# SKRIPSI

**RANCANG BANGUN SISTEM PENDUKUNG KEPUTUSAN UNTUK REKOMENDASI PEMBELIAN RUMAH BERBASIS WEB MENGGUNAKAN METODE *SIMPLE ADDITIVE WEIGHTING ( SAW )***

**(STUDI KASUS : PERUMAHAN DI KOTA BEKASI)**

****

**EDI WARYONO**

**1570231078**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS KRISNADWIPAYANA**

**JAKARTA**

**2019**

**RANCANG BANGUN SISTEM PENDUKUNG KEPUTUSAN UNTUK REKOMENDASI PEMBELIAN RUMAH BERBASIS WEB MENGGUNAKAN METODE *SIMPLE ADDITIVE WEIGHTING ( SAW )***

**(STUDI KASUS : PERUMAHAN DI KOTA BEKASI)**

****

**Diajukan sebagai salah satu syarat untuk memperoleh gelar**

**Sarjana Komputer ( S.Kom )**

**EDI WARYONO**

**1570231078**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS KRISNADWIPAYANA**

**JAKARTA**

**2019**

# PERNYATAAN ORSINILITAS

Yang bertanda tangan dibawah ini, saya :

Nama : Edi Waryono

NIM : 1570231078

Program Studi : Teknik Informatika

Jenjang : Strata Satu (S1)

Jenis Karya : Skripsi

Dengan ini menyatakan bahwa isi skripsi yang telah saya buat dengan judul : ***“Rancang Bangun Sistem Pendukung Keputusan Untuk Rekomendasi Pembelian Rumah Berbasis Web Menggunakan Metode Simple Additive Weighting ( SAW ) (Studi Kasus : Perumahan Di Kota Bekasi)”*** adalah hasil karya sendiri, semua sumber baik yang di kutip maupun yang di rujuk telah di tuliskan sumbernya secara jelas sesuai dengan norma, kaidah dan etika penulisan ilmiah dan skripsi belum pernah diterbitkan atau dipublikasikan dimanapun dan dalam bentuk apapun.

Demikian surat pernyataan ini saya buat dengan sebenar-benarnya. Apabila dikemudian hari ternyata saya memberikan keterangan palsu dan atau ada pihak lain yang mengklaim bahwa skripsi yang telah saya buat adalah milik seseorang atau badan tertentu, saya bersedia diproses baik secara pidana maupun perdata dan kelulusan saya dari program Teknik Informatika dan Fakultas Teknik Universitas Krisnadwipayana dicabut atau dibatalkan.

Jakarta, 21 Agustus 2019

Yang menyatakan,

Edi Waryono

1570231078

# HALAMAN PENGESAHAN

Tugas Akhir atau Skipsi ini diajukan oleh :

Nama : Edi Waryono

NIM : 1570231078

Program Studi : Teknik Informatika

Jenjang : Strata Satu (S1)

Judul Skripsi : Rancang Bangun Sistem Pendukung Keputusan Untuk Rekomendasi Pembelian Rumah Berbasis Web Menggunakan *Metode Simple Additive Weighting ( Saw )* (Studi Kasus : Perumahan Di Kota Bekasi)

Telah berhasil di pertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Komputer (S.Kom) pada Program Studi Teknik Informatika Fakultas Teknik Universitas Krisnadwipayana Jakarta.

**DEWAN PENGUJI**

|  |  |  |
| --- | --- | --- |
| Penguji I | : Junaidi, M.Kom | …………………………. |
|  |  |  |
| Penguji II | : Nurhikmah, ST, M.Kom | …………………………. |
|  |  |  |
| Penguji III | : Wargijono Utomo, S.Kom, M.Kom | …………………………. |
|  |  |  |
| Penguji IV/  Pembimbing | : Dr. Harjono P. Putro, ST, M.Kom | …………………………. |

# HALAMAN PERSETUJUAN

Tugas Akhir atau Skripsi ini diajukan oleh:

Nama : Edi Waryono

NIM : 1570231078

Program Studi : Teknik Informatika

Jenjang : Strata Satu (S1)

Judul Skripsi : Rancang Bangun Sistem Pendukung Keputusan Untuk Rekomendasi Pembelian Rumah Berbasis Web Menggunakan *Metode Simple Additive Weighting ( Saw )* (Studi Kasus : Perumahan Di Kota Bekasi)

**MENYETUJUI,**

Jakarta, 21 Agustus 2019

|  |  |
| --- | --- |
| DOSEN PEMBIMBING 1 | DOSEN PEMBIMBING 2 |
|  |  |
| Dr. HARJONO P. PUTRO, ST, M.Kom | M. SYARIF H, S.Kom, M.Kom |

**MENGETAHUI,**

Ketua Program Studi Teknik Informatika

ALI KHUMAIDI, S.Kom, M.Kom

# PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS

Yang bertanda tangan dibawah ini, saya :

Nama : Edi Waryono

NIM : 1570231078

Program Studi : Teknik Informatika

Jenjang : Strata Satu (S1)

Jenis Karya : Skripsi

Demi pengembangan ilmu pengetahuan, dengan ini saya menyetujui untuk memberikan ijin kepada pihak Universitas Krisnadwipayana Jakarta Hak bebas non-eksklusif *(Non-exlusive Royalti-free Right)* Atas karya ilmiah yang berjudul ***Rancang Bangun Sistem Pendukung Keputusan Untuk Rekomendasi Pembelian Rumah Berbasis Web Menggunakan Metode Simple Additive Weighting ( Saw ) (Studi Kasus : Perumahan Di Kota Bekasi)*** beserta perangkat yang diperlukan.

Dengan **Hak Bebas Royalti Non-Ekslusif** ini pihak Universitas Krisnadwipayana berhak menyimpan, mengalih-media atau bentuk-kan, pengelolaannya dalam pangkalan data *(database),* mendistribusikan dan menampilkan atau mempublikasikannya di internet atau media lain untuk kepentingan akademis tanpa perlu meminta ijin dari kami selama tetap mencantumkan nama kami sebagai penulis/pencipta karya ilmiah tersebut. Saya bersedia untuk menanggung secara pribadi, tanpa melibatkan pihak Universitas Krisnadwipayana, segala bentuk tuntutan hukum yang timbul atas pelanggaran Hak Cipta dalam karya ilmiah saya ini. Demikian pernyataan saya buat dengan sebenarnya.

Jakarta, 21 Agustus 2019

Yang menyatakan,

Edi Waryono

1570231078

UNIVERSITAS KRISNADWIPAYANA

FAKULTAS TEKNIK

PROGRAM STUDI TEKNIK INFORMATIKA

PEMINATAN REKAYASA PERANGKAT LUNAK

Skripsi, Agustus 2019

Edi Waryono, NIM. 1570231078

“Rancang Bangun Sistem Pendukung Keputusan Untuk Rekomendasi Pembelian Rumah Berbasis Web Menggunakan Metode *Simple Additive Weighting ( Saw )* (Studi Kasus : Perumahan Di Kota Bekasi)”

xv + 121 halaman, 58 gambar, 28 tabel

# ABSTRAK

Kebutuhan informasi tentang rumah yang semakin meningkat dan semakin bertambahnya jumlah pengembang perumahan di kota bekasi menghasilkan banyaknya alternatif rumah yang ditawarkan kepada calon pembeli. Hal itu membuat calon pembeli merasa kesulitan dalam menentukan pembelian rumah, calon pembeli rumah berharap mendapatkan rumah yang terbaik sesuai dengan kebutuhan nya. Untuk menentuan pembelian rumah calon pembeli perlu memperhatikan beberapa kriteria diantara nya adalah harga, jarak lokasi rumah ke pusat kota, akses menuju perumahan, legalitas rumah, fasilitas umum sekitar rumah, dan desain rumah suatu rumah. Berdasarkan masalah tersebut perlu adanya sistem pendukung keputusan sebagai alternatif pemecahan masalah bagi calon pembeli, sistem pendukung keputusan ini bertujuan merekomendasikan rumah yang sesuai dengan kebutuhan calon pembeli dan diharapkan menjadi alat bantu dalam proses pengambilan keputusan pembelian rumah yang sesuai dengan kebutuhan calon pembeli. Metode yang diterapkan dalam pembangunan *website* ini adalah metode *Simple Additive Weighting (SAW).* Hasil dari penelitian ini adalah sebuah sistem pendukung keputusan yang dapat mengelola data kriteria dan alternatif untuk dijadikan pendukung keputusan oleh calon pembeli.

**Kata kunci**

*Sistem Pendukung Keputusan, Simple Additive Weighting (SAW), Waterfall, UML.*

*KRISNADWIPAYANA UNIVERSITY*

*FACULTY OF ENGINEERING*

*INFORMATICS ENGINEERING STUDY PROGRAM SOFTWARE ENGINEERING*

*Thesis, August 2019*

Edi Waryono, NIM. 1570231078

*"* *Design and Build a Decision Support System for Web-Based House Purchase Recommendations Using the Simple Additive Weighting (Saw) Method (Case Study: Housing in Bekasi City)"*

*xv + 121 pages, 58 images, 28 tables*

# *ABSTRACT*

*The need for information about homes is increasing and the increasing number of housing developers in the city of Bekasi produces many alternative houses offered to prospective buyers. This makes prospective buyers find it difficult to determine the purchase of a house, prospective home buyers hope to get the best home according to his needs. To determine the purchase of a house prospective buyers need to consider several criteria including price, distance of the location of the house to the city center, access to housing, legality of the house, public facilities around the house, and home design of a house. Based on these problems the need for a decision support system as an alternative problem solving for prospective buyers, this decision support system aims at recommending a home that suits the needs of prospective buyers and is expected to be a tool in the decision-making process of purchasing a home that suits the needs of prospective buyers. The method applied in the construction of this website is the Simple Additive Weighting (SAW) method. The results of this study are a decision support system that can manage criteria and alternative data to be used as decision support by prospective buyers.*

***Keywords***

*Decision Support System, Simple Additive Weighting (SAW), Waterfall, UML.*

# KATA PENGHANTAR

Puji syukur alhamdulillah, penulis panjatkan kehadirat Allah, SWT, yang telah melimpahkan rahmat dan karunia-nya, sehingga pada akhir penulis dapat menyelesaikan skripsi ini tepat pada waktunya. Skripsi ini penulis sajikan dalam bentuk buku yang sederhana. Adapun judul skripsi, yang penulis ambil sebagai berikut “Rancang Bangun Sistem Pendukung Keputusan Untuk Rekomendasi Pembelian Rumah Berbasis Web Menggunakan Metode *Simple Additive Weighting ( Saw )* (Studi Kasus : Perumahan Di Kota Bekasi)”.

Tujuan penulisan skripsi ini adalah di buat sebagai salah satu untuk mendapatkan gelar Sarjana Komputer (S.Kom) pada Program Studi Teknik Informatika Fakultas Teknik Universitas Krisnadwipayana Jakarta.

Skripsi ini diambil berdasarkan penelitian mengenai Rancang Bangun Sistem Pendukung Keputusan Untuk Rekomendasi Pembelian Rumah Berbasis Web Menggunakan Metode S*imple Additive Weighting ( Saw )* (Studi Kasus : Perumahan Di Kota Bekasi). Penulis juga melakukan, mencari dan menganalisa berbagai macam sumber referensi, baik dalam bentuk jurnal ilmiah, buku-buku literatur, internet, dll yang terkait.

Penulis menyadari bahwa tanpa bimbingan dan dukungan dari semua pihak dalam pembuatan skripsi ini , maka penulis tidak dapat menyelesaikan laporan skripsi ini tepat pada waktunya. Untuk ini ijinkanlah penulis kesempatan ini untuk mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Bapak Dr. Harjono P. Putro, ST, M.Kom selaku pembimbing satu skripsi yang telah menyediakan waktu, pikiran dan tenaga dalam membimbing penulis dalam menyelesaikan skripsi ini.
2. Bapak M. Syarif Hartawan, S.Kom, M.Kom selaku pembimbing dua skripsi yang telah menyediakan waktu, pikiran dan tenaga dalam membimbing penulis dalam menyelesaikan skripsi ini.
3. Bapak Ali Khumaidi, S.Kom, M.Kom selaku Ketua Program Studi Teknik Informatika Universitas Krisnadwipayana.
4. Ibu Nuke L. Chusna, S.Si, M.Kom selaku Sektrtaris Ketua Program Studi Teknik Informatika Universitas Krisnadwipayana.
5. Bapak Junaidi, M.Kom selaku penguji satu Program Studi Teknik Informatika Universitas Krisnadwipayana.
6. Ibu Nurhikmah, ST, M.Kom selaku penguji dua Program Studi Teknik Informatika Universitas Krisnadwipayana.
7. Bapak Wargijono Utomo, S.Kom, M.Kom selaku penguji tiga Program Studi Teknik Informatika Universitas Krisnadwipayana.
8. Seluruh staf pengajar (dosen) Program Studi Teknik Informatika Fakultas Teknik Universitas Krisnadwipayana yang telah memberikan pelajaran yang berarti bagi penulis selama menempuh studi.
9. Orang tua tercinta yang telah memberikan dukungan material dan moral kepada penulis.
10. Seluruh staf dan karyawan Program Studi Teknik Informatika Fakultas Teknik Universitas Krisnadwipayana yang telah melayani penulis dengan baik selama kuliah.

Serta semua pihak yang terlalu banyak untuk penulis sebutkan satu persatu sehingga terwujudnya penulisan skripsi ini. Penulis menyadari bahwa penulisan Laporan ini masih jauh sekali dari sempurna, untuk itu penulis mohon kritik dan saran bersifat membangun demi kesempurnaan penulis karya ilmiah yang penulis hasilkan untuk yang akan datang.

Akhir kata semoga skripsi ini dapat bermanfaat bagi penulis khususnya dan bagi para pembaca yang berminat pada umumnya.

Jakarta, 21 Agustus 2019

Penulis,

Edi Waryono

NIM. 1570231078

# DAFTAR ISI

Halaman

[HALAMAN SAMPUL i](#_Toc18618156)

HALAMAN JUDUL ii

[PERNYATAAN ORSINILITAS iii](#_Toc18618157)

[HALAMAN PENGESAHAN iv](#_Toc18618158)

[HALAMAN PERSETUJUAN v](#_Toc18618159)

[PERSETUJUAN PUBLIKASI … vi](#_Toc18618160)

[ABSTRAK vii](#_Toc18618161)

[*ABSTRACT* viii](#_Toc18618162)

[KATA PENGHANTAR ix](#_Toc18618163)

[DAFTAR ISI xi](#_Toc18618164)

[DAFTAR GAMBAR xiii](#_Toc18618165)

[DAFTAR TABEL xv](#_Toc18618166)

[BAB I 1](#_Toc18618167)

[1.1 Latar Belakang 1](#_Toc18618168)

[1.2 Identifikasi Masalah 3](#_Toc18618169)

[1.3 Tujuan Penelitian 3](#_Toc18618170)

[1.4 Ruang Lingkup Penelitian 3](#_Toc18618171)

[1.5 Metodologi Penelitian 4](#_Toc18618172)

[1.5.1 Teknik Pengumpulan Data 4](#_Toc18618173)

[1.5.2 Metode Pengembangan 5](#_Toc18618174)

[BAB II 7](#_Toc18618175)

[2.1 Tinjauan Pustaka 7](#_Toc18618176)

[2.2 Tinjauan Studi 11](#_Toc18618177)

[*2.2.1* *Website* 11](#_Toc18618178)

[2.2.2 Sistem Pendukung Keputusan 14](#_Toc18618179)

[2.2.3 Rekayasa Perangkat Lunak 17](#_Toc18618180)

[*2.2.4* *Database* 32](#_Toc18618181)

[2.2.5 Alat Pengembangan Perangkat Lunak 35](#_Toc18618182)

[BAB III 41](#_Toc18618183)

[3.1 Analisis Kebutuhan Sistem 41](#_Toc18618184)

[3.1.1 Kebutuhan fungsional 41](#_Toc18618185)

[3.1.2 Kebutuhan non-fungsional 41](#_Toc18618186)

[3.1.3 Perancangan *Simple Additive Weighting (SAW)* 42](#_Toc18618187)

[3.1.4 Perancangan Proses Pendukung Keputusan 44](#_Toc18618188)

[3.1.5 Proses Perhitungan *Simple Additive Weighting (SAW)* 46](#_Toc18618189)

[3.2 Sistem Desain 54](#_Toc18618190)

[3.2.3 *Use Case Diagram* 54](#_Toc18618191)

[3.2.4 *Activity Diagram* 57](#_Toc18618192)

[3.2.5 *Sequence Diagram* 66](#_Toc18618193)

[3.2.6 *Class Diagram* 74](#_Toc18618194)

[3.2.7 Rancangan *Database* 75](#_Toc18618195)

[3.2.8 Rancangan *Interface* 78](#_Toc18618196)

[BAB IV 87](#_Toc18618197)

[4.1 Implementasi 87](#_Toc18618198)

[4.1.1 Implementasi *Database* 87](#_Toc18618199)

[4.1.2 Implementasi *Interface* 90](#_Toc18618200)

[4.1.3 Implementasi Metode SAW pada Sistem Pendukung Keputusan 113](#_Toc18618201)

[4.2 Pengujian Sistem 115](#_Toc18618202)

[4.2.1 Pengujian *White Box* 115](#_Toc18618203)

[4.2.2 Pengujian *Black Box* 118](#_Toc18618204)

[BAB V 120](#_Toc18618205)

[5.1 Kesimpulan 120](#_Toc18618206)

[5.2 Saran 120](#_Toc18618207)

[DAFTAR PUSTAKA 121](#_Toc18618208)

# DAFTAR GAMBAR

[**Gambar 2. 1** *Waterfall* Pressman (Pressman:2015:42) 17](#_Toc16329057)

[**Gambar 2. 2** *Use-Case Diagram* Sri Mulyani, (2016:109) 21](#_Toc16329058)

[**Gambar 2. 3** *Class Diagram* Sri Mulyani (2016:248) 22](#_Toc16329059)

[**Gambar 2. 4** *Sequence Diagram* Rosa dan M. Shalahudin, (2015:165). 24](#_Toc16329060)

[**Gambar 2. 5** *Activity Diagram* Rosa dan M. Shalahudin (2015:161). 26](#_Toc16329061)

[**Gambar 3. 1** Proses Pengambilan Keputusan 45](#_Toc18571241)

[**Gambar 3. 2** *Use Case Diagram* 54](#_Toc18571242)

[**Gambar 3. 3** *Activity Diagram* Mencari Alternatif Rumah 57](#_Toc18571243)

[**Gambar 3. 4** *Activity Diagram* Data Perumahan 58](#_Toc18571244)

[**Gambar 3. 5** *Activity Diagram* Melihat Panduan Penggunaan SPK 58](#_Toc18571245)

[**Gambar 3. 6** *Activity Diagram* Tentang SPK 59](#_Toc18571246)

[**Gambar 3. 7** *Activity Diagram Login* 60](#_Toc18571247)

[**Gambar 3. 8** *Activity Diagram* Data Nilai 61](#_Toc18571248)

[**Gambar 3. 9** *Activity Diagram* Kelola Data Kriteria 62](#_Toc18571249)

[**Gambar 3. 10** *Activity Diagram* Data Alternatif Rumah 63](#_Toc18571250)

[**Gambar 3. 11** *Activity Diagram* Data Proses Perangkingan 64](#_Toc18571251)

[**Gambar 3. 12** *Activity Diagram* Mencetak Laporan 65](#_Toc18571252)

[**Gambar 3. 13** *Sequence Diagram* Mencari Alternatif Rumah 66](#_Toc18571253)

[**Gambar 3. 14** *Sequence Diagram* Melihat Data Perumahan 67](#_Toc18571254)

[**Gambar 3. 15** *Sequence Diagram* Melihat Panduan 67](#_Toc18571255)

[**Gambar 3. 16** *Sequence Diagram* Melihat Tentang 68](#_Toc18571256)

[**Gambar 3. 17** *Sequence Diagram* Login Admin 69](#_Toc18571257)

[***Gambar 3. 18*** *Sequence Diagram Mengelola Data Nilai* 70](#_Toc18571258)

[**Gambar 3. 19** *Sequence Diagram* Mengelola Data Kriteria 71](#_Toc18571259)

[**Gambar 3. 20** *Sequence Diagram* Mengelola Data Kriteria 72](#_Toc18571260)

[**Gambar 3. 21** *Sequence Diagram* Mengelola Data Proses Perangkingan 73](#_Toc18571261)

[**Gambar 3. 22** *Sequence Diagram* Cetak Laporan 74](#_Toc18571262)

[**Gambar 3. 23** *Class Diagram* 75](#_Toc18571263)

[**Gambar 3. 24** Rancangan *Interface* halaman utama *Frontend* 79](#_Toc18571264)

[**Gambar 3. 25** Rancangan Menu Cari Rumah 80](#_Toc18571265)

[**Gambar 3. 26** Rancangan Hasil Rekomendasi Rumah 81](#_Toc18571266)

[**Gambar 3. 27** Rancangan Menu Peta Lokasi 81](#_Toc18571267)

[**Gambar 3. 28** Rancangan Menu Panduan 82](#_Toc18571268)

[**Gambar 3. 29** Rancangan Menu Tentang 82](#_Toc18571269)

[**Gambar 3. 30** Rancangan Menu Login 83](#_Toc18571270)

[**Gambar 3. 31** Rancangan Halaman Utama *Backend* 83](#_Toc18571271)

[**Gambar 3. 32** Rancangan Menu Nilai 84](#_Toc18571272)

[**Gambar 3. 33** Rancangan Menu Kriteria 84](#_Toc18571273)

[**Gambar 3. 34** Rancangan Menu Alternatif 85](#_Toc18571274)

[**Gambar 3. 35** Rancangan Menu Perangkingan 85](#_Toc18571275)

[**Gambar 3. 36** Rancangan Menu Laporan 86](#_Toc18571276)

[**Gambar 4. 1** Implementasi *Interface* Halaman Utama *frontend* 90](#_Toc18571277)

[**Gambar 4. 2** Implementasi Menu Cari Rumah 93](#_Toc18571278)

[**Gambar 4. 3** Hasil Rekomendasi Rumah 96](#_Toc18571279)

[**Gambar 4. 4** Implementasi Menu Peta Lokasi 97](#_Toc18571280)

[**Gambar 4. 5** Implementasi Menu Panduan 99](#_Toc18571281)

[**Gambar 4. 6** Implementasi Menu Tentang 100](#_Toc18571282)

[**Gambar 4. 7** Implementasi Menu *Login* 101](#_Toc18571283)

[**Gambar 4. 8** Implementasi *interface* Halaman Utama *backend* 102](#_Toc18571284)

[**Gambar 4. 9** Implementasi Menu Nilai 103](#_Toc18571285)

[**Gambar 4. 10** Implementasi Menu Kriteria 105](#_Toc18571286)

[**Gambar 4. 11** Implementasi Menu Alternatif 107](#_Toc18571287)

[**Gambar 4. 12** Implementasi Menu Perangkingan 109](#_Toc18571288)

[**Gambar 4. 13** Implementasi Menu Laporan 110](#_Toc18571289)

[**Gambar 4. 14** Laporan Nilai dan Normalisasi R 111](#_Toc18571290)

[**Gambar 4. 15** Laporan Hasil Perhitungan SAW 111](#_Toc18571291)

[**Gambar 4. 16** *Flowgraph Login* 116](#_Toc18571292)

# DAFTAR TABEL

[**Tabel 2. 1** Notasi *Use Case Diagram* 21](#_Toc16328813)

[**Tabel 2. 2** Notasi *Class Diagram* 23](#_Toc16328814)

[**Tabel 2. 3** Notasi *Sequence Diagram* 25](#_Toc16328815)

[**Tabel 2. 4** Notasi *Diagram Activity* 27](#_Toc16328816)

[**Tabel 3. 1** Penentuan Kriteria 44](#_Toc18617716)

[**Tabel 3. 2** Pembobotan Kriteria 46](#_Toc18617717)

[**Tabel 3. 3** Nilai Ranting Masing - Masing Kriteria 47](#_Toc18617718)

[**Tabel 3. 4** Data Alternatif Rumah 48](#_Toc18617719)

[**Tabel 3. 5** Normalisasi Kriteria 51](#_Toc18617720)

[**Tabel 3. 6** Hasil Perangkingan 53](#_Toc18617721)

[**Tabel 3. 7** Rancangan Alternatif 76](#_Toc18617722)

[**Tabel 3. 8** Rancangan Kriteria 76](#_Toc18617723)

[**Tabel 3. 9** Rancangan Nilai 76](#_Toc18617724)

[**Tabel 3. 10** Rancangan Pengguna 77](#_Toc18617725)

[**Tabel 3. 11** Rancangan Pengunjung 77](#_Toc18617726)

[**Tabel 3. 12** Rancangan Rangking 77](#_Toc18617727)

[**Tabel 3. 13** Rancangan Tentang 78](#_Toc18617728)

[**Tabel 3. 14** Rancangan Panduan 78](#_Toc18617729)

[**Tabel 4. 1** Implementasi Alternatif 87](#_Toc18571231)

[**Tabel 4. 2** Implementasi Kriteria 87](#_Toc18571232)

[**Tabel 4. 3** Implementasi Nilai 88](#_Toc18571233)

[**Tabel 4. 4** Implementasi Penguna 88](#_Toc18571234)

[**Tabel 4. 5** Implementasi Pengunjung 88](#_Toc18571235)

[**Tabel 4. 6** Implementasi Rangking 89](#_Toc18571236)

[**Tabel 4. 7** Implementasi Tentang 89](#_Toc18571237)

[**Tabel 4. 8** Implementasi Panduan 89](#_Toc18571238)

[**Tabel 4. 9** Tabel Pengujian *White Box Login.php* 115](#_Toc18571239)

[**Tabel 4. 10** Pengujian *Black Box* 118](#_Toc18571240)

# BAB I

**PENDAHULUAN**

## Latar Belakang

Kebutuhan informasi tentang rumah yang semakin meningkat dan pertumbuhan perumahan yang semakin banyak memunculkan banyaknya alternatif rumah yang ditawarkan oleh pengembang perumahan kepada calon pembeli, oleh karena itu calon pembeli merasa kesulitan dalam menentukan pemilihan rumah dan calon pembeli menginginkan rumah yang terbaik sesuai dengan kebutuhan dari banyaknya alternatif rumah yang di tawarkan oleh pengembang perumahan. Pemilihan rumah yang sesuai kebutuhan tidaklah mudah, dengan banyak sekali alternatif rumah yang ditawarkan oleh pengembang perumahan oleh karena itu dalam pemilihan rumah ada beberapa kriteria yang harus di perhatikan calon pembeli seperti harga rumah, luas tanah, luas bangunan, jumlah kamar tidur, jumlah kamar mandi, dan daya litrik rumah. Kriteria- kriteria tersebut harus diperhatikan agar mendapatkan pilihan rumah yang sesuai dengan kebutuhan dan tidak menyesal setelah menentukan pilihan rumah. Berdasarkan masalah tersebut dibutuhkan adanya sebuah sistem pendukung keputusan sebagai alternatif pemecahan masalah bagi calon pembeli, sistem pendukung keputusan ini bertujuan untuk merekomendasikan rumah yang sesuai dengan kebutuhan calon pembeli.

Sistem Pendukung Keputusan (SPK) adalah suatu sistem informasi spesifik yang ditujukan untuk membantu manajemen dalam mengambil keputusan yang berkaitan dengan persoalan yang bersifat semi terstruktur. Sistem ini memiliki fasilitas menghasilkan berbagai alternatif yang secara interaktif digunakan oleh pemakai. Sistem Pendukung Keputusan (SPK) biasanya dibangun untuk mendukung solusi atas suatu masalah atau suatu peluang (Dicky Nofriansyah & Sarjon Defit, 2017:2). Sistem Pendukung Keputusan digunakan untuk mempermudah pengambilan keputusan dalam menentukan sebuah pilihan dari berbagai alternatif – alternatif yang tersedia dari hasil pengolahan data dan informasi. Masalah pemilihan rumah memiliki karakteristik sebagai persoalan MADM *(Multiple Attribute Decision Making),* MADM memiliki kemampuan untuk melakukan penilaian secara lebih tepat karena didasarkan pada nilai kriteria dan bobot prefensi yang sudah ditentukan, kemudian dilanjutkan dengan proses perangkingan yang akan menyeleksi alternatif terbaik dari sejumlah alternatif yang ada (Nugroho & Wulandari, 2016). Dalam sebuah sistem pendukung keputusan terdapat beberapa jenis solusi pemecahan masalah diantaranya yaitu: *Multi Attribute Decision Making (MADM)* seperti: metode *Simple Additive Weighting (SAW)*. *Weighted Product (WP), Elimination Et Choix Traduisant la realite (ELECTRE), Technique for Order Preference by Similarity to Ideal Solution (TOPSIS),* dan *Analytic Hierarchy Process (AHP).* (Dicky Nofriansyah & Sarjon Defit, 2017:5).

Penyelesaian masalah dengan *Multi Attribute Decision Making (MADM)* dapat dilakukan dengan beberapa metode yang digunakan, salah satunya yaitu dengan menggunakan metode *Simple Additive Weighting (SAW)*. Metode *simple additive weighting* dapat diartikan sebagai metode pembobotan pada penyelesaian masalah dalam sebuah sistem pendukung keputusan. Konsep metode ini adalah dengan mencari rating kinerja (skala prioritas) pada setiap alternatif disemua atribut (Dicky Nofriansyah & Sarjon Defit,2017:33). Pemilihan metode *simple additive weighting (saw)* dikarenakan metode *simple additive weighting (saw)* dapat melakukan penilaian secara lebih tepat pada sistem pendukung keputusan, berdasarkan nilai kriteria dan bobot preferensi yang sudah ditentukan sebelumnya. Selain itu metode *simple additive weighting (saw)* mampu menyeleksi alternatif terbaik dari sejumlah alternatif yang ada.

Lokasi penelitian Sistem Pendukung Keputusan ini mengambil lokasi di kota Bekasi. Pengambilan lokasi di kota bekasi dikarenakan banyaknya perkembangan perumahan yang semakin banyak di bangun di daerah kota bekasi. Fasilitas umum yang dimiliki kota bekasi sangat lengkap dengan terdapat banyak lapangan kerja, sektor hunian, pusat perbelanjaan, sarana pendidikan, sarana kesehatan, dan fasilitas lainnya. Tidak hanya itu, akses yang dimiliki Bekasi pun sangat mudah dengan adanya tol Jakarta - Cikampek. Menurut data yang dihimpun oleh UrbanIndo, terdapat beberapa daerah yang paling banyak dicari di Bekasi. Daerah tersebut meliputi Medan Satria, Jatiasih, Bekasi Utara, Babelan, Serang baru, Pondok Gede, Tambun Utara, dan beberapa daerah lainnya.

## Identifikasi Masalah

Berdasarkan latar belakang yang telah dijabarkan di atas, maka permasalahan yang ada dalam penyusunan skripsi ini yaitu:

1. Calon pembeli yang dihadapkan banyaknya pilihan perumahan yang di tawarkan oleh pengembang perumahan dengan berbagai kriteria yang di tawarkan kepada calon pembeli perlu adanya sebuah aplikasi yang dapat mempermudah proses keputusan untuk membeli rumah.
2. Banyaknya kriteria – kriteria yang perlu di perhatikan dalam menentukan pilihan rumah yang sesuai dengan kebutuhan calon pembeli agar tidak menyesal dan menimbulkan permasalahan setelah melakukan pembelian rumah.
3. Pemilihan metode MADM *(Multiple Attribute Decision Making)* dalam membangun sistem pendukung keputusan dengan menggunakan metode *Simple Additive Weighting (SAW).*

## Tujuan Penelitian

Melihat fungsi dan tujuan sistem pendukung keputusan yang berkaitan dengan pembelian rumah yang membutuhkan berbagai kriteria maka tujuan penelitian ini adalah membangun sistem pendukung keputusan untuk rekomendasi pembelian rumah berbasis web menggunakan *metode simple additive weighting ( saw )* (studi kasus : perumahan di kota bekasi) dengan memberikan alternatif rumah untuk calon pembeli dalam menentukan pembelian rumah yang sesuai dengan kebutuhan calon pembeli.

## Ruang Lingkup Penelitian

Untuk memperjelas masalah yang akan dibahas dan agar tidak terjadi pembahasan yang meluas atau menyimpang, maka perlu kiranya dibuat suatu batasan masalah. Adapun ruang lingkup permasalahan yang akan dibahas dalam penulisan skripsi ini, yaitu sebagai berikut :

1. Sistem pendukung keputusan ini tidak memberikan fitur jual beli rumah sehingga tidak dapat melakukan transaksi pembelian rumah oleh calon pembeli.
2. Penelitian ini menggunakan metode pengembangan *waterfall* dengan melalui tahapan *requirement* (analisis kebutuhan), *design system* (desain sistem), *Coding* (pengkodean) & *Testing* (pengujian).
3. Data rumah yang digunakan dalam penelitian ini adalah data sekunder hasil dari observasi dengan menyajikan data hasil survei yaitu 10 perumahan yang ada di kota bekasi.
4. Kriteria yang digunakan untuk SPK pembelian rumah ini adalah harga, luas tanah, luas bangunan, jumlah kamar tidur, jumlah kamar mandi, dan daya listrik.

## Metodologi Penelitian

Untuk memperoleh data - data yang diperlukan untuk menunjang kesempurnaan dan kelengkapan laporan skripsi ini maka penulis menggunakan metode dan teknik penulisan sebagai berikut:

### Teknik Pengumpulan Data

Dalam mencapai tujuan penelitian perlu adanya proses pengumpulan data dan informasi perumahan yang dilakukan di kota bekasi. Teknik pengumpulkan data yang di lakukan adalah dengan 3 cara sebagai berikut:

1. Observasi

Observasi adalah sebagai satu proses wawasan sempit, yaitu memperhatikan sesuatu dengan menggunakan mata. Dalam pengertian psikologis, observasi atau yang disebut pengamatan, meliputi kegiatan pemuatan perhatian terhadap suatu objek dengan menggunakan seluruh alat indera. Jadi observasi adalah sebuah pengamatan secara langsung (Kesuma, 2017:16). Penelitian ini mengandung data yang telah didapat dari pengembang perumahan yang berada di kota bekasi yang telah membangun rumah - rumah yang siap dipasarkan kepada calon pembeli.

1. Wawancara

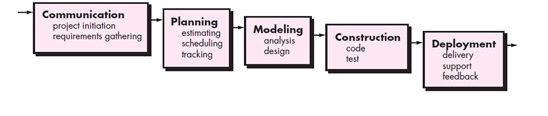
Wawancara dapat diartikan sebagai suatu proses memperoleh keterangan untuk tujuan penelitian dengan cara tanya jawab sambil bertatap muka antara pewawancara dengan informan. Teknik ini dilakukan dengan cara tanya jawab atau percakapan secara langsung. (Kesuma, 2017:16). Penelitian ini mengadakan wawancara dengan pihak pihak yang dapat memberikan informasi yang berhubungan dengan materi pembahasan seperti calon pembeli dan pengembang rumah yang ada di kota bekasi.

1. Studi Pustaka

"Metode penelitian pada dasarnya merupakan cara ilmiah untuk mendapatkan data dengan tujuan dan kegunaan tertentu." Dalam penelitian ini, penulis menggunakan metode kuantitatif dengan pendekatan penelitian deskriptif dan analisis asosiatif, karena adanya variabel - variabel yang akan ditelaah hubungannya serta tujuannya untuk menyajikan gambaran mengenai hubungan antara variabel-variabel yang diteliti (Sugiyono, 2017:2). Penelitian ini adalah hasil dari mempelajari dan mengumpulkan bahan - bahan melalui buku-buku dan internet yang berhubungan dengan pokok pembahasan yang akan dibahas dalam laporan skripsi ini.

### Metode Pengembangan

Model *waterfall* adalah model klasik yang bersifat sistematis , berurutan dalam membangun sebuah *software* . Nama model ini sebenarnya adalah "*Linear Sequential Model*" . Model ini termasuk ke dalam model generic pada rekayasa perangkat lunak dan pertama kali model *waterfall* ini diperkenalkan oleh Winston Royce pada tahun 1970 . Meskipun sering dianggap kuno , tetapi model waterfall ini merupakan model yang paling banyak dipakai dalam *Software Engineering (SE)* . Model *Waterfall* ini melakukan pendekatan secara sistematis dan berurutan . Model ini disebut dengan model *Waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan. (Pressman, 2015:42).



**Gambar 1. 1** Waterfall Pressman (Pressman:2015:42)

Dalam pengembangannya metode *waterfall* memiliki beberapa tahapan yang berurut yaitu :

A . *Communication (Project Initiation & Requirements Gathering)*

Sebelum memulai pekerjaan yang bersifat teknis, sangat diperlukan adanya komunikasi dengan *customer* demi memahami dan mencapai tujuan yang ingin dicapai. Hasil dari komunikasi tersebut adalah inisialisasi proyek, seperti menganalisis permasalahan yang dihadapi dan mengumpulkan data-data yang diperlukan, serta membantu mendefinisikan fitur dan fungsi *software*. Pengumpulan data-data tambahan bisa juga diambil dari jurnal, artikel dan internet.

B . *Planning (Estimating , Schedulling , Tracking)*

Tahap selanjutnya adalah tahap perencanaan yang menjelaskan mengenai estimasi tugas - tugas teknis yang akan dilakukan, resiko - resiko yang dapat terjadi , sumber daya yang diperlukan dalam membuat sistem, produk kerja yang ingin dihasilkan, penjadwalan kerja yang akan dilaksananakan, dan *tracking* proses pengerjaan sistem .

C . *Modelling (Analysis & Design)*

Tahap ini adalah tahap perancangan dan pemodelan arsitektur sistem yang berfokus pada perancangan struktur data, arsitektur *software*, tampilan *interface* dan algoritma program. tujuannya untuk lebih memahami gambaran besar dari apa yang akan dikerjakan.

D . *Construction (Coding & Testing)*

Tahapan *construction* ini merupakan proses penerjemahan bentuk desain menjadi kode atau bentuk atau bahasa yang dapat dibaca oleh mesin. Setelah pengkodean selesai, dilakukan pengujian terhadap sistem dan juga kode yang sudah dibuat. tujuannya untuk menemukan kesalahan yang mungkin terjadi untuk nantinya diperbaiki.

E . *Deployment (Delivery , Support , Feedback)*

Tahapan *Deployment* merupakan tahapan implementasi perangkat lunak ke *customer*, melakukan *maintenance* (perawatan perangkat lunak) secara berkala, perbaikan *software*, evaluasi *software* dan pengembangan *software* berdasarkan umpan balik yang diberikan agar sistem dapat tetap berjalan dan berkembang sesuai dengan fungsinya.

# BAB II

**LANDASAN TEORI**

## Tinjauan Pustaka

* + 1. **Sistem Pendukung Keputusan Pemilihan Rumah Tinggal Di Perumahan Menggunakan *Metode Simple Additive Weighting (SAW)* (Studi Kasus : Kota Samarinda) oleh Tomy Reza Adianto, Zainal Arifin, dan Dyna Marisa Khairina, Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Mulawarman (2017).**

Penelitian pernah di lakukan oleh Tomy Reza Adianto, Zainal Arifin, dan Dyna Marisa Khairina, (2017). Penelitian ini berjudul “Sistem Pendukung Keputusan Pemilihan Rumah Tinggal Di Perumahan Menggunakan Metode *Simple Additive Weighting (Saw)* (Studi Kasus : Kota Samarinda)”. Perkembangan di lapangan menunjukkan bahwa konsumen dalam memilih perumahan ada empat aspek setidaknya yang dijadikan sebagai bahan pertimbangan yaitu: harga, luas tanah, waktu tempuh kepusat kota dan akses menuju perumahan. Hasil dari penelitian tersebut menghasilkan aplikasi berbasis *website* guna membantu konsumen memilih perumahan yang dinginkan dengan menggunakan metode SAW. Metode *Simple Additive Weighting (SAW)* merupakan salah satu metode untuk penyelesaian masalah *multi-attribute decision making*. Konsep dasar metode SAW adalah mencari penjumlahan terbobot dengan rating kinerja pada setiap alternatif pada semua atribut.

* + 1. **Sistem pendukung keputusan pemilihan perumahan Dengan metode *simple additive weighting (SAW)* oleh Hartika Manao, Berto Nadeak dan Taronisokhi Zebua, Teknik Informatika STMIK Budi Darma (2017).**

Penelitian tentang sistem pendukung keputusan pemilihan rumah berbasis web pernah dilakukan oleh Hartika Manao, Berto Nadeak dan Taronisokhi Zebua (2017), dengan penelitian yang berjudul “Sistem pendukung keputusan pemilihan perumahan Dengan metode *simple additive weighting (SAW)”.* Hasil penelitiannya adalah sistem pendukung keputusan tidak dimaksudkan untuk menggantikan peran pengambil keputusan, tapi untuk membantu dan mendukung pengambil keputusan. Salah satu metode yang sering digunakan dalam sistem pendukung keputusan adalah metode *simple additive weighting (SAW).* Metode *simple additive weighting* ini dapat menentukan nilai preferensi setiap alternatif, kemudian melakukan proses perangkinan yang dapa menyeleksi alternatif terbaik dari sejumlah alternatif yang ada. Kriteria – kriteria yang digunakan dalam penelitian ini adalah dengan parameter harga perumahan, jarak dengan pasar terdekat, kepadatan penduduk disekitar lokasi, jarak dari pusat kota, jarak dari sekolah atau universitas. Pemanfaatan sistem pendukung keputusan dalam pemilihan perumahan sangat optimal dalam menentukan pilihan yang tepat, sehingga para pembeli perumahan tidak mengalami kerugian baik dari sisi kriteria yang diinginkan maupun masalah lainnya ke depan. Pendukung keputusan yang dilakukan berdasarkan metode *simple addative wighting* cukup baik untuk menghasilkan nilai preferensi setiap alternatif yang akan dijadikan sebagai kandidat keputusan.

* + 1. **Sistem Pendukung Keputusan Pemilihan Perumahan Di Bandar Lampung**

**oleh Hilda Dwi Yunita & Fatimah Fahurian, Studi Sistem Informasi Fakultas Komputer Universitas Mitra Indonesia (2019).**

Penelitian tentang sistem pendukung keputusan pemilihan rumah juga pernah dilakukan Hilda Dwi Yunita & Fatimah Fahurian (2019), dengan penelitian yang berjudul “Sistem Pendukung Keputusan Pemilihan Perumahan Di Bandar Lampung”. Penelitian ini sistem pendukung keputusan perumahan adalah terapan. Sistem pendukung keputusan dapat membantu pengambil keputusan, yaitu pembeli dalam memilih perumahan yang sesuai dengan kriteria yang diinginkan dan sesuai dengan kebutuhan dan keuangan.Sistem Pendukung Keputusan Pemilihan Perumahan di Bandar Lampung, akan membuat masyarakat khususnya pembeli rumah tidak mengalami kesulitan untuk mencari perumahan yang sesuai dengan kriteria dan pembeli dapat memilih/membeli rumah sesuai dengan kebutuhan dan keuangan. Pendukung Keputusan Pemilihan Rumah Tinggal di Perumahan Menggunakan Metode Simple Additive Weighting (SAW) (Studi Kasus : Kota Samarinda). Penelitian ini menyimpulkan bahwa sistem pendukung keputusan dengan menggunakan metode SAW dapat membantu memberikan rekomendasi kepada calon pembeli rumah berdasarkan sistem sesuai dengan data yang dimasukkan oleh admin.

**Tabel 2.5** Tinjauan Studi

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Peneliti | Judul | Metode | Variabel | Hasil |
| 1 | Tomy Reza Adianto, Zainal Arifin, dan Dyna Marisa Khairina, pada tahun 2017 | Sistem Pendukung Keputusan Pemilihan Rumah Tinggal Di Perumahan Menggunakan Metode SAW (Studi Kasus : Kota Samarinda | Metode SAW | harga, luas tanah, waktu tempuh kepusat kota dan akses menuju perumahan. | Hasil dari penelitian tersebut menghasilkan aplikasi berbasis website guna membantu konsumen memilih perumahan yang dinginkan dengan menggunakan metode SAW. |
| 2 | Hartika Manao, Berto Nadeak dan Taronisokhi Zebua, pada tahun 2017 | Sistem pendukung keputusan pemilihan perumahan Dengan metode *simple additive weighting (SAW)* | Metode SAW | harga, jarak dengan pasar terdekat, kepadatan penduduk disekitar lokasi, jarak dari pusat kota, jarak dari sekolah | Pemanfaatan sistem pendukung keputusan dalam pemilihan perumahan sangat optimal dalam menentukan pilihan yang tepat, sehingga para pembeli perumahan tidak mengalami kerugian baik dari sisi kriteria yang diinginkan maupun masalah lainnya ke depan. |
| 3 | Hilda Dwi Yunita & Fatimah Fahurian  Pada tahun 2019 | Sistem Pendukung Keputusan Pemilihan Perumahan Di Bandar Lampung | Metode  SAW | model rumah, harga rumah, luas  tanah, luas bangunan, jarak ke lokasi,  lingkungan | SPK yang dapat memberikan rekomendasi kepada calon pembeli rumah berdasarkan sistem sesuai dengan data yang dimasukkan oleh admin. |

Perbedaan dengan penelitian yang sudah dilakukan oleh Tomy Reza Adianto, Zainal Arifin, dan Dyna Marisa Khairina pada tahun 2017 dengan judul “Sistem Pendukung Keputusan Pemilihan Rumah Tinggal Di Perumahan Menggunakan Metode SAW (Studi Kasus : Kota Samarinda” adalah pada penelitian tersebut belum adanya penjelasan lengkap tentang proses perancagan metode *simple additive weighting* dan juga perhitungan data menggunakan metode saw. Di penelitian yang penulis lakukan akan menjabarkan bagaimana proses perhitungan dengan menggunakan metode *simple additive weighting (SAW)* dengan data yang telah ditentukan sebelumnya.

Perbedaan dengan penelitian yang pernah dilakukan oleh Hartika Manao, Berto Nadeak dan Taronisokhi Zebua pada tahun 2017 dengan judul “Sistem pendukung keputusan pemilihan perumahan Dengan metode *simple additive weighting (SAW)”* adalah penelitian yang akan penulis lakukan dengan menggunakan kriteria yang lebih banyak yaitu menggunakan kriteria harga, jarak lokasi rumah ke kota, aksesibilitas ke perumahan, legalitas, fasilitas, dan desain rumah. Dan juga penelitian ini menggunakan perancangan UML atau *Unified Modeling Language* dan pengembangan perangkat lunak mengunakan metode *waterfall* yang dalam penelitian tersebut belum digunakan.

Perbedaan dengan penelitian yang pernah dilakukan oleh Hilda Dwi Yunita & Fatimah Fahurian Pada tahun 2019 dengan judul “Sistem Pendukung Keputusan Pemilihan Perumahan Di Bandar Lampung” adalah penelitian tersebut menggunakan program berbasis objek sedangkan yang akan penulis rancang dan bangun adalah sistem pendukung keputusan untuk rekomendasi pembelian rumah berbasis web, dengan kriteria tambahan dari penelitian tersebut, yang mana penelitian yang penulis akan lakukan adalah menggunkan kriteria yaitu harga, jarak lokasi rumah ke kota, aksesibilitas ke perumahan, legalitas, fasilitas, dan desain rumah.

Berdasarkan penelitian – penelitian yang telah dilakukan sebelumnya penulis berharap akan memberikan pembaruan dalam perancangan sistem dan pengembangan perangkat lunak dalam sistem pendukung keputusan untuk rekomendasi pembelian rumah menggunakan metode *simple additive weighting (SAW).*

## Tinjauan Studi

### *Website*

Aplikasi atau perangkat lunak *(software)* merupakan bagian yang tidak terpisahkan dari suatu sistem komputer, disamping keberadaan pengguna *(brainware),* perangkat keras *(hardware)* dan jaringan *(networking).* Jika dilihat dari lingkungan pengembangannya, aplikasi dapat dibagi menjadi aplikasi berbasis *desktop*, aplikasi berbasis web dan aplikasi berbasis *mobile*. Aplikasi berbasis *desktop* merupakan aplikasi yang memerlukan proses instalasi di setiap komputer yang akan menggunakannya. Contoh aplikasi berbasis *desktop* antara lain Microsoft Office, Mozilla Firetox, Adobe Photoshop dan Macromedia Dreamweaver,

Sementara itu, aplikasi berbasis web tidak memerlukan instalasi di setiap komputer karena apikasi berada di suatu *server*. untuk membuka aplikasi cukup menggunakan *browser* yang terhubung melalui jaringan ke *server*. Situs web merupakan salah satu contoh jenis aplikasi berbasis web. Jenis aplikasi yang ketiga yaitu aplikasi berbasis *mobile* merupakan aplikasi yang hanya dapat dijalankan pada perangkat bergerak atau mobile seperti *handpnone*, *smartphone* dan PDA. Contoh dari jenis aplikasi ini antara lain *browser* Opera Mini, Blackberry Messenger (BBM), WhatsApp Messenger dan Polaris Office. (Achmad Solichin, 2016:1).

Saat ini, perkembangan aplikasi berbasis web sangat pesat karena memang memiliki beberapa kelebihan dibanding aplikasi berbasis *desktop*. Berikut ini beberapa kelebihan yang dimiliki oleh jenis aplikasi berbasis web :

* 1. Pada sisi *client* (pengguna), tidak memerlukan proses instalasi. Jika terjadi perubahan aplikasi, *client* juga tidak perlu repot-repot melalukan proses *update* karena cukup dilakukan di sisi *server*.
  2. Dapat diakses darimana saja melalui jaringan. Jika *server* aplikasi berada di jaringan intranet LAN apikasi dapat di akses ke seluruh komputer dalam jaringan internet tersebut. Dan jika jaringan internet ( Memiliki IP Public atau diletakkan di web hosting), maka aplikasi dapat diakses dari internet.
  3. Data disimpan di sisi *server*, sehingga akses terhadap data dari sisi *client* (pengguna) dapat diatur sesuai kebutuhan.
  4. *Cross-platform*, artinya aplikasi dapat diakses melalui komputer dengan berbagai sistem operasi (Windows, Linux atau Mac) asalkan memiliki browser.
  5. Dari sisi *client*, tidak memerlukan spesifikasi komputer yang besar karena hampir seluruh proses aplikasi dilakukan di sisi *server*.
  6. *Client* (pengguna) lebih aman dari virus atau gangguan keamanan lainnya karena aplikasi berjalan di atas *browser*.

Disamping kelebihan di atas, aplikasi berbasis web juga memiliki kekurangan terutama dari sisi performa aplikasi yang sangat bergantung pada kondisi jaringan yang digunakan. Pada jaringan yang lambat, performa aplikasi baik dari sisi kecepatan akses maupun kecepatan proses di dalamnya juga akan lambat. Selain itu, karena sifatnya yang dapat diakses dari mana saja, maka aplikasi berbasis web lebih rentan dari serangan yang dilakukan oleh pihak-pihak yang tidak bertanggung jawab. Oleh karena itu, perlindungan dan pengamanan terhadap aplikasi mutlak dilakukan oleh pengembang dengan sebaik-baiknya. (Achmad Solichin, 2016:2).

Teknologi jaringan lahir dari kebutuhan akan jalur komunikasi yang efektif dan efisien untuk keperluan militer dan pertahanan pada tahun 1970-an. Selanjutnya berkembang untuk keperluan di bidang yang lebih luas, dan lahirlah konsep TCP/IP pada awal tahun 1974. TCP/IP merupakan sekumpulan aturan yang memungkinkan komputer satu dengan yang lainnya dapat saling "berbicara" dan mengirimkan informasi. Dengan kata lain, TCP/IP merupakan "bahasa" baku yang dimengerti oleh komputer dalam berkomunikasi. TCP/IP terus dikembangkan, mulai dari menghubungkan antar komputer, antar ruangan, antar gedung. antar kota hingga antar negara sehingga seluruh komputer di seluruh dunia akhirnya dapat terhubung. Jaringan yang menghubungkan komputer di seluruh dunia tersebut, itulah internet.

Sementara itu, web atau istilah lengkapnya *World Wide Web* (WWW) dikembangkan pertama kali oleh Tim Berners-Lee pada tahun 1989. Tim merupakan bagian dari CERN, sebuah laboratorium penelitian ilmiah di Genewa, Swiss. Pada awalnya, Tim mengusulkan WWW sebagai suatu cara berbagi dokumen diantara para peneliti. Dokumen *online* dapat diakses melalui alamat unik yang *disebut Universal Resources Locator* atau URL. Dokumen dapat saling berkaitan melalui suatu *link*. WWW menggunakan protokol HTTP *(Hypertext Transfer Protocol)* untuk berkomunikasi di antara komputer dalam satu jaringan. Selanjutnya WWW tidak hanya dikembangkan untuk keperluan para peneliti, namun juga dikembangkan untuk kalangan pendidikan, bisnis dan perorangan. (Achmad Solichin, 2016:3).

*Website* dapat di artikan sebagai kumpulan halaman yang berisi informasi data digital baik berupa *text*, gambar, animasi, suara dan video atau gabungan dari semuanya yang di sediakan melalui jalur koneksi internet sehingga dapat di akses dan dilihat oleh semua orang diseluruh dunia. Halaman *website* dibuat menggunakan bahasa standar yaitu HTML. Skrip HTML ini akan diterjemahkan oleh *web browser* sehingga dapat ditampilkan dalam bentuk informasi yang dapat di baca oleh semua orang (Rohi Abdulloh, 2018:1).

Secara umum, *website* dibagi menjadi 3 jenis, yaitu *website* statis, dinamis, dan interaktif.

1. *Website* Statis

Website statis yaitu jenis *website* yang isinya tidak diperbaharui secara berkala, sehingga isinya dari waktu ke waktu akan selalu tetap. *Website* jenis ini biasanya hanya digunakan untuk menampilkan profil dari pemilik *website* seperti profil perusahaan atau organisasi.

1. *Website* Dinamis

Website dinamis yaitu jenis website yang isinya terus diperbaharui secara berkala oleh pengelola web atau pemilik *website*. *Website* jenis ini banyak dimiliki oleh perusahaan atau perorangan yang aktifitas bisnisnya memang berkaitan dengan *internet*. Contoh paling mudah dari *website* jenis ini yaitu web blog dan *website* berita.

1. *Website* Interaktif

*Website* interaktif pada dasarnya termasuk dalam kategori *website* dinamis, dimana isi informasinya selalu diperbaharui dari waktu ke waktu. Hanya saja, isi informasi tidak hanya diubah oleh pengelola *website* tetapi lebih banyak dilakukan oleh pengguna *website* itu sendiri. Contoh *website* jenis ini yaitu website jejaring sosial seperti facebook dan twitter atau website *marketplace* seperti bukalapak, tokopedia, dan sebagainya.

### Sistem Pendukung Keputusan

Konsep sistem pendukung keputusan (SPK) pertama kali diungkapkan pada tahun 1971 oleh Michael S. Scott Morton dengan istilah *Management Decision System*. Kemudian sejumlah perusahaan, lembaga penelitian dan perguruan tinggi mulai melakukan penelitian dan membangun sistem pendukung keputusan, sehingga dari produksi yang dihasilkan dapat disimpulkan bahwa sistem ini merupakan suatu sistem berbasis komputer yang ditujukan untuk membantu pengambilan keputusan dalam memanfaatkan data dan model tertentu untuk memecahkan berbagai persoalan yang tidak terstruktur. Little mendefinisikan Sistem Pendukung Keputusan sebagai suatu suatu informasi berbasis komputer yang menghasilkan berbagai alternatif keputusan untuk membantu manajemen dalam menangani berbagai permasalahan yang terstruktur maupun tidak terstruktur dengan menggunakan data dan model. (Dicky Nofriansyah & Sarjon Defit, 2017:1).

Moore dan Chang berpendapat bahwa konsep struktur pada definisi awal Sistem Pendukung Keputusan (bahwa Sistem Pendukung Keputusan dapat menangani situasi semistruktur dan tidak terstruktur), sebuah masalah dapat dijelaskan sebagai masalah terstruktur dan tidak terstruktur hanya dengan memperhatikan sipengambil keputusan atau suatu spesifik. Jadi mereka mendefinisikan DSS sebagai sistem yang dapat diperluas untuk mampu mendukung analisis data ad hoc dan pemodelan keputusan, berorientasi terhadap perencanaan masa depan, dan digunakanpada interval yang tidak reguler dan tak terencana. (Dicky Nofriansyah & Sarjon Defit, 2017:2).

Bonczek, dkk mendefinisikan Sistem Pendukung Keputusan sebagaisistem berbasis komputer yang terdiri dari tiga komponen yang saling berinteraksi: sistem bahasa (mekanisme untuk memberikan komunikasi antar pengguna dan komponen Sistem Pendukung Keputusan yang lain), sistem pengetahuan (repositoripengetahuan domain masalah yang ada entah sebagai data atau sebagai prosedur) dan sistem pemrosesan masalah (hubungan antara komponen lainnya terdiri dari satu atau lebih kapabilitas manipulasi masalah umum yang diperlukan untuk pengambilan keputusan). Konsep – konsep yang diberikan oleh definisi tersebut sangat penting untuk memahami hubungan antara Sistem Pendukung Keputusan dan pengetahuan. Dari berbagai definisi diatas dapat disimpulkan bahwa Sistem Pendukung Keputusan adalah suatu sistem informasi spesifik yang ditujukan untuk membantu manajemen dalam mengambil keputusan yang berkaitan dengan persoalan yang bersifat semi terstruktur. Sistem ini memiliki fasilitas untuk menghasilkan berbagai alternatif yang secara interaktif digunakan oleh pemakai. (Dicky Nofriansyah & Sarjon Defit, 2017:2).

Komponen-komponen Sistem Pendukung Keputusan terdiri dari :

1. *Data Management*. Termasuk database, yang mengandung data yang relevan untuk berbagai situasi dan diatur oleh *software* yang disebut *Database Management Systems (DBMS).*
2. *Model Management*. Melibatkan model finansial, statistikal, *management science*, atau berbagai model kuantitatif lainnya, sehingga dapat memberikan ke sistem suatu kemampuan analitis, dan manajemen *software* yang diperlukan.
3. *Communication (dialog subsystem).* *User* dapat berkomunikasi dan memberikan perintah pada DSS melalui subsistem ini. Ini berarti menyediakan antarmuka.
4. *Knowledge Management*. Subsistem optional ini dapat mendukung subsistem lain atau bertindak sebagai komponen yang berdiri sendiri. (Dicky Nofriansyah & Sarjon Defit, 2017:3).

Menurut Simon ada tiga fase dalam proses Pengambilan Keputusan diantaranya sebagai berikut :

* 1. *Intellegence*

Tahap ini merupakan proses penelusuran dan pendeteksian dari ruang lingkup problematika secara proses pengenalan masalah. Data masukan diperoleh, diproses dan diuji dalam rangka mengindentifikasi masalah.

* 1. *Design*

Tahap ini merupakan proses menemukan, mengembangkan dan menganalisis alternatif tindakan yang bisa dilakukan. Tahap ini meliputi menguji kelayakan solusi.

* 1. *Choice*

Pada tahap ini dilakukan proses pemilihan diantara berbagai alternatif tindakan yang mungkin dijalankan. Hasil pemilihan tersebut kemudian diimplementasikan dalam proses pengambilan keputusan. (Dicky Nofriansyah & Sarjon Defit, 2017:4).

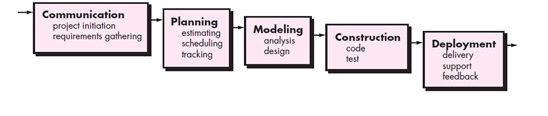
Sistem Pendukung Keputusan (SPK) biasanya dibangun untuk mendukung solusi atas suatu masalah atau suatu peluang. Aplikasi Sistem Pendukung Keputusan (SPK) digunakan dalam pengambilan keputusan. Aplikasi Sistem Pendukung Keputusan. Aplikasi Sistem Pendukung Keputusan (*SPK) menggunakan CBIS ( Computer base Information Systems)* yang fleksibel, interaktif, dan dapat diadaptasi yang dikembangkan untuk mendukung solusi atas masalah manajemen spesifik yang tidak terstruktur. Secara konsep ada 3(tiga) elemen yang terkait dengan Sistem Pendukung Keputusan, berikut ini adalah gambar dari setiap elemen yang terkait dalam sistem pendukung keputusan yaitu:

1. Masalah. Dalam sebuah sistem pendukung keputusan terdapat beberapa jenis masalah yaitu: masalah terstruktur, masalah semi terstruktur dan masalah tidak terstruktur. Mengikuti suatu pola atau model tingkah laku, baik yang terstruktur maupun tidak terstruktur.
2. Solusi. Dalam sebuah sistem pendukung terdapat beberapa jenis solusi pemecahaan masalah diataranya yaitu*: Multi Attribute Decision Making (MADM)* seperti: *Metode Simple Additive Weighting (SAW), Metode Weight Product (WP), Metode Analythical Hierarchy Process (AHP), Metode Topsis* dan lain-lain. Kemudian *Metode Multi Criteria Decision Making (MCDM)* seperti: *Metode Promethee, Metode Electre, Metode Oreste, Metode Entropi* dan Lain-lain. Selain terdapat juga *Metode Multi Factor Evaluation Process (MFEP),* *Metode Multi Attribute Utility Theory (MAUT)* serta *Metode FMADM (Fuzzy Multi Attribute Decision Making)* yang terdiri dari *F-AHP, F-SAW* dan lain-lain.
3. Hasil. Hasil atau keluaran dari sebuah sistem pendukung keputusan yaitu berupa sebuah keputusan yang dapat dijadikan sebagai tolak ukur sebuah kebijakan dari sebuah masalah yang diteliti atau dibahas. Keputusan merupakan kegiatan memilih suatu strategi atau tindakan dalam pemecahan masalah tersebut. Tindakan memilih strategi atau aksi yang diyakini manajer akan memberikan solusi. (Dicky Nofriansyah & Sarjon Defit, 2017:5).

### Rekayasa Perangkat Lunak

#### Metode *Waterfall*

*System Develoment Life Cycle (SDLC)* biasanya disebut juga dengan model *waterfall*. (Pressman: 2015). Model *waterfall* adalah model klasik yang bersifat sistematis , berurutan dalam membangun sebuah *software* . Nama model ini sebenarnya adalah "*Linear Sequential Model*" . Model ini termasuk ke dalam model generic pada rekayasa perangkat lunak dan pertama kali model *waterfall* ini diperkenalkan oleh Winston Royce pada tahun 1970 . Meskipun sering dianggap kuno , tetapi model waterfall ini merupakan model yang paling banyak dipakai dalam *Software Engineering (SE)* . Model *Waterfall* ini melakukan pendekatan secara sistematis dan berurutan . Model ini disebut dengan model *Waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan. (Pressman, 2015:42).



**Gambar 2. 1** Waterfall Pressman (Pressman:2015:42)

Pengembangannya metode *waterfall* memiliki beberapa tahapan yang berurut yaitu :

A . *Communication (Project Initiation & Requirements Gathering)*

Sebelum memulai pekerjaan yang bersifat teknis, sangat diperlukan adanya komunikasi dengan *customer* demi memahami dan mencapai tujuan yang ingin dicapai. Hasil dari komunikasi tersebut adalah inisialisasi proyek, seperti menganalisis permasalahan yang dihadapi dan mengumpulkan data-data yang diperlukan, serta membantu mendefinisikan fitur dan fungsi *software*. Pengumpulan data-data tambahan bisa juga diambil dari jurnal, artikel dan internet.

B . *Planning (Estimating , Schedulling , Tracking)*

Tahap selanjutnya adalah tahap perencanaan yang menjelaskan mengenai estimasi tugas - tugas teknis yang akan dilakukan, resiko - resiko yang dapat terjadi , sumber daya yang diperlukan dalam membuat sistem, produk kerja yang ingin dihasilkan, penjadwalan kerja yang akan dilaksananakan, dan *tracking* proses pengerjaan sistem .

C . *Modelling (Analysis & Design)*

Tahap ini adalah tahap perancangan dan pemodelan arsitektur sistem yang berfokus pada perancangan struktur data, arsitektur *software*, tampilan *interface* dan algoritma program. tujuannya untuk lebih memahami gambaran besar dari apa yang akan dikerjakan.

D . *Construction (Coding & Testing)*

Tahapan *construction* ini merupakan proses penerjemahan bentuk desain menjadi kode atau bentuk atau bahasa yang dapat dibaca oleh mesin. Setelah pengkodean selesai, dilakukan pengujian terhadap sistem dan juga kode yang sudah dibuat. tujuannya untuk menemukan kesalahan yang mungkin terjadi untuk nantinya diperbaiki.

E . *Deployment (Delivery , Support , Feedback)*

Tahapan *Deployment* merupakan tahapan implementasi perangkat lunak ke *customer*, melakukan *maintenance* (perawatan perangkat lunak) secara berkala, perbaikan *software*, evaluasi *software* dan pengembangan *software* berdasarkan umpan balik yang diberikan agar sistem dapat tetap berjalan dan berkembang sesuai dengan fungsinya.

Model *waterfall* ini terdapat beberapa sifat-sifat yang menojol dan cenderung menjadi permasalahan pada model *waterfall*.

* + 1. Ketika *problem* muncul, maka proses berhenti karena tidak dapat menuju ke tahapan selanjutnya. Apabila terdapat kemungkinan *problem* tersebut muncul akibat kesalahan dari tahapan sebelumnya, maka proses harus membenahi tahapan sebelumnya agar *problem* ini tidak muncul.
    2. Karena pendekatannya secara *sequential*, maka setiap tahap harus menunggu hasil dari tahap sebelumnya. Hal itu tentu membuang waktu yang cukup lama, artinya bagian lain tidak dapat mengerjakan hal lain selain hanya menunggu hasil dari tahap sebelumnya.

Metode pengembangan *waterfall* mempunyai keunggulan dalam membangun dan mengembangkan suatu sistem, antara lain:

1. Kualitas dari sistem yang dihasilkan akan baik. Ini dikarenakan oleh pelaksanaannya secara bertahap. Sehingga tidak terfokus pada tahapan tertentu.
2. Dokumen pengembangan sistem sangat terorganisir, karena setiap fase harus terselesaikan dengan lengkap sebelum melangkah ke fase berikutnya. Jadi setiap fase atau tahapan akan mempunyai dokumen tertentu.

Dalam proses membangun dan mengembangkan suatu sistem, metode *waterfall* mempunyai beberapa kelemahan, antara lain:

1. Diperlukan majemen yang baik, karena proses pengembangan tidak dapat dilakukan secara berulang sebelum terjadinya suatu produk.*.*
2. Kesalahan kecil akan menjadi masalah besar jika tidak diketahui sejak awal pengembangan.
3. Pelanggan sulit menyatakan kebutuhan secara eksplisit sehingga tidak dapat mengakomodasi ketidakpastian pada saat awal pengembangan.

Metode *waterfall* ini dianggap sebegai pendekatan yang lebih cocok digunakan untuk proyek pembuatan sistem baru dan juga pengembangan *software* dengan tingkat resiko yang kecil serta waktu pengembangan yang cukup lama. Tetapi salah satu kelemahan paling mendasar adalah menyamakan pengembangan *hardware* dan *software* dengan meniadakan perubahan saat pengembangan . Padahal , *error* diketahui saat *software* dijalankan dan perubahan - perubahan akan sering terjadi .

Di sisi lain. user tidak dapat mencoba sistem sebelum sistem benar-benar selesai. Selain itu, kinerja personil menjadi kurang optimal karena terdapat proses menunggu Suatu tahap selesai terlebih dahulu. Oleh karena itu seringkali diperlukan personil yang "*multi-skilled*” sehingga minimal dapat memnbantu pengerjaan untuk tahapan berikutnya. (Pressman, 2015:43).

#### *Unified Modeling Language* (UML)

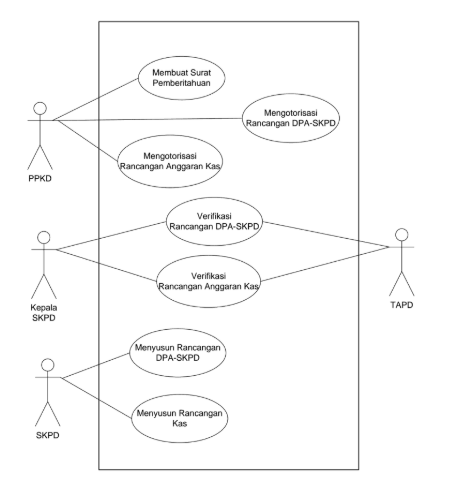
UML singkatan dari *Unified Modeling Language* yang berarti bahasa pemode lan standar. Chonoles mengatakan sebagai bahasa, berarti UML memiliki sintaks dan semantik. Ketika kita membuat model menggunakan konsep UML ada aturan -aturan yang harus diikuti. Bagaimana elemen pada model-model yang kita buat berhubungan satu dengan lainnya harus mengikuti standar yang ada. UML bukan hanya sekadar diagram, tetapi juga menceritakan konteksnya.( Muhamad Muslihudin & Oktafianto, 2016:58).

UML *(Unified Modeling Language)* merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram teks-teks pendukung. UML hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek. (Rosa A. S dan Shalahuddin. M, 2015:137). Terdapat empat macam *UML* yang umum digunakan pada saat pengembangan sebuah *software* atau *website* diagram diantara nya :

* + - 1. *Use Case Diagram*

*Use case diagram* adalah salah satu komponen dari *use case model* yang menggambarkan proses serta bagaimana sistem berinteraksi atau berhubungan dengan aktor.(Sri Mulyani, 2016:79). *Use-case diagram* memiliki unsur yang harus dipenuhi, yaitu :

1. *Use Cases*, yaitu sekumpulan fungsi yang terdapat dalam sistem dimana fungsi-fungsi tersebut dapat dilakukan oleh *actor* (*user*) untuk melakukan pekerjaannya dengan sistem yang ada.
2. *Actors*, yaitu segala sesuatu yang berinteraksi dengan sistem untuk bertukar informasi, baik *user* maupun sistem dari luar.
3. *Relationships*, yaitu garis yang menghubungkan antara *actors* dengan *use cases* yang dapat menggambarkan hubungan antara *actors* dengan *use cases* itu sendiri.



**Gambar 2. 2** Use-Case Diagram Sri Mulyani, (2016:109)

**Tabel 2. 1** Notasi Use-Case Diagram

|  |  |  |
| --- | --- | --- |
| **Nama Komponen** | **Keterangan** | **Simbol** |
| ***Use Case*** | Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama *use case*. |  |
| ***Actor*** | Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama *actor*. |  |
| ***Association*** | Komunikasi antar aktor dan *use case* yang berpartisipasi pada use case atau *use case* memiliki interaksi dengan *actor*. |  |

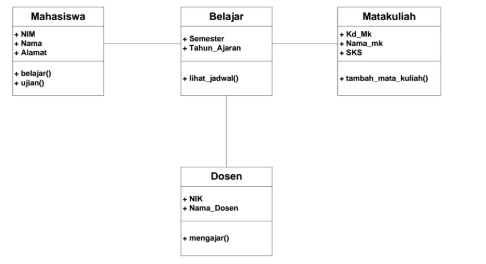
Sumber : Rosa A . S dan Shalahuddin M (2015: 156)

2. *Class Diagram*

*Class diagram* adalah diagram yang digunakan untuk merepresentasikan kelas, komponen-komponen kelas dan hubungan anatara masing-masing kelas. Selain itu class diagram mendeskripsikan jenis-jenis objek dalam sistem dan berbagai macam hubungan statis yang terdapat diantara mereka. *Class diagram* juga menunjukan *property* dan operasi sebuah kelas serta batasan-batasan yang terdapat dalam menggunakan istilah fitur sebagai istilah umum yang meliputi *property* dan operasi sebuah kelas. (Sri Mulyani, 2016:274).

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem”. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

1. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas.
2. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.



**Gambar 2. 3** Class Diagram Sri Mulyani (2016:248)

**Tabel 2. 2** Notasi class diagram

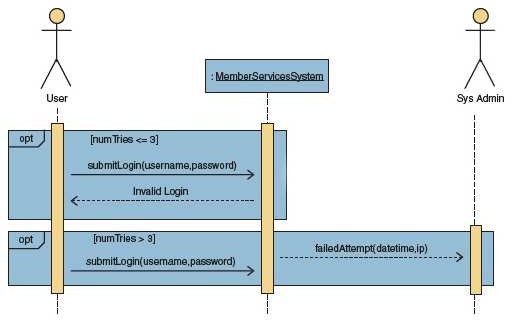
|  |  |
| --- | --- |
| **Symbol** | **Penjelasan** |
| |  | | --- | | Nama\_kelas | | + attribut | | +operasi() | | *Class*  Deskripsi dari objek yang terbagi atas 3 bagian, yaitu nama *class* pada bagian atas, atribut pada bagian tengah dan operasi pada bagian bawah. |
|  | *Aggregation*  Relasi antarkelas dengan makna semua- bagian ( whole-part). |
|  | *Association*  Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan *multiplicity*. |
|  | *Generalization*  Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus). |
|  | *Dependency*  Relasi antarkelas dengan makna kebergantungan antarkelas. |

Sumber : Rosa A . S dan Shalahuddin M (2015: 146)

* + - 1. *Sequence Diagram*

Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dengan massage yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah use case beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada use case.

Banyaknya diagram sekuen yang harus digambar adalah minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup dalam diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak. (Rosa dan M. Shalahudin, 2015:165).



**Gambar 2. 4** Sequence Diagram Rosa dan M. Shalahudin, (2015:165).

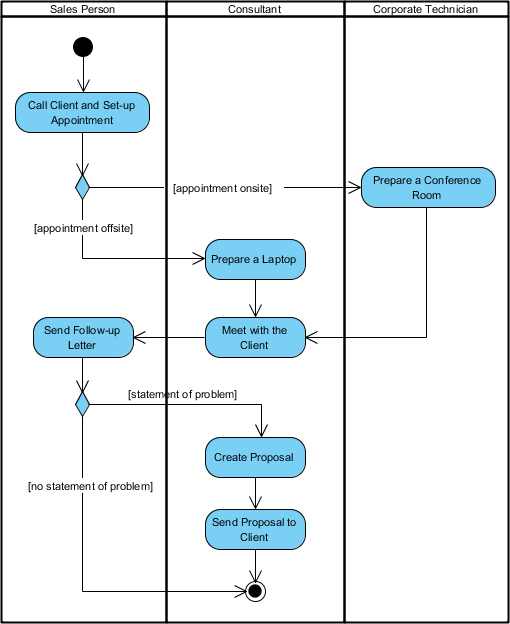
**Tabel 2. 3** Notasi sequence diagram

|  |  |  |
| --- | --- | --- |
| **Simbol** | **Nama** | **Keterangan** |
|  | *Actor* | Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama actor. |
|  | *Object Lifeline* | Objek entity, antarmuka yang saling berinteraksi. |
|  | *Message* | Spesifikasi dari komunikasi antar objek yang memuat informasi- informasi tentang aktifitas yang terjadi. |
|  | *Message (return)* | Spesifikasi dari komunikasi antar objek yang memuat informasi- informasi tentang aktifitas yang terjadi |
|  | *Activation* | Menyatakan objek dalam keadaan aktif dan berinteraksi. |
|  | Pesan tipe create  <<create>> | Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat. |

Sumber : Rosa A . S dan Shalahuddin M (2015: 165)

* + - 1. *Activity Diagram*

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Rosa dan M. Shalahudin (2015:161).



**Gambar 2. 5** Activity Diagram Rosa dan M. Shalahudin (2015:161).

**Tabel 2. 4** Notasi diagram activity

|  |  |  |
| --- | --- | --- |
| **Komponen** | **Simbol** | **Penjelasan** |
| *Initial node* |  | Diagram aktivitas memiliki sebuah status awal. |
| *Actions* |  | Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja. |
| *Flow* |  | Menunjukkan perkembangan tindakan. |
| *Decision* |  | Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu |
| *Fork* |  | Menunjukkan tindakan dilakukan secara bersamaan |
| *Join* |  | Menandakan akhir dan penggabungan pross yang berlangsung bersamaan. |
| *Activity Final* |  | Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir. |

Sumber : Rosa A . S dan Shalahuddin M (2015:162)

*Activity diagram*, sesuai dengan namanya diagram ini menggambarkan tentang aktifitas yang terjadi pada sistem. Dari pertama sampai akhir, diagram ini menunjukkan langkah – langkah dalam proses kerja sistem yang kita buat. Sebagai contoh, langkah – langkah memasak air. Tetapi kita akan menjelaskannya dengan bentuk grafik. Struktur diagram ini juga mirip dengan *flowchart*. Fungsi *Activity Diagram* Menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses Memperlihatkan urutan aktifitas proses pada sistem *Activity diagram* dibuat berdasarkan sebuah atau beberapa *use case* pada *use case diagram*.

#### *Simple Additive Weighting (SAW)*

Metode *Simple Additive Weighting (SAW)* sering juga dikenal istilah metode penjumlahan terbobot. Konsep dasar metode SAW adalah mencari penjumlahan terbobot dari rating kinerja pada setiap alternatif pada semua atribut (Fishburn. 1967) (MacCrimmon. 1968). Metode SAW membutuhkan proses normalisasi matriks keputusan (X ke suatu skala yang dapat diperbandingkan dengan semua rating alternatif yang ada. Metode ini merupakan metode yang paling terkenal dan paling banyak digunakan dalam menghadapi situasi *Multiple Attribute Decision Making (MADM)*. MADM itu sendiri merupakan suatu metode yang digunakan untuk mencari alternatif optimal dari sejumlah alternatif dengan kriteria tertentu. Metode SAW ini mengharuskan pembuat keputusan. (Febrina Sari, 2018:85).

Metode *Simple Additive Weighting (SAW)* dapat di artikan sebagai metode pembobotan sederhana atau penjumlahan terbobot pada penyelesaian masalah dalam sebuah sistem pendukung keputusan. Konsep metode ini adalah dengan mencari rating kinerja (skala prioritas) pada setiap alternatif disemua atribut (Dikcy Nofriansyah & Sarjon Defit, 2017:33).

Normalisasi adalah proses dua langkah yang meletakan data dalam bentuk tabulasi dengan menghilangkan kelompok berulang lalu menghilangkan data yang terduplikasi dari tabel relasional, teori normalisasi didasarkan pada bentuk normal. Sebuah tabel relasional dikatakan berada pada bentuk normal tertentu jika tabel memenuhi himpunan batasan tertentu (Febrina Sari, 2018:86). Pada waktu menormalisasi basis data, ada empat tujuan yang harus dicapai, yaitu :

1. Mengatur data dalam kelompok-kelompok sehingga masing-masing kelompok hanya menangani bagian kecil sistem.
2. Meminimalkan jumlah data yang berulang dalam basis data.
3. Membuat basis data yang datanya diakses dan dimanipulasi secara cepat dan efisien tanpa melupakan integrasi data.
4. Mengatur data sedemikian rupa sehingga ketika memodifikasi data anda hanya mengubah pada suatu tempat.

Berikut adalah aturan – aturan normalisasi :

1. Hilangkan kelompok berulang, buat tabel terpisah untuk setiap himpunan atribut yang berhubungan dan tentukan kunci utama pada masing - masing tabel.
2. Hilangkan data berulang, jika sebuah atribut hanya tergantung pada sebagian kunci utama gabungan, pindahkan atribut ke tabel lain.
3. Hilangkan kolom yang tidak tergantung pada kunci, jika atribut tidak tergantung pada kunci, pindahkan atribut ke tabel lain.
4. Pisahkan relasi majemuk, tidak ada tabel yang bisa mengandung dua atau lebih relasi majemuk yang tidak berhubngan langsung.
5. Pisahkan relasi majemuk yang berhubungan secara semantik, ada batasan pada informasi yang memperbolehkan pemisahan relasi many to many yang berhubungan secara logis.
6. Bentuk normal optimal, sebuah model hanya dibatasi oleh fakta sederhana.
7. Bentuk normal domain-key, sebuah model harus terbatas dari semua anomaly.

Metode SAW (*Simple Additive Weighting*) Merupakan suatu metode yang digunakan untuk mencari alternatif optimal dari sejumlah alternatif dengan kriteria tertentu. Inti dari SAW adalah menentukan nilai bobot untuk setiap atribut, kemudian dilanjutkan dengan proses perangkingan yang menyeleksi alternatif yang sudah diberikan. Pada dasarnya, ada 3 pendekatan untuk mencari nilai bobot atribut, yaitu pendekatan subyektif, pendekatan objektif dan pendekatan integrasi antara subyektif & objektif. Masing- masing pendekatan memiliki kelebihan dan kelemahan. Pada pendekatan subyektif, nilai bobot ditentukan berdasarkan subyektifitas dari pada pengambil keputusan, sehingga beberapa faktor dalam proses perangkingan alternatif bisa ditentukan secara bebas. Sedangkan pada pendekatan obyektif, nilai bobot dihitung secara matematis sehingga mengabaikan subyektifitas dari pengambil keputusan (Febrina Sari, 2018:87).

Metode SAW membutuhkan proses normalisasi matriks keputusan (X) ke suatu skala yang dapat diperbandingkan dengan semua rating alternatif yang ada. Metode ini merupakan metode yang paling terkenal dan paling banyak digunakan dalam menghadapi situasi *Multiple Attribute Decision Making* (MADM). MADM itu sendiri merupakan suatu metode yang digunakan untuk mencari alternatif optimal dari sejumlah alternatif dengan kriteria tertentu. Metode SAW ini mengharuskan pembuat keputusan menentukan bobot bagi setiap atribut. Skor total untuk alternatif diperoleh dengan menjumlahkan seluruh hasil perkalian antara rating (yang dapat dibandingkan lintas atribut) dan bobot tiap atribut. Rating tiap atribut haruslah bebas dimensi dalam arti telah melewati proses normalisasi matriks sebelumnya. Metode *Simple Additive Weighting (SAW),* adalah salah satu metode untuk menghadapi situasi *Multi Attribute Decision Making* (MADM) yang di gunakan untuk pengambilan keputusan*. Metode Simple Additive Weighting* disarankan untuk menyelesaikan masalah penyeleksian dalam sistem pengambilan keputusan multi proses.

Metode *Simple Additive Weighting* merupakan metode yang banyak digunakan dalam pengambilan keputusan yang memiliki banyak atribut. Metode *Simple Additive Weighting* membutuhkan proses normalisasi matriks keputusan (x) ke suatu skala yang dapat diperbandingkan dengan semua rating alternatif yang ada. Formula untuk melakukan normalisasi tersebut adalah sebagai berikut:

Keterangan:

Max = Nilai terbesar dari setiap kriteria i.

Min = Nilai terkecil dari setiap kriteria i.

= Nilai atribut yang dimiliki dari setiap kriteria.

Benefit = Jika nilai terbesar adalah yang terbaik.

Cost = Jika nilai terkecil adalah yang terbaik.

Dimana rij adalah rating kinerja ternormalisasi dari alternatif Ai pada atribut Cij i=1,2,…,n. Nilai preferensi untuk setiap alternatif (Vi) di berikan sebagai:

Keterangan:

Vi = Rangking untuk setiap alternatif.

Wj = Nilai bobot rangking (dari setiap kriteria).

rij = Nilai rating kinerja ternormalisasi.

Nilai Vi yang lebih besar mengidentifikasikan bahwa alternatif Ai lebih terpilih.

Nilai Vi yang lebih besar mengindikasikan bahwa alternatif (Ai) lebih terpilih. Berikut ini adalah tahapan-tahapan menggunakan metode SAW:

* 1. Menentukan kriteria-kriteria yang akan dijadikan acuan dalam pengambilan keputusan, yaitu Ci.
  2. Menormalisasi setiap nilai alternatif pada setiap atribut dengan cara menghitung rating kinerja.
  3. Membuat matriks keputusan berdasarkan kriteria (C1), kemudian melakukan normalisasi matriks berdasarkan persamaan yang disesuaikan dengan jenis atribut sehingga diperoleh matriks ternormalisasi R.
  4. Hasil akhir diperoleh dari proses perankingan yaitu penjumlahan dari perkalian matriks ternormalisasi R dengan bobot vektor sehingga diperoleh nilai terbesar yang dipilih sebagai alternatif terbaik (A1) sebagai solusi.

Kelebihan dari model Sinmple Additive Weighting SAW) dibandingkan dengan model pengambilan keputusan yang lain terletak pada kemampuannya untuk melakukan penilaian secara lebih tepat karena didasarkan pada nilai kriteria dan bobot preferensi yang sudah ditentukan, selain itu SAW juga dapat menyeleksi alternatif terbaik dari sejumlah alternatif yang ada karena adanya proses perangkingan setelah menentukan nilai bobot untuk setiap atribut.

Sedangkan kekurangan dari metode SAW adalah perhitungan dilakukan dengan menggunakan bilangan crisp dan adanya perbedaan perhitungan normalisasi. Sedangkan kekurangan dari metode SAW adalah perhitungan dilakukan dengan menggunakan bilangan crips dan adanya perbedaan perhitungan normalisasi (Febrina Sari, 2018:89).

### *Database*

Basis Data terdiri atas 2 kata, yaitu Basis dan Data. Basis kurang lebih dapat diartikan sebagai markas atau gudang, tempat bersarang atau berkumpul, sedangkan data adalah referensi fakta dunia nyata yang mewakili suatu objek seperti manusia, barang, hewan, peristiwa, konsep, keadaan dan sebagainnya, yang diwujudkan dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya. Basis data dapat di definisikan sebagai himpunan kelompok data yang saling berhubungan yang diorganisasikan sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah (Fathansyah, 2018:2).

Basis Data dan lemari arsip sesungguhnya memiliki prinsip kerja dan tujuan yang sama. Prinsip utamanya adalah pengaturan data/arsip. Dan tujuanutamanya adalah kemudahan dan kecepatan dalam pengambilan kembali data/arsip. Perbedaannya hanya terletak pada media penyimpanan yang digunakan. Jika lemari arsip menggunakan lemari dari besi atau kayu sebagai media penyimpanan, maka basis data menggunakan media penyimpanan elektronis seperti cakram magnetis (magnetic disk atau disingkat sebagai disk saja). Hal ini merupakan konsekuensi yang logis, karena lemari arsip langsung dikelola oleh manusia, sementara basis data dikelola melalui perantaraan mesin pintar elektronis (yang kita kenal sebagai komputer). Perbedaan media ini yang selanjutnya melahirkan perbedaan-perbedaan lain yang menyangkut jumlah dan jenis metode yang dapat digunakan dalam upaya penyimpanan. (Fathansyah, 2018:3).

Operasi-operasi dasar yang dapat kita lakukan berkenaan dengan basis data dapat meliputi:

1. Pembuatan basis data baru *(create database*), yang identik dengan pembuatan lemari arsip yang baru.
2. Penghapusan basis data (*drop database*), yang identik dengan perusakan lemari arsip (sekaligus beserta isinya, jika ada).
3. Pembuatan tabel baru ke suatu basis data (*create table*), yang identik dengan penambahan map arsip baru ke sebuah lemari arsip yang telah ada.
4. Penghapusan tabel dari suatu basis data (*drop table*), yang identik dengan perusakan map arsip lama yang ada di sebuah lemari arsip.
5. Penambahan/pengisian data baru ke sebuah tabel di sebuah basis data (*insert*). yang identik dengan penambahan lembaran arsip ke sebuah map arsip.
6. Pengambilan data dari sebuah tabel (*query*), yang identik dengan pencarian lembaran arsip dari sebuah map arsip
7. Pengubahan data dari sebuah tabel (*update*), yang identik dengan perbaikan isi lembaran arsip yang ada di sebuah map arsip.
8. Penghapusan data dari sebuah tabel (*delete*), yang identik dengan penghapusan sebuah lembaran arsip yang ada di sebuah map arsip. (Fathansyah, 2018:5).

Basis data sendiri dapat didefinisasikan dalam sejumlah sudut pandang seperti:

1. Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redundasasi) yang tidak perlu, untuk memenuhi berbagai kebutuhan.
3. Kumpulan *file*/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronis.

Dengan DBMS pengguna dapat melakukan:

1. Membuat *database* baru dan menentukan struktur data dengan menggunakan bahasa khusus data.
2. Melakukan query terhadap data dan memodifikasi data dengan menggunakan

bahasa yang tepat yang biasa disebut *query language* atau *data-manipulation language.*

1. Mendukung penggunaan penyimpanan yang sangat besar (*terabytes* dan lebih

dari itu).

1. Melakukan *recovery* terhadap *database* bila terjadi kegagalan, berbagai macam kerusakan, atau penggunaan yang salah.
2. Mengontrol akses data dari banyak pengguna sekaligus, tanpa terjadinya interaksi antar pengguna (*isolation*) dan tanpa tindakan pada data yang akan dilakukan secara parsial tetapi tidak sepenuhnya (*atomicy*).

Menurut Uus Rusmawan (2019:40), secara umum komponen basis data terdiri dari hal – hal berikut :

1. *Hardware*, sebagai pendukung operasi pengolahan data seperti CPU, memori, disk, terminal, dan sebagainya.
2. *Software* sistem operasi, (windows, linux, unix, dan sistem operasi lainnya).
3. *Software* pengelola basis data (DBMS) seperti MS-Access, SQL, Oracle, mysql, DB2, informix, sybase, postgree, IBM, AS400, dan sejenisnya.
4. *Software* program aplikasi misalnya visual basic, Delphi, php, dan lain – lain.
5. Basis data (semua data yang diperlukan dan dikelola oleh sistem basis data).
6. Pemakai atau pengguna basis data *(user).*

Menurut Uus Rusmawan (2019:41), basis data penting dalam pengolahan data dalam menggunakan komputer karena beberapa alasan, yaitu:

1. Sebagai komponen utama atau penting dalam sistem informasi, karena merupakan dasar dalam meyediakan informasi.
2. Menentukan kualitas informasi yang cepat, akurat, relevan sehingga informasi yang disajikan tidak basi. Informasi dapat dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya mendapatkannya.

*Database* atau basis data adalah sekumpulan infomasi yang disimpan dalam komputer secara sistematik sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi (Rohi Abdulloh, 2018:103). Dalam pembuatan aplikasi perlu memperlihatkan rancangan database agar aplikasi yang dibuat dapat berjalan sesuai dengan konsep yang di rencanakan. Perancangan *database* yang baik, hendaknya mengikuti aturan seperti berikut :

1. Tidak boleh ada dua data atau lebih yang sama atau *redundancy* data.
2. Setiap table harus memiliki kolom yang unik yang disebut dengan *primary key*, dimana kolom tersebut tidak dapat berisi data yang sama.
3. Besar dan ukuran *table* dibuat seminimal mungkin dengan pengaturan tipe daya yang tepat.
4. Pembuatan table dan kolom pada *table* hendaknya disesuaikan dengan kebutuhan aplikasi.

### Alat Pengembangan Perangkat Lunak

#### HTML

HTML merupakan singkatan dari *Hypertxt Markup Language* yaitu bahasa standar web yang dikelola penggunaanya oleh W3C *(World Wide Web Consotium)* berupa tag-tag yang menyusun setiap elemen dari website. HTML berperan sebaga penyusun struktur halaman website yang menempatkan setiap elemen website sesuai layout yang diinginkan. HTML biasanya disimpan dalam sebuah file berekstensi html. Untuk mengetikkan skrip HTML dapat menggunakan *text editor* seperti Notepad sebagai bentuk paling sederhana atau text editor khusus yang dapat mengenali setiap unsur skrip HTML dan menampilkanya dengan warna yang berbeda sehingga mudah dibaca, seperti Notepad++, Sublime Text, dan masih banyak lagi aplikasi lain yang sejenis.(Rohi Abdulloh, 2018:7).

HTML terdiri atas unsur-unsur yang membentuk struktur skrip HTML, yaitu tag, atribut, dan elemen. (Rohi Abdulloh, 2018:8).

* 1. Tag

Tag adalah simbol khusus (*markup*) berupa dua karakter "<" dan ">" yang mengapit suatu teks sebagai nama tag. Contohnya tag <body> adalah tag dengan nama body. Sebagian besar tag memiliki tag pembuka dan tag penutup yang membatasi pengaruh dari tag tersebut. Namun, terdapat sebagian kecil tag yang tidak memiliki tag penutup. Penutup tag ditambah simbol "" setelah tanda "", misalnya </body> yang merupakan penutup tag <body>. Berikut adalah karakteristik dari tag pada HTML:

1. Tag boleh ditulis dengan huruf besar (kapital), kecil atau campuran, karena sifatnya yang *incase sensitive*. Misalnya: <BODY>, <body> atau <Body> semuanya memiliki hasil yang sama. Namun, untuk standarnya lebih baik menggunakan huruf kecil semua.
2. Tag bisa berisi tag lain. Tag yang dibuka dahulu ditutup terakhir, sedangkan tag yang dibuka terakhir ditutup pertama. Namun, kesalahan dalam struktur ini tidak mengakibatkan error.
   1. Atribut

Atribut yaitu *property* yang mengatur bagaimana elemen dari suatu tag akan ditampilkan. Atribut ditulis di dalam simbol tag setelah nama tag dengan dipisahkan oleh spasi. Atribut ada yang memiliki nilai dan ada yang tidak memiliki nilai. Nilai suatu atribut ditulis di dalam tanda petik ganda ("."), dipisahkan dengan simbol sama dengan (=) dari nama atribut. Misalnya: <p align="center">. Berikut adalah karakteristik dari atribut pada HTML :

1. Atribut hanya ditulis pada tag pembuka.
2. Setiap tag memiliki atribut yang berbeda-beda yang dapat diterapkan pada tag tersebut.
3. Beberapa atribut ada yang dapat diterapkan ke semua tag atau disebut *global* *attribute*, misalnya: *class, id, style,* dan sebagainya.
4. Sebuah tag dapat ditulis dengan satu atribut, banyak atribut, atau tanpa atribut sama sekali.
5. Jika tag ditulis dengan banyak atribut, masing-masing atribut dipisahkan oleh spasi.
   1. Element

Element adalah bagian dari skrip HTML yang terdiri dari tag pembuka, isi element dan tag penutup. Jika sebuah element ditampilkan pada *browser*, maka hanya isi *element* yang tampil di *browser*.

HTML disimpan dengan ekstensi .html. jika di dalamnya tidak mengandung skrip PHP. Jika di dalamnya mengandung skrip PHP, maka disimpan dengan ekstensi ".php. Dalam penulisan HTML ada beberapa tag yang wajib dituliskan dengan struktur yang sudah ditentukan

<IDOCTYPE HTML>, adalah tag awal dari setiap dokumen HTML yang berfungsi untuk menginformasikan pada browser bahwa dokumen yang sedang dibuka adalah dokumen HTML. Tag ini perlu perlu dicantumkan diawal dokumen. <html> … </html>, Tag ini perlu adalah tag yang menunjukkan pembuka dan penutup dokumen HTML

#### PHP

*Hypertext Preprocessor* atau lebih akrab dengan sapaan PHP merupakan bahasa pemrograman script *server-side* yang didesain untuk pengembangan web. PHP disebut bahasa pemrograman *server-side* karena diproses pada komputer *server*. Hal ini berbeda dengan bahasa pemrograman *client-side* seperti Javascript yang diproses di dalam w*eb browser.* PHP dapat digunakan secara gratis dan bersifat *Open Source*. PHP dirilis dalam lisensi *PHP License*, sedikit berbeda dengan lisensi *GNU General Public License* yang biasa digunakan untuk proyek *Open Source*. (Miftahul Jannah, Sarwandi, dan Cyber Creative, : 2019:1).

Pada awalnya, PHP merupakan kependekan dari *Personal Home Page*. PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama *Form Interpreted (FI)* yang wujudnya berupa sekumpulan skrip dan digunakan untuk mengolah data formulir dari web. Selanjutnya, Rasmus merilis *Source Code* tersebut untuk umum dan menamakannya PHP/FL. Dengan perilisan *Source Code* ini menjadi *Open Source*, banyak pemrograman yang tertarik untuk ikut dalam pengembangan PHP. Pada November 1997, dirilis PHP/FI 2.0. Saat itu, interpreter PHP sudah diimplementasikan dalam program C. Dalam rilis ini disertakan juga modul-modul ekstensi yang bisa meningkatkan kemampuan PHP/FI secara signifikan. Tahun 1997, sebuah perusahaan bernama Zend menulis ulang interpreter PHP menjadi lebih bersih, lebih baik, dan lebih cepat. Kemudian pada bulan Juni 1998, perusahaan tersebut merilis interpreter baru untuk PHP dan meresmikannya sebagai PHP 3.0 dan singkatan PHP pun diubah menjadi Hypertext Preprocessing Pertengahan tahun 1999, Zend merilis interpreter PHP baru dan rilis tersebut dikenal dengan sebutan PHP 4.0. Pada tahun 2004, Zend kembali merilis PHP dengan versi terbaru, yaitu PHP 5.0. Versi ini dimasukkan model pemrograman berorientasi objek ke dalamnya, sehingga mampu menjawab perkembangan bahasa pemrograman ke arah paradigma berorientasi objek. Versi terbaru dan stabil dari bahasa pemrograman PHP saat ini adalah versi 7 yang dirilis pada tanggal 17 Februari 2017 kemarin. (Miftahul Jannah, Sarwandi, dan Cyber Creative, : 2019:3).

Konsep kerja PHP diawali dengan satu permintaan suatu halaman web oleh *browser*. Berdasarkan URL (*Uniform Resource Locator*) atau dikenal dengan alamat Internet, *browser* mendapat alamat dari *webserver*, mengidentifikasikan alamat yang dikehendaki, dan menyampaikan segala informasi yang dibutuhkan oleh *Web Server*. Selanjutnya *Web Server* akan mengirimkan isinya ke mesin php dan mesin inilah yang memproses dan memberikan hasilnya (berupa kode html ke *webserver*, selanjutnya *webserver* menyampaikan ke *client*. ( Rintho Rante Rerung, 2018:158).

PHP merupakan kependekan dari PHP *Hypertext Preprocessor* yaitu bahasa pemrograman web yang dapat disisipkan dalam skrip HTML dan bekerja di sisi *server*. Tujuan dari bahasa ini adalah membantu para pengembangan web untuk membuat web dinamis dengan cepat. (Rohi Abdulloh, 2018:127). Agar dapat menjalankan PHP harus menyediakan perangkat lunak berikut:

1. *Web Server (Apache, IIS, Personal Web Server/PWS)*
2. *PHP Server*
3. *Database Server (MySQL, Interbase, MS SQlL dll)*

Aplikasi di atas tidak perlu diinstal satu per satu, karena biasanya telah tersedia paket aplikasi yang telah menyediakan ketiganya dalam satu paket aplikasi seperti Xampp atau Appserv. Jika menggunakan Xampp, maka skrip PHP ditulis di dalam folder htdocs. Namun, jika menggunakan Appserv, maka skrip PHP di tulis di dalam folder www. Skrip PHP dituliskan di antara tanda <Pphp dan ? yang memisahkan skrip PHP dengan skrip lainnya. Satu file PHP dapat berisi full skrip PHP atau dapat disisipkan diantara skrip lain seperti HTML, CSS maupun JavaScript Namun, file yang berisi skrip PHP wajib disimpan dengan ekstensiphp dan disimpan di server (folder htdocs atau www). Jika disimpan dengan ekstensi HTML atau di simpan di sembarang tempat maka skrip PHP tidak diproses sebagaimana mestinya. Fungsi PHP dalam pemrograman web diantaranya menghasilkan halaman dinamis pada website, membuat, membuka, menulis, membaca, menghapus dan menutup, file pada server, memproses data yang dikirim dari form, mengirim dan mengakses cookie, modifikasi data pada database, mengontrol akses user, dan mengenkripsi data. (Rohi Abdulloh, 2018:128)

#### *Bootstrap*

*Bootstrap* adalah paket aplikasi siap pakai untuk membuat front end sebuah *website* bisa dikatakan *bootstrap* adalah *template* desain web dengan fitur plus. Bootstrap diciptakan untuk mempermudah proses desain web bagi berbagai tingkat pengguna level pemula hingga yang sudah berpengalaman. Cukup bermodalkan pengetahuan dasar mengenai *HTML* dan *CSS* anda sudah bisa menggunakan *bootstrap*.

Paket *bootstrap* berisi sekumpulan *file* CSS, *font* dan *javascript* yang siap diintegrasikan ke sebuah dokumen *HTML* menggunakan kaidah-kaidah tertentu dokumen *HTML* yang dihasilkan pun secara dinamis akan tampil dalam *layout* yang disesuaikan dengan ukuran layar piranti pengunjung daya tarik *bootstrap* terletak pada kemudahan dan kepraktisan penggunaannya tata *layout* dan pewarnaan juga bersih, *simple*, indah dan berkesan *modern*. Selain itu mengingat akses piranti mobil semakin meningkat tajam maka pendekatan desain web dan *mobile* *friendly* yang ditawarkanbushra pun juga menjadi solusi dan murah bagi pengembang web cukup dengan *bootstrap* website anda bisa diakses secara nyaman via ponsel tanpa menggunakan aplikasi khusus untuk peranti *mobile.*

*Bootstrap* merupakan proyek *open source* dan tentunya pemakaiannya bersifat gratis. Saat ini, *bootstrap* menjadi aplikasi yang paling populer di situs layanan hosting *software open source* GitHub.com. Jutaan *Website* telah menggunakan *bootstrap* sebagai *framework* desainnya. (Zaenal A. Rozi dan Smit Dev, 2015:1).

*Bootstrap* adalah *framework* ataupun *tools* untuk membuat aplikasi web ataupun *website* yang bersifat *responsive* secara cepat, mudah dan gratis. Kata ‘*responsive*’ disini berarti bahwa tampilan web (lebar dan susunan isinya) dapat berubah secara otomatis sesuai dengan lebar layar yang menampilkannya. *Bootstrap* terdiri dari CSS dan HTML untuk menghasilkan *Grid, Layout,* Tifografi, Tabel, *Form,* Navigasi, dan lain-lain. Di dalam *Bootstrap* juga sudah terdapat *Javascript (jQuery Plugins)* untuk menghasilkan komponen UI yang cantik seperti Transitions, Modal, DropDown, ScrollSpy, ToolTip, PopOver, Tab, Alert, Button, Carousel, dan lain-lain. Dengan *Bootstrap*, kita dapat membuat *responsive website* dengan cepat dan mudah dan dapat berjalan sempurna pada *browser-browser* modern seperti Chrome, Firefox, Safari, Opera dan Internet Explorer. (Husein Alatas, 2015:2).

Beberapa alasan mengapa saat ini cukup banyak pengembang yang menggunakan *Bootstrap* dalam membuat *front-end website*, yaitu karena beberapa kelebihan yang dimiliki oleh *Bootstrap* itu sendiri yang antara lain:

1. Dapat mempercepat waktu proses pembuatan *front-end website* dan tampilan *bootstrap* yang sudah cukup terlihat modern.
2. Tampilan *Bootsrap* sudah *responsive*, sehingga mendukung segala jenis resolusi, baik itu PC, tablet, dan juga *smartphone*.
3. *Website* menjadi Sangat ringan ketika diakses, karena *bootsrap* dibuat dengan sangat terstruktur.

*Bootstrap* merupakan salah satu *framework* CSS paling populer dari sekian banyak *framework* CSS yang ada. *Bootstrap* memungkinkan desain sebuah web menjadi reponsif sehingga dapat dilihat dari berbagai macam ukuran device dengan tampilan tetap menarik. *Bootstrap* juga membuat proses pengaturan desain menjadi lebih cepat karena tidak perlu lagi banyak menulis CSS, bahkan hampir tidak perlu kecuali jika memerlukan pengaturan desain yang berbeda dengan *style bootstrap*. *Bootstrap* telah didukung oleh hampir semua *browser* baik pada *desktop* maupun *mobile*. (Rohi Abdulloh, 2018:261).

*Bootstrap* dapat didapatkan melalui *website* resminya yaitu getbootstrap.com. Dari hasil *download* *bootstrap* terdapat banyak sekali *file* dan *folder* didalamnya, namun yang paling utama yaitu *folder dist.* Folder ini yang perlu disalin ke direktori yang akan menggunakan bootstrap Cara menjalankan *bootstrap* seperti berikut :

* 1. Salin *folder* *dist* ke *folder* tempat penyimpanan *file website*. Sebaiknya ubah namanya menjadi *bootstrap* agar mudah dikenali.
  2. Tambahkan *tag* <meta name = “viewport”> pada elemen <head> pada file template yang menggunakan bootstrap untuk membuat web responsive.
  3. Panggil library JQuery dan *file* bootstrap.min.js atau bootstrap.bundle.min.js dibagian akhir elemen<body>.

# BAB III

**ANALISIS DAN PERANCANGAN SISTEM**

## Analisis Kebutuhan Sistem

Analisis kebutuhan sistem adalah sebuah tahapan perancangan sistem yang membahas tentang kebutuhan dalam membangun sebuah sistem. Analisis kebutuhan sistem dapat digolongkan menjadi dua bagian yaitu :

### Kebutuhan fungsional

Kebutuhan fungsional adalah kebutuhan yang berisikan proses – proses yang perlu disediakan oleh sistem. Kebutuhan fungsional mencangkup proses *input, output,* *process*, antara lain adalah:

1. Sistem pendukung keputusan rekomendasi pembelian rumah ini membutuhkan input berupa data pembeli dan penilaian terhadap rumah yang diinginkan dari calon pembeli untuk kemudian di proses dan menghasilkan *output* analisa alternatif – alternatif rumah yang cocok untuk calon pembeli di kota bekasi.
2. Sistem ini juga diharapkan akan untuk mengimplementasikan metode *Simple Additive Weighting (SAW).*
3. Sistem pendukung keputusan ini akan menghasilkan analisa alternatif rumah yang cocok untuk calon pembeli di kota bekasi.
4. Sistem pendukung keputusan ini terdapat halaman depan *(front end)* dan halaman admin *(back end).*

### Kebutuhan non-fungsional

Kebutuhan non-fungsional adalah kebutuhan yang disediakan oleh sistem yang bertujuan untuk mendukung kebutuhan fungsional yang sudah ditentukan. Kebutuhan non-fungsional dari sistem sistem pendukung rekomendasi rumah ini adalah :

1. Tampilan sistem pendukung yang di bangun memiliki tampilan yang *user friendly*  yang mudah digunakan oleh calon pembeli dengan tampilan *(interface)* yang sederhana dan mudah dimengerti.
2. Sistem pendukung keputusan rekomendasi rumah ini apat dijalankan oleh beberapa *web browser* diantaranya Internet Explore, Google Chrome dan Mozilla Firefox dan lainnya.

### 3.1.3 Perancangan *Simple Additive Weighting (SAW)*

Sistem pendukung keputusan untuk rekomendasi pembelian rumah di perumahan kota bekasi ini menggunakan metode *simple additive weighting (saw)* dengan memerlukan data perumahan untuk dapat menghitung rekomendasi pembelian rumah yang akan dipilih oleh calon pembeli.

* + - 1. **Kriteria**

Penulis telah melakukan studi pustaka sebelum menentukan kriteria – kriteria yang menjadi acuan dalam proses pengambilan keputusan dalam membeli rumah. Hasil dari studi pustaka dari beberapa buku yang telah di lakukan menerangkan bahwa kriteria yang digunakan dalam pembelian rumah adalah sebagai berikut :

* 1. **Harga**

Beberapa perumahan memang menetapkan harga rumah yang selangit karena infrastrukturnya sudah lengkap. Pembangunan jalan tol, misalnya, bisa membuat harga

rumah naik 50 persen. Bertindaklah pandai, jika ada rencana pembangunan di kawasan tersebut jangan ragu membelinya. Tapi pastikan tempat tersebut aman dari pembebasan tanah. (Suprayitno, 2015:36). Kriteria harga memiliki tipe kriteria berjenis *cost* karena jika rumah semakin tinggi harga nya maka semakin kecil peminatnya.

* 1. **Jarak Lokasi Rumah Ke Pusat Kota**

Lokasi sangat berhubungan dengan akses rumah dari jalan dan kendaraan untuk mencapainya. Kebanyakan rumah yang letaknya dekat stasiun atau jalan raya, umumnya harganya akan lebih tinggi dibanding perumahan yang letaknya jauh dari jalan raya. Selain akses kendaraan, lokasi rumah yang dipilih juga sebaiknya berada di lingkungan yang aman dan tidak berisiko banjir saat musim hujan. Begitu juga fasilitas rumah, pastikan saat membeli sarana seperti listrik, air bersih, dan sarana lainnya telah tersedia secara memadai. (Suprayitno, 2015:34). Kriteria lokasi rumah di ambil dari jarak perumahan ke pusat kota bekasi. Kriteria lokasi rumah memiliki tipe kriteria berjenis *benefit* karena jika rumah dengan lokasi yang strategis maka semakin nyaman tinggal dirumah tersebut.

* 1. **Aksesbilitas Menuju Perumahan**

Aksesibilitas merupakan informasi akses jalan yang dapat ditempuh menuju perumahan sedangkan marketabilitas merupakan faktor pendukung yang dapat meningkatkan atau pun menurunkan nilai rumah yang akan dibeli. Calon pembeli hendaknya menanyakan master plan atas kawasan perumahan yang dikembangkan, agar dapat diperoleh informasi mengenal pengembangan perumahan di masa yang akan datang.(Suprayitno, 2015:35). Kriteria aksebilitas memiliki tipe kriteria berjenis *benefit* karena semakin mudah akses maka semakin mudah untuk menjangkau lokasi.

* 1. **Legalitas Rumah**

Hindari membeli rumah yang tidak jelas legalitasnya. Tidak jarang rumah berada di lokasi yang strategis namun tidak memiliki legalitas yang lengkap sehingga menimbulkan permasalahan di kemudian hari. Untuk pembelian rumah di developer, calon pembeli dapat menanyakan status tanah dan perizinan yang dimiliki oleh developer tersebut. (Suprayitno, 2015:35). Kriteria legalitas memiliki tipe kriteria berjenis *benefit* karena jika legalitas lengkap maka semakin aman dalam memiliki rumah.

* 1. **Fasilitas Umum Sekitar Rumah**

Untuk pembelian rumah di developer, calon pembeli dapat menanyakan fasum atau fasos yang akan disediakan oleh developer. Tidak jarang developer hanya menyediakan lahan saja kepada warga. Hal ini perlu diklarifikasi agar tidak menjadi kekecewaan di kemudian hari. Sedangkan pembangunan fasum/fasos diserahkan komplek perumahan umumnya akan dilengkapi dengan fasilitas sosial, seperti sekolah, rumah sakit, rumah ibadah, fasilitas hiburan seperti arena bermain anak, kolam renang dan sarana olahraga lainnya, taman, bioskop serta pertokoan. (Suprayitno, 2015:37). Kriteria fasilitas memiliki tipe kriteria berjenis *benefit* karena jika jumlah fasilitas rumah semakin banyak maka semakin banyak kemudahan yang didapat dalam rumah tersebut.

* 1. **Desain Rumah**

Mempunyai suatu rumah idaman memang menjadi impian bagi banyak orang, akan tetapi rumah seperti apa yang betul - betul anda idamkan? Apalagi sekarang banyak bermunculan berbagai model-model rumah yang mungkin menjadi salah satu pilihan sebagai kriteria rumah idaman anda, berdasarkan suatu survei yang telah dilakukan tentang kriteria apa yang diinginkan dalam sebuah bangunan rumah, didapatkan dari keseluruhan mengatakan bahwa rumah idaman ialah rumah yang minimalis, sehingga tidak heran jika rumah idaman minimalis menjadi salah satu rumah idaman dari kebanyakan orang. (Suprayitno, 2015:35). Kriteria deain rumah memiliki tipe kriteria berjenis *benefit* karena jika desain rumah semakin bagus maka semakin menarik rumah tersebut.

### Perancangan Proses Pendukung Keputusan

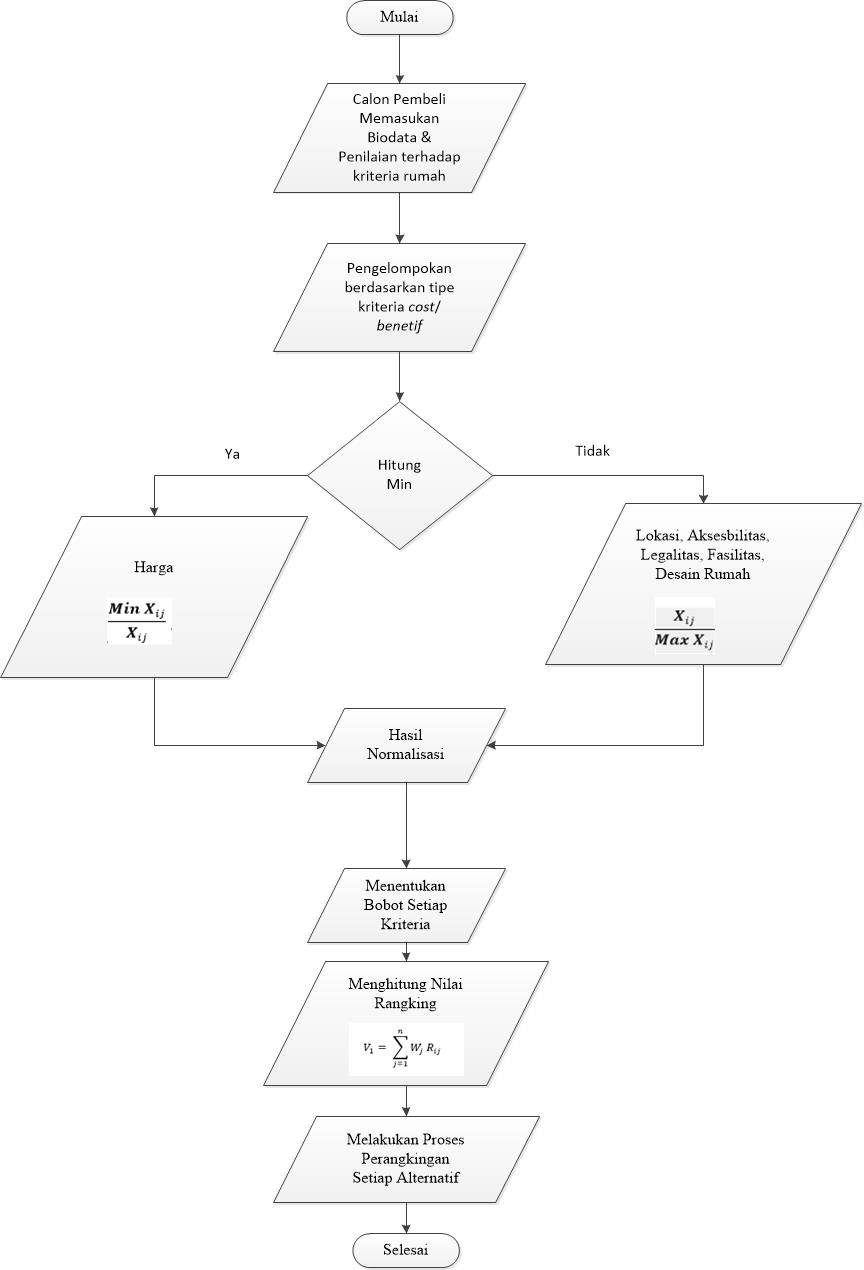
Pada sistem pengambilan keputusan untuk rekomendasi pembelian rumah ini terdapat empat proses utama yaitu data alternatif, data kriteria, data nilai dan perangkingan. proses perangkingan dimulai dengan langkah awal yaitu calon pembeli wajib memasukan biodata nama lengkap dan umur kemudian memasukan penilaian bobot terhadap kriteria yang sesuai dengan kebutuhan calon pembeli. Data kriteria tersebut diantaranya kelompok kriteria biaya (*cost*) yaitu harga. Kelompok kriteria keuntungan *(benefit)* yaitu lokasi rumah, aksebilitas, legalitas, fasilitas, desain rumah.

**Tabel 3. 1** Penentuan Kriteria

|  |  |
| --- | --- |
| Kriteria | Tipe Kriteria |
| Harga | *cost* |
| Jarak Lokasi Rumah Ke Pusat Kota | *benefit* |
| Aksesibilitas Menuju Perumahan | *benefit* |
| Legalitas Rumah | *benefit* |
| Fasilitas Umum Sekitar Rumah | *benefit* |
| Desain Rumah | *benefit* |

Setelah dilakukan perhitungan pada setiap alternatif berdasarkan kelompok rumus yang telah ditentukan, hasil dari perhitungannya yaitu berupa matriks normalisasi kemudian pengguna memasukkan bobot kepentingan dari setiap kriteria yang ada setelah memasukkan bobot kepentingan dari setiap kriteria maka akan dilakukan proses perhitungan.

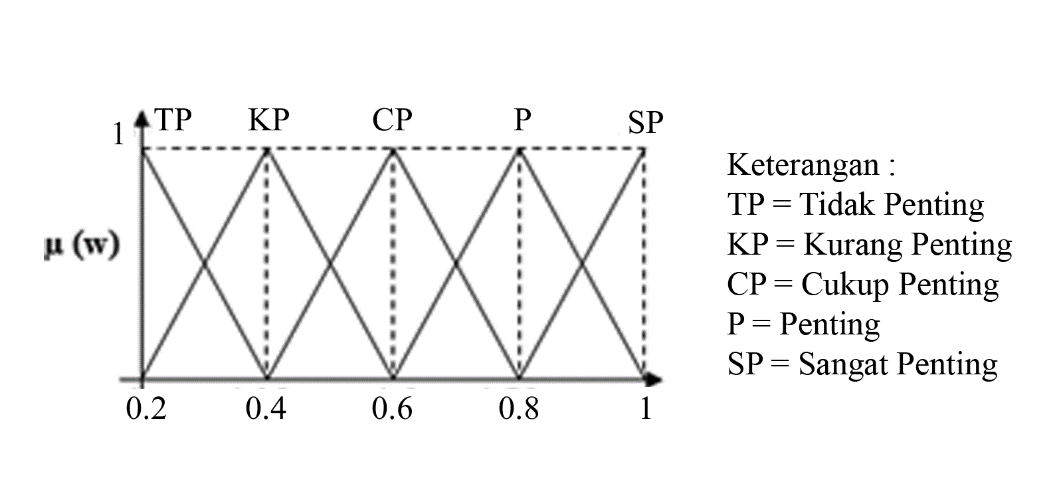
Pada tahap perangkingan dilakukan penyusunan peringkat atau rangking yang didasarkan pada hasil perkalian matrik ternormalisasi dengan bobot kepentingan berikut ini adalah proses proses pendukung keputusan pada sistem pendukung keputusan untuk rekomendasi pembelian rumah.



**Gambar 3. 1** Proses Pengam­bilan Keputusan

### Proses Perhitungan *Simple Additive Weighting (SAW)*

Masing-masing kriteria yang telah ditetapkan diatas akan dibobotkan berdasarkan bilangan *fuzzy*. Ada enam bilangan *fuzzy* yang digunakan dalam pembombotan ini, yaitu SP= Sangat Penting, KP= Kurang Penting CP= Cukup Penting, P= Penting, TP= Tidak Penting, seperti terlihat pada gambar berikut :



Bilangan-bilangan *fuzzy* di atas dikonversikan ke bilangan *crisp* sehingga seperti pada tabel berikut :

**Tabel 3. 2** Pembobotan Kriteria

|  |  |
| --- | --- |
| Bilangan *Fuzzy* | Nilai |
| Sangat Penting | 1 |
| Penting | 0,8 |
| Cukup Penting | 0,6 |
| Kurang Penting | 0,4 |
| Tidak Penting | 0,2 |

Penelitian ini diawali dengan pengamatan menggunakan metode observasi dengan menyajikan rangkuman hasil survey dan data yang akan digunakan adalah data sekunder dari 10 perumahan yang ada di Kota Bekasi. Kemudian melakukan penilain terhadap perumahan yang telah dijadikan alternatif berdasarkan pemilihan kriteria yang telah di tentukan sebelumnya.

**Tabel 3. 3** Nilai Ranting Masing - Masing Kriteria

|  |  |  |  |
| --- | --- | --- | --- |
| Alternatif | Kondisi | Bilangan Fuzzy | Nilai |
| Harga | < Rp. 450 Jt | SP | 1 |
| Rp. 510 – Rp. 600 Jt | P | 0.8 |
| Rp. 610 – 700 Jt | CP | 0.6 |
| Rp. 510 – Rp. 600 Jt | KP | 0.4 |
| > Rp. 800 Jt | TP | 0.2 |
| Jarak Lokasi Rumah Dari Pusat Kota | < 5 KM | SP | 1 |
| 6 - 10 KM | P | 0.8 |
| 11 - 15 KM | CP | 0.6 |
| 16 – 20 KM - | KP | 0.4 |
| > 20 KM | TP | 0.2 |
| Aksesbilitas Menuju Rumah | Sangat Bagus | SP | 1 |
| Bagus | P | 0.8 |
| Cukup Bagus | CP | 0.6 |
| Kurang Bagus | KP | 0.4 |
| Tidak Bagus | TP | 0.2 |
| Legalitas Rumah | Sangat Lengkap | SP | 1 |
| Lengkap | P | 0.8 |
| Cukup Lengkap | CP | 0.6 |
| Kurang Legkap | KP | 0.4 |
| Tidak Lengkap | TP | 0.2 |
| Fasilitas Rumah | Sangat Lengkap | SP | 1 |
| Lengkap | P | 0.8 |
| Cukup Lengkap | CP | 0.6 |
| Kurang Legkap | KP | 0.4 |
| Tidak Lengkap | TP | 0.2 |
| Desain Rumah | Sangat Bagus | SP | 1 |
| Bagus | P | 0.8 |
| Cukup Bagus | CP | 0.6 |
| Kurang Bagus | KP | 0.4 |
| Tidak Bagus | TP | 0.2 |

**Tabel 3. 4** Data Alternatif Rumah

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Alternatif | Harga | Lokasi Rumah | Aksesbilitas | Legalitas | Fasilitas | Desain Rumah |
| Golden Mozaik 3 | 1 | 0.4 | 0.6 | 1 | 0.8 | 1 |
| Komplek Wira Jatiasih Greenhills | 0.2 | 0.6 | 0.4 | 1 | 0.6 | 1 |
| Ziefa Cluster | 1 | 0.4 | 0.2 | 1 | 0.2 | 1 |
| Prima Pedurenan 5 | 1 | 0.6 | 0.6 | 1 | 0.4 | 1 |
| Mutiara Asgo Residence | 0.6 | 0.6 | 0.8 | 1 | 0.6 | 1 |
| Taman Firdaus 1 | 1 | 0.8 | 0.6 | 1 | 0.6 | 1 |
| Royal Benda Residence | 0.6 | 0.8 | 0.8 | 1 | 0.6 | 1 |
| Cluster Kiana Jatiasih | 0.6 | 0.8 | 0.8 | 1 | 0.8 | 1 |
| Green Pratama Residence | 1 | 0.6 | 0.6 | 1 | 0.6 | 1 |
| Perumahan Grand Victoria | 0.8 | 0.4 | 0.8 | 1 | 0.8 | 1 |

Dari data di atas kemudian kita rubah menjadi matriks.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 0.4 | 0.6 | 1 | 0.8 | 1 |
|  | 0.2 | 0.6 | 0.4 | 1 | 0.6 | 1 |
|  | 1 | 0.4 | 0.2 | 1 | 0.2 | 1 |
|  | 1 | 0.6 | 0.6 | 1 | 0.4 | 1 |
| x = | 0.6 | 0.6 | 0.8 | 1 | 0.6 | 1 |
|  | 1 | 0.8 | 0.6 | 1 | 0.6 | 1 |
|  | 0.6 | 0.8 | 0.8 | 1 | 0.6 | 1 |
|  | 0.6 | 0.8 | 0.8 | 1 | 0.8 | 1 |
|  | 1 | 0.6 | 0.6 | 1 | 0.6 | 1 |
|  | 0.8 | 0.4 | 0.8 | 1 | 0.8 | 1 |

Berdasarkan tabel diatas kemudian diubah menjadi matriks.

W = [ 1 0.8 0.6 0.4 0.2 0.2 ]

Nilai kriteria harga kemudian dinormalisasi seperti cara dibawah ini.

r1 =

r2 =

r3 =

r4 =

r5 =

r6 =

r7 =

r8 =

r9 =

r10 =

Nilai kriteria lokasi rumah kemudian dinormalisasi seperti cara dibawah ini.

r1 =

r2 =

r3 =

r4 =

r5 =

r6 =

r7 =

r8 =

r9 =

r10=

Nilai kriteria aksesbilitas kemudian dinormalisasi seperti cara dibawah ini.

r1 =

r2 =

r3 =

r4 =

r5 =

r6 =

r7 =

r8 =

r9 =

r10 =

Nilai kriteria jumlah legalitas kemudian dinormalisasi seperti cara dibawah ini.

r1 =

r2 =

r3 =

r4 =

r5 =

r6 =

r7 =

r8 =

r9 =

r10 =

Nilai kriteria fasilitas kemudian dinormalisasi seperti cara dibawah ini.

r1 =

r2 =

r3 =

r4 =

r5 =

r6 =

r7 =

r8 =

r9 =

r10 =

Nilai kriteria desain rumah kemudian dinormalisasi seperti cara dibawah ini.

r1 =

r2 =

r3 =

r4 =

r5 =

r6 =

r7 =

r8 =

r9 =

r10 =

Berdasarkan nilai normalisasi yang sudah di dapat maka jika disajikan dalam bentuk tabel akan terlihat seperti tabel berikut.

**Tabel 3. 5** Normalisasi Kriteria

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Nama | Harga | Lokasi Rumah | Aksesbilitas | Legalitas | Fasilitas | Desain Rumah |
| Golden Mozaik 3 | 0.2 | 0.5 | 0.75 | 1 | 1 | 1 |
| Komplek Wira Jatiasih Greenhills | 1 | 0.75 | 0.5 | 1 | 0.75 | 1 |
| Ziefa Cluster | 0.2 | 0.75 | 0.25 | 1 | 0.25 | 1 |
| Prima Pedurenan 5 | 0.2 | 1 | 0.75 | 1 | 0.5 | 1 |
| Mutiara Asgo Residence | 0.3333 | 0.75 | 1 | 1 | 0.75 | 1 |
| Taman Firdaus 1 | 0.2 | 1 | 0.75 | 1 | 0.75 | 1 |
| Royal Benda Residence | 0.3333 | 1 | 1 | 1 | 0.75 | 1 |
| Cluster Kiana Jatiasih | 0.3333 | 1 | 1 | 1 | 1 | 1 |
| Green Pratama Residence | 0.2 | 0.75 | 0.75 | 1 | 0.75 | 1 |
| Perumahan Grand Victoria | 0.25 | 0.5 | 1 | 1 | 1 | 1 |

Jika tabel hasil normalisasi diatas disajikan dalam bentuk matriks maka dapat dilihat dibawah ini.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 0.2 | 0.5 | 0.75 | 1 | 1 | 1 |
|  | 1 | 0.75 | 0.5 | 1 | 0.75 | 1 |
|  | 0.2 | 0.75 | 0.25 | 1 | 0.25 | 1 |
|  | 0.2 | 1 | 0.75 | 1 | 0.5 | 1 |
| x = | 0.3333 | 0.75 | 1 | 1 | 0.75 | 1 |
|  | 0.2 | 1 | 0.75 | 1 | 0.75 | 1 |
|  | 0.3333 | 1 | 1 | 1 | 1 | 1 |
|  | 0.3333 | 1 | 1 | 1 | 1 | 1 |
|  | 0.2 | 0.75 | 0.75 | 1 | 0.75 | 1 |
|  | 0.25 | 0.5 | 1 | 1 | 1 | 1 |

Untuk menentukan hasil rekomendasi pemilihan dilakukan perangkingan dengan rumus :

V = w x r

Dimana w adalah bobot yang sudah diteentukan oleh customer dan r merupakan normalisasi kriteria.

W = [ 1 0.8 0.6 0.4 0.2 0.2 ]

V1 = (0.2)1 + (0.5)0.8 + (0.75)0.6 + (1)0.4 + (1)0.2 + (1)0.2 = 1.8

V2 = (1)1 + (0.75)0.8 + (0.5)0.6 + (1)0.4 + (0.75)0.2 + (1)0.2 = 2.65

V3 = (0.2)1 + (0.75)0.8 + (0.25)0.6 + (1)0.4 + (0.75)0.2 + (1)0.2 = 1.7

V4 = (0.2)1 + (0.75)0.8 + (1)0.6 + (1)0.4 + (0.75)0.2 + (1)0.2 = 1.95

V5 = (0.3333)1 + (0.75)0.8 + (0.75)0.6 + (1)0.4 + (0.5)0.2 + (1)0.2 = 2.15

V6 = (0.2)1 + (1)0.8 + (0.75)0.6 + (1)0.4 + (0.75)0.2 + (1)0.2 = 2.2

V7 = (0.3333)1 + (1)0.8 + (1)0.6 + (1)0.4 + (0.75)0.2 + (1)0.2 = 2.4833

V8 = (0.3333)1 + (1)0.8 + (1)0.6 + (1)0.4 + (1)0.2 + (1)0.2 = 2.5333

V9 = (0.2)1 + (0.75)0.8 + (0.75)0.6 + (1)0.4 + (0.75)0.2 + (1)0.2 = 2

V10 = (0.25)1 + (0.5)0.8 + (1)0.6 + (1)0.4 + (1)0.2 + (1)0.2 = 2.05

Dari hasil diatas maka dapat dilihat data yang memiliki nilai paling tinggi adalah V2 atau Komplek Wira Jatiasih Greenhills. Jika disajikan dalam tabel maka dapat dilihat dari tabel dibawah ini.

**Tabel 3. 6** Hasil Perangkingan

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No | Nama | Harga | Lokasi Rumah | Aksesbilitas | Legalitas | Fasilitas | Desain Rumah | Jumlah SAW |
| 1 | Cluster Kiana Jatiasih | 0.2 | 0.4 | 0.45 | 0.4 | 0.2 | 0.2 | 1.8 |
| 2 | Royal Benda Residence | 1 | 0.6 | 0.3 | 0.4 | 0.15 | 0.2 | 2.65 |
| 3 | Perumahan Grand Victoria | 0.2 | 0.6 | 0.15 | 0.4 | 0.15 | 0.2 | 1.7 |
| 4 | Taman Firdaus 1 | 0.2 | 0.6 | 0.6 | 0.4 | 0.15 | 0.2 | 1.95 |
| 5 | Mutiara Asgo Residence | 0.3333 | 0.6 | 0.45 | 0.4 | 0.1 | 0.2 | 2.15 |
| 6 | Green Pratama Residence | 0.2 | 0.8 | 0.45 | 0.4 | 0.15 | 0.2 | 2.2 |
| 7 | Komplek Wira Jatiasih Greenhills | 0.333 | 0.8 | 0.6 | 0.4 | 0.15 | 0.2 | 2.4833 |
| 8 | Prima Pedurenan 5 | 0.333 | 0.8 | 0.6 | 0.4 | 0.2 | 0.2 | 2.5333 |
| 9 | Golden Mozaik 3 | 0.2 | 0.4 | 0.6 | 0.4 | 0.15 | 0.2 | 2 |
| 10 | Ziefa Cluster | 0.25 | 0.4 | 0.6 | 0.4 | 0.2 | 0.2 | 2.05 |

## Sistem Desain

Pada penelitian ini menggunakan *UML (Unified Modeling Language)* sebagai bahasa pemodelan untuk mendesain dan merancang Sistem Pendukung Keputusan Untuk Rekomendasi Pembelian Rumah di Kota Bekasi Dengan Menggunakan Metode *Simple Additive Weighting (SAW).* Model UML yang digunakan dalam perancangan sistem ini antara lain *use case diagram, calss diagram, activity diagram, & sequence diagram.*

### *Use Case Diagram*

*use case diagram* di bawah ini akan menggambarkan interaksi antara aktor dengan sistem pendukung keputusan rekomendasi pembelian rumah yang akan dibangun.



**Gambar 3. 2** Use Case Diagram

Di bawah ini terdapat beberapa skenario *Use Case* yang berjalan pada sistem pendukung keputusan untuk rekomendasi rumah diantaranya sebagai berikut :

1. *Use Case* Mencari Alternatif Rumah

Kondisi awal : Menampilkan Menu Cari Rumah

Skenario : Calon pembeli menekan menu cari kemudin memasukan biodata dan penilaian rumah terhadap beberapa kriteria sesuai penilaian, lalu tekan tombol proses untuk melihat hasil rekomendasi rumah.

Kondisi akhir : Dapat melihat hasil dari alternatif rumah.

1. *Use Case* Melihat Data Perumahan

Kondisi awal : Menampilkan menu peta lokasi.

Skenario : Calon pembeli menekan menu peta lokasi dan akan melihat beberapa lokasi yang menjadi alternatif perumahan yang tersedia.

Kondisi akhir : Dapat melihat hasil dari alternatif rumah melalui map.

1. *Use Case* Melihat Panduan Penggunaan

Kondisi awal : Menampilkan Melihat Panduan Penggunaan.

Skenario : Calon pembeli menekan menu panduan dan akan melihat langkah – langh dalam menjalankan sistem pendukung keputusan.

Kondisi akhir : Dapat melihat panduan penggunaan sistem pendukung keputusan.

1. *Use Case* Tentang Sistem Pendukung Keputusan

Kondisi awal : Tampilkan menu tentang*.*

Skenario : Calon pembeli menekan menu tentang dan akan melihat tentang sistem pendukung keputusan yang telah dibangun.

Kondisi akhir : calon pembeli akan melihat menu tentang pada halaman konten yang tersedia.

1. *Use Case Login*

Kondisi awal : Tampilan halaman login dengan tampilan berisi kolom username dan password, serta tombol *login*.

Skenario : Admin mengisi kolom *username* dan *password* dan menekan tombol login pada form yang tersedia.

Kondisi akhir : Jika berhasil maka admin diarahkan menuju halaman admin sistem rekomendasi pemilihan rumah, dan jika salah admin akan diberi peringatan gagal *login* serta harus mencoba melakukan login kembali.

1. *Use Case* Mengelola Data Nilai

Kondisi awal : Tampilan menu data nilai

Skenario : Admin dapat mengelola data nilai, mengahapus data nilai, mengedit data nilai dan menambah data nilai ke dalam sistem pendukung keputusan untuk rekomendasi pembelian rumah.

Kondisi akhir : Mendapatkan data nilai yang sudah di kelola.

1. *Use Case* Mengelola Data Kriteria

Kondisi awal : Tampilan data kriteria

Skenario : Admin dapat mengelola data kriteria, mengahapus data kriteria, mengedit data kriteria dan menambah data kriteria ke dalam sistem pendukung keputusan untuk rekomendasi pembelian rumah.

Kondisi akhir : Mendapatkan data kriteria yang sudah di kelola.

1. *Use Case* Mengelola Data Alternatif Rumah

Kondisi awal : Tampilan Data Alternatif Rumah

Skenario : Admin dapat mengelola data alternatif rumah, mengahapus data alternatif rumah, mengedit data alternatif rumah dan menambah data alternatif rumah ke dalam sistem pendukung keputusan untuk rekomendasi pembelian rumah.

Kondisi akhir : Mendapatkan data alternatif rumah yang sudah di kelola.

1. *Use Case* Mengelola Data Proses Rangking

Kondisi awal : Tampilan Data Proses Rangking

Skenario : Admin dapat mengelola data rangking, mengahapus, mengedit data dan menambah data ke dalam sistem pendukung keputusan untuk rekomendasi pembelian rumah.

Kondisi akhir : Mendapatkan data rangking yang sudah di kelola.

1. *Use Case* Mencetak Laporan

Kondisi awal : Tampilan Laporan

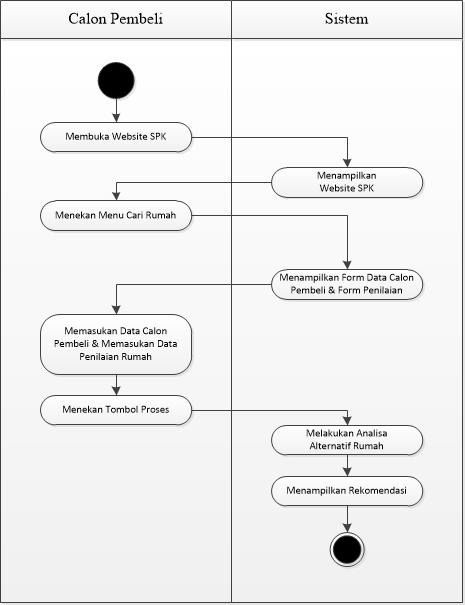
Skenario : Admin dapat mencetak data perangkingan rumah yang sudah dihitung menggunakan metode saw.

Kondisi akhir : Mendapatkan data alternatif rumah yang sudah di kelola.

### *Activity Diagram*

*Activity Diagram* atau diagram aktivitas adalah sebuah diagram yang menggambarkan urutan kegiatan atau urutan aktivitas dari sebuah sistem yang di bangun. Dibawah ini adalah *activity diagram* dari sistem pendukung keputusan untuk rekomendasi rumah yang akan di bangun.

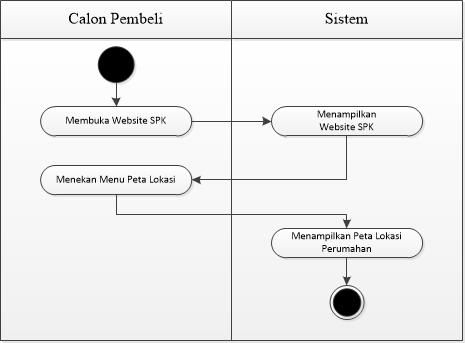
1. *Activity Diagram* Mencari Alternatif Rumah



**Gambar 3. 3** Activity Diagram Mencari Alternatif Rumah

Pada gambar di atas menerangkan bahwa calon pembeli membuka *website* SPK kemudian sistem menampilkan w*ebsite* SPK dan calon pembeli kemudian menekan menu cari rumah dan sistem akan menampilkan form data calon pembeli dan form penilaian, calon pembeli memasukan biodata dan memasukan data penilaian rumah yang sesuai kebutuhan calon pembeli.

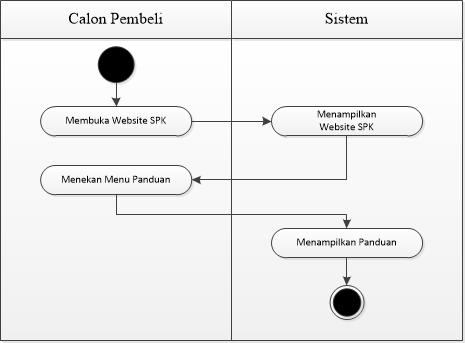
1. *Activity Diagram* Melihat Data Perumahan



**Gambar 3. 4** Activity Diagram Data Perumahan

Pada gambar di atas menjelaskan bahwa calon pembeli membuka *website* SPK dan kemudian sistem menampilkan *website* SPK, calon pembeli menekan menu peta lokasi dan kemudian sistem akan menampilkan peta lokasi perumahan.

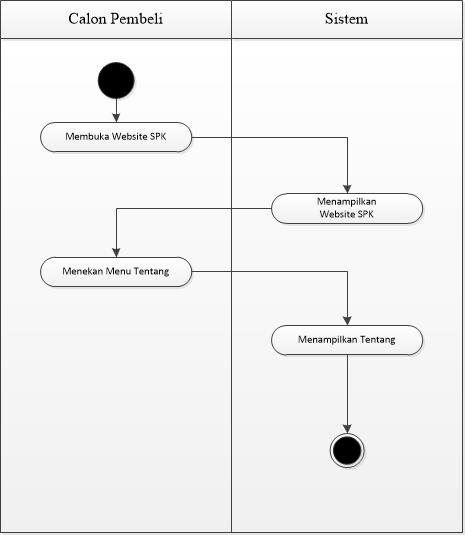
1. *Activity Diagram* Melihat Panduan Penggunaan SPK



**Gambar 3. 5** Activity Diagram Melihat Panduan Penggunaan SPK

Pada gambar di atas menjelaskan bahwa calon pembeli membuka *website* SPK dan kemudian sistem menampilkan *website* SPK, calon pembeli menekan menu panduan dan kemudian sistem akan menampilkan panduan dalam menggunakan SPK.

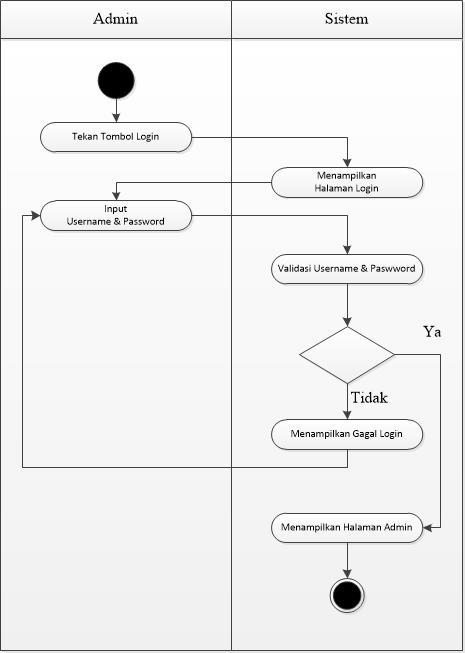
1. *Activity Diagram* Melihat Tentang SPK



**Gambar 3. 6** Activity Diagram Tentang SPK

Pada gambar di atas menjelaskan bahwa calon pembeli membuka *website* SPK dan kemudian sistem menampilkan *website* SPK, calon pembeli menekan menu tetang dan kemudian sistem akan menampilkan informasi tentang sistem ini dibangun.

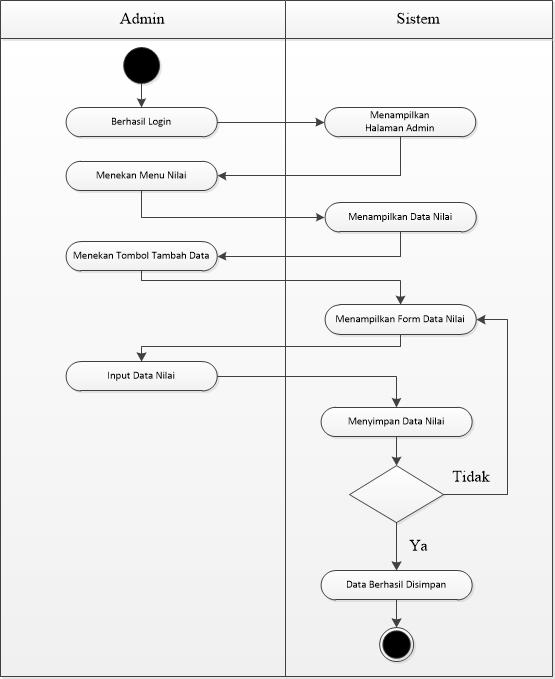
1. *Activity Diagram* *Login*



**Gambar 3. 7** Activity Diagram Login

Admin menekan menu login dan kemudian sistem akan menampilkan form *login* dan admin harus memasukan *username* dan *password*, setelah itu sistem akan memeriksa *username* dan *password*, jika benar maka sistem akan menampilkan halaman admin, dan jika salah maka sistem akan menampilkan pesan gagal.

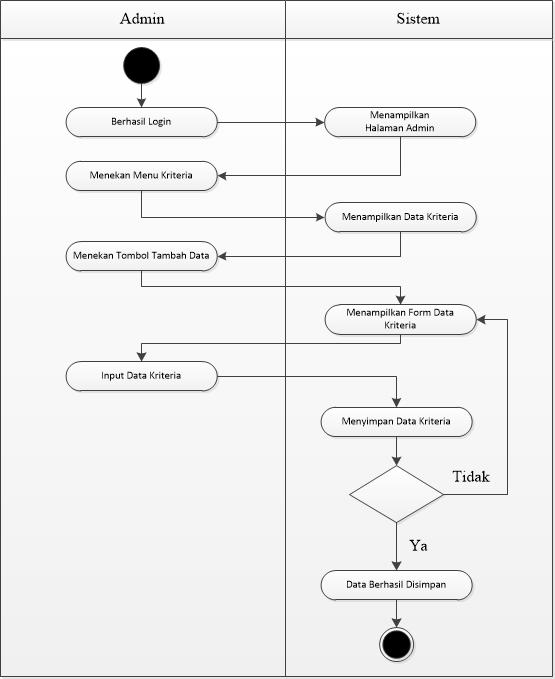
1. *Activity Diagram* Kelola Data Nilai



**Gambar 3. 8** Activity Diagram Data Nilai

Admin berhasil login dan kemudian sistem menampilkan halaman admin, admin menekan menu nilai dan sistem menampilkan data nilai, admin menekan tambah data dan sistem akan menampikan form data nilai, kemudian admin input data nilai, kemudian jika ya maka data berhasil di simpan, dan jika tidak maka akan kembali ke form data nilai.

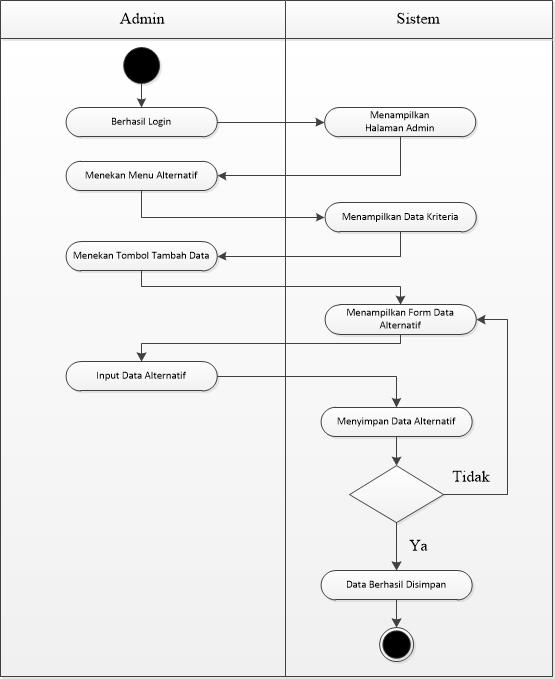
1. *Activity Diagram* Kelola Data Kriteria



**Gambar 3. 9** Activity Diagram Kelola Data Kriteria

Admin berhasil login dan kemudian sistem menampilkan halaman admin, admin menekan menu kriteria dan sistem menampilkan data kriteria, admin menekan tambah data dan sistem akan menampikan form data kriteria, kemudian admin input data kriteria, kemudian jika ya maka data berhasil di simpan, dan jika tidak maka akan kembali ke form data kriteria.

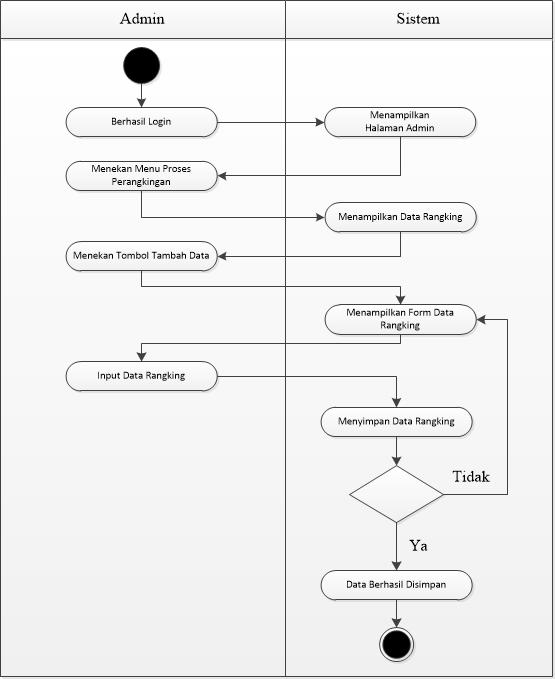
1. *Activity Diagram* Kelola Data Alternatif Rumah



**Gambar 3. 10** Activity Diagram Data Alternatif Rumah

Admin berhasil login dan kemudian sistem menampilkan halaman admin, admin menekan menu alternatif dan sistem menampilkan data alternatif, admin menekan tambah data dan sistem akan menampikan form data alternatif, kemudian admin input data alternatif, kemudian jika ya maka data berhasil di simpan, dan jika tidak maka akan kembali ke form data alternatif.

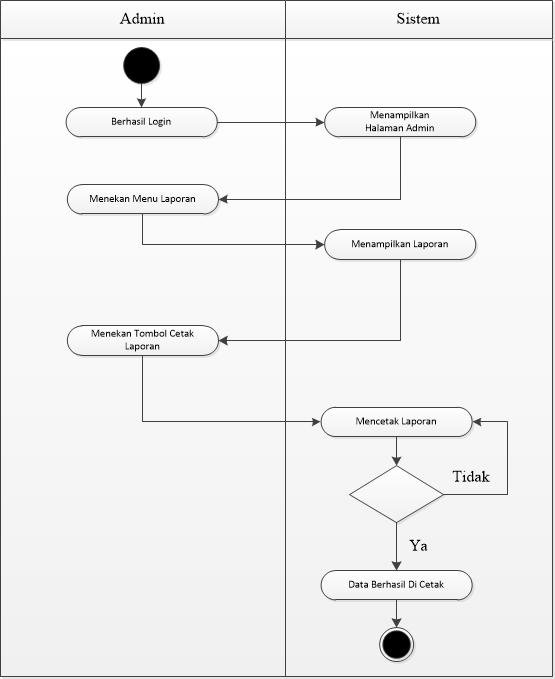
1. *Activity Diagram* Kelola Proses Perangkingan



**Gambar 3. 11** Activity Diagram Data Proses Perangkingan

Admin berhasil login dan kemudian sistem menampilkan halaman admin, admin menekan menu proses perangkingan dan sistem menampilkan data rangking, admin menekan tambah data dan sistem akan menampikan form data rangking, kemudian admin input data rangking, kemudian jika ya maka data berhasil di simpan, dan jika tidak maka akan kembali ke form data rangking.

1. *Activity Diagram* Mencetak Laporan



**Gambar 3. 12** Activity Diagram Mencetak Laporan

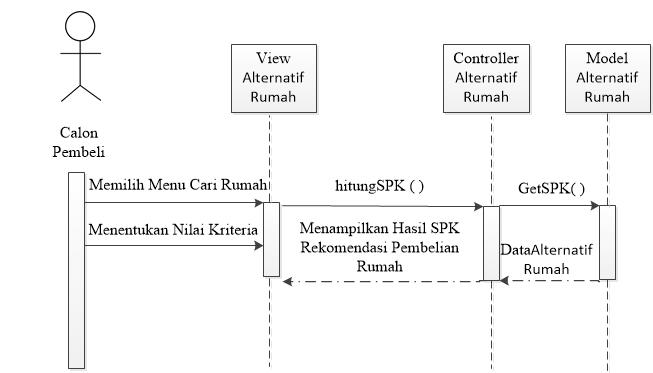
Admin berhasil login dan kemudian sistem menampilkan halaman admin, admin menekan menu laporan perangkingan dan sistem menampilkan laporan perangkingan, kemudian admin dapat menekan cetak laporan.

### *Sequence Diagram*

Sequence Diagram adalah sebuah diagram yang menggambarkan interaksi antar obyek dalam sebuah sistem. Diagram ini menunjukan diagram yang menggambarkan kolaborasi dinamis antara sejumlah object. Fungsinya adalah untuk menunjukkan rangkaian pesan yang dikirim antara object juga interaksi antara object. Sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem. Di bawah ini terdapat beberapa sequence diagram diantaranya sebagai berikut :

Dalam *sequence diagram* setiap *object* hanya memiliki garis yang digambarkan garis putus-putus ke bawah. Pesan antar *object* digambarkan dengan anak panah dari *object* yang mengirimkan pesan ke *object* yang menerima pesan.

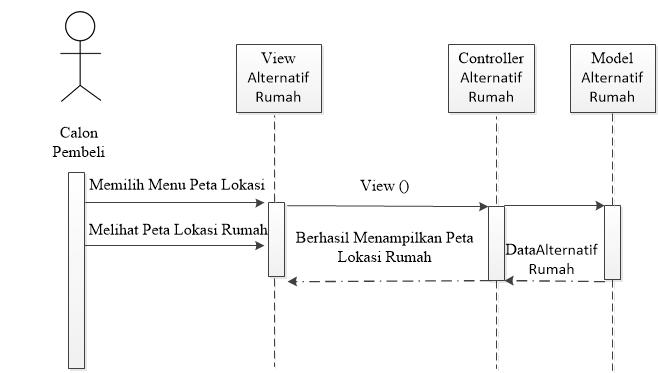
1. *Sequence Diagram* Mencari Alternatif Rumah



**Gambar 3. 13** Sequence Diagram Mencari Alternatif Rumah

Pada gambar diatas dapat dilihat bahwa calon pembeli memilih menu cari rumah pada View Rekomendasi Rumah, kemudian Controller Rekomendasi Rumah menjalankan fungsi hitungSPK dan diteruskan ke Model Alternatif Rumah dengan menjalankan fungsi getSPK. Hasil keluaran dari Model Alternatif Rumah berupa data hasil rekomendasi rumah yang sudah diolah oleh sistem pendukung keputusan.

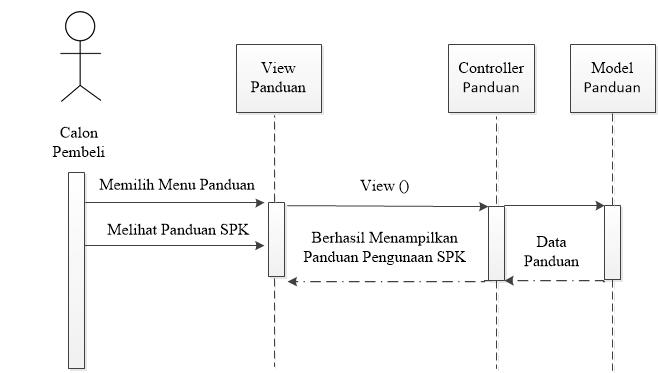
1. *Sequence Diagram* Melihat Data Perumahan



**Gambar 3. 14** Sequence Diagram Melihat Data Perumahan

Pada gambar diatas dapat dilihat bahwa calon pembeli memilih menu Peta Lokasi pada View Rekomendasi Rumah, kemudian Controller Rekomendasi Rumah menjalankan fungsi view dan diteruskan ke Model Alternatif Rumah. Hasil keluaran dari Model Alternatif Rumah berupa data peta loksai perumahan yang tersedia didalam rumah yang sudah diolah oleh sistem pendukung keputusan.

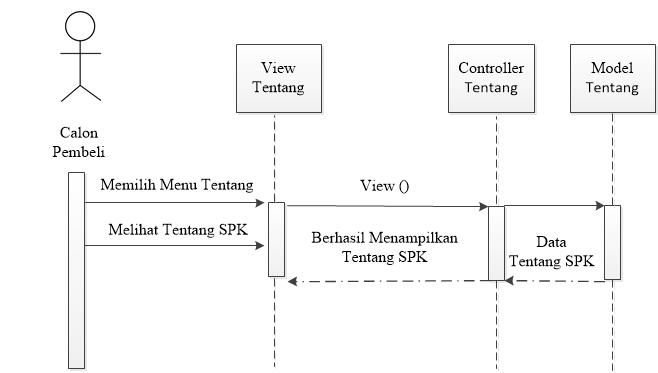
1. *Sequence Diagram* Melihat Panduan



**Gambar 3. 15** Sequence Diagram Melihat Panduan

Pada gambar diatas dapat dilihat bahwa calon pembeli memilih menu Panduan pada View Panduan, kemudian Controller Rekomendasi Rumah menjalankan fungsi view dan diteruskan ke Model Panduan. Hasil keluaran dari Model Panduan berupa isi panduan sistem pendukung keputusan rekomendasi pembelian rumah perumahan yang tersedia didalam sistem pendukung keputusan.

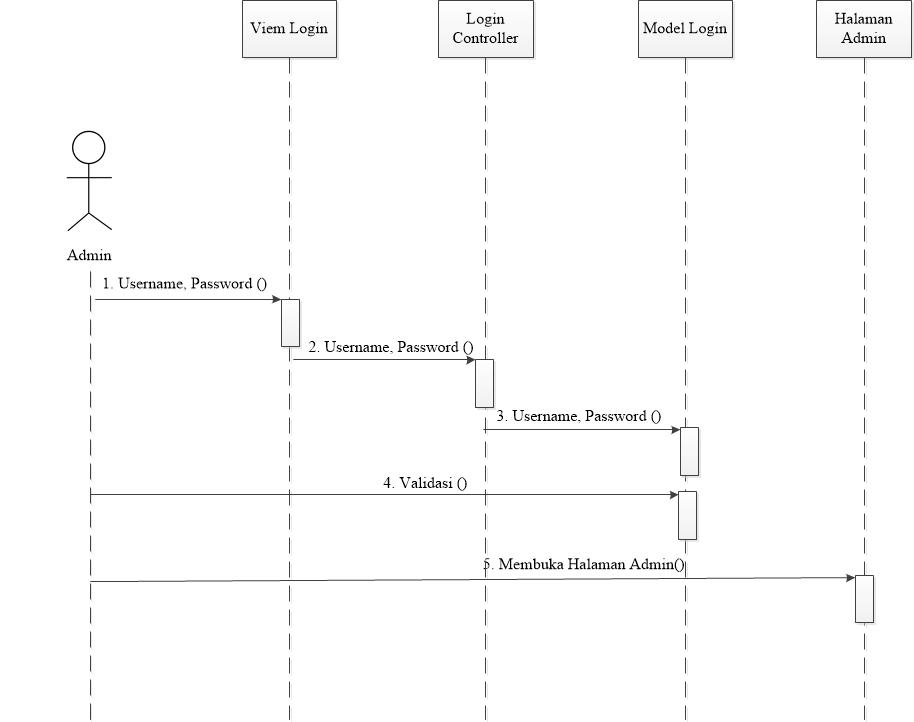
1. *Sequence Diagram* Melihat Tentang



**Gambar 3. 16** Sequence Diagram Melihat Tentang

Pada gambar diatas dapat dilihat bahwa calon pembeli memilih menu tentang pada View tentang, kemudian Controller Rekomendasi Rumah menjalankan fungsi view dan diteruskan ke Model tentang. Hasil keluaran dari Model tentang berupa isi tentang sistem pendukung keputusan rekomendasi pembelian rumah yang tersedia yang sudah diolah oleh.

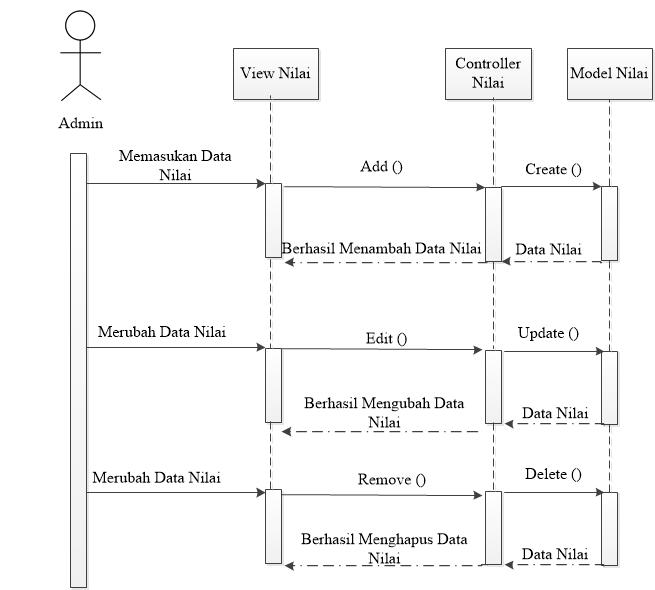
1. *Sequence Diagram* *Login Admin*



**Gambar 3. 17** Sequence Diagram Login Admin

Dari gambar diatas dapat dilihat bahwa admin memasukkan data *username* dan *password* ke dalam layar atau form *Login*, kemudian meneruskannya ke login controller lalu diverifikasi oleh model login. Model Login akan memberikan hasil verifikasi jika login sukses ataupun gagal. Jika *username* dan *password* yang di masukan adalah data yang benar maka admin akan berhasil melakukan login dan dapat melakukan pengolahaan data pada pada halaman *backend*.

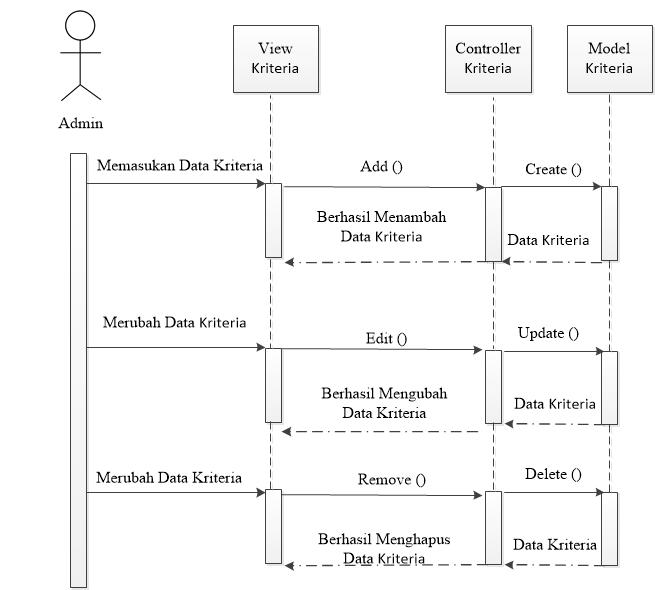
1. *Sequence Diagram* Mengelola Data Nilai



***Gambar 3. 18***Sequence Diagram *Mengelola Data Nilai*

Dari gambar diatas dapat dilihat bahwa admin memasukkan data kriteria ke View Nilai, kemudian menjalankan fungsi add, dan menjalankan fungsi create pada model nilai. Merubah data kriteria pada View Nilai, kemudian menjalankan fungsi edit dan menjalankan fungsi update pada model nilai. Merubah data nilai pada View Nilai, kemudian menjalankan fungsi remove pada model nilai.

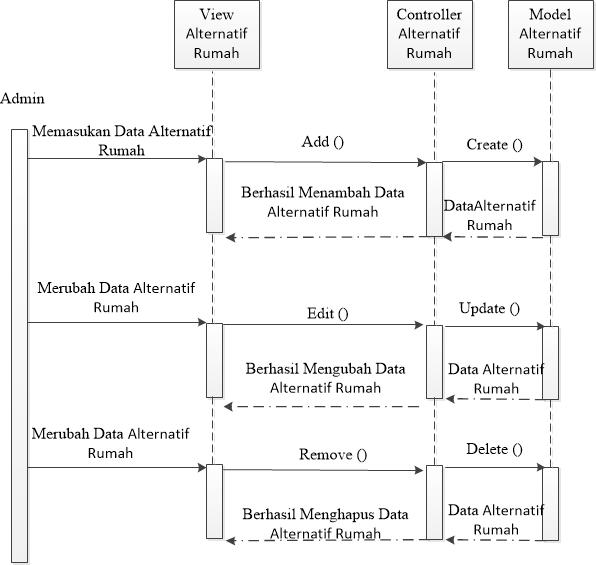
1. *Sequence Diagram* Mengelola Data Kriteria



**Gambar 3. 19** Sequence Diagram Mengelola Data Kriteria

Dari gambar diatas dapat dilihat bahwa admin memasukkan data kriteria ke View Kriteria, kemudian menjalankan fungsi add, dan menjalankan fungsi create pada model kriteria. Merubah data kriteria pada View Kriteria, kemudian menjalankan fungsi edit dan menjalankan fungsi update pada model kriteria. Merubah data kriteria pada View Kriteria, kemudian menjalankan fungsi remove pada model kriteria.

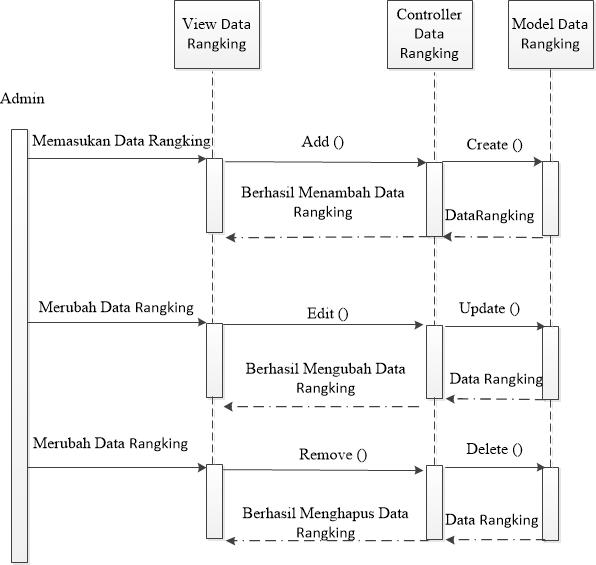
1. *Sequence Diagram* Mengelola Data Alternatif Rumah



**Gambar 3. 20** Sequence Diagram Mengelola Data Kriteria

Dari gambar diatas dapat dilihat bahwa admin memasukkan data rumah ke *View* Alternatif Rumah, kemudian menjalankan fungsi add, dan menjalankan fungsi create pada model Alternatif rumah. Merubah data rumah pada *View* Alternatif Rumah, kemudian menjalankan fungsi *edit* dan menjalankan fungsi update pada model Alternatif Rumah. Merubah data Alternatif Rumah pada *View* Alternatif Rumah, kemudian menjalankan fungsi remove pada model Alternatif Rumah.

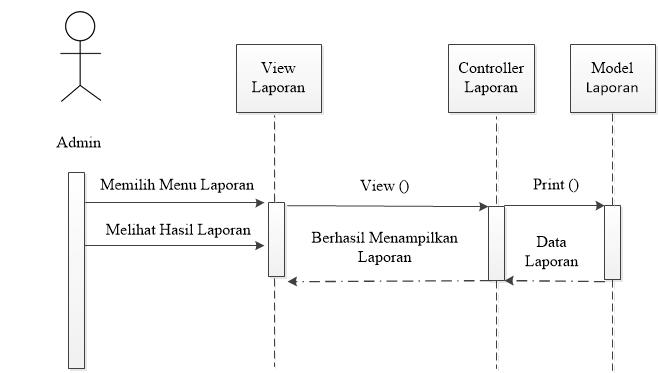
1. *Sequence Diagram* Mengelola Data Proses Perangkingan



**Gambar 3. 21** Sequence Diagram Mengelola Data Proses Perangkingan

Dari gambar diatas dapat dilihat bahwa admin memasukkan data rangking ke *View* Data Rangking, kemudian menjalankan fungsi add, dan menjalankan fungsi create pada model Data Rangking. Merubah data rumah pada *View* Data Rangking, kemudian menjalankan fungsi *edit* dan menjalankan fungsi update pada model Data Rangking. Merubah Data Rangking pada *View* Data Rangking, kemudian menjalankan fungsi remove pada model Data Rangking.

1. *Sequence Diagram* Mencetak Laporan

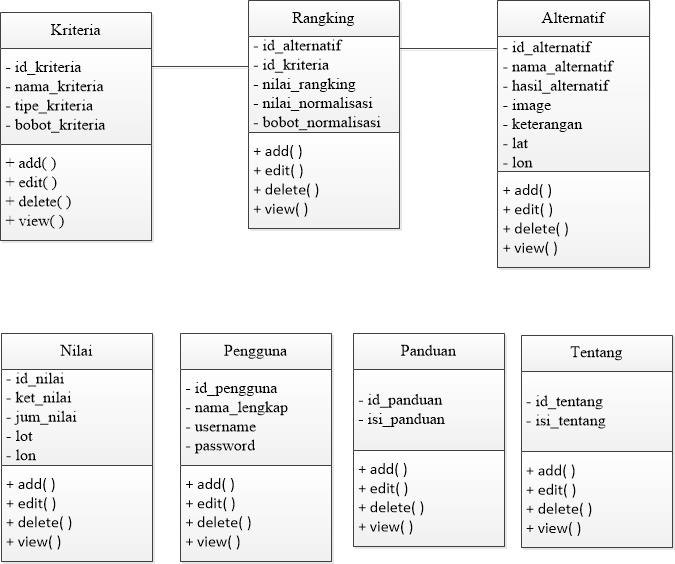


**Gambar 3. 22** Sequence Diagram Cetak Laporan

Dari gambar diatas dapat dilihat bahwa admin memilih menu laporan proses perangkingan ke *View* Data Laporan, kemudian menjalankan fungsi View, dan menjalankan fungsi *print* pada model laporan.

### *Class Diagram*

*Class Diagram* ini menggambarkan struktur dan deskripsi *class, package,* dan objek serta hubungan satu sama lain seperti *inheritance*, *association* dan lain – lain. Class memiliki tiga area pokok yaitu nama kelas, atribut dan metode.



**Gambar 3. 23** Class Diagram

Dari gambar *class diagram* diatas bahwa terdapat enam *class* yang terdiri dari *class* kriteria, *class* rangking, *class* alternatif, *class* nilai, *class* pengguna, dan *class* tentang. Dan setiap *class* memiliki uniknya masing – masing

### Rancangan *Database*

Perancangan *Database* merupakan suatu proses yang di lakukan untuk tujuan menentukan isi dan pengaturan data yang dibutuhkan dalam perancangan sistem yang akan dibangun. Rancangan *database* pada Sistem Pendukung Keputusan Untuk Rekomendasi Pembelian Rumah berbasis web di kota bekasi ini terdiri dari enam tabel.

Data-data yang diperlukan dalam sistem pendukung keputusan ini dapat disajikan pada tabel sebagai berikut:

* + 1. Tabel Alternatif

Tabel alternatif digunakan untuk menyimpan data alternatif rumah, struktur tabel alternatif seperti yang disajikan pada Tabel 3.1 di bawah ini :

**Tabel 3. 7** Rancangan Alternatif

|  |  |  |  |
| --- | --- | --- | --- |
| Nama Field | Tipe Data | Ukuran | Keterangan |
| id\_alternatif | Int | 11 | Primary Key |
| Nama\_alternatif | Varchar | 25 |  |
| Hasil Alternatif | Double |  |  |
| Image | Text |  |  |
| Keterangan | Text |  |  |
| Lat | Double |  |  |
| Ion | Double |  |  |

* + 1. Tabel Kriteria

Tabel Kriteria digunakan untuk menyimpan data kriteria yang menjadi pertimbangan dalam menentukan bobot penilian dalam proses, struktur tabel kriteria seperti yang disajikan pada Tabel 3.2 di bawah ini :

**Tabel 3. 8** Rancangan Kriteria

|  |  |  |  |
| --- | --- | --- | --- |
| Nama Field | Tipe Data | Ukuran | Keterangan |
| id\_kriteria | Int | 11 | Primary Key |
| nama\_kriteria | Varchar | 25 |  |
| tipe\_kriteria | Varchar | 10 |  |
| bobot\_kriteria | Double |  |  |

* + 1. Tabel Nilai

Tabel Nilai digunakan untuk menyimpan data penilian yang akan menjadi pertimbangan dalam proses perhitungan, struktur tabel nilai seperti yang disajikan pada Tabel 3.3 di bawah ini :

**Tabel 3. 9** Rancangan Nilai

|  |  |  |  |
| --- | --- | --- | --- |
| Nama Field | Tipe Data | Ukuran | Keterangan |
| id\_nilai | Int | 6 | Primary Key |
| ket\_nilai | Varchar | 20 |  |
| jum\_nilai | Double |  |  |

* + 1. Tabel Penguna

Tabel Penguna digunakan untuk menyimpan data penguna yang akan menjadi syarat dalam pengelolaan halaman admin pada sistem pendukung keputusan untuk rekomendasi pembelian rumah, struktur tabel pengguna seperti yang disajikan pada Tabel 3.4 di bawah ini :

**Tabel 3. 10** Rancangan Pengguna

|  |  |  |  |
| --- | --- | --- | --- |
| Nama Field | Tipe Data | Ukuran | Keterangan |
| id\_pengguna | Int | 7 | Primary Key |
| nama\_lengkap | Varchar | 25 |  |
| username | Varchar | 10 |  |
| Password | Varchar | 7 |  |

* + 1. Tebel Pengunjung

Tabel Pengunjung digunakan untuk menyimpan data pengunjung atau calon pembeli yang akan meminta rekomendasi alternatif rumah, struktur tabel pengunjung seperti yang disajikan pada Tabel 3.5 di bawah ini :

**Tabel 3. 11** Rancangan Pengunjung

|  |  |  |  |
| --- | --- | --- | --- |
| Nama Field | Tipe Data | Ukuran | Keterangan |
| id\_pengunjung | Int | 7 | Primary Key |
| nama | Varchar | 50 |  |
| jk | Varchar | 10 |  |
| umur | Varchar | 10 |  |

* + 1. Tabel Rangking

Tabel Ranking digunakan untuk melakukan perhitungan alternatif rumah, struktur tabel rangking seperti yang disajikan pada Tabel 3.6 di bawah ini :

**Tabel 3. 12** Rancangan Rangking

|  |  |  |  |
| --- | --- | --- | --- |
| Nama Field | Tipe Data | Ukuran | Keterangan |
| id\_alternatif | Int | 11 | Primary Key |
| Id\_kriteria | Int | 11 | Primary Key |
| nilai\_rangking | Double |  |  |
| nilai\_normalisasi | Double |  |  |
| bobot\_normalisasi | Double |  |  |

* + 1. Tabel Tentang

Tabel Tentang digunakan untuk untuk menyimpan isi panduan sistem pendukung keputusan, struktur tabel tentang dapat di lihat dalam Tabel 3.9 di bawah ini ::

**Tabel 3. 13** Rancangan Tentang

|  |  |  |  |
| --- | --- | --- | --- |
| Nama Field | Tipe Data | Ukuran | Keterangan |
| id\_tentang | Int | 11 | Primary Key |
| Isi\_tentang | Text |  |  |

* + 1. Tabel Panduan

Tabel Panduan digunakan untuk menyimpan isi panduan sistem pendukung keputusan , struktur tabel panduan dapat di lihat dalam Tabel 3.9 di bawah ini :

**Tabel 3. 14** Rancangan Panduan

|  |  |  |  |
| --- | --- | --- | --- |
| Nama Field | Tipe Data | Ukuran | Keterangan |
| id\_panduan | Int | 11 | Primary Key |
| isi\_panduan | Text |  |  |

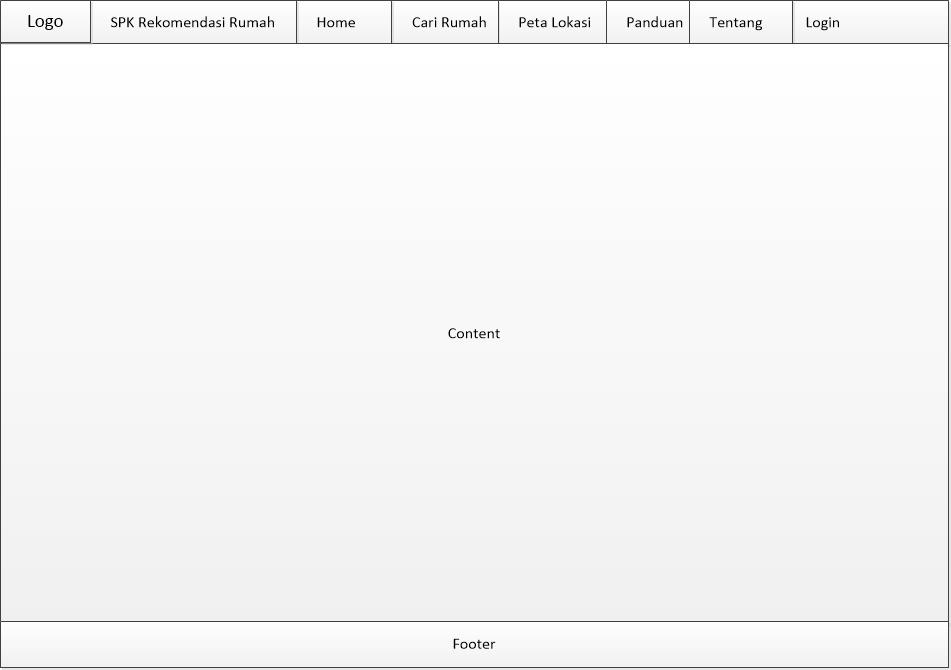
### Rancangan *Interface*

Rancangan Interface ini terdiri dari 2 bagian yaitu bagian *frontend* dan *backend*. Pada bagian *frontend* pengunjung atau calon pembeli dapat mengakses home, panduan, about, analisis, dan peta lokasi. Pada bagian *backend* terdiri dari home, nilai, kriteria, alternatif, proses perangkingan, laporan dan menu logout.

1. Rancangan Halaman Utama *Frontend*

Halaman *frontend* atau halaman bagian depan sebuah website di bangun atau di sediakan untuk para pengunjung website, untuk tampilan halaman depan sistem pendukung keputusan yang akan dibangun seperti pada rancangan dibawah ini :

* + - 1. Halaman Utama *Frontend*

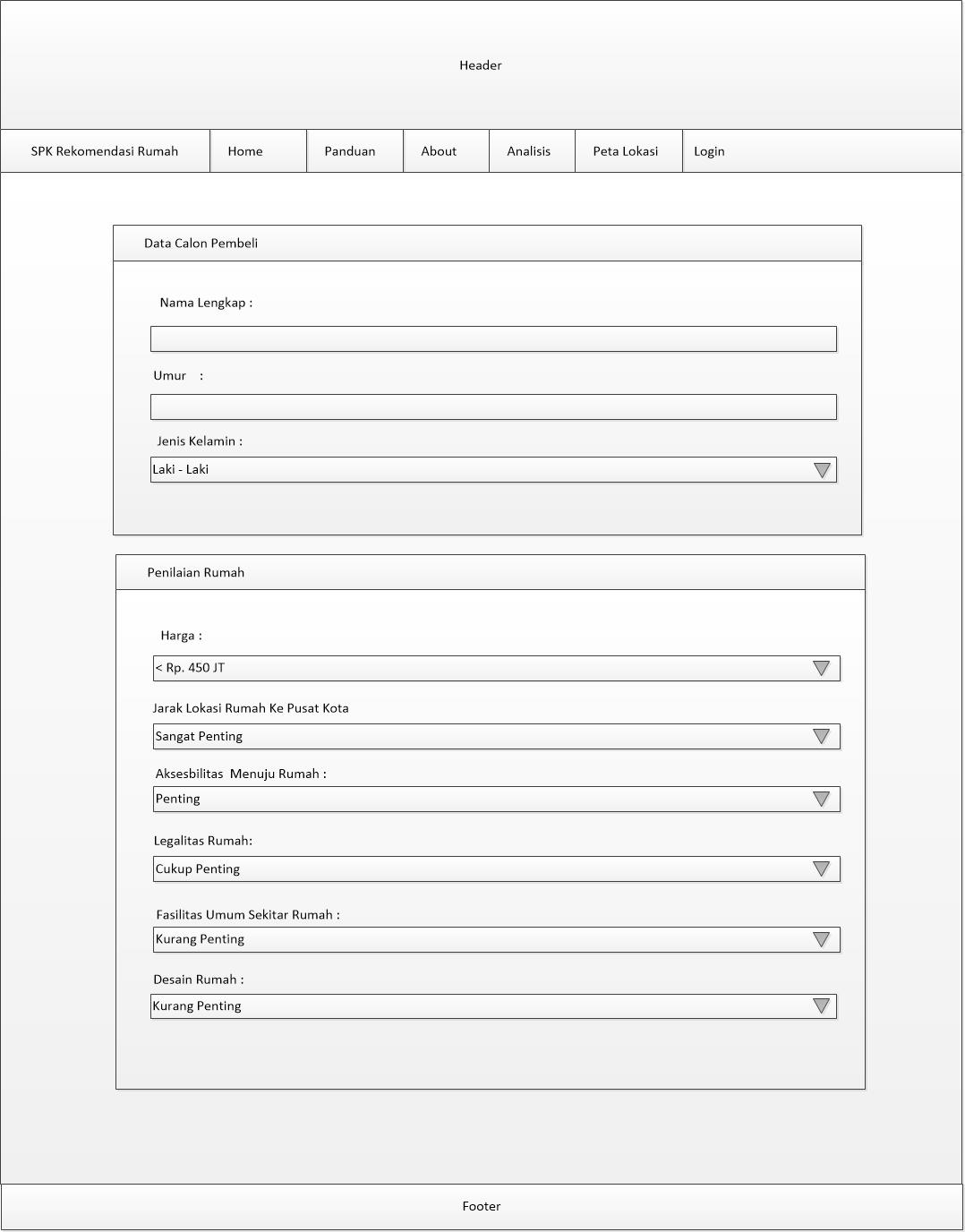


**Gambar 3. 24** Rancangan interface halaman utama frontend

Rancangan halaman depan website yang akan dibangun seperti terlihat pada gambar diatas. Pada halaman depan tersebut terdapat menu pengunjung atau calon pembeli dapat mengakses menu home, cari rumah, peta lokasi, panduan, tentang dan menu login untuk admin. Pada bagian tengah akan di isi oleh konten berdasarkan menu yang dipilih oleh calon pembeli, untuk halaman awal atau menu home akan berisi halaman selamat datang atau sekilas tentang sistem pendukung keputusan untuk rekomendasi pembelian rumah di kota bekasi.

Dalam Halaman *Frontend* ada beberapa menu utama yang disediakan oleh sistem pendukung keputusan, Untuk setiap menu mempunyai fungsi masing-masing dan untuk rancangan setiap menu halaman *frontend* akan di jelaskan di bawah ini :

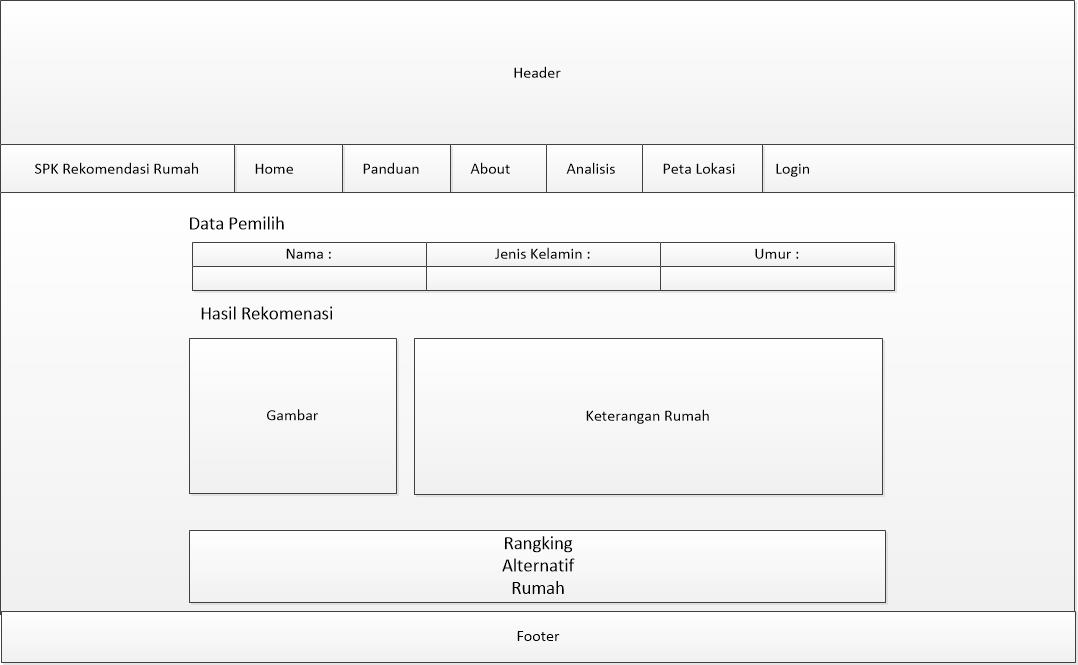
* 1. Rancangan Menu Cari Rumah



**Gambar 3. 25** Rancangan Menu Cari Rumah

Menu cari rumah dapat di akses oleh calon pembeli untuk melakukan proses pencarian alternatif rumah dengan cara memasukan data calon pembeli dan penilaian rumah oleh calon pembeli sesuai kebutuhannya.

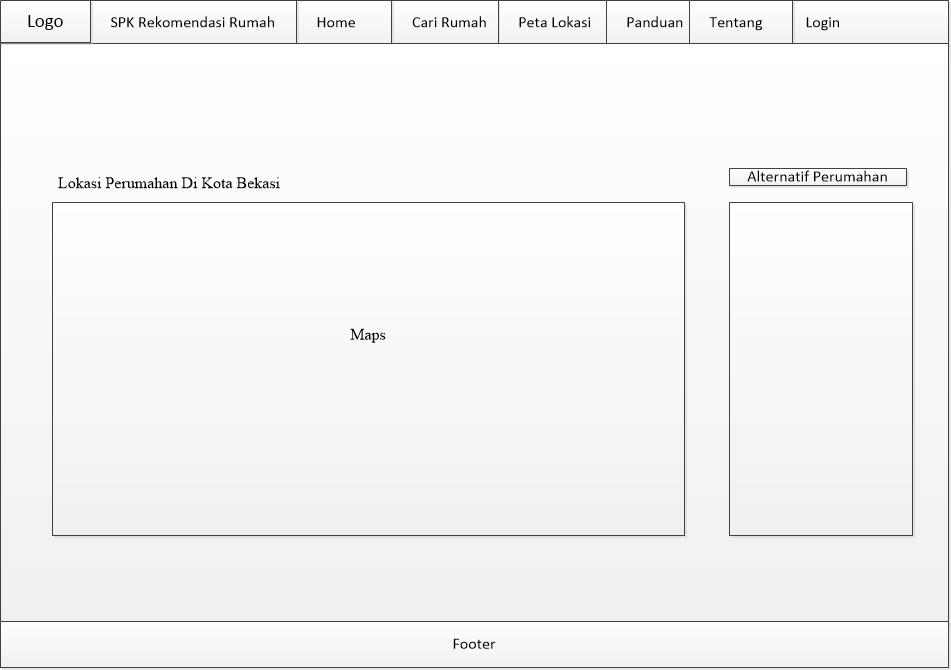
* 1. Rancangan Hasil Rekomendasi Rumah



**Gambar 3. 26** Rancangan Hasil Rekomendasi Rumah

Hasil rekomendasi rumah yang dimunculkan merupakan hasil dari inputan yang sudah di proses oleh sistem.

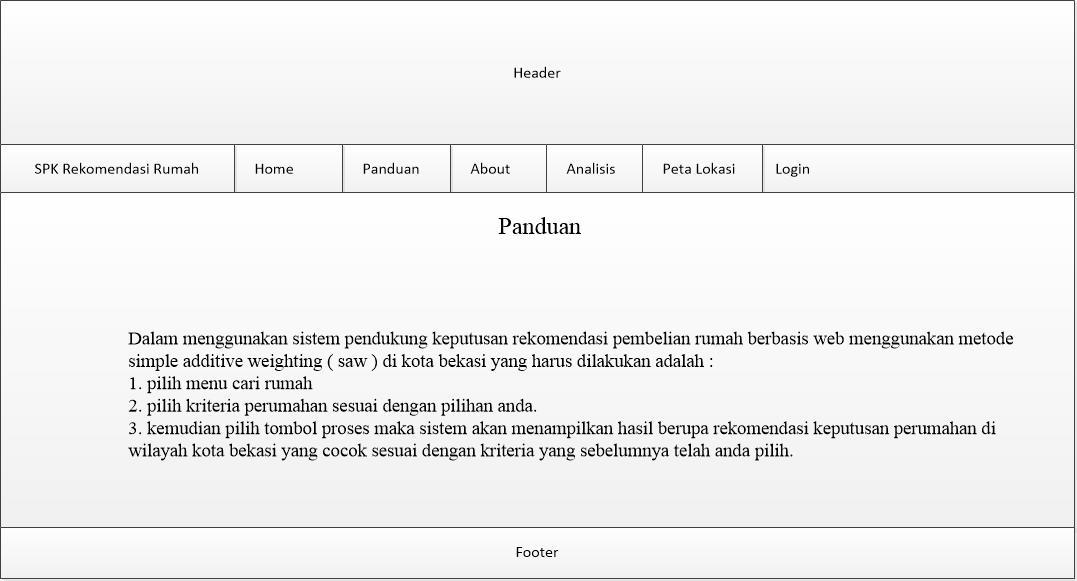
* 1. Rancangan Menu Peta Lokasi



**Gambar 3. 27** Rancangan Menu Peta Lokasi

Pada menu Peta Lokasi calon pembeli dapat melihat data – data perumahan yang ditawarkan oleh sistem pendukung keputusan.

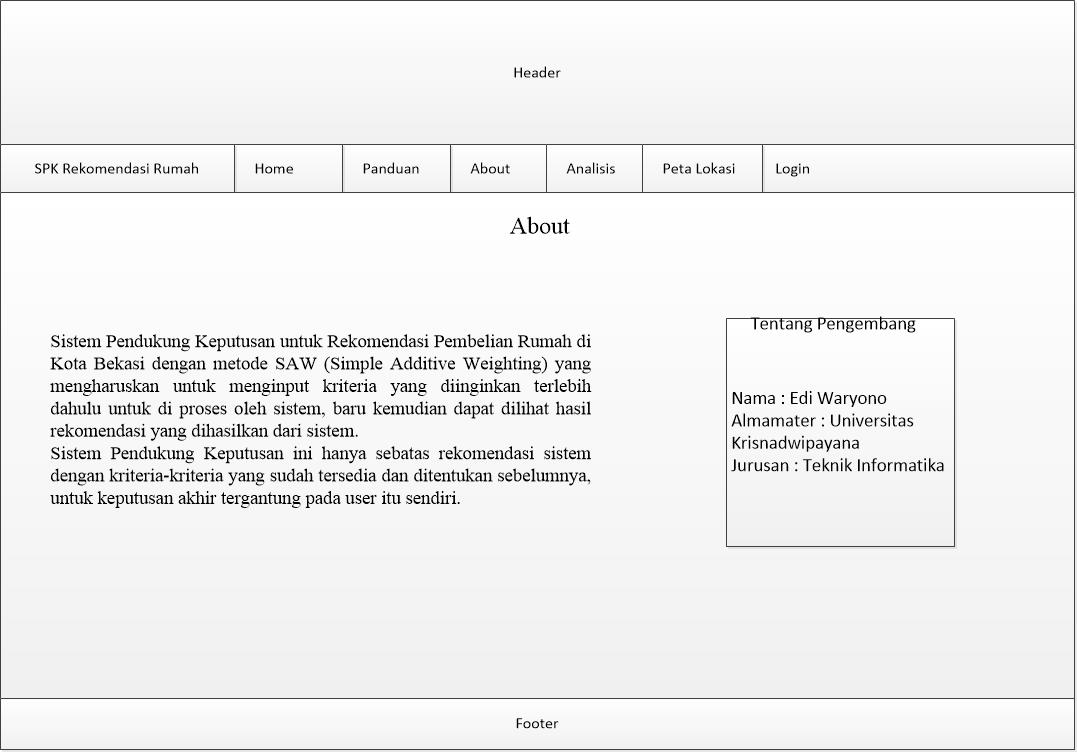
* 1. Rancangan Menu Panduan



**Gambar 3. 28** Rancangan Menu Panduan

Pada menu Panduan calon pembeli dapat melihat panduan penggunaan sistem, agar dalam mengunakan sistem ini dengan memahami fungsi pembuatannya.

* 1. Rancangan Menu Tentang



**Gambar 3. 29** Rancangan Menu Tentang

Pada menu Peta Tentang calon pembeli dapat melihat tentang sistem pendukung keputusan, dengan tujuan agar lebih mengenal terhadap sistem yang di bangun.

* 1. Rancangan Menu *Login*



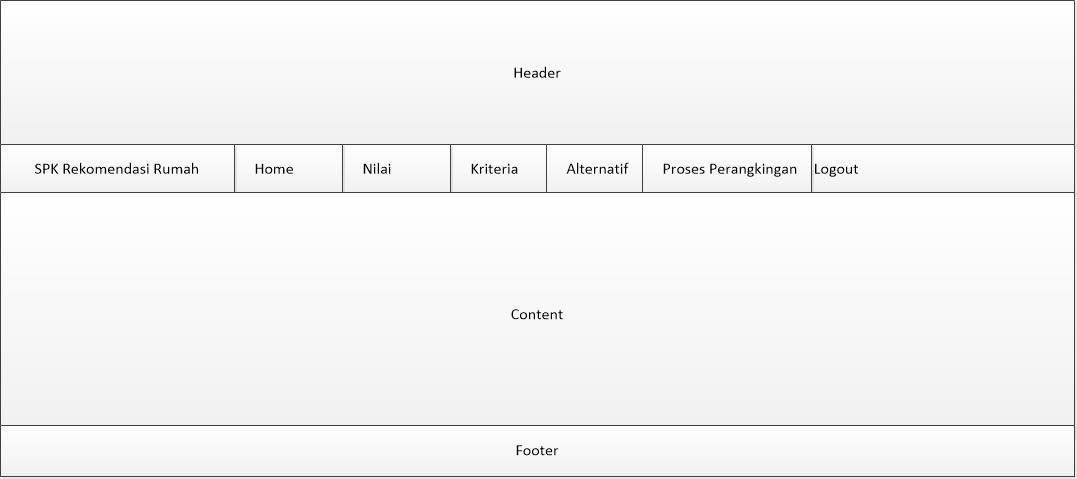
**Gambar 3. 30** Rancangan Menu Login

Halaman Menu Login digunakan oleh admin untuk masuk ke halaman *backend,* halaman ini mengharuskan admin memasukan username dan password dengan benar.

1. Rancangan Halaman *Backend*

Halaman *backend* adalah sebuah halaman belakang yang mempunyai tampilan khusus yang dibuat dengan tujuan untuk *me-manage* sebuah *website*. Halaman *backend* dapat dikendalikan oleh seorang admin dalam menambah, menghapus, dan mengedit data yang dikelola di dalam sistem pendukung keputusan.

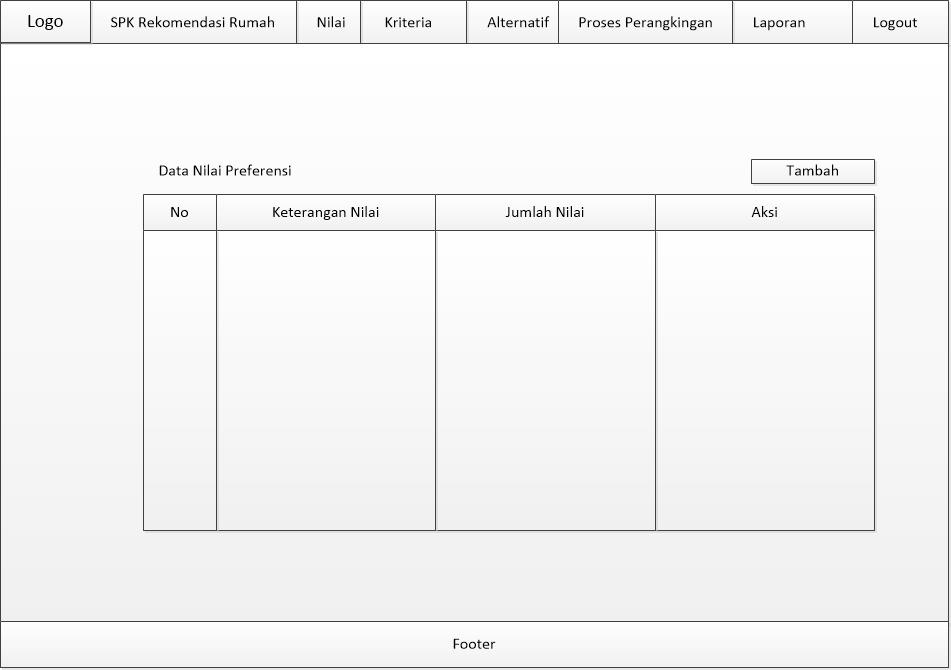
* + - 1. Halaman Utama *backend*



**Gambar 3. 31** Rancangan Halaman Utama backend

Rancangan halaman admin yang akan dibangun seperti terlihat pada gambar diatas. Pada halaman admin ini terdapat menu home, nilai, kriteria, alternatif, proses perangkingan dan logout.

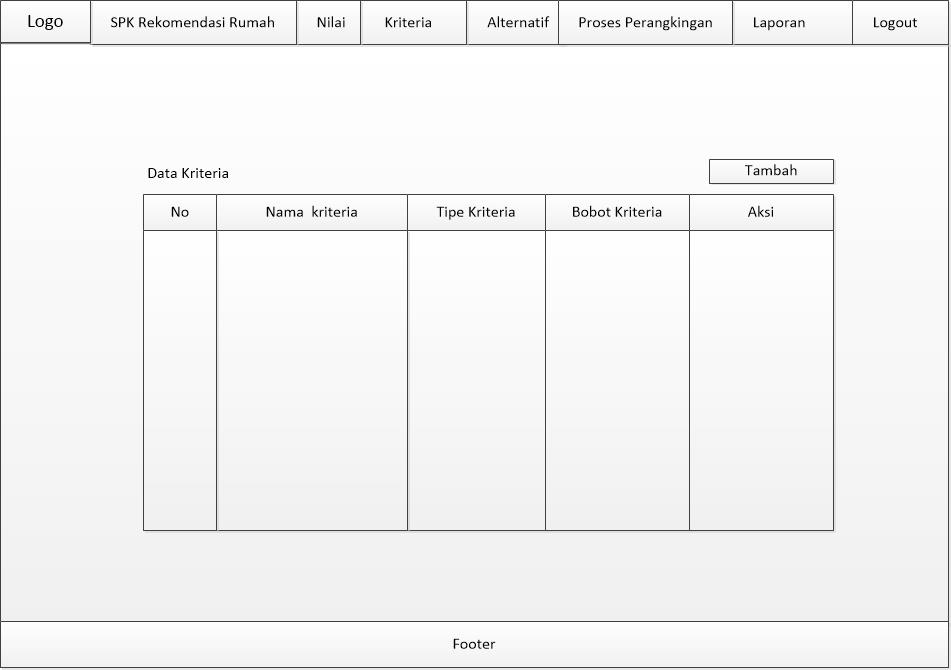
* + - 1. Rancangan Menu Nilai



**Gambar 3. 32** Rancangan Menu Nilai

Menu Nilai admin dapat mengelola data nilai dan jumlah nilai, yang disediakan untuk calon pembeli melakukan penilain terhadap alternatif yang di tawarkan.

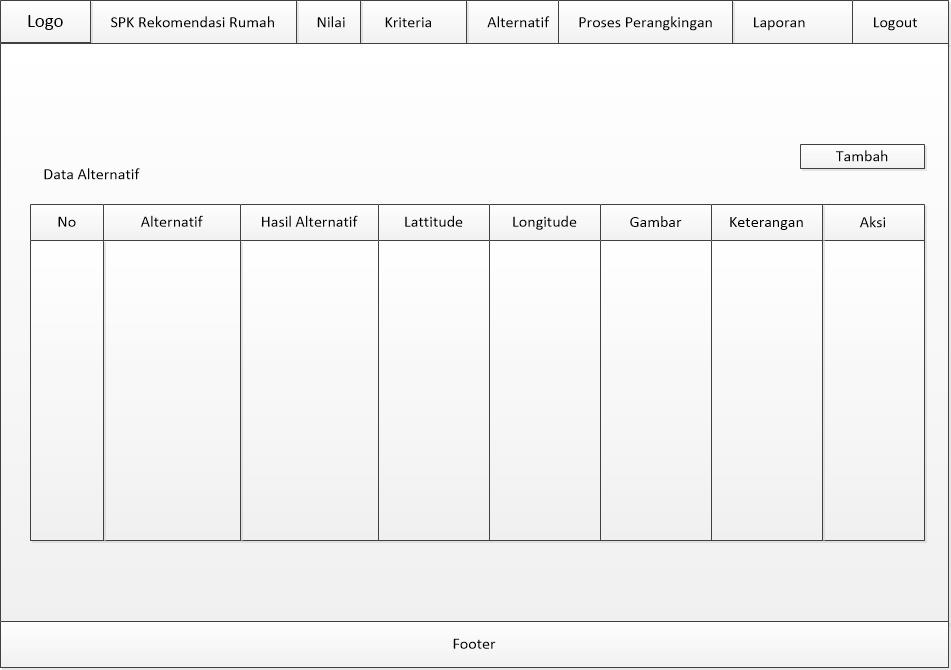
* + - 1. Rancangan Menu Kriteria



**Gambar 3. 33** Rancangan Menu Kriteria

Menu Kriteria dapat mengelola data kriteria yang disediakan untuk calon pembeli melakukan pilihan rumah berdasarkan alternatif yang di tawarkan.

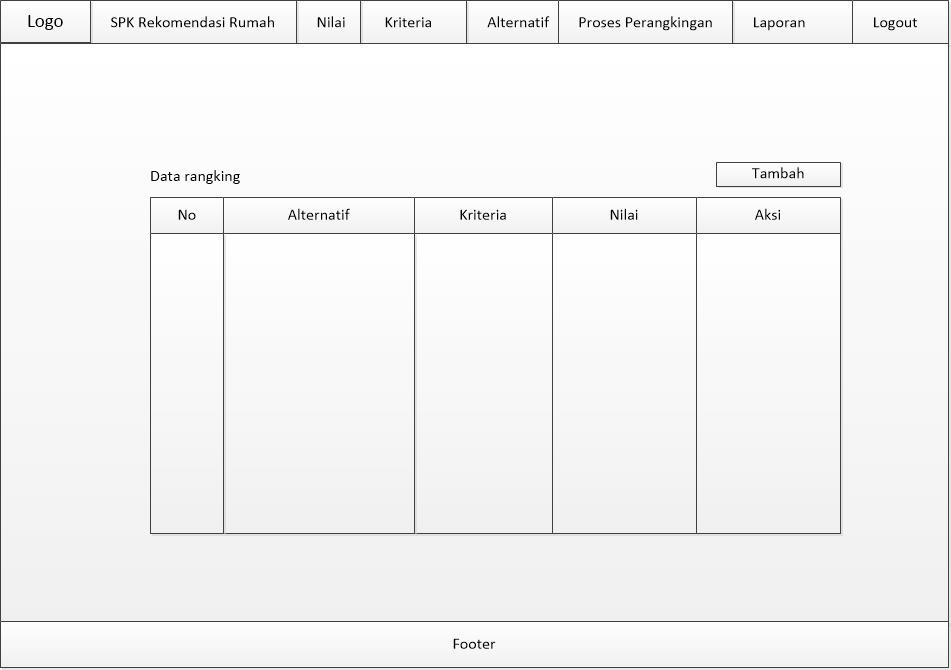
* + - 1. Rancangan Menu Alternatif



**Gambar 3. 34** Rancangan Menu Alternatif

Menu Alternatif dapat mengelola data alternatif yang disediakan untuk calon pembeli melakukan pilihan rumah berdasarkan alternatif yang di tawarkan.

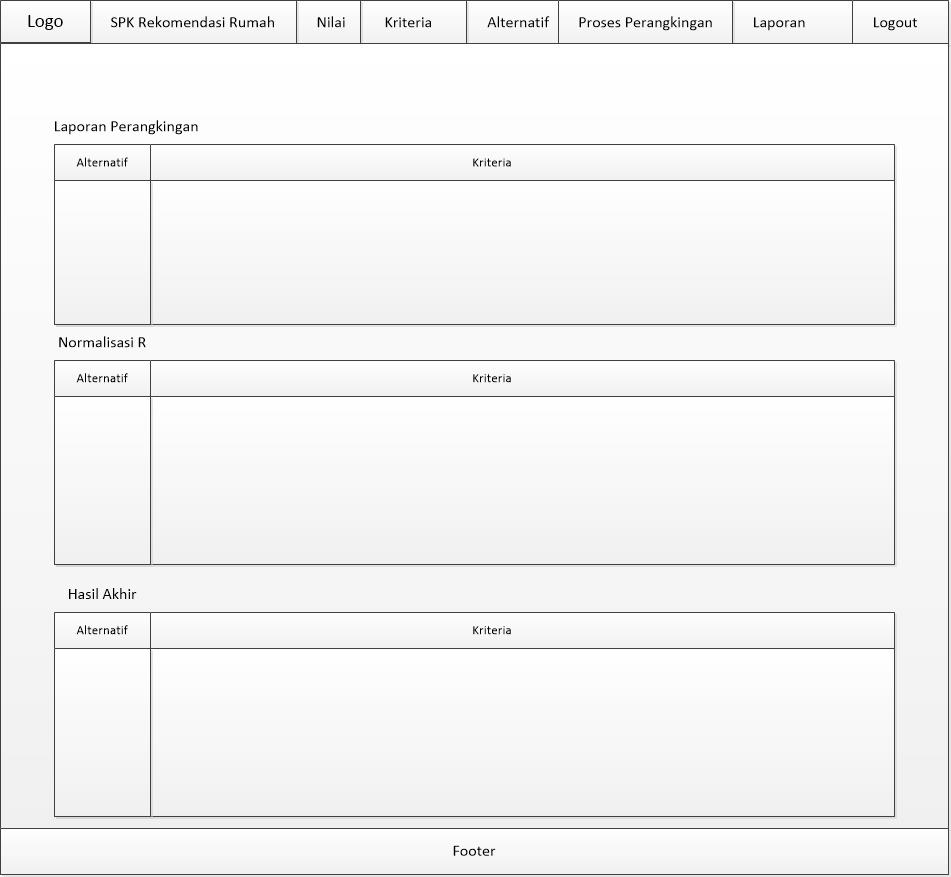
* 1. Rancangan Menu Perangkingan



**Gambar 3. 35** Rancangan Menu Perangkingan

Menu Perangkingan dapat mengelola data rangking yang disediakan untuk calon pembeli dalam merangking sesuai kebututuhan.

* 1. Rancangan Menu Laporan



**Gambar 3. 36** Rancangan Menu Laporan

Menu Laporan pada halaman admin dapat melihat laporan perangkingan dan melihat hasil normalisasi setiap kriteria dan melihat hasil akhir perhitungan dari proses perhitungan menggunakan metode *simple additive weighting*,

# BAB IV

**IMPLEMENTASI DAN PENGUJIAN SISTEM**

Untuk mengetahui keberhasilan dari program yang telah dirancang, maka perlu dilakukan pengujian terhadap aplikasi ini. Hasil dari implementasi yang telah di lakukan sebagai berikut :

## Implementasi

### Implementasi Database

* + 1. Tabel Alternatif

Tabel alternatif digunakan untuk menyimpan data alternatif rumah – rumah yang akan di tampilkan pada sistem pendukung keputusan, struktur tabel alternatif seperti yang disajikan pada Tabel 4.1 di bawah ini :

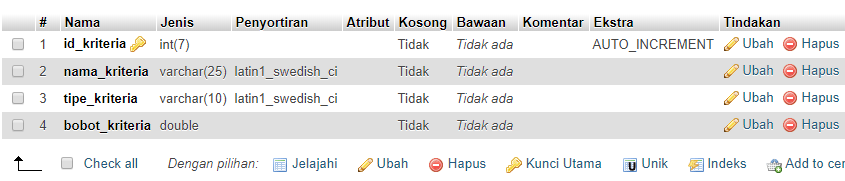
**Tabel 4. 1** Implementasi Alternatif



* + 1. Tabel Kriteria

Tabel Kriteria digunakan untuk menyimpan data kriteria – kriteria yang menjadi pertimbangan dalam menentukan bobot penilian dalam proses perhitungan pada sistem pendukung keputusan, struktur tabel kriteria seperti yang disajikan pada Tabel 4.2 di bawah ini :

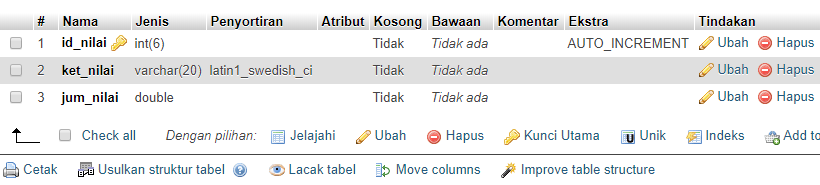
**Tabel 4. 2** Implementasi Kriteria



* + 1. Tabel Nilai

Tabel Nilai digunakan untuk menyimpan data penilian yang akan menjadi pertimbangan dalam proses perhitungan pada sistem pendukung keputusan, struktur tabel nilai seperti yang disajikan pada Tabel 4.3 di bawah ini :

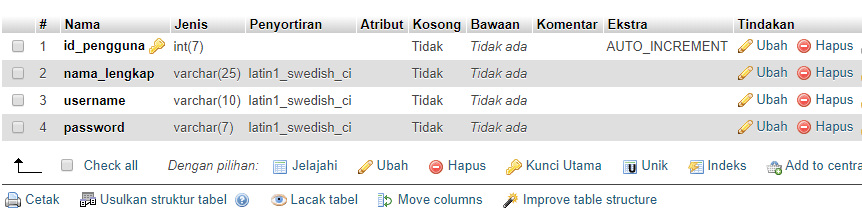
**Tabel 4. 3** Implementasi Nilai



* + 1. Tabel Penguna

Tabel Penguna digunakan untuk menyimpan data penguna yang akan menjadi syarat dalam pengelolaan halaman admin pada sistem pendukung keputusan, struktur tabel pengguna seperti yang disajikan pada Tabel 4.4 di bawah ini :

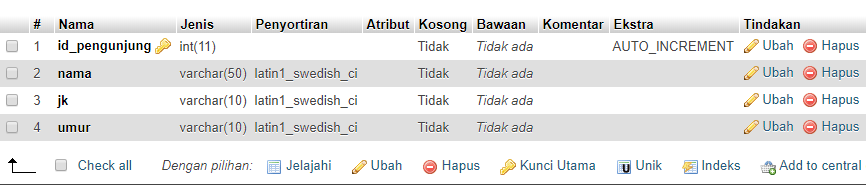
**Tabel 4. 4** Implementasi Penguna



* + 1. Tebel Pengunjung

Tabel Pengunjung digunakan untuk menyimpan data pengunjung atau calon pembeli yang akan meminta rekomendasi alternatif rumah, struktur tabel pengunjung seperti yang disajikan pada Tabel 4.5 di bawah ini :

**Tabel 4. 5** Implementasi Pengunjung



* + 1. Tabel Rangking

Tabel Ranking digunakan untuk melakukan perhitungan alternatif rumah, struktur tabel rangking seperti yang disajikan pada Tabel 4.6 di bawah ini :

**Tabel 4. 6** Implementasi Rangking



* + 1. Tabel Tentang

Tabel Tentang digunakan untuk untuk menyimpan isi panduan sistem pendukung keputusan, struktur tabel tentang dapat di lihat dalam Tabel 3.9 di bawah ini ::

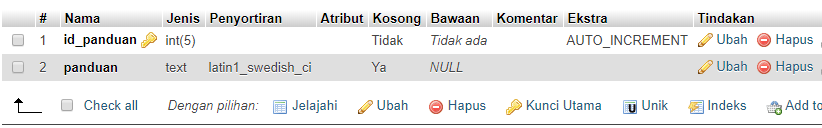
**Tabel 4. 7** Implementasi Tentang



* + 1. Tabel Panduan

Tabel Panduan digunakan untuk menyimpan isi panduan sistem pendukung keputusan , struktur tabel panduan dapat di lihat dalam Tabel 3.9 di bawah ini :

**Tabel 4. 8** Implementasi Panduan



### Implementasi *Interface*

Pada implementasi Interface ini terdiri dari 2 bagian yaitu bagian *frontend* dan *backend*. Pada bagian *frontend* pengunjung atau calon pembeli dapat mengakses menu home, panduan, about, analisis, dan peta lokasi. Pada bagian *backend* terdiri dari home, nilai, kriteria, alternatif, proses perangkingan, laporan dan menu logout.

* 1. Implementasi Halaman Utama *frontend*

Halaman Utama merupakan halaman bagian depan sebuah website dibangun yang disediakan untuk para pengunjung website, untuk tampilan halaman utama sistem pendukung keputusan yang telah dibangun seperti pada tampilan seperti dibawah ini :



**Gambar 4. 1** Implementasi interface halaman utama frontend

Halaman depan website yang telah dibangun seperti terlihat pada gambar diatas. Pada halaman depan tersebut terdapat menu pengunjung atau calon pembeli dapat mengakses menu home, cari rumah, peta lokasi, panduan, tentang dan menu login. Dibawah ini adalah pengkodingan untuk interface *halaman utama* frontend :

<?php

include\_once 'includes/config.php';

$config = new Config();

$db = $config->getConnection();

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<!-- The above 3 meta tags \*must\* come first in the head; any other head content must come \*after\* these tags -->

<title>Sistem Pendukung Keputusan Pemilihan Perumahan : Metode SAW</title>

<!-- Bootstrap -->

<link href="css/bootstrap.min.css" rel="stylesheet">

<style type="text/css">

.jumbotron {

position: relative;

background: url(images/background.jpg) center center;

color:#fff;

font :bold;

width: 100%;

height: 100%;

background-size: cover;

overflow: hidden;

}

</style>

<script src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script>

<script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>

<![endif]-->

</head>

<body>

<nav class="navbar navbar-inverse navbar-fixed-top navbar-expand-lg navbar navbar-light style="background-color: #187bb5";>

<div class="container">

<!-- Brand and toggle get grouped for better mobile display -->

<div class="navbar-header">

<button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#bs-example-navbar-collapse-1" aria-expanded="false">

<span class="sr-only">Toggle navigation</span>

<span class="icon-bar"></span>

<span class="icon-bar"></span>

<span class="icon-bar"></span>

</button>

<img src ="images/lg.png" height="50"width="50">

</div>

<div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">

<ul class="nav navbar-nav">

<a class="navbar-brand" >&nbsp;&nbsp;SPK Perumahan&nbsp;&nbsp;</a>

<li><a href="home.php">Home</a></li>

<li><a href="carirumah.php">Cari Rumah</a></li>

<li><a href="map.php">Peta Lokasi</a></li>

<li><a href="panduan.php">Panduan</a></li>

<li><a href="about.php">Tentang</a></li>

<li><a href="login.php">Login</a></li>

</ul>

</div><!-- /.navbar-collapse -->

</div><!-- /.container-fluid -->

</nav>

<div class="jumbotron">

<div class="container text-center">

<img src ="images/logo.png" width="500">

<p><font size="90" color="#187bb5"><b>SPK untuk Rekomendasi Pembelian Rumah</b></font></p><br>

<p><font size="5" color="#187bb5"><b>Merekomendasikan Rumah Yang Cocok Dengan Kebutuhan pembeli Berbasis Web<br>Menggunakan Metode Simple Additive Weighting ( SAW ) Di Kota Bekasi</font></p><br>

</div>

</div>

<footer>

<nav class="navbar navbar-inverse navbar-expand-lg navbar navbar-light style="background-color: #187bb5";

<div class="container">

<div class="col-lg-12 text-center">

<br><p>SPK Untuk Rekomendasi Pembelian Rumah &copy; 2019</p>

</div>

</div>

</footer>

<!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->

<script src="js/jquery-1.11.3.min.js"></script>

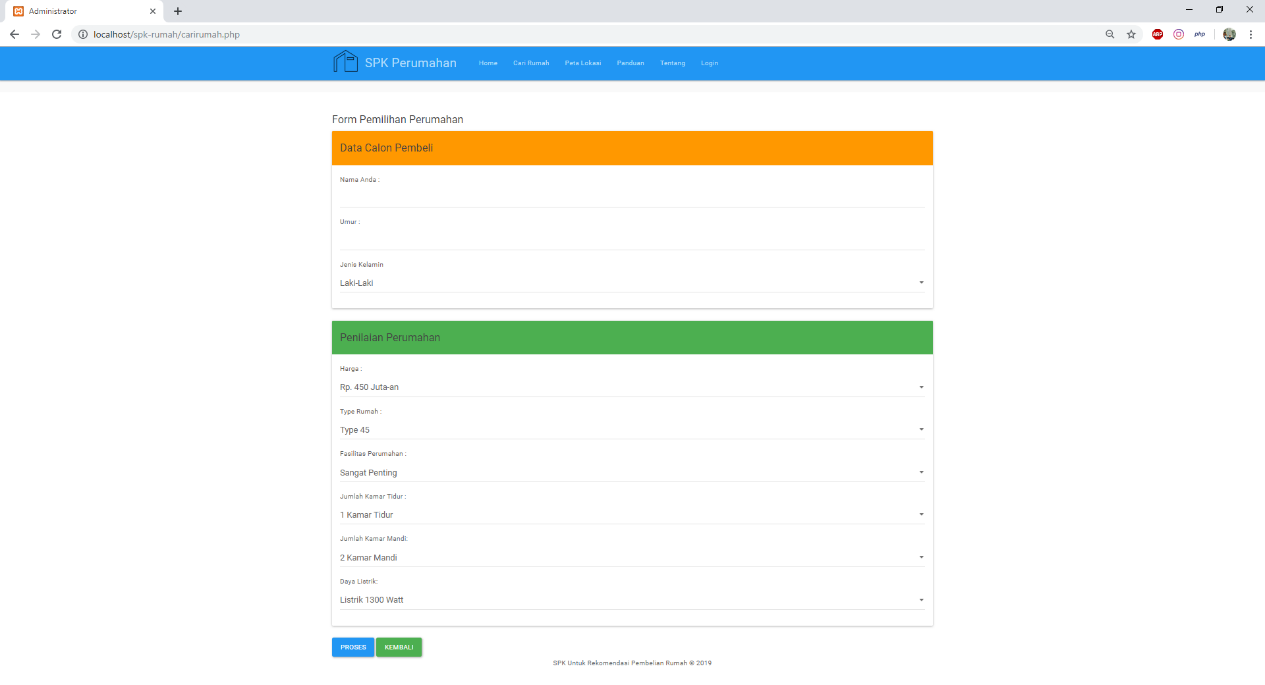
<!-- Include all compiled plugins (below), or include individual files as needed -->

<script src="js/bootstrap.min.js"></script>

</body>

</html>

* 1. Menu Cari Rumah



**Gambar 4. 2** Implementasi Menu Cari Rumah

Menu cari rumah dapat di akses oleh calon pembeli untuk melakukan proses pencarian alternatif rumah.

//MENDAPATKAN INPUT PENILAIAN BOBOT DARI CALON PEMBELI

?>

<div class="row">

<div class="col-xs-12 col-sm-12 col-md-12">

<h5>Form Pemilihan Perumahan</h5>

<form method="post" action="simpan\_analisa.php">

<div class="panel panel-warning">

<div class="panel-heading"><h5>Data Calon Pembeli</h5></div>

<div class="panel-body">

<div class="form-group">

<label for="nama">Nama Anda :</label>

<input type="text" class="form-control" id="nama" name="nama" oninvalid="this.setCustomValidity('data yang dimasukkan tidak lengkap')" required></div>

<div class="form-group">

<label for="umur">Umur :</label>

<input type="text" class="form-control" id="umur" name="umur" oninvalid="this.setCustomValidity('data yang dimasukkan tidak lengkap')" required>

</div>

<div class="form-group">

<label for="jk">Jenis Kelamin</label>

<select class="form-control" id="jk" name="jk">

<option value='Laki-Laki'>Laki-Laki</option>

<option value='Perempuan'>Perempuan</option>

</select></div></div></div>

<div class="panel panel-success">

<div class="panel-heading"><h5>Penilaian Perumahan</h5></div>

<div class="panel-body">

<div class="form-group">

<label for="kt1">Harga :</label>

<select class="form-control" id="kt1" name="kt1">

<?php

$stmt4 = $pgn3->readHarga();

while ($row4 = $stmt4->fetch(PDO::FETCH\_ASSOC)){

extract($row4);

echo "<option value='{$jum\_nilai}'>{$ket\_nilai}</option>";

}

?>

</select>

</div>

<div class="form-group">

<label for="kt2">Luas Tanah :</label>

<select class="form-control" id="kt2" name="kt2">

<?php

$stmt4 = $pgn3->readLuasTanah();

while ($row4 = $stmt4->fetch(PDO::FETCH\_ASSOC)){

extract($row4);

echo "<option value='{$jum\_nilai}'>{$ket\_nilai}</option>";

}

?>

</select>

</div>

<div class="form-group">

<label for="kt3">Luas Bangunan :</label>

<select class="form-control" id="kt3" name="kt3">

<?php

$stmt4 = $pgn3->readLuasBangunan();

while ($row4 = $stmt4->fetch(PDO::FETCH\_ASSOC)){

extract($row4);

echo "<option value='{$jum\_nilai}'>{$ket\_nilai}</option>";

}

?>

</select>

</div>

<div class="form-group">

<label for="kt4">Jumlah Kamar Tidur :</label>

<select class="form-control" id="kt4" name="kt4">

<?php

$stmt4 = $pgn3->readKamarTidur();

while ($row4 = $stmt4->fetch(PDO::FETCH\_ASSOC)){

extract($row4);

echo "<option value='{$jum\_nilai}'>{$ket\_nilai}</option>";

}

?>

</select>

</div>

<div class="form-group">

<label for="kt5">Jumlah Kamar Mandi:</label>

<select class="form-control" id="kt5" name="kt5">

<?php

$stmt4 = $pgn3->readKamarmandi();

while ($row4 = $stmt4->fetch(PDO::FETCH\_ASSOC)){

extract($row4);

echo "<option value='{$jum\_nilai}'>{$ket\_nilai}</option>";

}

?>

</select>

</div>

<div class="form-group">

<label for="kt6">Daya Listrik:</label>

<select class="form-control" id="kt6" name="kt6">

<?php

$stmt4 = $pgn3->readListrik();

while ($row4 = $stmt4->fetch(PDO::FETCH\_ASSOC)){

extract($row4);

echo "<option value='{$jum\_nilai}'>{$ket\_nilai}</option>";

}

?>

</select>

</div>

</div>

</div>

<button type="submit" name="proses" class="btn btn-primary">Proses</button>

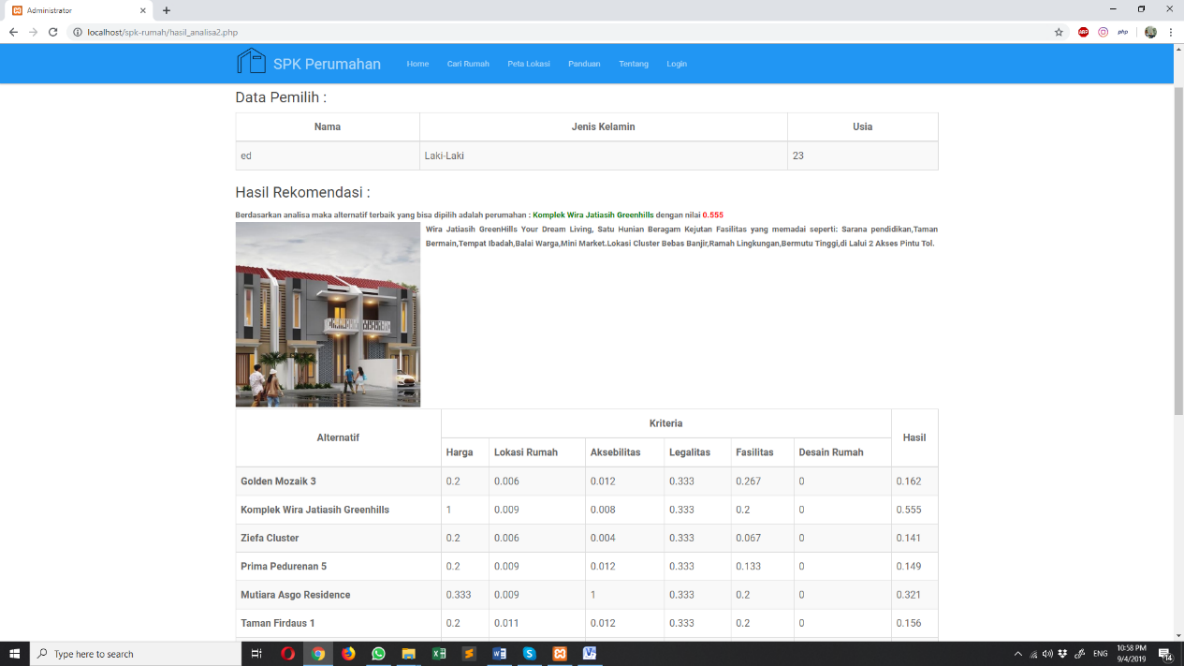
<button type="button" onclick="location.href='login.php'" class="btn btn-success">Kembali</button>

</form>

</div>

</div>

* 1. Hasil Rekomendasi Rumah



**Gambar 4. 3** Hasil Rekomendasi Rumah

Data dari calon pembeli akan diproses oleh sistem kemudian data tersebut akan dikelola dan menghasilkan rekomendasi rumah sesuai dengan data yang dimasukan oleh calon pembeli.

<h4>Hasil Rekomendasi :</h4>

<?php

sleep(5); //waktu agar query tidak langsung di proses untuk menunggu hasil perhitungan

$query1 = mysql\_query("SELECT nama\_alternatif, hasil\_alternatif, image, keterangan FROM alternatif

WHERE hasil\_alternatif = (SELECT MAX(hasil\_alternatif) FROM alternatif)") or die (mysql\_error());

while ($row1=mysql\_fetch\_array($query1)){ ?>

<strong>Berdasarkan analisa maka alternatif terbaik yang bisa dipilih adalah perumahan :

<font color="green"><?php echo $row1['nama\_alternatif'];?></font> dengan nilai

<font color="red"><?php echo round ($row1['hasil\_alternatif'],3); ?></font><br>

<p style="text-align:justify;">

<img src="upload/<?php echo $row1['image']; ?>" style="width:300px; height:300px; float:left; margin:0 9px 3px 0;">

<?php echo $row1['keterangan']; } ?>

</strong>

<br><br><table width="100%" class="table table-striped table-bordered">

<thead>

<tr>

<th rowspan="2" style="vertical-align: middle" class="text-center">Alternatif</th>

<th colspan="<?php echo $stmt2->rowCount(); ?>" class="text-center">Kriteria</th>

<th rowspan="2" style="vertical-align: middle" class="text-center">Hasil</th>

</tr>

<tr>

<?php

while ($row2 = $stmt2->fetch(PDO::FETCH\_ASSOC)){

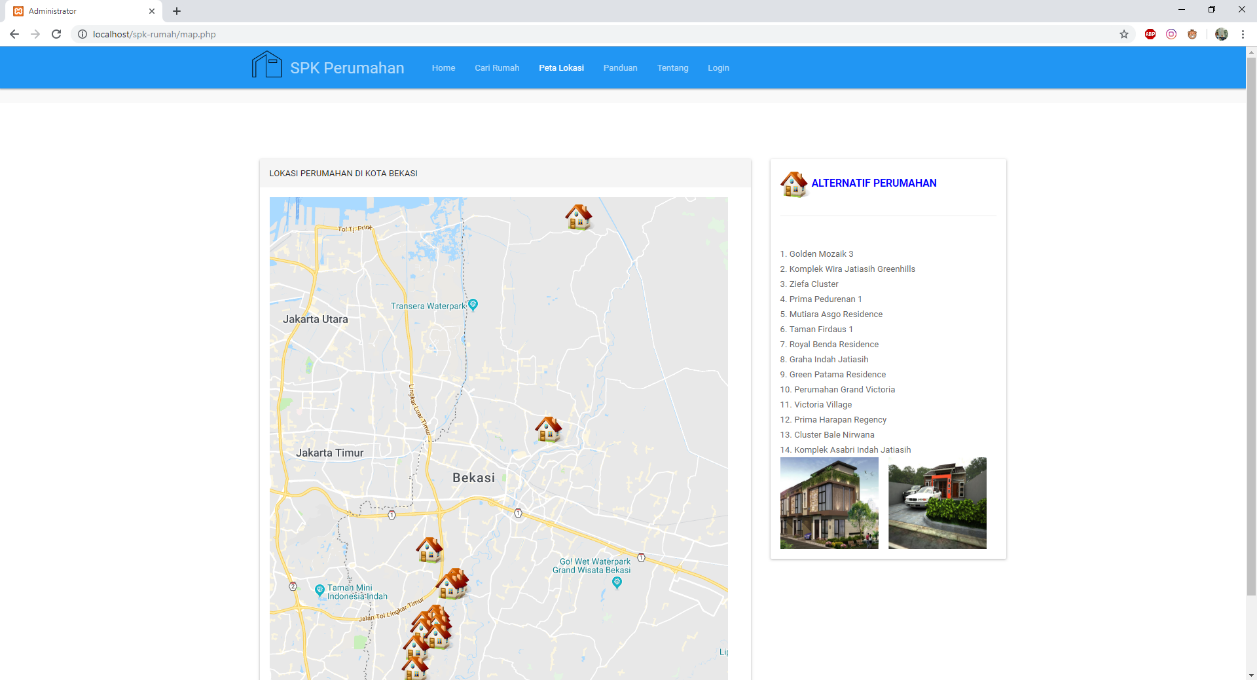
?> <th><?php echo $row2['nama\_kriteria'] ?></th>

<?php }

?> </tr>

</thead>

* 1. Menu Peta Lokasi



**Gambar 4. 4** Implementasi Menu Peta Lokasi

Pada menu Peta Lokasi calon pembeli dapat melihat data – data perumahan yang ditawarkan oleh sistem pendukung keputusan di daerah kota bekasi. Dibawah ini adalah pengkodingan untuk menu Peta Lokasi:

var marker;

function initialize() {

// Variabel untuk menyimpan informasi (desc)

var infoWindow = new google.maps.InfoWindow;

// Variabel untuk menyimpan peta Roadmap

var mapOptions = {

mapTypeId: google.maps.MapTypeId.ROADMAP, zoom: 24

}

// Pembuatan petanya

var map = new google.maps.Map(document.getElementById('map-canvas'), mapOptions);

// Variabel untuk menyimpan batas kordinat

var bounds = new google.maps.LatLngBounds();

// Pengambilan data dari database

<?php

$query = mysqli\_query($con,"select \* from alternatif");

while ($data = mysqli\_fetch\_array($query))

{

$nama = $data['nama\_alternatif'];

$lat = $data['lat'];

$lon = $data['lon'];

echo ("addMarker($lat, $lon, '<b>$nama</b>');\n");

}

?>

// Proses membuat marker

function addMarker(lat, lng, info) {

var lokasi = new google.maps.LatLng(lat, lng);

bounds.extend(lokasi);

var marker = new google.maps.Marker({

map: map,

position: lokasi,

icon: 'images/rumah.png'

});

map.fitBounds(bounds);

bindInfoWindow(marker, map, infoWindow, info);

}

// Menampilkan informasi pada masing-masing marker yang diklik

function bindInfoWindow(marker, map, infoWindow, html) {

google.maps.event.addListener(marker, 'click', function() {

infoWindow.setContent(html);

infoWindow.open(map, marker);

});

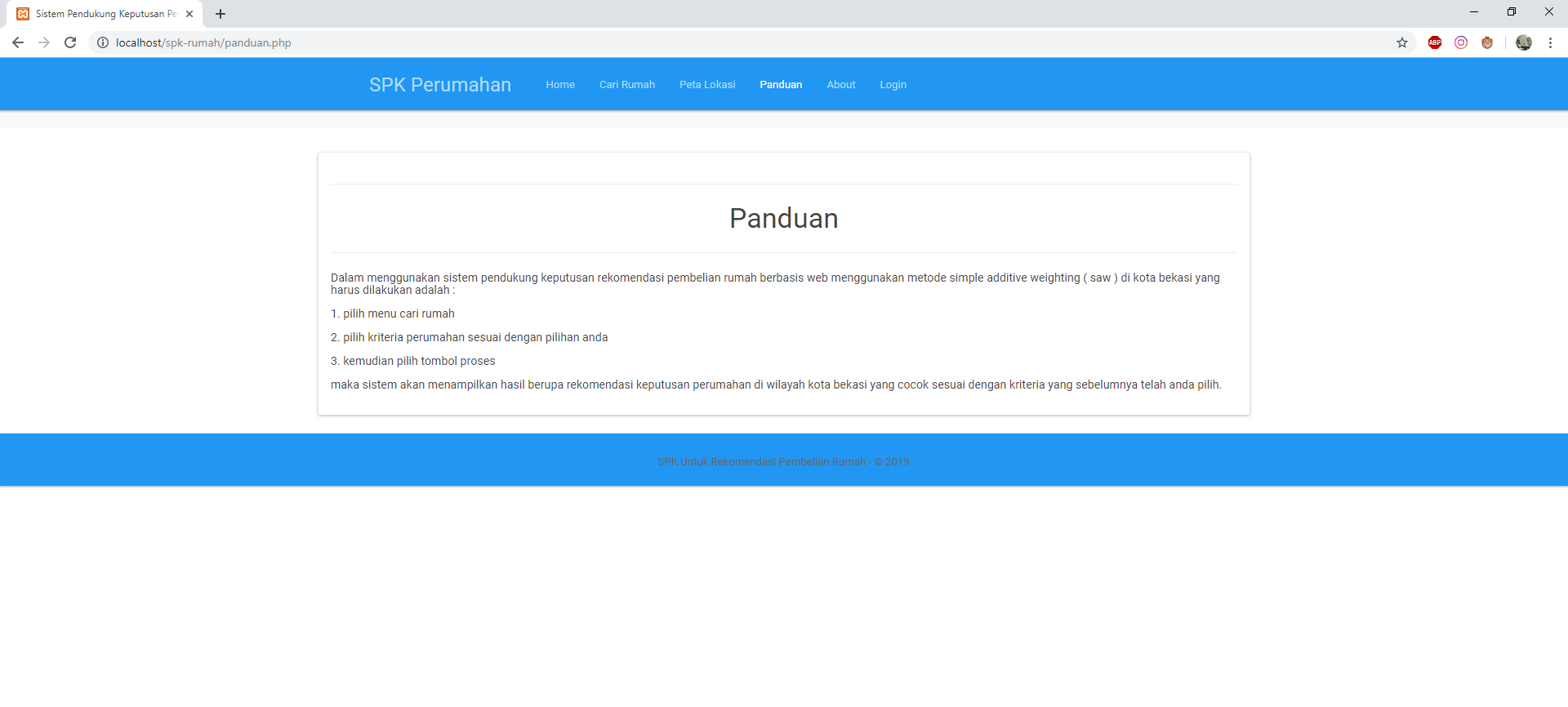
}

}

google.maps.event.addDomListener(window, 'load', initialize);

</script>

* 1. Menu Panduan



**Gambar 4. 5** Implementasi Menu Panduan

Pada menu Panduan calon pembeli dapat melihat panduan penggunaan sistem pendukung keputusan dan proses penggunaan sistem, agar dalam mengunakan sistem dengan memahami fungsi pembuatannya. Dibawah ini adalah pengkodingan untuk halaman tentang :

<div class="col-xs-12 col-sm-12 col-md-12">

<div class="panel panel-default">

<div class="panel-body">

<hr>

<h3 class="intro-text text-center">Panduan</h3>

<hr>

<?php

$sql="SELECT panduan FROM panduan";

$panduan = $db->query($sql);

$all = $panduan->fetch(PDO::FETCH\_ASSOC);

echo $all['panduan'];

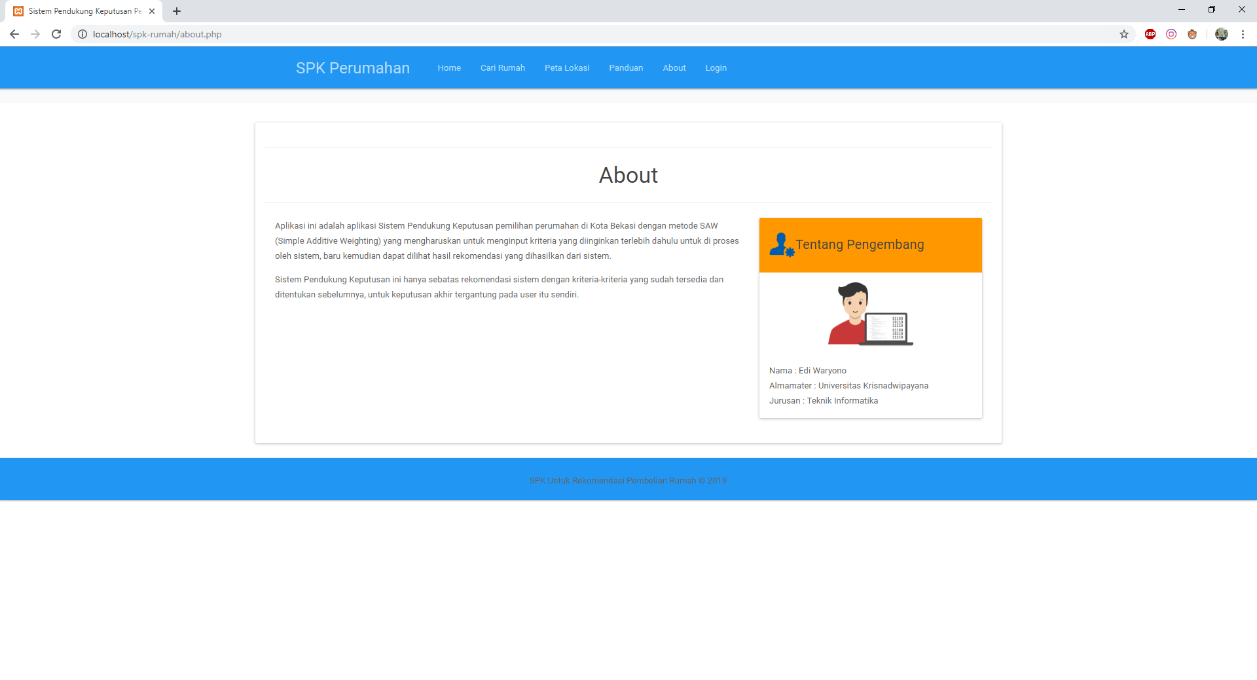
?>

</div>

</div>

</div>

* 1. Menu Tentang



**Gambar 4. 6** Implementasi Menu Tentang

Pada menu Peta Tentang calon pembeli dapat melihat sekilas mengenai sistem pendukung keputusan yang telah dibangun, dengan tujuan agar lebih mengenal terhadap sistem pendukung keputusan untuk rekomendasi pembelian rumah di kota bekasi. Dibawah ini adalah pengkodingan untuk halaman tentang :

<div style=’text-align:justify;’>

<?php

$sql="SELECT tentang FROM tentang";

$tentang = $db->query($sql);

$all = $tentang->fetch(PDO::FETCH\_ASSOC);

echo $all['tentang'];

?>

</div>

<div class="col-xs-12 col-sm-12 col-md-4">

<div class="panel panel-warning">

<div class="panel-heading"><h5><img src="images/admin.png" eight="40px" width="40px">Tentang Pengembang</h5></div>

<div class="panel-body">

<center><img src="images/pengembang.png" height="100px" width="130px"></center><br>

Nama : Edi Waryono <br>

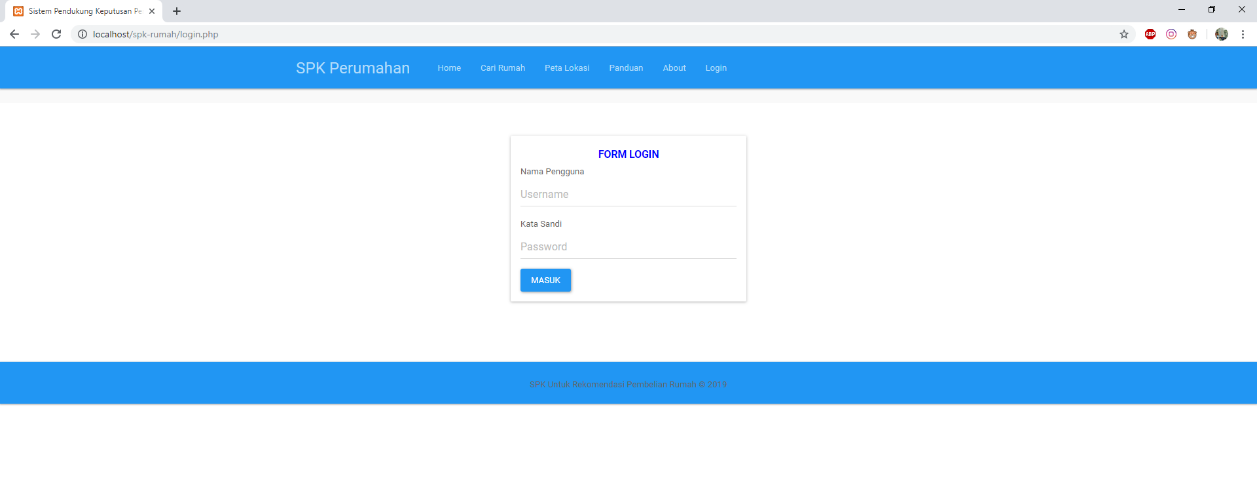
Almamater : Universitas Krisnadwipayana <br>

</div>

</div>

</div>

* 1. Menu *Login*



**Gambar 4. 7** Implementasi Menu Login

Halaman Menu *Login* digunakan oleh admin untuk masuk ke halaman utama *backend,* halaman ini mengharuskan admin memasukan *username* dan *password* dengan benar. Jika username benar maka akan berhasil masuk kedalam halaman utama *backend* jika *username* dan *password* salah maka akan ada peringatan data yang dimasukan salah, dan harus mengulangi dengan memasukan *username* dan *password* yg benar. Dibawah ini adalah pengkodingan untuk halaman *login :*

<div style="margin-top: 20px;" class="panel panel-default">

<div class="panel-body">

<div class="text-center"><font color="blue"><b><big>FORM LOGIN</big></b></font></div>

<form method="post">

<div class="form-group">

<label for="InputUsername1">Nama Pengguna</label>

<input type="text" class="form-control" name="username" id="InputUsername1" placeholder="Username">

</div>

<div class="form-group">

<label for="InputPassword1">Kata Sandi</label>

<input type="password" class="form-control" name="password" id="InputPassword1"

</div>

<button type="submit" class="btn btn-primary">Masuk</button>

</form>

</div>

</div>

1. Halaman *Backend*

Halaman *backend* adalah sebuah halaman belakang yang mempunyai tampilan khusus yang dibuat dengan tujuan untuk *me-manage* sebuah *website*.

* 1. Halaman Utama *backend*



**Gambar 4. 8** Implementasi interface halaman utama backend

<?php include\_once 'header.php'; include\_once 'includes/nilai.inc.php';

$pro3 = new Nilai($db);

$stmt3 = $pro3->readAll();

include\_once 'includes/alternatif.inc.php';

$pro1 = new Alternatif($db);

$stmt1 = $pro1->readAll();

$stmt4 = $pro1->readAll();

include\_once 'includes/kriteria.inc.php';

$pro2 = new Kriteria($db);

$stmt2 = $pro2->readAll();

?> <div class="row">

<div class="col-xs-12 col-sm-12 col-md-4">

<div class="panel panel-warning"></div>

<div class="col-xs-12 col-sm-12 col-md-4">

<div class="panel panel-warning"></div></div></div>

<div class="col-xs-12 col-sm-12 col-md-4">

<div class="panel panel-warning">

</ol></div></div></div></div> <div id="container2" style="min-width: 310px; height: 400px; margin: 0 auto"></div> var chart1; // globally available

$(document).ready(function() { chart1 = new Highcharts.Chart({

chart: { renderTo: 'container2', type: 'column' }, title: { text: 'Grafik Perangkingan '}, xAxis: { categories: ['Alternatif'] }, yAxis: { title: { text: 'Jumlah Nilai' } },

series: [ <?php while ($row4 = $stmt4->fetch(PDO::FETCH\_ASSOC)){

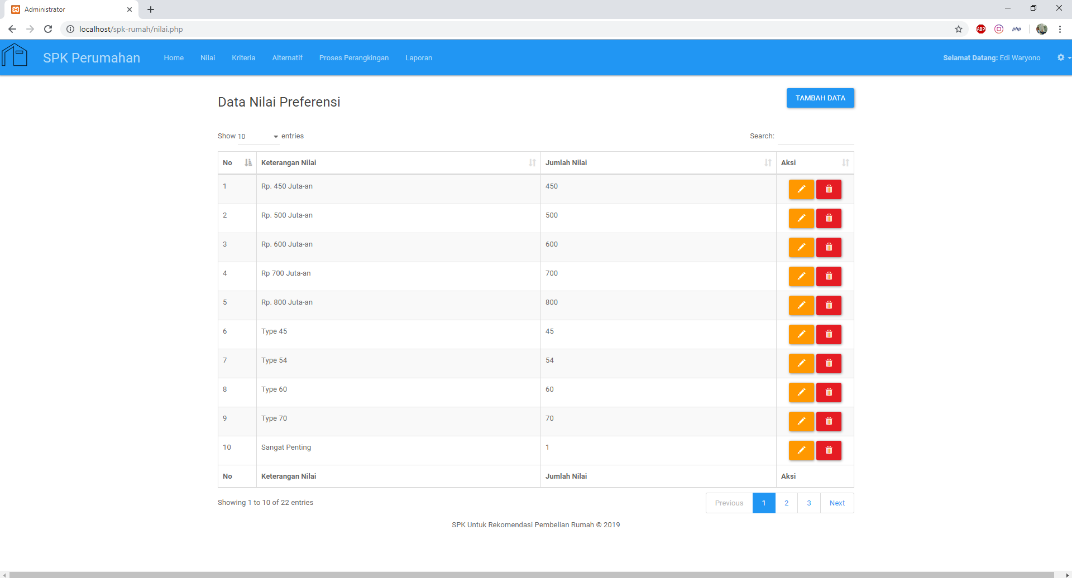
?> //data yang diambil dari database dimasukan ke variable nama dan data

{ name: '<?php echo $row4['nama\_alternatif'] ?>', data: [<?php echo $row4['hasil\_alternatif'] ?>] }, <?php } ?> ] }); }); </script> </body>

</html>

Halaman *Backend* ada beberapa menu, rancangan untuk setiap menu halaman *backend* adalah sebagai berikut :

* 1. Menu Nilai



**Gambar 4. 9** Implementasi Menu Nilai

Pada menu nilai admin dapat mengelola data nilai dan jumlah nilai, yang disediakan untuk calon pembeli melakukan penilain terhadap alternatif yang di tawarkan.

<?php

include\_once 'header.php';

include\_once 'includes/nilai.inc.php';

$pro = new Nilai($db);

$stmt = $pro->readAll();

?>

<div class="row">

<div class="col-md-6 text-left">

<h4>Data Nilai Preferensi</h4>

</div>

<div class="col-md-6 text-right">

<button onclick="location.href='nilai-baru.php'" class="btn btn-primary">Tambah Data</button>

</div></div>

<br/>

<table width="100%" class="table table-striped table-bordered" id="tabeldata">

<thead>

<tr>

<th width="30px">No</th>

<th>Keterangan Nilai</th>

<th>Jumlah Nilai</th>

<th width="100px">Aksi</th>

</tr></thead>

<tfoot>

<tr>

<th>No</th>

<th>Keterangan Nilai</th>

<th>Jumlah Nilai</th>

<th>Aksi</th>

</tr>

</tfoot>

<tbody>

<?php

$no=1;

while ($row = $stmt->fetch(PDO::FETCH\_ASSOC)){

?>

<tr>

<td><?php echo $no++ ?></td>

<td><?php echo $row['ket\_nilai'] ?></td>

<td><?php echo $row['jum\_nilai'] ?></td>

<td class="text-center">

<a href="nilai-ubah.php?id=<?php echo $row['id\_nilai'] ?>" class="btn btn-warning"><span class="glyphicon glyphicon-pencil" aria-hidden="true"></span></a>

<a href="nilai-hapus.php?id=<?php echo $row['id\_nilai'] ?>" onclick="return confirm('Yakin ingin menghapus data')" class="btn btn-danger"><span class="glyphicon glyphicon-trash" aria-hidden="true"></span></a>

</td>

</tr>

<?php

}

?></tbody>

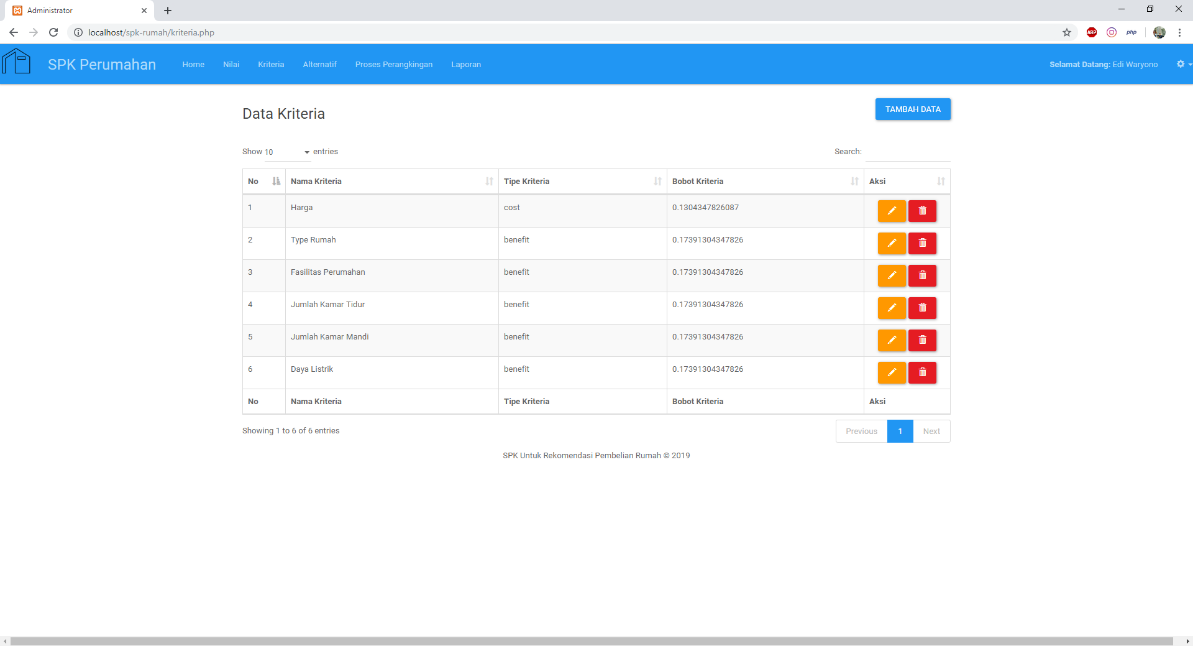
</table>

<?php

include\_once 'footer.php';

?>

* 1. Menu Kriteria



**Gambar 4. 10** Implementasi Menu Kriteria

Menu Kriteria dapat mengelola data kriteria yang disediakan untuk calon pembeli melakukan pilihan rumah berdasarkan alternatif yang di tawarkan.

<?php

include\_once 'header.php';

include\_once 'includes/kriteria.inc.php';

$pro = new Kriteria($db);

$stmt = $pro->readAll();

?>

<div class="row">

<div class="col-md-6 text-left">

<h4>Data Kriteria</h4>

</div>

<div class="col-md-6 text-right">

<button onclick="location.href='kriteria-baru.php'" class="btn btn-primary">Tambah Data</button>

</div>

</div>

<br/>

<table width="100%" class="table table-striped table-bordered" id="tabeldata">

<thead>

<tr>

<th>Nama Kriteria</th>

<th>Tipe Kriteria</th>

<th>Bobot Kriteria</th>

<th width="100px">Aksi</th>

</tr>

</thead>

<tfoot>

<tr>

<th>No</th>

<th>Nama Kriteria</th>

<th>Tipe Kriteria</th>

<th>Bobot Kriteria</th>

<th>Aksi</th>

</tr>

</tfoot>

<tbody>

<?php

$no=1;

while ($row = $stmt->fetch(PDO::FETCH\_ASSOC)){

?>

<tr>

<td><?php echo $no++ ?></td>

<td><?php echo $row['nama\_kriteria'] ?></td>

<td><?php echo $row['tipe\_kriteria'] ?></td>

<td><?php echo $row['bobot\_kriteria'] ?></td>

<td class="text-center">

<a href="kriteria-ubah.php?id=<?php echo $row['id\_kriteria'] ?>" class="btn btn-warning"><span class="glyphicon glyphicon-pencil" aria-hidden="true"></span></a>

<a href="kriteria-hapus.php?id=<?php echo $row['id\_kriteria'] ?>" onclick="return confirm('Yakin ingin menghapus data')" class="btn btn-danger"><span class="glyphicon glyphicon-trash" aria-hidden="true"></span></a>

</td>

</tr>

<?php

}

?>

</tbody>

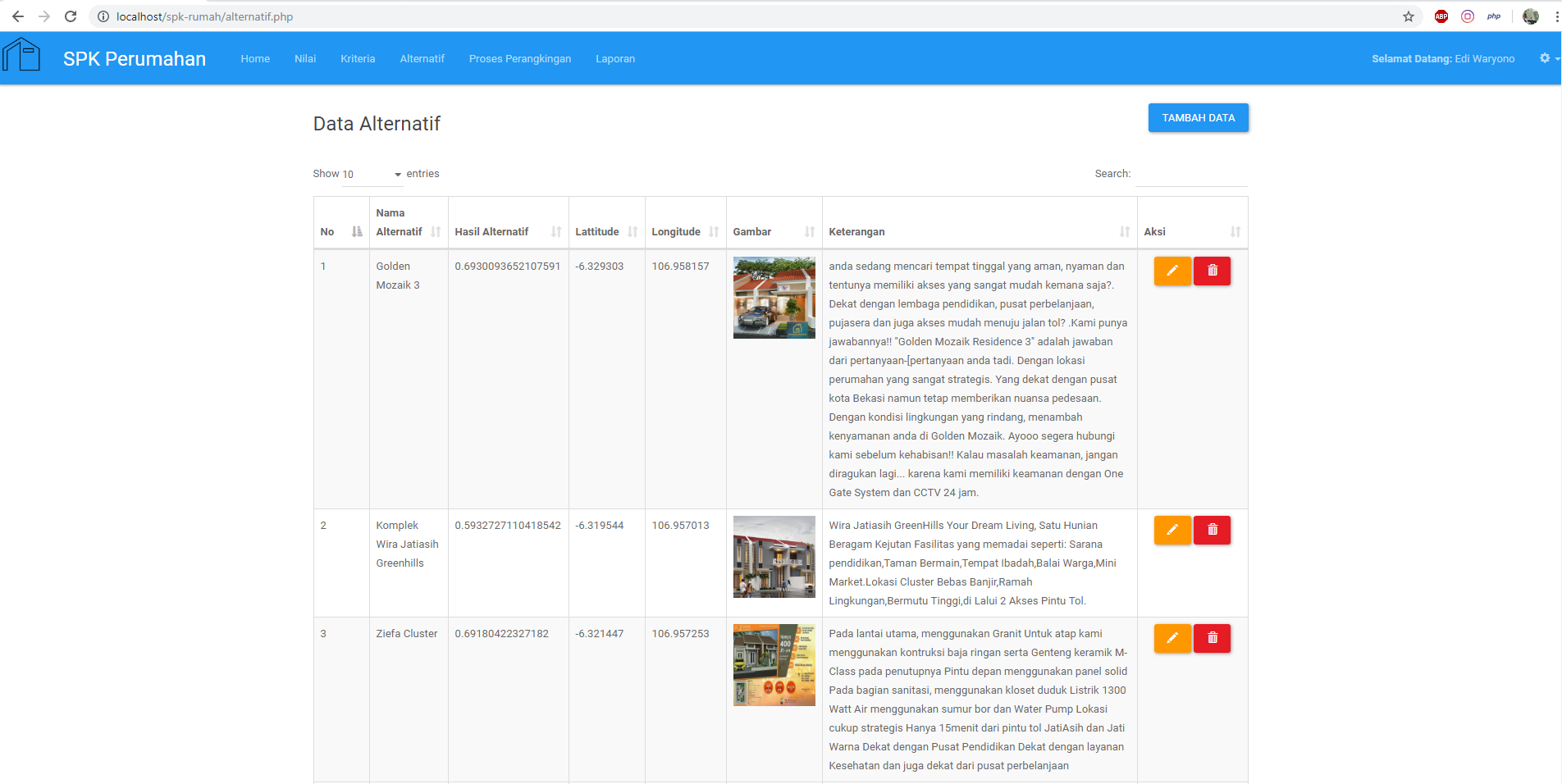
</table>

<?php

include\_once 'footer.php';

?>

* 1. Menu Alternatif



**Gambar 4. 11** Implementasi Menu Alternatif

Menu Alternatif dapat mengelola data alternatif yang disediakan untuk calon pembeli melakukan pilihan rumah berdasarkan alternatif yang di tawarkan.

<?php

include\_once 'header.php';

include\_once 'includes/alternatif.inc.php';

$pro = new Alternatif($db);

$stmt = $pro->readAll(); ?>

<div class="row">

<div class="col-md-6 text-left">

<h4>Data Alternatif</h4></div>

<div class="col-md-6 text-right">

<button onclick="location.href='alternatif-baru.php'" class="btn btn-primary">Tambah Data</button>

</div></div><br/>

<table width="100%" class="table table-striped table-bordered" id="tabeldata">

<thead>

<tr><th width="30px">No</th>

<th>Nama Alternatif</th>

<th>Hasil Alternatif</th>

<th>Lattitude</th>

<th>Longitude</th>

th>Gambar</th>

<th>Keterangan</th>

<th width="100px">Aksi</th></tr>

</thead>

<tfoot>

<tr>

<th>No</th>

<th>Nama Alternatif</th>

<th>Hasil Alternatif</th>

<th>Lattitude</th>

<th>Longitude</th>

th>Gambar</th>

<th>Keterangan</th>

<th>Aksi</th>

</tr>

</tfoot>

<tbody>

<?php

$no=1;

while ($row = $stmt->fetch(PDO::FETCH\_ASSOC)){

?>

<tr>

<td><?php echo $no++ ?></td>

<td><?php echo $row['nama\_alternatif'] ?></td>

<td><?php echo $row['hasil\_alternatif'] ?></td>

<td><?php echo $row['lat'] ?></td>

<td><?php echo $row['lon'] ?></td>

<td><?php echo $row['image'] ?></td>

<td><?php echo $row[keterangan] ?></td>

<td class="text-center">

<a href="alternatif-ubah.php?id=<?php echo $row['id\_alternatif'] ?>" class="btn btn-warning"><span class="glyphicon glyphicon-pencil" aria-hidden="true"></span></a>

<a href="alternatif-hapus.php?id=<?php echo $row['id\_alternatif'] ?>" onclick="return confirm('Yakin ingin menghapus data')" class="btn btn-danger"><span class="glyphicon glyphicon-trash" aria-hidden="true"></span></a>

</td>

</tr>

<?php

}

?>

</tbody>

</table>

<?php

include\_once 'footer.php';

?>

* 1. Menu Perangkingan



**Gambar 4. 12** Implementasi Menu Perangkingan

Menu Perangkingan dapat mengelola data rangking yang disediakan untuk calon pembeli dalam merangking sesuai kebututuhan.

<?php //MENAMPILKAN NILAI ALTERNATIF DARI MASING MASING KRITERIA , DENGAN MENGAMBIL DATA DARI BASIS DATA.

$no=1;

while ($row = $stmt->fetch(PDO::FETCH\_ASSOC)){

?>

<tr>

<td><?php echo $no++ ?></td>

<td><?php echo $row['nama\_alternatif'] ?></td>

<td><?php echo $row['nama\_kriteria'] ?></td>

<td><?php echo $row['nilai\_rangking'] ?></td>

<td class="text-center">

<a href="rangking-ubah.php?ia=<?php echo $row['id\_alternatif'] ?>&ik=<?php echo $row['id\_kriteria'] ?>" class="btn btn-warning"><span class="glyphicon glyphicon-pencil" aria-hidden="true"></span></a>

<a href="rangking-hapus.php?ia=<?php echo $row['id\_alternatif'] ?>&ik=<?php echo $row['id\_kriteria'] ?>" onclick="return confirm('Yakin ingin menghapus data')" class="btn btn-danger"><span class="glyphicon glyphicon-trash" aria-hidden="true"></span></a>

</td>

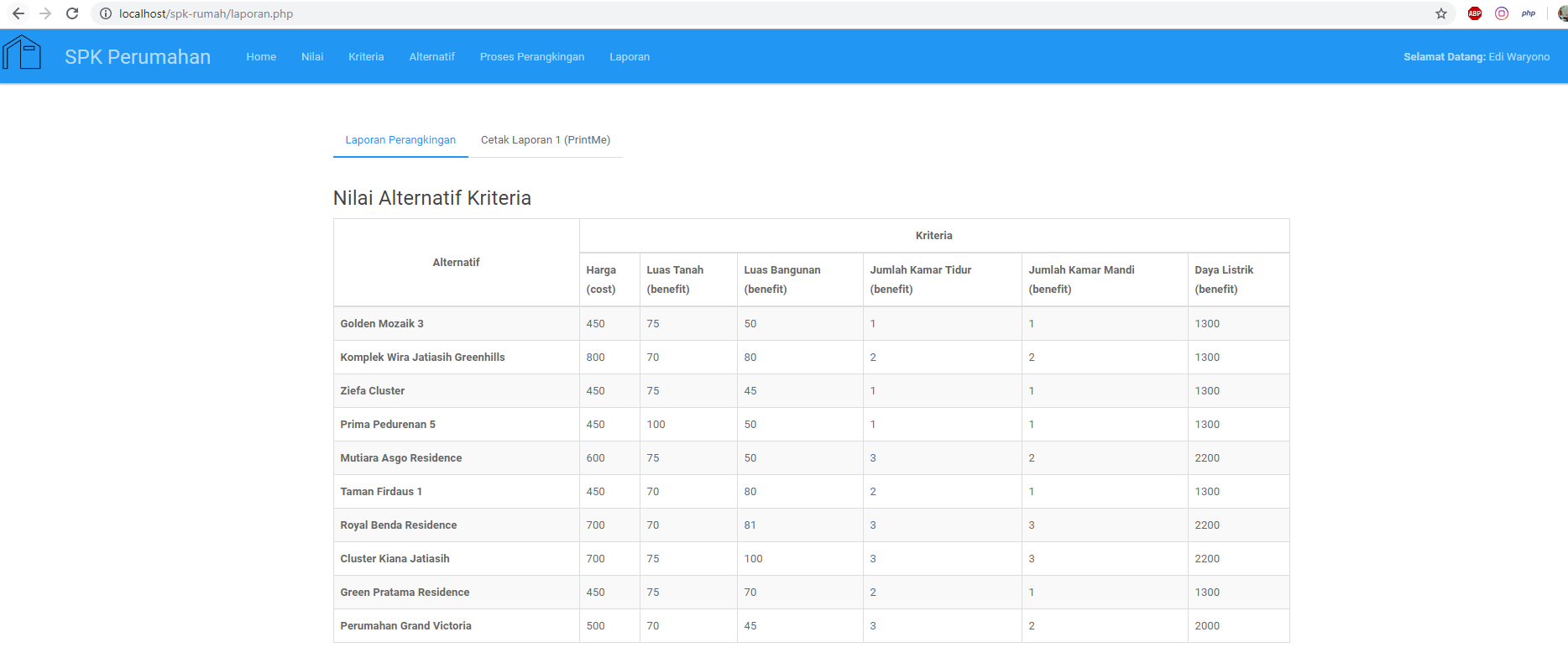
</tr>

<?php

}

?>

* 1. Menu Laporan



**Gambar 4. 13** Implementasi Menu Laporan

Menu Laporan dapat mencetak data rangking yang telah melalui tahap perhitungan menggunkan metode *simple additive weighting method* *(saw).*

<!-- MENAMPILKAN NILAI PREFERENSI DARI Alternatif TERHADAP MASING MASING Kriteria, SEBELUM DILAKUKAN NORMALISASI R -->

<h4>Nilai Alternatif Kriteria</h4>

<table width="100%" class="table table-striped table-bordered">

<thead>

<tr>

<th rowspan="2" style="vertical-align: middle" class="text-center">Alternatif</th>

<th colspan="<?php echo $stmt2x->rowCount(); ?>" class="text-center">Kriteria</th>

</tr>

<tr>

<?php

while ($row2x = $stmt2x->fetch(PDO::FETCH\_ASSOC)){

?>

<th><?php echo $row2x['nama\_kriteria'] ?><br/>(<?php echo $row2x['tipe\_kriteria'] ?>)</th>

<?php

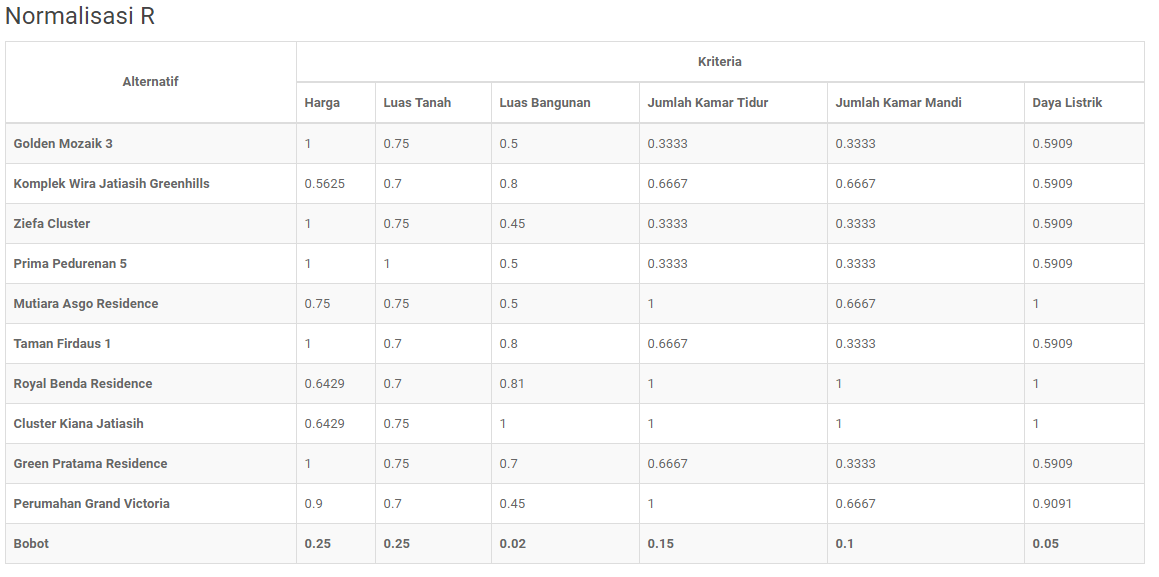
?>

</tr>

</thead>

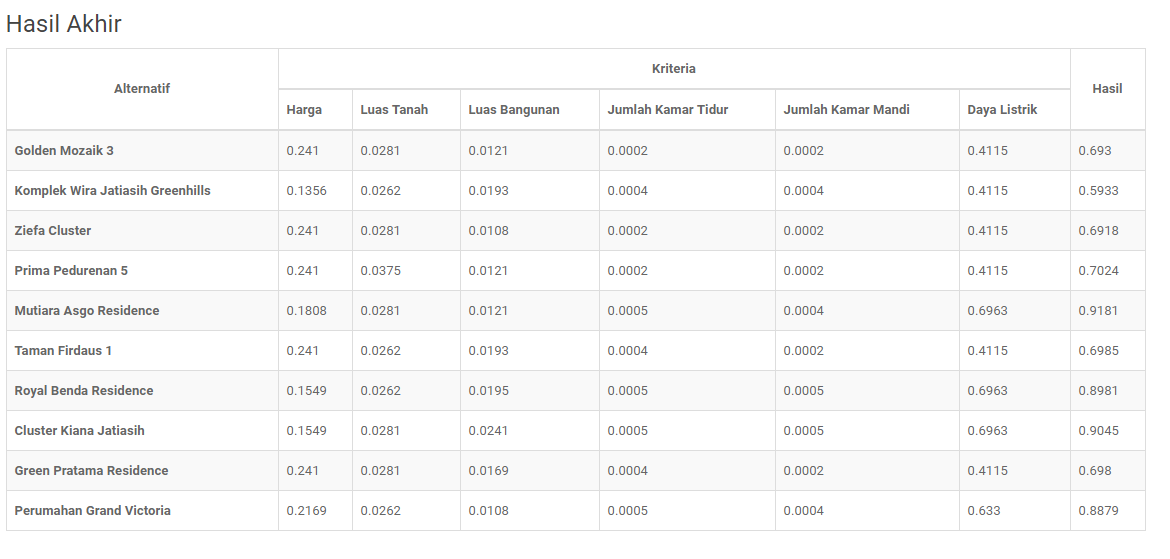
<tbody>

</table>



**Gambar 4. 14** Laporan Nilai dan Normalisasi R

Menu Laporan dapat mencetak data alternatif dan kriteria yang telah berisi data yang telah di input oleh admin, dan berdasarkan penilaian rumah yang sesuai dengan kebutuhan calon pembeli. Kemudian data – data tersebut di hitung dengan mencari nilai normalisasi setiap alternatif, dan terlihat hasil normalisasi pada gambar di atas, kemudian akan muncul hasil akhir perhitungan dengan mengalikan bobot dengan hasil normalisasi yang telah dihitung sebelumnya. Maka hasil akhir dari proses perhitungan pemilihan rumah dengan metode saw akan terlihat seperti dibawah ini.



**Gambar 4. 15** Laporan Hasil Perhitungan SAW

<!-- RUMUS UNTUK PERHITUNGAN AKHIR DARI METODE SAW , HASILNYA AKAN DISAJIKAN DENGAN GRAFIK DI HALAMAN BERANDA.. -->

<!-- HASIL YANG PALING BESAR ITULAH YANG MENJADI PILIHAN Alternatif -->

<h4>Hasil Akhir</h4>

<table width="100%" id="table-akhir" class="table table-striped table-bordered"><thead>

<tr><th rowspan="2" style="vertical-align: middle" class="text-center">Alternatif</th>

<th colspan="<?php echo $stmt2->rowCount(); ?>" class="text-center">Kriteria</th>

<th rowspan="2" style="vertical-align: middle" class="text-center">Hasil</th></tr>

<tr><?php

while ($row2 = $stmt2->fetch(PDO::FETCH\_ASSOC)){

?>

<th><?php echo $row2['nama\_kriteria'] ?></th>

<?php

}?>

</tr></thead>

<tbody> <?php

while ($row1 = $stmt1->fetch(PDO::FETCH\_ASSOC)){

?>

<tr><th><?php echo $row1['nama\_alternatif'] ?></th>

<?php

$a= $row1['id\_alternatif'];

$stmtr = $pro->readR($a);

while ($rowr = $stmtr->fetch(PDO::FETCH\_ASSOC)){

?>

<td>

<?php

echo round ($rowr['bobot\_normalisasi'],4);

?>

</td>

<?php

}?>

<td>

<?php echo round ($row1['hasil\_alternatif'],4);?>

</td></tr>

<?php

}

?> </tbody></table>

</div></div></div>

### Implementasi Metode SAW pada Sistem Pendukung Keputusan

* + 1. **Mendefiniskan setiap kriteria**

<?php //MENAMPILKAN NILAI ALTERNATIF DARI SETIAP KRITERIA

while ($row = $stmt->fetch(PDO::FETCH\_ASSOC)){ ?>

<tr>

<td><?php echo $no++ ?></td>

<td><?php echo $row['nama\_alternatif'] ?></td>

<td><?php echo $row['nama\_kriteria'] ?></td>

<td><?php echo $row['nilai\_rangking'] ?></td>

</tr>

<?php

}?>

* + 1. **Menormalisasi Setiap Alternatif**

//NORMALISASI SETIAP KRITERIA

function readMax($b){

$query = "SELECT max(nilai\_rangking) as mnr1 FROM " . $this->table\_name . " WHERE id\_kriteria='$b' LIMIT 0,1";

$stmt = $this->conn->prepare( $query );

$stmt->execute();

return $stmt;

}

function readMin($b){

$query = "SELECT min(nilai\_rangking) as mnr2 FROM " . $this->table\_name . " WHERE id\_kriteria='$b' LIMIT 0,1";

$stmt = $this->conn->prepare( $query );

$stmt->execute();

return $stmt;

}

function readHasil($a){

$query = "SELECT sum(bobot\_normalisasi) as bbn FROM " . $this->table\_name . " WHERE id\_alternatif='$a' LIMIT 0,1";

$stmt = $this->conn->prepare( $query );

$stmt->execute();

return $stmt;

}

1. **Menghitung Nilai Bobot Referensi**

<?php //MENAMPILKAN BOBOT DARI MASING-MASING Kriteria

}

?><tr>

<td><b>Bobot</b></td>

<?php

while ($row2yx = $stmt2yx->fetch(PDO::FETCH\_ASSOC)){

?>

<td><b><?php echo $row2yx['bobot\_kriteria'] ?></b></td>

<?php

}

?>

</tr>

</tbody>

</table>

1. **Melakukan Perangkingan**

<!--RUMUS JIKA JENIS KRITERIANYA BENEFIT-->

<?php

if($tipe=='benefit'){

$stmtmax = $pro->readMax($b);

$maxnr = $stmtmax->fetch(PDO::FETCH\_ASSOC);

echo $nor = $rowr['nilai\_rangking']/$maxnr['mnr1'];

//JIKA KRITERIANYA COST, MAKA YANG DIGUNAKAN RUMUS INI

} else{ $stmtmin = $pro->readMin($b);

$minnr = $stmtmin->fetch(PDO::FETCH\_ASSOC);

echo $nor = $minnr['mnr2']/$rowr['nilai\_rangking']; }

$pro->ia = $a;

$pro->ik = $b;

$pro->nn2 = $nor;

$pro->nn3 = $bobot\*$nor;

$pro->normalisasi(); ?></td>

<?php?> <td>

<?php //HASIL AKHIR DARI PROSES SAW

$stmthasil = $pro->readHasil($a);

$hasil = $stmthasil->fetch(PDO::FETCH\_ASSOC);

echo $hasil['bbn'];

$pro->ia = $a;

$pro->has = $hasil['bbn'];

$pro->hasil(); ?>

</td></tr>

<?php

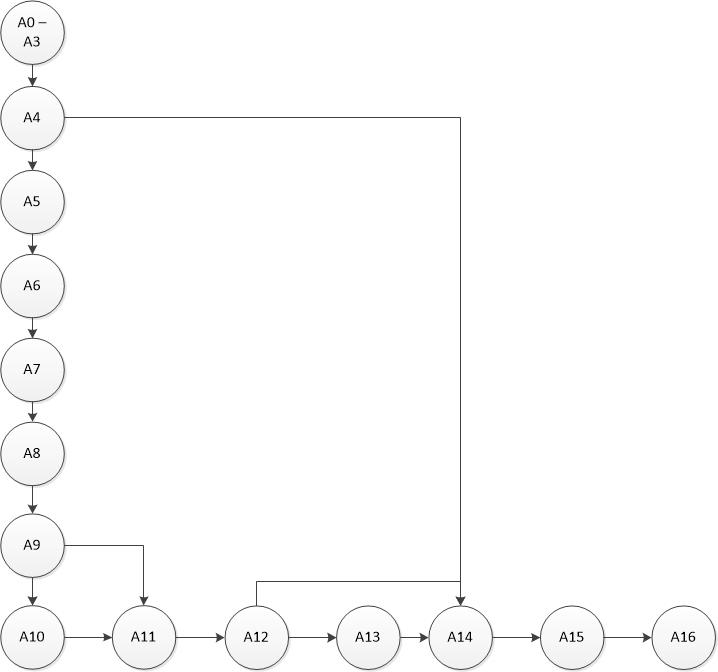
## Pengujian Sistem

Menurut Williams dalam jurnal Muhammad Afif Fakhri, Ismiarta Aknuranda & Djoko Pramono (2018) pengujian adalah sebuah proses untuk menganalisis sebuah perangkat lunak dalam mendeteksi perbedaan antara kondisi saat ini dan yang diharuskan, serta untuk menganalisis fitur-fitur dari sebuah perangkat lunak. Pengujian *blackbox* adalah pengujian yang tidak berfokus terhadap bagian dalam dari sistem, tetapi lebih berfokus kepada keluaran dari sistem yang dihasilkan dari respon input yang dipilih serta kondisi eksekusinya. Menurut Nidhira dan Jagruthi dalam jurnal pengujian *whitebox* digunakan untuk mendeteksi kesalahan pada kode program. Pengujian *whitebox* dilakukan pada saat kode program telah selesai diimplementasikan.

### Pengujian *White Box*

**Tabel 4. 9** Tabel Pengujian *White Box Login.php*

|  |  |
| --- | --- |
| A0 | <?php |
| A1 | include\_once 'includes/config.php'; |
| A2 | $config = new Config(); |
| A3 | $db = $config->getConnection(); |
| A4 | if($\_POST){ |
| A5 | include\_once 'includes/login.inc.php'; |
| A6 | $login = new Login($db); |
| A7 | $login->userid = $\_POST['username']; |
| A8 | $login->passid = $\_POST['password']; |
| A9 | if($login->login()){ |
| A10 | echo "<script>location.href='index.php'</script>"; |
| A11 | } |
| A12 | else{ |
| A13 | echo "<script>alert('Username & Password Salah')</script>"; |
| A14 | } |
| A15 | } |
| A16 | ?> |



**Gambar 4. 16** Flowgraph Login

* + 1. *Cyclomatic Complexity*

Yaitu *metric* perangkat lunak yang menyajikan ukuran kuantitatif dari kekompleksan logikal suatu program.

V(G) = Jumlah Minimum path yang kita perlukan agar dapat melintasi semua proses.

N = Jumlah node (berlaku untuk yang telah berubah menjadi tabel yang kanan).

E = Edge (garis pengubung antara node).

E (Edge) = 16, N (Node) = 14

V(G) = E - N + 2

= 16 – 14 + 2

= 4

* + 1. *Predicate Node* (P) yaitu kondisi yang terdapat pada node dan mempunyai

karakteristik dua atau lebih *edge* darinya.

V(G) = P + 1

= 3 + 1 = 4

= 4

* + 1. Jumlah *Region* (R)

*Region* yaitu daerah yang dibatasi oleh *node* dan *edge*. Untuk menghitung *region* daerah diluar *flowgraph* juga harus dihitung. *Flowgraph* *Login* adalah 4 region.

* + 1. *Independent Path*

*Independent Path* yaitu jalur yang melintasi atau melalui dalam program dimana sekurang-kurangnya terdapat proses perintah yang baru atau kondisi yang baru. *Path - path* yang terdapat pada *flowgraph* login adalah :

Path 1 : A0-A1-A2-A3-A4-A5-A6-A7-A8-A9-A10-A11-A12-A13-A14-A15-A16

Path 2 : A0-A1-A2-A3-A4-A14-A15-A16

Path 3 : A0-A1-A2-A3-A4-A5-A6-A7-A8-A9- A11-A12-A13-A14-A15-A16

Path 4 : A0-A1-A2-A3-A4-A5-A6-A7-A8-A9-A10-A11-A12-A14-A15-A16

* + 1. *Test Case*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Path | Data Uji | Pengujian | Keluaran | Hasil |
| Path 1 | Data Salah | Memulai kode dengan tag *<?php* kemudian dengan memanggil *file* *config.php* | Tidak Berhasil Melakukan Koneksi *Database* | Valid |
| Data Salah | Memasukan variabel *user* dan *password* tidak sesuai *database* | Peringatan *user* dan *password* salah | Valid |
| Path 2 | Data Normal | Memulai kode dengan tag *<?php* kemudian dengan memanggil *file* *config.php* | Berhasil Koneksi *Database* | Valid |
| Data Salah | Memasukan variabel *user* benar dan *password* salah | Peringatan *user* dan *password* salah | Valid |
| Path 3 | Data Normal | Memulai kode dengan tag *<?php* kemudian dengan memanggil *file* *config.php* | Berhasil Koneksi *Database* | Valid |
| Data Salah | Memasukan variabel *user* salah dan *password* benar | Menampilkan  Peringatan *user* dan *password* salah | Valid |
| Path 4 | Data Normal | Memulai kode dengan tag <?php kemudian dengan memanggil *file* *config.php* | Berhasil Koneksi *Database* | Valid |
| Data Normal | Memasukan variabel *user* dan *password* sesuai *database* | Menampilkan  *Index.php* | Valid |

### Pengujian *Black Box*

**Tabel 4. 10** Pengujian Black Box

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Fungsi** | **Pengujian** | **Masukan** | **Keluaran** | **Hasil** |
| 1 | Login | Data Normal | Input Username  & Password yang benar | Menampilkan Halaman admin | Berhasil karena  SPK telah dapat menampilkan halaman admin |
| Data Salah | *Input* Username  & Password yang salah | Kembali ke halaman login | Berhasil karena  SPK tidak dapat menampilkan halaman admin |
| 2 | Cari Rumah | Data Normal | Memasukan biodata dan penilaian terhadap rumah seperti harga, luas tnaha, luas bangunan, jumlah kamar tidur, jumlah kamar mandi, dan daya listrik | Menampilkan rumah yang sesuai penilaian yang di masukan | Berhasil karena  SPK dapat menampilkan alternatif rumah sesuai penilaian yang dimasukan oleh calon pembeli |
| Data Salah | Calon pembeli Tidak memasukan biodata terbelih dahulu | SPK menampikan *warning*  Data yang dimasukan tidak lengkap | Berhasil karena  SPK tidak dapat menampilkan alternatif rumah |
| 3 | Input Data Rumah | Data Normal | Admin memasukan data alternatif rumah yaitu nama rumah, dan lokasi rumah | SPK menampilkan data alternatif rumah yang sudah ditambahkan | Berhasil karena  SPK dapat menampilkan data yang ditambahkan. |
| Data Salah | Admin mengosongkan data rumah dan langsung menekan tombol tambah | SPK menampikan *warning*  Data yang dimasukan tidak lengkap | Berhasil karena  SPK tidak dapat menampilkan data yang ditambahkan. |
| 4 | Input  Data Nilai | Data Normal | Admin memasukan data nilai untuk penilaian calon pembeli terhadap kriteria – kriteria yang telah ditentukan. | SPK menampilkan data nilai yang sudah ditambahkan | Berhasil karena  SPK dapat menampilkan data nilai yang telah ditambahkan |
| Data Salah | Admin mengosongkan data nilai dan langsung menekan tombol tambah | SPK menampikan *warning*  Data yang dimasukan tidak lengkap | Berhasil karena  SPK tidak dapat menampilkan data yang ditambahkan. |
| 5 | Input  Data Kriteria | Data Normal | Admin memasukan data kriteria untuk penilaian calon pembeli dalam menentukan pilihan rumah. | SPK menampilkan data kriteria yang sudah ditambahkan | Berhasil karena  SPK dapat menampilkan data kriteria yang telah ditambahkan |
| Data Salah | Admin mengosongkan data kriteria dan langsung menekan tombol tambah | SPK menampikan *warning*  Data yang dimasukan tidak lengkap | Berhasil karena  SPK tidak dapat menampilkan data kriteria yang ditambahkan. |
| 6 | Input Data Rangking | Data Normal | Admin memasukan data rangking untuk mengurutkan alternatif rumah | SPK menampilkan data rangking perumahan | Berhasil karena  SPK dapat menampilkan data data rangking perumahan |
| Data Salah | Admin mengosongkan data rangking | SPK tidak menampilkan nilai rangking perumahan | Berhasil karena  SPK tidak dapat menampilkan data data rangking perumahan |

# BAB V

**PENUTUP**

## Kesimpulan

Kesimpulan yang dapat diambil dari skripsi ini adalah :

1. Sistem pendukung keputusan mampu memberikan rekomendasi rumah sehingga calon pembeli mendapatkan informasi mengenai alternatif rumah yang terbaik sesuai dengan input penilaian calon pembeli terhadap rumah yang dibutuhkan.
2. Sistem pendukung keputusan mampu menerapkan metode *simple additive weighting* dalam pemilihan rumah berdasarkan kriteria yang telah ditentukan sebelumnya.
3. Diharapkan dengan adanya sistem pendukung keputusan ini dapat memberikan informasi kepada calon pembeli dengan beberapa alternatif rumah yang tersedia di dalam sistem pendukung keputusan.
4. Sistem aplikasi ini memerlukan input data dari calon pembeli kemudian sistem pendukung keputusan ini akan memberikan alternatif sesuai dengan apa yang diinput.

## Saran

Saran yang dapat diberikan dari sistem pendukung keputusan ini adalah :

* + - 1. Hasil dari Sistem Pendukung Keputusan Untuk Rekomendasi Pembelian Rumah ini sebaiknya hanya digunakan sebagai alat bantu dalam mengambil keputusan bukan sebagai keputusan *final*.
      2. Sistem Pendukung Keputusan ini kedepannya diharapakan dapat dikembangkan dengan mencakup lokasi lainya dengan data – data alternatif yang lebih banyak dan beragam.
      3. Sistem Pendukung Keputusan ini kedepannya diharapakan dapat dikembangkan dengan lebih banyak menambahkan kriteria – kriteria yang lebih banyak dan sesuai dengan kriteria yang menjadi acuan calon pembeli dalam melakukan pembelian rumah.

# DAFTAR PUSTAKA

Nofriansyah, Dicky., dan Sarjon Defit. 2017. *Multi Criteria Decision Making(MCDM) Pada Sistem Pendukung Keputusan*. Yogyakarta: Deepublish.

Abdulloh, Rohi. 2017. *7 in 1 Pemrograman Web untuk Pemula*. Jakarta: PT Alex Media Komputindo.

Purbadian, Yenda. 2016. *Trik Cepat Membangun Aplikasi Berbasis Web dengan Framework CodeIgniter*. Yogyakarta: CV Andi Offset.

Jannah, Miftahul., Sarwandi,. dan Cyber Creative. 2019. *Mahir Bahasa Pemrograman PHP*. Jakarta: PT Alex Media Komputindo.

Sari, Febrina. 2018. *Metode Dalam Pengambilan Keputusan*. Yogyakarta: Deepublish.

Solichin, Achmad. 2016. *Pemrograman Web dengan PHP dan MySQL*. Jakarta : Penerbit Budi Luhur.

Fathansyah. 2018. *Basis Data*. Bandung : Informatika Bandung.

Rerung, Rintho Rante. 2018. *Pemrograman Web Dasar*. Yogyakarta : Deepublish.

Muslihudin, Muhamad., dan Oktafianto. 2016. *Analisis dan Perancangan Sistem Informasi Menggunakan Model Terstruktur dan UML.* Yogyakarta :Penerbit Andi.

Mulyani, Sri. 2016. *Metode Analisis dan Perancangan Sistem.* Bandung :Abdi Sistematika.

Pressman, R.S. (2015). *Rekayasa Perangkat Lunak: Pendekatan Praktisi Buku I*. Yogyakarta : Andi.

Lita Asyriati Latif, Mohamad Jamil dan Said HI Abbas. 2018. *Buku Ajar: Sistem Pendukung Keputusan Teori dan Implementasi*. Yogyakarta: Deepublish.

Hartika Manao, Berto Nadeak dan Taronisokhi Zebua, “Sistem pendukung keputusan pemilihan perumahan Dengan metode *simple additive weighting (SAW)”* Teknik Informatika STMIK Budi Darma, Vol 1, No 2, Juni 2017.

Tomy Reza Adianto, Zainal Arifin, dan Dyna Marisa Khairina, “Sistem Pendukung Keputusan Pemilihan Rumah Tinggal Di Perumahan Menggunakan *Metode Simple Additive Weighting (SAW)* (Studi Kasus : Kota Samarinda)*”* Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Mulawarman, Vol. 2, No. 1, Maret 2017.

Hilda Dwi Yunita & Fatimah Fahurian, “Sistem Pendukung Keputusan Pemilihan Perumahan Di Bandar Lampung*”* Fakultas Sistem Informasi Fakultas Komputer Universitas Mitra Indonesia, Volume 10, Nomor 1, Juni 2019.