

LAPORAN AKHIR



JUDUL:

**IMPLEMENTASI ALGORITMA *FILTER BASED*
RETRIEVAL PADA SISTEM PENDUKUNG
KEPUTUSAN PEMBELIAN RUMAH**

TIM PENGUSUL :

**Uung Ungkawa, M.T (NIDN: 0411105902)
Julio Nugraha Hursepuny. (NRP: 15-2013-021)**

Dibiayai oleh **Pribadi**

**INSTITUT TEKNOLOGI NASIONAL
AGUSTUS 2016**

Halaman Pengesahan

Judul Penelitian : IMPLEMENTASI ALGORITMA *FILTER BASED RETRIEVAL* PADA SISTEM PENDUKUNG KEPUTUSAN PEMBELIAN RUMAH

Kode/ Nama Rumpun Ilmu : 538/ Rekayasa Perangkat Lunak

Ketua Peneliti :

a. Nama Lengkap : Ir Ung Ungkawa, M.T.
b. NIDN : 0411105902
c. Jabatan Fungsional : Lektor
d. Program Studi : Teknik Informatika
e. Nomor HP : 08121443095
f. Alamat surel (e-mail) : uung@itenas.ac.id

Anggota Peneliti (1)

a. Nama Lengkap : Julio Nugraha Hursepuny
b. NRP : 15-2013-021
c. Instansi : Mahasiswa Institut Teknologi Nasional (ITENAS) Bandung

Anggota Peneliti (2)

a. Nama Lengkap :
b. NIDN :
c. Perguruan Tinggi :

Lama Penelitian Keseluruhan : 6 bulan
Penelitian Tahun ke : 1 (Satu)
Biaya Penelitian Keseluruhan : Rp. 3.000.000,-
Biaya Tahun Berjalan :
- diusulkan ke DIKTI Rp. 0,-
- dana internal PT Rp.0
- dana institusi lain Rp.0
- inkind sebutkan

Mengetahui

Dekan


Dani Rusriawan, S.T, M.T., Ph.D

NIP/NIK

Bandung, 24 - 8 - 2016



Ir Ung Ungkawa, M.T

NIP/NIK. 120071201

Menyetujui
Ketua LPPM Itenas



Dr. Tarsisius Kristyadi, S.T., M.T
NIP/NIK

DAFTAR ISI

BAB I	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Tujuan	3
1.4 Batasan Masalah.....	3
1.5 Metodologi.....	4
1.5.1 Metodologi Penelitian	4
1.5.2 Metodologi Pengembangan Sistem.....	4
1.6 Tinjauan Pustaka.....	6
1.7 Sistematika Penulisan	12
BAB II	13
2.1 Sistem Pendukung Keputusan	13
2.1.1 Karakteristik dan Kemampuan Sistem Pendukung Keputusan (SPK)	14
2.1.2 Tahapan Pengambilan Keputusan	16
2.1.3 Komponen Sistem Pendukung Keputusan.....	17
2.1.4 Keuntungan Sistem Pendukung Keputusan.....	18
2.2 Algoritma Filter Based Retrieval	18
2.2.1 <i>Obtain query from method</i>	19
2.2.2 <i>Filtering Proses</i>	20
2.2.3 <i>Conversational B</i>	26
2.3 Perumahan.....	27
2.4 Kota Bekasi.....	28
2.5 Penalaran Berbasis Kasus.....	28
2.6 Bahasa Pemrograman Java	30
2.6.1 Kata Kunci Simpanan	33
2.7 Eclipse	33
BAB III	35
3.1 Analisis	35
3.1.1 Cara Kerja Sistem	35
3.1.2 Analisis Perencanaan Sistem.....	35

3.1.3	Analisis Kebutuhan Sistem.....	36
3.2	Analisis Sistem Pendukung Keputusan Pembelian Rumah.....	37
3.2.1	Tabel Sample Data Rumah Dijual di Wilayah Kota Bekasi	39
3.2.2	Tabel Hasil Uji.....	40
3.3	Desain Sistem.....	42
3.3.1	Perancangan Sistem.....	42
Daftar Pustaka		43

BAB I

PENDAHULUAN

1.1 Latar Belakang

Rumah merupakan bangunan yang berfungsi sebagai tempat tinggal/hunian dan sarana pembinaan keluarga, sehingga rumah merupakan salah satu kebutuhan dasar manusia. Dari seluruh manusia yang membutuhkan rumah, kelompok yang memiliki kesulitan cukup besar dalam pemenuhan kebutuhan perumahannya adalah masyarakat berpenghasilan rendah. Dengan kondisi seperti ini para pengusaha properti semakin gencar dalam memberikan pelayanan terbaiknya kepada konsumen dengan membuat sistem perumahan mandiri. Untuk mewujudkan perumahan yang ideal bukanlah hal mudah, pengembangan perumahan seharusnya mengerti apa yang diinginkan oleh konsumen sehingga mereka dapat dengan tenang dan nyaman menempati rumah yang mereka beli. Untuk mendapatkan sebuah rumah yang sesuai dengan yang mereka inginkan, konsumen membutuhkan sebuah informasi yang baik serta akurat.

Perumahan merupakan salah satu kebutuhan sekunder, sehingga dalam pemilihan perumahan yang tepat agar sesuai dengan keinginan konsumen. Dalam pemilihan perumahan ada beberapa kriteria yang digunakan seperti harga, lokasi, fasilitas umum, perijinan, desain rumah, dan kredibilitas pengembang.

Kota Bekasi dipilih menjadi objek kota yang diteliti karena kota Bekasi merupakan salah satu daerah peyanggah kota Jakarta yang mana adalah Ibu Kota Negara, yang juga merupakan pusat pemerintahan dan perekonomian dimana banyak orang yang datang untuk mencari pekerjaan dan bahkan tinggal, tetapi seiring berjalannya waktu kota Jakarta tidaklah mampu menampung begitu banyak orang yang ingin tinggal dan menetap disana, maka penulis memilih kota Bekasi sebagai objek penelitian dalam menentukan rumah dan lokasi tempat tinggal.

Untuk dapat memenuhi hal tersebut, perlu suatu sistem pendukung keputusan berbasis komputer atau yang disebut Sistem Pendukung Keputusan (SPK) yang dapat memberikan informasi dan membantu menyediakan berbagai

alternatif yang dapat ditempuh dalam proses pengambilan keputusan. Keputusan yang akan diambil didasarkan pada alternatif-alternatif yang menjadi pertimbangan. Berdasarkan alternatif-alternatif pertimbangan yang ada, akan dibuat perankingan sehingga keputusan dapat diambil sesuai kebutuhan yang diharapkan.

Algoritma *Filter Based Retrieval* merupakan algoritma Sistem Pendukung Keputusan yang memperoleh keinginan pengguna melalui formulir. Ketika, ia melakukan pengambilan menyaring item yang mematuhi keinginan pengguna. Jika set pengambilan cukup kecil, maka item akan ditampilkan kepada pengguna. Jika set pengambilan terlalu besar atau pengguna tidak menemukan item yang diinginkan, sistem akan menyajikan lagi formulir untuk memodifikasi kebutuhan pengguna. Bentuk ini memiliki beberapa nilai awal dan label kustom.

Dalam proses pembelian rumah, konsumen perlu mempertimbangkan beberapa hal yang menjadi acuan standarisasi mereka. Contoh kecilnya bisa dilihat dari segi daerah atau nominal harga. Berdasarkan hal tersebut seharusnya industri properti memiliki fasilitas yang dapat membantu konsumen dalam memudahkan pemilihan standarisasi rumah sesuai yang mereka inginkan. Untuk itu perlu dibangun sebuah sistem rekomendasi yang menggunakan pendekatan khusus terhadap kasus-kasus yang ada seperti algoritma *filter based retrieval* berdasarkan parameter yang nantinya akan diteliti seperti daerah, harga maksimal, intensitas banjir, luas tanah, jumlah lantai, jumlah kamar, jumlah kamar mandi, fasilitas furniture.

Berdasarkan dari uraian yang telah dijelaskan diatas, untuk membangun sebuah aplikasi yang memudahkan konsumen dalam pengambilan keputusan maka dilakukan sebuah penelitian dengan judul **“IMPLEMENTASI ALGORITMA *FILTER BASED RETRIEVAL* PADA SISTEM PENDUKUNG KEPUTUSAN PEMBELIAN RUMAH”**.

1.2 Rumusan Masalah

Dari penjelasan latar belakang masalah yang terjadi, maka dapat dirumuskan permasalahan sebagai berikut:

1. Bagaimana merancang sistem rekomendasi yang dapat membantu konsumen dalam pembelian rumah dengan menggunakan algoritma *filter based retrieval* dalam proses *filtering* data kasus.
2. Bagaimana menentukan fitur – fitur yang akan dijadikan parameter penentu dalam proses *filtering* .

1.3 Tujuan

Tujuan dari penelitian tugas akhir ini adalah pengimplementasian algoritma *filter based retrieval* dalam pembuatan aplikasi sistem rekomendasi pembelian rumah.

1.4 Batasan Masalah

Batasan masalah yang dibahas dalam penelitian ini adalah sebagai berikut :

1. Aplikasi yang dibuat merupakan jenis aplikasi desktop.
2. Pembuatan aplikasi menggunakan bahasa pemrograman java.
3. Penelitian difokuskan pada pemilihan rumah pada perumahan di wilayah Kota Bekasi
4. Parameter yang menjadi acuan pendeteksi keputusan terletak pada hasil wawancara.
5. Data yang akan digunakan adalah data sekunder dari perumahan yang ada di kota Bekasi yang diambil di website www.bekasikota.go.id/read/573/daftar-perumahan-di-bekasi.
6. Data variable rumah diambil dari beberapa website:
 - www.urbanindo.com
 - rumahdijual.com
 - www.rumah.com
 - www.rumah123.com

1.5 Metodologi

Penelitian dan penyusunan tugas akhir ini dilakukan dengan menggunakan tahapan penelitian sebagai berikut :

1.5.1 Metodologi Penelitian

a. Studi Literatur

Literatur yang digunakan adalah yang terkait dengan Algoritma *Filter Based Retrieval*, Pembelajaran tersebut dilakukan dengan cara mencari referensi dan pengumpulan data yang diperoleh dari buku, jurnal, artikel sejenis yang telah dilakukan sebelumnya.

b. Analisis dan Perancangan

Pada tahap ini dilakukan analisis terhadap masalah untuk mencari solusi dalam Perancangan *prototype* sistem berdasarkan studi literatur yang diperoleh.

c. Pembangunan *Prototype* Sistem

Dari hasil analisis dan perancangan sistem yang telah dilakukan pada tahap sebelumnya, maka dilakukan pembangunan *Prototype* sistem.

d. Implementasi dan Pengujian *Prototype* Sistem

Setelah *Prototype* sistem selesai dibuat, maka pada tahap selanjutnya dilakukan pengujian-pengujian terhadap *Prototype* sistem.

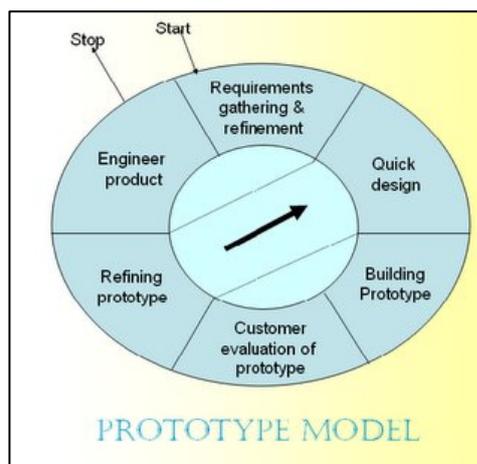
e. Penyusunan Laporan Tugas Akhir

Pada tahap ini dilakukan penulisan laporan dari aplikasi yang dibangun mulai dari perancangan hingga hasil dari pengujian *prototype* sistem.

1.5.2 Metodologi Pengembangan Sistem

Metode pengembangan sistem yang digunakan dalam membangun sistem rekomendasi pembeian rumah adalah metodologi *prototype*. Metodologi ini memungkinkan pihak developer dapat mengoreksi hasil rancangan berdasarkan penelitian dari user sebelum produk sesungguhnya dibuat Tujuan utama dari *Prototype* (Thompson & Wishbow, 1992) sebagai berikut :

- a. Proses revisi dan pengujian terhadap produk dilakukan secara terus menerus agar dapat mendapatkan produk yang diinginkan User. Proses pengujian dapat dilakukan secara keseluruhan maupun partial pada bagian tertentu produk.
- b. Proses pengujian harus memiliki perbandingan baku (Benchmark) sehingga menghasilkan produk yang secara empiris sehingga menghindari kegagalan produk atau terjadi perbedaan persepsi antara Developer atau User.
- c. Dengan adanya proses pengujian terus menerus antara Developer dan User, diharapkan menghasilkan produk yang sesuai dengan kebutuhan User.



Gambar 1 Model Prototype

(Sumber : roysarimilda.wordpress.com/2012)

Aktifitas *prototype* terdiri dari :

1. Mengidentifikasi kebutuhan : analisa terhadap kebutuhan calon *user*.
2. *Quick design* : pembuatan desain global untk membentuk *software*.
3. *Build prototype* : pembuatan *software prototype* termasuk pengujian.
4. Evaluasi hasil *prototype* : mengevaluasi *prototype* dengan analisis kebutuhan calon pemakai.
5. Menyempurnakan *prototype* sesuai dengan kebutuhan dan desain awal sistem.
6. Melakukan pengembangan terhadap *prototype*.

1.6 Tinjauan Pustaka

Penelitian mengenai Implementasi algoritma *filter based retrieval* pada sistem rekomendasi pembelian rumah, terinspirasi dari beberapa penelitian yang telah dilakukan oleh peneliti sebelumnya :

1. **Juan A. Recio-Garcia, Derek Bridge, 2008**, dalam framework *jCOLIBRI* 2.1 menjelaskan tentang **“IMPLEMENTASI ALGORITMA FILTER BASED RETRIEVAL PADA SISTEM PENDUKUNG KEPUTUSAN PEMBELIAN RUMAH”** dan menghasilkan kesimpulan bahwa Algoritma *Filter Based Retrieval* merupakan algoritma *recommender* ini memperoleh keinginan pengguna melalui formulir. Ketika, ia melakukan pengambilan menyaring item yang mematuhi keinginan pengguna. Jika set pengambilan cukup kecil, maka item akan ditampilkan kepada pengguna. Jika set pengambilan terlalu besar atau pengguna tidak menemukan item yang diinginkan, sistem akan menyajikan lagi formulir untuk memodifikasi kebutuhan pengguna. Bentuk ini memiliki beberapa nilai awal dan label kustom.
2. **Adinda Ria Rumondang Veranita, Uung Ungkawa, 2015**, dalam jurnalnya yang berjudul **“IMPLEMENTASI ALGORITMA NEAREST NEIGHBOR DAN CASE SELECTION PADA SISTEM REKOMENDASI PEMBELIAN RUMAH”** penelitian ini menjelaskan tentang pengkajian algoritma *nearest neighbor* dan mengimplementasikan pada klasifikasi data. Dalam proses pembelian rumah biasanya terdapat fitur-fitur yang menjadi standar keinginan konsumen. Dari fitur tersebut dapat dibangun sebuah aplikasi sistem rekomendasi pembelian rumah. Oleh karena itu, dilakukanlah penelitian terhadap implementasi algoritma *nearest neighbor* dan *case selection* pada aplikasi sistem rekomendasi pembelian rumah. Berdasarkan penelitian, algoritma *nearest neighbor* berhasil memberikan hasil akurat dalam proses *similarity* antara *query* dan basis kasus dengan tingkat keberhasilan 85% dan kegagalan 15%, serta dapat menampilkan hasil rekomendasi yang sesuai dengan permintaan *user*.

3. **Fanny Aryanti, Uung Ungkawa, Dewi Rosmala 2013**, dalam penelitian yang berjudul **“Pembangunan Aplikasi Travel Recommender Dengan Metode Case Based Reasoning”** menjelaskan tentang rekomendasi untuk calon wisatawan ketika ingin mencari objek wisata yang ada di Jawa Barat. Aplikasi ini merupakan basis pengetahuan yang memodelkan metode *Case Based Reasoning* dan pemrograman *PHP + JQuery* dalam pembuatan sistemnya. Dari penelitian tersebut ada beberapa teori yang digunakan termasuk pemanfaatan *case selection* pada proses pencocokan hasil rekomendasi.
4. **Meiga Andriyanto, Yofina Rizky Safitri, 2012**, dalam jurnalnya yang berjudul **“Penerapan Metode Simple Additive Weighting Pada Sistem Pendukung Keputusan Pemilihan Rumah Bagi Konsumen PT TERRASSIMA”** menjelaskan tentang Berdasarkan dari hasil pembahasan pada penjelasan sebelumnya terhadap Penerapan *Simple Additive Weighting* Pada Sistem Pendukung Keputusan Pemilihan Rumah Bagi Konsumen PT Terrasima, maka penulis mengambil simpulan, yaitu dengan adanya Sistem Pendukung Keputusan akan memudahkan konsumen dalam menentukan pemilihan rumah dan dapat memberikan rekomendasi rumah terbaik berdasarkan data yang di *input* oleh si pengguna secara cepat dan akurat.
5. **Lalu Arry Tri Laksono, Akuwan Saleh, M. Zen Samsono Hadi, 2013**, dalam jurnalnya yang berjudul **“Pembuatan Aplikasi Mobile Untuk Rekomendasi Pemilihan Rumah Berbasis Fuzzy Database Model Tahani”** menjelaskan tentang Aplikasi mobile diuji dengan diberi *query* dengan kombinasi (Harga Sedang DAN DP Kecil ATAU Besar Angsuran Sedang DAN Durasi Angsuran Lama), hasilnya memunculkan 2 rumah dengan tipe yang sama yaitu San Lorenzo bernomor ID 379 dan 383 dengan nilai rekomendasi paling tinggi untuk *query* ini = 1. Dari hasil rekomendasi tersebut dapat digunakan pertimbangan *user* dalam keputusan akhir.

6. **Tri Shandika Jaya, 2012**, dalam jurnalnya yang berjudul “**Sistem Pemilihan Perumahan Dengan Metode Kombinasi Fuzzy C-MEANS Clustering dan Simple Addtive Weighting**” menjelaskan bahwa Sistem Pemilihan Perumahan membantu pengambil keputusan dalam masalah pemilihan perumahan secara cepat dan mudah, sistem Pemilihan Perumahan dapat digunakan di berbagai platform sistem operasi dan browser, hasil pengujian sistem pada 10 kasus uji menghasilkan 9 kasus yang sesuai dan 1 kasus yang tidak sesuai, hasil rekomendasi perumahan menjadi lebih objektif karena user tidak menentukan alternatif yang akan dipilih secara langsung, penentuan atribut kriteria sangat mempengaruhi hasil perhitungan simple additive weigting

Berdasarkan tinjauan pustaka dari **Juan A. Recio-Garcia, Derek Bridge** yang membuat Algoritma *filter based retrieval* dan mengimplementasikannya pada sistem rekomendasi pembelian rumah, **Fanny Aryanti, Meiga** yang membangun Aplikasi Travel Recomender, **Andriyanto, Yofina Rizky Safitri, Lalu Arry Tri Laksono, Akuwan Saleh, M. Zen Samsono Hadi dan Tri Shandika Jayayang** membangun aplikasi sistem rekomendasi pembelian rumah dengan pengimplementasian metoda dan algoritma yang berbeda. Serta **Adinda Ria Rumondang Veranita** selaku kaka tingkat yang membangun sistem rekomendasi pembelian rumah dengan algoritma *nearest neighbor* sehingga penulis berkeinginan untuk melakukan penelitian terhadap **IMPLEMENTASI ALGORITMA FILTER BASED RETRIEVAL PADA SISTEM REKOMENDASI PEMBELIAN RUMAH.**

Tabel 1 - Perbandingan Tinjauan Pustaka

Judul	Nama	Metode	Tahun	Keterangan
Framework jCOLIBRI 2.1	Juan A. Recio-Garcia, Derek Bridge	<i>Filter Based Retrieval</i>	2008	Pembuatan dan pengimplementasian algoritma Filter Based Retrieval pada sistem pendukung keputusan pembelian rumah.
Implementasi Algoritma <i>Nearest Neighbor dan Case Selection</i> Pada Sistem Rekomendasi Pembelian Rumah	Adinda Ria Rumondang Veranita	<i>Nearest Neighbor dan Case Selection</i>	2015	Pengimplementasian algoritma nearest neighbor dan case selection dalam pembuatan aplikasi sistem rekomendasi pembelian rumah.
Pembangunan Aplikasi Travel <i>Recommender</i> Dengan Metode <i>Case Based Reasoning</i>	Fanny Aryanti	<i>Case Based Reasoning</i>	2013	Membangun aplikasi <i>Travel Recommender</i> untuk calon wisatawan yang berbasis web dengan menggunakan metoda case base reasoning dan menggunakan bahasa pemrograman PHP+jQuery sebagai pengganti dari sistem manual dan dapat digunakan oleh user sebagai alat untuk mencari rekomendasi tentang objek wisata di Jawa Barat.

Tabel 1 - Perbandingan Tinjauan Pustaka

Judul	Nama	Metode	Tahun	Keterangan
<p>Penerapan Metode <i>Simple Additive Weighthing</i> Pada Sistem Pendukung Keputusan Pemilihan Rumah Bagi Konsumen PT TERRASSIMA</p>	<p>Meiga Andriyanto Yofina Rizky Safitri</p>	<p><i>Simple Additive Weighthing</i></p>	<p>2012</p>	<p>Berdasarkan dari hasil pembahasan pada penjelasan sebelumnya terhadap Penerapan <i>Simple Additive Weighthing</i> Pada Sistem Pendukung Keputusan Pemilihan Rumah Bagi Konsumen PT Terrasima, maka penulis mengambil simpulan, yaitu dengan adanya Sistem Pendukung Keputusan akan memudahkan konsumen dalam menentukan pemilihan rumah dan dapat memberikan rekomendasi rumah terbaik berdasarkan data yang di <i>input</i> oleh si pengguna secara cepat dan akurat.</p>
<p>Pembuatan Aplikasi Mobile Untuk Rekomendasi Pemilihan Rumah</p>	<p>Lalu Arry Tri</p>	<p>Fuzzy Database</p>	<p>2013</p>	<p>Aplikasi mobile diuji dengan diberi <i>query</i> dengan kombinasi (Harga Sedang DAN DP Kecil ATAU Besar Angsuran Sedang DAN Durasi Angsuran Lama), hasilnya memunculkan 2 rumah dengan tipe yang sama yaitu San</p>

Tabel 1 - Perbandingan Tinjauan Pustaka

Judul	Nama	Metode	Tahun	Keterangan
Berbasis Fuzzy Database Model Tahani	Laksono, Akuwan Saleh, M. Zen Samsono Hadi	Model Tahani		Lorenzo bernomor ID 379 dan 383 dengan nilai rekomendasi paling tinggi untuk <i>query</i> ini = 1. Dari hasil rekomendasi tersebut dapat digunakan pertimbangan <i>user</i> dalam keputusan akhir.
Sistem Pemilihan Perumahan Dengan Metode Kombinasi Fuzzy C-MEANS Clustering dan Simple Addtive Weighting	Tri Shandika Jaya	<i>Fuzzy C- MEANS Clustering dan Simple Addtive Weighting</i>	2012	
Implementasi Algoritma <i>Filter Based Retrieval</i> Pada Sistem Rekomendasi Pembelian Rumah	Julio Nugraha Hursepuny	<i>Filter Based Retrieval</i>	2015	Pengimplementasian algoritma <i>filter based retrieval</i> dalam pembuatan aplikasi sistem rekomendasi pembelian rumah, dengan menentukan fitur-fitur untuk penentuan parameter sebagai tolak ukur proses filtering.

1.7 Sistematika Penulisan

Untuk memberikan kemudahan bagi penulis dalam menyusun laporan Tugas Akhir ini, maka sistematika penulisan laporan ini adalah sebagai berikut :

- **BAB I PENDAHULUAN**

Bab ini berisikan latar belakang masalah, rumusan masalah, batasan masalah, tujuan, batasan masalah, tinjauan pustaka, metodologi penelitian, sistematika penulisan.

- **BAB II LANDASAN TEORI**

Dalam bab ini akan diuraikan mengenai pengertian dari istilah-istilah maupun pengetahuan-pengetahuan yang digunakan sebagai penjelasan lebih dalam dari tinjauan pustaka yang ada.

- **BAB III ANALISIS DAN PERANCANGAN**

Bab ini berisikan penjelasan mengenai analisis dan perancangan dari sistem.

- **BAB IV IMPLEMENTASI DAN PENGUJIAN**

Bab ini berisikan mengenai implementasi dari sistem yang telah diciptakan dan hasil pengujian terhadap fungsionalitas sistem.

- **BAB V PENUTUP**

Dalam bab ini akan diuraikan mengenai kesimpulan dari hasil pengujian sistem.

- **DAFTAR PUSTAKA**

Pada bagian ini berisi tentang referensi - referensi yang telah dipakai oleh penulis sebagai acuan dan penunjang serta parameter yang mendukung penyelesaian Tugas Akhir ini baik secara praktis maupun sebagai teoritis.

BAB II

LANDASAN TEORI

2.1 Sistem Pendukung Keputusan

Sistem Pendukung Keputusan merupakan Sistem berbasis komputer yang interaktif, yang membantu pengambil keputusan dengan memanfaatkan data dan model untuk menyelesaikan masalah-masalah yang tak terstruktur (Surbakti, 2002). Ada yang mendefinisikan bahwa sistem pendukung keputusan merupakan suatu pendekatan untuk mendukung pengambilan keputusan. Sistem pendukung keputusan menggunakan data, memberikan antarmuka pengguna yang mudah, dan dapat menggabungkan pemikiran pengambil keputusan (Turban, 2005).

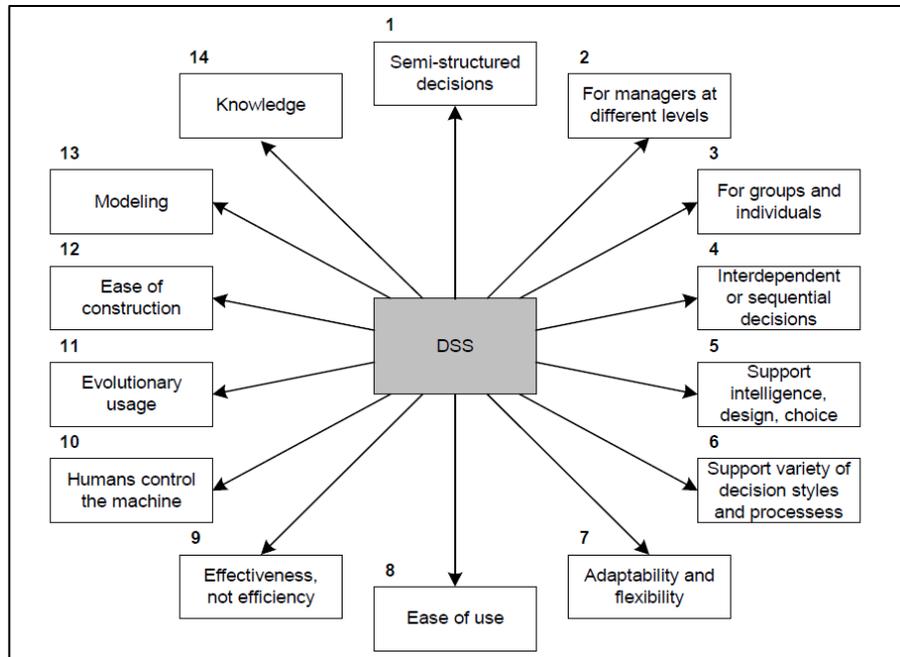
Sistem Pendukung Keputusan (SPK) atau Decision Support System (DSS) adalah sebuah sistem yang mampu memberikan kemampuan pemecahan masalah maupun kemampuan pengkomunikasian untuk masalah dengan kondisi semi terstruktur dan tak terstruktur. Sistem ini digunakan untuk membantu pengambilan keputusan dalam situasi semi terstruktur dan situasi yang tidak terstruktur, dimana tak seorangpun tahu secara pasti bagaimana keputusan seharusnya dibuat (Turban, 2001).

Sistem Pendukung keputusan bertujuan untuk menyediakan informasi, membimbing, memberikan prediksi serta mengarahkan kepada pengguna informasi agar dapat melakukan pengambilan keputusan dengan lebih baik. Sprague dan Watson mendefinisikan Sistem Pendukung Keputusan (SPK) sebagai sistem yang memiliki lima karakteristik utama yaitu (Sprague et.al, 1993):

1. Sistem yang berbasis komputer.
2. Dipergunakan untuk membantu para pengambil keputusan
3. Untuk memecahkan masalah-masalah rumit yang mustahil dilakukan dengan kalkulasi manual.
3. Melalui cara simulasi yang interaktif.
4. Data dan model analisis sebagai komponen utama.

2.1.1 Karakteristik dan Kemampuan Sistem Pendukung Keputusan (SPK)

Di bawah ini adalah karakteristik dan kemampuan ideal dari suatu SPK (Surbakti, 2002):



Gambar 2 Karakteristik Sistem Pendukung Keputusan

(Surbakti, 2002)

Keterangan :

1. SPK menyediakan dukungan bagi pengambil keputusan utamanya pada situasi semi terstruktur dan tak terstruktur dengan memadukan pertimbangan manusia dan informasi terkomputerisasi. Berbagai masalah tak dapat diselesaikan (atau tak dapat diselesaikan secara memuaskan) oleh sistem terkomputerisasi lain, seperti *Electronic Data Processing* atau *Management Information System*, tidak juga dengan metode atau *tool* kuantitatif standar.
2. Dukungan disediakan untuk berbagai level manajerial yang berbeda, mulai dari pimpinan puncak sampai manajer lapangan.
3. Dukungan disediakan bagi individu dan juga bagi grup. Berbagai masalah organisasional melibatkan pengambilan keputusan dari orang dalam grup. Untuk masalah yang strukturnya lebih sedikit seringkali hanya membutuhkan

keterlibatan beberapa individu dari departemen dan level organisasi yang berbeda.

4. SPK menyediakan dukungan ke berbagai keputusan yang berurutan atau saling berkaitan.
5. SPK mendukung berbagai fase proses pengambilan keputusan: *intelligence*, *design*, *choice* dan *implementation*.
6. SPK mendukung berbagai proses pengambilan keputusan dan *style* yang berbeda-beda; ada kesesuaian diantara SPK dan atribut pengambil keputusan individu (contohnya *vocabulary* dan *style* keputusan).
7. SPK selalu bisa beradaptasi sepanjang masa. Pengambil keputusan harus reaktif, mampu mengatasi perubahan kondisi secepatnya dan beradaptasi untuk membuat SPK selalu bisa menangani perubahan ini. SPK adalah fleksibel, sehingga *user* dapat menambahkan, menghapus, mengkombinasikan, mengubah, atau mengatur kembali elemenelemen dasar (menyediakan respon cepat pada situasi yang tak diharapkan). Kemampuan ini memberikan analisis yang tepat waktu dan cepat setiap saat.
8. SPK mudah untuk digunakan. *User* harus merasa nyaman dengan sistem ini. *User-friendliness*, fleksibilitas, dukungan grafis terbaik, dan antarmuka bahasa yang sesuai dengan bahasa manusia dapat meningkatkan efektivitas SPK. Kemudahan penggunaan ini diimplikasikan pada mode yang interaktif.
9. SPK mencoba untuk meningkatkan efektivitas dari pengambilan keputusan (akurasi, jangka waktu, kualitas), lebih daripada efisiensi yang bisa diperoleh (biaya membuat keputusan, termasuk biaya penggunaan komputer).
10. Pengambil keputusan memiliki kontrol menyeluruh terhadap semua langkah proses pengambilan keputusan dalam menyelesaikan masalah. SPK secara khusus ditujukan untuk mendukung dan tak menggantikan pengambil keputusan. Pengambil keputusan dapat menindaklanjuti rekomendasi komputer sembarang waktu dalam proses dengan tambahan pendapat pribadi atau pun tidak.
11. SPK mengarah pada pembelajaran, yaitu mengarah pada kebutuhan baru dan penyempurnaan sistem, yang mengarah pada pembelajaran tambahan, dan

begitu selanjutnya dalam proses pengembangan dan peningkatan SPK secara berkelanjutan.

12. *User* harus mampu menyusun sendiri sistem yang sederhana. Sistem yang lebih besar dapat dibangun dalam organisasi *user* tadi dengan melibatkan sedikit saja bantuan dari spesialis di bidang *Information Systems (IS)*.
13. SPK biasanya mendayagunakan berbagai model (standar atau sesuai keinginan *user*) dalam menganalisis berbagai keputusan. Kemampuan pemodelan ini menjadikan percobaan yang dilakukan dapat dilakukan pada berbagai konfigurasi yang berbeda. Berbagai percobaan tersebut lebih lanjut akan memberikan pandangan dan pembelajaran baru.
14. SPK dalam tingkat lanjut dilengkapi dengan komponen *knowledge* yang bisa memberikan solusi yang efisien dan efektif dari berbagai masalah yang pelik.

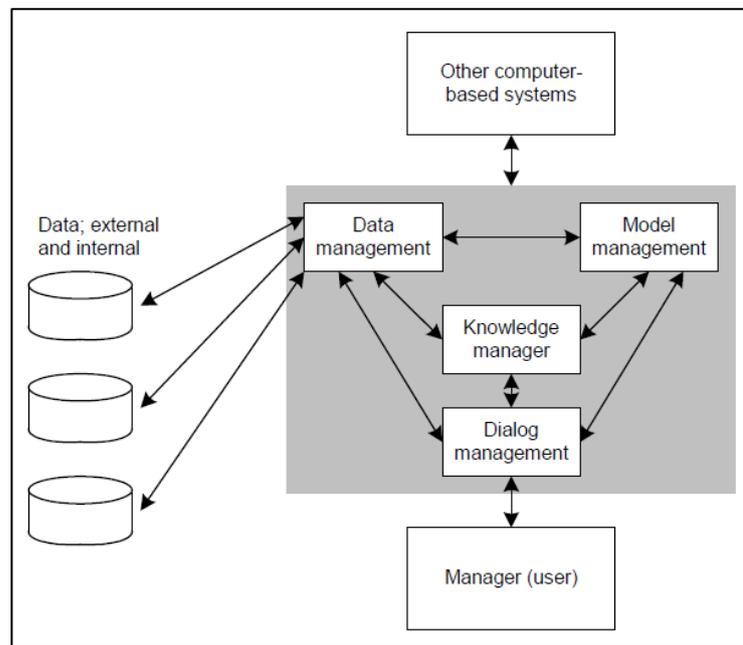
2.1.2 Tahapan Pengambilan Keputusan

Menurut Herbert A. Simon, tahap – tahap yang harus dilalui dalam proses pengambilan keputusan adalah sebagai berikut :

1. Tahap Pemahaman
Tahap ini merupakan proses penelusuran dan pendeteksian dari lingkup problematika serta proses pengenalan masalah. Data masukan diperoleh, diproses dan diuji dalam rangka mengidentifikasi masalah.
2. Tahap Perancangan
Tahap ini merupakan proses pengembangan dan pencarian alternatif tindakan / solusi yang dapat diambil.
3. Tahap Pemilihan
Tahap ini dilakukan pemilihan antara alternatif solusi yang dimunculkan pada tahap perencanaan agar ditentukan/dengan memperhatikan kriteria - kriteria berdasarkan tujuan yang akan dicapai.
4. Tahap Implementasi
Tahap ini dilakukan penerapan terhadap rancangan sistem yang telah dibuat pada tahap perancangan serta pelaksanaan alternatif tindakan yang telah dipilih pada tahap pemilihan.

2.1.3 Komponen Sistem Pendukung Keputusan

Sistem pendukung keputusan memiliki 4 komponen yaitu (Surbakti, 2002) :



Gambar 3 Desain Konseptual Sistem Pendukung Keputusan

(Subakti, 2002)

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 SPK melalui subsistem ini. Ini berarti menyediakan antarmuka.
4. *Knowledge Management*. Subsistem optional ini dapat mendukung subsistem lain atau bertindak sebagai komponen yang berdiri sendiri.

2.1.4 Keuntungan Sistem Pendukung Keputusan

Keuntungan yang akan didapat adalah sebagai berikut (Surbakti, 2002) :

1. Mampu mendukung pencarian solusi dari masalah yang kompleks.
2. Respon cepat pada situasi yang tak diharapkan dalam kondisi yang berubah-ubah.
3. Mampu untuk menerapkan berbagai strategi yang berbeda pada konfigurasi berbeda secara cepat dan tepat.
4. Pandangan dan pembelajaran baru.
5. Memfasilitasi komunikasi.
6. Meningkatkan kontrol manajemen dan kinerja.
7. Menghemat biaya.
8. Keputusannya lebih tepat.
9. Meningkatkan efektivitas manajerial, menjadikan manajer dapat bekerja lebih singkat dan dengan sedikit usaha.
10. Meningkatkan produktivitas analisis.

2.2 Algoritma Filter Based Retrieval

Algoritma *Filter Based Retrieval* merupakan algoritma SPK yang memperoleh keinginan pengguna melalui pengisian form. Proses *retrieval* dan *filtering* terjadi apabila item sesuai dengan keinginan pengguna. Jika set pengambilan cukup kecil, maka item akan ditampilkan kepada pengguna. Jika set pengambilan terlalu besar atau pengguna tidak menemukan item yang diinginkan, sistem akan menyajikan lagi formulir untuk memodifikasi kebutuhan pengguna. Bentuk ini memiliki beberapa nilai awal dan label kustom.

Filter Based Retrieval secara umum artinya pengambilan berdasarkan penyaringan, terdapat beberapa tahapan yang harus dilalui yaitu;

- Mengkonfigurasi aplikasi (kasus dasar)
- Sekuen 1

Merupakan permintaan kasus dan mendefinisikannya sebagai deskripsi dari masalah / kasus. Kasus yang disusun oleh deskripsi, solusi, pembenaran

solusi dan hasilnya, sehingga query hanya deskripsi bagian dari kasus. Ini adalah: kasus tanpa solusi atau hasil.

- Sekuen 2
Sebuah deskripsi (diwarisi dari data kasus), sebuah solusi, sebuah pembenaran Solusi, Hasilnya.
- Sekuen 3
Objek yang merangkum jawaban pengguna ketika kasus yang akan ditampilkan. Objek ini menyimpan integer internal nilai yang memungkinkan:
 - QUIT
 - REFINE QUERY
 - BUY

Hal ini juga berisi kasus yang dipilih dari daftar.

Jika jawabannya adalah BUY, kasus yang dipilih adalah hasil akhir.

Jika jawabannya adalah REFINE QUERY (ubah query), kasus yang dipilih dapat digunakan pada Navigasi dengan Mengusulkan untuk memperoleh query.

2.2.1 *Obtain query from method*

Menunjukkan dari dan untuk memperoleh query. Metode kelas ini memungkinkan untuk menggunakan nilai default menyembunyikan beberapa atribut dan menentukan label ditampilkan dengan setiap atribut.

- Mendapatkan Query Tanpa Nilai Awal
Memperoleh query tanpa menunjukkan nilai awal. Ini menunjukkan setiap atribut dan menggunakan nama atribut sebagai label. parameter: permintaan - untuk mendapatkan.
- Mendapatkan Query Dengan Nilai Awal
Memperoleh query yang menunjukkan nilai awal dari objek permintaan diterima. Ini menunjukkan setiap atribut dan menggunakan nama sebagai label. parameter: permintaan - dengan nilai awal dan di mana nilai-nilai baru pengguna disimpan.

Memperoleh query menunjukkan bentuk permintaan untuk mengisi dengan nilai yang diperoleh dan mengandung nilai awal menggunakan nilai Query menunjukkan jika nilai permintaan ditampilkan secara defaultAtribut tersembunyi adalah daftar tidak ditampilkan atribut label untuk setiap atribut. Jika tidak ada label untuk atribut,nama atribut yang digunakan.

Menambahkan Editor Parameter ke panel untuk setiap atribut dalam Komponen Kasus cc Kasus Komponen dengan panel atribut untuk menambahkan Parameter Editor Atribut tersembunyi atribut tidak ditampilkan label untuk atribut

2.2.2 *Filtering Proses*

Proses ini merupakan proses utama dalam Algoritma *Filter Based Retrieval* yang terdiri dari beberapa proses pendukung lainnya yaitu *Filter Predicate*, *Filter Based Retrieval Method*, *Filter Config*, *Filter Case*

2.2.2.1 *Filter Predicate*

- Mengambil kasus menurut predikat *boolean* (*equal*, *equal to*, *not equal*, *not equal to*, *query less*, *query less or equal*, *query more*, *query more or equal*) dari atribut.

➤ *Equal*

```
public class Equal implements FilterPredicate
{
    public boolean compute(Object caseObject, Object
queryObject) throws NoApplicableFilterPredicateException
    {
        if((caseObject == null)&&(queryObject==null))
            return true;
        else if(caseObject == null)
            return false;
        else if(queryObject == null)
            return true;
        else
            return caseObject.equals(queryObject);
    }
}
```

Predikat yang membandingkan jika dua objek yang sama.

➤ Not Equal

```
public class NotEqual implements FilterPredicate
{
    public boolean compute(Object caseObject, Object
queryObject) throws NoApplicableFilterPredicateException
    {
        if((caseObject == null)&&(queryObject==null))
            return false;
        else if(caseObject == null)
            return false;
        else if(queryObject == null)
            return true;
        else
            return !caseObject.equals(queryObject);
    }
}
```

Predikat yang membandingkan jika dua objek yang berbeda.

➤ *Query Less or Equal*

```
public class QueryLessOrEqual implements FilterPredicate
{
    public boolean compute(Object caseObject, Object
queryObject) throws NoApplicableFilterPredicateException
    {
        if((caseObject == null)&&(queryObject==null))
            return true;
        else if(caseObject == null)
            return false;
        else if(queryObject == null)
            return true;
        else if (! ((caseObject instanceof
java.lang.Number)|| (caseObject instanceof Enum)))
            throw new
jcolibri.exception.NoApplicableFilterPredicateException(
this.getClass(), caseObject.getClass());
        else if (! ((queryObject instanceof
java.lang.Number)|| (queryObject instanceof Enum)))
            throw new
jcolibri.exception.NoApplicableFilterPredicateException(
this.getClass(), queryObject.getClass());
        else
        {
            double caseValue;
            double queryValue;
            if((caseObject instanceof
Number)&&(queryObject instanceof Number))
            {
                Number n1 = (Number) caseObject;
                caseValue = n1.doubleValue();
                Number n2 = (Number) queryObject;
                queryValue = n2.doubleValue();
            }
            else
            {
                Enum enum1 = (Enum)caseObject;
                caseValue = enum1.ordinal();
                Enum enum2 = (Enum)queryObject;
                queryValue = enum2.ordinal();
            }
            return queryValue<=caseValue;
        }
    }
}
```

➤ *Query More or Equal*

```
public class QueryMoreOrEqual implements FilterPredicate
{
    public boolean compute(Object caseObject, Object
queryObject) throws NoApplicableFilterPredicateException
    {
        if((caseObject == null)&&(queryObject==null))
            return true;
        else if(caseObject == null)
            return false;
        else if(queryObject == null)
            return true;
        else if (! ((caseObject instanceof
java.lang.Number)|| (caseObject instanceof Enum)))
            throw new
jcolibri.exception.NoApplicableFilterPredicateException(
this.getClass(), caseObject.getClass());
        else if (! ((queryObject instanceof
java.lang.Number)|| (queryObject instanceof Enum)))
            throw new
jcolibri.exception.NoApplicableFilterPredicateException(
this.getClass(), queryObject.getClass());
        else
        {
            double caseValue;
            double queryValue;
            if((caseObject instanceof
Number)&&(queryObject instanceof Number))
            {
                Number n1 = (Number) caseObject;
                caseValue = n1.doubleValue();
                Number n2 = (Number) queryObject;
                queryValue = n2.doubleValue();
            }
            else
            {
                Enum enum1 = (Enum)caseObject;
                caseValue = enum1.ordinal();
                Enum enum2 = (Enum)queryObject;
                queryValue = enum2.ordinal();
            }
            return queryValue>=caseValue;
        }
    }
}
```

➤ *Filter Predicate*

```
public interface FilterPredicate
{
    public boolean compute(Object caseObject, Object
queryObject) throws
jcolibri.exception.NoApplicableFilterPredicateException;
}
```

2.2.2.2 Filter Config

Konfigurasi objek untuk Metode Ini berisi Peta dengan pasangan.

```
public class FilterConfig
{
    private HashMap<Attribute,FilterPredicate> config;

    public FilterConfig()
    {
        config = new HashMap<Attribute,FilterPredicate>();
    }

    public void addPredicate(Attribute attribute, FilterPredicate
predicate)
    {
        config.put(attribute, predicate);
    }

    public FilterPredicate getPredicate(Attribute attribute)
    {
        return config.get(attribute);
    }

    public Collection<Attribute> getDefinedAttributes()
    {
        return config.keySet();
    }
}
```

- Menambahkan predikat baru, parameter:
atribut - yang predikat terkait untuk
predikat - untuk atribut
- Mendapat predikat untuk atribut tertentu, parameter:
atribut - dari berpredikat
- Mengembalikan daftar atribut yang memiliki predikat yang didefinisikan.

2.2.2.3 *Filtercase*

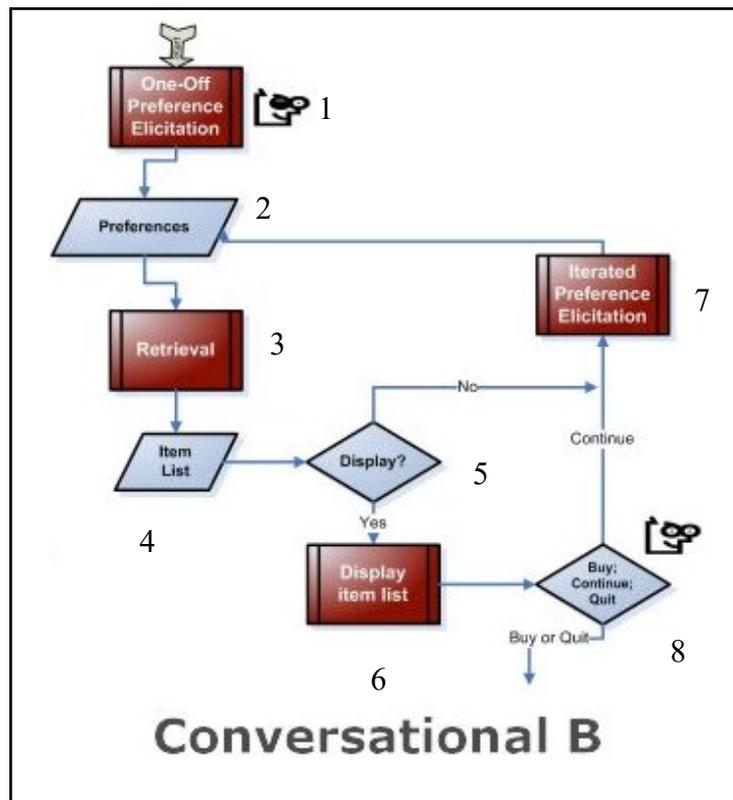
Mengambil kasus yang sesuai dengan predikat yang pengaturan filter,
parameter:

Kasus - untuk mengambil dari.

Permintaan - untuk membandingkan.

FilterConfig - mengandung predikat penyaring.

2.2.3 Conversational B



Gambar 4 Alur Proses Conversational B

(sumber : framework jCOLIBRI 2.1)

Seperti yang ditunjukkan pada Gambar 5 dijelaskan alur proses *data retrieval*. Berikut penjelasannya :

1. *State* 1 merupakan satu kasus dimana user melakukan pengisian form dan menentukan 1 pilihan.
2. *State* ini proses pemilihan data.
3. *State* ini merupakan proses retrieval(pengambilan kembali data).
4. *State* ini merupakan proses pemilihan daftar perumahan (filtering).
5. Decision making *state* memungkinkan sistem untuk melanjutkan proses yaitu dengan melihat list data atau kembali proses 2 (looping) sesuai dengan tindakan user.
6. *State* ini menampilkan list perumahan yang sudah melewati proses filtering.

7. *State* ini merupakan proses iterasi atau looping (pengulangan) membuat proses kembali ke *state* 2 yaitu proses pemilihan data.
8. Decision makin *state*, melakukan pembelian atau tidak.

2.3 Perumahan

Menurut Kamus Besar Bahasa Indonesia, perumahan adalah kumpulan dari rumah-rumah yang digunakan untuk berlindung bagi keluarga yang layak huni (dilengkapi dengan sarana dan prasarana). Menurut Sinulingga, 1999 pemukiman pada garis besarnya terdiri dari berbagai komponen (Sianturi, 2006):

1. Lahan atau tanah yang diperuntukkan untuk pemukiman itu, dimana kondisi tanah akan mempengaruhi harga satuan rumah yang akan dibangun diatas lahan itu.
 2. Prasarana pemukiman, yaitu jalan lokal, saluran *drainase*, saluran air kotor, saluran air bersih, serta jaringan listrik yang semuanya juga menentukan kualitas pemukiman yang akan dibangun.
 3. Perumahan (tempat tinggal) yang akan dibangun.
 4. Fasilitas umum dan fasilitas sosial yaitu fasilitas pendidikan, kesehatan, peribadatan, lapangan bermain, dan lain-lain dalam lingkungan pemukiman itu.
- Menurut Gallian Artur B. dan Eisner Simon, 1999 suatu pemukiman disebut “baik”, jika memenuhi ketentuan berikut (Sianturi, 2006) :

1. Lokasinya sedemikian rupa, sehinggal tidak terganggu oleh kegiatan lain seperti pabrik, yang umumnya dapat memberikan dampak pencemaran udara atau pencemaran lainnya.
2. Mempunyai akses terhadap pusat-pusat pelayanan, seperti pelayanan pendidikan, kesehatan, dan perdagangan.
3. Mempunyai fasilitas *drainase*, yang dapat mengalirkan air hujan dengan cepat dan tidak menimbulkan genangan air walaupun hujan.
4. Mempunyai fasilitas penyediaan air bersih, berupa jaringan distribusi yang siap disalurkan ke tiap rumah.

5. Dilengkapi dengan fasilitas pembuangan air kotor yang dapat dibuat dengan sistem individual yakni tangki septik dan lapangan rembesan atau tangki septik komunal.
6. Pemukiman harus dilayani dengan fasilitas pembuangan sampah secara teratur agar lingkungan tetap nyaman.
7. Dilengkapi dengan fasilitas umum seperti taman bermain bagi anak-anak, lapangan atau taman, tempat ibadah, pendidikan, dan kesehatan sesuai dengan skala besarnya pemukiman.
8. Dilayani oleh jaringan listrik dan telepon.

2.4 Kota Bekasi

Kota Bekasi merupakan salah satu kota yang terdapat di provinsi Jawa Barat, Indonesia. Nama Bekasi berasal dari kata bagasasi yang artinya sama dengan candrabaga yang tertulis dalam Prasasti Tugu era Kerajaan Tarumanegara, yaitu nama sungai yang melewati kota ini. Kota ini sekarang berada dalam lingkungan megapolitan Jabodetabek dan menjadi kota besar ke empat di Indonesia. Saat ini Kota Bekasi berkembang menjadi tempat tinggal kaum urban dan sentra industri, kota Bekasi juga dijuluki sebagai Kota Patriot dan Kota Pejuang.

Kota Bekasi dipilih menjadi objek kota yang diteliti karena kota Bekasi merupakan salah satu daerah peyanggah kota Jakarta yang mana adalah Ibu Kota Negara, yang juga merupakan pusat pemerintahan dan perekonomian dimana banyak orang yang datang untuk mencari pekerjaan dan bahkan tinggal, tetapi seiring berjalannya waktu kota Jakarta tidaklah mampu menampung begitu banyak orang yang ingin tinggal dan menetap disana, maka penulis memilih kota Bekasi sebagai objek penelitian dalam menentukan rumah dan lokasi tempat tinggal.

2.5 Penalaran Berbasis Kasus

Perkembangan sistem pakar telah demikian pesatnya karena kebutuhan masyarakat akan pengambilan keputusan yang cepat dan tepat terhadap suatu permasalahan. Dalam permasalahan tertentu, seringkali pengambilan keputusan tidak dapat dilakukan oleh seorang pakar, tetapi perlu memperhatikan kasus-kasus

yang pernah terjadi. Permasalahan yang sering muncul adalah ketersediaan pakar yang memiliki pengetahuan tertentu, seperti dokter spesialis tertentu, tidak mudah diperoleh, atau waktunya terbatas, padahal pasien ingin mengetahui penyakit yang dideritanya, dan perlu penanganan yang segera. Dengan pelajari kasus-kasus yang pernah terjadi, dimungkinkan seseorang dapat mempunyai gambaran suatu kesimpulan yang dapat diperoleh, misalkan suatu penyakit tertentu yang mungkin diderita pasien tersebut, tanpa harus menunggu lama untuk bertemu dengan dokter spesialis.

Dalam bidang meteorologi, misalnya, dengan mempelajari fenomenafenomena alam dan kejadian-kejadian yang pernah terjadi, maka dapat diperkirakan fenomena yang akan terjadi jika dijumpai fenomena alam yang mirip dengan kasus-kasus sebelumnya. Berdasarkan persoalan tersebut, munculah konsep sistem pakar menggunakan penalaran berbasis kasus.

Sistem ini bukan sebagai pengganti sistem pakar berbasis pengetahuan yang telah berkembang sebelumnya yang mayoritas merupakan sistem pakar berbasis aturan yang sulit untuk dibangun, tetapi sebagai pelengkap dan memperkuat sistempakar tersebut. Sistem ini menghasilkan kesimpulan berdasarkan kemiripan terhadap kasus-kasus yang pernah terjadi. Pada penalaran berbasis kasus, suatu basis kasus berisi kasus-kasus dengan solusi yang telah dicapai. Untuk menemukan solusi dari sebuah kasus baru yang diberikan, sistem akan mencari kasus-kasus dalam basis kasus yang memiliki tingkat kesamaan yang paling tinggi. Sebuah evaluasi terhadap perhitungan kesamaan kasus merupakan hal yang penting dalam penalaran berbasis kasus.

Penalaran berbasis kasus telah menjadi teknik yang sukses untuk sistem berbasis pengetahuan dalam banyak domain. Penalaran berbasis kasus berarti menggunakan pengalaman sebelumnya dalam kasus yang mirip untuk memahami dan memecahkan permasalahan baru. Penalaran berbasis kasus mengumpulkan kasus sebelumnya yang hampir sama dengan masalah yang baru dan berusaha untuk memodifikasi solusi agar sesuai dengan kasus yang baru (Aamodt dan Plaza,1994 Ide dasar dari penalaran berbasis kasus adalah asumsi bahwa permasalahan yang

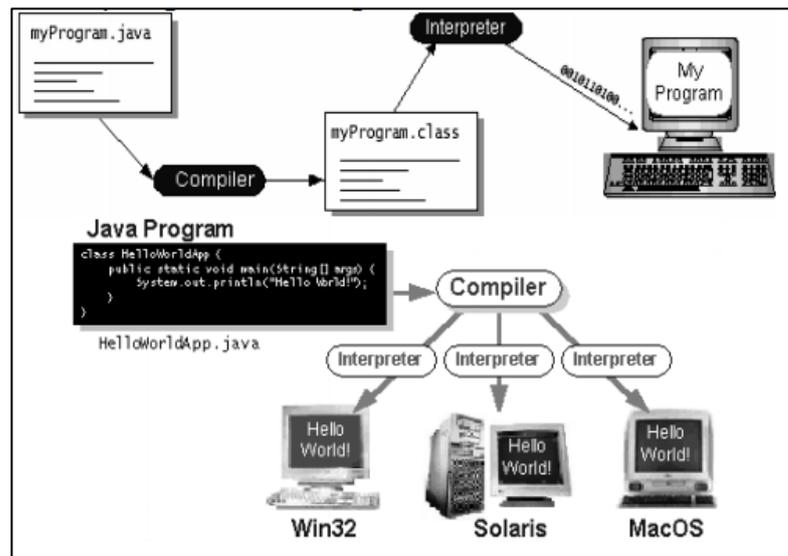
serupa mempunyai solusi serupa. Meskipun asumsi ini tidaklah selalu benar, hal ini tergantung pada banyaknya domain praktis.

2.6 Bahasa Pemrograman Java

Java adalah bahasa pemrograman berorientasi objek murni yang dibuat berdasarkan kemampuan-kemampuan terbaik bahasa pemrograman objek sebelumnya (C++, Ada, dan Simula). Java diciptakan oleh James Gosling, *developer* dari Sun Microsystems pada tahun 1991. Karakteristik yang dimiliki Java antara lain:

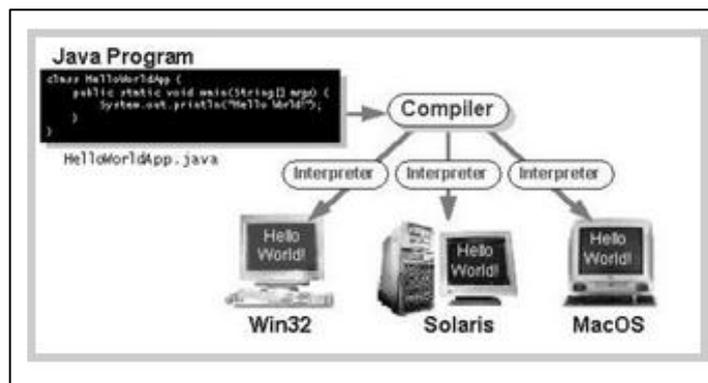
1. Sederhana (Simple)
2. Berorientasi Objek (Object Oriented)
3. Terdistribusi (Distributed)
4. Interpreted
5. Robust - Aman (Secure)
6. *Architecture Neutral – Portable*
7. *Performance*
8. Multithreaded
9. Dinamis

Kode pada bahasa pemrograman Java ditulis dalam file teks dan disimpan dengan nama file yang sama persis dengan nama.classnya serta diberikan ekstensi (*.java). File tersebut dikompilasi dengan compiler java (javac) sehingga akan menghasilkan 1 file dengan ekstensi class (*.class). file berekstensi class (*.class) yang akan dijalankan diatas Java Mechanic Virtual (JVM) dengan menggunakan java menggunakan java launcher tool (java). Skema kinerja dari bahasa pemrograman Java dijelaskan pada gambar 4 :



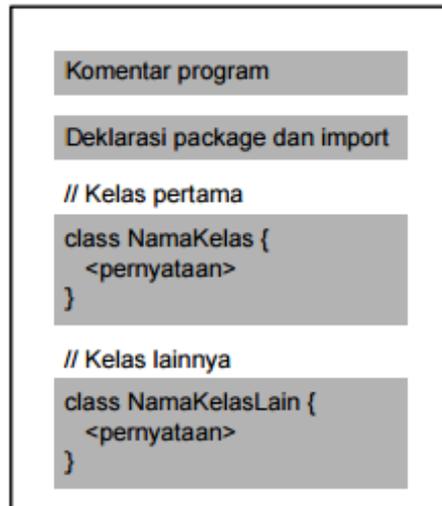
Gambar 5 Skema Kerja Bahasa Pemrograman Java

Adanya Java Virtual Mevhanic (JVM) program java yang telah dibuat dapat dijalankan pada berbagai komputer dengan arsitektur dan sistem operasi yang berbeda seperti dijelaskan pada gambar 5



Gambar 6 Skema Kerja Bahasa Pemrograman Java

Aplikasi-aplikasi pada java antara lain : stand alone, applets, aplikasi berbasis windows (GUI), java servlet - java server pages (JSP), java beans dan enterprise java beans, dan java micro edition. Contoh penulisan program pada java dengan model struktur program Java :



Gambar 7 Penulisan Program Java

Dapat ditulis dalam satu file .java:

- Berisi satu kelas
- Berisi beberapa kelas

Ditulis dalam beberapa file .java:

- Satu file satu kelas
- Satu file banyak kelas

Hanya boleh ada satu fungsi utama pada setiap program objek.

Kompilasi :

`javac <namafile.java>`

Contoh :

`javac HelloWorld.java`

Eksekusi :

`java <namafile hasil kompilasi> [<argumen baris perintah>]`

Contoh : `java HelloWorld`

2.6.1 Kata Kunci Simpanan

Kata kunci simpanan adalah *identifier* khusus yang disimpan oleh bahasa Java untuk mengendalikan bagaimana program didefinisikan. Kata kunci ini digunakan untuk mengenali tipe-tipe, pengubah, dan mekanisme pengaturan aliran program. Kata kunci ini hanya dapat digunakan untuk fungsi tertentu dan tidak dapat digunakan sebagai *identifier* nama suatu variabel, class dan method. Seperti terlihat dalam tabel 7:

Tabel 2 Kata kunci simpanan

abstract	boolean	break	byte	byvalue	case
cast	catch	char	class	const	continue
default	do	double	else	extends	false
final	finally	float	for	future	generic
goto	if	impleme nts	import	inner	instanceof
int	interface	long	native	new	null
operator	outer	package	private	protected	public
rest	return	short	static	super	switch
synchronized	this	throw	throws	transient	true
try	var	void	volatile	while	

2.7 Eclipse

Menurut wikipedia, Eclipse adalah sebuah IDE (Integrated Development Environment) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua platform (platform-independent). Jadi dengan *software* eclipse kita dapat mendvelop atau membuat sebuah program tidak hanya di android, kita bisa membuat program2 lain semisal berplatform java atau yang lainnya. Berikut ini adalah sifat dari Eclipse:

1. *Multi-platform*: Target sistem operasi Eclipse adalah Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X.
2. *Multi-language*: Eclipse dikembangkan dengan bahasa pemrograman Java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti C/C++, Cobol, Python, Perl, PHP, dan lain sebagainya.
3. *Multi-role*: Selain sebagai IDE untuk pengembangan aplikasi, Eclipse pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat

lunak, seperti dokumentasi, test perangkat lunak, pengembangan web, dan lain sebagainya.

Eclipse awalnya dikembangkan oleh IBM untuk menggantikan perangkat lunak IBM Visual Age for Java 4.0. Produk ini diluncurkan oleh IBM pada tanggal 5 November 2001, yang menginvestasikan sebanyak US\$ 40 juta untuk pengembangannya. Semenjak itu konsorsium Eclipse Foundation mengambil alih untuk pengembangan Eclipse lebih lanjut dan pengaturan organisasinya. Pada saat ini eclipse merupakan salah satu IDE favorit dikarenakan gratis dan open source, yang berarti setiap orang boleh melihat kode pemrograman perangkat lunak ini. Selain itu, kelebihan dari Eclipse yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan komponen yang dinamakan plug-in. Sampai saat sekarang ini Eclipse sudah mencapai versi 3.6 yang diberi nama Helios.

BAB III

ANALISIS DAN PERANCANGAN

Pada bab ini akan dibahas mengenai analisis filtering Algoritma *Filter Based Retrieval*. Setelah itu dilakukanlah perancangan aplikasi menggunakan metode *prototype* sebagai permodelan aplikasi seperti yang telah di paparkan pada sub Bab 1.5. Sehingga setelah dilakukan analisis dan perancangan maka dapat terlihat alur proses kerja aplikasi dan tahapan yang terjadi pada aplikasi.

3.1 Analisis

Pada perancangan aplikasi ini dilakukan tahap analisis proses kerja algoritma dan desain aplikasi. Analisis proses kerja algoritma dilakukan dengan cara menjelaskan setiap proses yang terjadi saat melakukan *obtaining query*, *filtering*, *retrieving*, *conversation* yang dilakukan oleh Algoritma *Filter Based Retrieval*. Setelah mengetahui proses setiap tahapan yang diatas maka dirancang sebuah aplikasi yang menjelaskan cara kerja dari sistem yang mengimplementasikan Algoritma *Filter Based Retrieval*.

3.1.1 Cara Kerja Sistem

3.1.2 Analisis Perencanaan Sistem

Berikut adalah output perencanaan dalam pembangunan aplikasi Sistem Rekomendasi Pembelian Rumah berdasarkan fase-fase yang terdapat pada metodologi *prototype* :

Tabel 3 Analisa Perencanaan Sistem

No	Fase	Tahap Perancangan
1	Analisis	<ol style="list-style-type: none">a. Melakukan analisis permasalahan observasi penelitian.b. Melakukan analisis kebutuhan sistem dan fungsionalitas sistem.c. Melakukan analisis mengenai skema kerja sistem.
2	<i>Design</i>	<ol style="list-style-type: none">a. Menjelaskan proses perancangan dengan <i>flowchart</i>.

		b. Menentukan design permodelan <i>layout prototype</i> .
3	Pengembangan <i>Prototype</i>	a. Mengembangkan bentuk <i>prototype</i> berdasarkan struktur model hasil tahapan <i>design</i> ke dalam bentuk bahasa pemrograman.
4	Pengujian	a. Melakukan bentuk pengujian dengan menggunakan tabel uji.
5	Evaluasi	a. Melakukan evaluasi dan penyempurnaan pada sistem secara keseluruhan.

3.1.3 Analisis Kebutuhan Sistem

Untuk mencapai tujuan penelitian dalam membangun sebuah aplikasi dilakukan analisis sistem yang terdiri dari analisis kebutuhan sistem dan analisis fitur sistem untuk memahami apa saja yang dibutuhkan dan yang harus dicapai dalam setiap tahapan-tahapan proses. Dalam membangun aplikasi identifikasi sidik jari, terdapat beberapa kebutuhan sistem yang digunakan. Kebutuhan sistem ini akan dijelaskan dalam dua bagian, yakni kebutuhan *hardware* dan *software*, masing-masing dari kebutuhan tersebut akan dijelaskan sesuai fungsinya masing-masing.

1. Kebutuhan *Hardware*

Berikut ini adalah beberapa perangkat keras (*Hardware*) yang mendukung pembangunan aplikasi :

- Processor Intel Core i3 2.2 GHz
- Memory Internal DDR3 4096 MB
- Internal Storage HDD 640GB
- RADEON GRAPHICS HD 6730 2 GB

2. Kebutuhan *Software*

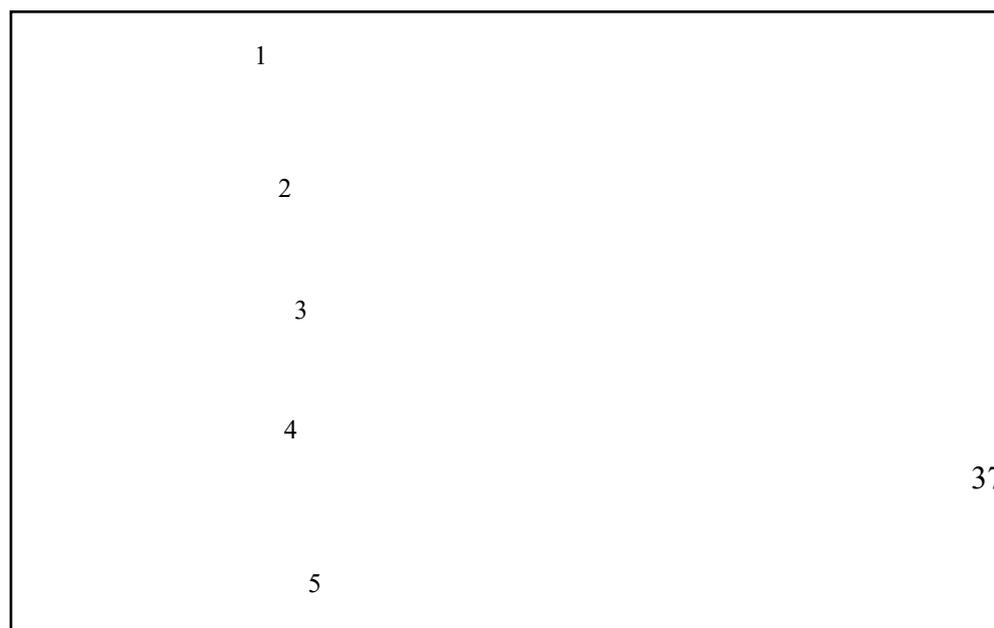
Spesifikasi perangkat lunak yang digunakan dalam pengembangan aplikasi Sistem Rekomendasi Pembelian Rumah ini adalah sebagai berikut :

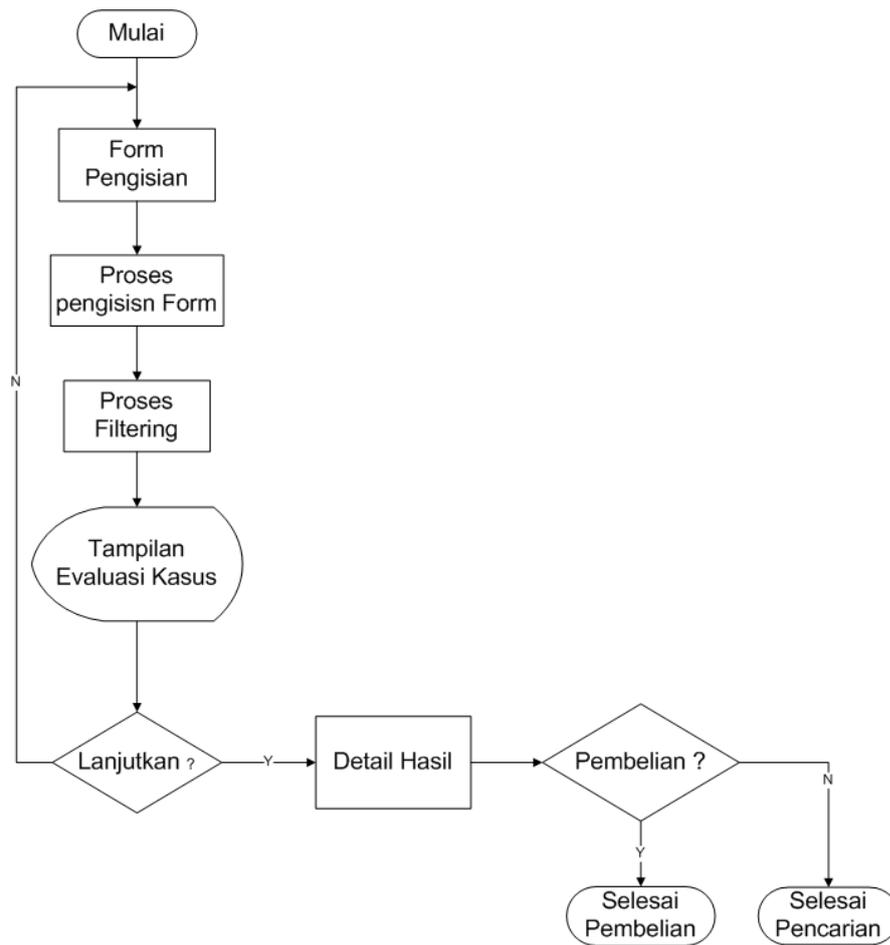
- Sistem Operasi Windows 7

- *Java SDK 1.8 (Java Standard Development Kit)*
- *Java Compiler*
- *Java Debugger*
- Eclipse
- jCOLIBRI2

3.2 Analisis Sistem Pendukung Keputusan Pembelian Rumah

Perancangan sistem pendukung keputusan pembelian rumah dikembangkan melalui *sample* data rumah dijual yang telah disusun dengan fitur – fitur yang menjadi standar informasi *user* dalam pembelian rumah. Pengembangan aplikasi sistem rekomendasi pembelian rumah membutuhkan data tentang model dan fitur-fitur rumah yang akan dijual di wilayah Kota Bekasi. Data yang dibutuhkan terkumpul melalui tahapan observasi data rumah dijual yang informasinya dapat ditemui di beberapa situs online milik industri properti. Selain itu, dibutuhkan juga sebuah data hasil wawancara terhadap beberapa orang koresponden untuk menentukan tingkatan fitur – fitur yang menjadi pilihan konsumen ketika ingin membeli rumah. Fitur – fitur yang memiliki persentase tertinggi akan melalui tahap penilaian yang akan digunakan pada proses *filtering* dengan basis kasus agar sistem dapat menampilkan hasil terbaik. Untuk menggambarkan langkah- langkah dan urutan prosedur dari aplikasi sistem rekomendasi pembelian rumah maka dibuat sebuah *flowchart* atau diagram alir secara grafik dari langkah-langkah dan urutan prosedur dari aplikasi. Berikut adalah *flowchart* cara kerja sistem rekomendasi pembelian rumah yang dijelaskan pada gambar 6.





Gambar 8 Flowchart Sistem Pendukung Keputusan Pembelian Rumah

Seperti yang ditunjukkan pada Gambar 9 dijelaskan *flowchart* arsitektur sistem. Berikut penjelasan langkah pada *flowchart* arsitektur sistem rekomendasi pembelian rumah :

1. Merupakan *state* awal dari aplikasi.
2. Pada *state* ini disediakan form pemilihan fitur-fitur rumah.
3. Pada *state* ini dilakukan pengisian form yang disediakan.

4. Pada *state* ini sistem akan melakukan penyaringan data atau query sesuai dengan yang isi melalui form sebelumnya sesuai dengan predikat *boolean*, *filter config* dan *filter case* yang sudah ditetapkan.
5. Pada *state* ini sistem menampilkan evaluasi hasil dari proses *filtering*, jika hasil tidak didapat atau tidak ada yang sesuai maka proses pengisian form harus dilakukan kembali.
6. Pada *state* ini yaitu *decision* menampilkan semua proses *filtering* yang telah dilakukan, jika tidak sesuai maka akan dilakukan pengulangan *state* ke 3, untuk meminta user mengisi kembali form.
7. *State* ini menampilkan data rekomendasi rumah dari hasil *filtering* yang menandakan terdapat kecocokan hasil dengan apa yang diinginkan user.
8. *State decision* ini menampilkan semua hasil data yang cocok dan hanya tinggal menunggu keputusan user untuk membeli atau tidaknya rekomendasi rumah yang sudah dihasilkan.
9. Pada *state* ini merupakan akhir dari aplikasi dengan keputusan user menyetujui dan melakukan proses pembelian.
10. Pada *state* ini juga merupakan akhir dari aplikasi tetapi dengan keputusan akhir user yang tidak melakukan proses pembelian rumah.

3.2.1 Tabel Sample Data Rumah Dijual di Wilayah Kota Bekasi

Tabel sample data rumah dijual di wilayah kota Bandung merupakan data *real* tentang rumah-rumah yang akan dijual lengkap dengan informasi detail harga, luas bangunan, jumlah kamar, kamar mandi, tipe rumah, fasilitas furnitur, dan daerah rumah yang dijual. Sample data didapatkan dari informasi rumah dijual yang dipromosikan melalui website www.urbanindo.com, www.rumahdijual.com, www.rumah.com, www.rumah123.com. Website tersebut merupakan website yang menyajikan informasi rumah dijual secara online, dimana setiap data rumah yang akan dijual memiliki informasi yang akurat dan lengkap. Tabel sample data terlampir di halaman lampiran.

Pada penelitian ini sample data rumah yang diambil dibatasi 4 – 5 data perumahan dari tiap wilayah. Data rumah dijual di setiap daerah yang dijadikan basis kasus terlampir pada halaman lampiran.

3.2.2 Tabel Hasil Uji

Tabel hasil uji adalah kumpulan data dari hasil wawancara terhadap dua puluh orang koresponden dimana hasil wawancara tersebut akan digunakan untuk penentuan terhadap fitur-fitur standarisasi konsumen dalam pembelian rumah. Fitur-fitur tersebut adalah daerah, harga, intensitas banjir, luas tanah, jumlah lantai, jumlah kamar, jumlah kamar mandi, furnitur, luas bangunan, tipe rumah pada rumah. Dari 10 fitur yang diberikan, koresponden harus memberikan nilai persentase untuk 8 fitur, dimana 8 fitur tersebut merupakan fitur-fitur yang akan menjadi variabel penentu dalam proses penghitungan rekomendasi. Data hasil wawancara dijelaskan pada tabel 4.

Data hasil wawancara diambil dari 10 orang perempuan dan 10 orang laki-laki yang memiliki rencana membeli sebuah rumah di wilayah Kota Bekasi. Semua koresponden merupakan seseorang yang telah memiliki penghasilan tetap yang dianggap bisa mewakili calon pembeli lainnya. Dari hasil wawancara dapat dilihat persentase dua penialian terendah dimiliki oleh fitur luas bangunan dan tipe rumah. Sehingga dapat disimpulkan bahwa 8 fitur teratas yang menjadi standarisasi koresponden dalam pemilihan rumah terletak pada fitur daerah, harga, intensitas banjir, luas tanah, jumlah lantai, jumlah kamar, jumlah kamar mandi, furnitur. Dari total 8 fitur yang telah diteliti dari hasil wawancara, masing –masing fitur memiliki persentase yang berbeda sehingga penilaian untuk masing-masing fitur juga akan berbeda.

Tabel 4 Data Hasil Wawancara

Nama Koresponden	Penilaian Pilihan Hasil Wawancara									
	Daerah	Harga	Intensitas Banjir	Luas Tanah	Jumlah Lantai	Minimal Kamar	Minimal Kamar Mandi	Furnitur	Luas Bangunan	Tipe Rumah
Ayu Lestari	SP	P	SP	SP	P	P		P	P	
Angga Kurnia	P	SP	SP	C	P	C	P		C	
Amelia Indriani	SP	P	SP	P	C	P	C		C	
Gugi Gustiana Agung	SP	SP	P	SP		SP	P		C	P
Yarra Eka Putra	SP	P	C	SP	C	P		P	P	
Malla Hilma Charisma	P	SP	P	P	P	SP	P	P		
Anis Elistya	P	SP	P	P	P	SP	P			C
Tesara Anggraeni	P	SP	SP	SP	P	C		C		C
Dina Oktavia	P	SP	P	C	SP	SP		C	C	
Evi Sylvia	SP	P	SP	C	P	P	C	P		
Oskar Mahendra	SP	SP	C	P	P	P	C		P	
Darma Setiawan	SP	P	SP	P		C	P	P		C
Desi Susanti Siregar	SP	SP	SP	P	SP	P		P		P
Willy Sie	SP	SP	SP	SP		P	P	C		C
M Windy S	SP	P	C	SP		P	P	P	C	
Melia Piskawati	SP	P	SP	C	SP	C		C	P	
M Iskandar Zulkarnaen	SP	P	P	P	P	SP	P		C	
Ian Van Khaidir	P	SP	SP	SP	SP	P	C	P		
Anisa Windy	P	SP	SP	C	P	C	C	P		
M Bayu Hernowo	P	P	C	P	C	P	P			C

Penentuan bobot prefensi dengan memberikan tingkat kepentingan menggunakan nilai SP(Sangat Penting), P(Penting), C(Cukup).

3.3 Desain Sistem

Pada tahap ini dilakukan analisis terhadap kebutuhan aplikasi berdasarkan skenario. Berikut adalah skenario singkat aplikasi dibangun :

1. Pengguna aplikasi adalah konsumen yang ingin mencari referensi rumah.
2. *User* terlebih dahulu mengisi kolom fitur yang menjadi standarisasi user dalam pembelian rumah.
3. Sistem akan mencocokkan data fitur yang diinginkan *user* dengan data pada basis kasus
4. Informasi yang akan ditampilkan pada aplikasi ini yaitu hasil rekomendasi meliputi : area, harga, luas bangunan, fasilitas furnitur, tipe, jumlah kamar dan jumlah kamar mandi.
5. *User* dapat melakukan perevisian fitur –fitur rumah yang diinginkan apabila data referensi rumah yang ditampilkan tidak sesuai dengan keinginan user.

3.3.1 Perancangan Sistem

Tahapan perancangan sistem yang dilakukan pada penelitian ini menggunakan pemodelan UML (Undefined Model Language). Aplikasi sistem rekomendasi pembelian rumah ini menggunakan *use case* diagram, skenario *use case* diagram, *activity* diagram dan *sequence* diagram untuk menggambar dan menjelaskan alur aplikasi.

3.3.1.1 Workflow Sistem

Daftar Pustaka

- [1] Meiga Andriyanto, Yofina Rizky Safitri. "*Penerapan Metode Simple Additive Weighthing Pada Ssistem Pendukung Keputusan Pemilihan Rumah Bagi Konsumen PT TERRASSIMA*". Palembang: STMIK PalComTech.

- [2] Lalu Arry Tri Laksono, Akuwan Saleh, M. Zen Samsono Hadi. “*Pembuatan Aplikasi Mobile Untuk Rekomendasi Pemilihan Rumah Berbasis Fuzzy Database Model Tahani*”. Surabaya: Institut Teknologi Sepuluh November.
- [3] Adinda Ria Rumondang Veranita, 2015. “*Implementasi Algoritma Nearest Neighbor dan Case Selection Pada Sistem Rekomendasi Pembelian Rumah*”. (Studi kasus : Kota Bandung dan Sekitarnya). Bandung: ITENAS.
- [4] Fanny Aryanti, 2013. “*Pembangunan Aplikasi Travel Recommender Dengan Metode Case Based Reasoning*”
- [5] _____, www.bekasikota.go.id/read/573/daftar-perumahan-di-bekasi
- [6] Aamold A. dan Plaza E., 1994. *Case-based Reasoning : foundation issues, methodological variation and System approach, AI Communication* .
- [7] Belen Diaz Agudo, Ian Watson, 2012. *Case-Base Reasoning Research and Development*.
- [8] Belen Diaz-Agudo, Pedro A. Gonzalez-Calero, Juan A. Recio Garcia, Antonio A. Shancez-Ruiz-Granados, 2007. *Building CBR systems with jCOLIBRI*.
- [9] Juan A. Recio-Garcia, Derek Bridge, 2008. Framework jCOLIBRI 2.1, *Conversational (type B) recommender using form-filling and Filter-Based retrieval*.