

**APLIKASI PENCARIAN RUMAH KOS
MENGUNAKAN METODE PERBANDINGAN
EKSPONENSIAL (MPE)**

TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Teknik Pada
Jurusan Teknik Informatika

oleh :

KHAIRIL AZHAR
10651004380



**FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SULTAN SYARIF KASIM RIAU
PEKANBARU
2013**

APLIKASI PENCARIAN RUMAH KOS MENGUNAKAN METODE PERBANDINGAN EKSPONENSIAL (MPE)

KHAIRIL AZHAR
10651004380

Tanggal Sidang : 25 Juni 2013

Periode Wisuda : November 2013

Jurusan Teknik Informatika

Fakultas Sains dan Teknologi

Universitas Islam Negeri Sultan Syarif Kasim Riau

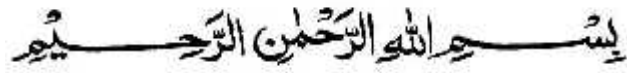
Jl. Soebrantas KM 15 No. 155 Pekanbaru

ABSTRAK

Usaha Rumah Kost merupakan salah satu sumber investasi yang tidak tergoncang oleh krisis global. Saat ini sistem pencarian rumah kost yang ada masih menggunakan metode pencarian penyaringan biasa tanpa memperdulikan tingkat kepentingan suatu kriteria. Salah satu alternatif untuk permasalahan tersebut adalah dengan menggunakan metode perbandingan eksponensial pada proses pencarian. Proses pencarian pada sistem ini dimulai dengan memasukkan tingkat kepentingan dan kemudian sistem menampilkan hasil sesuai dengan perhitungan metode perbandingan eksponensial. Setiap rumah kost yang ditampilkan pada pencarian akan diarahkan ke halaman profil rumah kost yang menampilkan informasi lebih lanjut mengenai rumah kost. Pada halaman profil juga ditampilkan peta rumah kost menggunakan *web service* Google Maps. Sistem ini dapat memberikan manfaat bagi pencari rumah kost berupa urutan rumah kost yang disarankan. Berdasarkan kuesioner terhadap beberapa pencari rumah kost, diperoleh hasil sebanyak 80,5% responden menyatakan bahwa sistem menyediakan informasi dengan sangat baik kepada pencari rumah kost karena hasil pencarian tidak hanya ditampilkan tetapi juga diurut berdasarkan tingkat kebutuhan pencari rumah kost dengan menggunakan perhitungan metode perbandingan eksponensial.

Kata kunci : Rumah kost, *Google Maps*, Metode Perbandingan Eksponensial

KATA PENGANTAR



Assalammu'alaikumwrwb.

Alhamdulillah *rabbil'alam*, penulis ucapkan sebagai tanda syukur yang sebesar-besarnya kepada Allah SWT, atas segala karunia dan rahmat yang diberikan-Nya, sehingga penulis dapat menyelesaikan tugas akhir ini dengan baik. Shalawat serta salam terucap buat junjungan Baginda Rasulullah Muhammad SAW, karena jasa Beliau kita bisa menikmati zaman yang penuh dengan ilmu pengetahuan seperti sekarang ini.

Tugas akhir ini disusun sebagai salah satu syarat untuk mendapatkan gelar kesarjanaan pada jurusan Teknik Informatika Universitas Islam Negeri Sultan Syarif Kasim Riau. Penulisan dan penyusunan laporan tugas akhir ini tidak terlepas dari bantuan dari berbagai pihak yang telah memberikan masukan-masukan kepada penulis. Pada kesempatan ini, penulis mengucapkan banyak terimakasih kepada :

1. Bapak Prof. DR. H.M. Nazir, selaku Rektor Universitas Islam Negeri Sultan Syarif Kasim Riau.
2. Ibu Dra. Hj. Yenita Morena, M.Si, selaku Dekan Fakultas Sains dan Teknologi, Universitas Islam Negeri Sultan Syarif Kasim Riau.
3. Ibu Dr. Okfalisa, S.T, M.Sc, selaku Ketua Jurusan Teknik Informatika.
4. Bapak Benny Sukma Negara, S.T, M.T, selaku pembimbing I yang selalu memberikan bimbingan dan petunjuk sehingga tugas akhir ini dapat diselesaikan dengan baik.
5. Bapak M. Safrizal, S.T, M.Cs, selaku Penguji I tugas akhir.
6. Bapak Reski Mai Candra, S.T, M.Sc, selaku Penguji II tugas akhir.
7. Orang tuaku tercinta terutama ibu saya yang selalu memberikan doa, motivasi, bimbingan yang tiada hentinya, tiada habisnya bait doa yang diucapkan, serta telah banyak berkorban demi keberhasilan anak-anaknya.

Semoga selalu dalam lindungan Allah SWT dan segala pengorbanan beliau mendapat ridho dari Allah SWT Amiin.

8. Bg Arafat salahudin, S,Hi yang telah memberikan semangat dan membantu tiap ada kesulitan dalam menyelesaikan tugas akhir ini.
9. Segenap dosen Teknik Informatika yang tidak dapat saya sebutkan satu persatu yang telah banyak memberikan ilmu dan bimbingan akademis kepada penulis selama masa perkuliahan.
10. Teman-teman seperjuangan Jurusan Teknik Informatika angkatan 2006 UIN Suska Riau.
11. Seluruh pihak yang belum penulis cantumkan, terima kasih atas dukungannya.

Penulis menyadari bahwa dalam penulisan laporan ini masih banyak kesalahan dan kekurangan. Oleh karena itu, kritik dan saran yang sifatnya membangun sangat diharapkan untuk kesempurnaan laporan ini. Akhirnya, penulis berharap semoga laporan ini dapat memberikan sesuatu yang bermanfaat bagi siapa saja yang membacanya. Amin.

Wassalamu'alaikumwr.wb.

Pekanbaru, Juni 2013

KHAIRIL AZHAR

10651004380

DAFTAR ISI

| | |
|---|------|
| HALAMAN JUDUL LAPORAN..... | i |
| LEMBAR PERSETUJUAN | ii |
| LEMBAR PENGESAHAN | iii |
| LEMBAR HAK ATAS KEKAYAAN INTELEKTUAL..... | iv |
| LEMBAR PERSEMBAHAN | v |
| ABSTRAK | vi |
| <i>ABSTRACT</i> | vii |
| KATA PENGANTAR | viii |
| DAFTAR ISI..... | ix |
| DAFTAR GAMBAR | x |
| DAFTAR TABEL | xi |
| DAFTAR LAMPIRAN | xii |
| BAB I PENDAHULUAN | I-1 |
| 1.1.Latar Belakang | I-1 |
| 1.2.Rumusan Masalah | I-3 |
| 1.3.Batasan Masalah..... | I-3 |
| 1.4.Tujuan | I-4 |
| 1.5.Sistematika Penulisan | I-4 |
| BAB II LANDASAN TEORI | II-1 |
| 2.1.Algoritma Metode Perbandingan Eksponensial..... | II-1 |
| 2.2.Google Maps | II-3 |
| 2.3.PHP (<i>Hypertext Preprocessor</i>) | II-4 |
| 2.3.1.Client-side dan Server-side Programming | II-5 |
| 2.4.Framework | II-5 |
| 2.5.Codeigniter | II-6 |
| 2.5.1.Kelebihan Codeigniter | II-6 |
| 2.5.2.Fitur Codeigniter | II-7 |

| | |
|---|-------|
| 2.5.3.Sistem Codeigniter | II-9 |
| 2.5.4.MVC (Model – View – Controller) | II-10 |
| BAB III METODOLOGI PENELITIAN..... | III-1 |
| 3.1.Identifikasi Masalah | III-2 |
| 3.2.Perumusan Masalah | III-2 |
| 3.3.Pengumpulan Data | III-2 |
| 3.3.1.Studi Pustaka..... | III-2 |
| 3.4.Analisa Sistem..... | III-2 |
| 3.4.1.Analisa Sistem Sebelumnya..... | III-2 |
| 3.4.1.Analisa Sistem Baru | III-4 |
| 3.5.Perancangan | III-7 |
| 3.5.1.Subsistem Data..... | III-7 |
| 3.5.2.Subsistem Modal..... | III-7 |
| 3.5.3.Subsistem Dialog | III-8 |
| 3.6.Implementasi | III-8 |
| 3.7.Pengujian..... | III-8 |
| 3.7.Kesimpulan dan Saran..... | III-8 |
| BAB IV ANALISA DAN PERANCANGAN..... | IV-1 |
| 4.1.Deskripsi Umum Sistem | IV-1 |
| 4.2.Analisa Sistem..... | IV-2 |
| 4.1.1.Analisa Sistem Yang Berjalan | IV-2 |
| 4.1.2.Analisa Yang Akan Dibangun | IV-2 |
| 4.3.Analisa Fungsional Sistem..... | IV-2 |
| 4.3.1.Site map..... | IV-3 |
| 4.3.2.FlowChart..... | IV-3 |
| 4.3.2.1. Flowchart Login | IV-3 |
| 4.3.2.2. Flowchart Registrasi..... | IV-4 |
| 4.3.2.3. Flowchart Guest | IV-5 |
| 4.3.2.4. Flowchart Member | IV-6 |

| | |
|---|-------|
| 4.3.2.6. Flowchart Admin | IV-7 |
| 4.3.2.7. Flowchart Pencarian Rumah Kos | IV-8 |
| 4.3.2.8. Flowchart Menampilkan Map Dari Google Map | IV-10 |
| 4.3.2.9. Flowchart Penentuan Kordinat Rumah Kos | IV-10 |
| 4.4. Use Case Diagram | IV-11 |
| 4.4.1. Use Case Diagram Guest | IV-11 |
| 4.4.2. Use Case Diagram Member | IV-12 |
| 4.4.3. Use Case Diagram Admin | IV-13 |
| 4.5. Class Diagram | IV-13 |
| 4.6. Siquence Diagram | IV-14 |
| 4.6.1. Siquence Diagram Admin Login | IV-14 |
| 4.6.2. Siquence Diagram Member Login | IV-15 |
| 4.6.3. Siquence Diagram Pendaftaran | IV-15 |
| 4.6.4. Siquence Diagram Pencarian | IV-16 |
| 4.6.5. Siquence Diagram Ubah Info Pada Member | IV-17 |
| 4.6.6. Siquence Diagram Update Foto Galeri Pada Member | IV-17 |
| 4.6.7. Siquence Diagram Hapus Foto Galeri Pada Member | IV-18 |
| 4.6.8. Siquence Diagram Ubah Foto Profil Pada Member | IV-18 |
| 4.6.9. Siquence Diagram Kelola Akun Pada Member | IV-19 |
| 4.6.10. Siquence Diagram Kelola Member Pada Admin | IV-19 |
| 4.6.11. Siquence Diagram Kelola fasilitas Pada Admin | IV-20 |
| 4.7. Perancangan <i>Entity Relational Diagram (ERD)</i> | IV-20 |
| 4.8. Perancangan Tabel | IV-21 |
| 4.9. Perancangan <i>interface</i> | IV-23 |
| 4.9.1. Rancangan halaman <i>Home</i> | IV-23 |
| 4.9.2. Rancangan halaman pencarian | IV-24 |
| 4.9.3. Rancangan halaman <i>Registrasi</i> | IV-24 |
| 4.9.4. Rancangan halaman Tambah Kos | IV-25 |
| 4.9.5. Rancangan halaman <i>Login User</i> | IV-25 |

| | |
|---|-------|
| 4.9.6.Rancangan halaman <i>Login Admin</i> | IV-26 |
| 4.9.7.Rancangan halaman <i>Admin</i> | IV-27 |
| 4.9.8.Rancangan halaman Tambah Rumah Kos Pada <i>Member</i> | IV-28 |
| 4.9.9.Rancangan halaman Ubah Kordinat Lokasi Pada <i>Member</i> | IV-29 |
| 4.9.10.Rancangan halaman Tambah Galeri Pada <i>member</i> | IV-29 |
| 4.8.Perhitungan Metode Perbandingan Eksponensial | IV-30 |
| BAB V IMPLEMENTASI DAN PENGUJIAN | V-1 |
| 5.1.Implementasi | V-1 |
| 5.1.1.Lingkungan Implementasi | V-2 |
| 5.1.2.Batasan Implementasi | V-2 |
| 5.1.2.Hasil Implementasi..... | V-2 |
| 5.2.Pengujian Gues | V-3 |
| 5.3.Pengujian Member | V-8 |
| 5.4.Pengujian Admin..... | V-21 |
| 5.5.Pengujian <i>Blackbox</i> | V-26 |
| BAB IV PENUTUP | VI-1 |
| 6.1.Kesimpulan | VI-1 |
| 6.2.Saran..... | VI-2 |
| DAFTAR PUSTAKA | xiii |
| LAMPIRAN A | A-1 |
| LAMPIRAN B | B-1 |
| LAMPIRAN C | C-1 |

BAB I

PENDAHULUAN

1.1 Latar Belakang

Menurut data yang telah kami terima, tingkat kelulusan ujian nasional (UN) siswa SMA sederajat tahun ini di Pekanbaru mencapai lebih 99,8 persen," kata Kepala Dinas Pendidikan (Disdik) Pekanbaru Yuzamri Yakub di Pekanbaru, Sabtu, (26/5). Perkembangan perguruan tinggi di Provinsi Riau khususnya kota Pekanbaru selalu meningkat dari tahun ketahun. Persentase kelulusan UN SMA/MA di Provinsi Riau, sesuai data yang diterima, mencapai 99,87 persen dari jumlah peserta 47.838 siswa dan untuk SMK mencapai 99,91 persen dari jumlah peserta 18.268 siswa. Jika dikurangi dengan jumlah peserta, diketahui siswa SMA yang tidak lulus tahun ini 63 orang dan untuk SMK 17 orang. Dengan tingginya persentase kelulusan siswa di Provinsi Riau memberikan kuota yang besar peminat kampus UIN Suska. Banyaknya mahasiswa baru yang berasal dari kota Pekanbaru memberikan peluang bisnis baru bagi pengusaha kos-kosan untuk mengembangkan usahanya. Banyaknya kos-kosan di sekitar kampus UIN memberikan kesulitan bagi para mahasiswa baru ataupun masyarakat Pekanbaru dalam memilih kos-kosan sesuai dengan keinginan dan keuangan yang ada.

Peta digital pada saat ini sudah menjadi salah satu aplikasi penting mencari lokasi dan membantu memberikan petunjuk arah untuk bepergian ke luar kota. Ada beragam jenis aplikasi ini yang bisa didapat baik secara berbayar atau gratis, misalnya saja Google Map. Google map menyediakan beberapa fitur, seperti menentukan arah mengemudi, mencari lokasi SPBU dan tempat singgah terdekat, hingga memantau lokasi kemacetan di beberapa ruas jalan utama.

Adapun penelitian sebelumnya oleh Yayan Riek Santara alumni Sekolah Tinggi Management Informatika dan Komputer AMIKOM Yogyakarta dengan judul "Aplikasi Peta Penunjuk Lokasi Object Wisata di Kabupaten Wonogiri Menggunakan Macromedia Director MX. Pada tugas akhir tersebut menggunakan aplikasi Macromedia Director MX, Macromedia Director MX 2004 adalah untuk membuat profile pribadi interaktif sederhana yang terdiri dari file

bitmap (jpg), animasi (gif) flash (swf) dan tombol standar. yang hasilnya berupa application / projector (exe) yang bisa dijalankan di komputer lain. Saat ini telah ada website yang menyediakan informasi mengenai rumah kost di internet seperti www.carikost.net. Pada sistem ini proses pencarian masih dilakukan dengan metode filter biasa. Pencarian dilakukan dengan cara memilih kriteria-kriteria sesuai kebutuhan tanpa memperdulikan tingkat kepentingan tiap fasilitas. Berbeda dengan sistem yang nantinya akan dibuat. Pengguna dapat menentukan tingkat kepentingan tiap kriteria saat melakukan pencarian.

Adapun Metode Perbandingan Eksponensial sebelumnya telah diteliti oleh Ezza Wilona Caulica, mahasiswi Politeknik Caltex Riau dengan judul penelitian Sistem pendukung keputusan penerimaan beasiswa pemerintah Provinsi Riau menggunakan Metode Perbandingan Eksponensial. Pada penelitian sebelumnya, kriteria-kriteria pada sistem yang diteliti oleh Ezza berbasis *Application Desktop* dengan bahasa pemrograman *Visual Basic.NET* karena yang menggunakan sistem tersebut hanya pegawai dari Kantor Pemerintah Provinsi Riau. Kemudian Metode Perbandingan Eksponensial juga diteliti oleh Dedi Rahma Wijaya, mahasiswa Manajemen Informatika Politeknik Telkom Bandung dengan judul penelitian “Pembanguna Sistem Pengambilan Keputusan untuk Pemilihan Proyek Pada CV. Garut Langit”. Pada penelitian tersebut Dedi Rahmana Wijaya Menggunakan Bahasa Pemrograman PHP dan database MySQL tanpa menggunakan framework *codeigniter*. Metode perbandingan eksponensial mempunyai keuntungan dalam mengurangi bias yang mungkin terjadi dalam analisis. Nilai skor yang menggambarkan urutan prioritas menjadi besar (fungsi eksponensial) ini mengakibatkan urutan prioritas alternatif keputusan lebih nyata (Marimin, 2009).

Oleh karena itu, penulis membuat suatu aplikasi pencarian rumah kost berbasis *website* yang bertujuan untuk memudahkan masyarakat dalam melakukan pencarian lokasi rumah kost menggunakan Metode Perbandingan Eksponensial. Dengan sistem ini, masyarakat tidak perlu lagi langsung untuk melakukan survey ke lokasi rumah kost. Cukup dengan duduk di depan komputer yang terhubung ke jaringan internet, mereka dapat mengetahui informasi rumah kost pada daerah tertentu.

Adapun manfaat dari penelitian ini membantu pengguna untuk mendapatkan informasi mengenai rumah kost. Dengan adanya aplikasi ini pengguna juga dapat mengetahui lokasi rumah kost tersebut melalui map dan memberikan kemudahan kepada pemilik rumah kost dalam hal promosi. Pemilik rumah kost tidak perlu lagi bersusah payah dalam mempromosikan rumah kostnya. Melalui aplikasi ini pemilik rumah kost cukup mendaftarkan informasi rumah kost mereka.

1.2 Rumusan Masalah

Perumusan masalah dalam tugas akhir ini adalah :

Bagaimana (sistem yang manual) dengan membangun suatu *web* aplikasi pencarian rumah kos dengan algoritma Metode Perbandingan Eksponensial.

1.3 Batasan Masalah

Adapun batasan masalah pada tugas akhir ini yaitu :

1. *Web* ini menggunakan Google Maps sebagai penyedia layanan pencarian kost.
2. Metode yang dilakukan menggunakan bobot berdasarkan harga dan beberapa fasilitas, diantaranya fasilitas kasur, lamari, meja belajar, wc, parker, petugas keamanan dan AC pada rumah kost.
3. Penelitian *Entry* data dilakukan dengan pendekatan *cloud computing*. Dimana data menjadi tanggung jawab pengguna aplikasi.
4. *Website* ini tidak menyertakan fitur *ecommerce* karena sistem ini lebih difokuskan pada penyediaan informasi tentang pencarian menggunakan metode perbandingan eksponensial

1.4 Tujuan

Adapun tujuan dari tugas akhir ini adalah :

1. Membangun sebuah aplikasi yang mampu memberikan informasi kos-kosan yang sesuai dengan kebutuhan mahasiswa dan mampu memberikan kemudahan para pemilik.

2. Menerapkan metode MPE dalam pencarian lokasi kos-kosan yang sesuai dengan kebutuhan para pencari kos-kosan.

1.5 Sistematika Penulisan

Sistematika penulisan dalam penyusunan laporan tugas akhir ini adalah sebagai berikut:

BAB I PENDAHULUAN

Bab ini menjelaskan dasar-dasar dari penulisan laporan tugas akhir, yang terdiri dari latar belakang, rumusan masalah, batasan masalah, tujuan, serta sistematika penulisan laporan tugas akhir.

BAB II LANDASAN TEORI

Bab ini membahas teori-teori yang berhubungan dengan topik penelitian, yang terdiri dari Pencarian Rumah Kos Menggunakan Algoritma Pencarian MPE Berbasis PHP Codeigniter 2.0 Framework dan AJAX.

BAB III METODOLOGI PENELITIAN

Bab ini membahas tentang metodologi yang digunakan dalam penelitian dan pengembangan perangkat lunak.

BAB IV ANALISA DAN PERANCANGAN

Bab ini membahas tentang hasil analisa, deskripsi sistem, karakteristik pengguna, deskripsi umum kebutuhan, deskripsi perancangan rinci dan perancangan antar muka sistem

BAB V IMPLEMENTASI DAN PENGUJIAN

Bab ini membahas implementasi dan pengujian yang dilakukan terhadap Pencarian Rumah Kos Menggunakan Algoritma Pencarian MPE Berbasis PHP Codeigniter 2.0 Framework

BAB VI PENUTUP

Bab ini berisi kesimpulan yang dihasilkan dari pembahasan tentang Pencarian Rumah Kos Menggunakan Algoritma Pencarian MPE Berbasis PHP Codeigniter 2.0 Framework dan AJAX. dan beberapa saran sebagai hasil akhir dari penelitian yang telah dilakukan.

BAB II

LANDASAN TEORI

2.1 Algoritma Metode Perbandingan Eksponensial

Algoritma Metode Perbandingan Eksponensial (MPE) merupakan salah satu metode untuk menentukan urutan prioritas alternatif keputusan dengan kriteria jamak dengan cara mengkuantifikasikan pendapat seseorang atau lebih dalam skala tertentu (Marimin, 2009).

Langkah-Langkah MPE:

1. Menyusun alternatif keputusan yang akan dipilih.
2. Menentukan kriteria atau perbandingan kriteria keputusan yang penting untuk dievaluasi.
3. Menentukan tingkat kepentingan dari setiap kriteria keputusan atau pertimbangan kriteria.
4. Melakukan penilaian terhadap semua alternatif pada setiap kriteria.
5. Menghitung total nilai setiap alternatif.
6. Menentukan urutan prioritas keputusan didasarkan pada skor atau nilai total masing-masing alternatif

Formulasi Perhitungan

$$\text{Total nilai } (TN_i) = \sum_{j=1}^m (RK_{ij})^{TKK_j}, 1 \leq i \leq n$$

Dimana :

1. TN_i merupakan total alternative ke-i.
2. RK_{ij} merupakan derajat kepentingan relatif ke-j pada pilihan keputusan i.
3. TKK_j merupakan derajat kepentingan criteria keputusan ke-j yang nilainya lebih besar dari 0. TKK_j merupakan bilangan bulat.
4. n merupakan jumlah alternatif pilihan keputusan.
5. m merupakan jumlah kriteria keputusan.

Penentuan tingkat kepentingan kriteria dilakukan dengan cara wawancara. Sedangkan penentuan skor alternatif pada kriteria tertentu dilakukan dengan memberi nilai setiap alternatif berdasarkan nilai kriterianya. Semakin besar nilai alternatif, semakin besar pula skor alternatif tersebut. Total skor masing-masing alternatif keputusan akan relatif berbeda secara nyata karena adanya fungsi eksponensial (Marimin, 2009).

Untuk skala tingkat kepentingan yang digunakan pada sistem ini yaitu dengan menggunakan skala likert. Skala likert yang digunakan pada sistem ini terdiri dari sembilan bobot nilai, yaitu sebagai berikut:

1. Sama sekali tidak penting
2. Sangat tidak penting
3. Tidak penting
4. Agak tidak penting
5. Netral
6. Agak penting
7. Penting
8. Sangat penting
9. Sangat setuju penting

Fasilitas yang di gunakan dalam tugas akhir ini adalah :

1. Kasur
2. Lemari
3. Meja Belajar
4. WC
5. Tepat Parkir
6. AC
7. Petugas Keamanan

Keuntungan menggunakan Metode MPE

Metode perbandingan eksponensial mempunyai keuntungan dalam mengurangi bias yang mungkin terjadi dalam analisis. Nilai skor yang menggambarkan urutan prioritas menjadi besar (fungsi eksponensial) ini mengakibatkan urutan prioritas alternatif keputusan lebih nyata.

2.2 Google Maps

Google Maps merupakan sebuah jasa peta *virtual* gratis yang disediakan oleh Google. Jasa ini dapat diakses secara online. Sejak diluncurkan pada tahun 2007, Google Maps menjadi salah satu pilihan para pengguna *internet* untuk melihat peta dunia secara *online*. Beberapa fitur yang ada di Google Maps, mampu menampilkan peta dunia, foto satelit, kepadatan jalan, topografi suatu tempat, foto satelit, topografi suatu tempat, serta fasilitas *street view*, atau pencitraan tingkat jalan. *Street view* ditempatkan di setiap perempatan jalan untuk menampilkan kondisi dan situasi jalan secara nyata.



Gambar 2.1 Tampilan Google Maps

Dengan menggunakan fasilitas Google Maps pengguna dapat mengetahui arah serta jarak perjalanan yang harus ditempuh untuk menuju ke lokasi yang dituju. Google Maps juga memberikan info tentang kepadatan jalan pada saat-saat tertentu. Selain itu, Google Maps juga mampu menampilkan foto satelit untuk mengenali ciri-ciri fisik suatu lokasi alam seperti gunung, laut dan hutan. Tak hanya itu, Google Maps juga dapat melihat dan bernavigasi dalam pencitraan tingkat jalan.

Google Maps menyediakan sebuah fasilitas untuk membuat peta sendiri. Dengan fasilitas tersebut, menambahkan daerah kedalam peta, serta memberikan arah perjalanan menuju lokasi sebelumnya. Google Maps ini dapat digunakan oleh

siapa saja yang membutuhkan informasi dengan cara langsung mengakses situs : <http://maps.google.com/>.

Lars and Jens Rasmussen mengembangkan Google Maps, sebagai sebuah aplikasi perangkat lunak untuk perusahaan Where Technologies. Google Inc, mengakuisisi perusahaan tersebut pada Oktober 2004, dan berubah menjadi aplikasi *web* Google Maps. Google mengumumkan *web* Google Maps untuk pertama kali di Google blog pada tanggal 8 Februari 2005. Pada awalnya tidak semua *browser* dapat mengakses halaman ini. Hanya Internet Explorer dan Mozilla yang dapat mengakses layanan ini. Pada tanggal 25 Februari 2005, Google memberikan kesempatan pada Opera dan Safari untuk mengakses layanan ini. Dan saat ini, hampir semua *browser* dapat mengakses halaman ini.

2.3 PHP (*Hypertext Preprocessor*)

PHP (*Hypertext Preprocessor*) yang merupakan bahasa pemrograman berbasis web memiliki kemampuan untuk memproses data dinamis. PHP dikatakan sebagai sebuah *server-side embedded script language* artinya sintaks-sintaks dan perintah yang kita berikan akan sepenuhnya dijalankan oleh *server* tetapi disertakan pada halaman HTML biasa. Aplikasi-aplikasi yang dibangun oleh PHP pada umumnya akan memberikan hasil pada *web browser*, tetapi prosesnya secara keseluruhan dijalankan di *server*. (PHP dan MySQL Untuk Pemula, MADCOMS).

Pada prinsipnya *server* akan bekerja apabila ada permintaan dari client. Dalam hal ini client menggunakan kode-kode PHP untuk mengirimkan permintaan ke *server*. Ketika menggunakan PHP sebagai *server-side embedded script language* maka *server* akan melakukan hal-hal sebagai berikut :

1. Membaca permintaan dari client / browser.
2. Mencari halaman / page di server.
3. Melakukan instruksi yang diberikan oleh PHP untuk melakukan modifikasi pada halaman / page.
4. Mengirim kembali halaman tersebut kepada client melalui internet atau intranet.

2.3.1 Client-side dan Server-side Programming

Perbedaan utama antara *server-side programming* dan *client-side programming* adalah tempat mengeksekusi skripnya. Pada *client-side programming*, skrip di eksekusi di *browser*, sedangkan pada *server-side programming*, skrip dieksekusi di *client (web browser)*.

Karakteristik *client-side programming* :

1. Kode program di download bersama dengan halaman *web*.
2. Bersifat interpreter dan diterjemahkan oleh *browser*.
3. Model eksekusinya simpel, dan skrip dapat dijadikan satu dengan HTML.

Karakteristik *server-side programming* bekerja pada :

1. Ada klien yang meminta *request*.
2. Eksekusi program dilakukan di *server*.
3. Mengirimkan hasil ke klien.

Keuntungan *server-side programming* :

1. *Cross-platform* : tidak tergantung pada *browser* tertentu karena program dieksekusi di *server*.
2. Optimasi dan pemeliharaan dilakukan di *server*.
3. Dapat dikembangkan menjadi aplikasi lain. *Server-side* tidak hanya terbatas dalam keamanan, tetapi dapat mengakses file dan *database*.
4. Menambah kekuatan *server*. Mesin *server* cenderung harus lebih *powerful* dengan beberapa tool.
5. Kode program aman : Client tidak dapat mengakses kode program.

2.4 Framework

Dalam bahasa Inggris, *framework* berarti kerangka kerja. *Framework* dapat diartikan sebagai sebuah tools atau alat yang berfungsi untuk membantu sebuah pekerjaan. Karena PHP adalah script untuk membuat website, maka framework PHP dalam kasus ini bisa diartikan sebagai alat yang berfungsi untuk mempermudah proses pembuatan sebuah website. Lalu apa bedanya dengan Content Management System (CMS), seperti WordPress, Joomla dan lain-lain.

2.5 Codeigniter

Codeigniter merupakan sebuah *framework* yang digunakan pada *web* yang dibuat dalam format PHP. Format yang dibuat ini nantinya dapat digunakan untuk membuat sistem aplikasi *web* yang kompleks. Codeigniter dapat mempercepat proses pembuatan *web*, karena semua *class* dan modul yang dibutuhkan sudah ada dan kita hanya tinggal menggunakannya kembali pada aplikasi *web* yang akan kita buat.

Framework secara umum yaitu sebuah susunan atau rangkaian kerja yang tetap dan dibuat sedemikian rupa yang kemudian dapat digunakan kembali dalam sebuah aktifitas kerja yang lain tapi tetap dalam satu area kerja dengan rangkaian kerja sebelumnya. *Framework* juga berarti kumpulan sebuah maupun banyak modul-modul dalam bentuk *class library* yang dapat kita gunakan lagi untuk membentuk sebuah aplikasi *web* yang lebih besar dengan memanfaatkan modul-modul *class library* yang ada (Kolang Wiswakarma, 2010).

Penulis berencana akan menggunakan bahasa pemrograman PHP. Namun demi terciptanya suatu sistem yang rapi, tidak akan cukup jika hanya menggunakan PHP murni. Penulis nantinya akan menggunakan *PHP Framework Codeigniter 2.0* yang memiliki konsep *MVC (Model, View, Controller)* bertujuan agar sistem menjadi lebih rapi jika dibandingkan PHP yang terstruktur.

2.5.1 Kelebihan Codeigniter

Codeigniter adalah sebuah *framework* berbasis PHP. Codeigniter sudah menyiapkan beberapa kumpulan *class-class* untuk membuat aplikasi *web*. Beberapa keunggulan *framework* Codeigniter diantaranya:

1. Open Source.

Codeigniter adalah *framework* yang gratis untuk digunakan dan dapat dikembangkan secara legal. Codeigniter berlisensi Apache/BSD-style open source yang memperbolehkan pengguna dari *framework* ini sesuai dengan keinginan. Ini adalah salah satu kelebihan yang paling nyata sangat menguntungkan user yang menggunakannya.

2. Ringan dijalankan pada semua platform.
Codeigniter ringan dijalankan pada berbagai *platform*. Sistem utama dari Codeigniter hanya memerlukan sedikit ruang pada *server* untuk membentuk sebuah *class library*. Ini yang membedakan Codeigniter dengan *framework* sejenisnya. *Class library* tambahan hanya digunakan ketika diminta oleh pengguna, sehingga hanya dipakai ketika diperlukan. Ini yang menyebabkan Codeigniter sangat ringan jika dijalankan.
3. Menggunakan MVC.
Model MVC adalah singkatan untuk “Model–View–Controller”. Model *framework* ini dapat mempermudah perbedaan antara tampilan dan program. Ini secara khusus sangat baik untuk suatu proyek dimana *designer* bekerja dengan file template yang akan memperkecil banyaknya kode disetiap template karena sudah dipisahkan dengan baik.
4. Dikemas dalam sebuah framework yang lengkap.
Codeigniter dikemas dalam kumpulan *class* dalam sebuah *library* yang cukup lengkap yang dapat menunjang proses pembuatan *web*, seperti mengakses database, mengirim *email*, validasi form data, membuat *session*, memanipulasi gambar dan banyak lagi.
5. Dokumentasi yang lengkap.
Codeigniter mempunyai dokumentasi yang sangat rapi dan sangat jelas. Sehingga dengan adanya dokumentasi ini akan banyak membantu para programmer pemula yang ingin mempelajari *framework* Codeigniter.
6. Komunitas pengguna yang besar.
Codeigniter merupakan sebuah *framework* yang mempunyai komunitas yang sangat besar. Dengan adanya komunitas ini juga dimungkinkan untuk sharing ilmu tentang Codeigniter dari satu pengguna ke pengguna Codeigniter lainnya.

2.5.2 Fitur Codeigniter

Beberapa fitur dari *framework* Codeigniter yang handal dalam membuat aplikasi *web* yang kompleks, diantaranya:

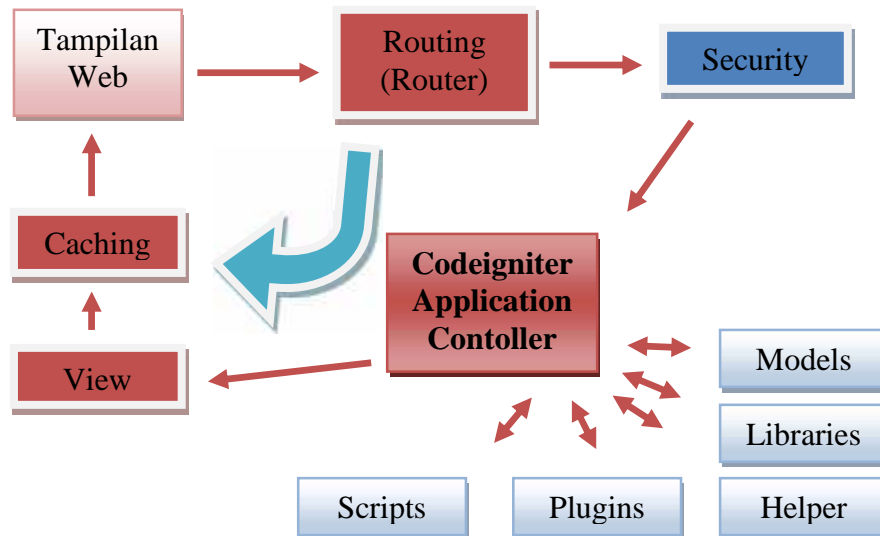
1. Sistem berbasis Model-View-Controller (MVC).
2. Kompetibel dengan PHP 4.
3. Sangat ringan sehingga menghemat banyak *space*.
4. *Class* database lengkap yang *support* untuk berbagai *platform*.
5. Penyimpanan data aktif pada *database*.
6. Validasi *form* dan *data*.
7. Keamanan dan *XSS filter*.
8. Manajemen *session*.
9. *Class* kirim email, *support* untuk *attachment*, HTML/Text email, protocol yang bervariasi (sendmail, SMTP dan Mail) dan banyak lagi.
10. *Library* untuk memanipulasi gambar seperti memotong, menskalasi ukuran gambar, memutar dan lain-lain.
11. *Class* untuk mengupload file.
12. *Class* FTP.
13. Pengurutan halaman/*paging*.
14. Enkripsi data.
15. Testing performa kode (*Code benchmarking*).
16. Menyimpan file halaman *web/webpage caching*. *Caching* merupakan suatu aktifitas untuk menyimpan file yang pernah diakses dalam *browser* kemudian disimpan di computer pengguna. Jika suatu saat halaman *web* tersebut dibuka kembali, *cache* ini akan muncul sehingga halaman *web* akan lebih cepat dibuka dan secara otomatis akan menghemat *bandwidth* yang keluar.
17. *Error log*.
18. *Scaffolding*. *Scaffolding* merupakan suatu fitur aplikasi yang berfungsi sebagai tampilan virtual dari sebuah database dimana kita dapat membuat, mengedit, melihat dan menghapus sebuah data pada database.
19. *Class* untuk tanggal kalender.
20. *Class* untuk melakukan *zip encoding*.
21. *Class template engine*.

22. *Class traceback*. *Traceback* merupakan suatu metode untuk mengetahui halaman *web* lain yang mempunyai link pada halaman *web* kita.
23. *Class* untuk unit testing.
24. *Search-engine Friendly URLs*.
25. *URL routing* yang fleksibel.
26. *Library* yang besar untuk fungsi helper.

2.5.3 Sistem Codeigniter

Pada gambar 2.1 terlihat susunan sistem yang dimiliki *framework* Codeigniter. Tampilan *web* adalah bagian akhir dimana halaman *web* tersebut terlihat pada *browser*. Tahapannya adalah sebagai berikut:

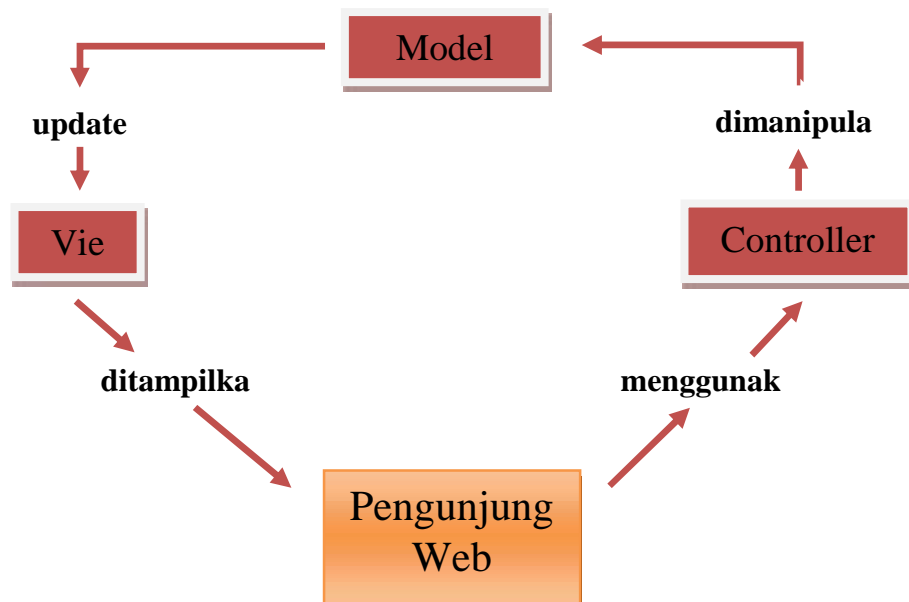
1. File Tampilan *Web* bertindak sebagai kontroler utama, yang berfungsi sebagai penampilan halaman *web* yang memuat sumber kode utama yang berfungsi untuk menjalankan Codeigniter.
2. Bagian *Routing* berfungsi untuk menerima permintaan HTTP untuk menentukan langkah yang selanjutnya diambil.
3. Jika file *Cache* ada, maka sistem langsung akan menuju bagian *Caching* yang kemudian akan ditampilkan pada halaman *web*, tanpa melalui sistem Codeigniter yang normal.
4. Pada bagian *Security*, semua permintaan HTTP dan form yang dikirim oleh pengguna akan disaring untuk pengamanan.
5. Kemudian data akan menuju bagian *Application Controller* yang akan dimunculkan model, *library*, *script helper* dan *plugins* dan semua sumber yang dibutuhkan untuk melengkapi permintaan tersebut.
6. Tampilan akhir (*View*) akan dikirim ke dalam *browser* untuk dilihat pada halaman *web*. Jika *Caching* diterima maka tampilan akhir akan menjalani *caching* terlebih dahulu sehingga permintaan yang sama akan dapat berjalan dengan lebih cepat pada *browser*.



Gambar 2.2 Struktur kerja Codeigniter pada halaman web

2.5.4 MVC (*Model – View – Controller*)

Framework Codeigniter adalah *framework* yang berbasis Model-View-Controller (MVC). MVC adalah sebuah *software* yang memisahkan antara aplikasi logika dengan presentasi pada halaman web (Komang Wiswakarma, 2010). Sehingga hal ini akan menyebabkan halaman web akan mengandung kode yang sedikit karena sudah terjadi pemisahan antara tampilan dan pemrograman. Ilustrasinya dapat dilihat pada gambar 4.3.



Gambar 2.3 MVC pada halaman web

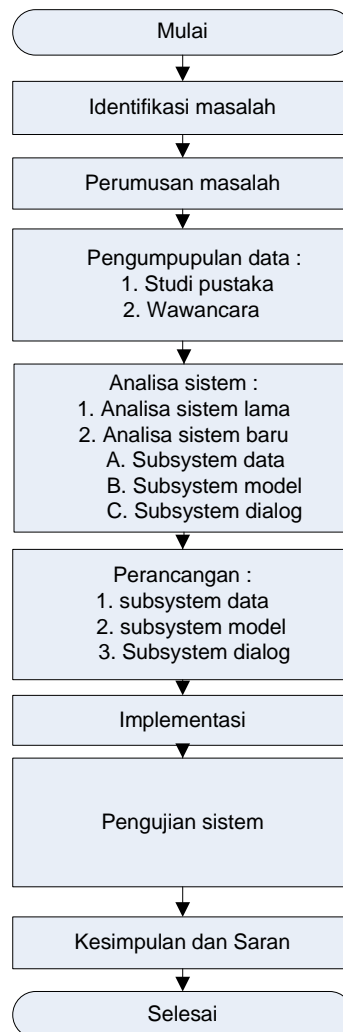
1. Model merupakan struktur data. Secara spesifik class model akan mengandung fungsi kode yang akan membantu dalam segala proses yang berhubungan dengan database seperti memasukkan, mengedit, mendapatkan dan menghapus data dalam sebuah database.
2. *View* merupakan informasi yang disampaikan ke pengguna. Sebuah *view* biasanya berupa halaman *web*, tetapi dalam Codeigniter, sebuah *view* juga bisa berupa sebuah *fragmen* halaman seperti *header* dan *footer*. *View* juga bisa berupa halaman RSS atau jenis halaman *view* yang lain.
3. Controller merupakan sebuah perantara antara Model dan *View* dan semua sumber yang dibutuhkan untuk memproses permintaan HTTP dan dalam membuat halaman *web*.

Dalam menggunakan *framework* Codeigniter, kehadiran bagian Model tidak sering dibutuhkan. *Web* dapat dibangun hanya dengan bagian *Controller* dan *View* atau dapat juga membangun fungsi sendiri secara minimal menggunakan dua bagian tersebut. Codeigniter memperbolehkan untuk membangun aplikasi sesuai kemampuan dan keinginan programmer (Komang Wiswakarma, 2010).

BAB III

METODOLOGI PENELITIAN

Pada bab ini akan dipaparkan tentang langkah-langkah yang digunakan untuk membahas permasalahan yang diambil dalam penelitian atau yang disebut dengan metodologi penelitian. Metodologi penelitian tugas akhir ini dapat digambarkan sebagai berikut :



Gambar 3.1 *Flowchart* Metodologi Penelitian

3.1 Identifikasi Masalah

Identifikasi masalah merupakan sekelompok aspek yang berada disekitar masalah utama yang dapat diteliti untuk menjawab permasalahan utama. Adapun permasalahan yang dapat diidentifikasi untuk pelaksanaan tugas akhir ini adalah perhitungan yang masih manual.

3.2 Perumusan Masalah

Berdasarkan permasalahan yang telah diidentifikasi, maka dapat dirumuskan bahwa bagaimana merancang dan membangun suatu Pencarian Rumah Kos menggunakan algoritma pencarian MPE berbasis PHP Codeigniter 2.0 framework dan AJAX.

3.3 Pengumpulan Data

3.3.1 Studi Pustaka

Studi pustaka dilakukan untuk mencari dan mempelajari serta mengumpulkan seluruh informasi yang terkait dan mendukung pelaksanaan penelitian pada tugas akhir ini. Studi pustaka ini membahas tentang Pencarian Rumah Kos menggunakan algoritma pencarian MPE berbasis PHP Codeigniter 2.0 framework dan AJAX. Sumber kepustakaan diambil dari karya ilmiah yang berasal dari buku-buku maupun internet. Karya ilmiah yang dimaksud adalah berupa tulisan ilmiah yang berbentuk artikel, buku, *e-book* (buku elektronik), dan lain-lain.

3.4 Analisa Sistem

3.4.1 Analisa Penelitian sebelumnya

Metode Perbandingan Eksponensial sebelumnya telah diteliti oleh Ezza Wilona Caulica (Sistem pendukung keputusan penerimaan beasiswa pemerintah Provinsi Riau menggunakan Metode Perbandingan Eksponensial). Pada penelitian sebelumnya diteliti oleh Ezza berbasis *Application Desktop* dengan bahasa pemrograman *Visual Basic.NET* karena yang menggunakan sistem

tersebut hanya pegawai dari Kantor Pemerintah Provinsi Riau. Sedangkan pada sistem yang akan dibangun pada proyek akhir ini berbasis *Web Application* dengan bahasa pemrograman PHP dan *Codeigniter 2.0 Framework* karena pengguna sistem ini ditujukan untuk semua masyarakat di Pekanbaru yang membutuhkan rumah kost.

Tabel 3.1 Perbandingan aplikasi yang akan di bangun terhadap penelitian sebelumnya

| Objek | Penelitian Sebelumnya | Penelitian Sekarang |
|--------------------|----------------------------------|----------------------------------|
| Jumlah Kriteria | Statis | Dinamis |
| Metode | Metode Perbandingan Eksponensial | Metode Perbandingan Eksponensial |
| Teknologi | Application Desktop | Web Based Application |
| Bahasa Pemrograman | Visual Basic.NET | PHP |
| <i>Framework</i> | Tidak Ada | Codeigniter 2.0 |
| <i>Database</i> | MySQL | MySQL |
| GIS | Tidak Ada | Google Maps API |

Saat ini telah ada *website* yang menyediakan informasi mengenai rumah kost di internet seperti www.carikost.net. Pada sistem ini proses pencarian masih dilakukan dengan metode *filter* biasa. Pencarian dilakukan dengan cara memilih kriteria-kriteria sesuai kebutuhan tanpa memperdulikan tingkat kepentingan tiap fasilitas. Berbeda dengan sistem yang nantinya akan dibuat. Pengguna dapat menentukan tingkat kepentingan tiap kriteria saat melakukan pencarian.

Tabel 3.2 Perbandingan Sistem Pencarian Kost terhadap Penelitian Sebelumnya

| Objek | Penelitian Sebelumnya | Penelitian Sekarang |
|--------------------|------------------------------|----------------------------|
| Metode | Filtering | MPE |
| Teknologi | Web Based Application | Web Based Application |
| Bahasa Pemrograman | PHP | PHP |
| GIS | Tidak Ada | Google Maps API |

3.4.2 Analisa Sistem Baru

Analisa sistem baru dilakukan untuk menyusun langkah - langkah dalam mengidentifikasi permasalahan - permasalahan yang akan terjadi pada sistem yang akan dibangun. Serta kebutuhan - kebutuhan apa saja yang diinginkan untuk mengatasi permasalahan yang ada pada sistem nantinya. Dalam analisa sistem terdiri atas beberapa subsistem, subsistem data, subsistem model, dan subsistem dialog

3.5 Perancangan

3.5.1 Subsistem Data

Tahap perancangan subsistem data merupakan hasil dari analisa data yaitu ERD, yang selanjutnya akan dibuat suatu perancangan tabel secara utuh dan lengkap dengan berbagai komponennya.

3.5.2 Subsistem Model

Perancangan model merupakan hasil dari analisa model yaitu metode yang digunakan dalam pembuatan aplikasi tersebut. Pada subsistem ini akan dibuat suatu desain model system berupa *Site Map*, *Flowchart*, *Use Case Diagram*, *Class Diagram*, *Sequence Diagram*, *Entity Relational Diagram (ERD)*, dan *Tabel* dari proses Pencarian Rumah Kos menggunakan algoritma pencarian MPE berbasis PHP Codeigniter 2.0 framework dan AJAX

3.5.3 Subsistem Dialog

Perancangan subsistem dialog akan menghasilkan sebuah perancangan struktur menu aplikasi dan desain *User Interface* pada aplikasi, yang diperoleh dari analisa subsistem dialog atau implementasi dari analisa DFD.

3.6 Implementasi

Implementasi sistem merupakan hasil dari desain sistem yang telah dirancang kemudian diimplementasikan pada sebuah program komputer. Bahasa pemrograman yang digunakan adalah PHP.

3.7 Pengujian

Tahap pengujian diperlukan untuk menjadi ukuran bahwa sistem dapat dijalankan sesuai dengan tujuan, yang akan dilakukan dengan cara sebagai berikut:

Setelah sistem ini selesai dibuat akan dilakukan pengujian menjalankan fungsi aplikasi yang telah dibangun sesuai perancangan, untuk mengetahui apakah sistem sudah mampu memberikan hasil pencarian rumah kos.

Pengujian menggunakan *balckbox* yaitu penguji yang dilakukan terhadap *interface* perangkat lunak, pengujian ini dilakukan untuk mengetahui apakah aplikasi sudah sesuai dengan yang diharapkan dan menghasilkan output yang tepat dan berjalan dengan baik.

3.8 Kesimpulan dan Saran

Kesimpulan ini merupakan kesimpulan dari suatu pembahasan untuk menjawab pertanyaan-pertanyaan yang dikemukakan pada masalah dan tujuan serta saran-saran yang dikemukakan.

BAB IV

ANALISA DAN PERANCANGAN

Dalam membuat suatu aplikasi pencarian rumah kos berbasis web, terlebih dahulu dilakukan analisa. Analisa dilakukan untuk memahami persoalan atau identifikasi masalah sebelum melakukan tahap perancangan. Sementara pada perancangan, yang dibuat harus memiliki kesesuaian dengan analisa sistem yang sebelumnya telah dilakukan.

4.1 Deskripsi Umum Sistem

Aplikasi yang akan penulis bangun pada tugas akhir ini adalah pencarian rumah kos berbasis web menggunakan metode perbandingan eksponensial. Pada aplikasi tersebut memiliki tiga hak akses, hak akses tersebut adalah :

1. Hak Akses Member

Sebelum memiliki hak akses member, terlebih dahulu member mesti melakukan registrasi agar memiliki akses untuk melakukan pengolahan data rumah kos yang dimiliki, member dapat melakukan pengolahan data seperti melakukan pemilihan lokasi rumah kos dan menampilkan data fasilitas yang dimiliki oleh rumah kos.

2. Hak Akses Admin

Pada aplikasi tugas akhir ini hak akses tertinggi dimiliki oleh admin. Admin mengelola member yang mendaftar, admin dapat menonaktifkan dan mengaktifkan member serta dapat melakukan pengolahan data fasilitas.

3. Hak akses *Guest* (masyarakat)

Masyarakat dapat menggunakan aplikasi ini tanpa harus melakukan login. Masyarakat dapat melihat kos-kosan yang berada di sekitar wilayah kampus UIN Suska beserta deskripsi dari fasilitas yang dimiliki oleh masing-masing kosan

4.2 Analisa Sistem

Analisa sistem dibagi dalam dua bagian pembahasan yaitu analisa pada sistem yang berjalan dan analisa pada sistem yang akan dibangun. Pada analisa sistem yang akan dibangun akan dibahas juga analisa aplikasi pencarian rumah kos berbasis web dan analisa fungsional sistem yang terdiri dari analisa berupa *Site Map*, *Flowchart*, *Use Case Diagram*, *Class Diagram*, *Sequence Diagram*, *Entity Relational Diagram (ERD)*, *Tabel*

4.2.1 Analisa sistem yang berjalan

Analisa pada sistem yang berjalan :

1. Admin memasukkan batasan fasilitas rumah kos serta mengelola user
2. Member setelah di aktivasi oleh admin dapat melakukan login ke system sehingg mendapatkan hak akses member. Setelah login sebagai member, member dapat melakukan promosi terhadap rumah kos yang dimiliki dengan cara memberikan informasi rumah kos berupa data fasilitas-fasilitas yang dimiliki dan memberikan lokasi rumah kos dalam bentuk peta *google*.
3. Masyarakat melakukan pencarian terhadap rumh kos yang akan di cari sesuai dengan kriteria yang dibutuhkan.

4.2.2 Analisa yang akan dibangun

Aplikasi yang akan dibangun adalah bagaimana masyarakat dapat menerima hasil pencarian rumah kos sesuai dengan kriteria yang diinginkan. Serta memberikan kemudahan bagi pemilik kos untuk mempromosikan kos.

Pembahasan pada analisa yang akan dibangun ini meliputi analisa aplikasi pada sisi administrator dan pada sisi member, selain itu juga dibahas analisa pada sisi masyarakat berupa hasil kos-kosan yng sesuai dengan kriteria yang diinginkan.

4.3 Analisa fungsional sistem

Analisa fungsional akan dilakukan dengan menggambarkan bagaimana masukan diproses oleh aplikasi menjadi keluaran yang diharapkan oleh pengguna sistem. Adapun cakupan analisa fungsional disini membahas, yaitu :

1. Perancangan *Site Map*
2. Perancangan *Flowchart*
3. Perancangan *Use Case Diagram*
4. Perancangan *Class Diagram*
5. Perancangan *Sequence Diagram*
6. Perancangan *Entity Relational Diagram (ERD)*
7. Perancangan *Tabel*

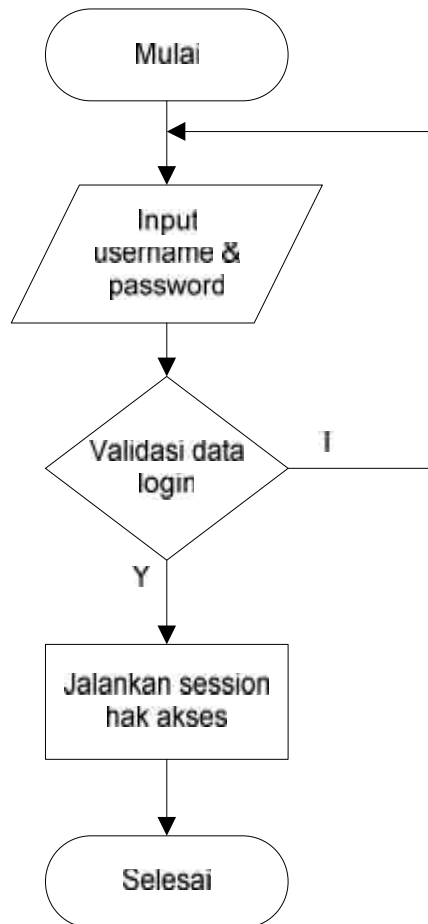
Site Map merupakan salah satu cara sederhana untuk menginformasikan kepada pengguna tentang halaman-halaman pada suatu situs. Pada Gambar 4.1 menampilkan bagaimana bentuk *site map* sistem yang akan dibuat.

Gambar 4.1 *Site Map* Aplikasi pencarian rumah kos

Flowchart merupakan alur proses dari suatu sistem. Berikut ini merupakan beberapa proses yang ada pada sistem yang akan dibuat.

Gambar 4.2 merupakan gambaran alur proses login pada *user*, yaitu *member* dan *admin*. Terlebih dahulu *admin* ataupun *member* memasukkan *username* dan *password* yang selanjutnya data yang dimasukkan tersebut akan

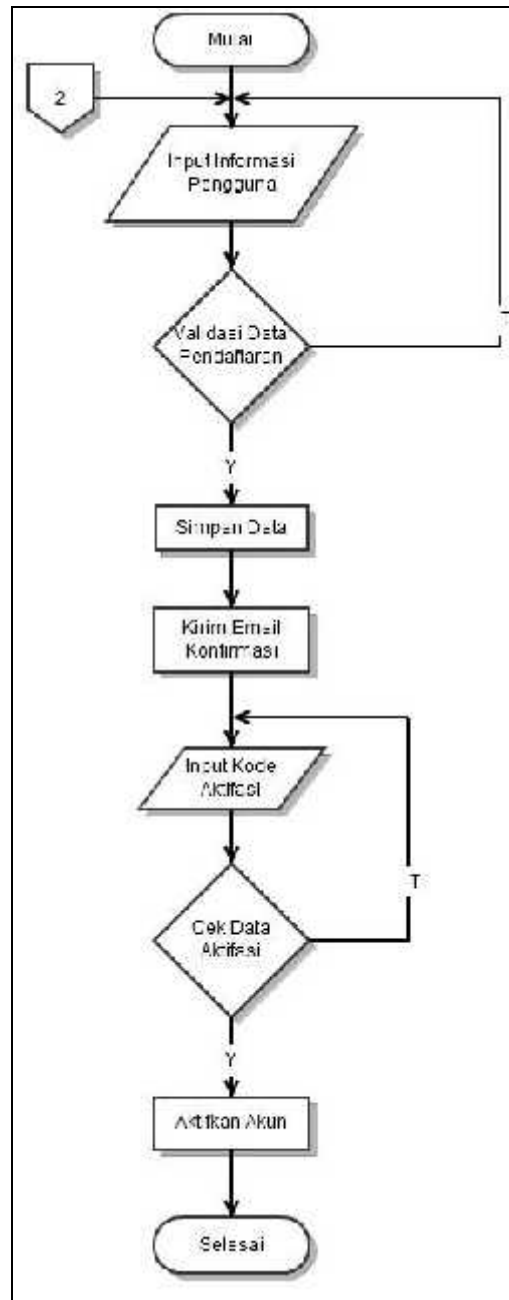
dikirim ke *database user* untuk memeriksa kebenaran data yang dimasukkan tersebut. Apabila login berhasil maka sistem akan menjalankan *session login* sesuai dengan hak akses yang dimiliki si pengguna



Gambar 4.2 *Flowchart login user*

4.3.3.2 Flowchart Registrasi

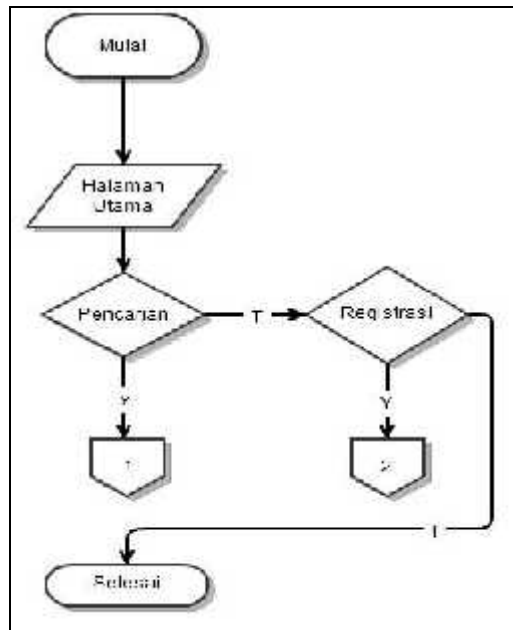
Gambar 4.3 merupakan gambaran alur dalam proses pendaftaran. Pendaftaran ini ditujukan pada pengguna yang ingin mendaftarkan rumah kostnya. Pertama, pendaftar harus mengisi informasi yang dibutuhkan oleh sistem yang kemudian akan diproses. Kemudian sistem akan mengirim *email* yang berisikan *link* yang mengarah pada halaman aktivasi akun.



Gambar 4.3 *Flowchart* registrasi

4.3.3.3 *Flowchart* guest

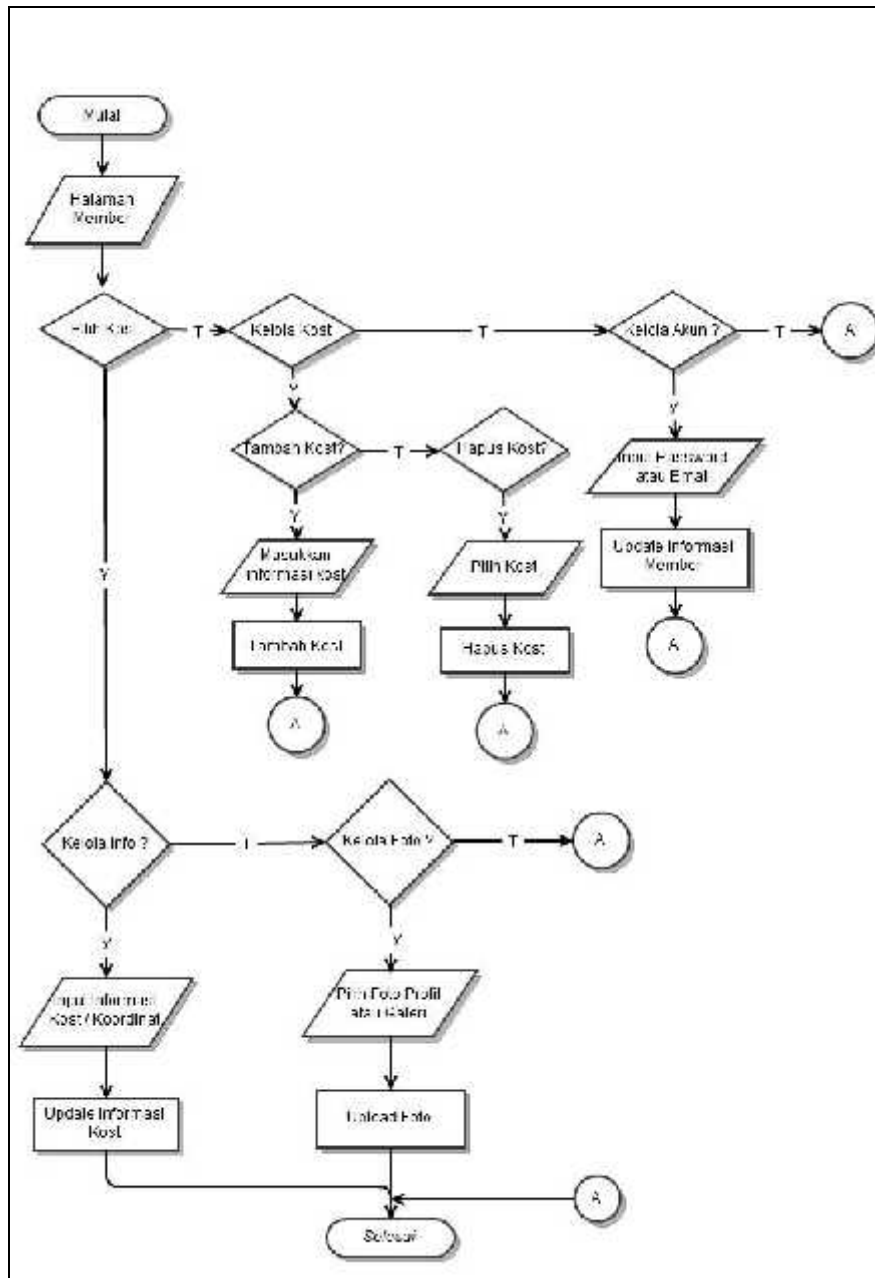
Gambar 4.4 merupakan *flowchart* proses yang dapat dilakukan oleh *guest*. *Guest* dapat melakukan pencarian dan registrasi.



Gambar 4.4 *Flowchart* guest

4.3.3.4 *Flowchart* Member

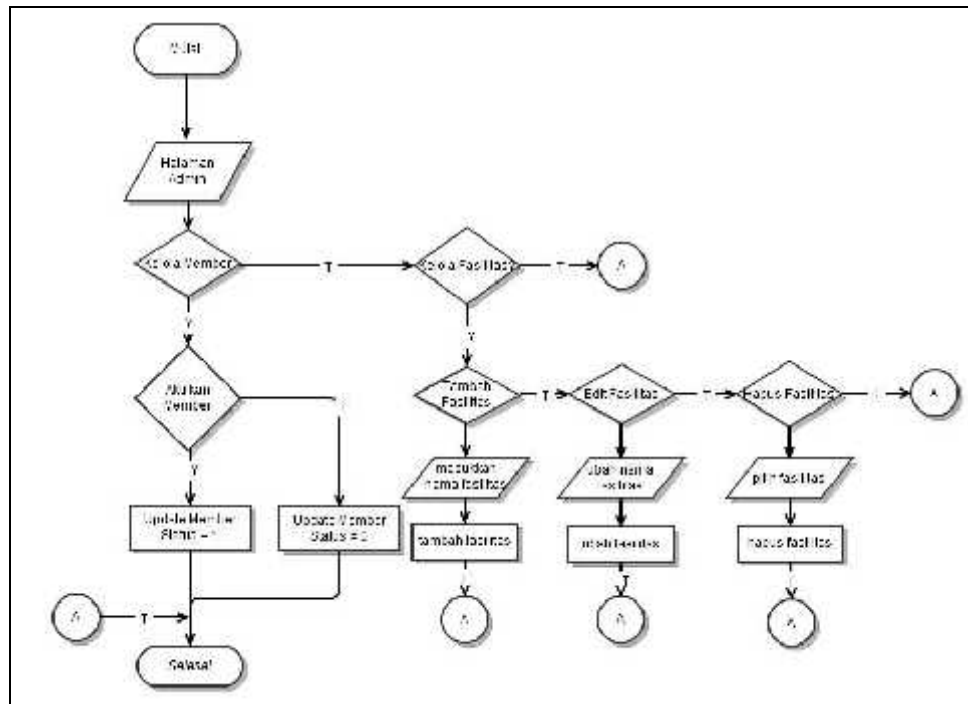
Gambar 4.5 merupakan *flowchart* proses yang dapat dilakukan oleh *member* setelah melewati proses *login*. Terlebih dahulu *member* menentukan apakah akan mengelola rumah kost, mengelola akun, atau langsung memilih rumah kost. Untuk mengubah informasi rumah kost terlebih dahulu *member* harus terlebih dahulu memilih rumah kost yang akan diubah. Begitu juga dengan kelola foto dan pencarian. Foto yang diunggah akan dimasukkan kedalam direktori server sementara *url* direktorinya disimpan pada *database*.



Gambar 4.5 Flowchart member

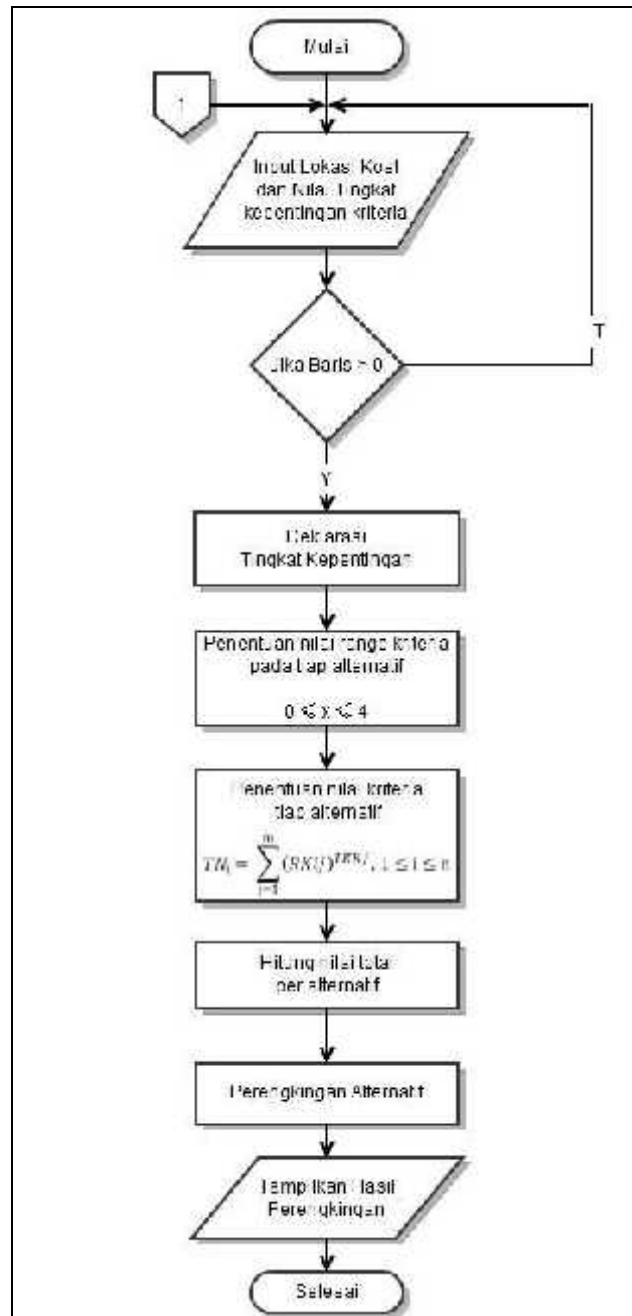
4.3.3.5 Flowchart admin

Gambar 3.6 merupakan *flowchart* proses yang dapat dilakukan oleh admin setelah melewati proses login. Admin dapat mengubah status *member* menjadi aktif atau tidak aktif. Admin juga bisa menambah, mengubah, atau menghapus fasilitas.



4.3.3.6 Flowchart Pencarian Rumah Kost

Gambar 4.7 merupakan gambaran alur proses pencarian yang dapat dilakukan oleh *member*, admin, dan *guest*. Terlebih dahulu pengguna memasukkan lokasi rumah kost dan nilai tingkat kepentingan dari tiap kriteria yang berguna untuk perhitungan MPE. Kemudian sistem mengambil data dari *database* yang selanjutnya dilakukan perhitungan sesuai dengan data yang telah dimasukkan sebelumnya.

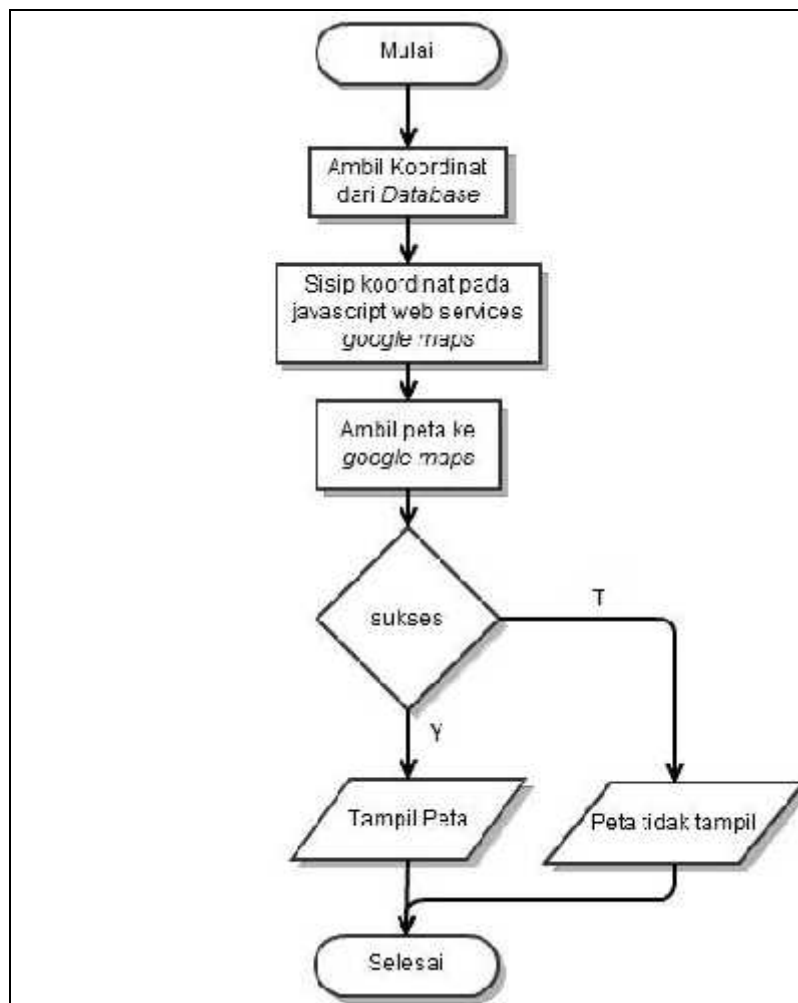


Gambar 4.7 Flowchart pencarian rumah kost

Pertama sistem mendeklarasi nilai tingkat kepentingan berdasarkan nilai yang telah dimasukkan oleh pengguna pada halaman pencarian sebelumnya. Kemudian sistem melakukan proses perhitungan berdasarkan nilai range kriteria terhadap alternatif-alternatif yang ada. Pada sistem ini satu rumah kost merupakan satu alternatif.

4.3.3.7 Flowchart menampilkan map dari Google Maps Web Services

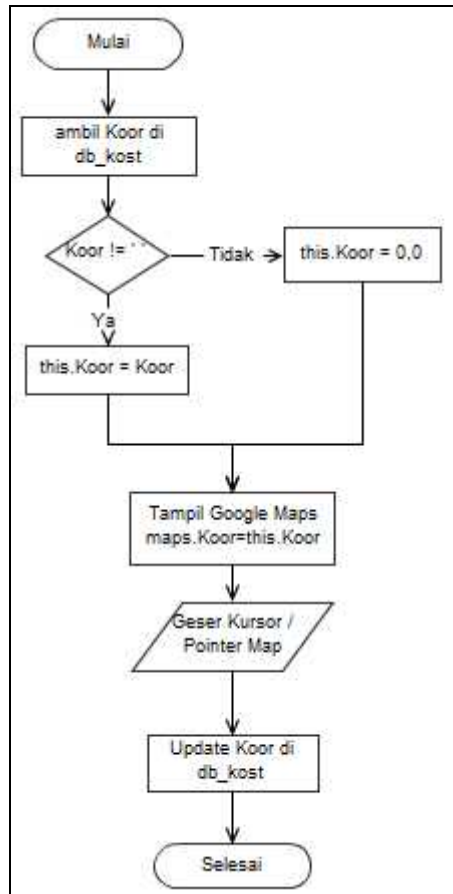
Gambar 4.8 merupakan *flowchart* proses menampilkan map dengan menggunakan fitur *web services* dari *Google Maps*. Sistem terlebih dahulu mengambil koordinat yang terdapat pada *database server*. Koordinat yang telah diambil tersebut kemudian disisipkan pada *javascript* yang telah disediakan *Google Maps* untuk menampilkan *map*. Apabila proses pengambilan peta sukses, maka peta akan ditampilkan.



Gambar 4.8 Flowchart proses menampilkan map

4.3.3.8 Flowchart Penentuan Koordinat Rumah Kost

Gambar 3.9 di bawah ini merupakan *flowchart* untuk menentukan koordinat suatu rumah kost menggunakan *Google Maps API*. Koordinat diambil dari *database* dan disimpan pada *database*.



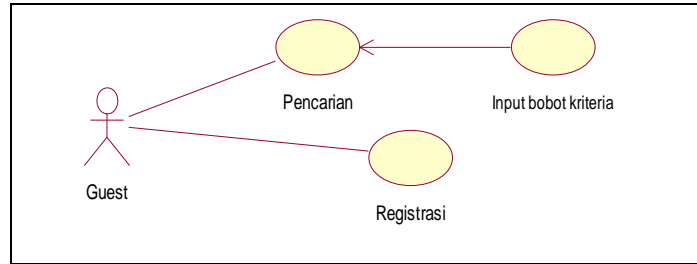
Gambar 4.9 *Flowchart* Penentuan Koordinat Rumah Kost

4.4 *Use Case Diagram*

Use Case Diagram merupakan gambaran fungsionalitas pada suatu sistem. Berikut ini adalah *Use Case Diagram* yang akan dibuat pada sistem ini.

4.4.1 *Use Case Diagram guest*

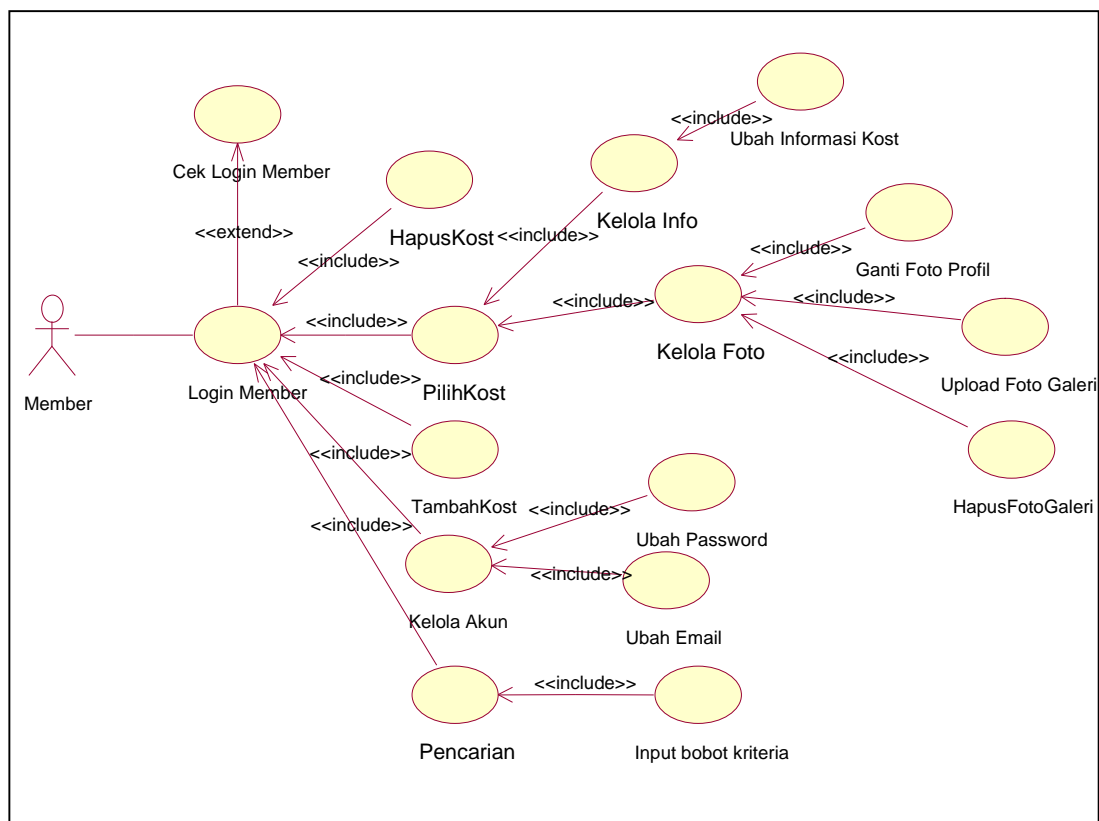
Gambar 4.10 merupakan gambaran *Use Case Diagram* pada *guest*. Pada umumnya *guest* dapat melakukan pencarian dan *registrasi*. Untuk pencarian, *guest* dapat menentukan nilai tingkat kepentingan masing-masing kriteria yang ada.



Gambar 4.10 Use Case Diagram guest

4.4.2 Use Case Member

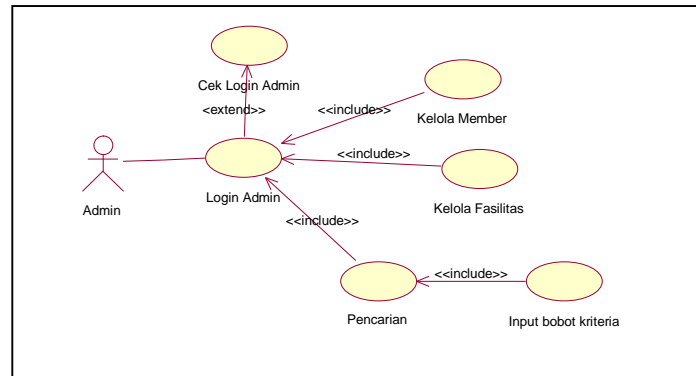
Gambar 4.11 merupakan gambaran *Use Case Diagram* pada *member*. Pada sistem ini, *member* merupakan pemilik rumah kost. *Member* dapat memiliki lebih dari satu rumah kost. *Member* juga dapat menghapus rumah kost yang sudah ada. Untuk mengelola suatu rumah kost, *member* terlebih dahulu harus memilih rumah kost yang terdaftar pada *member*. Setelah melakukan pemilihan rumah kost, *member* dapat mengelola info dan foto.



Gambar 4.11 Use Case Diagram Member

4.4.3 Use Case admin

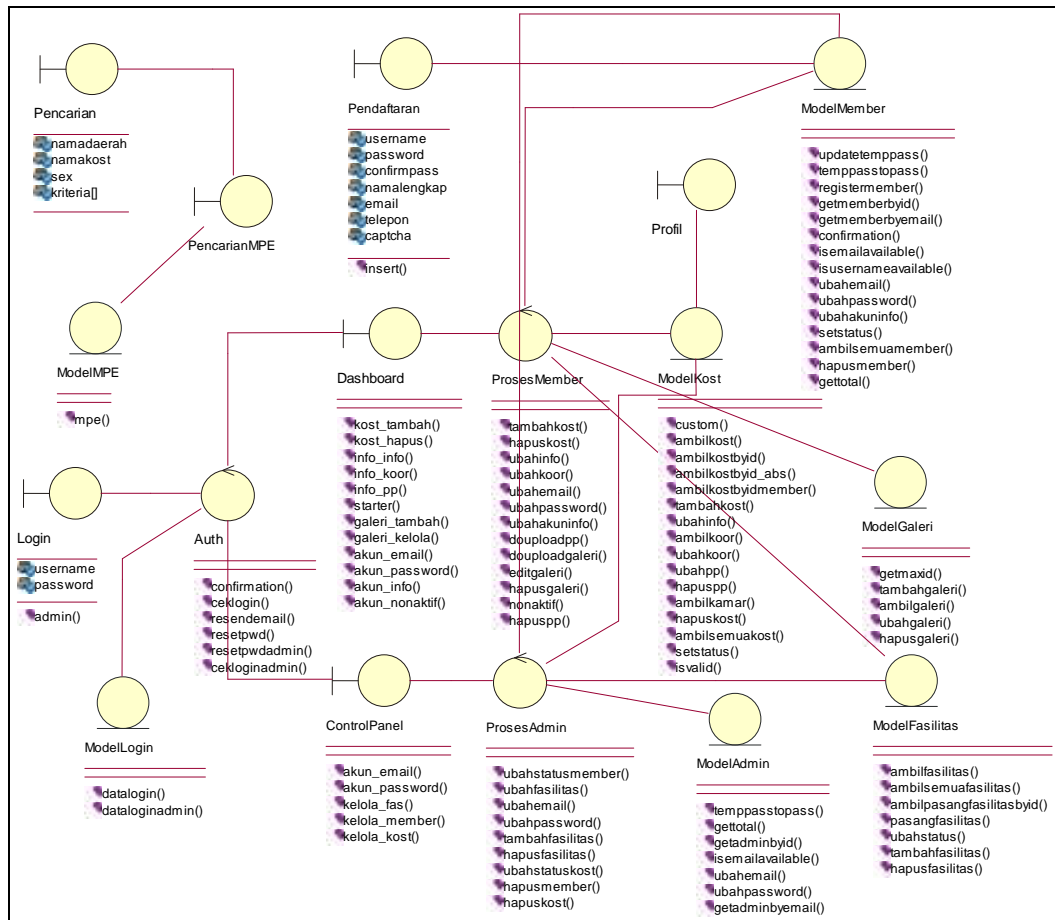
Gambar 4.12 merupakan gambaran *Use Case* pada admin. Pada sistem ini, admin merupakan pengelola *web*. Admin dapat mengelola *member* yang terdaftar pada sistem. Admin juga dapat mengubah fasilitas yang nantinya digunakan untuk kriteria pencarian.



Gambar 4.12 Use Case Diagram admin

4.5 Class Diagram

Gambar 4.13 berikut merupakan *Class Diagram* dari sistem yang akan dibuat. Secara garis besar gambar tersebut menjelaskan alur dari proses yang nantinya akan diimplementasikan pada sistem. Pada implementasinya nanti masih terdapat kemungkinan adanya perubahan proses dengan apa yang ada pada *Class Diagram* dibawah ini.

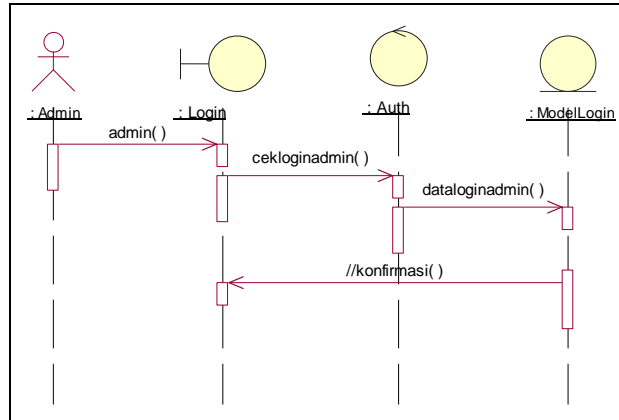


4.6 Sequence Diagram

Sequence Diagram merupakan gambaran interaksi antar objek dan mengindikasikan komunikasi diantara obyek-obyek tersebut. Berikut ini adalah gambaran *Sequence Diagram* pada sistem yang akan dibuat.

4.6.1 Sequence Diagram admin login

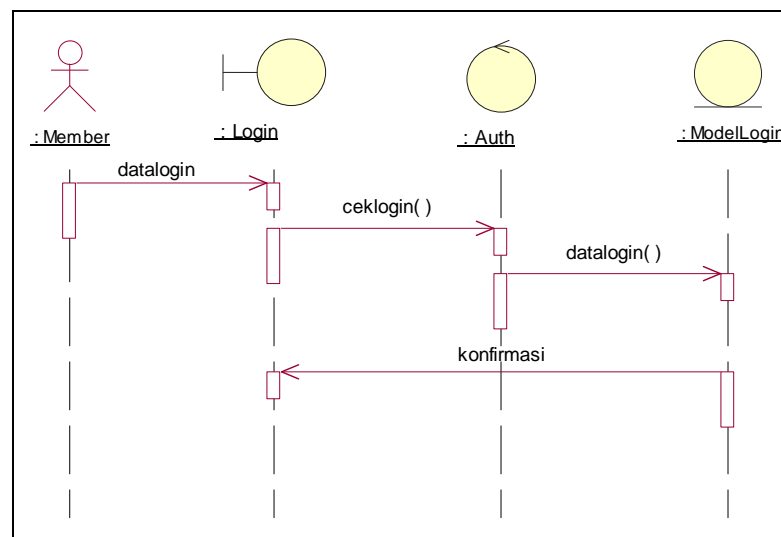
Pada Gambar 4.14 di bawah ini admin terlebih dahulu mengisi data login yang nantinya akan dilakukan pengecekan ke *database*.



Gambar 4.14 *Sequence Diagram admin login*

4.6.2 *Sequence Diagram member login*

Pada Gambar 4.15 di bawah ini mirip halnya dengan *Sequence Diagram* admin *Login*. Yang membedakan hanyalah aktor yang melakukannya dan *session* yang akan dijalankan ketika sukses.

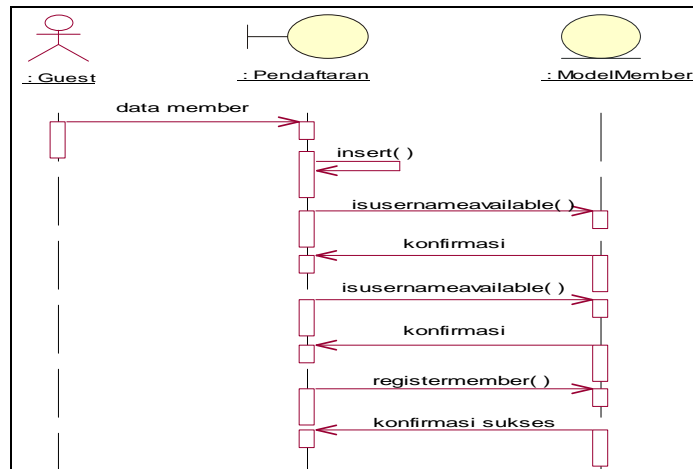


Gambar 4.15 *Sequence Diagram member login*

4.6.3 *Sequence Diagram Pendaftaran*

Pada Gambar 4.16 di bawah ini menjelaskan bagaimana *guest* melakukan pendaftaran. *Guest* terlebih dahulu mengisi *form* yang disediakan pada halaman registrasi atau pendaftara. Sebelum data diterima, sistem terlebih dahulu melakukan pengecekan ke *database* apakah *username* yang dimasukkan telah

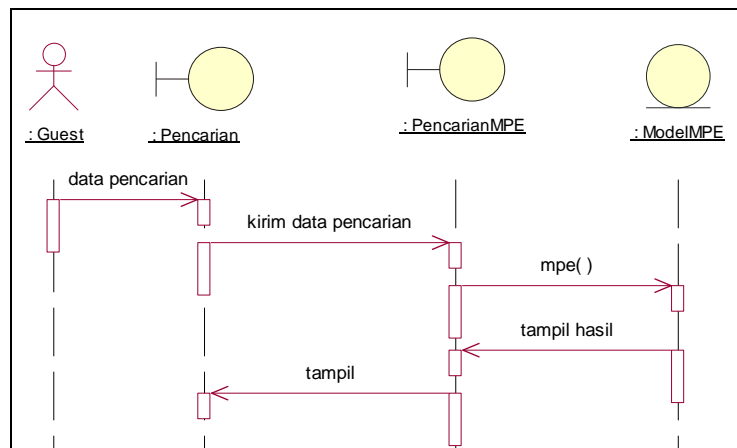
terdaftar atau belum. Apabila belum terdaftar maka sistem akan langsung memasukkan data ke *database*.



Gambar 4.16 *Sequence Diagram* Pendaftaran

4.6.4 *Sequence Diagram* Pencarian

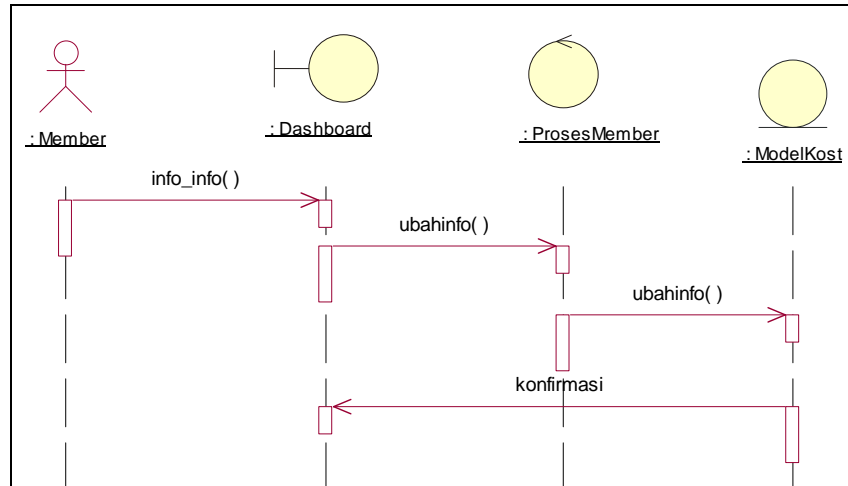
Semua Aktor dapat melakukan pencarian rumah kost. Pada Gambar 4.17, Aktor yang digunakan merupakan perwakilan dari dua aktor lainnya. Terlebih dahulu pengguna memasukkan data pencarian berupa kata kunci dan nilai tingkat kepentingan. Hasil yang ditampilkan melalui proses MPE terlebih dahulu



Gambar 4.17 *Sequence Diagram* Pencarian

4.6.5 *Sequence Diagram* ubah info pada member

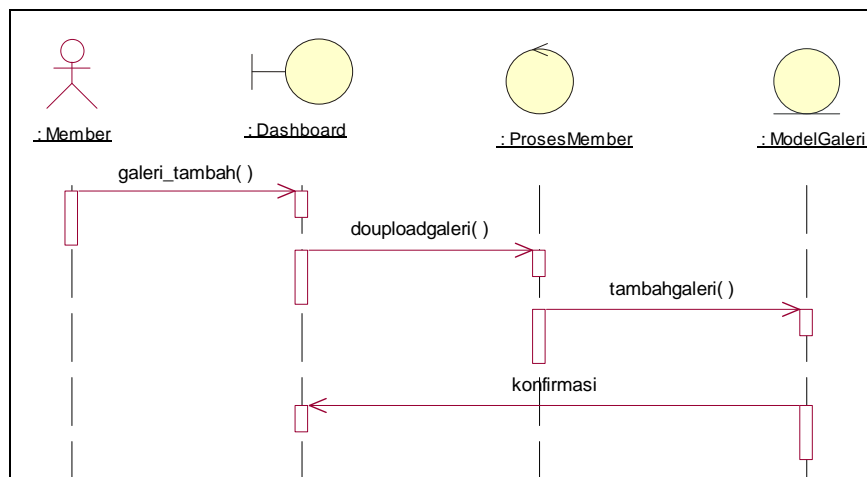
Pada Gambar 4.18 di bawah ini menjelaskan bagaimana *member* melakukan perubahan informasi rumah kost. Informasi disini termasuk koordinat dan info lainnya.



Gambar 4.18 *Sequence Diagram* Ubah Info pada Member

4.6.6 *Sequence Diagram* upload foto galeri pada member

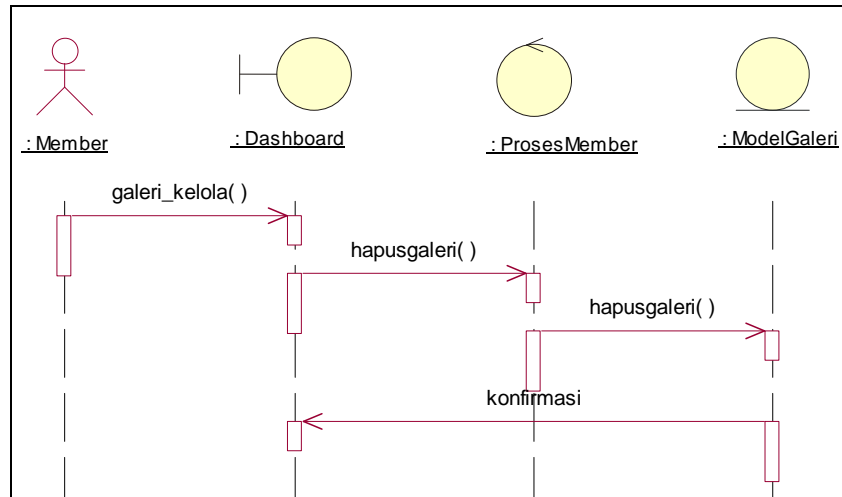
Pada Gambar 4.19 di bawah ini menjelaskan bagaimana *member* melakukan *upload* foto galeri. Apabila sukses menambahkan data foto ke dalam *database*, sistem terlebih dahulu mengunggah foto ke direktori *server* sebelum menampilkan konfirmasi sukses.



Gambar 4.19 *Sequence Diagram* upload foto pada member

4.6.7 *Sequence Diagram* hapus foto galeri pada *member*

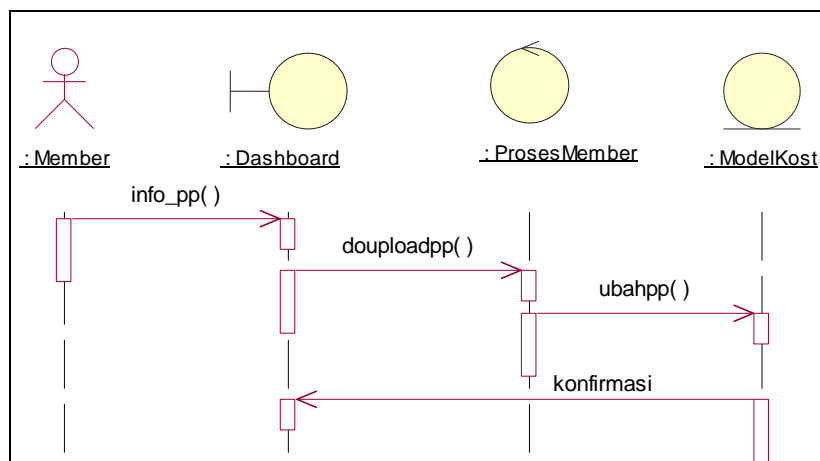
Pada Gambar 4.20 di bawah ini menjelaskan bagaimana *member* menghapus foto galeri. Apabila sukses menghapus data foto ke dalam *database*, sistem terlebih dahulu menghapus foto pada direktori *server* sebelum menampilkan konfirmasi sukses.



Gambar 4.20 *Sequence Diagram* Ubah Foto Profil pada *Member*

4.6.8 *Sequence Diagram* ubah foto profil pada *member*

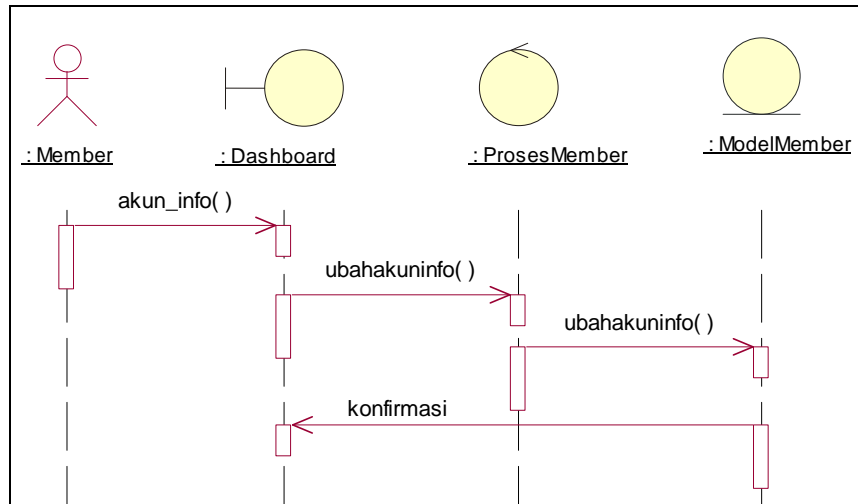
Pada Gambar 3.21 di bawah ini menjelaskan bagaimana *member* melakukan *upload* foto profil. Sama halnya dengan *upload* foto galeri. Data foto yang dipilih terlebih dahulu dimasukkan ke dalam *database* sebelum diunggah ke direktori *server*.



Gambar 4.21 *Sequence Diagram* ubah foto profil pada *member*

4.6.9 Sequence Diagram kelola akun pada member

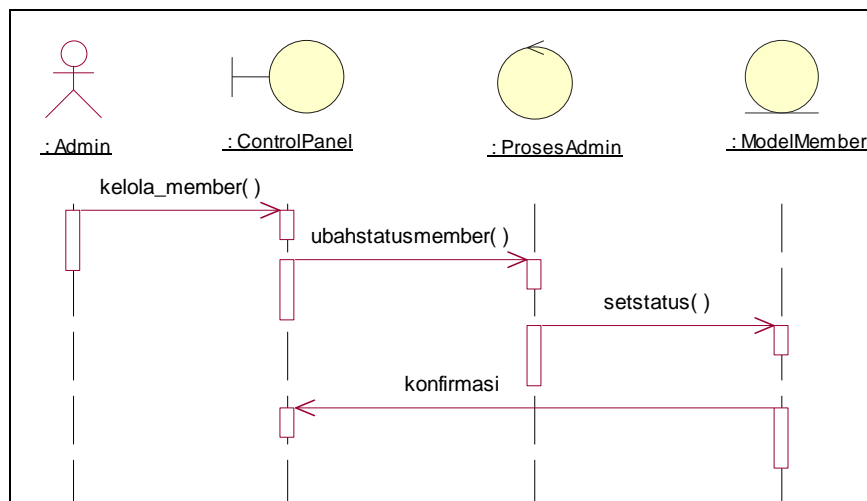
Pada Gambar 4.22 di bawah ini menjelaskan bagaimana *member* melakukan perubahan info akun.



Gambar 4.22 Sequence Diagram kelola akun pada *member*

4.6.10 Sequence Diagram kelola member pada admin

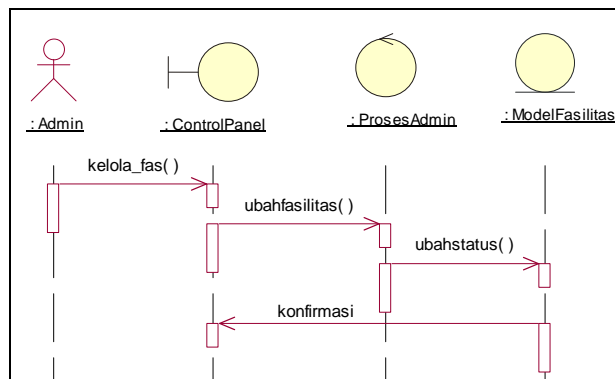
Pada Gambar 4.23 di bawah ini menjelaskan bagaimana admin melakukan pengelolaan *member*. Admin dapat melakukan pengaktifan ataupun pemblokiran *member* tertentu.



Gambar 4.23 Sequence Diagram kelola *member* pada admin

4.6.11 *Sequence Diagram* kelola fasilitas pada admin

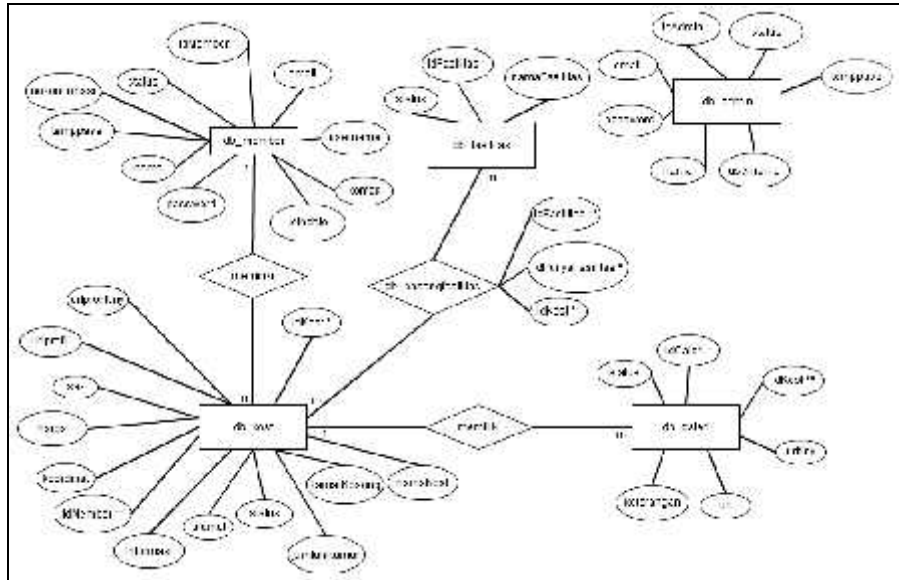
Pada Gambar 3.24 di bawah ini menjelaskan bagaimana admin melakukan pengelolaan terhadap fasilitas. Admin dapat mengubah, menambah, atau menghapus fasilitas. Fasilitas ini merupakan kriteria untuk pemrosesan MPE.



Gambar 4.24 *Sequence Diagram* kelola fasilitas pada admin

4.7 Perancangan *Entity Relational Diagram* (ERD)

Pada Gambar 3.25 berikut merupakan *Entity Relational Diagram* (ERD) dari database yang nantinya akan digunakan oleh sistem. Setiap tabel memiliki *Primary Key* yang menjadi ciri khas pada tabel berikut. Tiap *member* dapat memiliki lebih dari satu rumah kost. Sedangkan tiap rumah kost dapat memiliki lebih dari satu fasilitas. Tiap kost juga dapat memiliki lebih dari satu foto galeri yang menampilkan kumpulan foto rumah kost tersebut.



Gambar 4.25 ERD Pencarian Rumah Kos

4.8 Perancangan Tabel

Tabel-tabel berikut merupakan perancangan tabel berdasarkan *Entity Relational Diagram (ERD)* yang telah dibuat sebelumnya, tabel-tabel ini juga berisikan type data yang digunakan pada setiap field dalam tabel. Selain itu tabel-tabel berikut juga berisikan tentang *data field* yang menjadi *Primary Key* ataupun *Foreign Key* pada tabel hasil relasi.

Tabel 4.1 Tabel Member

| Nama Field | Type | Keterangan |
|--------------|--------------|---|
| idMember | AutoNumber | Primary Key |
| Username | Varchar(30) | |
| Password | Varchar(100) | |
| Email | Varchar(100) | |
| Nama | Varchar(200) | |
| Status | Integer | 0:belum konfirmasi, 1:aktif, 2:nonaktif, 4:banned |
| NoKonfirmasi | Varchar(100) | Nomor konfirmasi untuk mengaktifkan akun |
| Kontak | Varchar(30) | |
| Temppass | Varchar(100) | Password sementara ketika reset password |
| Joindate | Date | Tanggal pendaftaran |

Tabel 4.2 Tabel *Admin*

| Nama Field | Type | Keterangan |
|-------------------|--------------|--|
| idAdmin | AutoNumber | Primary Key |
| username | Varchar(100) | |
| password | Varchar(100) | |
| email | Varchar(100) | |
| nama | Varchar(100) | |
| status | Integer | |
| temppass | Varchar(100) | Password sementara ketika reset password |

Tabel 4.3 Tabel Kost

| Nama Field | Type | Keterangan |
|-------------------|--------------|-----------------------------------|
| idKost | AutoNumber | Primary Key |
| idMember | Integer | Foreign Key |
| namaKost | Varchar(100) | |
| urlprofil | Text | url foto profil |
| urlprofiltiny | Text | url foto profil berukuran kecil |
| harga | Varchar(100) | |
| alamat | Varchar(100) | |
| jumlahKamar | Integer | |
| kamarKosong | Integer | |
| koordinat | Varchar(100) | |
| informasi | Text | |
| sex | Varchar(20) | Pria, Wanita, dan Pria dan Wanita |
| status | Integer | 0:nonaktif, 1:aktif, 2:banned |

Tabel 4.4 Tabel Fasilitas

| Nama Field | Type | Keterangan |
|-------------------|--------------|-------------------|
| idFasilitas | AutoNumber | Primary Key |
| namaFasilitas | Varchar(100) | |
| status | Integer | |

Tabel 4.5 Tabel galeri Kost

| Nama Field | Type | Keterangan |
|-------------------|--------------|-------------------------------|
| idGaleri | AutoNumber | Primary Key |
| idKost | Integer | Foreign Key |
| url | Text | |
| urltiny | Text | url galeri dalam ukuran kecil |
| keterangan | Varchar(100) | |
| status | Integer | |

Tabel 4.6 Tabel pasangFasilitas

| Nama Field | Type | Keterangan |
|-------------------|------------|-------------|
| idPasangFasilitas | AutoNumber | Primary Key |
| idKost | Integer | Foreign Key |
| idFasilitas | Integer | Foreign Key |

4.9 Perancangan *Interface* (Tampilan)

Interface adalah rancangan tatap muka atau tampilan dari sistem informasi yang akan dibuat. Tampilan tersebut dapat mempermudah pembuatan desain tampilan sistem ketika menggunakan *coding*.

4.9.1 Rancangan halaman *home*

Berikut ini adalah tampilan halaman depan Aplikasi pencarian rumah kos menggunakan metode eksponensial

| | |
|-------------------------------|------------------|
| Header | |
| Beranda Pencarian bantuan | Register Login |
| Content | |
| Rumah Kos | |

Gambar 4.26 Rancangan halaman *home user*

4.9.2 Rancangan halaman pencarian

Berikut ini adalah tampilan halaman pencarian Aplikasi pencarian rumah kos menggunakan metode eksponensial (MPE)

| Header | |
|-------------------------------|-----------------|
| Beranda Pencarian bantuan | |
| Register Login | |
| Form Pencarian | Hasil Pencarian |

Gambar 4.27 Rancangan halaman *pencarian user*

4.9.3 Rancangan halaman registrasi

Berikut ini adalah tampilan halaman depan Aplikasi pencarian rumah kos menggunakan metode eksponensial (MPE)

| Header | |
|--|--|
| Beranda Pencarian bantuan | |
| Register Login | |
| <div>REGISTRASI FORM</div> <div>USERNAME <input type="text"/></div> <div>PASSWORD <input type="password"/></div> <div>ULANGI PASSWORD <input type="password"/></div> <div>Nama Lengkap <input type="text"/></div> <div>Email <input type="text"/></div> <div>No Telpn <input type="text"/></div> <div>Kode Keamanan <input type="text"/></div> <div>SUBMIT</div> | |

Gambar 4.28 Rancangan halaman *registrasi user*

4.9.4 Rancangan halaman tambah kos

Berikut ini adalah tampilan halaman depan Aplikasi pencarian rumah kos menggunakan metode eksponensial (MPE)

| Header | |
|-------------------------------|---------------------|
| Beranda Pencarian bantuan | |
| Register Login | |
| Foto Rumah Kos | Deskripsi Rumah Kos |
| Peta Lokasi Rumah Kos | |
| Galery Rumah Kss | |

Gambar 4.29 Rancangan halaman *tambah kos user*

4.9.5 Rancangan halaman login user

Berikut ini adalah tampilan halaman depan Aplikasi pencarian rumah kos menggunakan metode eksponensial (MPE)

The wireframe shows a web page layout for user login. At the top is a header box labeled 'Header'. Below it is a navigation bar containing the links 'Beranda | Pencarian | bantuan' on the left and 'Register | Login' on the right. The main content area contains a centered box titled 'LOGIN USER'. Inside this box are two input fields: 'USERNAME' and 'PASSWORD'. Below the password field is a button labeled 'Masuk'.

Gambar 4.30 Rancangan halaman *login user*

4.9.6 Rancangan halaman login admin

Berikut ini adalah tampilan halaman depan Aplikasi pencarian rumah kos menggunakan metode eksponensial (MPE)

The wireframe shows a web page layout for admin login. It has the same header and navigation bar as the user login page. The main content area contains a centered box titled 'LOGIN ADMIN'. Inside this box are two input fields: 'USERNAME' and 'PASSWORD'. Below the password field is a button labeled 'Masuk'.

Gambar 4.31 Rancangan halaman *admin user*

4.9.7 Rancangan halaman admin

Berikut ini adalah tampilan halaman depan Aplikasi pencarian rumah kos menggunakan metode eksponensial (MPE)

| Header | | |
|--|--|----------------------|
| Beranda Pencarian bantuan | | |
| Register Login | | |
| <div>Rumah Kos</div> <div>Tambah Kos</div> <div>Hapus Kos</div> <div>Akun</div> <div>Ubah email</div> <div>Ubah Password</div> | | <div>Info Akun</div> |
| | | |
| | | |
| | | |

Gambar 4.32 Rancangan halaman menu *admin*

4.9.8 Rancangan halaman tambah rumah kos pada member

Berikut ini adalah tampilan halaman depan Aplikasi pencarian rumah kos menggunakan metode eksponensial (MPE)

Header

Beranda | Pencarian | bantuanRegister | Logout

Tambah Rumah Kos

Nama Kos

Alamat

Kamar Kosong

Gender

harga

Info

Fasilitas

☐ Kasur

☐ Lemari

☐ WC

Tambah

Gambar 4.33 Rancangan halaman menu tambah rumah kos pada member

4.9.9 Rancangan halaman ubah koordinat lokasi pada member

Berikut ini adalah tampilan halaman depan Aplikasi pencarian rumah kos menggunakan metode eksponensial (MPE)

| Header | |
|-------------------------------|--|
| Beranda Pencarian bantuan | Register Logout |
| Ubah Koordinat Lokasi | |
| <div>Peta Google</div> | <div>Cari kota <input type="text"/></div> <div>Titik Koordinat</div> <div>Latitude <input type="text"/></div> <div>Longitude <input type="text"/></div> <div>Ganti</div> |

Gambar 4.34 Rancangan halaman menu ubah koordinat lokasi pada member

4.9.10 Rancangan halaman tambah galeri pada member

Berikut ini adalah tampilan halaman depan Aplikasi pencarian rumah kos menggunakan metode eksponensial (MPE)

| Header | |
|---------------------------------------|---|
| Beranda Pencarian bantuan | Register Logout |
| Tambah Galery | |
| Foto Galeri | <input type="button" value="Pilih Berkas"/> |
| Keterangan | <div></div> |
| <input type="button" value="Upload"/> | |

Gambar 4.35 Rancangan halaman tambah galeri pada member

4.10 Perhitungan Perbandingan Metode Eksponensial

Pada sub bab ini akan dijelaskan analisa menggunakan perhitungan secara manual dari metode yang digunakan yaitu metode eksponensial. Pada proses pencarian rumah kost ini telah ditentukan 9 (sembilan) kriteria penilaian sebagai contoh kasus yang dapat dilihat pada Tabel 3.3.

Tabel 4.7 Kriteria pencarian rumah kost

| No. | Kriteria | TKK |
|-----|--------------|-----|
| 1 | Harga | 9 |
| 2 | WC | 8 |
| 3 | Kasur | 7 |
| 4 | Lemari | 6 |
| 5 | Meja Belajar | 5 |
| 6 | Parkiran | 4 |
| 7 | AC | 3 |
| 8 | Wifi | 2 |
| 9 | Satpam | 1 |

Penentuan tingkat kepentingan dilakukan oleh pencari rumah kost pada saat melakukan pencarian rumah kost. Pada kriteria harga, semakin tinggi tingkat kepentingan yang dimasukkan oleh pencari rumah kost maka berarti mereka lebih memilih harga yang murah. Sedangkan semakin rendah tingkat kepentingan yang dimasukkan pada kriteria harga maka itu berarti pencari tidak mempermasalahkan harga. Pada kriteria fasilitas (baik itu berupa kasur, lemari ataupun fasilitas lainnya), semakin tinggi tingkat kepentingan maka semakin tinggi tingkat kebutuhan pencari rumah kost terhadap fasilitas tersebut. Sedangkan semakin rendah tingkat kepentingan yang dimasukkan pada kriteria fasilitas maka itu berarti pencari rumah kost tidak terlalu mempermasalahkan ada tidaknya fasilitas tersebut.

Berikut ini adalah tabel *range* pembobotan pada perhitungan metode perbandingan eksponensial. Pembobotan harga ditentukan pada saat melakukan pencarian rumah kost.

Tabel 4.8 Nilai bobot *range*

| Tingkat | Keterangan |
|----------------|-------------------|
| 1 | Tidak baik |
| 1,25 | Tidak baik |
| 1,50 | Kurang baik |
| 2,25 | Cukup baik |
| 2,50 | Baik |
| 3 | Lebih baik |
| 3,50 | Sangat baik |
| 4 | Sangat baik |

Berikut ini akan dijelaskan mengenai penentuan nilai dari setiap kriteria yang telah ditentukan.

1. Harga

Semakin murah harga suatu rumah kost maka semakin tinggi nilai bobot yang diberikan. Sebaliknya, semakin mahal harga suatu rumah kost maka semakin rendah nilai bobot yang diberikan. Untuk menentukan komposisi harga, sistem terlebih dahulu mengambil harga terendah dan tertinggi pada suatu daerah yang terdapat di dalam *database*. Sebagai contoh “panam”, harga terendah yang didapat adalah 350.000 dan harga tertinggi adalah 750.000. Untuk mendapatkan penambahan tiap *range* harga maka dapat dilakukan dengan menggunakan persamaan statistik distribusi frekuensi:

$$P = \frac{\text{range}}{\text{jumlah kelompok}}$$

$$P = \frac{(X_{\max} - X_{\min})}{\text{jumlah kelompok}}$$

$$P = \left(\frac{750.000 - 350.000}{9} \right)$$

$$P = 44.444,45$$

Dengan didapatnya nilai penambahan *range* tiap harga maka didapat nilai komposisi harga seperti yang ditampilkan pada Tabel 4.5 berikut ini.

Tabel 4.9 Komposisi penilaian harga

| Harga | RK _{harga} |
|--------------------|---------------------|
| 350.000 - 375.571 | 4 |
| 375.572 - 400.142 | 3.5 |
| 400.143 - 500.713 | 3 |
| 500.714 - 550.285 | 2.50 |
| 550.286 - 600.857 | 2.25 |
| 600.858 - 650.457 | 1.50 |
| 650.858 - 675.428 | 1.25 |
| 6765.429 - 700.427 | 1.25 |
| 700.428 - 750.0000 | 1 |

2. Fasilitas

Pembobotan fasilitas disini mencakup seluruh fasilitas yang ada. Seperti yang ditampilkan pada Tabel 4.6 berikut ini.

Tabel 4.10 Komposisi penilaian fasilitas

| Fasilitas | $RK_{fasilitas}$ |
|------------------|------------------------------------|
| Ada | 4 |
| Tidak Ada | 1 |

Dalam menggunakan metode perbandingan eksponensial ada beberapa tahapan yang harus dilakukan yaitu :

- Menyusun alternatif keputusan yang akan dipilih.
- Menentukan kriteria atau perbandingan kriteria keputusan yang penting untuk dievaluasi.
- Menentukan tingkat kepentingan dari setiap kriteria keputusan atau pertimbangan kriteria.
- Melakukan penilaian terhadap semua alternatif pada setiap kriteria.
- Menghitung total nilai setiap alternatif.
- Menentukan urutan prioritas keputusan didasarkan pada skor atau nilai total masing-masing alternatif.

Berdasarkan contoh kasus yang telah dilakukan perhitungan secara manual dapat dianalisa bahwa penentuan bobot kriteria dari 1 (satu) sampai 4 (empat) menjadi 1 (satu) sampai 9 (sembilan) tidak mempengaruhi hasil akhir dari perengkingan. Yang menjadi pembeda adalah besarnya score setelah dilakukan perhitungan perbandingan eksponensial. Pada saat menggunakan bobot kriteria dari 1 (satu) sampai 9 (Sembilan) menghasilkan nilai bobot yang sangat besar. Hal ini akan berpengaruh pada waktu proses apabila digunakan pada sistem.

BAB V

IMPLEMENTASI DAN PENGUJIAN

Implementasi dan pengujian merupakan tahap yang dilakukan setelah tahap analisa dan perancangan selesai.

5.1 Implementasi

Implementasi aplikasi meliputi tahapan konfigurasi server local (*localhost*), penulisan program dan uji aplikasi. Pada tahapan ini, aplikasi yang telah selesai akan dioperasikan dan dilakukan pengujian untuk melihat sejauh mana aplikasi yang dibuat dapat menghasilkan tujuan yang ingin dicapai.

Implementasi aplikasi ini berupa konfigurasi server *local* dan dibuatnya suatu aplikasi pencarian rumah kos. Aplikasi yang telah dibangun diletakkan didalam *server localhost*, pembuatan aplikasi ini menggunakan bahasa pemrograman PHP, sementara untuk media penyimpanan data atau database menggunakan MySQL. Aplikasi yang dibangun memiliki hak akses admin, member, dan *guest* (masyarakat).

5.1.1 Lingkungan implementasi

Lingkungan implementasi terdiri dari lingkungan perangkat keras atau fisik fisik dan lingkungan perangkat lunak atau aplikasi penunjang yang digunakan untuk membuat aplikasi pencarian rumah kos berbasis web.

Pada Lingkungan perangkat keras dalam implementasi memiliki spesifikasi sebagai berikut :

Perangkat Keras

1. *Processor* : Intel Celeron 1.83 GHz
2. RAM : 1 G
3. *Harddisk* : 80 G
4. Modem : Smartfrend

Sementara implementasi lingkungan perangkat lunak, aplikasi yang digunakan memiliki spesifikasi sebagai berikut :

1. Sistem operasi : Windows Seven Premium 32 bit
2. framework : Codeigniter
3. Language : PHP
4. MySQL Version : 5.0.45
5. Apache : 2.2.5

5.1.2 Batasan implementasi

Batasan implementasi dari tugas akhir ini adalah :

1. Menggunakan bahasa pemrograman PHP sebagai pengkodean.
2. Menggunakan framework PHP *Codeigniter* sebagai bantuan pengkodean.
3. Menggunakan aplikasi wamp5 sebagai *server localhost* dan *database mysql*.
4. Pada implementasi ini, lokasi kosan berada disekitar wilayah kampus UIN Suska Riau.

5.1.3 Hasil implementasi

Hasil Implementasi yang telah dilakukan adalah berupa pengecekan hak akses masing-masing user. Adapapun hak akses tersebut, terdiri dari :

1. Hak Akses Member

Sebelum memiliki hak akses member, terlebih dahulu member mesti melakukan registrasi agar memiliki akses untuk melakukan pengolahan data rumah kos yang dimiliki, member dapat melakukan pengolahan data seperti melakukan pemilihan lokasi rumah kos dan menampilkan data fasilitas yang dimiliki oleh rumah kos.

2. Hak Akses Admin

Pada aplikasi tugas akhir ini hak akses tertinggi dimiliki oleh admin. Admin mengelola member yang mendaftar, admin dapat menon aktifkan dan mengaktifkan member serta dapat melakukan pengolahan data fasilitas.

3. Hak akses *Guest* (masyarakat)

Masyarakat dapat menggunakan aplikasi ini tanpa harus melakukan login. Masyarakat dapat melihat kos-kosan yang berada di sekitar wilayah kampus UIN Suska beserta deskripsi dari fasilitas yang dimiliki oleh masing-masing kosan.

Aplikasi web ini dibuat dengan menggunakan pemrograman PHP Codeigniter 2.0 *framework*. Adapun pengujian terfokus pada proses kerja sistem dan keluarannya.

Terdapat 3 (tiga) jenis pengguna yang dapat mengakses sistem ini. Pengguna sistem tersebut yaitu:

1. *Guest*, merupakan pencari kost yang tidak membutuhkan login untuk dapat menggunakan sistem. *Guest* hanya dapat melakukan pencarian dan pendaftaran sebagai pemilik kost (*Member*).
2. *Member*, merupakan pemilik kost yang telah terdaftar pada sistem. *Member* dapat menambah dan mengelola data rumah kost mereka pada sistem.
3. *Admin*, merupakan pengguna yang memiliki hak untuk mengelola sistem secara keseluruhan. *Admin* memiliki hak akses penuh dalam sistem.

5.2 Pengujian *Guest*

Guest memiliki keterbatasan dalam melakukan hal pada sistem. *Guest* hanya dapat melakukan pencarian dan pendaftaran. Namun setiap halaman yang dapat diakses oleh *guest* dapat juga diakses oleh *member* dan *admin*.

1. Tampilan beranda Web Pencarian Kost

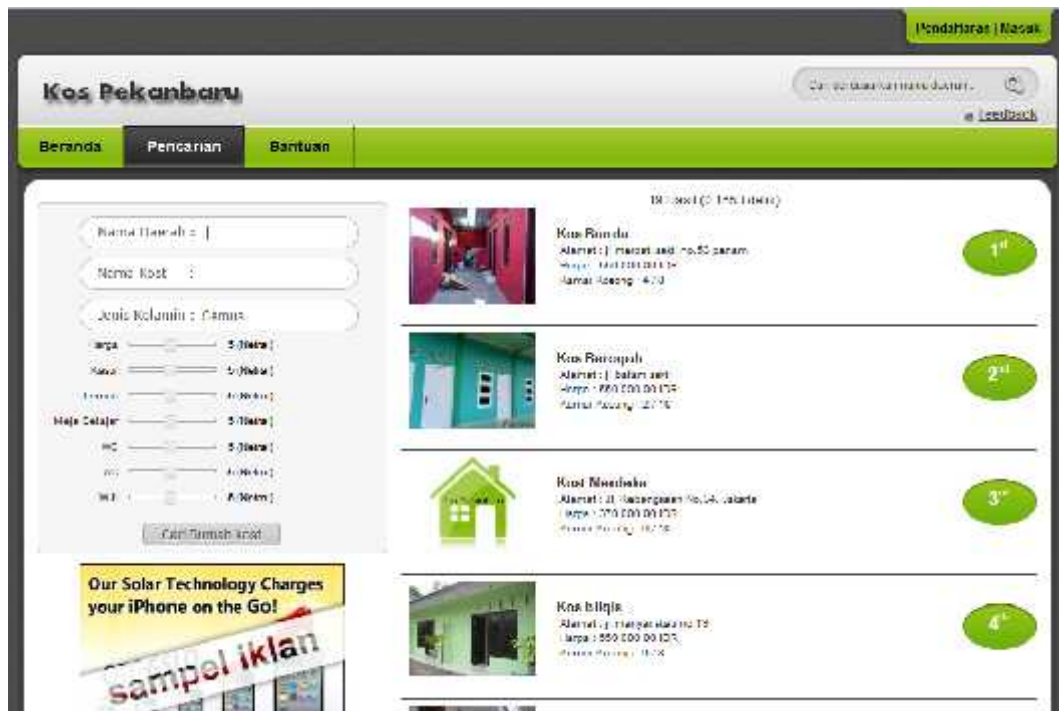
Halaman ini akan ditampilkan pertama kali saat web diakses. Disini pengguna dapat memilih apa yang akan dilakukan seperti melakukan pencarian, login, ataupun mendaftarkan diri sebagai pemilik kost. Pada halaman ini tidak ada validasi yang dibutuhkan. Seperti yang ditampilkan pada Gambar 5.1 berikut:



Gambar 5.1 Halaman Beranda pada Web

2. Tampilan halaman pencarian

Halaman ini merupakan tempat dimana pengguna dapat melakukan pencarian menggunakan metode MPE. Pengguna menentukan tingkat kepentingan suatu kriteria seperti harga, kasur, lemari ataupun fasilitas lainnya. Pengguna juga dapat menentukan daerah rumah kost yang diinginkan serta jenis kelamin penghuni rumah kost dengan mengisi *field* yang telah disediakan. Pada halaman ini tidak ada proses validasi yang dibutuhkan. Seperti yang ditampilkan pada Gambar 5.2 berikut:



Gambar 5.2 Halaman Pencarian pada Web

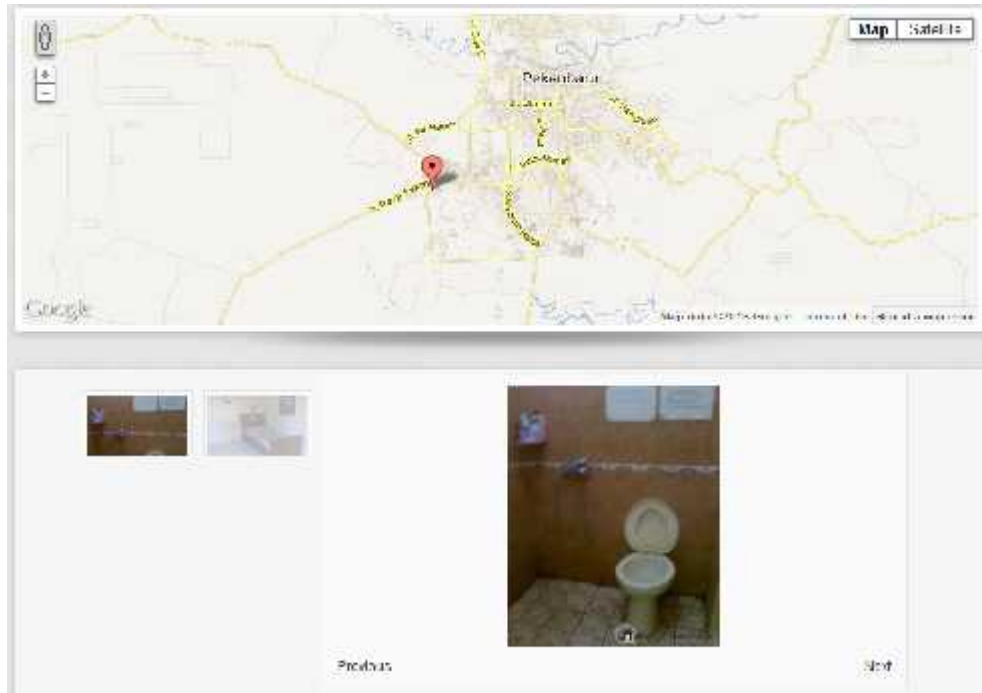
3. Tampilan halaman profil rumah kost

Halaman ini menampilkan informasi lebih mendetail mengenai suatu rumah kost. Apabila pengguna mengklik salah satu rumah kost pada halaman pencarian, maka pengguna akan diarahkan ke halaman profil yang berisi informasi mengenai rumah kost yang dipilihnya. Seperti yang ditampilkan pada Gambar 5.3 berikut:



Gambar 5.3 Halaman Profil pada Web bagian 1

Apabila pemilik rumah kost menentukan koordinat rumah kost mereka secara tepat, maka pengguna dapat melihatnya pada bagian map di halaman profil ini. Sama halnya dengan foto galeri. Pengguna juga dapat melihat foto galeri suatu rumah kost apabila pemilik mengunggahnya. Seperti yang ditampilkan pada Gambar 5.4 berikut:



Gambar 5.4 Halaman Profil pada Web bagian 2

4. Tampilan halaman pendaftaran

Pada halaman ini pengguna dapat melakukan pendaftaran sebagai pemilik rumah kost. Di halaman ini terdapat beberapa validasi yang telah berjalan dengan baik. Seperti pemilihan username yang dapat digunakan, password dengan minimal lima karakter serta format email yang sesuai. Apabila terdapat kesalahan, maka sistem akan menampilkan pesan bahwa terdapat kesalahan. Untuk *field* nomor telepon, pengguna hanya dapat memasukkan berupa angka saja. Pemilihan email juga dilakukan validasi apakah username telah terdaftar pada sistem atau belum. Apabila telah terdaftar maka pengguna harus menggunakan email yang lain. Form juga dilengkapi dengan pengisian kode keamanan. Hal ini sangat membantu untuk mencegah adanya pendaftaran menggunakan bot. Seperti yang ditampilkan pada Gambar 5.5 berikut:

Username: Username harus lebih dari 5 karakter

Password: Gunakan password minimal 8 karakter

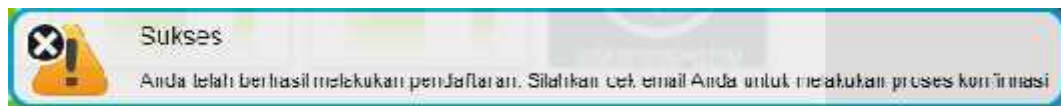
Email: Masukkan alamat email yang valid

Phone Number:

Daftar Sekarang

Gambar 5.5 Halaman pendaftaran

Setelah melakukan pendaftaran, maka sistem akan mengirim link konfirmasi ke email pengguna. Pengguna cukup mengklik link tersebut untuk mengaktifkan akun.



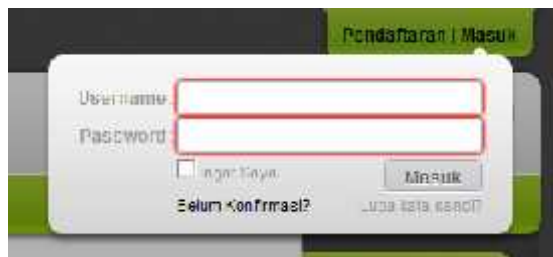
Gambar 5.6 Notifikasi sukses melakukan pendaftaran

5.3 Pengujian *Member*

Untuk menjadi *member*, pengguna harus terlebih dahulu melakukan pendaftaran. Apabila telah melakukan pendaftaran serta mengaktifkan akun, maka pengguna dapat langsung melakukan login sebagai *member*. Berikut adalah aktifitas yang dapat dilakukan oleh *member*.

1. Halaman Login Member

Terdapat dua cara untuk melakukan login. Yang pertama, pengguna cukup mengklik link “masuk” yang terdapat pada bagian kanan atas halaman. Apabila sukses, maka sistem akan menampilkan form login. Pada form ini sistem mencegah pengguna untuk memasukkan data kosong yaitu dengan cara menggunakan *attribute required* pada *textbox*. Seperti yang ditampilkan pada Gambar 5.7 berikut:

A screenshot of a web application's login page. At the top right, there is a green button labeled 'Pendaftaran / Masuk'. Below it is a white login form with a green border. The form contains two input fields: 'Username' and 'Password'. The 'Password' field has a red border around it, indicating a validation error. Below the fields is a checkbox labeled 'Ingat Saya' and a 'Masuk' button. At the bottom of the form, there is a link that says 'Belum Konfirmasi?'. The background of the page is dark green.

Gambar 5.7 Halaman login pertama

Apabila proses login gagal, maka pengguna akan diarahkan ke halaman login. Seperti yang ditampilkan pada Gambar 5.8 berikut:

Kost Pekanbaru

Member Login

Back

Username

Password

☐ Ingat Saya

Submit

Belum Konfirmasi?

Lupa Password?

Gambar 5.8 Halaman login kedua

Pada form dan halaman login ini memiliki fitur reset *password* apabila pengguna lupa. Pengguna cukup dengan mengklik “lupa *password*” dan kemudian sistem akan menampilkan form yang mengharuskan pengguna untuk memasukkan alamat email. Seperti yang ditampilkan pada Gambar 5.9 berikut:

Kost Pekanbaru

Member Login

Back

Username

Password

☐ Ingat Saya

Submit

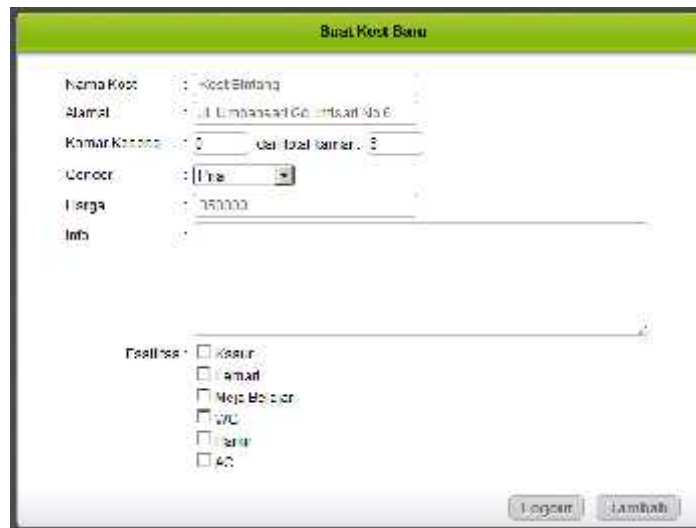
Belum Konfirmasi?

Lupa Password?

Gambar 5.9 Halaman lupa password

2. Halaman Tambah Rumah Kost Pertama

Jika pengguna baru saja mendaftar dan belum mempunyai data rumah kost, maka pengguna secara otomatis akan diarahkan ke halaman untuk membuat data kost baru. Pada halaman ini, terdapat beberapa validasi seperti jumlah kamar kosong harus lebih kecil dari total kamar. Pada *field* ini terdapat *javascript* yang mencegah pengguna untuk menginputkan selain angka. *Javascript* ini juga digunakan sebagai validasi pada *field* harga. Pada *field* info di form ini, jumlah karakter komentar tidak dibatasi, sehingga bisa memasukkan ribuan jumlah karakter. Seperti yang ditampilkan pada Gambar 5.10 berikut:



Gambar 5.10 Halaman Tambah Kost Pertama

3. Dashboard

Apabila berhasil menambah data rumah kost baru, maka pengguna akan diarahkan ke halaman *dashboard*. Disini pengguna dapat melakukan banyak hal yang berhubungan dengan rumah kost. Seperti yang ditampilkan pada Gambar 5.11 berikut:



Gambar 5.11 Halaman *dashboard*.

Pada halaman ini ada beberapa bagian yang perlu diperhatikan. Yang pertama adalah *panel* pada bagian tengah yang berisi beberapa icon yang akan mengarahkan pengguna ke halaman yang diinginkan. Seperti yang ditampilkan pada Gambar 4.12 berikut:



Gambar 5.12 Panel icon pada *dashboard*

Kemudian pada bagian kiri atas halaman *dashboard*, terdapat form yang berguna untuk menentukan rumah kost yang ingin dikelola. Seperti yang ditampilkan pada Gambar 5.13 berikut:



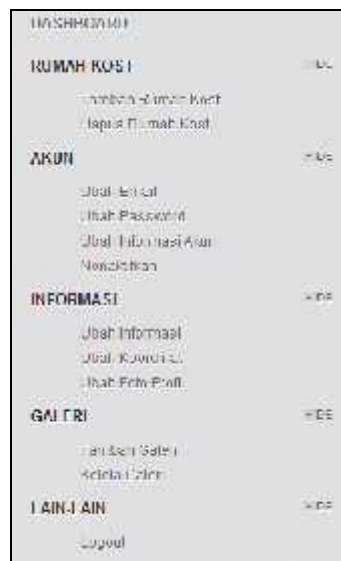
Gambar 5.13 From ubah rumah kost pada *dashboard*

Pada bagian kanan terdapat informasi mengenai jumlah rumah kost yang telah pengguna masukkan pada sistem dan jumlah galeri pada seluruh rumah kost serta pada rumah kost yang sedang dikelola. Seperti yang ditampilkan pada Gambar 5.14 berikut:



Gambar 4.14 Panel informasi akun pada *dashboard*

Pada bagian kiri halaman *dashboard* terdapat *link* menu yang akan mengarahkan pengguna ke halaman yang diinginkan. Seperti yang ditampilkan pada Gambar 5.15 berikut:



Gambar 5.15 Panel *side-menu* pada *dashboard*

Ada terdapat beberapa fungsi yang dapat dilakukan pada halaman *dashboard* ini. Berikut adalah fungsi/fitur yang dapat dilakukan pengguna pada halaman *dashboard*:

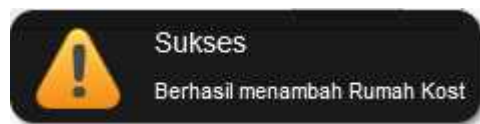
a. Halaman Tambah Rumah Kost

Sama halnya dengan form tambah rumah kost sebelumnya, pada form ini juga terdapat *javascript* yang mencegah pengguna untuk menginputkan selain angka pada jumlah kamar kosong ,total kamar dan harga. Pada *field* info di form ini, jumlah karakter komentar tidak dibatasi, sehingga bisa memasukkan ribuan jumlah karakter. Seperti yang ditampilkan pada Gambar 5.16 berikut:



Gambar 5.16 Halaman Tambah Kost pada *dashboard*

Setiap aktifitas akan memberikan pesan apakah suatu aktifitas yang dilakukan berhasil atau gagal. Apabila sukses menambah data rumah kost baru, maka akan muncul notifikasi. Seperti yang ditampilkan pada Gambar 5.17 berikut:



Gambar 5.17 Contoh notifikasi sukses

b. Halaman Hapus Rumah Kost

Halaman ini berfungsi untuk menghapus suatu rumah kost pengguna. Sistem menampilkan terlebih dahulu informasi rumah kost yang hendak dihapus. Apabila data rumah kost telah sesuai maka pengguna dapat langsung mengklik tombol “Hapus Rumah Kost”. Seperti yang ditampilkan pada Gambar 5.18 berikut:

Gambar 4.18 Halaman Hapus Rumah Kost

Sebelum menghapus rumah kost, sistem terlebih dahulu menampilkan pesan konfirmasi apakah data rumah kost siap untuk dihapus. Apabila pengguna mengklik tombol “OK” maka sistem akan menghapus data rumah kost. Seperti yang ditampilkan pada Gambar 5.19 berikut:



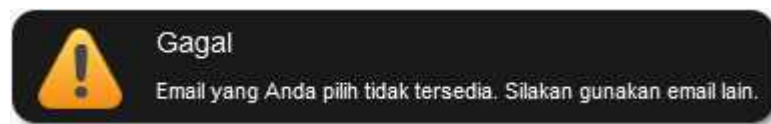
Gambar 5.19 Pesan Konfirmasi Hapus Rumah Kost

c. Halaman Ubah Email

Pada halaman ini pengguna dapat mengubah email yang sudah terdaftar menjadi email yang lain. Seperti yang ditampilkan pada Gambar 5.20 berikut:

Gambar 5.20 Halaman Ubah Email

Pada setiap *field* diberikan *attribute required* yang mengharuskan pengguna untuk mengisi *field* tersebut. Sistem juga memberikan validasi terhadap email yang diinputkan. Apabila email baru yang dimasukkan telah digunakan, maka sistem akan menampilkan notifikasi bahwa email yang dimasukkan tidak tersedia untuk pengguna. Seperti yang ditampilkan pada Gambar 5.21 berikut:



Gambar 5.21 Notifikasi Gagal Mengubah Email

Apabila sukses mengubah email maka sistem akan menampilkan pesan sukses. Seperti yang ditampilkan pada Gambar 4.22 berikut:



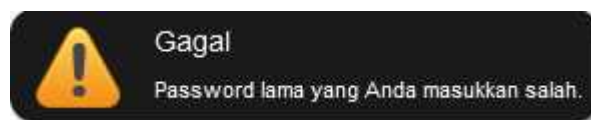
Gambar 5.22 Notifikasi Sukses Mengubah Email

d. Halaman Ubah Password

Pada halaman ini member dapat mengubah password. Untuk mengubah password, pengguna harus terlebih dahulu memasukkan password lama dengan benar sebelum memasukkan password baru. Seperti yang ditampilkan pada Gambar 5.23 berikut:

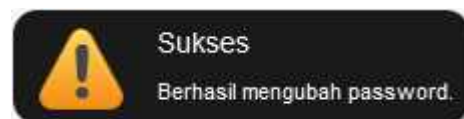
Gambar 5.23 Halaman Ubah Password

Apabila pengguna salah memasukkan password lama, maka sistem akan memberikan notifikasi “Password lama yang Anda masukkan salah”. Seperti yang ditampilkan pada Gambar 5.24 berikut:



Gambar 5.24 Notifikasi Kesalahan Password Lama

Untuk mengubah password, pengguna juga diharuskan memasukkan password sebanyak dua kali. Apabila terdapat ketidaksamaan dalam menginputkan password baru, maka sistem akan memberikan pesan kesalahan. Apabila pengguna mengisi *field* dengan benar maka sistem akan menampilkan pesan sukses. Seperti yang ditampilkan pada Gambar 5.25 berikut:



Gambar 4.25 Notifikasi Sukses Mengubah Password

e. Halaman Ubah Informasi Akun

Pada halaman ini pengguna dapat mengubah informasi berupa nama lengkap dan nomor kontak. Disini hanya ada satu validasi yaitu pada *field* kontak yang mengharuskan pengguna untuk memberikan masukan berupa angka. Seperti yang ditampilkan pada Gambar 5.26 berikut:

Gambar 5.26 Halaman Ubah Informasi Akun

f. Halaman Nonaktif

Halaman ini bertujuan untuk menonaktifkan pengguna. Dengan menonaktifkan akun maka seluruh data rumah kost yang telah dimasukkan oleh pengguna tidak dapat diakses oleh pencari rumah kost. Halaman ini berguna untuk pemilik kost yang ingin menghentikan publikasi terhadap rumah kost mereka. Seperti yang ditampilkan pada Gambar 4.57 berikut:

Gambar 5.27 Halaman Nonaktif

Sebelum menonaktifkan akun, sistem terlebih dahulu menampilkan konfirmasi apakah pengguna yakin akan menonaktifkan akun. Apabila pengguna yakin maka sistem akan langsung menonaktifkan akun.

Penting untuk diketahui bahwa walaupun pengguna telah menonaktifkan akunnya, pengguna tetap bisa mengaktifkan kembali. Pengguna cukup login dengan menggunakan data login seperti biasa. Kemudian sistem akan kembali mengaktifkan akun pengguna dan akun dapat kembali digunakan seperti biasa.

g. Halaman Ubah Informasi Rumah Kost

Pada halaman ini pengguna dapat mengubah informasi rumah kost. Validasi yang digunakan sama halnya dengan halaman tambah kost baru. Begitu juga dengan kekurangannya. Pada *field* info di form ini, jumlah karakter komentar tidak dibatasi, sehingga bisa memasukkan ribuan jumlah karakter. Seperti yang ditampilkan pada Gambar 5.28 berikut:

Gambar 5.28 Halaman Ubah Informasi Rumah Kost

h. Halaman Ubah Koordinat Rumah Kost

Pada halaman ini pengguna dapat menentukan titik koordinat rumah kost mereka. Seperti yang ditampilkan pada Gambar 5.29 berikut:

Gambar 4.29 Halaman Ubah Koordinat Rumah Kost

Pengguna lebih dimudahkan dengan menggunakan Google Maps API dalam menentukan koordinat rumah kost mereka. Hal pertama yang harus dilakukan yaitu memasukkan nama kota dimana rumah kost pengguna berada pada *field* yang telah disediakan. Setelah itu klik tombol “Cari” maka Google Maps akan langsung mengarah pada kota sesuai dengan nama kota yang pengguna masukkan. Kemudian pengguna cukup menggeser pointer yang ada pada Google

Maps ke arah dimana rumah kost pengguna berada. Pengguna dapat menggunakan mode “Hibrida” untuk menghasilkan tampilan yang lebih jelas.

i. Halaman Ubah Foto Profil

Pada halaman ini pengguna dapat mengubah foto profil rumah kost. Seperti yang ditampilkan pada Gambar 5.30 berikut:



Gambar 5.30 Halaman Ubah Foto Profil

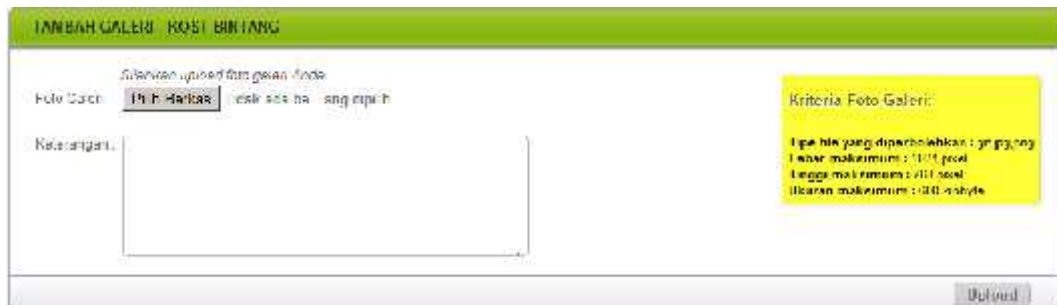
Kriteria foto yang diperbolehkan untuk diunggah telah ditampilkan dengan sangat jelas. Apabila foto yang diunggah diluar persyaratan, maka sistem akan menampilkan pesan kesalahan. Seperti yang ditampilkan pada Gambar 5.31 berikut:



Gambar 5.31 Notifikasi Gagal Mengunggah Foto

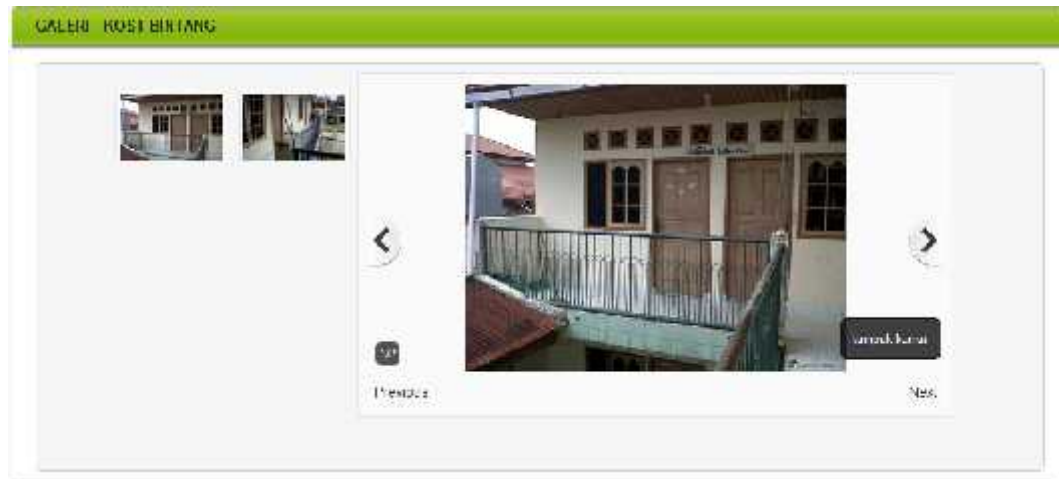
j. Halaman Tambah Galeri

Pada halaman ini pengguna dapat mengunggah foto rumah kost mereka. Fitur ini sangat berguna untuk memberikan gambaran kepada pencari kost seperti apa suasana atau bentuk dari rumah kost pengguna. Pada *field* keterangan di form ini, jumlah karakter komentar tidak dibatasi, sehingga bisa memasukkan ribuan jumlah karakter. Seperti yang ditampilkan pada Gambar 5.32 berikut:



Gambar 5.32 Halaman Tambah Galeri

Pada bagian bawah halaman ini pengguna dapat melihat foto galeri apa saja yang telah diunggah pada suatu rumah kost. Seperti yang ditampilkan pada Gambar 5.33 berikut:



Gambar 5.33 Tampilan Galeri pada Halaman Tambah Galeri

k. Halaman Kelola Galeri

Pada halaman ini pengguna dapat mengubah informasi ataupun menghapus galeri. Pada *field* keterangan di form ini, jumlah karakter komentar tidak dibatasi, sehingga bisa memasukkan ribuan jumlah karakter. Seperti yang ditampilkan pada Gambar 5.34 berikut:



Gambar 5.34 Halaman Kelola Galeri

5.4 Pengujian Admin

1. Halaman Login Admin

Cara mengakses halaman ini sedikit unik. Admin harus menginputkan *url* secara manual. Misalnya localhost/login/admin/29/05/13. Bagian “29/05/13” pada

link tersebut merupakan tanggal pada server. Jadi *url* halaman login admin tidak tetap. Setiap hari halaman login admin memiliki *url* yang berbeda-beda. Berikut adalah potongan script yang digunakan:

```
function admin($tgl="$bln=$thn"){
    $date=date("dmy");
    if($tgl.$bln.$thn==$date){
        $this->load->view("login_admin");
        $this->load->view("notif");
    }else{
        header("location:".site_url("login"));
    }
}
```

Potongan *script* di atas berada pada halaman controller yang diberi nama login.php. Fungsi admin dijalankan dengan mengakses *class* login terlebih dahulu. Oleh sebab itu untuk mengakses halaman login admin harus dengan melalui *url* localhost/login/admin dengan langsung memasukkan parameter tanggal seperti yang telah disebutkan sebelumnya.

Validasi yang dilakukan pada halaman login admin sama halnya dengan login member. Sistem melakukan pengecekan apakah data login yang dimasukkan benar atau tidak. Apabila data login yang dimasukkan benar maka sistem akan mengarahkan pengguna akan diarahkan ke halaman *Control Panel*.

2. *Control Panel*

Setelah sukses melakukan login admin, pengguna diarahkan ke halaman *Control Panel*. Seperti yang ditampilkan pada Gambar 5.35 berikut:



Gambar 4.35 Halaman *Control Panel*

Pada *Control Panel* ini memiliki beberapa halaman yang memiliki fungsi tersendiri. Berikut adalah halaman yang terdapat pada halaman *Control Panel*:

a. Halaman Kelola Fasilitas

Pada halaman ini pengguna dapat mengelola fasilitas apa saja yang akan dimasukkan dalam perhitungan MPE. Pengguna dapat menghapus, mengubah maupun menambah fasilitas. Setiap proses yang dilakukan menggunakan AJAX. Seperti yang ditampilkan pada Gambar 5.36 berikut:



Gambar 5.36 Halaman Kelola Fasilitas

Apabila pengguna mengklik tombol “Tambah” pada tabel kelola fasilitas maka akan muncul form tambah fasilitas. Seperti yang ditampilkan pada Gambar 5.37 berikut:

Gambar 5.37 Form Tambah Fasilitas

b. Halaman Kelola Rumah Kost

Pada halaman ini pengguna dapat mengubah status ataupun menghapus rumah kost. Seperti yang ditampilkan pada Gambar 5.38 berikut:

| ID | Username | Nama Kost | Status | Kelas |
|-----------|-----------|--------------------|--------|-------|
| 100000001 | ara | Kost Dimping | Aktif | |
| 100000002 | widhik | Gaya Jelas | Aktif | |
| 100000003 | perman | Kost GRK | Aktif | |
| 100000004 | perman | Kost Prima | Aktif | |
| 100000005 | fransesca | Kost Orange | Aktif | |
| 100000006 | ara | Kost Cendrawasih I | Aktif | |
| 100000007 | widhik | Sendawa K139 | Aktif | |
| 100000008 | widhik | Kost Jusdolo | Aktif | |
| 100000009 | ara | Kost Bawang | Aktif | |

Showing 1 to 9 of 9 entries

Gambar 5.38 Halaman Kelola Rumah Kost

c. Halaman Kelola Member

Pada halaman ini pengguna dapat mengubah status ataupun menghapus rumah kost. Seperti yang ditampilkan pada Gambar 5.39 berikut:

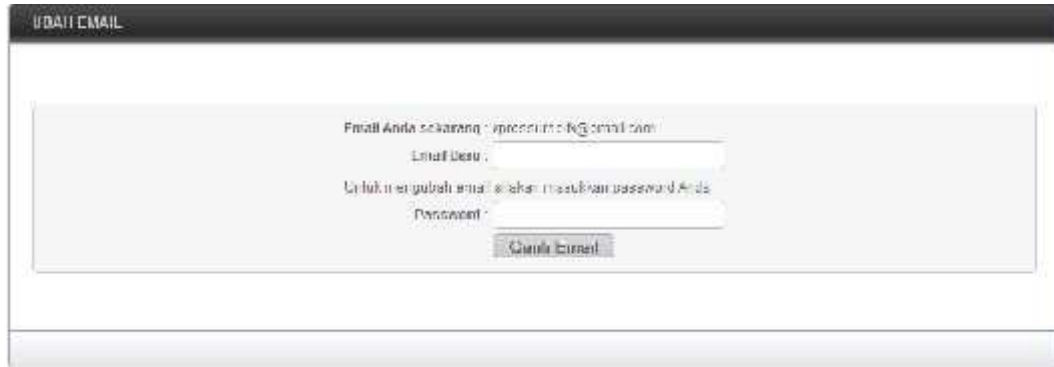
| ID | Username | Nama Kost | Status | Kelas |
|-----------|----------|---------------|--------|-------|
| 100000001 | ara | Kost Merdeka | Aktif | |
| 100000002 | ara | Kost pak do | Aktif | |
| 100000003 | ara | Kost Pangeran | Aktif | |
| 100000004 | ara | Kost Tanaman | Aktif | |
| 100000005 | ara | Kost Teratai | Aktif | |
| 100000006 | ara | Kost Hitam | Aktif | |
| 100000007 | ara | Kost Pink | Aktif | |
| 100000008 | ara | Kost Palang | Aktif | |
| 100000009 | ara | Kost Hitam | Aktif | |
| 100000010 | ara | Kost Bawang | Aktif | |
| 100000011 | ara | Kost Bawang | Aktif | |
| 100000012 | ara | Kost Bawang | Aktif | |
| 100000013 | ara | Kost Bawang | Aktif | |
| 100000014 | ara | Kost Bawang | Aktif | |
| 100000015 | ara | Kost Bawang | Aktif | |
| 100000016 | ara | Kost Bawang | Aktif | |
| 100000017 | ara | Kost Bawang | Aktif | |
| 100000018 | ara | Kost Bawang | Aktif | |
| 100000019 | ara | Kost Bawang | Aktif | |
| 100000020 | ara | Kost Bawang | Aktif | |

Showing 1 to 20 of 20 entries

Gambar 5.39 Halaman Kelola Member

d. Halaman Ubah Email

Pada halaman ini pengguna dapat mengubah email yang sudah terdaftar menjadi email yang lain. Seperti yang ditampilkan pada Gambar 5.40 berikut:



Gambar 5.40 Halaman Ubah Email Admin

Pada setiap *field* diberikan *attribute required* yang mengharuskan pengguna untuk mengisi *field* tersebut. Sistem juga memberikan validasi terhadap email yang diinputkan. Apabila email baru yang dimasukkan telah digunakan, maka sistem akan menampilkan notifikasi bahwa email yang dimasukkan tidak tersedia untuk pengguna. Seperti yang ditampilkan pada Gambar 5.41 berikut:



Gambar 5.41 Notifikasi Gagal Mengubah Email Admin

Apabila sukses mengubah email maka sistem akan menampilkan pesan sukses. Seperti yang ditampilkan pada Gambar 5.42 berikut:



Gambar 5.42 Notifikasi Sukses Mengubah Email

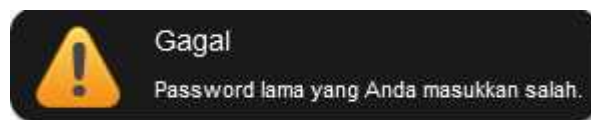
e. Halaman Ubah Password

Pada halaman ini pengguna dapat mengubah password. Untuk mengubah password, pengguna harus terlebih dahulu memasukkan password lama dengan benar sebelum memasukkan password baru. Seperti yang ditampilkan pada Gambar 5.43 berikut:



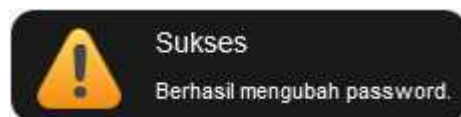
Gambar 5.43 Halaman Ubah Password

Apabila pengguna salah memasukkan password lama, maka sistem akan memberikan notifikasi “Password lama yang Anda masukkan salah”. Seperti yang ditampilkan pada Gambar 4.44 berikut:



Gambar 5.44 Notifikasi Kesalahan Password Lama

Untuk mengubah password, pengguna juga diharuskan memasukkan password sebanyak dua kali. Apabila terdapat ketidaksamaan dalam menginputkan password baru, maka sistem akan memberikan pesan kesalahan. Apabila pengguna mengisi *field* dengan benar maka sistem akan menampilkan pesan sukses. Seperti yang ditampilkan pada Gambar 5.45 berikut:



Gambar 5.45 Notifikasi Sukses Mengubah Password

5.5 Pengujian Dengan Menggunakan *Blackbox*

5.5.1 Modul Pengujian Tampil Register

Tabel 5.1. Pengujian modul tampil register

| Deskripsi | Prosedur pengujian | Masukkan | Keluaran yang diharapkan | Kriteria evaluasi hasil | Hasil yang didapat | Kesimpulan |
|---------------------------|---------------------------|--|---|--|---|------------|
| Pengujian tampil register | Klik tombol menu register | 1. Username 2. Passsword 3. Ulangi password 4. Nama Lengkap 5. email 6. Nomor telepon 7. Kode keamanan | Muncul Pesan Register berhasil | Data berhasil diproses Tampilan menu sesuai yang diharapkan | Muncul Pesan Register berhasil | Diterima |

5.5.2 Modul Pengujian Tampil Login member

Tabel 5.2. Pengujian modul tampil login

| Deskripsi | Prosedur pengujian | Masukkan | Keluaran yang diharapkan | Kriteria evaluasi hasil | Hasil yang didapat | Kesimpulan |
|------------------------|------------------------|----------------------------|---|--|----------------------------------|------------|
| Pengujian tampil login | Klik tombol menu login | 1. Username 2. Password | Muncul Pesan Register berhasil | Data berhasil diproses Tampilan menu sesuai yang diharapkan | Masuk halaman akses member | Diterima |

5.5.3 Modul Pengujian Tampil Tambah Rumah Kos

Tabel 5.3. Pengujian modul tampil tambah rumah kos

| Deskripsi | Prosedur pengujian | Masukkan | Keluaran yang diharapkan | Kriteria evaluasi hasil | Hasil yang didapat | Kesimpulan |
|-----------------------------------|-----------------------------------|---|--|--|------------------------|------------|
| Pengujian tampil tambah rumah kos | Klik tombol menu tambah rumah kos | 1. Nama kos 2. Alamat 3. Kamar kosong 4. Gender 5. Harga 6. Info 7. Fasilitas | Data berhasil diproses Tampilan menu sesuai yang diharapkan | Data berhasil diproses Tampilan menu sesuai yang diharapkan | Data berhasil diproses | Diterima |

5.5.4 Modul Pengujian Ubah Email

Tabel 5.4. Pengujian modul tampil ubah email

| Deskripsi | Prosedur pengujian | Masukkan | Keluaran yang diharapkan | Kriteria evaluasi hasil | Hasil yang didapat | Kesimpulan |
|-----------------------------|-----------------------------|------------------------------|---|--|---|------------|
| Pengujian tampil ubah email | Klik tombol menu ubah email | 1. Email baru 2. password | Data berhasil diproses dan tampilan sesuai yang di inginkan | Data berhasil diproses Tampilan menu sesuai yang diharapkan | Data berhasil diproses dan tampilan sesuai yang di inginkan | Diterima |

5.5.5 Modul Pengujian Ubah Password

Tabel 5.5. Pengujian modul tampil ubah password

| Deskripsi | Prosedur pengujian | Masukkan | Keluaran yang diharapkan | Kriteria evaluasi hasil | Hasil yang didapat | Kesimpulan |
|--------------------------------|--------------------------------|--|---|--|---|-------------------|
| Pengujian tampil ubah password | Klik tombol menu ubah password | 1. Password lama 2. Password baru 3. Ulangi password | Data berhasil diproses dan tampilan sesuai yang di inginkan | Data berhasil diproses Tampilan menu sesuai yang diharapkan | Data berhasil diproses dan tampilan sesuai yang di inginkan | Diterima |

5.5.6 Modul Pengujian Tampil Ubah Informasi Akun

Tabel 5.6. Pengujian modul ubah informasi akun

| Deskripsi | Prosedur pengujian | Masukkan | Keluaran yang diharapkan | Kriteria evaluasi hasil | Hasil yang didapat | Kesimpulan |
|--------------------------------------|--------------------------------------|------------------------------|---|--|---|-------------------|
| Pengujian tampil ubah informasi akun | Klik tombol menu ubah informasi akun | 1. Nama lengkap 2. Kontak | Data berhasil diproses dan tampilan sesuai yang di inginkan | Data berhasil diproses Tampilan menu sesuai yang diharapkan | Data berhasil diproses dan tampilan sesuai yang di inginkan | Diterima |

5.5.7 Modul Pengujian Tampil Ubah Informasi Rumah Kos

Tabel 5.7. Pengujian modul tampil ubah informasi rumah kos

| Deskripsi | Prosedur pengujian | Masukkan | Keluaran yang diharapkan | Kriteria evaluasi hasil | Hasil yang didapat | Kesimpulan |
|---------------------------------|---------------------------------|---|--------------------------------|--|--|------------|
| Pengujian tampil ubah informasi | Klik tombol menu ubah informasi | 1. Nama kos 2. Alamat 3. Kamar kosong 4. Gender 5. Harga 6. Info 7. Fasilitas | Muncul Pesan Register berhasil | Data berhasil diproses Tampilan menu sesuai yang diharapkan | Masuk halaman akses operator rumah sakit | Diterima |

5.5.8 Modul Pengujian Tampil Ubah Koordinat

Tabel 5.8. Pengujian modul tampil ubah koordinat

| Deskripsi | Prosedur pengujian | Masukkan | Keluaran yang diharapkan | Kriteria evaluasi hasil | Hasil yang didapat | Kesimpulan |
|---------------------------------|---------------------------------|---|--------------------------|--|------------------------|------------|
| Pengujian tampil ubah koordinat | Klik tombol menu ubah koordinat | 1. Cari kota 2. Latitude 3. Longitude | Data berhasil diproses | Data berhasil diproses Tampilan menu sesuai yang diharapkan | Perubahan data profile | Diterima |

5.5.9 Modul Pengujian Ubah Foto Profile

Tabel 5.9. Pengujian modul tampil ubah foto profile

| Deskripsi | Prosedur pengujian | Masukkan | Keluaran yang diharapkan | Kriteria evaluasi hasil | Hasil yang didapat | Kesimpulan |
|------------------------------------|------------------------------------|--|--|--|--|------------|
| Pengujian tampil ubah foto profile | Klik tombol menu ubah foto profile | 1. Hapus foto lama 2. <i>Browse</i> foto baru | Data berhasil diproses dan tampilan sesuai yang diinginkan | Data berhasil diproses Tampilan menu sesuai yang diharapkan | Data berhasil diproses dan tampilan sesuai yang diinginkan | Diterima |

5.5.10 Modul Pengujian Login Admin

Tabel 5.10. Pengujian modul tampil login admin

| Deskripsi | Prosedur pengujian | Masukkan | Keluaran yang diharapkan | Kriteria evaluasi hasil | Hasil yang didapat | Kesimpulan |
|------------------------|------------------------|----------------------------|--|--|--|------------|
| Pengujian tampil login | Klik tombol menu login | 1. Username 2. Password | Data berhasil diproses dan tampilan sesuai yang diinginkan | Data berhasil diproses Tampilan menu sesuai yang diharapkan | Data berhasil diproses dan tampilan sesuai yang diinginkan | Diterima |

5.5.11 Modul Pengujian Kelola Fasilitas

Tabel 5.11. Pengujian modul kelola fasilitas

| Deskripsi | Prosedur pengujian | Masukkan | Keluaran yang diharapkan | Kriteria evaluasi hasil | Hasil yang didapat | Kesimpulan |
|-----------------------------------|-----------------------------------|--------------------------------|--|--|--|-------------------|
| Pengujian tampil kelola fasilitas | Klik tombol menu kelola fasilitas | 1. Nama Fasilitas 2. Status | Data berhasil diproses dan tampilan sesuai yang diinginkan | Data berhasil diproses Tampilan menu sesuai yang diharapkan | Data berhasil diproses dan tampilan sesuai yang diinginkan | Diterima |

5.5.12 Modul Pengujian Kelola Member

Tabel 5.12. Pengujian modul kelola member

| Deskripsi | Prosedur pengujian | Masukkan | Keluaran yang diharapkan | Kriteria evaluasi hasil | Hasil yang didapat | Kesimpulan |
|--------------------------------|--------------------------------|-----------------|--|--|--|-------------------|
| Pengujian tampil kelola member | Klik tombol menu kelola member | 1. Satus | Data berhasil diproses dan tampilan sesuai yang diinginkan | Data berhasil diproses Tampilan menu sesuai yang diharapkan | Data berhasil diproses dan tampilan sesuai yang diinginkan | Diterima |

5.5.13 Modul Pengujian Kelola Rumah Kos

Tabel 5.13. Pengujian modul kelola rumah kos

| Deskripsi | Prosedur pengujian | Masukkan | Keluaran yang diharapkan | Kriteria evaluasi hasil | Hasil yang didapat | Kesimpulan |
|-----------------------------------|-----------------------------------|-----------------|--|--|--|-------------------|
| Pengujian tampil kelola rumah kos | Klik tombol menu kelola rumah kos | 1. Status | Data berhasil diproses dan tampilan sesuai yang diinginkan | Data berhasil diproses Tampilan menu sesuai yang diharapkan | Data berhasil diproses dan tampilan sesuai yang diinginkan | Diterima |

5.5.14 Modul Pengujian Ubah Email

Tabel 5.14. Pengujian modul ubah email

| Deskripsi | Prosedur pengujian | Masukkan | Keluaran yang diharapkan | Kriteria evaluasi hasil | Hasil yang didapat | Kesimpulan |
|-----------------------------|-----------------------------|------------------------------|--|--|--|-------------------|
| Pengujian tampil ubah email | Klik tombol menu ubah email | 1. Email baru 2. Password | Data berhasil diproses dan tampilan sesuai yang diinginkan | Data berhasil diproses Tampilan menu sesuai yang diharapkan | Data berhasil diproses dan tampilan sesuai yang diinginkan | Diterima |

5.5.15 Modul Pengujian Ubah Password

Tabel 5.15. Pengujian modul ubah password

| Deskripsi | Prosedur pengujian | Masukkan | Keluaran yang diharapkan | Kriteria evaluasi hasil | Hasil yang didapat | Kesimpulan |
|--------------------------------|--------------------------------|--|---|--|---|------------|
| Pengujian tampil ubah password | Klik tombol menu ubah password | 1. Password lama 2. Password baru 3. Ulangi Password | Data berhasil diproses dan tampilan sesuai yang di inginkan | Data berhasil diproses Tampilan menu sesuai yang diharapkan | Data berhasil diproses dan tampilan sesuai yang di inginkan | Diterima |

5.5.16 Modul Pengujian Pencarian

Tabel 5.16. Pengujian modul pencarian

| Deskripsi | Prosedur pengujian | Masukkan | Keluaran yang diharapkan | Kriteria evaluasi hasil | Hasil yang didapat | Kesimpulan |
|----------------------------|----------------------------|---|---|--|---|------------|
| Pengujian tampil pencarian | Klik tombol menu pencarian | 1. Nama daerah 2. Nama Kos 3. Jenis Kelamin 4. Bobot Harga 5. Bobot Kasur 6. Bobot Lemari 7. Bobot Meja Belajar | Data berhasil diproses dan tampilan sesuai yang di inginkan | Data berhasil diproses Tampilan menu sesuai yang diharapkan | Data berhasil diproses dan tampilan sesuai yang di inginkan | Diterima |

BAB VI

PENUTUP

Pada bab ini akan dipaparkan kesimpulan yang didapat dari pelaksanaan Tugas Akhir. Selain itu, disampaikan beberapa saran yang berguna untuk kelanjutan pengembangan topik yang diambil.

6.1 Kesimpulan

Beberapa hal yang dapat disimpulkan yaitu :

1. Penerapan metode eksponensial dalam sistem ini telah berhasil memberikan alternatif atau solusi keputusan yang baik bagi pencari rumah kost. Dengan bukti nilai hasil kuisioner yang telah dilakukan pada Lampiran B.
2. Menentukan lokasi rumah kost telah berhasil dilakukan dengan memanfaatkan *web service* dari Google Maps. Dengan menggunakan fitur *drag and drop* pengguna dapat dengan mudah menentukan dimana lokasi rumah kost berada.
3. Sistem pencarian rumah kost ini berhasil dibangun dengan kemampuan untuk memusatkan informasi rumah kost dengan menggunakan metode pencarian MPE berbasis PHP Codeigniter *Framework*.

6.2 Saran

Saran yang dapat diberikan penulis untuk pengembangan selanjutnya yaitu:

1. Menambahkan *e-commerce* pada sistem. Dengan adanya *e-commerce*, pengguna dapat melakukan pemesanan rumah kost secara langsung baik dengan menggunakan kartu kredit, *e-banking* ataupun media transaksi lainnya tanpa harus mendatangi suatu rumah kost.
2. Pengguna dapat mengakses informasi rumah kost pada suatu daerah melalui *sms gateway*. Dengan adanya fitur ini, pengguna menjadi lebih mudah mendapatkan informasi rumah kost yang paling disarankan pada daerah tertentu tanpa harus terhubung ke internet terlebih dahulu.

3. Menambahkan tipe file polygon pada peta google untuk menampilkan lokasi jalan yang tidak di tampilkan oleh google map.
4. Pada penentuan bobot harga sebaiknya dilakukan *survey* pada tiap daerah. Dengan melakukan *survey* ke setiap daerah maka akan didapat data bobot yang lebih akurat untuk digunakan dalam perhitungan metode perbandingan eksponensial.

DAFTAR PUSTAKA

- Anonim (t,t). *Rancang Bangun Rumah Kost* . Diunduh pada 5 September 2012 dari <http://repository.ipb.ac.id/bitstream/handle/123456789/20025/Bagian%20ISI%20PKM%20GT%20KIC.pdf?sequence=2>
- Anonim (t,t). *Teman Kampus – Memilih Tempat Kost Ideal* . Diunduh pada 5 September 2012 dari <http://temankampus.com/anak-kost/memilih-tempat-kost-ideal/>
- Caulica, Ezza Wilona. (2011). *Sistem Pendukung Keputusan Penerimaan Beasiswa Pemerintah Propinsi Riau Menggunakan Metode Perbandingan Eksponensial*. Pekanbaru: Politeknik Caltex Riau
- Hall, Shane. 2010. *How to Use the Likert Scale in Statistical Analysis*. Diunduh pada 31 Oktober 2012 dari http://www.ehow.com/how_4855078_use-likert-scale-statistical-analysis.html.
- Kadir, Abdul. (2011). *Buku Pintar JQUERY dan PHP*. Yogyakarta : MediaKom
- Kadir, Abdul. (2008). *Dasar pemrograman web dinamis menggunakan PHP*. Yogyakarta : Andi
- Marimin. (2009). *Teknik dan Aplikasi Pengambilan Keputusan Kriteria Majemuk*. Yogyakarta: Grasindo
- Markusic, Mayflor. 2009. *Simplifying the Likert Scale*. Diunduh pada 31 Oktober 2012 dari <http://www.brighthub.com/education/special/articles/13507.aspx>
- Santara, Yayan Riek, 2012, *Aplikasi Peta Penunjuk Lokasi Object Wisata di Kabupaten Wonogiri Menggunakan Macromedia Director MX*, Diunduh 25 November 2012 dari [http:// repository .amikom .ac .id /index .php / add-downloader/Publikasi 10.21.0490.pdf/306](http://repository.amikom.ac.id/index.php/add-downloader/Publikasi/10.21.0490.pdf/306)
- Subakti, Irfan. 2002. *Sistem Pendukung Keputusan*. Surabaya: Institut Teknologi Sepuluh Nopember.
- Sunyoto, Andi. (2007). *AJAX membangun web dengan teknologi Asynchroneuse JavaScript & XML*. Yogyakarta : Andi
- Trochim, William M.K.2006. *Likert Scaling -Research Methods Knowledge Based*. Diunduh pada 27 Oktober 2012 dari [http://www.anatomyfacts.com/Research/ResearchMethodsKnowledgeBase .pdf](http://www.anatomyfacts.com/Research/ResearchMethodsKnowledgeBase.pdf)

Winarno, Edy., Zaki, Ali.,Community, SmitDev. (2011). *Mudah Membuat Website dan e-Commerce dengan PHP Framework*. Semarang : Elex Media Komputindo

Wiswakarma, Komang. (2010). *9 Langkah Menjadi Master Framework Codeigniter*. Yogyakarta: Lokomedia