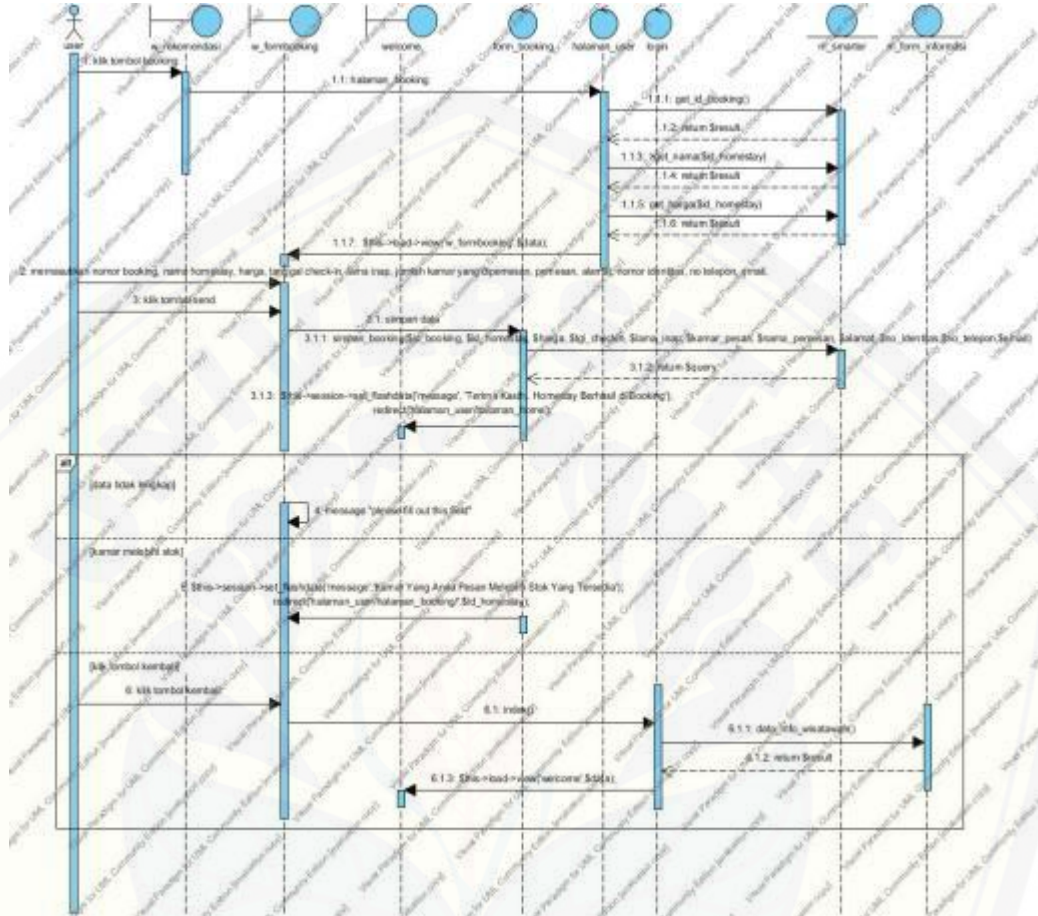
B.4. Sequence Diagram Lihat Informasi



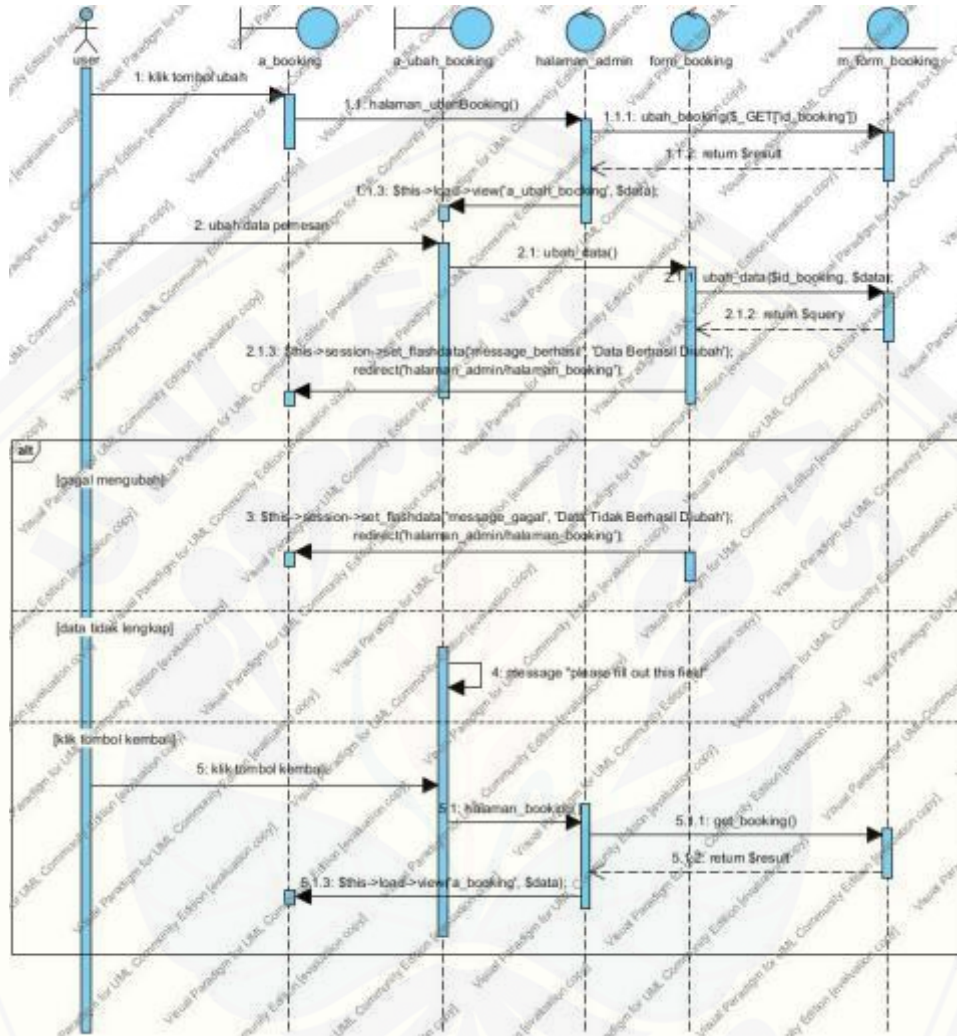
B.5. Sequence Diagram Ubah Admin



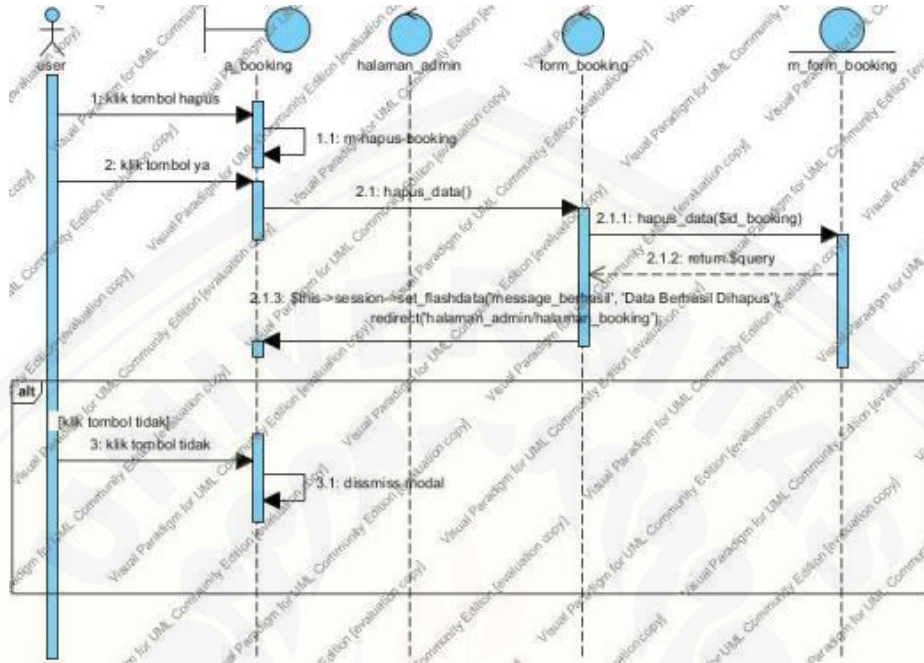
B.6. Sequence Diagram Booking Homestay



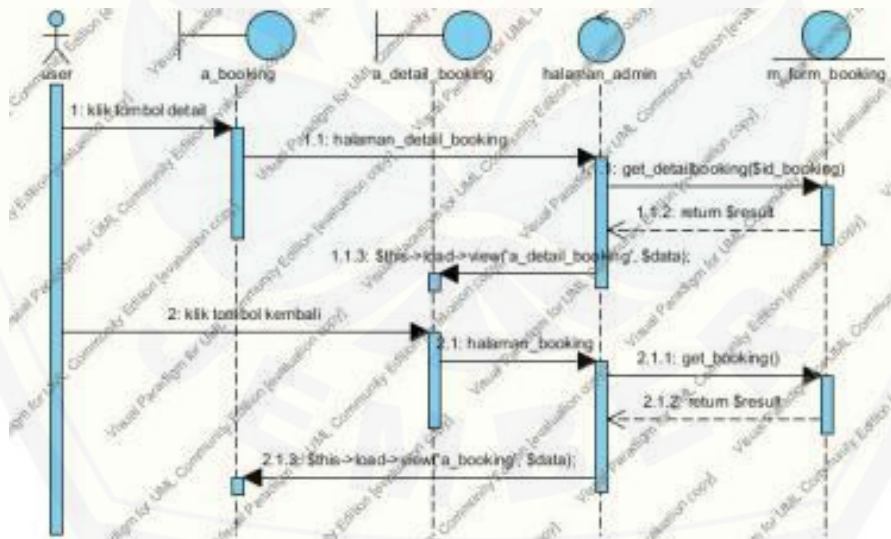
B.7. Sequence Diagram Ubah Booking



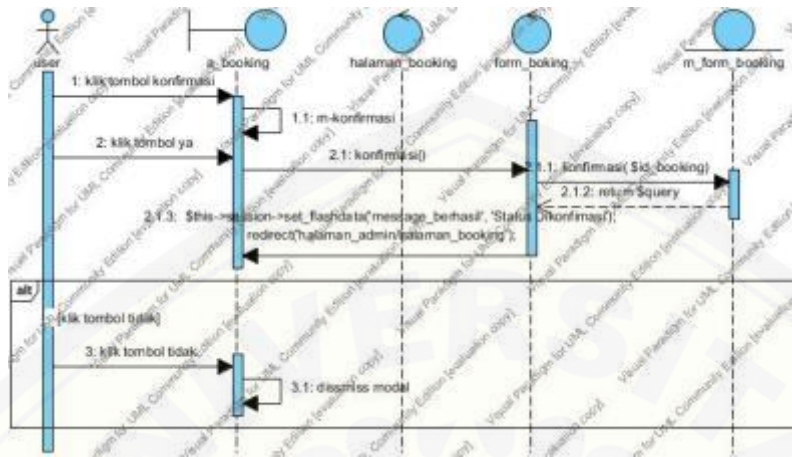
B.8. *Sequence Diagram Hapus Data Pemesan*



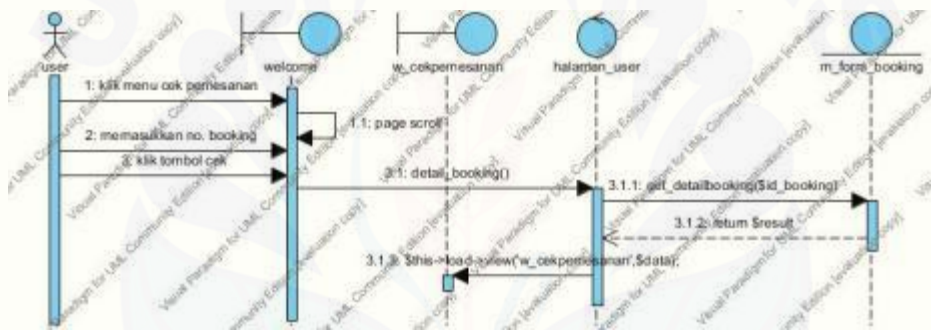
B.9. *Sequence Diagram Lihat Detail Booking*



B.10. *Sequence Diagram* Konfirmasi Booking



B.11. *Sequence Diagram* Cek Detail Pemesanan

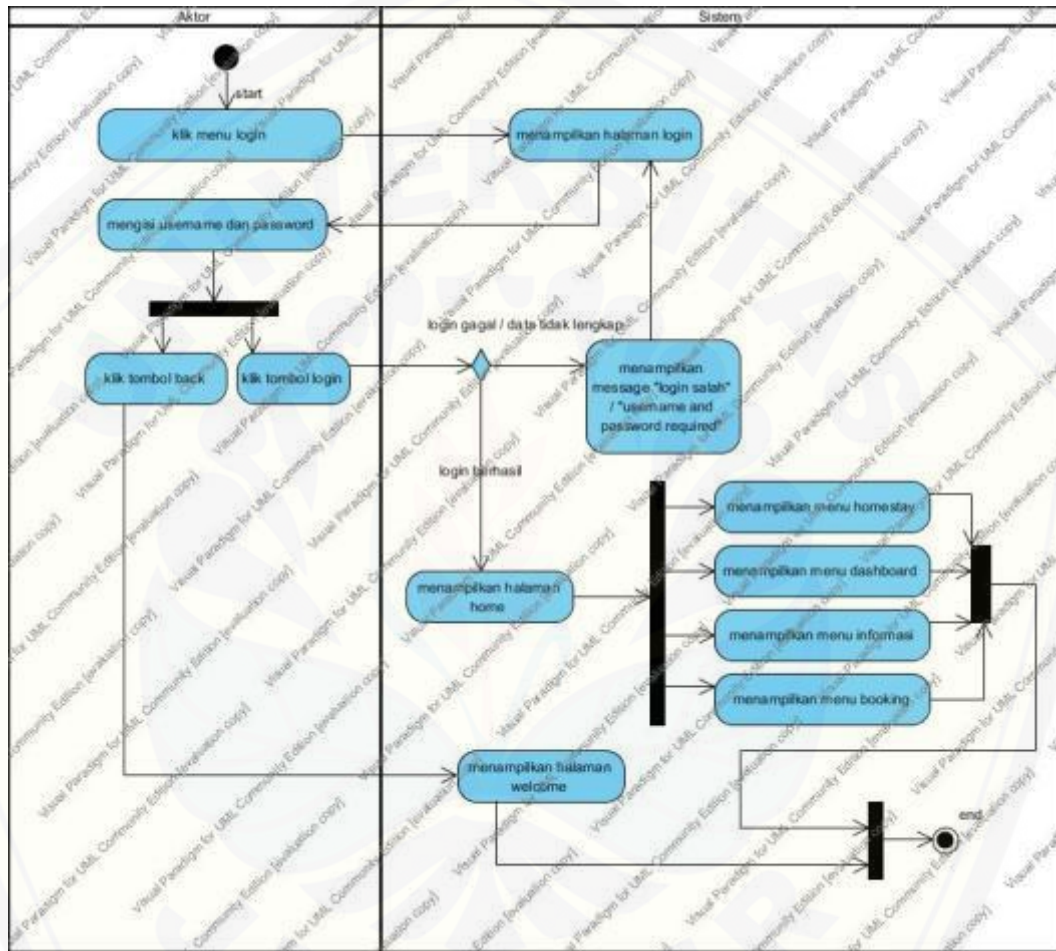


B.12. Sequence Diagram Lihat Detail Perhitungan

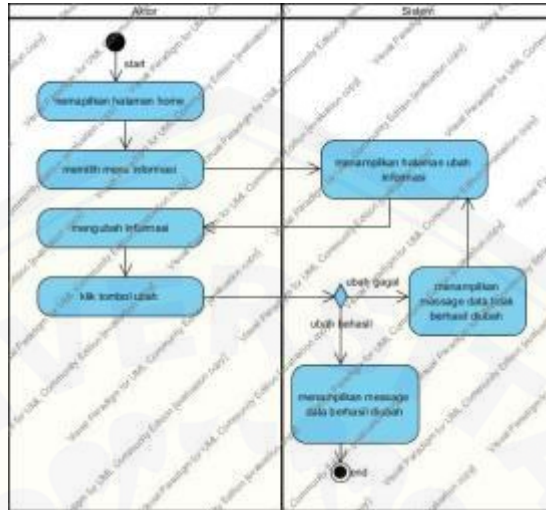


LAMPIRAN C

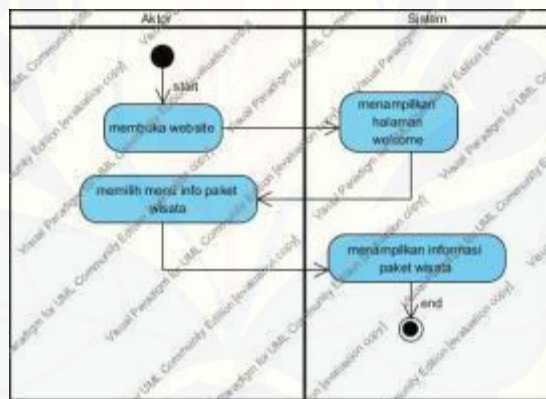
C.1. Activity Diagram Login



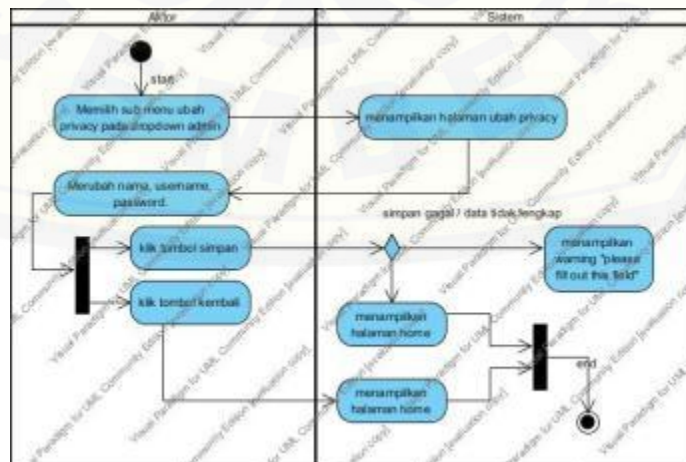
C.3. Activity Diagram Ubah Informasi



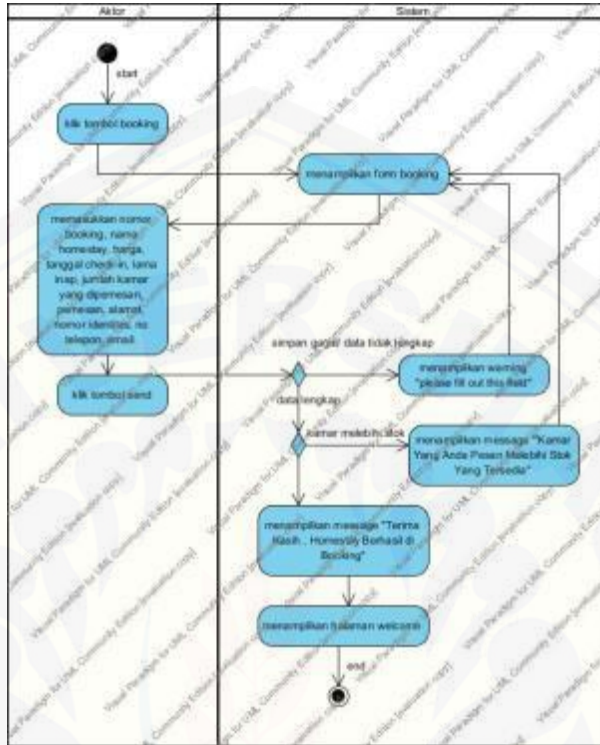
C.4. Activity Diagram Lihat Informasi



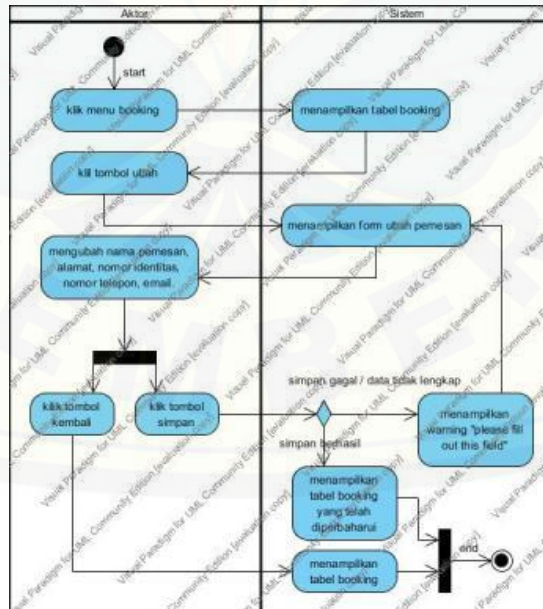
C.5. Activity Diagram Ubah Admin



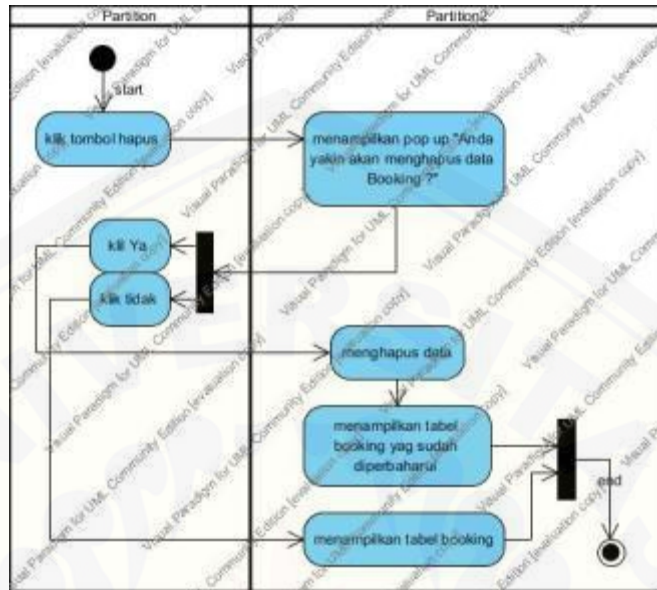
C.6. Activity Diagram Booking Homestay



C.7. Activity Diagram Ubah Data Pemesan



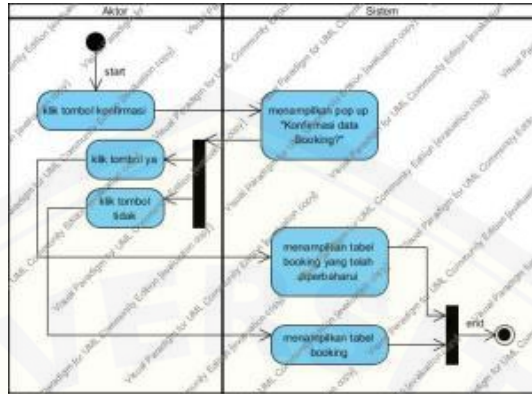
C.8. Activity Diagram Hapus Data Pemesan



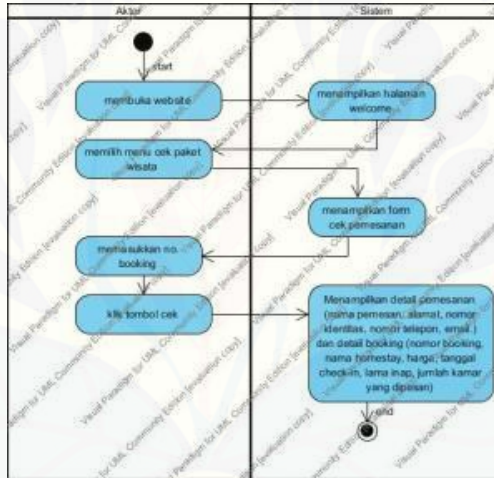
C.9. Activity Diagram Lihat Detail Booking



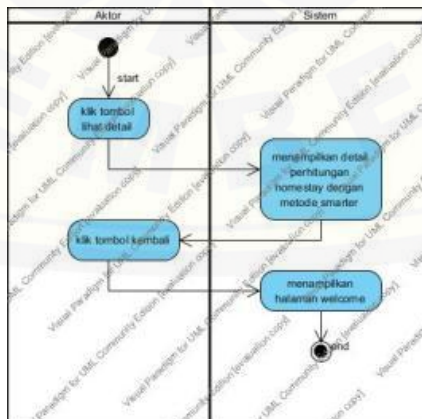
C.10. Activity Diagram Konfirmasi Booking



C.11. Activity Diagram Cek Detail Pemesanan



C.12. Activity Diagram Lihat Detail Perhitungan



LAMPIRAN D

D.1. Kode Program *Login*a. *Controller login*

```
function index() {
    $data=array(
        'title'=>'Data Informasi',
        'info_wisatawan'=>$this->M_form_informasi->data_info_wisatawan()
    );

    $this->load->view('welcome',$data);
}

function halaman_login() {
    $this->load->view('v_login');
}

function proses() {
    $user = $this->session->userdata('username');
    $data['detail'] = $this->madmin->detail_user($user)->result();

    $username = $this->input->post('username');
    $password = $this->input->post('password');

    $status = $this->M_Login->Auth($username, $password);
    if($status){
        //jika username & password match
        $data = $this->M_Login->getDataLogin($username)->result();
        $sesi = array(
            'username' => $username,
            'id_level' => $data[0]->id_level
        );

        $this->session->set_userdata($sesi);
        if($this->session->userdata('id_level') == 1){
            //Load view admin
            $user = $this->session->userdata('username');
            $data['detail'] = $this->madmin->detail_user($user)->result();

            $this->load->view('a_home',$data);
        }
    }
    else{
        //username & password salah
        $this->session->set_flashdata ('message', 'Login salah');
        redirect('login/halaman_login');
        //echo "<script>alert('Username atau Password Tidak Sesuai');</script>";
    }
}

function logout() {
    $this->M_session->destroy_session();
    $this->index();
}
```


b. Model `m_login`

```

public function Auth($username, $password) {
    $query = "select count(*) as hasil from user where username = '". $username. "' and password = '". $password. "'";
    $result = $this->db->query($query)->result();

    if($result[0]->hasil > 0){
        return true;
    }else{
        return false;
    }
}

}

public function getDataLogin($username){
    // $query = select * from Login where username = $username
    // return $this->db->query($query);
    $array = array(
        'username' => $username
    );
    $this->db->where($array);
    return $this->db->get('user');
}

function get_data($username, $password) {
    return $this->db->query("SELECT username, level FROM user WHERE username='$username' AND password='$password'");
}

```

D.2. Kode Program Kelola Data *Homestay*

a. Controller halaman_admin

```

function halaman_homestay() {

    $data=array(
        'title'=>'Data Homestay',
        'id_homestay'=>$this->M_form_homestay->get_id_homestay(),
        'data_homestay'=>$this->M_form_homestay->data_homestay(),
    );

    $user = $this->session->userdata('username');
    $data['detail'] = $this->madmin->detail_user($user)->result();

    $this->load->view('a_homestay',$data);
}

function halaman_detail_homestay($id_homestay) {
    $data=array(
        'title'=>'Data Detail Homestay',
        'detail_homestay'=>$this->M_form_homestay->detail_homestay($id_homestay)
    );

    $user = $this->session->userdata('username');
    $data['detail'] = $this->madmin->detail_user($user)->result();

    $this->load->view('a_detail_homestay', $data);
}

function halaman_ubahHomestay() {
    $data=array(
        'title'=>'Data Ubah Homestay',
        'get_homestay'=>$this->M_form_homestay->get_homestay($_GET['id_homestay']),
    );

    $user = $this->session->userdata('username');
    $data['detail'] = $this->madmin->detail_user($user)->result();

    $this->load->view('a_ubah_homestay', $data);
}

```

b. *Controller form_homestay*

```

function simpan_data() {
    $id = $this->input->post('id_homestay');
    $id_kriteriahomestay = $this->input->post('id_kriteriahomestay');
    $nama_homestay = $this->input->post('nama_homestay');
    $pemilik = $this->input->post('pemilik');
    $alamat_homestay = $this->input->post('alamat_homestay');
    $no_identities = $this->input->post('no_identities');
    $no_telepon = $this->input->post('no_telepon');
    $harga = $this->input->post('harga');
    $jarak_ijen = $this->input->post('jarak_ijen');
    $jarak_saranaum = $this->input->post('jarak_saranaum');
    $jumlah_kamar = $this->input->post('jumlah_kamar');
    $wifi = $this->input->post('wifi');
    $transportasi = $this->input->post('transportasi');
    $katering = $this->input->post('katering');
    $config['upload_path'] = './img/';
    $config['allowed_types'] = 'gif|jpg|png';
    $this->load->library('upload', $config);
    $this->upload->do_upload();
    $upload_data = $this->upload->data();
    $foto_homestay = $upload_data['file_name'];
    $id_status = 6;

    $data = array(
        'id_homestay' => $id,
        'nama_homestay' => $nama_homestay,
        'pemilik' => $pemilik,
        'alamat_homestay' => $alamat_homestay,
        'no_identities' => $no_identities,
        'no_telepon' => $no_telepon,
        'foto_homestay' => $foto_homestay,
        'id_status' => $id_status
    );

    $this->load->model('M_form_homestay');
    $logic = $this->M_form_homestay->simpan_data($data);
    if ($logic == true) {
        $this->session->set_flashdata('message_berhasil', 'Data Berhasil Ditambahkan');
        $this->M_form_homestay->simpan_kriteria($id_kriteriahomestay, $id,$harga,$jumlah_kamar,$jarak_ijen,
            $jarak_saranaum,$wifi,$transportasi,$katering);
        redirect('halaman_admin/halaman_homestay');
    } else {
        $this->session->set_flashdata('message_gagal', 'Data Tidak Berhasil Ditambahkan');
        redirect('halaman_admin/halaman_homestay');
    }
}

function ubah_data() {
    $id = $this->input->post('id_homestay');
    $id_kriteriahomestay = $this->input->post('id_kriteriahomestay');
    $nama_homestay = $this->input->post('nama_homestay');
    $pemilik = $this->input->post('pemilik');
    $alamat_homestay = $this->input->post('alamat_homestay');
    $no_identities = $this->input->post('no_identities');
    $no_telepon = $this->input->post('no_telepon');
    $harga = $this->input->post('harga');
    $jarak_ijen = $this->input->post('jarak_ijen');
    $jarak_saranaum = $this->input->post('jarak_saranaum');
    $jumlah_kamar = $this->input->post('jumlah_kamar');
    $wifi = $this->input->post('wifi');
    $transportasi = $this->input->post('transportasi');
    $katering = $this->input->post('katering');
    $config['upload_path'] = './img/';
    $config['allowed_types'] = 'gif|jpg|png';
    $this->load->library('upload', $config);
    $this->upload->do_upload();
    $upload_data = $this->upload->data();
    $foto_homestay = $upload_data['file_name'];

    $data = array(
        'id_homestay' => $id,
        'nama_homestay' => $nama_homestay,
        'pemilik' => $pemilik,
        'alamat_homestay' => $alamat_homestay,
        'no_identities' => $no_identities,
        'no_telepon' => $no_telepon,
        'foto_homestay' => $foto_homestay
    );

    $this->load->model('M_form_homestay');
    $logic = $this->M_form_homestay->ubah_data($id, $data);
    if ($logic == true) {
        $this->session->set_flashdata('message_berhasil', 'Data Berhasil Diubah');
        $this->M_form_homestay->ubah_kriteria($id_kriteriahomestay, $id,$harga,$jumlah_kamar,
            $jarak_ijen,$jarak_saranaum,$wifi,$transportasi,$katering);
        redirect('halaman_admin/halaman_homestay');
    } else {
        $this->session->set_flashdata('message_gagal', 'Data Tidak Berhasil Diubah');
        redirect('halaman_admin/halaman_homestay');
    }
}

function hapus_data() {
    $id = $this->input->post('id_homestay');
    $this->load->model('M_form_homestay');
    $this->M_form_homestay->hapus_data($id);
    $this->session->set_flashdata('message_berhasil', 'Homestay Berhasil Dihapus');
    redirect('halaman_admin/halaman_homestay');
}

```

c. Model *m_form_homestay*

```

function data_homestay() {
    return $this->db->query("SELECT * FROM data_homestay dh join kriteria_homestay kh
    on dh.id_homestay=kh.id_homestay join status s
    on dh.id_status=s.id_status where dh.id_status='6' ")>result_array();
}

function get_homestay($id_homestay) {
    $query = $this->db->query(
    "SELECT * FROM data_homestay dh join kriteria_homestay kh
    on dh.id_homestay=kh.id_homestay where dh.id_homestay='$id_homestay'">result_array();
    return $query;
}

function detail_homestay($id_homestay) {
    $query = $this->db->query(
    "SELECT * FROM data_homestay dh join kriteria_homestay kh
    on dh.id_homestay=kh.id_homestay join status s
    on dh.id_status=s.id_status where dh.id_homestay='$id_homestay'">result_array();
    return $query;
}

function get_id_homestay() {
    $q = $this->db->query("select MAX(RIGHT(id_homestay,4)) as id_max from data_homestay");
    $skh = '';
    if ($q->num_rows() > 0) {
        foreach ($q->result_array() as $k) {
            $tmp = ((int) $k['id_max']) + 1;
            $skh = sprintf("%04s", $tmp);
        }
    } else {
        $skh = '0001';
    }
    return 'H-' . $skh;
}

function simpan_data($data) {
    return $this->db->insert('data_homestay', $data);
}

function simpan_kriteria($id_kriteriahomestay, $id, $harga, $jumlah_kamar, $jarak_ijen, $jarak_saranaum, $wifi, $transportasi, $katering) {
    $this->db->query("INSERT INTO 'kriteria_homestay' ( 'id_kriteriahomestay', 'id_homestay', 'harga',
    'jumlah_kamar', 'jarak_ijen', 'jarak_saranaum', 'wifi', 'transportasi', 'katering' )
    VALUES ('$id_kriteriahomestay', '$id', '$harga', '$jumlah_kamar', '$jarak_ijen',
    '$jarak_saranaum', '$wifi', '$transportasi', '$katering')");
}

function ubah_data($id, $data) {
    $this->db->where('id_homestay', $id);
    return $this->db->update('data_homestay', $data);
}

function ubah_kriteria( $id_kriteriahomestay, $id, $harga, $jumlah_kamar, $jarak_ijen, $jarak_saranaum, $wifi, $transportasi, $katering) {
    $this->db->query("UPDATE 'kriteria_homestay' SET 'id_homestay'='".$id."',
    'harga'='".$harga."',
    'jumlah_kamar'='".$jumlah_kamar."',
    'jarak_ijen'='".$jarak_ijen."',
    'jarak_saranaum'='".$jarak_saranaum."',
    'wifi'='".$wifi."',
    'transportasi'='".$transportasi."',
    'katering'='".$katering.'"
    where 'id_kriteriahomestay'='".$id_kriteriahomestay.'"");
}

function hapus_data($id) {
    $this->db->query("UPDATE 'data_homestay' SET 'id_status'='7' where 'id_homestay' = '".$id.'"");
}

```

D.3. Kode Program Ubah Informasi

a. Controller Halaman _Admin

```

function halaman_informasi() {
    $data=array(
        'title'=>'Data Informasi',
        'info_wisatawan'=>$this->M_form_informasi->data_info_wisatawan()
    );

    $user = $this->session->userdata('username');
    $data['detail'] = $this->admin->detail_user($user)->result();

    $this->load->view('a_info', $data);
}

```

b. *Controller form_informasi*

```
function ubah_data_info_wisatawan() {  
    $infowisatawan = $this->input->post('infowisatawan');  
    $logic = $this->M_form_informasi->ubah_data_wisatawan($infowisatawan);  
    if($logic == true) {  
        $this->session->set_flashdata('message_berhasil_wisatawan', 'Data Berhasil Diubah');  
        redirect('halaman_admin/halaman_informasi');  
    } else {  
        $this->session->set_flashdata('message_gagal_wisatawan', 'Data Tidak Berhasil Diubah');  
        redirect('halaman_admin/halaman_informasi');  
    }  
}
```

c. *Model m_form_informasi*

```
function ubah_data_wisatawan($infowisatawan) {  
    return $this->db->query("UPDATE informasi SET info = " . $infowisatawan . " WHERE id_info = 'wisatawan'");  
}
```

D.4. Kode Program Lihat Informasia. *Controller login*

```
function index() {  
    $data=array(  
        'title'=>'Data Informasi',  
        'info_wisatawan'=>$this->M_form_informasi->data_info_wisatawan()  
    );  
    $this->load->view('welcome',$data);  
}
```

b. *Model halaman_admin*

```
function data_info_wisatawan() {  
    return $this->db->query("SELECT info FROM informasi ")->result_array();  
}
```

D.5. Kode Program Ubah Admin

a. Controller halaman_admin

```
function halaman_ubahAdmin() {
    $data=array(
        'title'=>'Data Ubah User',
        'get_admin'=>$this->M_form_user->get_admin(),
    );

    $user = $this->session->userdata('username');
    $data['detail'] = $this->madmin->detail_user($user)->result();

    $this->load->view('a_ubah_admin', $data);
}
```

b. Controller form_admin

```
function ubah_data() {
    $userid = $this->input->post('userid');
    $nama_user = $this->input->post('nama_user');
    $username = $this->input->post('username');
    $password = $this->input->post('password');
    $data = array(
        'userid' => $userid,
        'nama_user' => $nama_user,
        'username' => $username,
        'password' => $password,
    );

    $this->load->model('M_form_user');
    $logic = $this->M_form_user->ubah_data_admin($userid, $data);
    if ($logic == true) {
        $this->session->set_flashdata('message_berhasil', 'Data Berhasil Diubah');
        redirect('halaman_admin/halaman_home');
    } else {
        $this->session->set_flashdata('message_gagal', 'Data Tidak Berhasil Diubah');
        redirect('halaman_admin/halaman_ubahAdmin');
    }
}
```

c. Model m_form_user

```
function get_admin() {
    $query = $this->db->query("SELECT * FROM user u join level 1 on u.id_level=l.id_level where u.id_level='1'");
    return $query;
}

function ubah_data_admin($userid, $data) {
    $this->db->where('userid', $userid);
    return $this->db->update('user', $data);
}
```


c. Model *m_form_booking*

```

function get_booking() {
    return $this->db->query("SELECT * FROM booking b join detail_booking db on db.id_booking=b.id_booking join data_homestay dh
    on dh.id_homestay=db.id_homestay join kriteria_homestay kh on dh.id_homestay=kh.id_homestay join status s
    on s.id_status=b.id_status where b.id_status='2' or b.id_status='4' or b.id_status='5' order by b.id_status asc")->result_array();
}

function get_booking1() {
    return $this->db->query("SELECT id_status,b.id_booking as id_booking,(datediff(tgl_booking,now()))*-1) as batas_booking
    FROM booking b join detail_booking db on db.id_booking=b.id_booking");
}

function get_nama($id_homestay) {
    return $this->db->query("SELECT * FROM `data_homestay` where id_homestay = '$id_homestay'");
}

function cek_tgl($id_homestay) {
    return $this->db->query("SELECT tgl_checkin, tgl_checkout FROM `detail_booking` where id_homestay = '$id_homestay'");
}

function cek_kamar($id_homestay,$tgl_checkin) {
    return $this->db->query("SELECT b.kamar_pesan as kamar_pesan, k.jumlah_kamar as jumlah_kamar, b.tgl_checkin as tgl_checkin
    FROM kriteria_homestay k join data_homestay d on d.id_homestay=k.id_homestay
    join detail_booking b on d.id_homestay=b.id_homestay where d.id_homestay = '$id_homestay' and b.tgl_checkin='$tgl_checkin'");
}

function cek_kamarKosong($id_homestay) {
    return $this->db->query("SELECT jumlah_kamar
    FROM kriteria_homestay k where id_homestay = '$id_homestay'");
}

function get_harga($id_homestay) {
    return $this->db->query("SELECT * FROM `kriteria_homestay` where id_homestay = '$id_homestay'");
}

function simpan_booking($id_booking, $id_homestay, $harga, $tgl_checkin,$lama_inap, $kamar_pesan, $nama_pemesan, $alamat,
    $no_identitas, $no_telepon,$email ){
    $insert= $this->db->query("INSERT INTO `booking` (`id_booking`,`tgl_booking`, `nama_pemesan`, `alamat`, `no_identitas`,
    `no_telepon`, `email`, `id_status`)
    VALUES ('".$id_booking."',now(), '".$nama_pemesan."', '".$alamat."', '".$no_identitas."', '".$no_telepon."', '".$email."', '2')");

    $insert= $this->db->query("INSERT INTO `detail_booking` (`id_booking`,`id_homestay`, `tgl_checkin`, `tgl_checkout`, `harga`, `kamar_pesan`)
    VALUES ('".$id_booking."', '".$id_homestay."', '".$tgl_checkin."', ADDDATE('$tgl_checkin', INTERVAL $lama_inap DAY) ,
    |'".$harga*$kamar_pesan."', '".$kamar_pesan.'")");

    return $insert;
}

function get_id_booking() {
    $sq = $this->db->query("select MAX(RIGHT(id_booking,6)) as id_max from booking");
    $skd = '';
    if ($sq->num_rows() > 0) {
        foreach ($sq->result_array() as $sk) {
            $tmp = ((int) $sk['id_max']) + 1;
            $skd = sprintf("%06s", $tmp);
        }
    } else {
        $skd = '000001';
    }
    return 'B-' . $skd;
}

```

D.7. Kode Program Ubah Data Pemesan

a. Controller halaman_admin

```

function halaman_ubahBooking() {
    $data=array(
        'title'=>'Data Ubah Booking',
        'ubah_booking'=>$this->M_form_booking->ubah_booking($_GET['id_booking'])
    );

    $user = $this->session->userdata('username');
    $data['detail'] = $this->madmin->detail_user($user)->result();

    $this->load->view('a_ubah_booking', $data);
}

```

b. Controller form_booking

```
function ubah_data() {
    $id_booking = $this->input->post('id_booking');
    $nama_pemesan = $this->input->post('nama_pemesan');
    $alamat = $this->input->post('alamat');
    $no_identitas = $this->input->post('no_identitas');
    $no_telepon = $this->input->post('no_telepon');
    $email = $this->input->post('email');

    $data = array(
        'id_booking' => $id_booking,
        'nama_pemesan' => $nama_pemesan,
        'alamat' => $alamat,
        'no_identitas' => $no_identitas,
        'no_telepon' => $no_telepon,
        'email' => $email
    );

    $this->load->model('M_form_booking');
    $logic = $this->M_form_booking->ubah_data($id_booking, $data);
    if ($logic == true) {
        $this->session->set_flashdata('message_berhasil', 'Data Berhasil Diubah');
        redirect('halaman_admin/halaman_booking');
    } else {
        $this->session->set_flashdata('message_gagal', 'Data Tidak Berhasil Diubah');
        redirect('halaman_admin/halaman_booking');
    }
}
```

c. Model m_form_booking

```
function ubah_data($id_booking, $data) {
    $this->db->where('id_booking', $id_booking);
    return $this->db->update('booking', $data);
}
```

D.8. Kode Program Hapus Data Pemesan**a. Controller form_booking**

```
function hapus_data() {
    $id_booking = $this->input->post('id_booking');
    $this->load->model('M_form_booking');
    $this->M_form_booking->hapus_data($id_booking);
    $this->session->set_flashdata('message_berhasil', 'Data Berhasil Dihapus');
    redirect('halaman_admin/halaman_booking');
}
```

b. Model *m_form_booking*

```
function hapus_data($id_booking) {
    $query = $this->db->query("UPDATE `booking` SET `id_status` = '8'
                               where `id_booking` = '$id_booking'");
}
```

D.9. Kode Program Lihat Detail *Booking*a. Controller *halaman_admin*

```
function halaman_detail_booking($id_booking) {
    $data=array(
        'title'=>'Data Detail Booking',
        'detail_booking'=>$this->M_form_booking->get_detailbooking($id_booking)
    );

    $user = $this->session->userdata('username');
    $data['detail'] = $this->madmin->detail_user($user)->result();

    $this->load->view('a_detail_booking', $data);
}
```

b. Model *m_form_booking*

```
function get_detailbooking($id_booking) {
    return $this->db->query("SELECT * FROM booking b join detail_booking db on db.id_booking=b.id_booking join data_homestay dh
                           on dh.id_homestay=db.id_homestay join kriteria_homestay kh on dh.id_homestay=kh.id_homestay join status s
                           on s.id_status=b.id_status where b.id_booking like '%" . $id_booking . "%' ")->result_array();
}
```

D.10. Kode Program Konfirmasi *Booking*a. Controller *form_booking*

```
function konfirmasi() {
    $id_booking = $this->input->post('id_booking');
    $this->load->model('M_form_booking');
    $logic = $this->M_form_booking->konfirmasi($id_booking);
    $this->session->set_flashdata('message_berhasil', 'Status Dikonfirmasi');
    redirect('halaman_admin/halaman_booking');
}
}
```

b. Model *m_form_booking*

```
function konfirmasi( $id_booking) {
    $query = $this->db->query("UPDATE `booking` SET `id_status`='4' where `id_booking`= '$id_booking' ");
    return $query;
}
```

D.11. Kode Program Cek Detail Pemesanan

a. Controller halaman_user

```
public function detail_booking(){
    $id_booking= $this->input->post('id_booking');

    $data=array(

        'detail_booking' =>$this->M_form_booking->get_detailbooking($id_booking)
    );

    $this->load->view('w_cekpemesanan',$data);
}
```

b. Model *m_form_booking*

```
function get_detailbooking($id_booking) {
    return $this->db->query("SELECT * FROM booking b join detail_booking db on db.id_booking=b.id_booking join data_homestay dh
    on dh.id_homestay=db.id_homestay join kriteria_homestay kh on dh.id_homestay=kh.id_homestay join status s
    on s.id_status=b.id_status where b.id_booking like '%" . $id_booking . "%' ")>result_array();
}
```

D.12. Kode Program Lihat Detail Perhitungan

a. Controller halaman_user

```
function halaman_detail_hitung($id_homestay) {
    $data=array(
        'title'=>'Data Detail Homestay',
        'detail_subkriteriau'=>$this->M_smarter->detail_subkriteriau(),
        'detail_subkriteriai'=>$this->M_smarter->detail_subkriteriai(),
        'detail_subsubharga'=>$this->M_smarter->detail_subsubharga(),
        'detail_subsubjarakijen'=>$this->M_smarter->detail_subsubjarakijen(),
        'detail_subsubjaraksarana'=>$this->M_smarter->detail_subsubjaraksarana(),
        'detail_subsubjumlahkamar'=>$this->M_smarter->detail_subsubjumlahkamar(),
        'detail_subsubwifi'=>$this->M_smarter->detail_subsubwifi(),
        'detail_subsubtransportasi'=>$this->M_smarter->detail_subsubtransportasi(),
        'detail_subsubkatering'=>$this->M_smarter->detail_subsubkatering(),
        'score' => $this->M_smarter->score($id_homestay),
        'perkalianumum' => $this->M_smarter->perkalian($id_homestay, 1),
        'perkaliankhusus' => $this->M_smarter->perkalian($id_homestay, 2),
        'jumlahumum' => $this->M_smarter->jumlah($id_homestay, 1),
        'jumlahkhusus' => $this->M_smarter->jumlah($id_homestay, 2),
        'jumlah' => $this->M_smarter->jumlah2($id_homestay)
    );

    $this->load->view('w_detail_hitung', $data);
}
```


b. Model m_smarter

```

function detail_subkriteriaiu() {
    return $this->db->query("SELECT * FROM subkriteria s join prioritas_subkriteria ps on s.id_subkriteria=ps.id_subkriteria
    where ps.id_prioritas in(1,2,3,4) ")>result_array();
}

function detail_subkriteriak() {
    return $this->db->query("SELECT * FROM subkriteria s join prioritas_subkriteria ps on s.id_subkriteria=ps.id_subkriteria
    where ps.id_prioritas in(5,6,7) ")>result_array();
}

function detail_subsubharga() {
    return $this->db->query("SELECT * FROM subsubkriteria ss join prioritas_subsubkriteria pss on ss.id_subsubkriteria=pss.id_subsubkriteria
    where pss.id_prioritas in(1,2,3) ")>result_array();
}

function detail_subsubjarakijen() {
    return $this->db->query("SELECT * FROM subsubkriteria ss join prioritas_subsubkriteria pss on ss.id_subsubkriteria=pss.id_subsubkriteria
    where pss.id_prioritas in(4,5,6) ")>result_array();
}

function detail_subsubjaraksarana() {
    return $this->db->query("SELECT * FROM subsubkriteria ss join prioritas_subsubkriteria pss on ss.id_subsubkriteria=pss.id_subsubkriteria
    where pss.id_prioritas in(7,8,9) ")>result_array();
}

function detail_subsubjumlahkamar() {
    return $this->db->query("SELECT * FROM subsubkriteria ss join prioritas_subsubkriteria pss on ss.id_subsubkriteria=pss.id_subsubkriteria
    where pss.id_prioritas in(10,11,12) ")>result_array();
}

function detail_subsubwifi() {
    return $this->db->query("SELECT * FROM subsubkriteria ss join prioritas_subsubkriteria pss on ss.id_subsubkriteria=pss.id_subsubkriteria
    where pss.id_prioritas in(13,14) ")>result_array();
}

function detail_subsubtransportasi() {
    return $this->db->query("SELECT * FROM subsubkriteria ss join prioritas_subsubkriteria pss on ss.id_subsubkriteria=pss.id_subsubkriteria
    where pss.id_prioritas in(15,16) ")>result_array();
}

function detail_subsubkatering() {
    return $this->db->query("SELECT * FROM subsubkriteria ss join prioritas_subsubkriteria pss on ss.id_subsubkriteria=pss.id_subsubkriteria
    where pss.id_prioritas in(17,18) ")>result_array();
}

function perkalian($id_homestay, $id) {
    return $this->db->query("SELECT * FROM perkalian r join data_homestay dh on r.id_homestay=dh.id_homestay join subkriteria s
    on r.id_subkriteria = s.id_subkriteria where r.id_homestay= 'h-0001' and s.id_kriteria = $id group by r.id_subkriteria ")>result_array();
}

function jumlah($id_homestay, $id) {
    return $this->db->query("SELECT sum(nilai) as jumlah FROM perkalian r join data_homestay dh on r.id_homestay=dh.id_homestay
    join subkriteria s on r.id_subkriteria = s.id_subkriteria where r.id_homestay= 'h-0001' and s.id_kriteria = $id")>result_array();
}

function jumlah2($id_homestay) {
    return $this->db->query("SELECT nilai, nama_kriteria, bobot FROM perkalian2 p join data_kriteria k on p.id_kriteria = k.id_kriteria
    where p.id_homestay = '$id_homestay'")>result_array();
}

function score($id_homestay) {
    return $this->db->query("SELECT * FROM ranking r join data_homestay dh on r.id_homestay=dh.id_homestay
    where r.id_homestay= '$id_homestay' ")>result_array();
}

```