

**RANCANG BANGUN APLIKASI MOBILE MESIN Pencari
INFORMASI PENJUALAN RUMAH**

SKRIPSI

LABORATORIUM

PEMROGRAMAN APLIKASI PERANGKAT BERGERAK

Untuk memenuhi sebagian persyaratan mencapai gelar Sarjana Komputer



Disusun Oleh :

ITTO WIRAWAN MULYOTO

NIM. 115060800111045

KEMENTRIAN RISET TEKNOLOGI DAN PENDIDIKAN TINGGI

UNIVERSITAS BRAWIJAYA

PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER

PROGRAM STUDI INFORMATIKA/ILMU KOMPUTER

MALANG

2015

LEMBAR PERSETUJUAN

RANCANG BANGUN APLIKASI MOBILE MESIN PENCARI INFORMASI PENJUALAN RUMAH

SKRIPSI

LABORATORIUM PEMROGRAMAN APLIKASI PERANGKAT BERGERAK

Untuk memenuhi sebagian persyaratan mencapai gelar Sarjana Komputer



Disusun Oleh :

ITTO WIRAWAN MULYOTO

NIM. 115060800111045

Skrripsi ini telah disetujui oleh dosen pembimbing pada tanggal 29 Juni 2015

Dosen Pembimbing I

Dosen Pembimbing II

Agi Putra Kharisma, ST., MT.
NIK.2013048604301001

Drs . Mardji, M.T.
NIP. 196708011992031001

LEMBAR PENGESAHAN

RANCANG BANGUN APLIKASI *MOBILE* MESIN Pencari INFORMASI PENJUALAN RUMAH

SKRIPSI

LABORATORIUM
PEMROGRAMAN APLIKASI PERANGKAT BERGERAK

Diajukan untuk memenuhi persyaratan memperoleh gelar Sarjana Komputer

Disusun oleh:

ITTO WIRAWAN MULYOTO

NIM. 115060800111045

Skripsi ini telah diuji dan dinyatakan lulus pada tanggal 29 Juli 2015

Penguji I

Eriq Muhammad Adams J,ST., M.Kom
NIP. 19850410 201212 1 001

Penguji II

Sabriansyah Rizqika Akbar, S.T., M.Eng.
NIP. 19820809 201212 1 004

Penguji III

Wibisono Sukmo Wardhono, ST., MT
NIK. 820404 06 1 1 0091

Mengetahui

Ketua Program Studi Informatika/Illmu Komputer

Drs. Marji, M.T.
NIP. 19670801 199203 1 001

**PERNYATAAN
ORISINALITAS SKRIPSI**

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).



Malang, 29 Juli 2015

Itto Wirawan Mulyoto
NIM. 115060800111045

KATA PENGANTAR

Dengan nama Allah SWT Yang Maha Pengasih Lagi Maha Penyayang. Segala puji dan syukur penulis panjatkan atas kehadiran-Nya, karena hanya atas rahmat, pertolongan dan kasih sayang-Nya penulis dapat menyelesaikan skripsi dengan judul “Rancang Bangun Aplikasi *Mobile* Mesin Pencari Informasi Penjualan Rumah” dengan baik.

Shalawat serta salam atas junjungan nabi besar kita Nabi Muhammad SAW beserta keluarga dan para sahabat sekalian. Melalui kesempatan ini, penulis menyampaikan rasa hormat dan terimakasih kepada pihak-pihak yang telah memberikan bantuan kepada penulis dalam penyelesaian skripsi ini. Pihak-pihak tersebut antara lain :

1. Bapak Agi Putra Kharisma ST., MT. dan Bapak Drs. Mardji, M.T. selaku dosen pembimbing I dan dosen pembimbing II yang telah banyak memberikan bimbingan, ilmu dan saran dalam penyusunan skripsi ini.
2. Kedua orang tua penulis yang telah memberi motivasi, kasih sayang serta dukungan moril dan materil kepada penulis. Saudara dari penulis yang telah memberikan semangat dan bantuan yang sangat besar dalam penyelesaian skripsi ini.
3. Bapak Sabriansyah Rizqika Akbar, ST., M.Eng. selaku dosen pembimbing akademik yang telah memberikan bimbingan, ilmu dan saran selama penulis belajar.
4. Seluruh dosen Program Studi Informatika/Ilmu Komputer atas kesediaannya membagi ilmunya kepada penulis.
5. Teman-teman kuliah diantaranya *member of TIF-E class*, serta semua teman-teman angkatan 2012, 2011 dan 2010, terimakasih atas segala bantuannya selama menjadi mahasiswa serta dukungannya dalam menyelesaikan skripsi ini.

Semoga jasa dan amal baik kita semua mendapatkan balasan dari Allah SWT. Penulis menyadari bahwa skripsi ini masih banyak kekurangan dan masih jauh dari

sempurna. Oleh karena itu, kritik dan saran untuk kesempurnaan skripsi ini, senantiasa penulis harapkan dari berbagai pihak.

Malang, 29 Juli 2015

Penulis



ABSTRAK

Itto Wirawan Mulyoto. 2015. Rancang Bangun Aplikasi *Mobile* Mesin Pencari Informasi Penjualan Rumah. Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya, Malang.

Dosen Pembimbing : Agi Putra Kharisma, ST., MT. dan Drs. Mardji, M.T.

Kebutuhan rumah yang semakin meningkat setiap tahunnya sudah tidak dapat dipungkiri lagi. Sejalan dengan meningkatnya kebutuhan rumah maka kebutuhan akan informasi penjualannya pun akan meningkat. Internet menyediakan banyak *website* mengenai informasi penjualan rumah. Akan tetapi keterbatasan perangkat bergerak dalam membuka *browsing* berbagai *website* informasi penjualan rumah satu per satu membutuhkan waktu yang lama. Dengan pergerakan informasi penjualan rumah yang selalu bertambah tiap waktu, calon pembeli yang ingin selalu *update* info baru harus selalu aktif dalam mengakses *website* penjualan rumah dan apabila *website* rujukan lebih dari satu maka ia harus memantau setiap *website* semuanya satu per satu. Dibutuhkan sebuah sistem yang dapat mencari sekaligus memantau informasi baru untuk pencari informasi penjualan rumah.

Pada penelitian ini, penulis akan membuat sebuah aplikasi perangkat bergerak mesin pencari informasi penjualan rumah yang menggunakan teknik *web scraping* dan notifikasi. Teknik *web scraping* digunakan untuk mendapatkan informasi penjualan rumah dari 3 situs. Sedangkan notifikasi akan digunakan untuk memantau informasi penjualan rumah baru dengan melakukan pengecekan secara otomatis pada ketiga situs.

Hasil pengujian validasi yang dilakukan, menunjukkan semua fitur pada sistem telah terpenuhi sesuai kebutuhan. Selain itu, pada pengujian *usability* menunjukkan bahwa aplikasi dapat digunakan dengan mudah oleh pengguna dengan index persentase lebih dari 50% dengan status sangat memuaskan.

Kata Kunci : mesin pencari, *web scraping*, *push notification*, perangkat bergerak

ABSTRACT

Itto Wirawan Mulyoto. 2015. Rancang Bangun Aplikasi Mobile Mesin Pencari Informasi Penjualan Rumah. Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya, Malang.

Dosen Pembimbing : Agi Putra Kharisma, ST., MT. dan Drs. Mardji, M.T.

The needs of houses is increasing every year can't be denied anymore. With the increasing needs of the house so it's the same with the needs of the sale information for house. The internet provides many websites about home sales information. However, the limitation of mobile devices for browsing various house sale information websites one by one it takes a long time. With the currency of houses sales information increasing every time, prospective buyers who want to always update new info should always active accessing the website and if the website is more than one, then he/she has to monitoring every websites all one by one. We need a system that can search simultaneously monitor new information to search the house sales.

In this study, writer will make a mobile devices application for searching house sales information using web scraping techniques and notifications. Web scraping technique is used to obtain information from the home sale 3 sites. While the notification will be used to monitor information on new home sales by automatically checking on all three sites.

Results of validation testing performed, showing all the features of the system have been met as required. In addition, the usability testing indicates that the application can be used easily by users to index a percentage of more than 50% with a very satisfactory status.

Keywords : search engine, web scraping, push notification, mobile application

DAFTAR ISI

KATA PENGANTAR.....	i
ABSTRAK	iii
ABSTRACT.....	iv
DAFTAR ISI.....	v
DAFTAR GAMBAR	viii
DAFTAR TABEL	x
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah.....	3
1.4 Tujuan.....	4
1.5 Manfaat.....	4
1.6 Sistematika Penulisan.....	4
BAB II DASAR TEORI.....	6
2.1 Search Engines	6
2.2 HTTP Request	7
2.3 Web Scraping	8
2.3.1 Library JSOUP	11
2.4 Aplikasi Perangkat Bergerak.....	12
2.5 Sistem Operasi Android	13
2.8.1 Siklus Hidup Android Activity	13
2.8.2 Arsitektur Android	15
2.8.3 Eclipse	16
2.8.4 SQLite	16
2.6 Push Notification.....	16

2.7	Rekayasa Perangkat Lunak	16
2.8	Pengujian Perangkat lunak	17
2.11.1	Teknik Pengujian.....	17
2.9	Strategi Pengujian	18
2.12.1	Pengujian Validasi.....	19
2.12.2	Pengujian Usability	19
2.12.3	Pengujian Unit.....	20
BAB III METODELOGI PENELITIAN		22
3.1	Studi Literatur	23
3.2	Analisis dan Perancangan.....	23
3.2.1	Analisa Kebutuhan	23
3.2.2	Perancangan Sistem.....	23
3.3	Implementasi	24
3.4	Pengujian dan Analisis	24
3.5	Pengambilan Kesimpulan dan Saran.....	25
BAB IV PERANCANGAN		26
4.1	Analisis Kebutuhan	27
4.1.1	Gambaran Aplikasi.....	27
4.1.2	Identifikasi Aktor	28
4.1.3	Analisis Sumber Data	28
4.1.4	Analisis Kebutuhan Fungsional	31
4.1.5	Analisis Kebutuhan Non Fungsional.....	37
4.2	Perancangan Sistem.....	38
4.2.1	Perancangan Arsitektur Sistem	38
4.2.2	Perancangan Aktifitas	39
4.2.3	Perancangan <i>Sequence</i>	44
4.2.4	Alur proses <i>Scraping</i>	48
4.2.5	Alur proses Notifikasi	49
4.2.6	Perancangan Kelas	50

4.2.7	Perancangan Skema Basis Data	55
4.2.8	Perancangan Antarmuka Aplikasi.....	55
BAB V IMPLEMENTASI DAN PENGUJIAN		59
5.1	Implementasi	60
5.1.1	Spesifikasi Lingkungan Sistem	60
5.1.2	Batasan Implementasi	61
5.1.3	Implementasi Basis Data	62
5.1.4	Implementasi <i>Class</i> dan <i>Assets</i> Pada File Program.....	62
5.1.5	Implementasi Kode Program.....	63
5.1.6	Implementasi Antarmuka Aplikasi.....	75
5.2	Pengujian.....	78
5.2.1	Pengujian Unit.....	78
5.2.2	Pengujian Validasi.....	93
5.2.3	Pengujian Usability	100
5.3	Analisis Hasil Pengujian	104
5.3.1	Analisis Hasil Pengujian Unit	104
5.3.2	Analisis Hasil Pengujian Validasi	105
5.3.3	Analisis Hasil Pengujian <i>Usability</i>	105
BAB VI PENUTUP		109
6.1	Kesimpulan.....	109
6.2	Saran.....	109
DAFTAR PUSTAKA		111
LAMPIRAN		114

DAFTAR GAMBAR

Gambar 2. 1 Ilustrasi <i>web scraping</i>	10
Gambar 2. 2 Kurs Bank Mandiri.....	11
Gambar 2. 3 Hasil <i>web scraping</i>	11
Gambar 2. 4 Siklus Hidup <i>Activity</i> di Android	14
Gambar 2. 5 Arsitektur Android	15
Gambar 3. 1 Tahapan Penelitian.....	22
Gambar 4. 1 Pohon Perancangan.....	26
Gambar 4. 2 Gambaran aplikasi mesin pencari informasi penjualan rumah	28
Gambar 4. 3 Tampilan daftar penjualan rumah situs rumahku.com	29
Gambar 4. 4 Tampilan daftar penjualan rumah situs rumah123.com	29
Gambar 4. 5 Tampilan daftar penjualan rumah situs lamudi.co.id	30
Gambar 4. 6 Diagram <i>Use Case</i> Aplikasi Mobile Mesin Pencari Penjualan Rumah	33
Gambar 4. 7 Arsitektur Sistem.....	38
Gambar 4. 8 <i>Activity Diagram</i> Mencari Informasi Penjualan Rumah	40
Gambar 4. 9 <i>Activity Diagram</i> Simpan Kata Kunci Pencarian.....	40
Gambar 4. 10 <i>Activity Diagram</i> Melihat Informasi Detail Penjualan Rumah	41
Gambar 4. 11 <i>Activity Diagram</i> Lihat Kata Kunci Tersimpan	42
Gambar 4. 12 <i>Activity Diagram</i> Hapus Kata Kunci Tersimpan.....	42
Gambar 4. 13 <i>Activity Diagram</i> Mendapat Notifikasi	43
Gambar 4. 14 <i>Sequence Diagram</i> Mencari Informasi Penjualan Rumah	44
Gambar 4. 15 <i>Sequence Diagram</i> Simpan Kata Kunci Pencarian	45
Gambar 4. 16 <i>Sequence Diagram</i> Melihat Detail Informasi Penjualan Rumah	45
Gambar 4. 17 <i>Sequence Diagram</i> Lihat Kata Kunci Pencarian.....	46
Gambar 4. 18 <i>Sequence Diagram</i> Hapus Kata Kunci Pencarian	46
Gambar 4. 19 <i>Sequence Diagram</i> Mendapat Notifikasi	47
Gambar 4. 20 Alur proses <i>scraping</i>	48
Gambar 4. 21 Alur proses Notifikasi	50
Gambar 4. 22 <i>Class Diagram</i>	51

Gambar 4. 23 Diagram logikal strktur basis data	55
Gambar 4. 24 Peta navigasi aplikasi	56
Gambar 4. 25 Halaman Utama	56
Gambar 4. 26 Halaman daftar informasi penjualan rumah	57
Gambar 4. 27 Halaman detail informasi penjualan rumah.....	57
Gambar 4. 28 Halaman riwayat.....	58
Gambar 4. 29 Halaman daftar penjualan rumah + informasi baru.....	58
Gambar 5. 1 Diagram pohon implementasi dan pengujian.....	59
Gambar 5. 2 DDL tabel Cari	62
Gambar 5. 3 DDL tabel Scraping	62
Gambar 5. 4 Impelementasi alur proses <i>scraping</i>	67
Gambar 5. 5 Implementasi alur proses notifikasi.....	75
Gambar 5. 6 Halaman Utama	76
Gambar 5. 7 Halaman Daftar Informasi Penjualan Rumah	76
Gambar 5. 8 Halaman Detail Informasi Penjualan Rumah	77
Gambar 5. 9 Halaman Riwayat	77
Gambar 5. 10 Pengujian Unit Alur Proses <i>Scraping</i>	83
Gambar 5. 11 <i>Flowchart</i> dan <i>Flowgraph</i> Alur Proses <i>Scraping</i>	85
Gambar 5. 12 Pengujian Unit Alur Proses Notifikasi	91
Gambar 5. 13 <i>Flowchart</i> dan <i>Flowgraph</i> Alur Proses Notifikasi.....	91
Gambar 5. 14 Tampilan hasil pencarian.....	94
Gambar 5. 15 Tampilan detail iklan rumah.....	95
Gambar 5. 16 Tampilan saat akan menyimpan kata kunci	96
Gambar 5. 17 Tampilan halaman riwayat	97
Gambar 5. 18 Tampilan menghapus kata kunci.....	98
Gambar 5. 19 Tampilan notifikasi.....	99

DAFTAR TABEL

Tabel 2. 1 Perbandingan GET dan POST	7
Tabel 4. 1 Identifikasi Aktor	28
Tabel 4. 2 Perbandingan jumlah informasi pada masing-masing situs penjualan rumah .	31
Tabel 4. 3 Daftar Kebutuhan Fungsional	32
Tabel 4. 4 Skenario <i>Use case</i> Mencari Informasi Penjualan Rumah.....	33
Tabel 4. 5 Skenario <i>Use case</i> Simpan Kata Kunci Pencarian.....	34
Tabel 4. 6 Skenario <i>Use case</i> Melihat Informasi Detail Penjualan Rumah	35
Tabel 4. 7 Skenario <i>Use case</i> Lihat Kata kunci Pencarian.....	35
Tabel 4. 8 Skenario <i>Use case</i> Hapus Kata kunci Pencarian.....	36
Tabel 4. 9 Skenario <i>Use case</i> Mendapat Notifikasi	36
Tabel 4. 10 Spesifikasi Kebutuhan Non Fungsional	38
Tabel 4. 11 Deskripsi <i>Class</i> Kata_Kunci	51
Tabel 4. 12 Deskripsi <i>Class</i> Cari_info	52
Tabel 4. 13 Deskripsi <i>Class</i> Database	53
Tabel 4. 14 Deskripsi <i>Class</i> Notifikasi.....	53
Tabel 4. 15 Deskripsi <i>Class</i> Notifikasi.....	54
Tabel 5. 1 Spesifikasi Perangkat Keras Komputer.....	60
Tabel 5. 2 Spesifikasi Perangkat Keras <i>Smartphone</i> Android	61
Tabel 5. 3 Spesifikasi perangkat lunak komputer	61
Tabel 5. 4 Spesifikasi perangkat lunak perangkat bergerak	61
Tabel 5. 5 Implementasi <i>class</i> pada kode program *.java	63
Tabel 5. 6 Kasus Uji Untuk Alur proses <i>Scraping</i>	86
Tabel 5. 7 Kasus Uji Untuk Pengujian Unit Alur Proses Notifikasi	92
Tabel 5. 8 Kasus Uji Mencari Informasi Penjualan Rumah.....	93
Tabel 5. 9 Kasus Uji Melihat Informasi Detail Penjualan Rumah.....	94
Tabel 5. 10 Kasus Uji Menyimpan Kata Pencarian	95
Tabel 5. 11 Kasus Uji Melihat Kata Pencarian Tersimpan.....	96

Tabel 5. 12 Kasus Uji Menghapus Kata Pencarian.....	97
Tabel 5. 13 Kasus Uji Mendapat Notifikasi.....	98
Tabel 5. 14 Hasil Pengujian Validasi	99
Tabel 5. 15 Hasil Pengujian <i>Usability</i>	103
Tabel 5. 16 Interpretasi Skor Likert	105
Tabel 5. 17 Hasil indeks persentase	106
Tabel 5. 18 Hasil status pengujian <i>usablitiy</i>	108



BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi telah merubah banyak gaya hidup orang dalam mencari sebuah informasi yang dibutuhkan. Dengan internet, orang bisa mengakses sebuah informasi apapun dan kapanpun ia inginkan. Apalagi ditambah dengan perkembangan teknologi perangkat bergerak atau *smartphone* yang sedemikian cepat di seluruh dunia memungkinkan orang bisa mengakses informasi lewat internet dimanapun hanya dengan genggam tangan. Di Indonesia sendiri, perkembangan perangkat bergerak atau *smartphone* sangat pesat. Persentase penggunaan *smartphone* di Indonesia pada Januari-Oktober 2014 terus meningkat. Mereka menggunakan ponsel pintar untuk mencari berbagai macam informasi. Dikutip dari koran Sindo, Pengguna *smartphone* yang mencari informasi belanja lewat ponsel sebanyak 95%, disusul pengguna *smartphone* yang mencari informasi lokal lewat ponsel sebanyak 94% [SIN-14]. Dengan teknologi yang ada pada perangkat bergerak ini, banyak informasi yang bisa diakses dan didapatkan melalui internet salah satunya adalah informasi mengenai penjualan rumah.

Rumah merupakan istilah untuk tempat dimana manusia tinggal dan hidup di dalamnya. Kebutuhan akan rumah sudah pasti menjadi salah satu kebutuhan pokok yang tidak bisa dipisahkan dari kehidupan manusia. Seiring dengan bertambahnya jumlah penduduk Indonesia tiap tahunnya, maka begitu juga dengan kebutuhan rumah sebagai tempat menetap akan turut meningkat. Hal itu tercermin dari maraknya penjualan apartemen dan rumah di awal tahunnya. Direktur Utama BTN, Maryono, mengemukakan, konsumen rumah kian bertambah untuk memenuhi kebutuhan papan [KOM-14]. Melihat banyaknya kebutuhan akan rumah, sudah pasti akan sejalan dengan kebutuhan pencarian informasi dalam hal ini pencarian informasi terkait penjualan rumah.

Saat ini masyarakat dalam mencari informasi penjualan rumah memakai cara-cara seperti mencari dari koran, maupun agen/broker. Namun dengan perkembangan teknologi dan informasi saat ini membuat beberapa orang mulai memanfaatkan internet dan perangkat bergerak untuk mencari informasi karena

kemudahan akses dimana saja. 59,9% pengguna Internet di Indonesia mengakses dunia maya itu melalui ponsel pintar atau *smartphone*. Angka itu mengalahkan persentase pengguna yang mengakses internet melalui laptop atau *netbook* [TEM-14].

Internet menyediakan banyak *website* mengenai informasi penjualan rumah. *Browsing* pada satu *website* saja dalam mencari informasi penjualan rumah akan kurang memuaskan sehingga calon pembeli harus mencari informasi dari *website* lain. Hal tersebut dikarenakan calon pembeli cenderung membandingkan beberapa rumah dalam situs penjualan rumah satu dengan yang lainnya untuk mendapatkan kriteria yang diinginkan seperti harga, lokasi, kondisi rumah dll. Akan tetapi keterbatasan perangkat bergerak dalam membuka *browsing* berbagai *website* informasi penjualan rumah satu per satu membutuhkan waktu yang lama. Diperlukan sebuah sistem yang bisa mengekstrak dan mengambil data semua situs untuk digabung menjadi satu. Akan tetapi, tidak semua situs penjualan rumah menyediakan API untuk memfasilitasi ekstrak data melalui *web service* situs tersebut. Oleh karena itu, dibutuhkan sistem untuk mengekstrak data melalui situs-situs tersebut secara langsung.

Selain itu, apabila dalam *website* tersebut ada informasi baru mengenai penjualan rumah calon pembeli tidak akan mengetahuinya tanpa mengeceknya secara langsung pada *website* tersebut. Hal tersebut akan membuat calon pembeli harus selalu aktif dalam mengakses *website* penjualan rumah untuk memantau informasi terbaru dan apabila *website* rujukan lebih dari satu maka ia harus memantau setiap *website* semuanya satu per satu. Oleh karena itu, dibutuhkan sebuah sistem yang dapat memperbarui informasi penjualan rumah tersebut secara otomatis dan memberitahukannya ke pengguna. Dalam hal ini, efisiensi waktu dan pembaruan informasi dalam pencarian penjualan rumah menjadi masalah yang ingin diatasi oleh penulis.

Berdasarkan uraian tersebut maka perlu dibuat solusi yaitu dengan membuat aplikasi pencari informasi penjualan rumah yang bisa menghimpun informasi menjadi satu dari berbagai *website* dengan menggunakan teknik *web scraping* dan pemberitahuan jika ada informasi penjualan rumah yang baru yang akan diterapkan berupa notifikasi pada aplikasi perangkat bergerak.

Diharapkan dengan adanya aplikasi ini, masyarakat bisa mencari informasi penjualan rumah yang diinginkan lebih cepat dan efisien sehingga dapat menghemat waktu pengguna serta dapat membantu untuk mengetahui keterbaruan informasi secara otomatis melalui fitur notifikasi.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, dapat dirumuskan masalah-masalah yang akan dibahas sebagai berikut :

1. Bagaimana mengekstrak data dan menyeragamkan informasi penjualan rumah dari beberapa situs yang berbeda?
2. Bagaimana menyatukan data hasil pencarian penjualan rumah dari beberapa situs yang berbeda?
3. Bagaimana membuat notifikasi atau pemberitahuan jika ada informasi baru mengenai penjualan rumah ?
4. Bagaimana tingkat kepuasan pengguna dalam menggunakan aplikasi *mobile* mesin pencari informasi penjualan rumah ?

1.3 Batasan Masalah

Berdasarkan latar belakang dan rumusan masalah yang telah diuraikan, agar permasalahan yang dirumuskan dapat lebih terfokus, maka penelitian ini dibatasi dalam hal :

1. Mesin pencari informasi penjualan rumah dari beberapa situs penjualan rumah memakai teknik *web scraping*. Situs penjualan rumah yang akan dilakukan proses *web scraping* antara lain :
 - Rumah123.com
 - Rumahku.com
 - Lamudi.co.id
2. Perangkat bergerak yang digunakan dalam pembuatan aplikasi ini adalah android.

1.4 Tujuan

Tujuan dari penelitian ini adalah untuk membuat mesin pencari informasi penjualan rumah menggunakan teknik *web scraping* pada perangkat bergerak untuk memudahkan masyarakat dalam mencari informasi secara efisien dan hemat waktu yang didapat dari situs-situs informasi penjualan rumah.

1.5 Manfaat

Adapun manfaat yang diharapkan dari hasil penelitian ini adalah sebagai berikut :

1. Membantu masyarakat dalam mencari informasi penjualan rumah yang diinginkan dengan cepat dan efisien
2. Memberikan pengetahuan dan pemahaman penggunaan teknik *scraping* pada situs-situs penyedia informasi penjualan rumah.

1.6 Sistematika Penulisan

Untuk mencapai tujuan yang diharapkan, maka sistematika penulisan yang disusun dalam laporan tugas akhir ini adalah sebagai berikut :

BAB I Pendahuluan

Berisi tentang latar belakang masalah, rumusan masalah, batasan masalah, tujuan dan manfaat dari penelitian serta sistematika penulisan.

BAB II Landasan Teori

Menguraikan tentang dasar teori dan referensi secara luas serta informasi yang diperlukan dalam permasalahan yang akan dibahas.

BAB III Metode Penelitian

Membahas tentang metode yang digunakan dalam penelitian dan langkah-langkah dalam penulisan yang terdiri dari studi literatur, analisis kebutuhan, perancangan sistem, implementasi sistem, pengujian dan analisis, serta pengambilan kesimpulan dan saran.

BAB IV Analisa dan Perancangan

Membahas analisis kebutuhan dan perancangan sistem sesuai dengan teori yang ada.

BAB V Implementasi dan Pengujian

Membahas tentang implementasi dari sistem berdasarkan metodologi dan perancangan serta memuat proses dan hasil pengujian terhadap sistem yang telah diimplementasikan.

BAB VI Penutup

Memuat kesimpulan yang diperoleh dari pembuatan dan pengujian perangkat lunak yang dikembangkan serta saran-saran untuk pengembangan lebih lanjut.



BAB II

DASAR TEORI

2.1 *Search Engines*

Search engine adalah sebuah sistem yang diperuntukan untuk pencarian dan pengambilan informasi serta menampilkan hasilnya. Biasanya sistem ini berbasis indeks beberapa dokumen HTML, sehingga pencarian dapat dengan mudah dilakukan [RAT-12]. Pada awalnya semua website yang ada pada *world wide web* terdaftar pada satu direktori yang juga tampil di *world wide web*. Saat seorang pengguna internet hendak mencari sesuatu informasi yang ada pada website-website terdaftar cukup dengan melihat pada direktori yang dimaksud. Tapi dikarenakan perkembangan pertambahan jumlah website di dunia maya begitu cepat dan karena kecepatannya membuat direktori tersebut tidak mampu menampung pertambahan jumlah web yang ada. Akhirnya hal itulah yang mendasari kemunculan web *search engine* atau lebih dikenal dengan nama *search engine*. Pengguna internet memerlukan sebuah alat untuk menemukan informasi yang dibutuhkan yang terdapat pada *world wide web*, itulah alasan yang mendorong para pembuat *search engines* membangun alat untuk mencari informasi yang ada pada *world wide web* [DEW-11].

Fungsi *search engine* tidak lain adalah untuk mencari informasi dengan cara mengetikkan kata kunci yang dimaksud, sehingga akan ditampilkan pada hasil pencarian yang berupa website website asli yang berisi berbagai bentuk informasi seperti tulisan, gambar, video, dsb. dengan mudah dan cepat dimana informasi tersebut berada [TES-11].

Dalam penelitian ini, *search engine* atau mesin pencari adalah aplikasi yang ingin dibangun oleh penulis. Aplikasi mesin pencari akan memakai teknik *web scraping* guna mengekstrak data dari situs-situs penjualan rumah yang telah ditentukan dan disimpan dalam sebuah database. Pengguna bisa mencari informasi penjualan rumah dari aplikasi tersebut sesuai kebutuhan dengan menggunakan query tertentu.

2.2 HTTP Request

Hypertext Transfer Protocol (HTTP) didesain untuk membangun komunikasi antara klien dan server. HTTP bekerja sebagai *request-response protocol* antara klien dan server. Sebuah *web browser* bisa menjadi klien dan aplikasi pada komputer yang menjadi *host* terhadap sebuah *web site* bisa menjadi server. Sebagai contoh, sebuah klien semisal *browser* mengirim sebuah HTTP *request* kepada server. Kemudian server akan mengirimkan respon kembali ke klien. Respon tersebut berisi informasi status mengenai *request* beserta kontennya [W3C-15].

Terdapat 2 metode HTTP *request* yang digunakan untuk *request-response* antara klien dan server yaitu GET dan POST [W3C-15].

- GET – *Request* data dari sumber yang spesifik
- POST – Pengiriman data untuk diproses ke sumber yang spesifik

Perbandingan GET dan POST dapat dilihat pada tabel dibawah ini [W3C-15] :

Tabel 2. 1 Perbandingan GET dan POST

	GET	POST
BACK button / Reload	Tidak berbahaya	Data akan dikirimkan ulang (<i>browser</i> akan memperingatkan pengguna bahwa data akan dikirimkan ulang)
<i>Bookmarked</i>	Bisa dilakukan <i>bookmarked</i>	Tidak bisa melakukan <i>bookmarked</i>
<i>Cached</i>	Bisa dilakukan <i>cached</i>	Tidak melakukan <i>cached</i>
Tipe <i>Encoding</i>	application/x-www-form- urlencoded	application/x-www-form- urlencoded atau multipart/form-data. Menggunakan <i>multipart encoding</i> untuk data biner

<i>History</i>	Parameter tersimpan berada pada <i>browser history</i>	Parameter tidak tersimpan di <i>browser history</i>
Batasan pada panjang data	Ya, ketika mengirim data, metode GET menambahkan data pada URL; dan panjang URL terbatas (panjang URL maksimum adalah 2048 karakter)	Tidak ada batasan
Batasan pada tipe data	Hanya karakter ASCII yang diperbolehkan	Tidak ada batasan. Data biner juga diperbolehkan
Keamanan	GET kurang aman bila dibandingkan dengan POST karena data yang dikirimkan adalah bagian dari URL. Jangan pernah menggunakan GET ketika mengirimkan <i>password</i> atau data sensitif lainnya	POST lebih aman daripada GET dikarenakan parameter tidak tersimpan pada <i>browser history</i> atau dalam <i>web server logs</i>
<i>Visibility</i>	Data dapat terlihat oleh semua orang di URL	Data tidak ditampilkan di URL

Pada penelitian ini, sistem memanfaatkan HTTP request untuk mengirimkan kriteria data yang dibutuhkan untuk scraping informasi penjualan rumah pada 3 situs penjualan rumah yang telah ditentukan menggunakan metode GET maupun POST.

2.3 Web Scraping

Web scraping merupakan sekumpulan teknik untuk mendapatkan informasi dari sebuah website secara otomatis tanpa harus menyalinnya secara manual. Tujuan dari *web scraper* adalah untuk mencari informasi tertentu, mengekstraksi informasi tersebut, kemudian menghimpunnya dalam halaman web yang baru.

Beberapa *web scraper* memfokuskan dirinya untuk mengubah data yang tidak terstruktur dan menyimpannya ke dalam data yang terstruktur [VAR-12].

Secara teknis *web scraping* tidak termasuk dalam bidang data mining karena data mengandung upaya untuk memahami pola semantik atau tren dalam kumpulan data yang besar yang telah diperoleh. Aplikasi *web scraping* (umumnya disebut juga dengan agen yang cerdas dan otomatis) berfokus hanya pada mendapatkan data dengan cara pengambilan dan ekstraksi serta dapat melibatkan kumpulan data dengan perbedaan ukuran yang signifikan [TUR-10].

Web scraping berhubungan dengan pengindeksan web yang merupakan suatu teknik universal yang dipakai hampir semua cari engine. Perbedaannya *web scraping* lebih berfokus pada transformasi dari suatu isi web tidak terstruktur, umumnya dalam format HTML (Hyper Text Markup Language) menjadi suatu format data terstruktur yang dapat disimpan dan dianalisis pada database atau pada lembar kerja.

Web scraping juga terkait dengan otomasi web yang mensimulasikan aktifitas web browsing dari manusia menggunakan perangkat lunak komputer. Contoh penggunaannya antara lain [WID-11]:

- Perbandingan harga online/katalog produk.
- Monitoring data cuaca.
- Deteksi perubahan website.
- Penelitian web.
- Integrasi data web.
- Web content mashup.
- Mengumpulkan informasi seputar perusahaan (nama, lokasi, Harga, kontak, dan lain-lain).
- Mengkliping artikel (judul, isi, kata kunci, sumber referensi, dan lain-lain).
- Otomasi sistem lelang.
- Mengekstraksi undian judi, dan lainnya.

Untuk mengekstraksi data dari suatu website perlu dilakukan beberapa hal seperti :

- Menemukan halaman HTML sasaran dalam sebuah situs dengan mengikuti hypertexts.

- Ekstraksi potongan-potongan data yang relevan dari halamannya.
- Penyaringan dan pemrosesan data

Dalam penelitian ini, *web scraping* digunakan untuk mengambil data dan informasi yang di situs penjualan rumah yang telah ditentukan oleh peneliti sebelumnya. *Web scraping* ini akan dibangun dalam aplikasi perangkat bergerak menggunakan *library* JSOUP. Data *web scraping* yang diambil dari beberapa website akan dihimpun menjadi satu dalam database menggunakan SQLite. Ilustrasi *web scraping* dapat dilihat dalam gambar 2.1 di bawah ini.



Gambar 2.1 Ilustrasi *web scraping*
Sumber : [VAR-12].

Pada Gambar 2.1 terlihat bagaimana ilustrasi *web scraping* yang mengambil data HTML sebuah halaman situs menjadi sebuah data yang terstruktur.

Sebuah contoh kasus *web scraping* yang akan dilakukan untuk mengambil informasi kurs mata uang Austalian Dollar untuk ditampilkan, yang ditunjukkan pada Gambar 2.2 yang merupakan sebuah HTML halaman situs.

Mata Uang	Symbol	Beli	Jual
United States Dollar	USD	12.033,00	12.197,00
Australian Dollar	AUD	10.214,00	10.508,00
Canadian Dollar	CAD	10.630,00	10.898,00
Swiss Franc	CHF	12.378,00	12.816,00
New Zealand Dollar	NZD	9.409,00	9.668,00
Danish Krone	DKK	1.955,00	2.120,00
British Pound	GBP	18.937,00	19.315,00
Hongkong Dollar	HKD	1.508,00	1.618,00
Japanese Yen	JPY	101,46	104,69
Singapore Dollar	SGD	9.213,00	9.456,00
EURO Spot Rate	EUR	14.976,00	15.305,00
Saudi Arabian Real	SAR	3.095,00	3.370,00
Swedish Krona	SEK	1.583,00	1.689,00
Norwegian Krone	NOK	1.689,00	1.860,00
China Yuan	CNY	1.928,00	2.020,00

Gambar 2. 2 Kurs Bank Mandiri
Sumber : [MAN-14]

Setelah melakukan proses *web scraping* dapat dilihat hasil ekstrak data yang diinginkan pada Gambar 2.4 di bawah ini.

```
Australian Dollar AUD 10.214,00 10.508,00
```

Gambar 2. 3 Hasil *web scraping*

2.1.1 Library JSOUP

JSOUP merupakan *library* JAVA yang ditujukan untuk pengekstrakan data dari HTML. JSOUP menyediakan API yang sangat sesuai untuk mengekstrak dan memanipulasi data menggunakan metode dari DOM, CSS dan jquery. Dalam memarsing HTML JSOUP memiliki metode DOM yang sama seperti yang dilakukan oleh peramban modern lakukan [JSO-15].

- Mengambil dan memarsing HTML dari URL, file atau string
- Mencari dan mengekstrak data menggunakan DOM *traversal* atau CSS *selectors*
- Memanipulasi elemen HTML, atribut, dan text
- Keluaran berupa HTML yang teratur

JSOUP didesain untuk bekerja pada berbagai macam variasi HTML yang ditemukan, mulai dari struktur HTML yang mudah hingga struktur tersulit sekalipun [JSO-15].

Pada penelitian ini JSOUP digunakan sebagai *library* untuk JAVA dalam membantu mengambil dan mengekstrak data penjualan rumah yang dibutuhkan situs-situs sumber data yang telah ditentukan sebelumnya.

2.4 Aplikasi Perangkat Bergerak

Aplikasi perangkat bergerak adalah sebuah program komputer yang di desain untuk berjalan di *smartphones*, *computer tablet* dan perangkat bergerak lainnya. Aplikasi biasanya tersedia melalui *application distribution platforms*, yang mulai muncul di tahun 2008 dan biasanya di operasikan oleh pemilik dari *mobile operating system*, seperti Apple App Store, Google Play, Windows Phone Store dan BlackBerry App World [SIE-08].

Beberapa aplikasi gratis, sedangkan lainnya harus dibeli. Biasanya, aplikasi diunduh dari platform ke perangkat tujuan, seperti iPhone, BlackBerry, Android atau Windows Phone, namun terkadang bisa juga diunduh ke komputer jinjing dan komputer *desktop*. Untuk aplikasi yang memiliki banderol harga, umumnya persentase keuntungan sebesar 20-30%, diberikan kepada *distribution provider* (seperti iTunes), dan sisanya diberikan kepada pembuat aplikasi [SIE-08].

Aplikasi perangkat bergerak awalnya menawarkan produktivitas yang umum dan pengambilan informasi, termasuk kalender, kontak, *stock market* dan informasi cuaca. Bagaimanapun juga, keinginan publik dan ketersediaan dari *developer tools* mendorong perluasan yang singkat di kategori lainnya, seperti pemrosesan kata, *social media*, *picture sharing*, *mobile games*, *factory automation*, *GPS mapping* dan

location-based service, perbankan, *networking and file transfer*, pendidikan, *video streaming*, *order tracking*, pembelian tiket dan *mobile medical apps*[SIE-08].

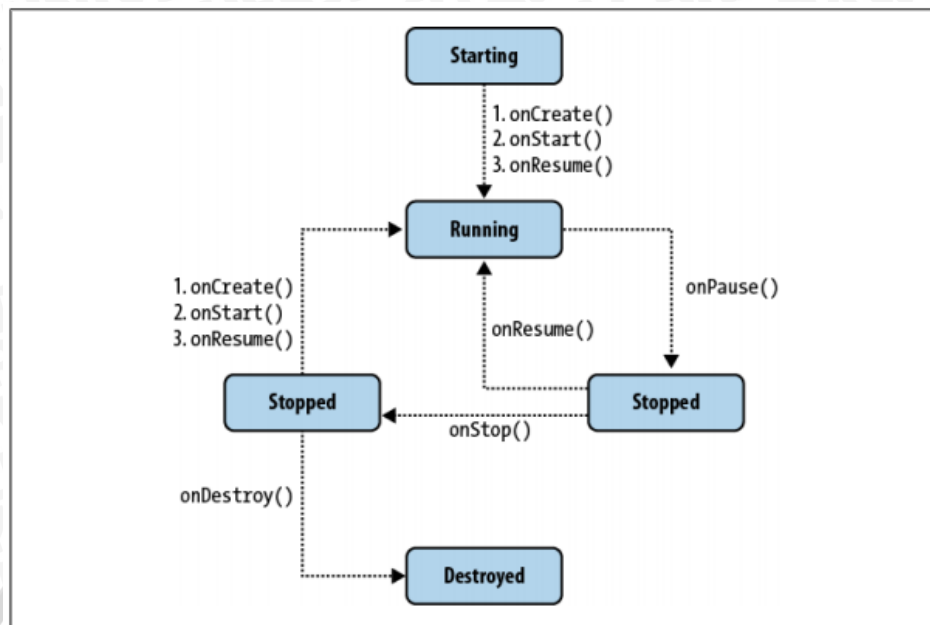
Popularitas dari aplikasi perangkat bergerak terus meningkat, sebagaimana penggunaannya semakin lazim di antara pengguna perangkat bergerak [LUD-12]. Pada Mei 2012 studi yang dilakukan *comScore* melaporkan bahwa kuartar sebelumnya, lebih banyak pelanggan perangkat bergerak menggunakan *apps* dibanding membuka *web* di perangkatnya: masing-masing sebesar 51.1% vs 49.8% [PER-12]. Peneliti menemukan bahwa penggunaan aplikasi perangkat bergerak berkorelasi kuat dengan konteks pengguna dan tergantung pada lokasi pengguna dan waktu pada hari [BOH-11].

2.5 Sistem Operasi Android

Android adalah platform *open source* komprehensif yang dirancang untuk perangkat *mobile*. Android didesain untuk berjalan pada segala bentuk perangkat fisik. Android tidak membuat asumsi apapun tentang ukuran perangkat layar, resolusi, *chipset* dan sebagainya. Inti dari Android di desain agar dapat *portable* [GAR-11].

2.8.1 Siklus Hidup Android Activity

Aplikasi Android biasanya terdiri dari beberapa *activity* yang saling terkait. *Acitivity* sendiri merupakan suatu komponen aplikasi yang menyediakan antarmuka pada layar pengguna, sehingga pengguna dapat melakukan interaksi dengan mudah pada aplikasi tersebut. Di bawah ini terdapat gambar dari siklus hidup Android *activity*.



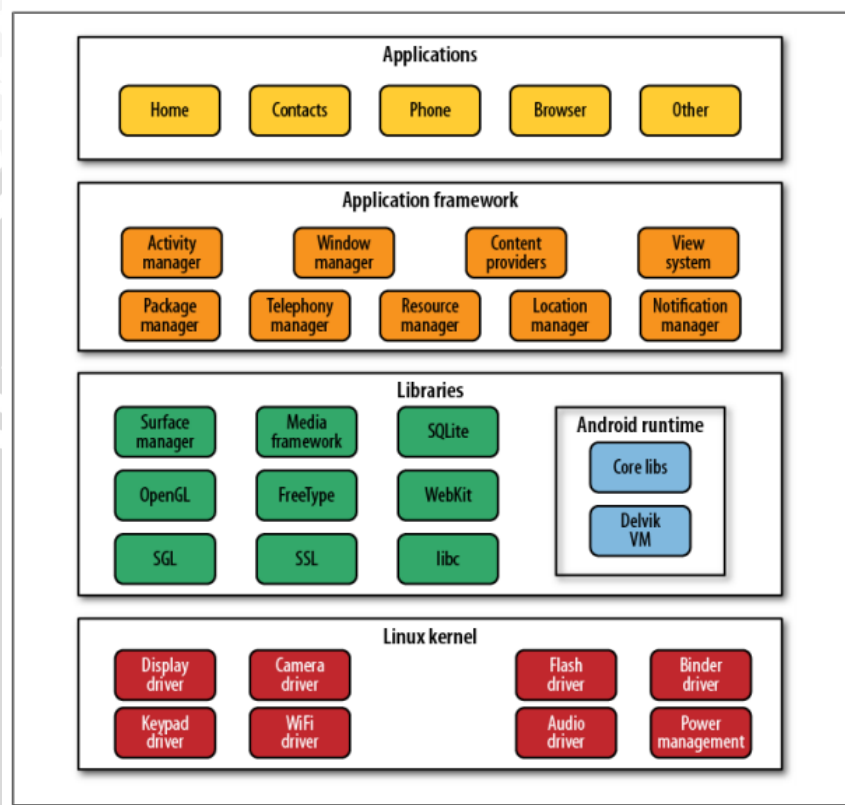
Gambar 2. 4 Siklus Hidup Activity di Android
Sumber : [GAR-11]

Terdapat beberapa *state* dalam siklus hidup *activity* Android, antara lain [GAR-11] :

1. *Starting State* yaitu keadaan ketika *activity* belum ada di memori.
2. *Running State* yaitu kondisi ketika berada di layar dan sedang berinteraksi dengan pengguna.
3. *Paused State* yaitu keadaan ketika *activity* tidak terlihat, namun masih ada di memori.
4. *Destroyed State* yaitu keadaan ketika *activity* sudah tidak berada lagi di memori.

2.8.2 Arsitektur Android

Arsitektur Android sendiri terdiri 5 lapisan yaitu antara lain *linux kernel*, *android runtime*, *libraries*, *application framework*, dan *applications*. Dimana tiap lapisan mempunyai karakteristik dan tujuan masing-masing [GAR-11].



Gambar 2.5 Arsitektur Android

Sumber : [GAR-11]

Dari gambar di atas diketahui beberapa lapisannya yaitu :

1. *Kernel Linux*, dimana Android sendiri dibangun di atas *kernel linux*. Terdapat beberapa alasan mengapa *linux* dipakai sebagai dasar arsitekturnya, yaitu karena fiturnya, keamanannya, dan juga *portability*.
2. Di lapisan *Android runtime*, terdapat *dalvik* yang merupakan *virtual-machine* yang dirancang khusus untuk Android.
3. Di lapisan *libraries*, biasanya terdapat pustaka umum yang diambil dari komunitas *open source* yang menyediakan layanan tertentu untuk lapisan aplikasi.

4. *Application framework* menyediakan lingkungan kaya yang menyediakan *services* yang dapat membantu *app developer* untuk menyelesaikan pekerjaan.
5. Di lapisan *application*, merupakan lapisan yang kita lihat secara langsung dan setiap aplikasi Android harus di *signed* sebelum di *install* di sebuah *device*.

2.8.3 Eclipse

Eclipse adalah sebuah IDE (*Integrated Development Environment*) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua *platform*. Sifat dari *Eclipse* ialah *Multi-platform*, *Multi-language*, *Multi-role*. Kelebihan dari *Eclipse* adalah gratis dan *open source*, yang berarti setiap orang boleh melihat kode pemrograman perangkat lunak ini, selain itu memiliki kemampuan untuk dapat dikembangkan oleh pengguna dengan menggunakan komponen yang dinamakan *plug-in* [MUL-10].

2.8.4 SQLite

SQLite merupakan alat untuk melakukan penanganan data (*data store*), yang telah ter-include dalam Android. *SQLite* sendiri terkenal sebagai *database* yang *opensource*, *stand alone SQL database*, berukuran kecil, tidak membutuhkan administrasi, tanpa *server*, tanpa *file* konfigurasi dan juga telah digunakan pada banyak aplikasi terkenal seperti *Browser Firefox* dan *IPhone* [MUL-10].

2.6 Push Notification

Dalam dunia teknologi perangkat bergerak, *notification* atau pemberitahuan adalah pesan yang ditujukan untuk pengguna aplikasi. Sedangkan *push notification* adalah suatu fitur yang memberitahukan pengguna bila ada pesan baru atau suatu *event* baru. Pemberitahuan ini secara otomatis akan muncul walaupun pengguna sedang tidak aktif dalam aplikasi tersebut [DEV - 14].

2.7 Rekayasa Perangkat Lunak

Rekayasa perangkat lunak adalah disiplin ilmu yang membahas semua aspek produksi perangkat lunak, mulai dari tahap awal spesifikasi sistem sampai

pemeliharaan sistem setelah digunakan. Pada definisi ini, ada dua kata kunci, yaitu : [SOM-11]

1. Disiplin ilmu rekayasa

Seorang perekayasa membuat suatu alat bekerja. Perekayasa menerapkan teori, metode dan alat bantu yang sesuai. Perekayasa menggunakannya dengan selektif dan selalu mencoba mencari solusi terhadap permasalahan, walaupun tidak ada teori atau metode yang mendukung. Perekayasa juga menyadari bahwa mereka harus bekerja dalam batasan organisasi dan keuangan, sehingga mereka berusaha mencari solusi dalam batasan-batasan ini.

2. Semua aspek proses pembuatan aplikasi

Rekayasa perangkat lunak tidak hanya berhubungan dengan proses teknis dari pengembangan perangkat lunak, tetapi juga dengan kegiatan seperti manajemen proyek perangkat lunak dan pengembangan alat bantu, metode dan teori untuk mendukung produksi perangkat lunak.

2.8 Pengujian Perangkat lunak

Tujuan dari pengujian atau *tesing* perangkat lunak adalah untuk menemukan dan memperbaiki sebanyak mungkin kesalahan dalam program sebelum menyerahkan program kepada *customer*. Salah satu pengujian yang baik adalah pengujian yang memiliki probabilitas yang tinggi dalam menemukan kesalahan perangkat lunak [PRE - 01:482].

2.11.1 Teknik Pengujian

Pengujian perangkat lunak memerlukan perancangan kasus uji (*test case*) agar dapat menemukan kesalahan dalam waktu singkat dan usaha minimum. Berbagai macam metode perancangan kasus uji telah berevolusi. Metode-metode ini menyediakan developer pendekatan sistematis untuk pengujian. Terlebih lagi metode-metode ini menyediakan mekanisme yang dapat membantu memastikan kelengkapan dari pengujian dan menyediakan kemungkinan tertinggi untuk menemukan kesalahan-kesalahan dalam perangkat lunak [PRE- 01:443].

2.11.1.1 Black-Box Testing

Black-box testing atau *behavioral testing* berfokus pada persyaratan fungsional perangkat lunak [PRE - 01:459]. Dengan demikian, pengujian *blackbox* memungkinkan perekayasa perangkat lunak untuk mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk sebuah program. Pengujian *black-box testing* berusaha untuk menemukan kesalahan dalam kategori berikut [PRE - 01:460]:

1. Fungsi yang tidak benar atau fungsi yang hilang
2. Kesalahan antarmuka
3. Kesalahan dalam struktur data atau akses *database* eksternal
4. Kesalahan perilaku (*behavior*) atau kesalahan kinerja
5. Inisialisasi dan kesalahan terminasi

Tidak seperti pengujian *white-box*, yang dilakukan pada saat awal proses pengujian, pengujian *black-box* cenderung diaplikasikan selama tahap akhir pengujian. Karena pengujian *black-box* memperhatikan struktur kontrol, maka perhatian berfokus pada domain informasi. Pengujian didesain untuk menjawab pertanyaan-pertanyaan berikut [PRE - 01:460]:

1. Bagaimana validitas fungsional diuji?
2. Kelas *input* apa yang akan membuat test case menjadi baik?
3. Apakah sistem sangat sensitif terhadap harga input tertentu?
4. Bagaimana batasan dari suatu data diisolasi?
5. Kecepatan dan volume data apa yang dapat ditolerir oleh sistem?
6. Apa pengaruh kombinasi data tertentu terhadap operasi sistem?

2.11.1.2 White-Box Testing

White-box testing atau *glass-box testing* merupakan sebuah metode perancangan kasus uji yang menggunakan struktur kontrol dari perancangan prosedural untuk memperoleh kasus uji [PRE - 01:444].

2.9 Strategi Pengujian

Strategi untuk pengujian perangkat lunak mengintegrasikan metode desain testcase perangkat lunak ke dalam sederetan langkah yang direncanakan dengan

baik, dan hasilnya adalah konstruksi perangkat lunak yang berhasil [PRE - 01:477]. Strategi pengujian harus mengakomodasi pengujian tingkat rendah yang diperlukan untuk membuktikan bahwa segmen kode sumber yang kecil telah diimplementasikan dengan tepat, demikian juga pengujian tingkat tinggi yang memvalidasi fungsi-fungsi sistem mayor yang berlawanan dengan kebutuhan pelanggan. Proses pengujian dimulai dengan menguji validasi sistem atau dikenal sebagai pengujian validasi, dan dilanjutkan dengan pengujian *usability*.

2.12.1 Pengujian Validasi

Pada kulminasi pengujian terintegrasi, perangkat lunak secara lengkap dirakit sebagai suatu paket; kesalahan *interfacing* telah diungkap dan dikoreksi, dan seri akhir dari pengujian perangkat lunak, yaitu pengujian validasi dapat dimulai. Validasi dapat ditentukan dengan berbagai cara, tetapi definisi yang sederhana adalah bahwa validasi berhasil bila perangkat lunak berfungsi dengan cara yang dapat diharapkan secara bertanggung jawab oleh pelanggan. Validasi perangkat lunak dicapai melalui sederetan pengujian *black-box* yang memperlihatkan konformitas dengan persyaratan. Rencana pengujian menguraikan kelas-kelas pengujian yang akan dilakukan, dan prosedur pengujian menentukan *test case* spesifik yang akan digunakan untuk mengungkap kesalahan dalam konformitas dengan persyaratan. Baik rencana dan prosedur didesain untuk memastikan apakah semua persyaratan fungsional dipenuhi; semua persyaratan kinerja dicapai; dokumentasi betul dan direkayasa oleh manusia; dan persyaratan lainnya dipenuhi (transportabilitas, kompatibilitas, pembetulan kesalahan, maintainabilitas) [PRE - 01:495].

2.12.2 Pengujian Usability

Pengujian *usability* adalah pengujian terhadap atribut kualitas yang menilai seberapa mudahnya user *interface* yang digunakan dalam pengembangan perangkat lunak [JAK - 03:1]. Pengujian ini menunjukkan apakah perangkat lunak yang sudah dibuat dapat digunakan dengan baik oleh calon pengguna. Terdapat 5 komponen dalam pengujian *usability* [JAK - 94:152], yaitu :

1. *Learnability*

Menjelaskan tingkat kemudahan pengguna atau user untuk menyelesaikan tugas-tugas dasar ketika pertama kali mereka berhadapan dengan sistem yang ada.

2. *Efficiency*

Menjelaskan seberapa cepat pengguna dapat menyelesaikan tugas-tugas yang ada saat pengguna pertama kali menggunakan aplikasi.

3. *Memorability*

Menjelaskan tentang tingkat kemudahan pengguna dalam menggunakan sistem dengan baik, setelah beberapa lama tidak menggunakannya.

4. *Errors*

Menjelaskan kemungkinan terjadinya *error* atau kesalahan yang dilakukan oleh pengguna dan seberapa mudah mereka dapat mengatasinya.

5. *Satisfaction*

Menjelaskan tingkat kepuasan pengguna dalam menggunakan sistem yang telah dibuat. Pertanyaan yang diajukan dalam kuisioner ini didapatkan dari pihak yang sebelumnya sudah melakukan kuisioner terhadap perangkat lunak berbasis *mobile* dengan pengujian *usability* yang sejenis. Pertanyaan-pertanyaan tersebut merupakan terjemahan dari pertanyaan yang ada pada 5 *quality components* [JAK-94:158] yang akan disertakan sebagai lampiran.

2.12.3 Pengujian Unit

Pengujian unit berfokus pada usaha verifikasi pada inti terkecil dari desain perangkat lunak, yakni modul. Dengan menggunakan gambaran desain prosedural sebagai panduan, jalur kontrol yang penting diuji untuk mengungkap kesalahan di dalam batas modul tersebut. Kompleksitas relatif dari pengujian dan kesalahan yang diungkap dibatasi oleh ruang lingkup batasan yang dibangun untuk pengujian unit. Pengujian unit biasanya berorientasi pada *white-box*, dan langkahnya dapat dilakukan secara paralel untuk model bertingkat [PRE - 01:485].

Pengujian yang terjadi sebagai bagian dari inti digambarkan secara skematis pada Gambar 2.7. *Interface* modul diuji untuk memastikan bahwa informasi secara tepat mengalir masuk dan keluar dari inti program yang diuji. Struktur data lokal diuji untuk memastikan bahwa data yang tersimpan secara temporal dapat tetap

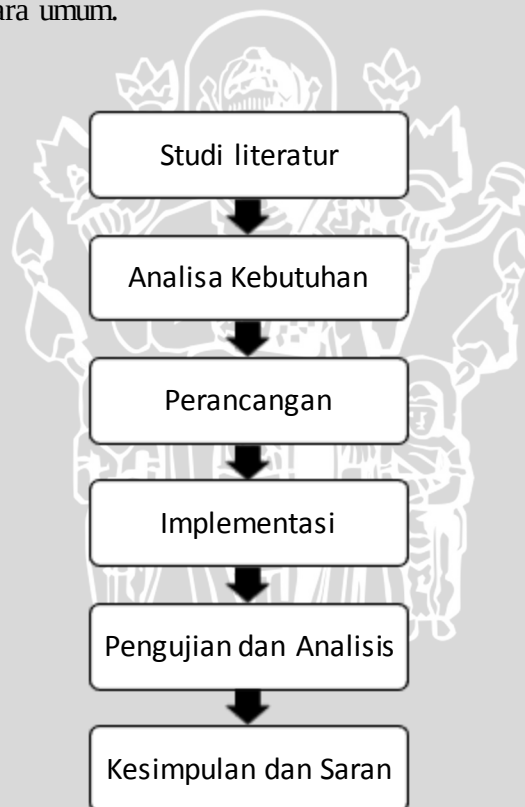
menjaga integritasnya selama semua langkah di dalam suatu algoritma dieksekusi. Kondisi batas diuji untuk memastikan bahwa modul beroperasi dengan tepat pada batas yang ditentukan untuk membatasi pemrosesan. Semua jalur independen (jalur dasar) yang melalui struktur kontrol dipakai sedikitnya satu kali. Dan akhirnya, penanganan kesalahan diuji [PRE - 01:485]



BAB III

METODELOGI PENELITIAN

Pada bab ini dijelaskan langkah-langkah yang akan dilakukan dalam penelitian ini. Langkah pertama yang dilakukan yaitu mengumpulkan teori-teori pendukung dan mengemasnya ke dalam studi literatur. Kemudian dilanjutkan dengan proses analisis kebutuhan. Setelah analisis kebutuhan dilakukan, proses selanjutnya yaitu melakukan perancangan dan melakukan implementasi sesuai dengan perancangan. Kemudian dilakukan pengujian dan analisis pada sistem yang telah dibuat. Kesimpulan dan saran disertakan sebagai catatan atas sistem dan kemungkinan arah pengembangan sistem selanjutnya. Gambar 3.1 menunjukkan tahapan penelitian secara umum.



Gambar 3. 1 Tahapan Penelitian

3.1 Studi Literatur

Studi literatur menjelaskan dasar teori yang digunakan untuk menunjang penulisan skripsi. Teori-teori pendukung tersebut meliputi :

- a. *Search Engine*
- b. *HTTP Request*
- c. *Web Scraping*
- d. Sistem Operasi Android
- e. *Push Notification*
- f. Rekayasa Perangkat Lunak
- g. Pengujian Perangkat Lunak
- h. Strategi Pengujian

3.2 Analisa dan Perancangan

Analisa dan perancangan merupakan sebuah tahapan yang berfungsi untuk merumuskan kebutuhan-kebutuhan yang diperlukan dalam pengimplementasian aplikasi ini. Tahapan-tahapan yang ada dalam analisis dari perancangan antara lain:

3.2.1 Analisa Kebutuhan

Dalam tahapan ini diperlukan gambaran tentang bagaimana jalannya aplikasi dan analisa tentang kebutuhan sumber data yaitu menjelaskan data-data apa saja yang akan dibutuhkan dari situs penjualan rumah. Kebutuhan sumber data didapatkan dengan melakukan observasi pada 3 situs penjualan rumah yang telah ditentukan.

Selain itu, pada tahapan ini diperlukan analisa kebutuhan sistem yang dibutuhkan, yaitu menjelaskan aktor yang terlibat dan interaksinya terhadap aplikasi. Analisa kebutuhan dilakukan dengan mengidentifikasi semua kebutuhan sistem untuk pengguna yang kemudian akan dimodelkan dalam bentuk *use case diagram*.

3.2.2 Perancangan Sistem

Perancangan sistem dilakukan setelah kebutuhan-kebutuhan dari sistem telah terkumpul dari proses analisis kebutuhan. Perancangan aplikasi pada

penelitian ini akan berdasarkan orientasi objek. Perancangan akan terdiri dari lima model perancangan, yaitu perancangan arsitektur sistem, perancangan aktifitas yang dimodelkan dengan *activity diagram*, perancangan interaksi yang dimodelkan dengan *sequence diagram*, perancangan alur proses sistem yang dimodelkan dengan *flowchart*, perancangan skema basis data, perancangan kelas dan perancangan antarmuka sistem.

3.3 Implementasi

Implementasi merupakan tahap pembangunan sistem. Setelah melakukan tahapan perancangan sistem barulah dibuat aplikasi ini dengan menggunakan bahasa pemrograman Java untuk platform Android. Selain itu, pada tahapan ini terdapat beberapa hal yang akan diterapkan diantaranya:

- Implementasi basis data, dengan menggunakan sebuah SQLite yang bertujuan untuk memudahkan manipulasi data.
- Implementasi pengkodean sistem, dengan menggunakan bahasa pemrograman java menggunakan Eclipse ADT.

3.4 Pengujian dan Analisis

Pengujian perangkat lunak merupakan tahapan untuk mengetahui terpenuhinya kinerja dan performa sistem sesuai dengan kebutuhan yang telah ditentukan sebelumnya melalui serangkaian proses uji. Ada tiga macam pengujian yang dilakukan pada aplikasi ini antara lain :

1. *White-box testing*

Pengujian unit menggunakan metode *white-box testing* yang berfokus pada usaha verifikasi pada inti terkecil dari desain perangkat lunak, yakni modul.

2. *Black-box testing*

Pengujian validasi menggunakan metode *black-box testing* untuk memeriksa hasil eksekusi aplikasi dan fungsionalitasnya. Kesesuaian fitur-fitur aplikasi dengan kebutuhan sistem dilakukan untuk mengetahui validitas aplikasi.

3. Pengujian *Usability*

Pengujian *usability* menggunakan metode kuisiner dan pengujian kompatibilitas. Kuisiner yang dibuat dengan menggunakan metode skala likert dimana responden memberikan nilai atau skor 1 sampai 5. Kuisiner ini nantinya akan disebarakan secara langsung dengan cara mendatangi responden yang telah terbiasa menggunakan *smart phone* dalam kehidupan sehari-harinya.

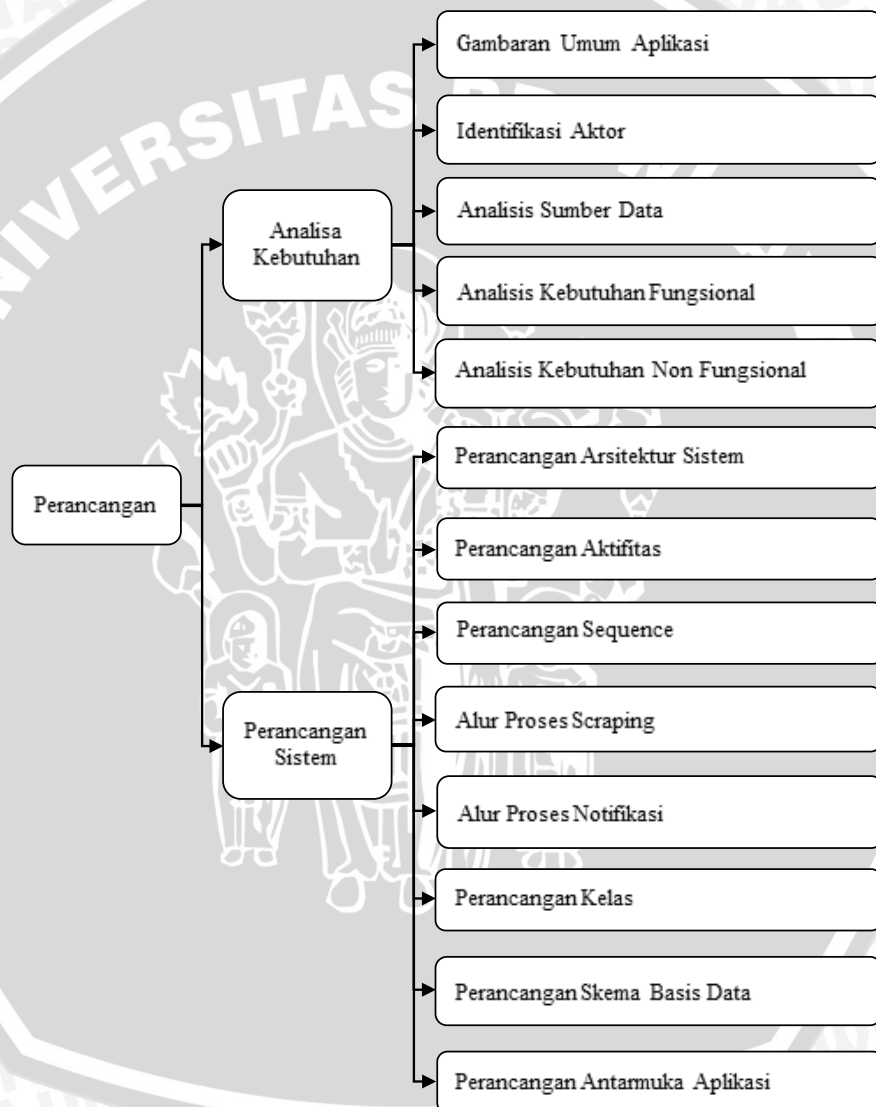
Setelah tahap pengujian, dilakukan analisis untuk mengetahui hasil dari pengujian perangkat lunak sehingga bisa didapatkan kesimpulan dan sistem yang telah dibuat.

3.5 Pengambilan Kesimpulan dan Saran

Pengambilan kesimpulan dilakukan setelah semua tahapan perancangan sistem, implementasi sistem, dan pengujian sistem telah selesai dilakukan. Kesimpulan diambil dari hasil pengujian dan analisis Aplikasi Mesin Pencari Informasi Penjualan Rumah. Penulisan saran berguna untuk memberikan pertimbangan atas hasil yang telah dilakukan.

BAB IV PERANCANGAN

Bab ini akan menjelaskan mengenai analisis dan perancangan aplikasi Mesin Pencari Informasi Penjualan Rumah. Ada 2 tahap pada bab ini, yaitu analisa kebutuhan dan perancangan sistem. Gambar 4.1 akan menjelaskan prosedur yang akan dilakukan pada bab ini.



Gambar 4. 1 Pohon Perancangan

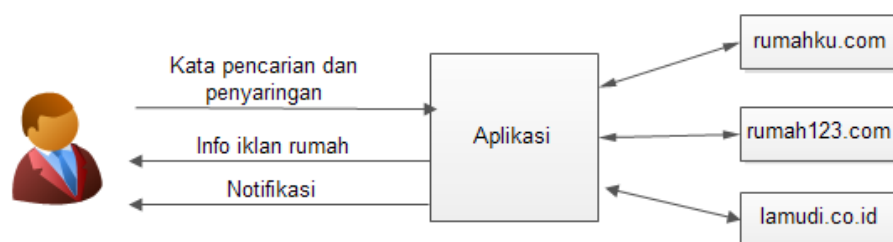
4.1 Analisis Kebutuhan

Tahap analisis kebutuhan merupakan identifikasi terhadap semua kebutuhan aplikasi perangkat bergerak mesin pencari informasi penjualan rumah. Terdapat beberapa tahapan dalam analisa kebutuhan ini diantaranya meliputi gambaran aplikasi, identifikasi aktor, daftar kebutuhan fungsional dan non-fungsional dengan menggunakan pemodelan diagram *use case* , dan analisis sumber data scraping. Analisis kebutuhan ini bertujuan untuk memberikan analisa secara jelas tentang apa-apa saja kebutuhan yang disediakan sistem demi memenuhi kebutuhan pengguna.

4.1.1 Gambaran Aplikasi

Aplikasi *mobile* mesin pencari informasi penjualan rumah merupakan sebuah aplikasi yang ditujukan untuk membantu masyarakat dalam mencari informasi penjualan rumah melalui perangkat bergerak. Pengguna bisa mencari informasi tersebut dengan memasukkan kata kunci pencarian dan penyaringan. Informasi yang diberikan oleh aplikasi seperti link informasi, nama informasi, lokasi rumah, harga , jumlah kamar tidur & kamar mandi. Informasi penjualan rumah didapat dari hasil data scraping pada 3 situs penjualan rumah yang digabung menjadi satu lalu disimpan ke dalam basis data lokal pada perangkat bergerak kemudian akan ditampilkan ke pengguna.

Fitur notifikasi atau pemberitahuan berfungsi memberitahu pengguna apabila terdapat informasi baru terkait penjualan rumah berdasarkan data kata kunci pencarian yang tersimpan. Fitur tersebut dilakukan dengan menjalankan *scraping* secara otomatis berdasarkan kata kunci pencarian yang tersimpan di basis data dengan selang waktu tertentu. Semua proses tersebut dilakukan secara *background* yang artinya aplikasi berjalan sendiri dan tanpa mengganggu aplikasi lain. Hasil *scraping* otomatis yang didapat akan disimpan ke basis data dan dibedakan antara informasi lama dan informasi baru berdasarkan perbandingan dalam basis data. Informasi baru yang telah terkonfirmasi akan dinotifikasi ke pengguna pada perangkat bergerak berupa jumlah informasi baru dan tampilan daftar informasi yang berbeda. Pada Gambar 4.2 dibawah ini memperlihatkan gambaran aplikasi mesin pencari informasi penjualan rumah.



Gambar 4. 2 Gambaran aplikasi mesin pencari informasi penjualan rumah

4.1.2 Identifikasi Aktor

Tahap ini bertujuan untuk melakukan identifikasi terhadap aktor yang terlibat dengan aplikasi *mobile* mesin pencari informasi penjualan rumah.

Tabel 4. 1 Identifikasi Aktor

Aktor	Deskripsi
Pengguna	Pengguna merupakan semua orang yang menggunakan aplikasi ini untuk mencari kebutuhan informasi penjualan rumah.

4.1.3 Analisis Sumber Data

Informasi mengenai penjualan rumah diambil dari 3 situs jual rumah yang sudah ditentukan sebelumnya. Untuk setiap situs jual rumah akan dilakukan analisa data dan informasi apa saja yang terdapat di dalamnya. Setelah itu, dari tiap situs akan dilakukan perbandingan dari informasi yang ada. Pemilihan sumber data diambil dari 3 situs jual rumah, yaitu :

1. Situs Rumahku.com

Berikut adalah tampilan dari situs rumahku.com dengan memasukkan kata kunci pencarian rumah “Malang” yang ditunjukkan pada Gambar 4.3

Rumahku.com Cari Properti Pencarian Lainnya Bantuan Artikel Info Lainnya Daftar / Login Pasang Iklan Properti

Urutkan: Premium 1 - 20 dari 1,521 listing



Rumah Dijual: RUMAH MEWAH, GUNUNG MALANG, BALIKPAPAN
Damai, Balikpapan 76114
Rp. 1,700,000,000

K. Tidur	3	K. Mandi	2
Luas	100	Sertifikat	SHM
Tanah	220	Panjang	10
Lebar	22	ID	2JGD0M2K

Dipasarkan oleh: HILAL HAITAMI



Rumah Dijual: Rumah di Bendungan Sigura Gura Barat
Karang Besuki, Malang 65149
Rp. 1,000,000,000

K. Tidur	3	K. Mandi	2
Luas	100	Sertifikat	SHM
Tanah	132	ID	Y3DK076X

Dipasarkan oleh: Yuni Agustina



Tanah Dijual: Tanah Kavling Daerah Tidar Karang Besuki
Tidar, Malang 65119

Properti Baru Pilihan

Bogor Raya Residential
Rumah baru di Sukaraja, Bogor
Hunian Yang Prestisius
Harga Mulai 300 Jutaan

The Greenhill
Rumah baru di Depok, Depok
View Terbaik
Area Hijau 12,5 Ha

Riverside Residence
Rumah baru di Sentul City, Bogor
My First Choice

Desain & Arsitektur

Gambar 4.3 Tampilan daftar penjualan rumah situs rumahku.com
Sumber : [RKU-15]

Pada Gambar 4.3 merupakan halaman awal pencarian yang menampilkan daftar penjualan rumah situs rumahku.com. Terdapat beberapa informasi yang bisa diambil dari halaman tersebut seperti judul iklan, harga, lokasi, jumlah kamar tidur, jumlah kamar mandi, luas tanah, luas bangunan dan sertifikat.

2. Situs Rumah123.com

Berikut adalah tampilan dari situs rumahku.com dengan memasukkan kata kunci pencarian rumah “Malang” yang ditunjukkan pada Gambar 4.4.

LIST GRID Featured Terbaru Harga Terpopuler

Rumah dijual di Malang, Jawa Timur (1,124 listing)



Perumahan Minimalis Idaman Keluarga Naman Dan Asri Di
• Rumah Dijual
★ FEATURED
Singosari, Malang
K. TIDUR: 2 K. MANDI: 1 GARASI: 1
LUAS BANGUNAN: 300m²
LUAS TANAH: 60m²
IDR 162.000.000
Simulasikan dengan Permata KPR
Independent Property Agent
Tayang sejak 5 hari yang lalu



Rumah Tirtasani Malang Cokoh Homestay/villa
• Rumah Dijual
★ FEATURED
Karangploso, Malang
K. TIDUR: 3 K. MANDI: 4 GARASI: 2
LUAS BANGUNAN: 300m²
LUAS TANAH: 367m²
IDR 1.300.000.000
Simulasikan dengan Permata KPR
Almhira Property
Tayang sejak 12 hari yang lalu

Info Lokal
Malang yang mempunyai luas area 252.136 km² adalah kota terbesar ke-2 di provinsi Jawa Timur. Populasi yang tercatat pada tahun 2010 adalah 820.243 jiwa (menurut data BPS) dengan jumlah...

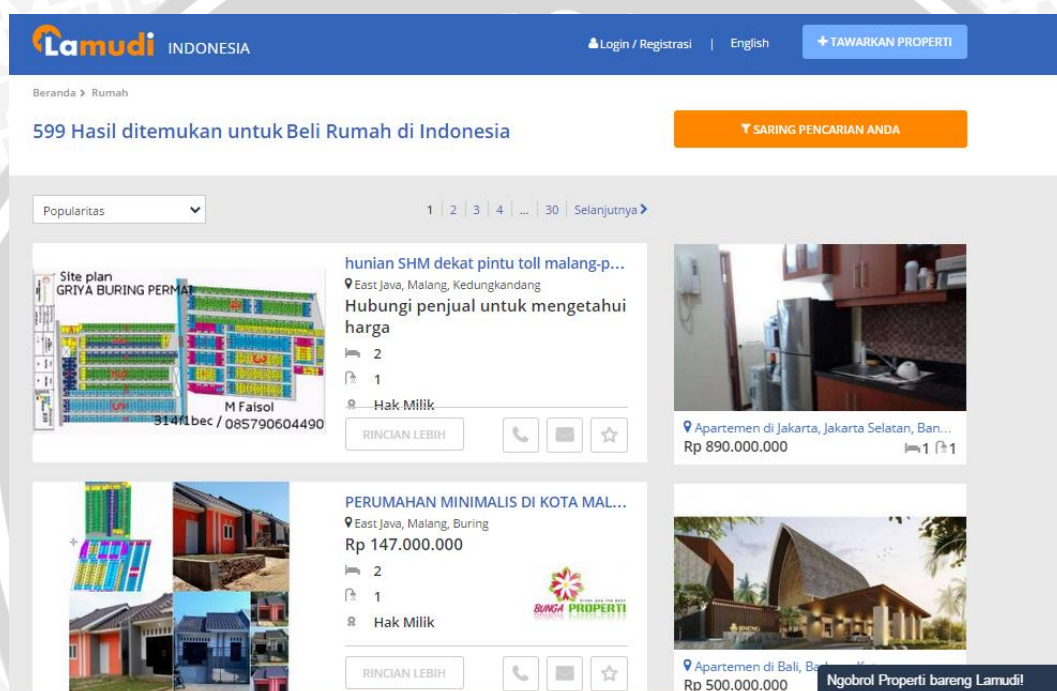
Berita Properti Terbaru
1 hari yang lalu
Usul Hapuskan NJOP dan PBB, Pajak...
1 hari yang lalu
Pusat Belanja Terbesar di Indonesia Itu...
1 hari yang lalu
Rumah Seperti Ini Akan Sulit Dapatkan...

Gambar 4.4 Tampilan daftar penjualan rumah situs rumah123.com
Sumber : [RUM-15]

Pada Gambar 4.4 merupakan halaman awal pencarian yang menampilkan daftar penjualan rumah situs rumah123.com. Terdapat beberapa informasi yang bisa diambil dari halaman tersebut seperti judul iklan, harga, lokasi, jumlah kamar tidur, jumlah kamar mandi, luas tanah, garasi, dan luas bangunan.

3. Situs lamudi.co.id

Berikut adalah tampilan dari situs lamudi.co.id dengan memasukkan kata kunci pencarian rumah “Malang” yang ditunjukkan pada Gambar 4.5



Gambar 4.5 Tampilan daftar penjualan rumah situs lamudi.co.id
Sumber : [LAM-15]

Pada Gambar 4.5 merupakan halaman awal pencarian yang menampilkan daftar penjualan rumah situs lamudi.co.id. Terdapat beberapa informasi yang bisa diambil dari halaman tersebut seperti judul iklan, harga, lokasi, jumlah kamar tidur, dan jumlah kamar mandi .

Dari penjelasan informasi yang ada di setiap situs, dapat dipastikan bahwa setiap situs memiliki jumlah informasi penjualan rumah yang ditampilkan berbeda-beda. Pada Tabel 4.2. di bawah ini menampilkan perbandingan informasi yang ada pada setiap situs.

Tabel 4. 2 Perbandingan jumlah informasi pada masing-masing situs penjualan rumah

No.	Informasi	Rumahku.com	Rumah123.com	Lamudi.co.id
1.	Judul / Tautan	Ada	Ada	Ada
2.	Gambar Rumah	Ada	Ada	Ada
3.	Alamat Rumah	Ada	Ada	Ada
4.	Harga	Ada	Ada	Ada
5.	Kamar tidur	Ada	Ada	Ada
6.	Kamar Mandi	Ada	Ada	Ada
7.	Luas tanah	Ada	Ada	Tidak ada
8.	Luas bangunan	Ada	Ada	Tidak ada
9.	Sertifikat	Ada	Ada	Tidak ada

Dari Tabel 4.2 bisa dilihat perbandingan jumlah dan ketersediaan informasi pada setiap situs berbeda-beda. Untuk keperluan data *scraping* pada aplikasi mesin pencari informasi penjualan rumah diperlukan ketersediaan informasi yang sama pada setiap situs. Oleh karena itu, proses *scraping* akan mengambil data yang tersedia pada ketiga situs seperti judul/tautan url, gambar, kamar tidur, kamar mandi dan harga.

4.1.4 Analisis Kebutuhan Fungsional

Daftar kebutuhan terdiri dari 2 bagian yaitu kebutuhan fungsional dan kebutuhan non-fungsional. Pada daftar kebutuhan fungsional terdiri dari spesifikasi kebutuhan fungsional perangkat lunak bagi pengguna yaitu berupa kebutuhan fungsional, aktor yang menggunakan sistem, serta nama use case yang dapat dilihat pada tabel 4.3 berikut ini.

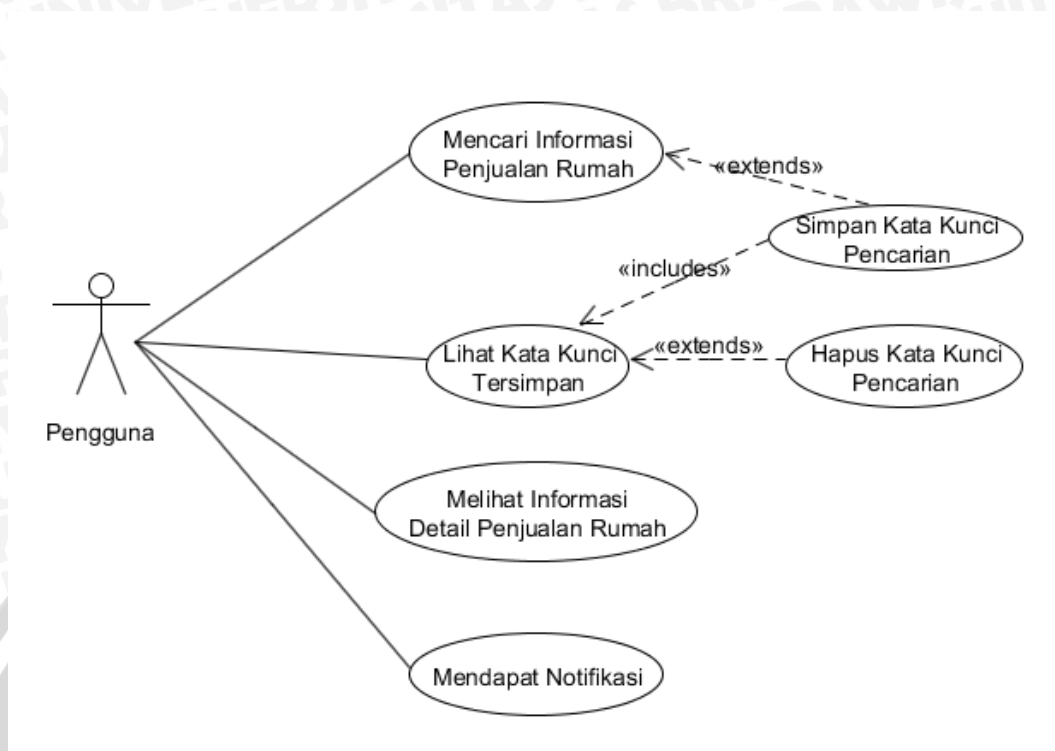
Tabel 4. 3 Daftar Kebutuhan Fungsional

No.	Kebutuhan Fungsional	Nama <i>Use Case</i>
1.	Aplikasi harus menyediakan fasilitas mencari informasi penjualan rumah.	Mencari Informasi Penjualan Rumah
2.	Aplikasi harus menyediakan fasilitas menyimpan kata kunci pencarian yang nantinya pada setiap kata kunci akan diterapkan <i>auto scraping</i> untuk notifikasi.	Simpan Kata Kunci Pencarian
3.	Aplikasi harus menyediakan fitur melihat informasi detail penjualan rumah dari situs asalnya.	Melihat Informasi Detail Penjualan Rumah
4.	Aplikasi harus menyediakan fasilitas lihat kata kunci tersimpan	Lihat Kata Kunci Tersimpan
5.	Aplikasi harus menyediakan fasilitas hapus kata kunci tersimpan	Hapus Kata Kunci Tersimpan
6.	Aplikasi harus menyediakan fasilitas notifikasi pengguna.	Mendapat Notifikasi

Selanjutnya daftar kebutuhan fungsional akan dijabarkan lebih lanjut dengan *use case* dan skenario *use case*.

4.1.4.1 Diagram *Use Case*

Gambar 4.6 merupakan diagram *use case* dari aplikasi ini yang menunjukkan spesifikasi fungsionalitas yang dibutuhkan oleh sistem dari aktor.



Gambar 4. 6 Diagram *Use Case* Aplikasi Mobile Mesin Pencari Penjualan Rumah

4.1.4.2 Skenario *Use Case*

Skenario *use case* berisi penjabaran dari tiap-tiap *use case* yang ada pada Gambar 4.6. Pada skenario *use case*, diberikan uraian nama *use case*, aktor yang terlibat, tujuan, deskripsi, langkah-langkah *use case*, kondisi awal yang harus dipenuhi (*pre-condition*), dan kondisi akhir yang diharapkan (*post-condition*). Skenario *use case* dapat dilihat pada Tabel 4.4 – Tabel 4.9 di bawah ini

Tabel 4. 4 Skenario *Use case* Mencari Informasi Penjualan Rumah

Nama <i>use case</i>	Mencari Informasi Penjualan Rumah
Tujuan	Mencari informasi penjulalan rumah dari sistem.
Deskripsi	<i>Use case</i> ini menjelaskan bagaimana user mencari informasi penjualan rumah yang diinginkan berdasarkan

	kata kunci pencarian dan penyaringan yang diinginkan.
Aliran Utama	
1. Pengguna memasukkan kata kunci pencarian dan penyaringan. 2. Sistem menampilkan informasi penjualan rumah berupa daftar iklan dari gabungan 3 situs. Daftar iklan terdiri dari masing-masing berupa judul iklan, gambar, harga, lokasi, kamar tidur dan kamar mandi.	
Variant	
1a. Memilih kata kunci pencarian dan penyaringan 1a.1. Pengguna memilih kata kunci pencarian dan penyaringan dari daftar riwayat. 1a.2 Lanjut ke langkah no. 2	
Exception	
1a. Pengguna tidak mengisi kata kunci 1a.1. Sistem menampilkan pesan kesalahan. 1a.2. Pengguna memasukkan kata kunci pencarian 1a.3. Lanjut ke langkah no. 2	

Tabel 4. 5 Skenario *Use case* Simpan Kata Kunci Pencarian

Nama use case	Simpan Kata kunci Pencarian
Tujuan	Menyimpan kata kunci pencarian ke basis data.
Deskripsi	<i>Use case</i> ini menjelaskan bagaimana menyimpan kata kunci pencarian pada basis data.
Aliran Utama	
1. Pengguna memutuskan untuk menyimpan kata kunci pencarian. 2. Sistem menyimpan kata kunci pencarian.	
Variant	
-	

Exception
-

Tabel 4. 6 Skenario *Use case* Melihat Informasi Detail Penjualan Rumah

Nama use case	Melihat Informasi Detail Penjualan Rumah
Tujuan	Melihat informasi detail penjualan rumah.
Deskripsi	<i>Use case</i> ini menjelaskan bagaimana pengguna melihat informasi detail penjualan rumah yang diinginkan dari situs asalnya.
Aliran Utama	
1. Pengguna memilih salah satu iklan dari daftar iklan yang telah ditampilkan. Daftar iklan terdiri dari masing-masing berupa judul iklan, gambar, harga, lokasi, kamar tidur dan kamar mandi. 2. Sistem menampilkan informasi detail penjualan rumah yang sesuai dengan pilihan.	
Variant	
-	
Exception	
-	

Tabel 4. 7 Skenario *Use case* Lihat Kata kunci Pencarian

Nama use case	Lihat Kata kunci Tersimpan
Tujuan	Melihat kata kunci pencarian yang tersimpan.
Deskripsi	<i>Use case</i> ini menjelaskan bagaimana melihat kata kunci pencarian yang tersimpan.

Aliran Utama
1. Pengguna membuka halaman riwayat. 2. Sistem menampilkan halaman riwayat yang berisi daftar kata kunci pencarian yang telah tersimpan.
Variant
-
Exception
1a. Pengguna tidak menyimpan kata kunci pada halaman daftar iklan 1a.1. Sistem menampilkan pesan pemberitahuan bahwa tidak ada kata kunci tersimpan.

Tabel 4. 8 Skenario *Use case* Hapus Kata kunci Pencarian

Nama use case	Lihat Kata kunci Tersimpan
Tujuan	Menghapus kata kunci pencarian yang tersimpan.
Deskripsi	<i>Use case</i> ini menjelaskan bagaimana menghapus kata kunci pencarian yang tersimpan sebelumnya.
Aliran Utama	
1. Pengguna masuk halaman Riwayat. 2. Sistem menampilkan halaman riwayat yang berisi daftar kata kunci pencarian yang tersimpan. 3. Pengguna memilih kata kunci pencarian yang ingin dihapus. 4. Sistem menghapus kata kunci pencarian yang dipilih.	
Variant	
-	
Exception	
-	

Tabel 4. 9 Skenario *Use case* Mendapat Notifikasi

Nama use case	Lihat Kata kunci Tersimpan
----------------------	-----------------------------------

Tujuan	Mendapatkan notifikasi mengenai informasi penjualan rumah terbaru.
Deskripsi	<i>Use case</i> ini menjelaskan bagaimana mendapatkan notifikasi mengenai informasi penjualan rumah terbaru.
Aliran Utama	
1. Pengguna yang sebelumnya telah menyimpan kata kunci menuju “setting notifikasi”. 2. Pada halaman “setting notifikasi” pengguna mengaktifkan notifikasi dan mengeset waktu apabila diperlukan. 3. Sistem melakukan <i>scraping</i> otomatis berdasarkan kata kunci yang tersimpan. 4. Informasi penjualan rumah baru yang didapatkan dinotifikasi ke pengguna.	
Variant	
-	
Exception	
1a. Pengguna belum menyimpan kata kunci pencarian 1a.1. Pengguna menyimpan kata kunci pencarian saat berada pada halaman daftar iklan. 1a.2. Lanjut ke langkah no. 1	

4.1.5 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional adalah analisis untuk mengetahui spesifikasi yang dibutuhkan oleh sistem. Spesifikasi kebutuhan non fungsional pada aplikasi ini akan dijabarkan pada tabel 4.10 dibawah ini.

Tabel 4. 10 Spesifikasi Kebutuhan Non Fungsional

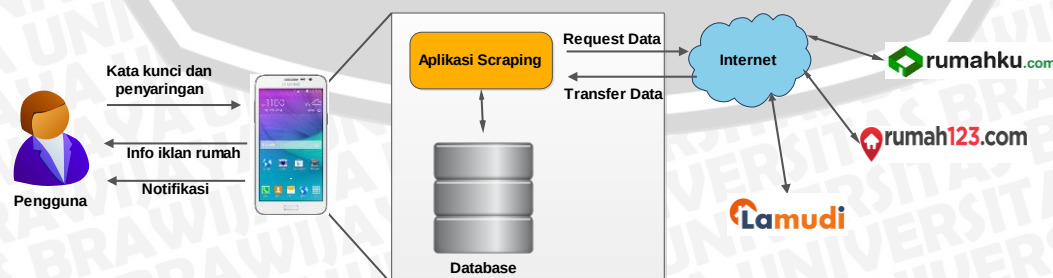
Parameter	Deskripsi Kebutuhan
Usability	Aplikasi harus dapat digunakan dengan mudah oleh pengguna. Pengguna dapat mengetahui dan menggunakan fitur-fitur yang ada pada aplikasi mesin pencari informasi penjualan rumah.

4.2 Perancangan Sistem

Perancangan sistem meliputi 8 tahap yaitu perancangan arsitektur sistem, perancangan aktifitas, perancangan sequence, alur proses *scraping*, alur proses notifikasi, perancangan skema basis data, perancangan kelas, dan perancangan antarmuka aplikasi. Tahap perancangan arsitektur sistem menjelaskan arsitektur sistem yang akan dibuat dan dikembangkan. Tahap perancangan basis data menjelaskan bagaimana data disimpan dan relasinya masing-masing dan tahap perancangan antarmuka aplikasi menjelaskan tampilan yang digunakan.

4.2.1 Perancangan Arsitektur Sistem

Perancangan arsitektur sistem dapat digambarkan seperti pada gambar 4.7. Pertama-tama pengguna mengakses aplikasi informasi penjualan rumah melalui perangkat bergerak Android. Aplikasi akan memproses pencarian informasi dengan cara *Scraping data* untuk mengambil informasi dari 3 situs penjualan rumah yang ada di internet. Hasil *scraping* akan ditampilkan ke pengguna sekaligus menyimpannya ke basis data untuk keperluan fitur notifikasi. Semua proses data dilakukan pada sisi *client* atau dalam hal ini yaitu perangkat bergerak itu sendiri.



Gambar 4. 7 Arsitektur Sistem

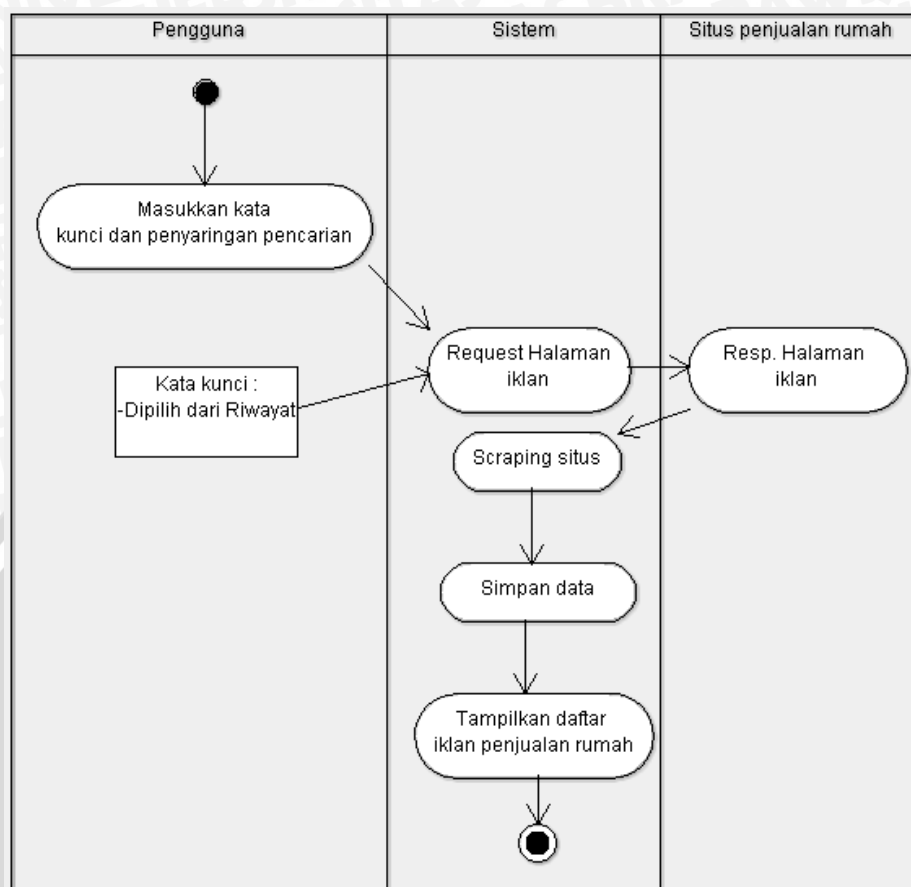
Ketika memasuki halaman awal aplikasi, pengguna memasukkan kata kunci pencarian dan penyaringan lalu menekan tombol cari. Aplikasi akan menindaklanjuti dengan menampilkan informasi berupa daftar tautan penjualan rumah dari hasil scraping pada 3 situs. Saat pengguna memilih salah satu informasi penjualan rumah dari daftar, aplikasi akan menuju tautan situs yang asli untuk menampilkan informasi detail penjualan rumah.

Selain untuk pencarian, kata kunci yang dimasukkan oleh pengguna juga akan disimpan ke basis data untuk dipakai pada fitur pemberitahuan informasi terbaru atau notifikasi. Fitur ini membantu pengguna untuk mengetahui informasi penjualan rumah terbaru dengan cara aplikasi melakukan *scraping* otomatis dengan interval waktu tertentu yang selanjutnya akan dinotifikasi ke pengguna dan menampilkan informasi baru tersebut dengan tampilan berbeda dengan informasi yang lama.

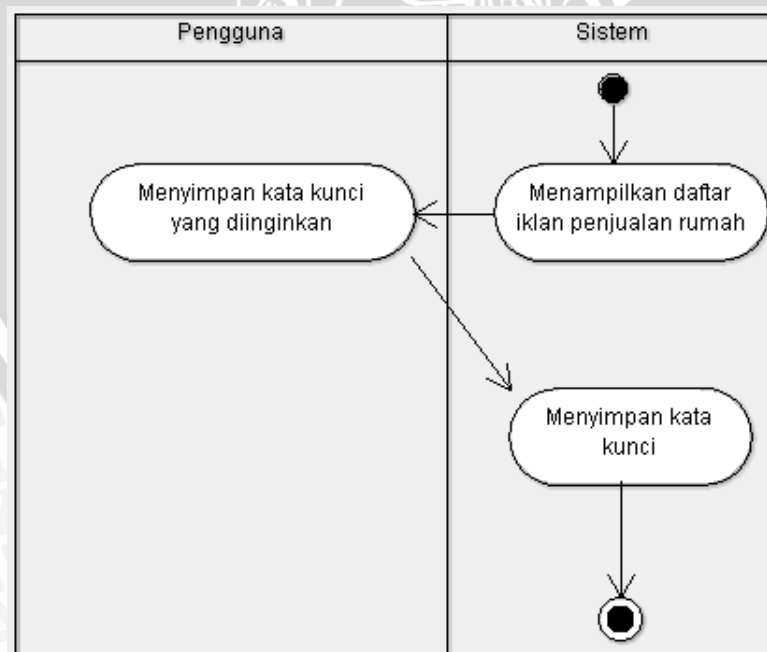
4.2.2 Perancangan Aktifitas

Perancangan aktifitas memodelkan aktifitas antara pengguna dengan sistem yang berjalan berdasarkan pada skenario *use case*. Aktifitas dipresentasikan menggunakan Diagram *Activity* yang bisa dilihat pada Gambar 4.8 – 4.13.

Gambar 4.8 menunjukkan aktifitas mencari informasi penjualan rumah yang dilakukan oleh pengguna, sistem dan situs penjualan rumah. Deskripsi *activity diagram* sesuai dengan skenario pada *Use Case* Mencari Informasi Penjualan Rumah dalam Tabel 4.4. Pengguna melakukan proses pencarian dengan memasukkan kata kunci pencarian dan penyaringan. Alternatif lain pengguna juga bisa memilih kata kunci pencarian yang telah disimpan sebelumnya. Ketika pengguna telah memasukkan kata kunci maupun memilihnya pada halaman Riwayat sistem akan melakukan req. HTML yang selanjutnya akan di respon oleh situs penjualan rumah berupa HTML. Kemudian sistem akan memulai melakukan *scraping data*. Hasil *scraping* akan ditampilkan ke pengguna sekaligus disimpan untuk keperluan fitur notifikasi. Alur proses *scraping* dapat dilihat pada sub bab 4.4.4

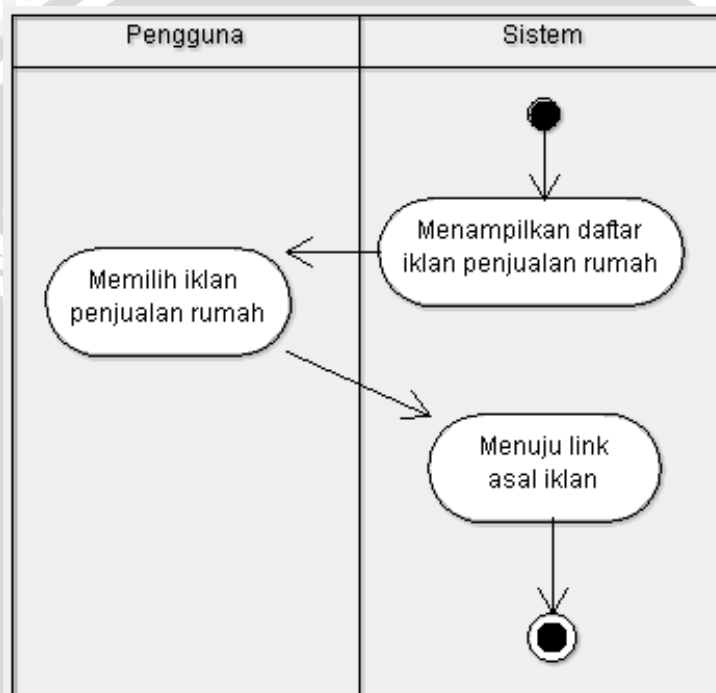


Gambar 4. 8 Activity Diagram Mencari Informasi Penjualan Rumah



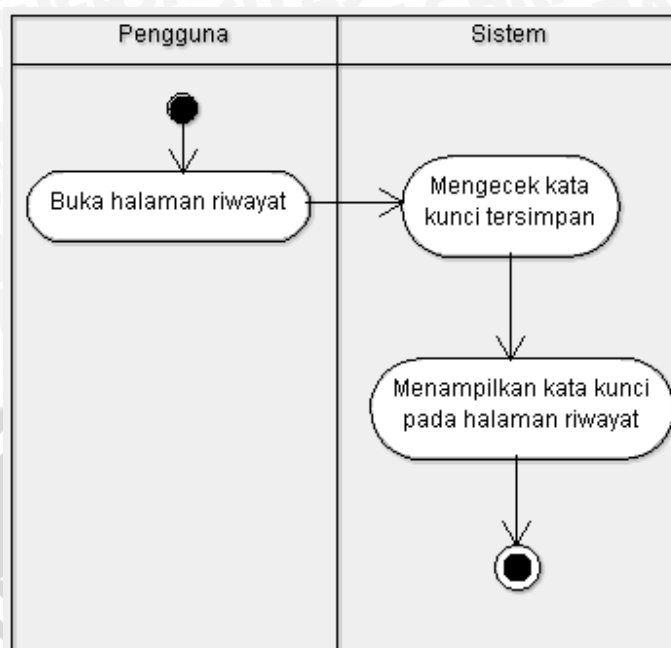
Gambar 4. 9 Activity Diagram Simpan Kata Kunci Pencarian

Gambar 4.9 menunjukkan aktifitas simpan kata kunci pencarian yang dilakukan oleh pengguna dan sistem. Deskripsi *activity diagram* sesuai dengan skenario pada *Use Case* Simpan Kata Kunci Pencarian dalam Tabel 4.5. Sistem menampilkan daftar informasi penjualan rumah dari hasil pencarian sebelumnya. Pengguna menyimpan kata kunci yang diinginkan maka sistem akan menyimpan kata kunci pencarian pada basis data. Ketika kata kunci disimpan akan dilakukan proses notifikasi. Alur proses notifikasi dapat dilihat pada sub bab 4.2.4



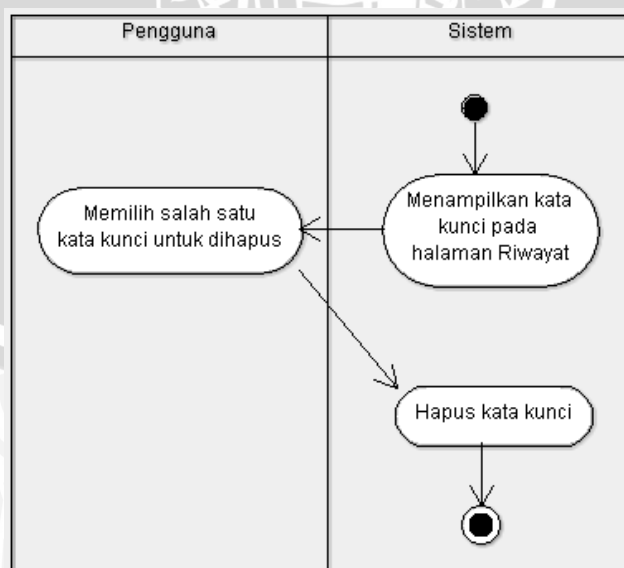
Gambar 4. 10 Activity Diagram Melihat Informasi Detail Penjualan Rumah

Gambar 4.10 menunjukkan aktifitas melihat informasi detail penjualan rumah yang dilakukan oleh pengguna dan sistem. Deskripsi *activity diagram* sesuai dengan skenario pada *Use Case* Melihat Informasi Detail Penjualan Rumah dalam Tabel 4.6. Pengguna memilih salah satu dari daftar informasi penjualan rumah yang ada. Sistem akan menuju tautan asal dari informasi tersebut dan menampilkannya.



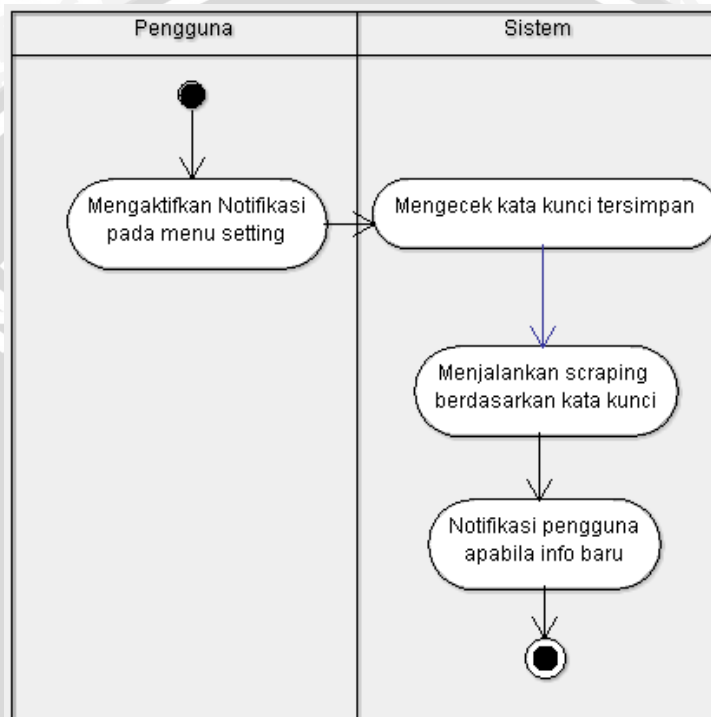
Gambar 4. 11 Activity Diagram Lihat Kata Kunci Tersimpan

Gambar 4.11 menunjukkan aktifitas lihat kata kunci tersimpan yang dilakukan oleh pengguna dan sistem. Deskripsi *activity diagram* sesuai dengan skenario pada *Use Case* Lihat Kata Kunci tersimpan dalam Tabel 4.7. Pengguna membuka halaman Riwayat dengan memilih menu Riwayat. Sistem akan mengecek apakah ada kata kunci tersimpan sebelumnya. Apabila ada akan menampilkan kata kunci pada halaman Riwayat.



Gambar 4. 12 Activity Diagram Hapus Kata Kunci Tersimpan

Gambar 4.12 menunjukkan aktifitas hapus kata kunci tersimpan yang dilakukan oleh pengguna dan sistem. Deskripsi *activity diagram* sesuai dengan skenario pada *Use Case* Mencari Informasi Penjualan Rumah dalam Tabel 4.8 Pengguna memilih salah satu dari kata kunci tersimpan pada halaman Riwayat lalu menekan tombol delete. Sistem akan menghapus kata kunci dari basis data.



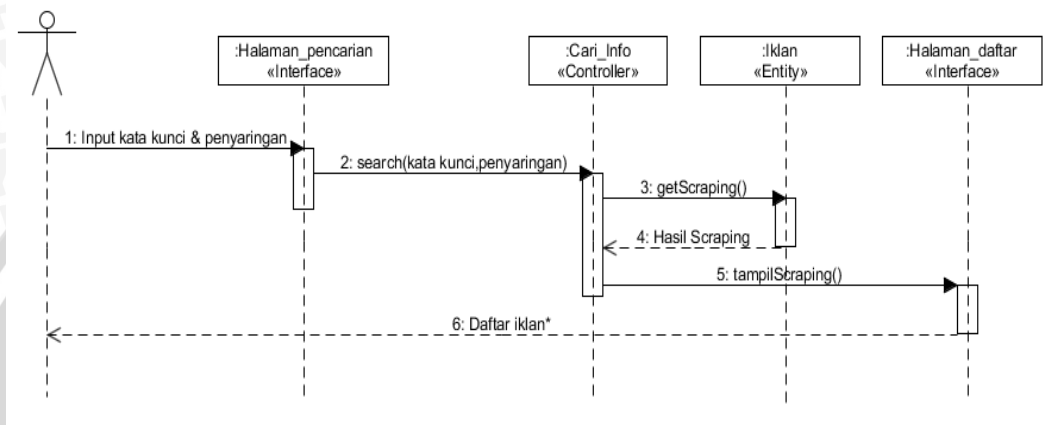
Gambar 4. 13 *Activity Diagram* Mendapat Notifikasi

Gambar 4.13 menunjukkan aktifitas mendapat notifikasi yang dilakukan oleh pengguna dan sistem. Deskripsi *activity diagram* sesuai dengan skenario pada *Use Case* Mencari Informasi Penjualan Rumah dalam Tabel 4.9. Pengguna terlebih dahulu mengaktifkan notifikasi pada menu “setting notifikasi”. Sistem akan merespon dengan terlebih dahulu mengecek kata kunci yang tersimpan. Kata kunci yang tersimpan akan dipakai untuk melakukan *scraping* informasi penjualan rumah secara otomatis. Apabila terdapat informasi baru, sistem akan mengirimkan notifikasi kepada pengguna.

4.2.3 Perancangan *Sequence*

Sequence diagram adalah suatu diagram yang menggambarkan interaksi antar obyek dan mengindikasikan komunikasi di antara obyek-obyek tersebut. *Sequence diagram* untuk aplikasi mesin pencari informasi penjualan rumah dapat dilihat pada Gambar 4.14 – Gambar 4.19 di bawah ini :

1. Mencari Informasi Penjualan Rumah

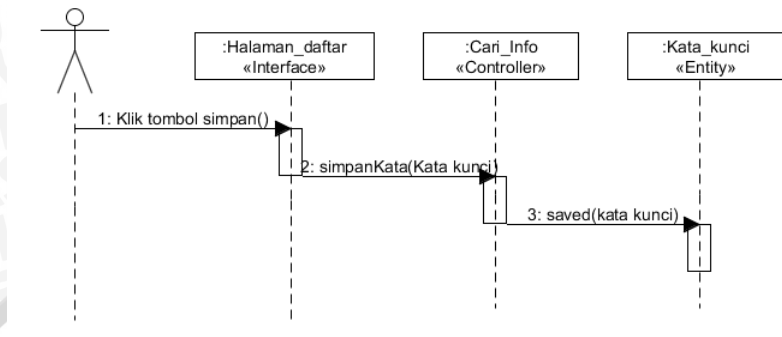


Daftar iklan: (Judul iklan, gambar, link URL, alamat, harga, kamar tidur dan kamar mandi)*

Gambar 4. 14 *Sequence Diagram* Mencari Informasi Penjualan Rumah

Gambar 4.14 merupakan diagram *sequence* mencari informasi penjualan rumah yang akan menampilkan proses masukkan dari pengguna sampai mendapat informasi penjualan rumah. Pengguna terlebih dahulu memasukkan kata pencarian pada halaman utama/pencarian yang ditampilkan oleh objek *interface* Halaman_pencarian yang selanjutnya akan mengirimkan variabel masukkan ke objek *controller*:Cari_info melalui method *search()*. Kemudian objek *controller* Cari_info akan melakukan *scraping*. Hasil *scraping* yang telah didapat akan dikirimkan kembali ke objek *controller* Cari_info oleh *entity* informasi. Pada akhirnya *controller* Cari_info akan melakukan perintah menampilkan informasi ke *interface* Halaman_daftar untuk diperlihatkan ke pengguna.

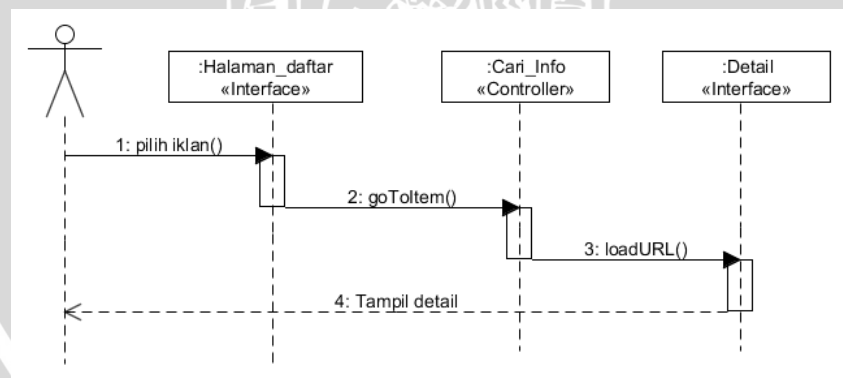
2. Simpan Kata Kunci Pencarian



Gambar 4. 15 *Sequence Diagram* Simpan Kata Kunci Pencarian

Gambar 4.15 merupakan diagram *sequence* simpan kata kunci pencarian yang akan menjelaskan bagaimana pengguna menyimpan kata kunci pencarian yang nantinya akan dipakai sebagai notifikasi. Pengguna terlebih dahulu menekan tombol “simpan” pada *Interface* halaman_daftar atau daftar informasi penjualan rumah. Data kata kunci yang dipilih akan dikirim ke objek *controller* Cari_info yang akan memproses perintah *saved()* untuk menyimpannya ke basis data.

3. Melihat Detail Informasi Penjualan Rumah

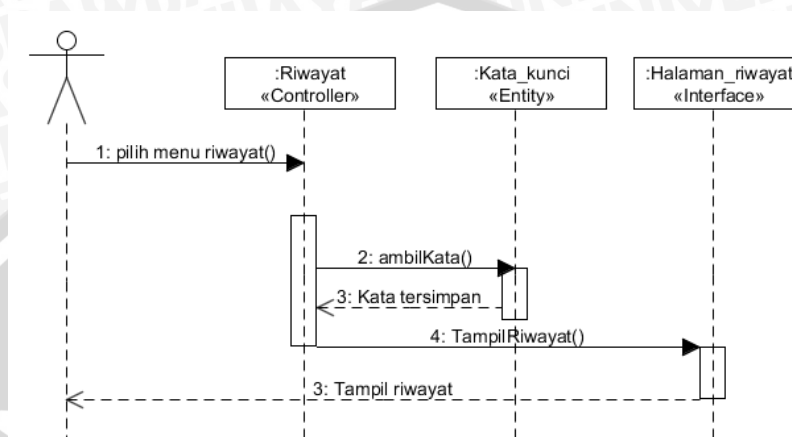


Gambar 4. 16 *Sequence Diagram* Melihat Detail Informasi Penjualan Rumah

Gambar 4.16 merupakan diagram *sequence* melihat detail informasi penjualan rumah yang akan menampilkan link asal dari informasi penjualan rumah. Pengguna terlebih dahulu memilih salah satu informasi yang diinginkan pada *interface* halaman_daftar. Informasi yang terpilih akan diproses oleh *controller* Cari_info untuk selanjutnya dilakukan perintah *loadURL()* ke *interface* Detail.

Akhirnya *interface* Detail akan menampilkan alamat asal informasi pada pengguna menggunakan *browser* eksternal perangkat bergerak.

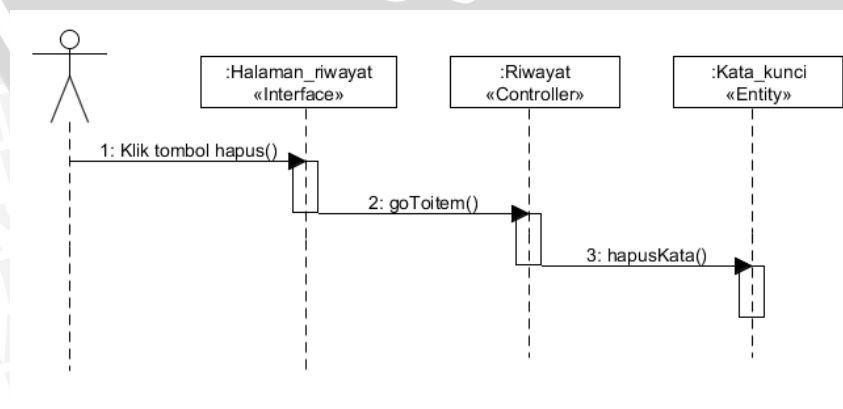
4. Lihat Kata Kunci Tersimpan



Gambar 4. 17 *Sequence Diagram* Lihat Kata Kunci Pencarian

Gambar 4.17 merupakan diagram *sequence* lihat kata kunci pencarian yang akan menjelaskan proses melihat kata kunci pencarian yang telah disimpan oleh user. Pengguna terlebih dahulu memilih menu riwayat . Objek *controller* Riwayat akan memproses perintah pengguna dengan mengambil data kata kunci pada basis data. Kata kunci tersimpan akan ditampilkan dalam bentuk riwayat pada *interface* halaman_riwayat oleh *controller* Riwayat untuk pada akhirnya ditampilkan ke pengguna.

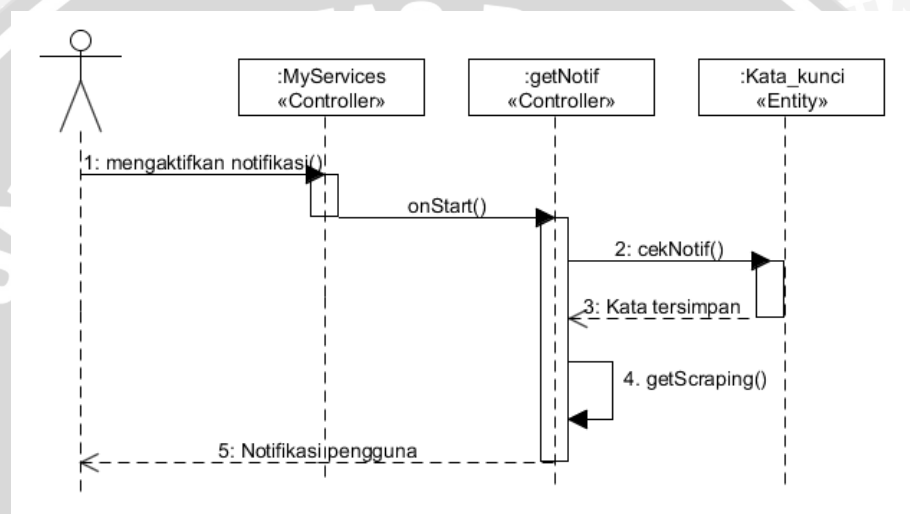
5. Hapus Kata Kunci Tersimpan



Gambar 4. 18 *Sequence Diagram* Hapus Kata Kunci Pencarian

Gambar 4.18 merupakan diagram *sequence* hapus kata kunci pencarian yang akan menjelaskan proses menghapus kata kunci pencarian yang telah disimpan oleh user. Pengguna terlebih dahulu menekan tombol “hapus” pada halaman riwayat yang disediakan oleh objek *interface* halaman_riwayat. Kata yang telah terpilih akan diproses oleh *controller* Riwayat untuk selanjutnya dihapus dari basis data.

6. Mendapat Notifikasi

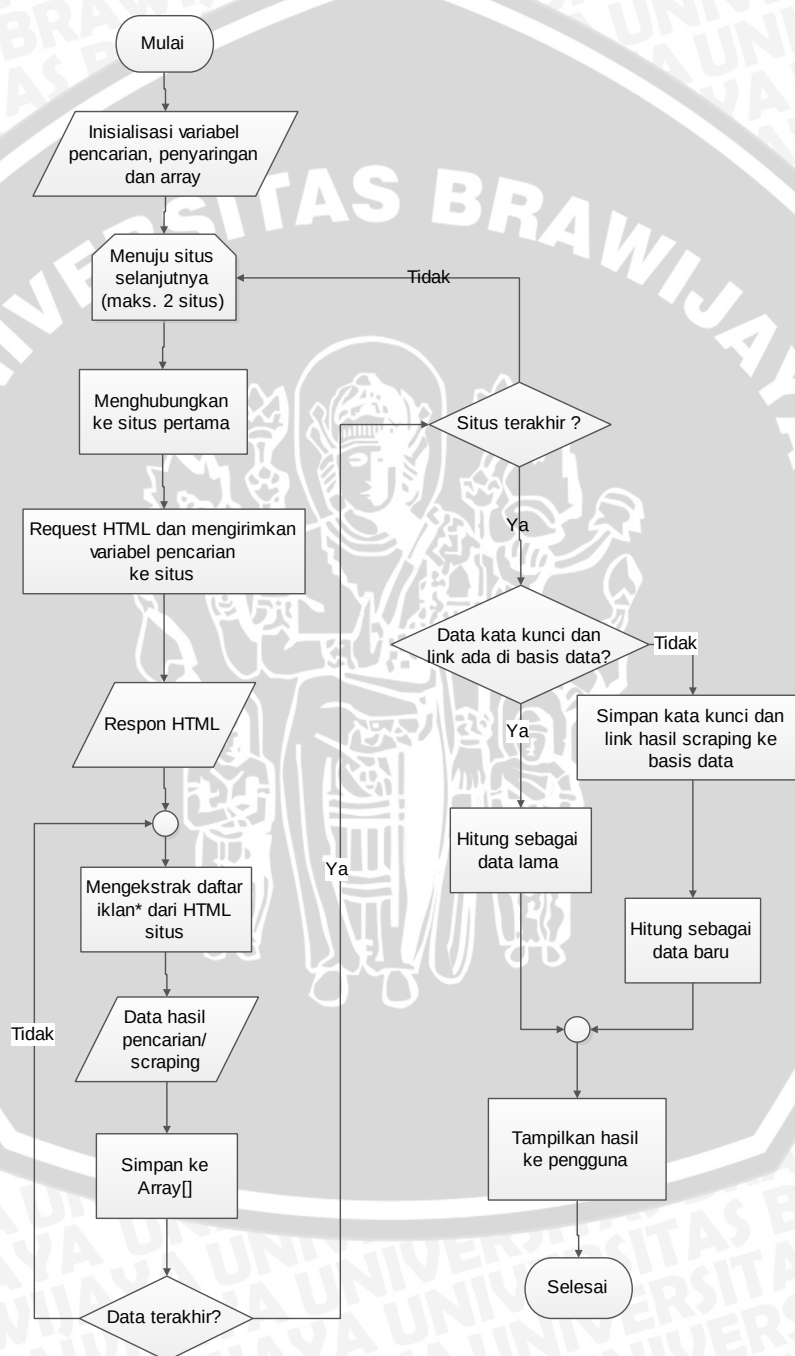


Gambar 4. 19 *Sequence Diagram* Mendapat Notifikasi

Gambar 4.19 merupakan diagram *sequence* mendapat notifikasi yang akan menjelaskan proses bagaimana pengguna bisa mendapatkan notifikasi mengenai informasi penjualan rumah terbaru. Pengguna terlebih dahulu mengaktifkan notifikasi pada menu “setting notifikasi” dan megeset interval waktu apabila diperlukan. Selanjutnya sistem akan mengaktifkan *background process* sesuai interval waktu yang ditentukan. Selanjutnya sistem akan mengecek kata kunci yang telah tersimpan pada basis data. Kata kunci yang tersimpan akan digunakan untuk melakukan *scraping* secara otomatis. Informasi penjualan rumah baru yang didapatkan akan dinotifikasi ke pengguna.

4.2.4 Alur proses *Scraping*

Alur proses *scraping* menjelaskan urutan proses *scraping* data secara mendetail dan hubungannya antar proses dengan proses lain dalam sistem. Diagram alur proses *scraping data* dapat dilihat pada Gambar 4.20 di bawah ini.



Daftar iklan: (Judul iklan, gambar, link URL, alamat, harga, kamar tidur dan kamar mandi)*

Gambar 4. 20 Alur proses *scraping*

Dari Gambar 4.20 alur proses *web scraping* akan dimulai dengan Inisialisasi variabel kata pencarian dan penyaringan, lalu sistem akan mencoba membangun koneksi dan mengirimkan *request* HTML beserta variabel pencarian ke situs yang pertama.. Selanjutnya, situs akan mengirimkan respon HTML. Kemudian sistem akan mulai mencari data penjualan rumah yang dibutuhkan.

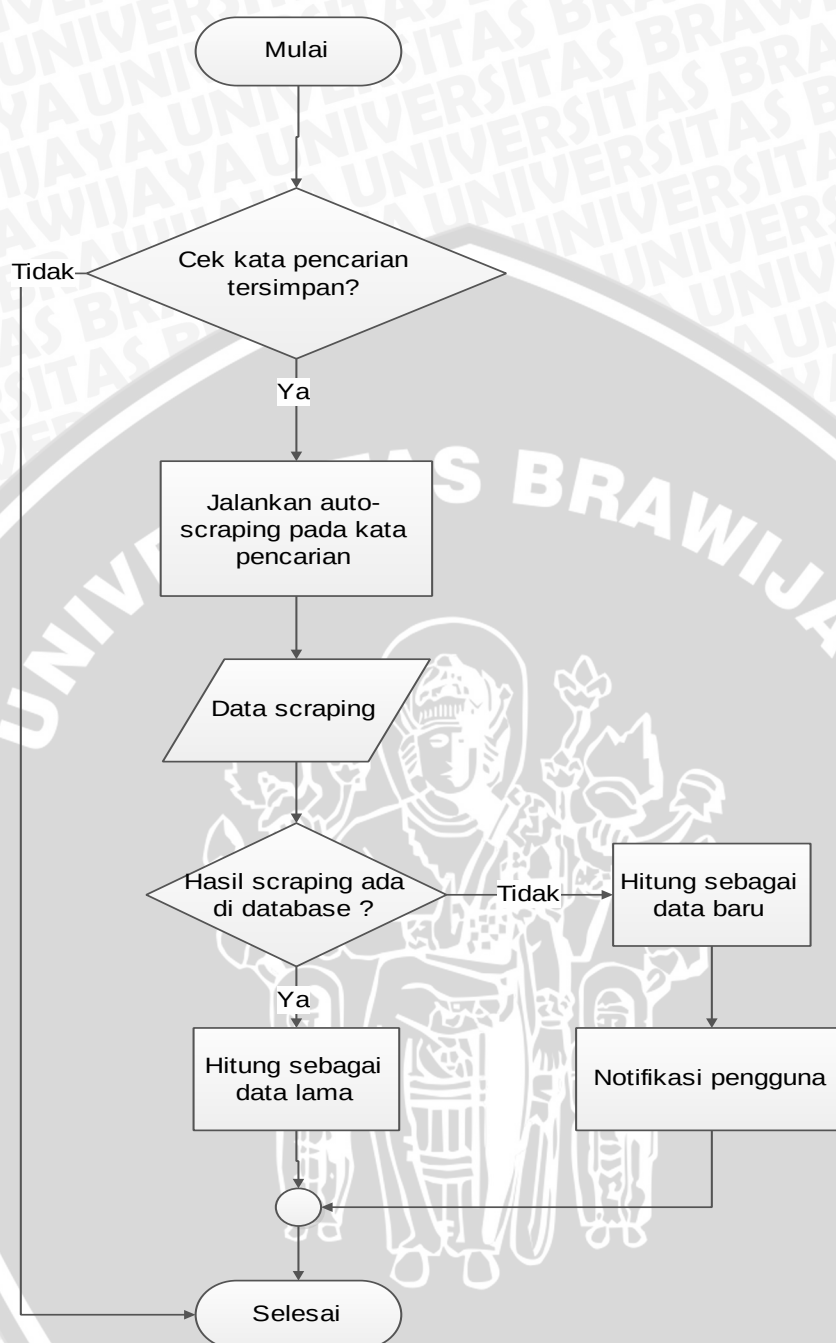
Data yang telah didapat akan disimpan dalam sebuah array. Begitu seterusnya dilakukan secara berulang hingga semua data penjualan rumah pada situs pertama didapat semua. Apabila semua data pada situs pertama sudah diambil, sistem akan mengulangi lagi ke fase membangun koneksi pada situs kedua seterusnya hingga maksimal pada situs ketiga. Selanjutnya, apabila *scraping* telah dilakukan pada ketiga situs, sistem akan mengecek kata kunci dan link URL hasil *scraping* pada basis data.

Apabila tidak ada, maka akan disimpan ke basis data dan dihitung sebagai data baru atau belum terbaca yang akan digunakan sebagai informasi apabila kata kunci dipakai untuk notifikasi. Sedangkan apabila data sudah ada di basis data, maka akan dihitung sebagai data lama atau sudah terbaca. Terakhir, sistem akan menampilkan hasil *scraping* ke pengguna dari data yang terkumpul dalam array.

4.2.5 Alur proses Notifikasi

Alur proses *scraping* menjelaskan urutan proses notifikasi data secara mendetail dan hubungannya antar proses dengan proses lain dalam sistem. Diagram alur proses notifikasi dapat dilihat pada Gambar 4.21

Dari Gambar 4.21 alur proses notifikasi akan dimulai mengecek apabila kata kunci pencarian dan penyaringan yang berstatus notifikasi aktif pada basis data. Bila ada maka akan mengaktifkan *auto-scraping* atau *scraping* otomatis pada kata kunci pencarian tersebut. *Auto-scraping* akan menjalankan *scraping* data secara otomatis dengan rentan waktu tertentu. Setelah hasil *scraping* didapat akan dicek di basis data apakah sudah ada atau tidak. Apabila sudah ada, maka akan dihitung sebagai data lama atau sudah terbaca. Sedangkan apabila tidak ada di basis data akan disimpan dan dihitung sebagai data baru untuk dinotifikasi ke pengguna.

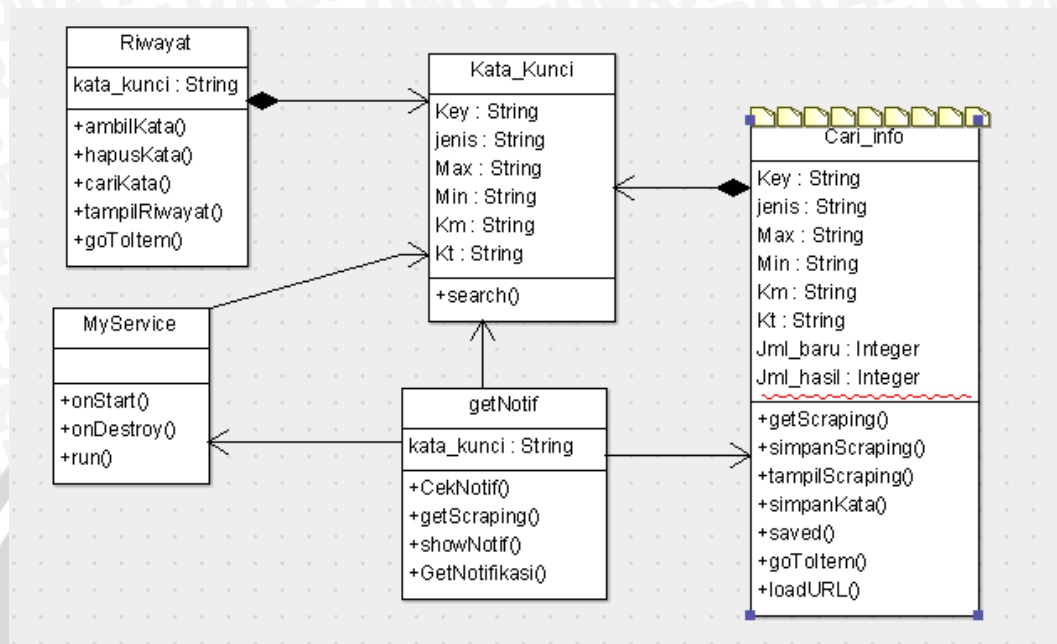


Gambar 4. 21 Alur proses Notifikasi

4.2.6 Perancangan Kelas

Class Diagram adalah diagram yang digunakan untuk menampilkan beberapa kelas yang ada dalam sistem perangkat lunak yang akan dikembangkan. Class diagram menunjukkan hubungan antar class dalam sistem yang sedang dibangun dan bagaimana mereka saling berkolaborasi untuk mencapai suatu tujuan.

Berikut ini digambarkan class diagram dari aplikasi mesin pencari informasi penjualan rumah yang bisa dilihat pada Gambar 4.22 di bawah ini.



Gambar 4. 22 Class Diagram

Berdasarkan pada Gambar 4.22 di atas, dapat dijelaskan deskripsi dan fungsi masing-masing *class* sebagai berikut :

1. *Class* Kata_Kunci

Tabel 4. 11 Deskripsi Class Kata_Kunci

Nama Class : Kata_Kunci
Deskripsi : <i>Class</i> ini merupakan <i>class</i> yang pertama kali dieksekusi ketika aplikasi dijalankan oleh pengguna dan akan menampilkan masukkan kata pencarian dan penyaringan untuk mencari informasi penjualan rumah.
Nama Method : search()
Fungsi : Method ini adalah method yang berisi deklarasi komponen-komponen penyusun <i>layout</i> utama yang ditampilkan pada pengguna

2. *Class Cari_info*

Tabel 4. 12 Deskripsi *Class Cari_info*

Nama Class : Cari_info()
Deskripsi : <i>Class</i> ini merupakan <i>class</i> yang akan menampilkan daftar informasi penjualan rumah hasil dari <i>scraping</i> pada situs penjualan rumah. <i>Class</i> ini memiliki hubungan komposit terhadap <i>class</i> kata_kunci didasarkan pada masukkan kata pencarian dan penyaringan.
Nama Method : getScraping() Fungsi : Method ini berfungsi untuk melakukan <i>scraping</i> pada situs penjualan rumah
Nama Method : simpanScraping() Fungsi : Method ini berfungsi untuk melakukan penyimpanan hasil <i>scraping</i> pada situs penjualan rumah.
Nama Method : tampilScraping() Fungsi : Method ini berfungsi untuk menampilkan hasil <i>scraping</i> pada pengguna.
Nama Method : simpankata () Fungsi : Method ini berfungsi untuk menyimpan kata pencarian yang digunakan untuk <i>scraping</i> .
Nama Method : saved() Fungsi : Method ini berfungsi untuk menyimpan kata kunci ke basis data.
Nama Method : goToItem() Fungsi : Method ini berfungsi untuk mengarahkan ke method lain apabila ditekan oleh pengguna.
Nama Method : loadURL() Fungsi : Method ini berfungsi untuk membuka halaman asal situs.

3. Class Riwayat

Tabel 4. 13 Deskripsi Class Riwayat

Nama Class : Riwayat()
Deskripsi : Class ini merupakan class yang menangani kata pencarian yang tersimpan. Class ini memiliki hubungan komposit dengan class kata_kunci yang dimana kata pencarian yang tersimpan berasal dari class tersebut.
Nama Method : ambilKata () Fungsi : Method ini digunakan untuk mengambil data yang tersimpan pada basis data.
Nama Method : hapusKata() Fungsi : Method ini digunakan untuk menghapus kata pencarian yang tersimpan dari basis data.
Nama Method : cariKata() Fungsi : Method ini digunakan untuk mencari informasi penjualan rumah terbaru melalui kata pencarian yang telah tersimpan .
Nama Method : TampilRiwayat() Fungsi : Method ini digunakan untuk menampilkan daftar kata kunci pencarian tersimpan .
Nama Method : goToItem() Fungsi : Method ini berfungsi untuk mengarahkan ke method lain apabila ditekan oleh pengguna.

4. Class getNotif

Tabel 4. 14 Deskripsi Class getNotif

Nama Class : getNotif()
Deskripsi : Class ini merupakan class yang digunakan untuk menampilkan notifikasi kepada pengguna terhadap informasi baru mengenai penjualan rumah. Class ini memiliki hubungan asosiasi dengan class

kata_kunci, cari_info dan service. Dimana masing-masing memiliki peran dalam menyediakan kata kunci, <i>scraping</i> , dan <i>background process</i>
Nama Method : cekNotif() Fungsi : Method ini digunakan untuk mengecek kata kunci yang tersimpan tersedia atau tidak.
Nama Method : getScraping() Fungsi : Method ini digunakan untuk menjalankan <i>scraping</i> data.
Nama Method : showNotif() Fungsi : Method ini digunakan untuk menampilkan notifikasi ke pengguna.
Nama Method : GetNotifikasi() Fungsi : Method ini merupakan method induk yang digunakan untuk mendapatkan notifikasi.

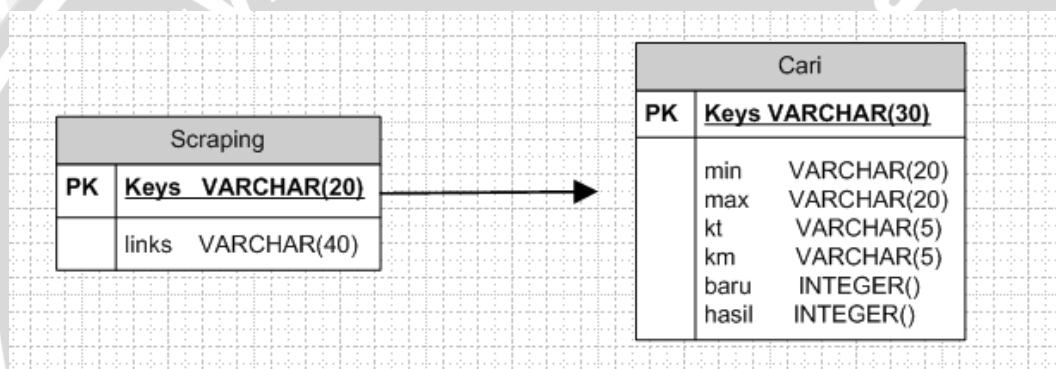
5. Class MyService

Tabel 4. 15 Deskripsi Class MyService

Nama Class : MyService()
Deskripsi : Class ini merupakan <i>class</i> yang digunakan <i>background process</i> dan pengaturan waktu interval notifikasi. Class ini memiliki hubungan asosiasi dengan <i>class</i> cari_info
Nama Method : onStart() Fungsi : Method ini digunakan untuk memulai proses dengan waktu intrval tertentu.
Nama Method : onDestroy() Fungsi : Method ini digunakan untuk menghentikan proses yang telah berjalan.
Nama Method : run() Fungsi : Method ini digunakan untuk menjalankan proses <i>auto scraping</i> .

4.2.7 Perancangan Skema Basis Data

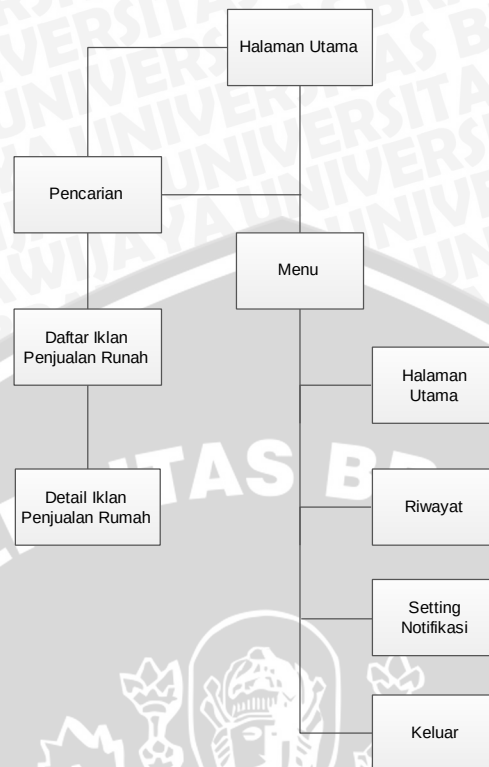
Perancangan basis data digunakan untuk menyimpan data-data yang didapat dari hasil proses *web scraping*. Perancangan basis data ini direpresentasikan dalam bentuk diagram logikal. Diagram logikal menyediakan tampilan grafis dari struktur sistem yang bertujuan untuk menganalisa struktur sistem melalui entitas dan relasi antar basis data. Berikut ini digambarkan skema basis data dari aplikasi mesin pencari informasi penjualan rumah yang bisa dilihat pada Gambar 4.23 di bawah ini.



Gambar 4. 23 Diagram logikal strktur basis data

4.2.8 Perancangan Antarmuka Aplikasi

Bagian ini menjelaskan mengenai perancangan antarmuka aplikasi mesin pencari informasi penjualan rumah. Aplikasi ini akan digunakan oleh pengguna untuk memudahkan pencarian dan mengetahui informasi berbagai penjualan rumah yang ada. Berikut adalah *sitemap* aplikasi yang ditunjukkan pada Gambar 4.24



Gambar 4. 24 Peta navigasi aplikasi

a. Halaman Utama

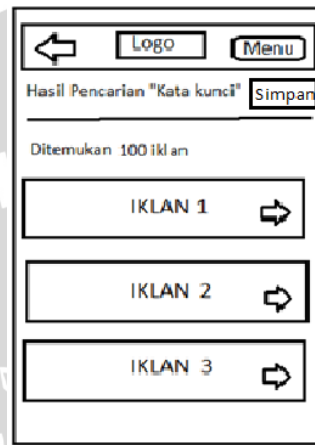
Halaman Utama adalah halaman awal saat pengguna membuka aplikasi pertama kali. Halaman utama berisi tampilan masukkan untuk pengguna berupa kata pencarian dan penyaringan. Gambar 4.25 dibawah ini adalah rancangan antarmuka halaman utama.

LOGO		MENU	
Dijual			
Kata Kunci Pencarian			
Harga Max	Harga Min		
K. Tidur	K. Mandi		
SEARCH			

Gambar 4. 25 Halaman Utama

b. Daftar Informasi Penjualan Rumah

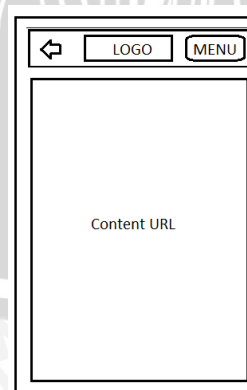
Halaman daftar penjualan rumah akan tampil saat pengguna menekan tombol cari pada halaman awal atau menekan daftar pencarian sebelumnya. Halaman ini berisi daftar informasi penjualan rumah. Gambar 4.26 adalah rancangan antarmuka halaman daftar informasi penjualan rumah.



Gambar 4. 26 Halaman daftar informasi penjualan rumah

c. Detail Informasi Penjualan Rumah

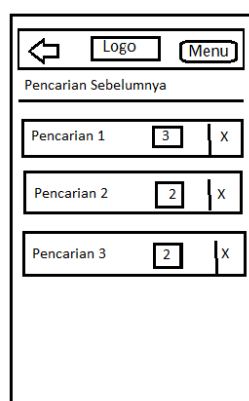
Halaman detail informasi penjualan rumah akan tampil saat pengguna menekan salah satu dari daftar informasi penjualan rumah. Halaman ini berisi tampilan web menggunakan URL dari informasi yang telah dipilih sebelumnya. Gambar 4.27 adalah rancangan antarmuka halaman detail informasi penjualan rumah.



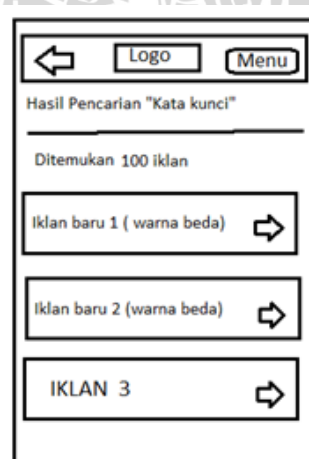
Gambar 4. 27 Halaman detail informasi penjualan rumah

d. Halaman Riwayat

Halaman riwayat akan tampil saat pengguna menekan pilihan riwayat pada menu. Halaman ini berisi daftar pencarian sebelumnya. Apabila salah satu pencarian ditekan maka akan memuat halaman daftar penjualan rumah. Pada informasi baru, warna daftar akan dibuat berbeda dari informasi yang lama. Gambar 4.28 adalah tampilan halaman riwayat dan Gambar 4.29 adalah tampilan halaman daftar penjualan rumah yang muncul saat memilih salah satu riwayat, tampilan informasi terbaru akan muncul dengan warna berbeda.



Gambar 4. 28 Halaman riwayat

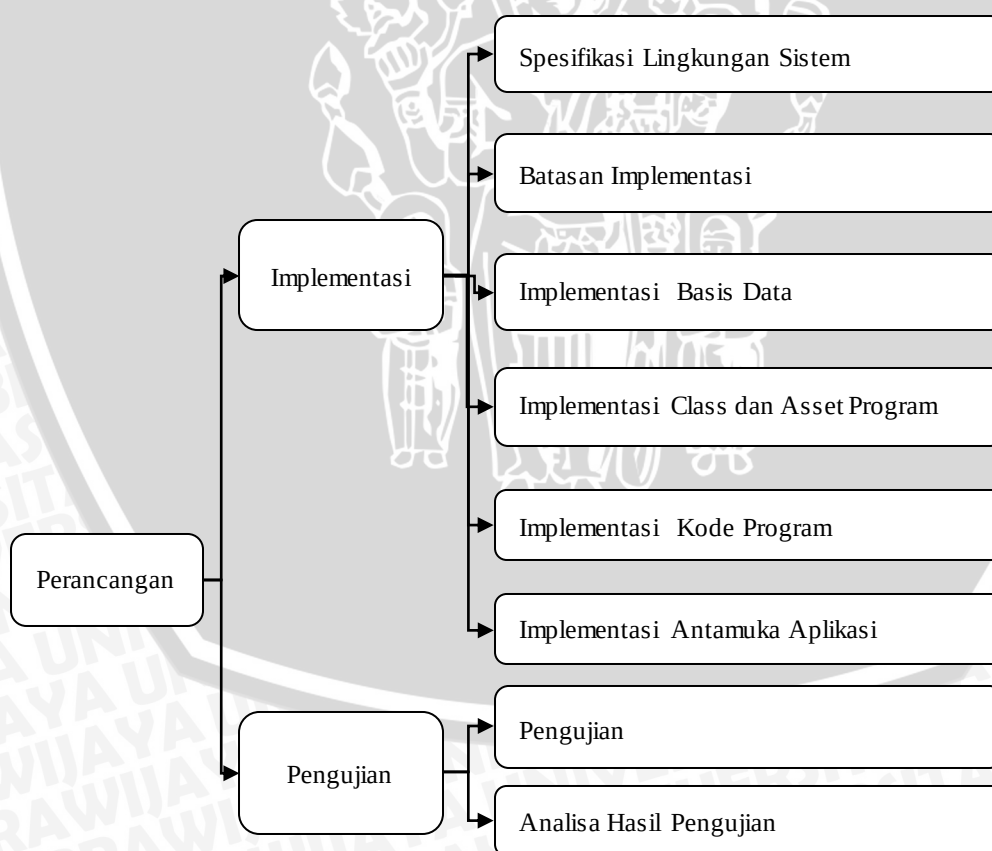


Gambar 4. 29 Halaman daftar penjualan rumah + informasi baru

BAB V

IMPLEMENTASI DAN PENGUJIAN

Bab ini akan membahas tahapan implementasi dan pengujian dari aplikasi mesin pencari informasi penjualan rumah berdasarkan hasil yang telah didapatkan dari tahapan analisis kebutuhan dan perancangan sistem. Tahapan implementasi dan pengujian terdiri dari tiga tahap. Tahap pertama adalah penjelasan tentang spesifikasi lingkungan pengembangan sistem, batasan-batasan dalam implementasi, implementasi basis data, implementasi *class* dan *assets* pada file program, implementasi kode program, dan implementasi antarmuka aplikasi. Tahap kedua adalah penjelasan tentang pengujian sistem yang terdiri dari pengujian validasi dan pengujian non-fungsional. Tahap ketiga adalah analisis hasil pengujian untuk mendapatkan kesimpulan berdasarkan hasil pengujian yang telah dilakukan. Gambar 5.1 adalah diagram pohon dari implementasi dan pengujian.



Gambar 5. 1 Diagram pohon implementasi dan pengujian

5.1 Implementasi

Tahap ini menjelaskan tentang spesifikasi lingkungan pengembangan sistem, batasan-batasan implementasi, implementasi basis data, implementasi *class* dan *assets* pada *file* program, implementasi kode program dan implementasi antarmuka aplikasi.

5.1.1 Spesifikasi Lingkungan Sistem

Spesifikasi aplikasi mesin pencari informasi penjualan rumah diimplementasikan dalam lingkungan implementasi yang terdiri dari spesifikasi perangkat keras dan spesifikasi perangkat lunak.

5.1.1.1 Spesifikasi Perangkat Keras

Spesifikasi perangkat keras yang digunakan dalam proses pembuatan aplikasi mesin pencari informasi penjualan rumah yaitu menggunakan komputer dengan spesifikasi *processor*, *memory*, dan *display* ditunjukkan pada Tabel 5.1

Tabel 5. 1 Spesifikasi Perangkat Keras Komputer

Nama Komponen	Spesifikasi
<i>System Model</i>	Toshiba Satellite L745
<i>Processor</i>	AMD A6-3400m APU with Radeon (tm) HD Graphics x 4 (1.40Ghz)
<i>Memory</i>	4096 MB RAM
<i>Display</i>	AMD Radeon HD 6520G

Untuk proses implementasi dan pengujian pada perangkat bergerak *smartphone* Android menggunakan spesifikasi perangkat keras yang ditunjukkan pada Tabel 5.2

Tabel 5. 2 Spesifikasi Perangkat Keras *Smartphone* Android

Nama Komponen	Spesifikasi
<i>System Model</i>	Galaxy S3 Mini GT-I8190
<i>Processor</i>	1 GHz dual-core Cortex-A9
<i>Memory</i>	804 MB RAM
<i>Display</i>	SUPER Amoled capacitive Touchscreen ,16m Colors

5.1.1.2 Spesifikasi Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan dalam proses pembuatan aplikasi mesin pencari informasi penjualan rumah yaitu menggunakan perangkat lunak dengan spesifikasi yang ditunjukkan pada Tabel 5.3

Tabel 5. 3 Spesifikasi perangkat lunak komputer

Nama Komponen	Spesifikasi
<i>Operating System</i>	Microsoft Windows 7
Bahasa Pemrograman	- Java
<i>Text Editor</i>	- Eclipse ADT
<i>Database</i>	SQLite

Untuk proses implementasi dan pengujian pada perangkat bergerak Android menggunakan spesifikasi perangkat lunak yang ditunjukkan pada Tabel 5.4

Tabel 5. 4 Spesifikasi perangkat lunak perangkat bergerak

Nama Komponen	Spesifikasi
<i>Operating System</i>	- Android 4.1.2 (Jelly Bean)

5.1.2 Batasan Implementasi

Dalam implementasi aplikasi mesin pencari informasi penjualan rumah terdapat batasan-batasan yang diberikan yaitu sebagai berikut :

1. Aplikasi mesin pencari informasi penjualan rumah dibuat pada perangkat bergerak Android menggunakan *library* Jsoup sebagai *scraping*.

2. Aplikasi menggunakan koneksi internet karena data informasi penjualan rumah berada di *website*.
3. Pembuatan antarmuka aplikasi menggunakan bahasa pemrograman XML
4. Penyimpanan data informasi penjualan rumah berada pada penyimpanan lokal menggunakan SQLite

5.1.3 Implementasi Basis Data

Implementasi penyimpanan data pada aplikasi mesin pencari informasi penjualan rumah menggunakan basis data untuk menyimpan data informasi penjualan rumah hasil dari proses *web scraping*. Implementasi dilakukan dengan menggunakan SQLite. Implementasi basis data dimodelkan dalam DDL (*Data Definition Language*). Gambar 5.2 dan 5.3 di bawah ini memperlihatkan DDL dari aplikasi mesin pencari informasi penjualan rumah.

1	CREATE TABLE IF NOT EXISTS cari(
2	keys VARCHAR PRIMARY KEY NOT NULL,
3	jenis VARCHAR,
4	maxs VARCHAR ,
5	mins VARCHAR ,
6	km VARCHAR,
7	kt VARCHAR,
8	baru int,
9	hasil int);

Gambar 5. 2 DDL tabel Cari

1	CREATE TABLE IF NOT EXISTS scraping(
2	keys VARCHAR,
3	links VARCHAR);

Gambar 5. 3 DDL tabel Scraping

5.1.4 Implementasi Class dan Assets Pada File Program

Setiap *class* dan *Assets* yang telah dirancang pada proses perancangan direalisasikan pada beberapa file program. Tabel 5.5 di bawah menjelaskan mengenai *class* yang ada pada file program.

Tabel 5. 5 Implementasi *class* pada kode program *.java

Folder	Nama	Nama File Program
src	KataKunci	KataKunci.java
src	CariInfo	CariInfo.java
src	DetailInfo	DetailInfo.java
src	Riwayat	Riwayat.java
src	GetNotif	GetNotif.java
src	MyService	MyService.java
src	CounterManager	CounterManager.java
src	imgAdapter	imgAdapter.java
Android Private Libraries	Jsoup 1.8.1	Jsoup_1.8.1.jar

5.1.5 Implementasi Kode Program

Aplikasi mesin pencari informasi penjualan rumah mempunyai beberapa proses fungsi utama yang terbagi dalam beberapa *file* program. Pada penulisan Tugas Akhir ini hanya dicantumkan implementasi dari beberapa proses saja sehingga tidak semua implementasi akan dicantumkan. Implementasi proses yang dicantumkan antara lain adalah implementasi alur proses *scraping*, dan alur proses notifikasi.

5.1.5.1 Implementasi Alur proses *Scraping*

Alur proses *scraping* dilakukan dengan mengestrak data penjualan rumah pada 3 situs sebagai sumber data. Proses *scraping* dilakukan saat pengguna telah memasukkan kata pencarian dan penyaringan pada halaman awal. Gambar 5.4 berikut adalah implementasi alur proses *scraping* yang terdapat pada file program CariInfo.java

```
1 private boolean getScraping() {
2     try {
3         Elements links;
4         String url = "http://www.rumahku.com/properties/search_engine/";
5
6         Document doc = Jsoup
7             .connect(url)
8             .userAgent("Mozilla/5.0 (Macintosh; U; Intel Mac OS X;
9                 en-en) AppleWebKit/523.10.3 (KHTML, like Gecko) Version/3.0.4
10                 Safari/523.10")
11             .timeout(0).data("data[Property][keyword]", KEY)
12             .data("data[Property][property_action_id]", JENIS)
13             .data("data[Property][property_type_id][0]", "1")
14             .data("data[Property][min_price]", MIN)
15             .data("data[Property][max_price]", MAX)
16             .data("data[Property][beds_range]", KT)
17             .data("data[Property][baths_range]", KM).post();
18         ArrayList<Home> dataHome = new ArrayList<Home>();
19         links = doc.select("#list-properties");
20
21         for (Element link : links) {
22             String link_url, link_gambar, judul, lokasi, harga, k_tidur, k_mandi,
23             l_bangunan, l_tanah;
24             link_url = link.select("h2 > a").attr("abs:href");
25
26             link_gambar = link.select("div.img-property > a > img").attr(
27                 "abs:data-original");
28
29             judul = link.select("h2 ").text();
30             lokasi = link.select("p").text().split("#")[0];
31             String hargaxx = link.select("div[class=price]").text();
32             String hargax = hargaxx;
33             harga = hargax;
34             String ktidurx = link.select("div[class=property-spec] > ul >
35                 li:contains(K. Tidur:)").text();
36             k_tidur = ktidurx.replace("K. Tidur: ", "");
37             String kmandix = link.select("div[class=property-spec] > ul >
38                 li:contains(K. Mandi: )").text();
39             k_mandi = kmandix.replace("K. Mandi: ", "");
40
41             Home newHome = new Home("www.rumahku.com", link_url, link_gambar,
42                 judul, lokasi, harga, k_tidur, k_mandi);
```

```
43 dataHome.add(newHome);
44     }
45 String jenis_rumah;
46 if (JENIS == "1") {
47     jenis_rumah = "dijual";
48 } else {
49     jenis_rumah = "disewa";
50 }
51 url = "http://www.rumah123.com/search";
52
53 doc = Jsoup
54     .connect(url)
55     .userAgent("Mozilla/5.0 (Macintosh; U; Intel Mac OS X; en-en)
56     AppleWebKit/523.10.3 (KHTML, like Gecko) Version/3.0.4
57     Safari/523.10")
58     .timeout(0).data("category", jenis_rumah).data("area", KEY)
59     .data("pricemin", MIN).data("pricemax", MAX)
60     .data("bedroommin", KT).data("bathroommin", KM).get();
61 links = doc.select("div[class=listing default-listing]");
62
63     for (Element link : links) {
64 String link_url, link_gambar, judul, lokasi, harga, k_tidur, k_mandi,
65 l_bangunan, l_tanah;
66 link_url = link.select("h2.listing-title > a").attr("abs:href");
67
68 link_gambar = link.select("div.photo-container > a > img ")
69     .attr("abs:data-original");
70 judul = link.select("h2.listing-title > a").text();
71 lokasi = link.select("div.listing-location").text();
72 String hargaxx = link.select("div.price-info").text();
73 String hargaxy = hargaxx.replace("IDR ", "");
74 harga = hargaxy;
75 String ktidurx = link.select("div.bed-info > b").text();
76 k_tidur = ktidurx.replace("K. Tidur: ", "");
77 String kmandix = link.select("div.bath-info > b").text();
78 k_mandi = kmandix.replace("K. Mandi: ", "");
79 Home newHome = new Home("www.rumah123.com", link_url, link_gambar,
80     judul, lokasi, harga, k_tidur, k_mandi);
81 dataHome.add(newHome);
82     }
83 String offer_type;
84 if (JENIS == "1") {
```



```

85         offer_type = "offer_type:buy";
86     } else {
87         offer_type = "offer_type:rent";
88     }
89
90     url = "http://www.lamudi.co.id/classifieds/";
91
92     doc = Jsoup
93         .connect(url)
94         .userAgent("Mozilla/5.0 (Macintosh; U; Intel Mac OS X; en-en)
95         AppleWebKit/523.10.3 (KHTML, like Gecko) Version/3.0.4
96         Safari/523.10")
97         .timeout(0).data("attribute_option", offer_type)
98         .data("q", KEY).data("price", MIN + "-" + MAX)
99         .data("bedrooms", KT + "+").data("bathrooms", KM + "+")
100        .get();
101    links = doc.select("li[class=highlight-box]");
102    for (Element link : links) {
103        String link_url, link_gambar, judul, lokasi, harga, k_tidur, k_mandi,
104        l_bangunan, l_tanah;
105        link_url = link.select("h3 > a").attr("abs:href");
106
107        link_gambar = link.select("div.listing-image > a > img").attr(
108            "abs:data-src");
109        judul = link.select("h3 > a").text();
110        lokasi = link.select("p[class=listing-address icon-location]").text();
111        String hargaxx = link.select("span.listing-price").text();
112
113        harga = hargaxx;
114        String ktidurx = link.select("span[original-title=Kamar
115            tidur]").text();
116        k_tidur = ktidurx;
117        String kmandix = link.select("span[original-title=Kamar
118            tidur]").text();
119        k_mandi = kmandix;
120
121        Home newHome = new Home("www.lamudi.co.id", link_url, link_gambar,
122            judul, lokasi, harga, k_tidur, k_mandi);
123        dataHome.add(newHome);
124    }
125    dataList = new ArrayList<HashMap<String, String>>();
126    for (int i = 0; i < dataHome.size(); i++) {

```

```

127 HashMap<String, String> tmpHasil = new HashMap<String, String>();
128 tmpHasil.put("A", dataHome.get(i).link_url);
129 tmpHasil.put("B", dataHome.get(i).link_gambar);
130 tmpHasil.put("C", dataHome.get(i).judul);
131 tmpHasil.put("D", dataHome.get(i).lokasi);
132 tmpHasil.put("E", dataHome.get(i).harga);
133 tmpHasil.put("F", dataHome.get(i).k_tidur);
134 tmpHasil.put("G", dataHome.get(i).k_mandi);
135 tmpHasil.put("H", "");
136 tmpHasil.put("I", dataHome.get(i).links);
137 tmpHasil.put("J", STATUS);
138 tmpHasil.put("IMG", dataHome.get(i).link_gambar);
139 tmpHasil.put("INFO", "K.Mandi : " + dataHome.get(i).k_mandi
140                      + " - K.Tidur : " +
141                      dataHome.get(i).k_tidur);
142 dataList.add(tmpHasil);
143     }
144 } catch (Exception e) {
145     e.printStackTrace();
146     return false;
147 }
148     return true;
149 }

```

Gambar 5.4 Impelementasi alur proses *scraping*

Pada Gambar 5.4, digambarkan implementasi kode alur proses *scraping* dari aplikasi mesin pencari informasi penjualan rumah. Berikut penjelasan dari kode program Gambar 5.4 :

1. Baris 1 merupakan nama fungsi yang digunakan untuk *scraping* data pada 3 situs penjualan rumah.
2. Baris 2 merupakan kode program untuk mengecek error.
3. Baris 3-6 merupakan inisialisasi variabel-variabel dari URL rumahku.com.
4. Baris 7-17 merupakan kode program untuk membangun koneksi dan mengirimkan variabel-variabel pencarian sesuai dengan masukkan pada halaman awal aplikasi oleh pengguna ke situs rumahku.com, variable yang dimaksud antara lain :
 - Jenis rumah (dijual/disewa)

- Kata pencarian
 - Harga rumah (minimal/maksimal)
 - Jumlah kamar tidur dan kamar mandi
5. Baris 19-39 merupakan kode program untuk mengekstrak data penjualan rumah yang dibutuhkan setelah situs rumahku.com memberikan respon sesuai variabel-variabel pencarian yang dikirimkan sebelumnya melalui *HTTP request*. Data yang diekstrak antara lain :
- Link url
 - Link gambar
 - Judul iklan
 - Harga rumah
 - Lokasi rumah
 - Jumlah kamar tidur dan kamar mandi
6. Baris 41-43 merupakan kode program untuk menyimpan hasil ekstrak data ke array.
7. Baris 45-50 merupakan kode program mengecek kondisi masukkan jenis rumah yang nanti akan berpengaruh pada pengiriman variabel status rumah ke situs rumah123.com.
8. Baris 51 merupakan inisialisasi variable URL rumah123.com.
9. Baris 53-60 merupakan kode program untuk membangun koneksi dan mengirimkan variabel-variabel pencarian sesuai dengan masukkan pada halaman awal aplikasi oleh pengguna ke situs rumah123.com, variable yang dimaksud antara lain :
- Jenis rumah (dijual/disewa)
 - Kata pencarian
 - Harga rumah (minimal/maksimal)
 - Jumlah kamar tidur dan kamar mandi
10. Baris 61-78 merupakan kode program untuk mengekstrak data penjualan rumah yang dibutuhkan setelah situs rumah123.com memberikan respon sesuai variabel-variabel pencarian yang dikirimkan sebelumnya melalui *HTTP request*. Data yang diekstrak antara lain :

- Link url
- Link gambar
- Judul iklan
- Harga rumah
- Lokasi rumah
- Jumlah kamar tidur dan kamar mandi

11. Baris 79-81 merupakan kode program untuk menyimpan hasil ekstrak data ke array.

12. Baris 83-88 merupakan kode program mengecek kondisi masukkan jenis rumah yang nanti akan berpengaruh pada pengiriman variabel status rumah ke situs lamudi.co.id.

13. Baris 90 merupakan inisialisasi variable URL lamudi.co.id.

14. Baris 92-100 merupakan kode program untuk membangun koneksi dan mengirimkan variabel-variabel pencarian sesuai dengan masukkan pada halaman awal aplikasi oleh pengguna ke situs lamudi.co.id, variable yang dimaksud antara lain :

- Jenis rumah (dijual/disewa)
- Kata pencarian
- Harga rumah (minimal/maksimal)
- Jumlah kamar tidur dan kamar mandi

15. Baris 101-119 merupakan kode program untuk mengekstrak data penjualan rumah yang dibutuhkan setelah situs lamudi.co.id memberikan respon sesuai variabel-variabel pencarian yang dikirimkan sebelumnya melalui HTTP request. Data yang diekstrak antara lain :

- Link url
- Link gambar
- Judul iklan
- Harga rumah
- Lokasi rumah
- Jumlah kamar tidur dan kamar mandi

16. Baris 121-123 merupakan kode program untuk menyimpan hasil ekstrak data ke array.
17. Baris 125-143 merupakan kode program untuk menggabungkan semua data yang telah didapat dari ketiga situs menjadi satu pada sebuah array yang baru.

5.1.5.2 Implementasi Alur proses Notifikasi

Alur proses notifikasi dilakukan dengan melakukan *scraping* otomatis dengan interval waktu tertentu untuk mendapatkan data terkini dan menampilkannya ke pengguna sebagai notifikasi. Proses notifikasi dilakukan setelah pengguna memiliki kata pencarian tersimpan di basis data. Gambar 5.5 berikut adalah implelementasi alur proses notifikasi yang terdapat pada file program GetNotif.java

```

1 public class GetNotifikasi extends AsyncTask<String, Void, Void> {
2     protected void onPreExecute() {
3     }
4     ArrayList<String> arEvent = new ArrayList<String>();
5     protected Void doInBackground(String... urls) {
6         dataList = new ArrayList<HashMap<String, String>>();
7         arEvent = new ArrayList<String>();
8         Cursor c = db.rawQuery("SELECT keys,jenis,maxs,mins,km,kt,baru,hasil
9             FROM tb_cari",null);
10        dataKosong = false;
11        if (c.getCount() == 0) {
12            dataKosong = true;
13        }
14        StringBuffer buffer = new StringBuffer();
15        while (c.moveToNext()) {
16            HashMap<String, String> tmpHasil = new HashMap<String, String>();
17            int hasil = getScraping(c.getString(0), c.getString(1),c.getString(2),
18                c.getString(3), c.getString(4), c.getString(5));
19            if (hasil!=0) {arEvent.add("Kata kunci-"+c.getString(0) + " : " +
20                hasil + " iklan baru\n");
21            db.execSQL("UPDATE tb_cari SET baru= " + hasil + " WHERE keys='" +
22                c.getString(0) + "'");
23        }
24    }

```

```

25         return null;
26     }
27     protected void onPostExecute(Void unused) {
28
29         if (arEvent != null) {
30             String[] events;
31             events = new String[arEvent.size()];
32             for (int i = 0; i < events.length; i++) {
33                 events[i] = arEvent.get(i);
34             }
35             if ( arEvent.size() != 0) {
36                 if (events != null) {
37                     showNotif(events, 0);
38                 }else {
39                     events = new String[] {"Iklan baru tidak ditemukan"};
40                     showNotif(events, 0);
41                 }
42             }else {
43                 events = new String[] {"Iklan baru tidak ditemukan"};
44                 showNotif(events, 0);
45             }
46             }else {
47                 String[] events = new String[] {"Iklan baru tidak ditemukan"};
48                 showNotif(events, 0);
49             }
50         }
51     }
52     ArrayList<HashMap<String, String>> dataList;
53     private NotificationManager mNotificationManager;
54     private int notificationID = 100;
55     private int numMessages = 0;
56     private int JML_HASIL;
57     private int JML_BARU = 0;
58     boolean isExist;
59     private int getScraping(String KEY, String JENIS, String MIN, String
60         MAX,
61         String KT, String KM) {
62         int hasil = 0 ;
63         try {
64             Elements links;
65             String url = "http://www.rumahku.com/properties/search_engine/";
66

```



```

67 Document doc;
68 ArrayList<Home> dataHome = new ArrayList<Home>();
69     try {
70         doc = Jsoup.connect(url)
71             .userAgent("Mozilla/5.0 (Macintosh; U; Intel Mac OS X;
72             en-en) AppleWebKit/523.10.3 (KHTML, like Gecko) Version/3.0.4
73             Safari/523.10")
74             .timeout(0).data("data[Property][keyword]", KEY)
75             .data("data[Property][property_action_id]", JENIS)
76             .data("data[Property][property_type_id][0]", "1")
77             .data("data[Property][min_price]", MIN)
78             .data("data[Property][max_price]", MAX)
79             .data("data[Property][beds_range]", KT)
80             .data("data[Property][baths_range]", KM).post();
81     links = doc.select("#list-properties");
82     for (Element link : links) {
83         String link_url, link_gambar, judul, lokasi, harga, k_tidur, k_mandi,
84         l_bangunan, l_tanah;
85         link_url = link.select("h2 > a
86         Home newHome = new Home("www.rumahku.com", link_url, "", "", "", "",
87         "", "");
88         dataHome.add(newHome);
89     }
90     } catch (Exception ex) {
91         ex.printStackTrace();
92     }
93     try {
94         url = "http://www.rumah123.com/search";
95
96         doc = Jsoup.connect(url)
97             .userAgent("Mozilla/5.0 (Macintosh; U; Intel Mac OS X; en-en)
98             AppleWebKit/523.10.3 (KHTML, like Gecko) Version/3.0.4
99             Safari/523.10")
100             .timeout(0).data("category", JENIS).data("area", KEY)
101             .data("pricemin", MIN).data("pricemax", MAX)
102             .data("bedroommin", KT).data("bathroommin", KM).get();
103         links = doc.select("div[class=listing default-listing]");
104         for (Element link : links) {
105             String link_url;
106             link_url = link.select("h2.listing-title > a").attr("abs:href");
107             Home newHome = new Home("www.rumah123.com", link_url, "", "", "", "",
108             "", "");

```

```
109 dataHome.add(newHome);
110     }
111     } catch (Exception ex) {
112         ex.printStackTrace();
113     }
114     String offer_type;
115     if (JENIS == "1") {
116         offer_type = "offer_type:buy";
117     } else {
118         offer_type = "offer_type:rent";
119     }
120     try {
121         url = "http://www.lamudi.co.id/classifieds/";
122
123         doc = Jsoup.connect(url)
124             .userAgent("Mozilla/5.0 (Macintosh; U; Intel Mac OS X; en-en)
125                 AppleWebKit/523.10.3 (KHTML, like Gecko) Version/3.0.4
126                 Safari/523.10")
127             .timeout(0).data("attribute_option", offer_type)
128             .data("q", KEY).data("price", MIN + "-" + MAX)
129             .data("bedrooms", KT + "+").data("bathrooms", KM + "+")
130             .get();
131         links = doc.select("li[class=highlight-box]");
132         for (Element link : links) {
133             String link_url;
134             link_url = link.select("h3 > a").attr("abs:href");
135             Home newHome = new Home("www.lamudi.co.id", link_url, "", "", "", "",
136                 "", "");
137             dataHome.add(newHome);
138         }
139     } catch (Exception ex) {
140         ex.printStackTrace();
141     }
142     dataList = new ArrayList<HashMap<String, String>>();
143     isExist = false;
144     for (int i = 0; i < dataHome.size(); i++) {
145         HashMap<String, String> tmpHasil = new HashMap<String, String>();
146         tmpHasil.put("A", dataHome.get(i).link_url);
147         isExist = true;
148         dataList.add(tmpHasil);
149     }
150     JML_HASIL = 0;
```

```

151     hasil = 0 ;
152     for (int i = 0; i < dataHome.size(); i++) {
153         if (isExist) {
154             if (cekNotif(dataHome.get(i).link_url,KEY,JENIS,MIN,
155                 MAX, KT,   KM)) {
156             } else
157             {
158                 System.out.println(dataHome.get(i).link_url);
159                 hasil ++;
160             }
161         }
162         JML_HASIl++;
163     }
164     db.execSQL("UPDATE tb_cari SET hasil= "
165 + JML_HASIl + " WHERE keys='" + KEY + "'");
166
167 } catch (Exception e) {
168     e.printStackTrace();
169     return hasil;
170 }
171 return hasil;
172 }
173 public void showNotif(String[] events, int position) {
174
175     Notification.Builder mBuilder = new Notification.Builder(c);
176     mBuilder.setContentTitle("Notifikasi Home Finder");
177     mBuilder.setSmallIcon(R.drawable.ic_launcher);
178     mBuilder.setNumber(++numMessages);
179     Notification.InboxStyle inboxStyle = new Notification.InboxStyle();
180     inboxStyle.setBigContentTitle("Notifikasi Home Finder :");
181     for (int i = 0; i < events.length; i++) {
182         inboxStyle.addLine(events[i]);
183     }
184     mBuilder.setStyle(inboxStyle);
185     mBuilder.setContentText(events[0]);
186     mBuilder.setTicker(events[0]);
187     {
188         Intent resultIntent = new Intent(c, Riwayat.class);
189         TaskStackBuilder stackBuilder = TaskStackBuilder.create(c);
190         stackBuilder.addParentStack(Riwayat.class);
191         stackBuilder.addNextIntent(resultIntent);
192         PendingIntent resultPendingIntent = stackBuilder.getPendingIntent(

```



```

193         0, PendingIntent.FLAG_UPDATE_CURRENT);
194         mBuilder.setContentIntent(resultPendingIntent);
195     }
196     mNotificationManager = (NotificationManager)
197         c.getSystemService(Context.NOTIFICATION_SERVICE);
198     mNotificationManager.notify(notificationID, mBuilder.build());
199     }
    
```

Gambar 5.5 Implementasi alur proses notifikasi

Pada Gambar 5.5, digambarkan implementasi kode alur proses notifikasi dari aplikasi mesin pencari informasi penjualan rumah. Berikut penjelasan dari kode program Gambar 5.5 :

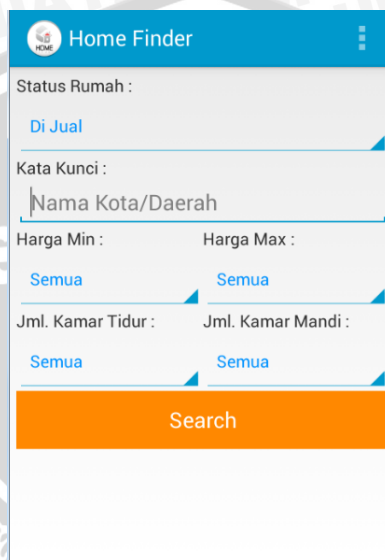
1. Baris 1 merupakan nama fungsi yang digunakan untuk notifikasi.
2. Baris 8-13 merupakan kode program untuk mengambil kata kunci yang tersimpan di basis data.
3. Baris 17-26 merupakan kode program untuk mengecek kondisi apabila terdapat data kata kunci di basis data maka akan melakukan *scraping* otomatis untuk mengecek keberadaan data baru pada situs penjualan rumah dan menampilkan jumlah iklan baru yang didapat.
4. Baris 29-48 merupakan kode program untuk mengecek apabila tidak terdapat data baru maka akan menampilkan notifikasi data baru tidak ditemukan.
5. Baris 52-62 merupakan inialisasi variabel-variabel untuk keperluan notifikasi
6. Baris 63-172 merupakan kode program untuk *scraping* data penjualan rumah pada 3 situs.
7. Baris 173-199 merupakan kode program untuk menampilkan notifikasi dan menangani eksekusi apabila notifikasi ditekan akan menuju halaman riwayat.

5.1.6 Implementasi Antarmuka Aplikasi

Bagian ini akan menunjukkan hasil implementasi antarmuka pada aplikasi mesin pencari informasi penjualan rumah yang terdiri dari halaman utama, halaman daftar informasi rumah, halaman detail informasi, dan halaman riwayat.

5.1.6.1 Halaman Utama

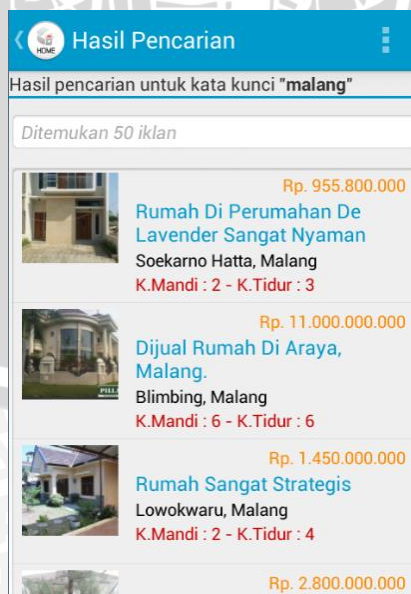
Halaman utama akan ditampilkan setelah pengguna membuka aplikasi pertama kali. Halaman utama berisi kolom input pencarian dan penyaringan serta tombol “cari”.



Gambar 5. 6 Halaman Utama

5.1.6.2 Halaman Daftar Informasi Penjualan Rumah

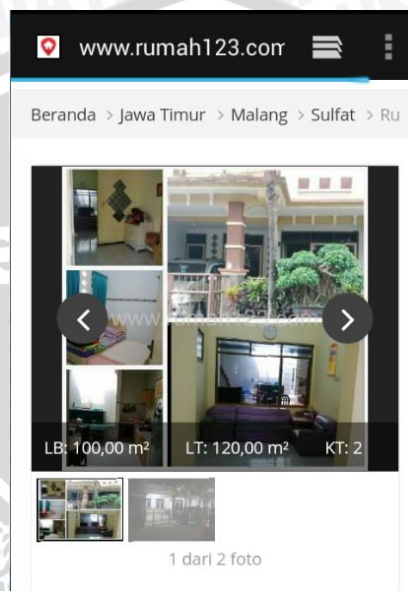
Halaman detail informasi rumah akan ditampilkan setelah pengguna memasukkan kata pencarian dan penyaringan.



Gambar 5. 7 Halaman Daftar Informasi Penjualan Rumah

5.1.6.3 Halaman Detail Informasi Penjualan Rumah

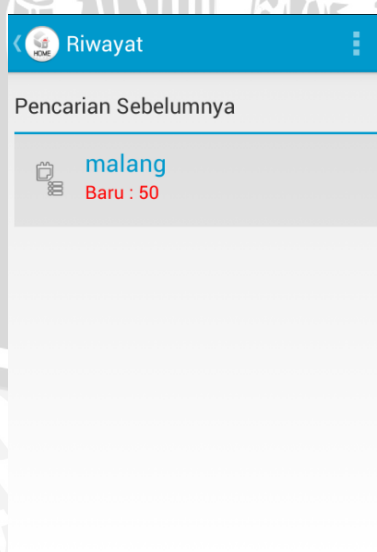
Halaman detail informasi penjualan rumah akan ditampilkan setelah pengguna memilih salah satu informasi. Halaman ini, akan menampilkan alamat asal informasi penjualan rumah dengan cara membuka browser eksternal.



Gambar 5.8 Halaman Detail Informasi Penjualan Rumah

5.1.6.4 Halaman Riwayat

Halaman Riwayat akan ditampilkan saat pengguna memilih menu riwayat. Halaman ini akan berisi daftar kata pencarian yang tersimpan oleh user.



Gambar 5.9 Halaman Riwayat

5.2 Pengujian

Bagian ini menjelaskan tentang pengujian dan analisis yang akan dilakukan pada aplikasi yang telah dibangun. Pengujian yang dilakukan antara lain pengujian unit, pengujian validasi dan pengujian non fungsional. Setelah melalui tahap pengujian, selanjutnya dilakukan analisis terhadap hasil dari pengujian yang telah dilakukan. Analisis dilakukan untuk mendapatkan kesimpulan dari hasil setiap pengujian yang dilakukan.

5.2.1 Pengujian Unit

Pengujian unit dilakukan untuk melihat apakah algoritma berjalan dengan benar. Teknik yang digunakan dalam melakukan pengujian unit ini adalah teknik *Basis Path Testing*. Pada teknik *basis path testing*, proses pengujian dilakukan dengan memodelkan algoritma pada suatu *flowgraph*, kemudian menentukan jumlah kompleksitas siklomatis (*cyclomatic complexity*), basis set dari jalur independen dan memberikan kasus uji (*test case*) pada setiap basis set yang telah ditentukan.

5.2.1.2 Kasus Uji Unit *Scraping*

Gambar 5.10 berikut menunjukkan proses pengujian unit pada alur proses *scraping* dan Gambar 5.11 menunjukkan pemodelan *flowchart* dan *flowgraph* pada alur proses *scraping*.

```

1 private boolean getScraping() {
2     try {
3         Elements links;.....1
4         String url = "http://www.rumahku.com/properties/search_engine/";
5
6         Document doc = Jsoup.....2
7             .connect(url)
8             .userAgent("Mozilla/5.0 (Macintosh; U; Intel Mac OS X;
9                 en-en) AppleWebKit/523.10.3 (KHTML, like Gecko) Version/3.0.4
10                Safari/523.10")
11             .timeout(0).data("data[Property][keyword]", KEY)
12             .data("data[Property][property_action_id]", JENIS)
13             .data("data[Property][property_type_id][0]", "1")
14             .data("data[Property][min_price]", MIN)
15             .data("data[Property][max_price]", MAX)
16             .data("data[Property][beds_range]", KT)

```

```

17         .data("data[Property][baths_range]", KM).post();
18     ArrayList<Home> dataHome = new ArrayList<Home>();
19     links = doc.select("#list-properties");
20
21     for (Element link : links) {.....3
22     String link_url, link_gambar, judul, lokasi, harga, k_tidur, k_mandi,
23     1 bangunan, 1 tanah;
24     link_url = link.select("h2 > a").attr("abs:href");
25
26     link_gambar = link.select("div.img-property > a > img").attr(
27         "abs:data-original");
28
29     judul = link.select("h2 ").text();
30     lokasi = link.select("p").text().split("#")[0];
31     String hargaxx = link.select("div[class=price]").text();
32     String hargax = hargaxx;
33     harga = hargax;
34     String ktidurx = link.select("div[class=property-spec] > ul >
35         li:contains(K. Tidur:)").text();
36     k_tidur = ktidurx.replace("K. Tidur: ", "");
37     String kmandix = link.select("div[class=property-spec] > ul >
38         li:contains(K. Mandi:)").text();
39     k_mandi = kmandix.replace("K. Mandi: ", "");
40
41     Home newHome = new Home("www.rumahku.com", link_url, link_gambar,
42         judul, lokasi, harga, k_tidur, k_mandi);
43     dataHome.add(newHome);.....4
44         }.....5
45     String jenis_rumah;
46     if (JENIS == "1") {
47         jenis_rumah = "dijual";
48     } else {
49         jenis_rumah = "disewa";
50     }
51     url = "http://www.rumah123.com/search";
52
53     doc = Jsoup
54         .connect(url)
55         .userAgent("Mozilla/5.0 (Macintosh; U; Intel Mac OS X; en-en)
56         AppleWebKit/523.10.3 (KHTML, like Gecko) Version/3.0.4
57         Safari/523.10")

```

```

58         .timeout(0).data("category", jenis_rumah).data("area", KEY)
59         .data("pricemin", MIN).data("pricemax", MAX)
60         .data("bedroommin", KT).data("bathroommin", KM).get();
61     links = doc.select("div[class=listing default-listing]");
62
63     for (Element link : links) {
64         String link_url, link_gambar, judul, lokasi, harga, k_tidur, k_mandi,
65         l_bangunan, l_tanah;
66         link_url = link.select("h2.listing-title > a").attr("abs:href");
67
68         link_gambar = link.select("div.photo-container > a > img ")
69             .attr("abs:data-original");
70         judul = link.select("h2.listing-title > a").text();
71         lokasi = link.select("div.listing-location").text();
72         String hargaxx = link.select("div.price-info").text();
73         String hargaxy = hargaxx.replace("IDR ", "");
74         harga = hargaxy;
75         String ktidurx = link.select("div.bed-info > b").text();
76         k_tidur = ktidurx.replace("K. Tidur: ", "");
77         String kmandix = link.select("div.bath-info > b").text();
78         k_mandi = kmandix.replace("K. Mandi: ", "");
79         Home newHome = new Home("www.rumah123.com", link_url, link_gambar,
80             judul, lokasi, harga, k_tidur, k_mandi);
81         dataHome.add(newHome);
82     }
83     String offer_type;
84     if (JENIS == "1") {
85         offer_type = "offer_type:buy";
86     } else {
87         offer_type = "offer_type:rent";
88     }
89
90     url = "http://www.lamudi.co.id/classifieds/";
91
92     doc = Jsoup
93         .connect(url)
94         .userAgent("Mozilla/5.0 (Macintosh; U; Intel Mac OS X; en-en)
95         AppleWebKit/523.10.3 (KHTML, like Gecko) Version/3.0.4
96         Safari/523.10")
97         .timeout(0).data("attribute_option", offer_type)
98         .data("q", KEY).data("price", MIN + "-" + MAX)
99         .data("bedrooms", KT + "+").data("bathrooms", KM + "+")

```



```

100         .get();
101     links = doc.select("li[class=highlight-box]");
102     for (Element link : links) {
103         String link_url, link_gambar, judul, lokasi, harga, k_tidur, k_mandi,
104         l_bangunan, l_tanah;
105         link_url = link.select("h3 > a").attr("abs:href");
106
107         link_gambar = link.select("div.listing-image > a > img").attr(
108             "abs:data-src");
109         judul = link.select("h3 > a").text();
110         lokasi = link.select("p[class=listing-address icon-location]").text();
111         String hargaxx = link.select("span.listing-price").text();
112
113         harga = hargaxx;
114         String ktidurx = link.select("span[original-title=Kamar
115             tidur]").text();
116         k_tidur = ktidurx;
117         String kmandix = link.select("span[original-title=Kamar
118             tidur]").text();
119         k_mandi = kmandix;
120
121         Home newHome = new Home("www.lamudi.co.id", link_url, link_gambar,
122             judul, lokasi, harga, k_tidur, k_mandi);
123         dataHome.add(newHome);
124     }
125     dataList = new ArrayList<HashMap<String, String>>();
126     for (int i = 0; i < dataHome.size(); i++) {
127         HashMap<String, String> tmpHasil = new HashMap<String, String>();
128         tmpHasil.put("A", dataHome.get(i).link_url);
129         tmpHasil.put("B", dataHome.get(i).link_gambar);
130         tmpHasil.put("C", dataHome.get(i).judul);
131         tmpHasil.put("D", dataHome.get(i).lokasi);
132         tmpHasil.put("E", dataHome.get(i).harga);
133         tmpHasil.put("F", dataHome.get(i).k_tidur);
134         tmpHasil.put("G", dataHome.get(i).k_mandi);
135         tmpHasil.put("H", "");
136         tmpHasil.put("I", dataHome.get(i).links);
137         tmpHasil.put("J", STATUS);
138         tmpHasil.put("IMG", dataHome.get(i).link_gambar);
139         tmpHasil.put("INFO", "K.Mandi : " + dataHome.get(i).k_mandi
140             + " - K.Tidur : " +
141             dataHome.get(i).k_tidur);

```

```
142         dataList.add(tmpHasil);
143     }
144     } catch (Exception e) {...6
145         e.printStackTrace();
146         return false;
147     }
148     return true;
149 }
150 public class TampilScraping extends AsyncTask<String, Void, Void> {
151     private ProgressDialog Dialog = new ProgressDialog(CariInfo.this);
152     boolean status = false;
153     protected void onPreExecute() {
154         Dialog.setMessage("Proses pencarian ke server...");
155         Dialog.show();
156         Dialog.setCancelable(true);
157     }
158     protected Void doInBackground(String... urls) {
159         status = getScraping();
160         return null;
161     }
162     @Override
163     protected void onCancelled(Void unused) {
164         // TODO Auto-generated method stub
165         CariInfo.this.finish();
166     }
167     @Override
168     protected void onCancelled() {
169         // TODO Auto-generated method stub
170         CariInfo.this.finish();
171     }
172     protected void onPostExecute(Void unused) {
173         Dialog.dismiss();
174         if (status && dataList.size() != 0) {
175             JML_HASIL = 0;
176             JML_BARU = 0;
177             for (int i = 0; i < dataList.size(); i++) {
178                 if (isExist) {
179                     if (cekKata(dataList.get(i).get("A")) {...7
180                         dataList.get(i).remove("H");
181                         dataList.get(i).put("H", "");...8
182                     } else {
183                         dataList.get(i).remove("H");
184                         dataList.get(i).put("H", "BARU");
```

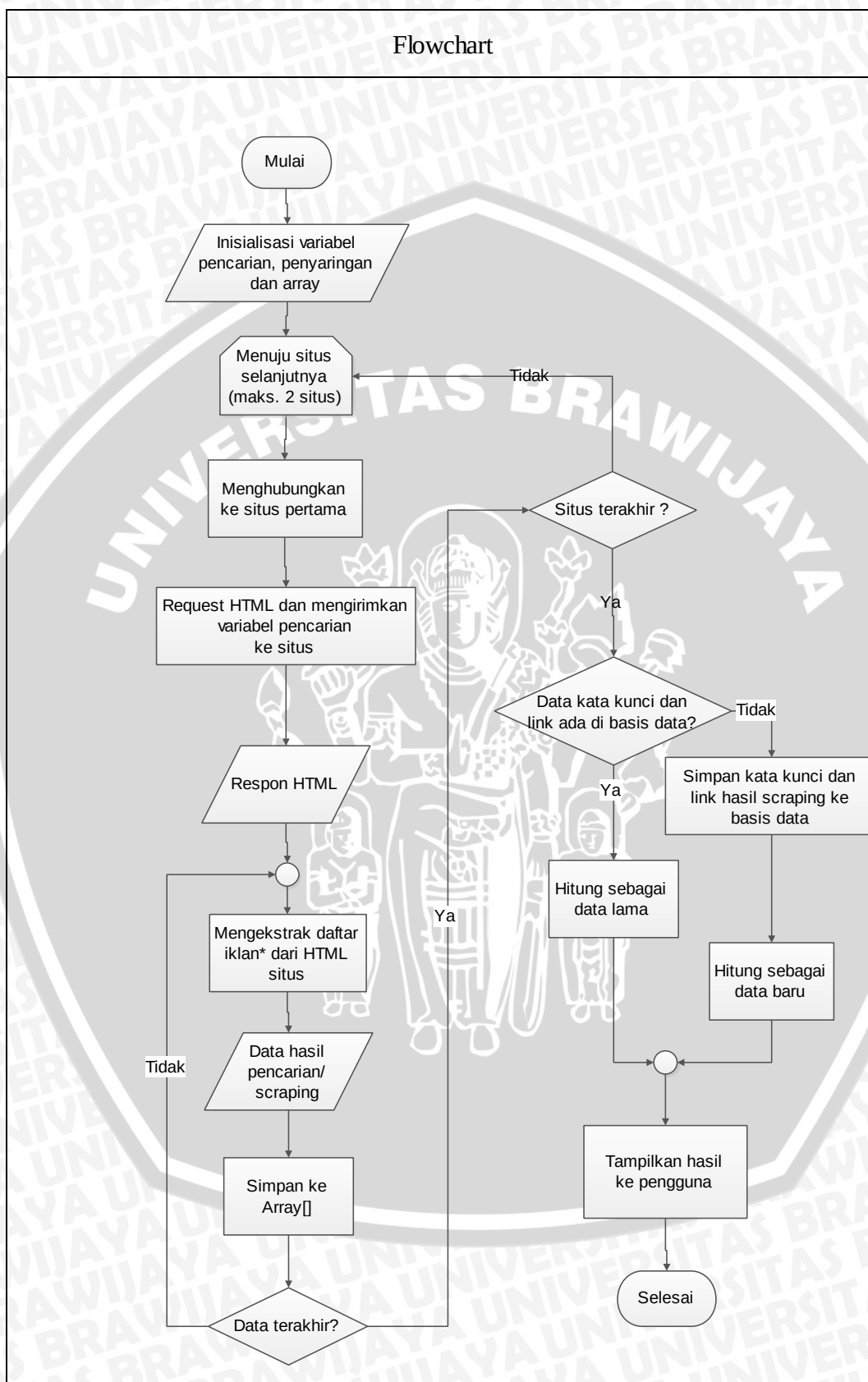
```

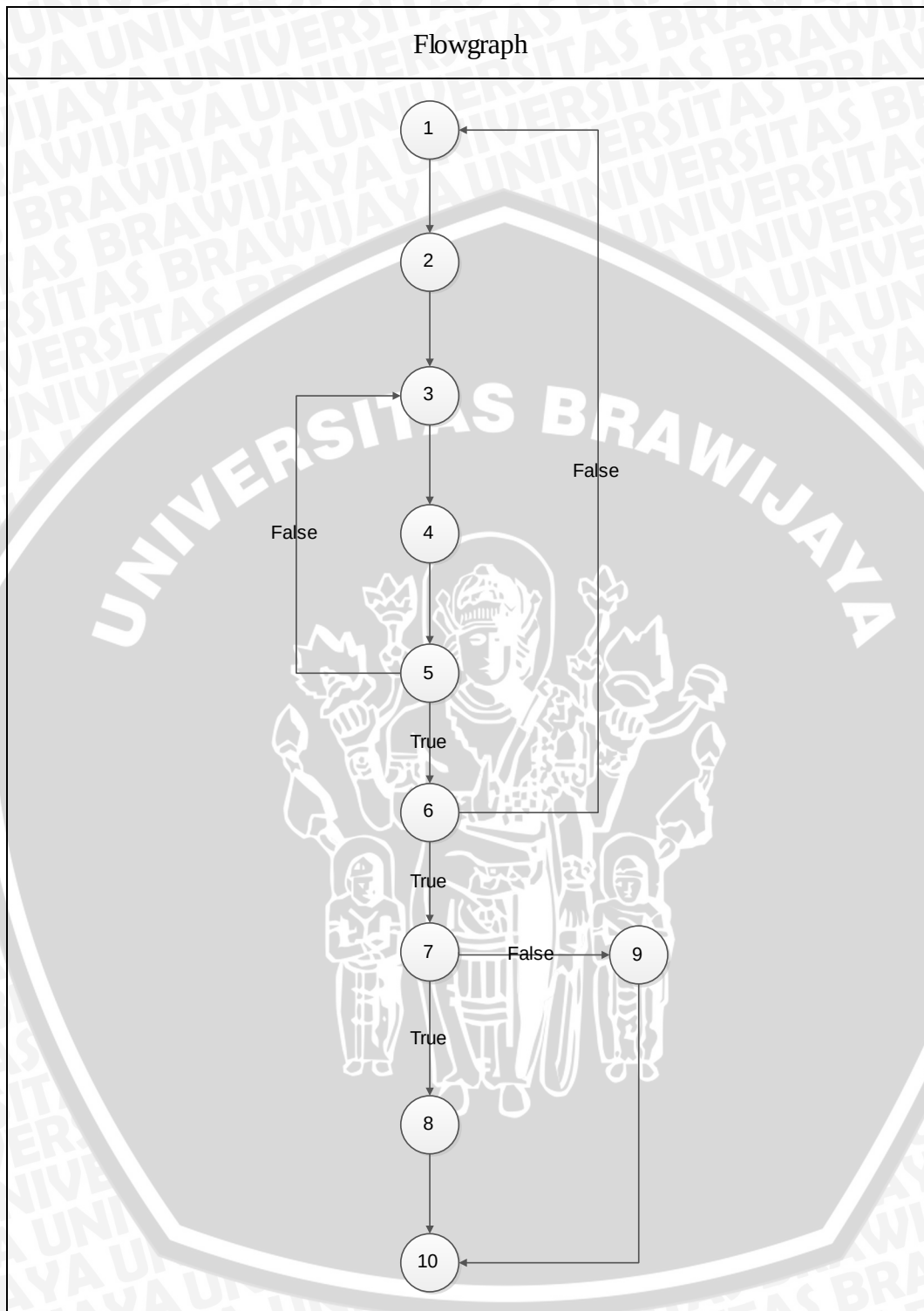
184         db.execSQL("INSERT INTO scraping VALUES('" + KEY.....9
185         + "','" + dataList.get(i).get("A") + "' );");
186         JML_BARU++;
187     }
188 }
189     JML_HASIL++;
190 }
191     imgAdapter adapter = new imgAdapter(CariInfo.this, dataList, "C", "D",
192     "INFO", "H", "E", "I" , "J" );.....10
193     lvData.setAdapter(adapter);
194     lvData.setOnItemClickListener(new OnItemClickListener() {
195     public void onItemClick(AdapterView<?> parent, View view,
196     int position, long id) {
197         gotoItem(position);
198     });
199     txtHasil.setText("Ditemukan " + dataList.size() + " iklan");
200     } else {
201     u.Alert("Proses pengambilan data gagal, silahkan coba lagi..");
202     finish();
203     } }

```

Gambar 5. 10 Pengujian Unit Alur Proses *Scraping*

Flowchart





Gambar 5. 11 Flowchart dan Flowgraph Alur Proses Scraping

Pemodelan *flowgraph* yang telah dilakukan terhadap alur proses *scraping* menghasilkan kompleksitas siklomatis (*cyclomatic complexity*) melalui persamaan $V(G) = E - N + 2$, dimana $V(G)$ merupakan jumlah kompleksitas siklomatis, E

merupakan sisi atau edge (garis penghubung antar node) dan N merupakan jumlah simpul (node).

$$V(G) = E - N + 2 = 12 - 10 + 2 = 4$$

Berdasarkan dari nilai *cyclomatic complexity* yang telah didapatkan dari perhitungan, maka ditentukan dua buah basis set dari jalur *independent*, yaitu :

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

Jalur 2 : 1-2-3-5-4-5-6-7-9-10

Jalur 3 : 1-2-3-4-5-3-n

Jalur 4 : 1-2-3-4-5-6-1-n

Setiap jalur yang didapatkan akan dieksekusi sebanyak 1 kali jika syarat yang ada pada perintah if-else terpenuhi

Tabel 5. 6 Kasus Uji Untuk Alur proses *Scraping*

Jalur	Kasus Uji	Hasil yang diharapkan	Hasil yang didapatkan
1	Menampilkan informasi iklan rumah lama	Sistem tidak memberi informasi “baru” pada iklan yang menandakan iklan sudah pernah dilihat	Sistem tidak memberi informasi “baru” pada iklan yang menandakan iklan sudah pernah dilihat
2	Menampilkan informasi iklan rumah baru	Sistem memberi informasi “baru” pada iklan yang menandakan iklan belum pernah dilihat	Sistem memberi informasi “baru” pada iklan yang menandakan iklan belum pernah dilihat

Untuk beberapa jalur tidak diberikan kasus uji dikarenakan jalur tersebut merupakan satu kesatuan terhadap alur proses *scraping* yang mana tidak terdapat kasus yang perlu diujikan. Seperti contoh pada pengecekan di *flowgraph* nomor 5

dan 6 masing-masing adalah pengecekan data terakhir dan situs terakhir saat melakukan proses *scraping*.

5.2.1.3 Kasus Uji Unit Notifikasi

Gambar 5.12 berikut menunjukkan proses pengujian unit pada alur proses notifikasi dan Gambar 5.13 menunjukkan pemodelan *flowchart* dan *flowgraph* pada alur proses notifikasi.

```

1 public class GetNotifikasi extends AsyncTask<String, Void, Void> {
2     protected void onPreExecute() {
3     }
4     ArrayList<String> arEvent = new ArrayList<String>();
5     protected Void doInBackground(String... urls) {
6         dataList = new ArrayList<HashMap<String, String>>();
7         arEvent = new ArrayList<String>();
8         Cursor c = db.rawQuery("SELECT keys,jenis,maxs,mins,km,kt,baru,hasil
9             FROM tb_cari",null);
10        dataKosong = false;
11        if (c.getCount() == 0) {.....1
12            dataKosong = true;
13        }
14        StringBuffer buffer = new StringBuffer();
15        while (c.moveToNext()) {.....2
16        HashMap<String, String> tmpHasil = new HashMap<String, String>();
17        int hasil = getScraping(c.getString(0), c.getString(1),c.getString(2),
18            c.getString(3), c.getString(4), c.getString(5));
19        if (hasil!=0) {arEvent.add("Kata kunci-"+c.getString(0) + " : " +
20            hasil + " iklan baru\n");.....5
21        db.execSQL("UPDATE tb_cari SET baru= " + hasil + " WHERE keys='" +
22            c.getString(0) + "'");
23        }
24        }
25        return null;
26    }
27    protected void onPostExecute(Void unused) {
28
29    if (arEvent != null) {
30        String[] events;
31        events = new String[arEvent.size()];
32        for (int i = 0; i < events.length; i++) {
33            events[i] = arEvent.get(i);

```

```

34         }
35         if ( arEvent.size() != 0 ) {
36             if (events != null) {
37                 showNotif(events, 0);
38             }else {
39                 events = new String[] {"Iklan baru tidak ditemukan"};
40                 showNotif(events, 0);
41             }
42             }else {
43                 events = new String[] {"Iklan baru tidak ditemukan"};
44                 showNotif(events, 0);
45             }
46             }else {
47                 String[] events = new String[] {"Iklan baru tidak ditemukan"};
48                 showNotif(events, 0);
49             }
50         }
51     }
52     ArrayList<HashMap<String, String>> dataList;
53     private NotificationManager mNotificationManager;
54     private int notificationID = 100;
55     private int numMessages = 0;
56     private int JML_HASIL;
57     private int JML_BARU = 0;
58     boolean isExist;
59     private int getScraping(String KEY, String JENIS, String MIN, String
60         MAX,
61     String KT, String KM) {
62         int hasil = 0 ;
63         try {
64             Elements links;
65             String url = "http://www.rumahku.com/properties/search_engine/";
66
67             Document doc;
68             ArrayList<Home> dataHome = new ArrayList<Home>();
69             try {
70                 doc = Jsoup.connect(url)
71                     .userAgent("Mozilla/5.0 (Macintosh; U; Intel Mac OS X;
72                     en-en) AppleWebKit/523.10.3 (KHTML, like Gecko) Version/3.0.4
73                     Safari/523.10")
74                     .timeout(0).data("data[Property][keyword]", KEY)
75                     .data("data[Property][property_action_id]", JENIS)

```

```

76         .data("data[Property][property_type_id][0]", "1")
77         .data("data[Property][min_price]", MIN)
78         .data("data[Property][max_price]", MAX)
79         .data("data[Property][beds_range]", KT)
80         .data("data[Property][baths_range]", KM).post();
81     links = doc.select("#list-properties");
82     for (Element link : links) {
83         String link_url, link_gambar, judul, lokasi, harga, k_tidur, k_mandi,
84             l_bangunan, l_tanah;
85         link_url = link.select("h2 > a
86         Home newHome = new Home("www.rumahku.com", link_url, "", "", "", "",
87             "", "");
88         dataHome.add(newHome);
89     }
90     } catch (Exception ex) {
91         ex.printStackTrace();
92     }
93     try {
94         url = "http://www.rumah123.com/search";
95
96         doc = Jsoup.connect(url)
97             .userAgent("Mozilla/5.0 (Macintosh; U; Intel Mac OS X; en-en)
98             AppleWebKit/523.10.3 (KHTML, like Gecko) Version/3.0.4
99             Safari/523.10")
100             .timeout(0).data("category", JENIS).data("area", KEY)
101             .data("pricemin", MIN).data("pricemax", MAX)
102             .data("bedroommin", KT).data("bathroommin", KM).get();
103             links = doc.select("div[class=listing default-listing]");
104             for (Element link : links) {
105                 String link_url;
106                 link_url = link.select("h2.listing-title > a").attr("abs:href");
107                 Home newHome = new Home("www.rumah123.com", link_url, "", "", "", "",
108                     "", "");
109                 dataHome.add(newHome);
110             }
111             } catch (Exception ex) {
112                 ex.printStackTrace();
113             }
114         String offer_type;
115         if (JENIS == "1") {
116             offer_type = "offer_type:buy";
117         } else {

```



```

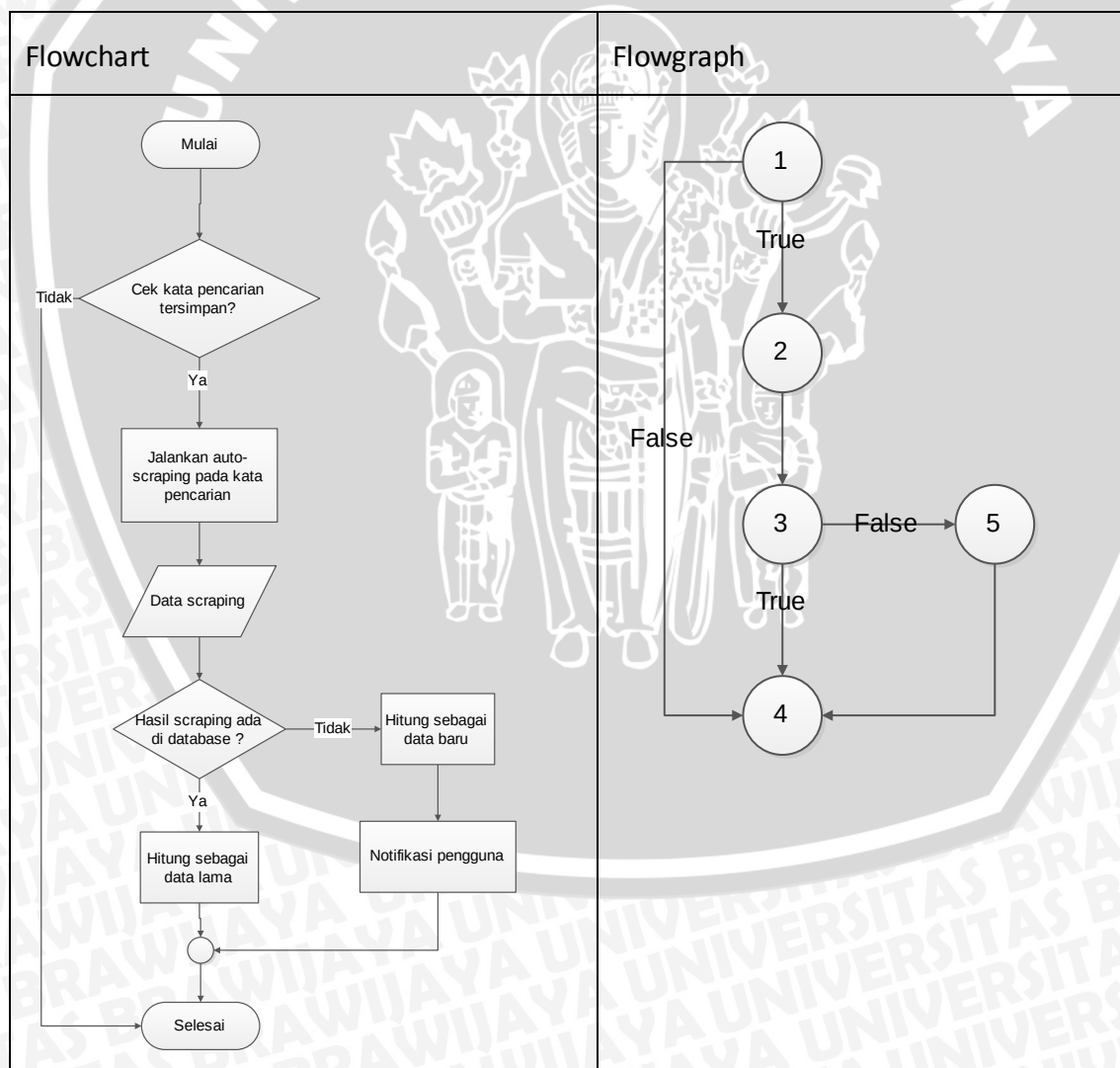
118         offer_type = "offer_type:rent";
119     }
120     try {
121         url = "http://www.lamudi.co.id/classifieds/";
122
123         doc = Jsoup.connect(url)
124             .userAgent("Mozilla/5.0 (Macintosh; U; Intel Mac OS X; en-en)
125             AppleWebKit/523.10.3 (KHTML, like Gecko) Version/3.0.4
126             Safari/523.10")
127             .timeout(0).data("attribute_option", offer_type)
128             .data("q", KEY).data("price", MIN + "-" + MAX)
129             .data("bedrooms", KT + "+").data("bathrooms", KM + "+")
130             .get();
131         links = doc.select("li[class=highlight-box]");
132         for (Element link : links) {
133             String link_url;
134             link_url = link.select("h3 > a").attr("abs:href");
135             Home newHome = new Home("www.lamudi.co.id", link_url, "", "", "", "",
136                 "", "");
137             dataHome.add(newHome);
138         }
139         } catch (Exception ex) {
140             ex.printStackTrace();
141         }
142         dataList = new ArrayList<HashMap<String, String>>();
143         isExist = false;
144         for (int i = 0; i < dataHome.size(); i++) {
145             HashMap<String, String> tmpHasil = new HashMap<String, String>();
146             tmpHasil.put("A", dataHome.get(i).link_url);
147             isExist = true;
148             dataList.add(tmpHasil);
149         }
150         JML_HASIL = 0;
151         hasil = 0 ;
152         for (int i = 0; i < dataHome.size(); i++) {
153             if (isExist) {.....3
154                 if (cekNotif(dataHome.get(i).link_url,KEY,JENIS,MIN,
155                     MAX, KT, KM)) {.....4
156                     } else
157                     {
158                     System.out.println(dataHome.get(i).link_url);
159                     hasil ++;

```

```

160         }
161     }
162     JML_HASIL++;
163 }
164 db.execSQL("UPDATE tb_cari SET hasil= "
165 + JML_HASIL + " WHERE keys='" + KEY + "'");
166
167 } catch (Exception e) {
168     e.printStackTrace();
169     return hasil;
170 }
171 return hasil;}
    
```

Gambar 5. 12 Pengujian Unit Alur Proses Notifikasi



Gambar 5. 13 Flowchart dan Flowgraph Alur Proses Notifikasi

Pemodelan *flowgraph* yang telah dilakukan terhadap notifikasi menghasilkan kompleksitas siklomatis (*cyclomatic complexity*) melalui persamaan $V(G) = E - N + 2$, dimana $V(G)$ merupakan jumlah kompleksitas siklomatis, E merupakan sisi atau edge (garis penghubung antar node) dan N merupakan jumlah simpul (node).

$$V(G) = E - N + 2 = 6 - 5 + 2 = 3$$

Berdasarkan dari nilai *cyclomatic complexity* yang telah didapatkan dari perhitungan, maka ditentukan dua buah basis set dari jalur *independent*, yaitu :

Jalur 1 : 1-2-3-4

Jalur 2 : 1-2-3-5-4

Jalur 3 : 1-4

Setiap jalur yang didapatkan akan dieksekusi sebanyak 1 kali jika syarat yang ada pada perintah *if-else* terpenuhi.

Tabel 5. 7 Kasus Uji Untuk Pengujian Unit Alur Proses Notifikasi

Jalur	Kasus Uji	Hasil yang diharapkan	Hasil yang didapatkan
1	Tidak terdapat iklan rumah baru	Sistem mendapatkan tidak mendapatkan iklan rumah terbaru dan menampilkan pesan pemberitahuan	Sistem mendapatkan tidak mendapatkan iklan rumah terbaru dan menampilkan pesan pemberitahuan
2	Mendapat notifikasi	Sistem mendapatkan iklan rumah baru dan menotifikasi pengguna	Sistem mendapatkan iklan rumah baru dan menotifikasi pengguna
3	Tidak terdapat kata pencarian tersimpan	Sistem tidak melakukan <i>update</i> informasi	Sistem tidak melakukan <i>update</i> informasi

5.2.2 Pengujian Validasi

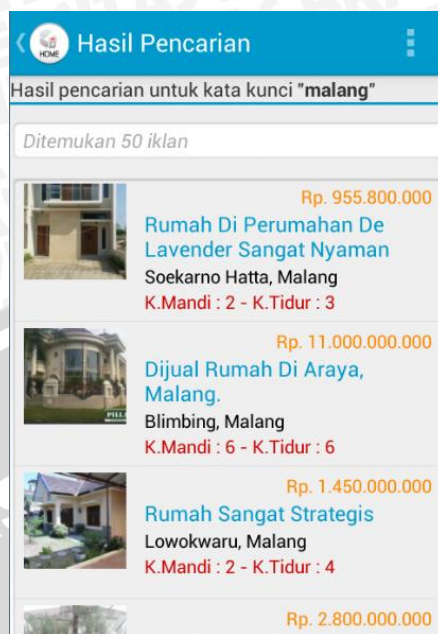
Pengujian validasi dilakukan untuk mengetahui apakah sistem yang dibangun sudah berjalan seperti yang diharapkan. Pengujian validasi ini dilakukan berdasarkan skenario *use case* yang ada pada tahapan perancangan. Pengujian ini menggunakan metode *black-box testing*, karena tidak diperlukan konsentrasi terhadap alur jalannya program dan lebih ditekankan untuk menemukan kesesuaian antara kinerja sistem dengan daftar kebutuhan.

5.2.2.1 Kasus Uji Mencari Informasi Penjualan Rumah

Tabel 5. 8 Kasus Uji Mencari Informasi Penjualan Rumah

Nama Kasus Uji	Mencari Informasi Penjualan Rumah
Objek uji	Kebutuhan Fungsional (SRS_01)
Tujuan pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional mencari informasi penjualan rumah.
Prosedur uji	1. Membuka aplikasi 2. Memasukkan kata pencarian dan penyaringan 3. Menekan tombol cari
Hasil yang diharapkan	Aplikasi akan menampilkan daftar informasi penjualan rumah yang sesuai dengan kata pencarian dan penyaringan yang dimasukkan.

Terlihat pada Gambar 5.14 merupakan hasil pencarian informasi penjualan rumah dengan memasukkan kata pencarian “malang” dan penyaringan lalu menekan tombol *search* .



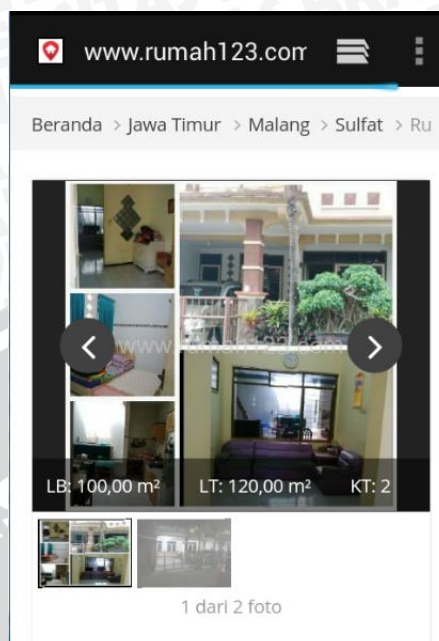
Gambar 5. 14 Tampilan hasil pencarian

5.2.2.2 Kasus Uji Melihat Informasi Detail Penjualan Rumah

Tabel 5. 9 Kasus Uji Melihat Informasi Detail Penjualan Rumah

Nama Kasus Uji	Melihat Informasi Detail Penjualan Rumah
Objek uji	Kebutuhan Fungsional (SRS_02)
Tujuan pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional melihat informasi detail penjualan rumah.
Prosedur uji	1. Melakukan proses memasukkan kata pencarian dan penyaringan 2. Memilih salah satu iklan yang diinginkan yang tampil di daftar informasi penjualan rumah
Hasil yang diharapkan	Aplikasi akan menampilkan informasi detail penjualan rumah dengan cara memuat alamat asal iklan rumah.

Terlihat pada Gambar 5.15 merupakan tampilan informasi detail penjualan rumah dengan memilih salah satu informasi pada daftar penjualan rumah.



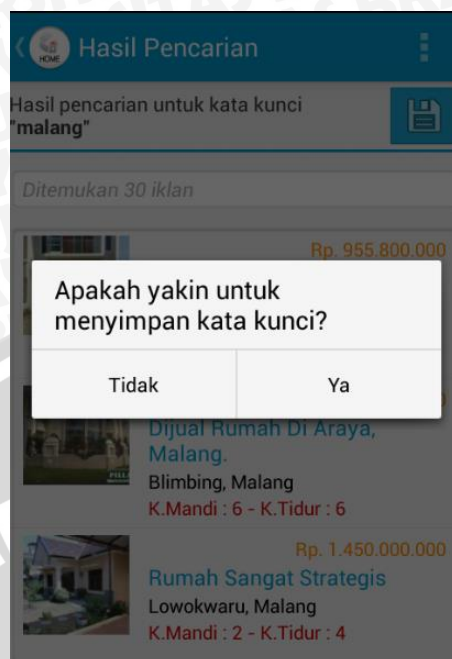
Gambar 5. 15 Tampilan detail iklan rumah

5.2.2.3 Kasus Uji Menyimpan Kata Pencarian

Tabel 5. 10 Kasus Uji Menyimpan Kata Pencarian

Nama Kasus Uji	Melihat Menyimpan Kata Pencarian
Objek uji	Kebutuhan Fungsional (SRS_03)
Tujuan pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional menyimpan kata pencarian.
Prosedur uji	1. Melakukan proses memasukkan kata pencarian dan penyaringan 2. Pada halaman daftar informasi penjualan rumah menekan tombol simpan untuk meyimpan kata pencarian.
Hasil yang diharapkan	Aplikasi akan menyimpan kata pencarian ke basis data dan dapat dilihat pada halaman riwayat.

Terlihat pada Gambar 5.16 merupakan tampilan saat menyimpan kata kunci pencarian.



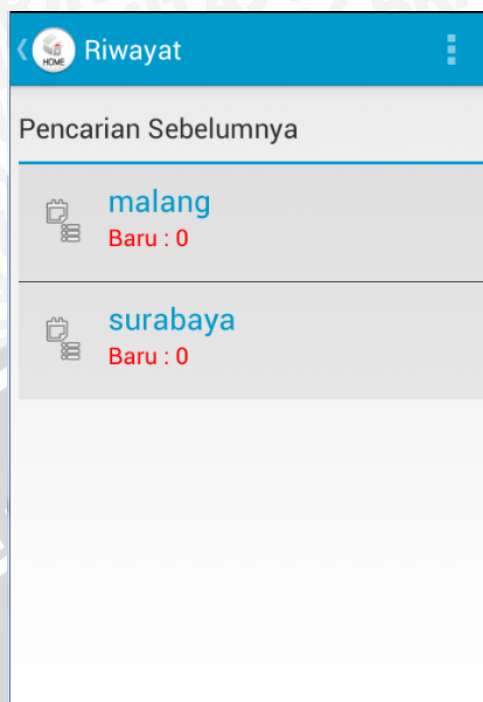
Gambar 5. 16 Tampilan saat akan menyimpan kata kunci

5.2.2.4 Kasus Uji Melihat Kata Pencarian Tersimpan

Tabel 5. 11 Kasus Uji Melihat Kata Pencarian Tersimpan

Nama Kasus Uji	Melihat Kata Pencarian Tersimpan
Objek uji	Kebutuhan Fungsional (SRS_04)
Tujuan pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional melihat kata pencarian tersimpan.
Prosedur uji	1. Membuka aplikasi. 2. Memilih menu riwayat.
Hasil yang diharapkan	Aplikasi akan menampilkan kata pencarian yang tersimpan pada halaman riwayat.

Terlihat pada Gambar 5.17 merupakan halaman riwayat dimana kata pencarian yang telah disimpan sebelumnya ditampilkan.



Gambar 5. 17 Tampilan halaman riwayat

5.2.2.5 Kasus Uji Menghapus Kata Pencarian

Tabel 5. 12 Kasus Uji Menghapus Kata Pencarian

Nama Kasus Uji	Menghapus Kata Pencarian
Objek uji	Kebutuhan Fungsional (SRS_05)
Tujuan pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional menghapus kata pencarian.
Prosedur uji	1. Membuka aplikasi 2. Memilih menu riwayat 3. Memilih kata pencarian yang ingin dihapus
Hasil yang diharapkan	Aplikasi akan menghapus kata pencarian yang dipilih.

Terlihat pada Gambar 5.18 merupakan tampilan saat akan menghapus kata kunci pencarian.



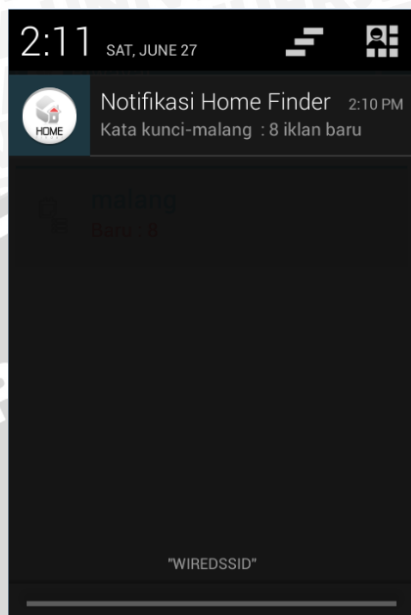
Gambar 5. 18 Tampilan menghapus kata kunci

5.2.2.6 Kasus Uji Mendapat Notifikasi

Tabel 5. 13 Kasus Uji Mendapat Notifikasi

Nama Kasus Uji	Mendapat Notifikasi
Objek uji	Kebutuhan Fungsional (SRS_06)
Tujuan pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional mendapat notifikasi
Prosedur uji	1. Menyimpan kata kunci pada halaman daftar iklan 2. Mengaktifkan notifikasi dan mengatur waktu interval. 3. Menghubungkan emulator pada localhost. 4. Menyiapkan data plain HTML sebagai perumpamaan data baru.
Hasil yang diharapkan	Aplikasi akan mengirimkan notifikasi ke pengguna apabila terdapat informasi penjualan rumah baru.

Terlihat pada Gambar 5.19 merupakan tampilan notifikasi kata kunci “malang” dengan jumlah iklan baru.



Gambar 5. 19 Tampilan notifikasi

5.2.2.7 Hasil Pengujian Validasi

Tabel 5. 14 Hasil Pengujian Validasi

No .	Nama Kasus Uji	Hasil Yang Diharapkan	Hasil Yang Didapatkan	Status Validasi
1.	Mencari Informasi Penjualan Rumah	Aplikasi akan menampilkan daftar informasi penjualan rumah yang sesuai dengan kata pencarian dan penyaringan yang dimasukkan.	Aplikasi berhasil menampilkan daftar informasi penjualan rumah sesuai dengan kata pencarian dan penyaringan yang dimasukkan	Valid
2.	Melihat Informasi Detail Penjualan Rumah	Aplikasi akan menampilkan informasi detail penjualan rumah dengan cara memuat	Aplikasi berhasil menampilkan informasi detail penjualan rumah dengan cara memuat	Valid

		alamat asal iklan rumah.	alamat asal iklan rumah.	
3.	Melihat Menyimpan Kata Pencarian	Aplikasi akan menyimpan kata pencarian ke basis data dan dapat dilihat pada halaman riwayat.	Aplikasi berhasil menyimpan kata pencarian ke basis data dan dapat dilihat pada halaman riwayat.	Valid
4.	Melihat Kata Pencarian Tersimpan	Aplikasi akan menampilkan kata pencarian yang tersimpan pada halaman riwayat.	Aplikasi berhasil menampilkan kata pencarian yang tersimpan pada halaman riwayat.	Valid
5.	Menghapus Kata Pencarian	Aplikasi akan menghapus kata pencarian yang dipilih dari basis data.	Aplikasi berhasil menghapus kata pencarian yang dipilih dari basis data.	Valid
6.	Mendapat Notifikasi	Aplikasi akan mengirimkan notifikasi ke pengguna apabila terdapat informasi penjualan rumah baru.	Aplikasi berhasil mengirimkan notifikasi ke pengguna apabila terdapat informasi penjualan rumah baru.	Valid

5.2.3 Pengujian Usability

Pengujian *usability* dilakukan dengan menggunakan teknik kuisioner dengan jumlah pertanyaan sebanyak 15 butir serta jumlah responden sebanyak 30 orang. Dari 30 responden tersebut 5 diantaranya adalah masyarakat Indonesia yang bergerak di bidang non-IT dan 25 diantaranya di bidang IT. Metode skala likert digunakan pada kuisioner ini dalam mendapatkan jawaban atau *feedback* dari responden. Skala Likert dirancang untuk meyakinkan responden menjawab dalam

berbagai tingkatan pada setiap butir pertanyaan atau pernyataan yang terdapat dalam kuisioner. Responder diminta melakukan penilaian dengan cara melakukan *check* pada kolom range angka antara 1 sampai 5 pada setiap pertanyaan atau pernyataan dengan keterangan sebagai berikut.

1 = Sangat Tidak Setuju

2 = Tidak Setuju

3 = Netral

4 = Setuju

5 = Sangat Setuju

Kuisioner yang diberikan pada responder, dibuat berdasarkan 5 *quality components usability* yaitu *learnability*, *efficiency*, *memorability*, *errors* dan *satisfaction*. Kuisioner dibuat dengan referensi pada sebuah jurnal mengenai pengukuran *usability* sistem menggunakan kuisioner [USA - 14].

Pengujian *usability* digunakan untuk menunjukkan apakah perangkat lunak yang sudah dibuat dapat digunakan dengan baik oleh calon pengguna dengan menggunakan kaidah 5 *quality components* pada *usability* yaitu *learnability*, *efficiency*, *memorability*, *errors* dan *satisfaction*.

1. *Learnability*

Pada komponen *learnability*, diberikan pertanyaan dengan tujuan untuk mengetahui tingkat kemudahan pengguna atau *user* dalam menyelesaikan tugas-tugas dasar ketika pertama kali melihat atau berhadapan dengan sistem yang ada.

No.	Pertanyaan
1	Apakah teks yang ada pada system jelas dan tidak menimbulkan ambigui?
2	Apakah nama menu yang ada sudah ditulis dengan logis dan dapat anda pahami?
3	Apakah informasi yang ditampilkan pada setiap halaman telah memungkinkan pengguna untuk dapat mengambil sebuah keputusan?
4	Apakah setiap halaman memiliki judul yang jelas dan informatif?

2. Efficiency

Pada komponen *efficiency*, diberikan pertanyaan dengan tujuan untuk mengetahui seberapa cepat pengguna dapat menyelesaikan tugas-tugas yang ada saat pengguna pertama kali menggunakan aplikasi.

No.	Pertanyaan
1	Apakah saat menu yang anda klik dapat menampilkan dengan cepat?
2	Apakah saat mengetik nama lokasi pada saat pencarian, system memberikan pilihan daftar nama lokasi sesuai dengan awalan teks yang anda ketik?
3	Apakah pilihan daftar nama lokasi yang disajikan oleh sistem pada saat pencarian, sesuai dengan yang anda maksud?

3. Memorability

Pada komponen *memorability*, diberikan pertanyaan dengan tujuan untuk mengetahui tingkat kemudahan pengguna dalam menggunakan sistem dengan baik, setelah beberapa lama tidak menggunakannya.

No.	Pertanyaan
1	Apakah sistem mudah dioperasikan?
2	Apakah terminologi penamaan pada menu dan halaman telah sesuai dengan konten?
3	Apakah menu dan halaman pada sistem mudah diingat?
4	Apakah <i>icon</i> yang ada telah lazim dan sudah anda kenal?

4. Errors

Pada komponen *errors*, diberikan pertanyaan dengan tujuan untuk mengetahui kemungkinan terjadinya *error* atau kesalahan yang dilakukan oleh pengguna dan seberapa mudah mereka dapat mengatasinya.

No.	Pertanyaan
1	Apakah pesan kesalahan menginformasikan penyebab kesalahan terjadi?

2	Apakah sistem sebisa mungkin mencegah anda untuk melakukan kesalahan?
---	---

5. Satisfaction

Pada komponen *satisfaction*, diberikan pertanyaan dengan tujuan untuk mengetahui tingkat kepuasan pengguna dalam menggunakan sistem yang telah dibuat.

No.	Pertanyaan
1	Apakah semua fitur yang ada telah mencakup informasi atau kebutuhan yang diharapkan?
2	Apakah anda merasa terbantu dengan adanya aplikasi ini dalam memonitoring cuaca dan gempa di Indonesia?

Tabel 5. 15 Hasil Pengujian *Usability*

No	Pertanyaan	Jumlah				
		1	2	3	4	5
1	Apakah teks yang ada pada sistem jelas dan tidak menimbulkan ambigu?	0	2	1	14	13
2	Apakah nama menu yang ada sudah ditulis dengan logis dan dapat anda pahami?	0	0	1	12	17
3	Apakah informasi yang ditampilkan pada setiap halaman telah memungkinkan pengguna untuk dapat mengambil sebuah keputusan?	0	0	4	10	16
4	Apakah setiap halaman memiliki judul yang jelas dan informatif?	0	0	2	7	21
5	Apakah saat menu yang anda klik dapat menampilkan dengan cepat?	0	0	4	9	17
6	Apakah saat mengetik kata pencarian, sistem memberikan pilihan daftar iklan rumah sesuai dengan masukkan awal?	0	0	3	12	15
7	Apakah pilihan daftar iklan rumah yang disajikan oleh sistem pada saat pencarian, sesuai dengan yang anda maksud?	0	0	0	16	14

8	Apakah sistem mudah dioperasikan?	0	0	0	9	21
9	Apakah terminologi penamaan pada menu dan halaman telah sesuai dengan konten?	0	0	0	11	19
10	Apakah menu dan halaman pada sistem mudah diingat?	0	0	3	5	22
11	Apakah <i>icon</i> yang ada telah lazim dan sudah anda kenal?	0	0	1	13	16
12	Apakah pesan kesalahan menginformasikan penyebab kesalahan terjadi?	0	0	1	15	14
13	Apakah sistem sebisa mungkin mencegah anda untuk melakukan kesalahan?	0	0	5	16	9
14	Apakah semua fitur yang ada telah mencakup informasi atau kebutuhan yang diharapkan?	0	0	2	10	18
15	Apakah anda merasa terbantu dengan adanya aplikasi ini dalam mencari informasi penjualan rumah?	0	0	1	11	18

Keterangan :

STS : Sangat Tidak Setuju

S : Setuju

TS : Tidak Setuju

SS : Sangat Setuju

N : Setuju

5.3 Analisis Hasil Pengujian

Proses analisis terhadap hasil pengujian dilakukan untuk mendapatkan kesimpulan dari hasil pengujian aplikasi mesin pencari informasi penjualan rumah yang telah dilakukan. Proses analisis mengacu pada hasil pengujian yang didapatkan. Analisis dilakukan meliputi analisis hasil pengujian unit, pengujian validasi dan analisis hasil pengujian *usability*.

5.3.1 Analisis Hasil Pengujian Unit

Proses analisis pengujian unit dilakukan dengan melihat kesesuaian fungsi dari implementasi unit fungsi yang diujikan dengan hasil perancangan perangkat lunak yang telah dirancang sebelumnya. Berdasarkan hal tersebut, maka dapat

diambil kesimpulan bahwa unit fungsi yang diujikan dari program sudah memenuhi kebutuhan fungsional yang telah dirancang pada tahap perancangan.

5.3.2 Analisis Hasil Pengujian Validasi

Proses analisis terhadap hasil pengujian validasi dilakukan dengan melihat kecocokan antara hasil kinerja sistem dengan daftar kebutuhan. Berdasarkan hasil pengujian validasi dapat disimpulkan bahwa implementasi dan fungsionalitas aplikasi mesin pencari informasi penjualan rumah telah memenuhi kebutuhan fungsional yang dijabarkan pada tahap analisis kebutuhan.

5.3.3 Analisis Hasil Pengujian Usability

Proses analisis terhadap hasil pengujian usability dilakukan dengan menggunakan skala likert. Interpretasi skor likert atau presentase dari setiap skor likert ditunjukkan pada Tabel 5.16. Hasil perhitungan index persentase dari setiap pertanyaan ditunjukkan pada Tabel 5.17 dan hasil status pengujian usability ditunjukkan pada Tabel 5.18. Untuk rumus perhitungan total skor ditunjukkan pada persamaan 5.1 sedangkan rumus perhitungan index persentase ditunjukkan pada persamaan 5.2 [EVO - 14].

Tabel 5. 16 Interpretasi Skor Likert

Skor Likert	Interpretasi skor dengan Interval = 20	Pilihan
1	0% - 19%	Sangat Tidak Setuju
2	20% - 39%	Tidak Setuju
3	40% - 60%	Netral
4	60% - 79%	Setuju
5	80% - 100%	Sangat Setuju

[Sumber : SIS - 14]

Tabel 5. 17 Hasil indeks persentase

No	Pertanyaan	STS	TS	N	S	SS	Total	Index(%)
<i>Learnability</i>								
1	Apakah teks yang ada pada sistem jelas dan tidak menimbulkan ambigu?	0	2	1	14	13	128	85,33
2	Apakah nama menu yang ada sudah ditulis dengan logis dan dapat anda pahami?	0	0	1	12	17	136	90,67
3	Apakah informasi yang ditampilkan pada setiap halaman telah memungkinkan pengguna untuk dapat mengambil sebuah keputusan?	0	0	4	10	16	132	88,00
4	Apakah setiap halaman memiliki judul yang jelas dan informatif?	0	0	2	7	21	139	92,67
<i>Efficiency</i>								
5	Apakah saat menu yang anda klik dapat menampilkan dengan cepat?	0	2	1	14	13	128	85,33
6	Apakah saat mengetik kata pencarian, sistem memberikan pilihan daftar iklan rumah sesuai dengan masukkan awal?	0	0	1	12	17	136	90,67
7	Apakah pilihan daftar iklan rumah yang disajikan oleh	0	0	4	10	16	132	88,00

	sistem pada saat pencarian, sesuai dengan yang anda maksud?							
Memorability								
8	Apakah sistem mudah dioperasikan?	0	2	1	14	13	128	85,33
9	Apakah terminologi penamaan pada menu dan halaman telah sesuai dengan konten?	0	0	1	12	17	136	90,67
10	Apakah menu dan halaman pada sistem mudah diingat?	0	0	4	10	16	132	88,00
11	Apakah <i>icon</i> yang ada telah lazim dan sudah anda kenal?	0	0	2	7	21	139	92,67
Errors								
12	Apakah pesan kesalahan menginformasikan penyebab kesalahan terjadi?	0	2	1	14	13	128	85,33
13	Apakah sistem sebisa mungkin mencegah anda untuk melakukan kesalahan?	0	0	1	12	17	136	90,67
Satisfaction								
14	Apakah semua fitur yang ada telah mencakup informasi atau kebutuhan yang diharapkan?	0	2	1	14	13	128	85,33
15	Apakah anda merasa terbantu dengan adanya aplikasi ini	0	0	1	12	17	136	90,67

	dalam mencari informasi penjualan rumah?								
--	--	--	--	--	--	--	--	--	--

Keterangan :

STS : Sangat Tidak Setuju

S : Setuju

TS : Tidak Setuju

SS : Sangat Setuju

N : Setuju

$$\text{Total Skor} = S_{sts} \times 1 + S_{ts} \times 2 + S_n \times 3 + S_s \times 4 + S_{ss} \times 5 \quad (5.1)$$

$$\text{Indeks (\%)} = \frac{\text{Total Skor}}{Y} \times 100\% \quad (5.2)$$

Y = Skor Likert tertinggi x jumlah responden

Tabel 5. 18 Hasil status pengujian *usablitiy*

No	Aspek Penilaian	Rata-rata Persentase (%)	Status
1	<i>Learnability</i>	89,17	Sangat Memuaskan
2	<i>Efficiency</i>	88,67	Sangat Memuaskan
3	<i>Memorability</i>	92,33	Sangat Memuaskan
4	<i>Errors</i>	85,67	Sangat Memuaskan
5	<i>Satisfaction</i>	91,00	Sangat Memuaskan

Dari hasil pengujian *usability* yang telah dilakukan, menyatakan bahwa index persentase (%) yang didapatkan pada setiap aspek penilaian *usability*, menunjukkan hasil rata-rata sebesar 89,37% yaitu dengan status sangat memuaskan. Dengan adanya hasil tersebut menunjukkan aplikasi layak diterima dan memudahkan pengguna dengan hasil sangat memuaskan.

BAB VI

PENUTUP

6.1 Kesimpulan

Berdasarkan hasil analisis, perancangan, implementasi dan pengujian yang dilakukan, maka diambil kesimpulan sebagai berikut :

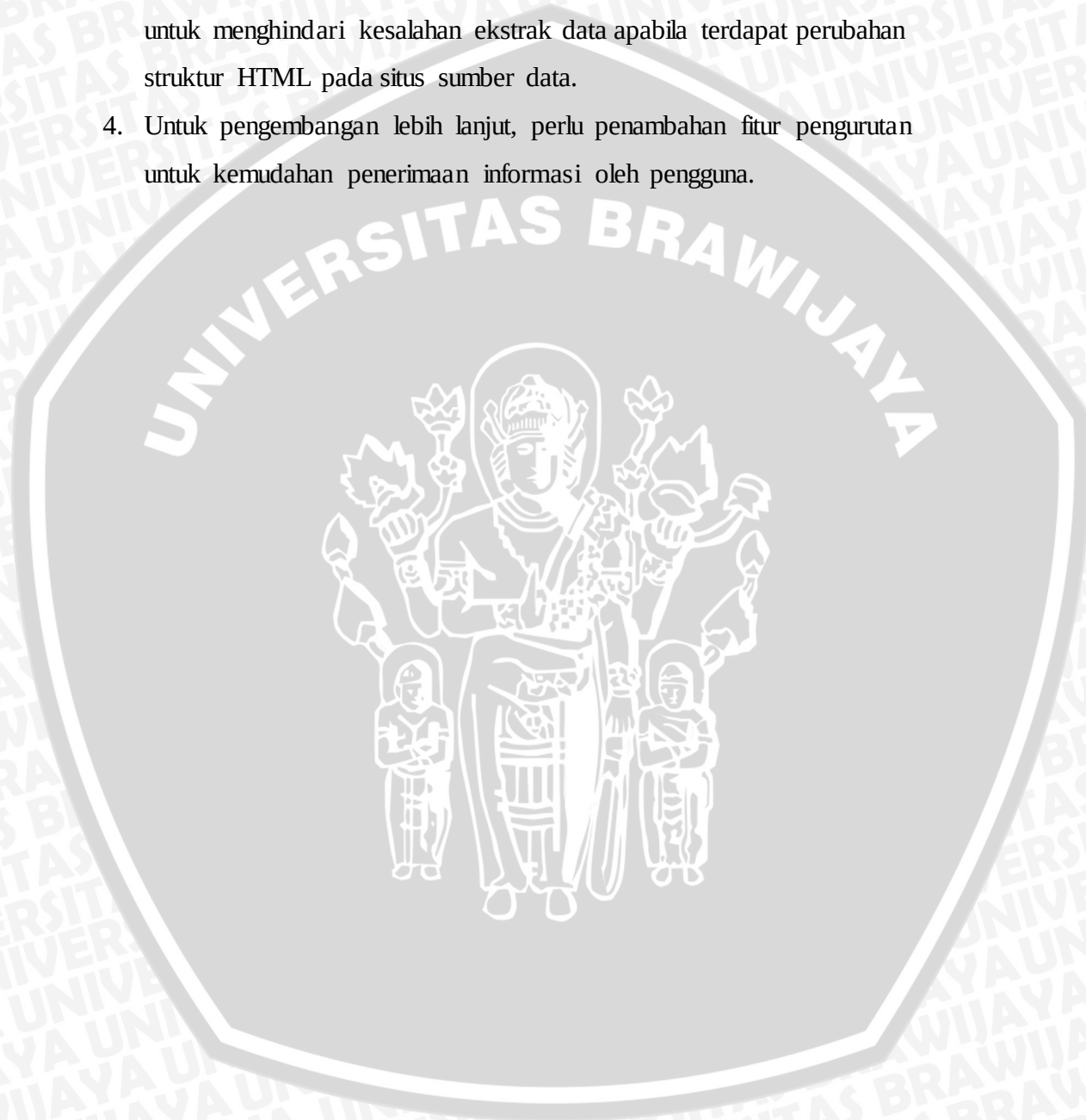
1. Berdasarkan hasil pengujian, disimpulkan bahwa data berhasil diekstrak dan diseragamkan hasil dari *scraping* 3 situs penjualan rumah.
2. Perancangan aplikasi mesin pencari informasi penjualan rumah telah dibuat sesuai dengan spesifikasi kebutuhan yang telah dianalisis.
3. Implementasi aplikasi mesin pencari informasi penjualan rumah dibuat dengan menggunakan *library* JSOUP dan data informasi penjualan rumah diperoleh dari proses *web scraping* 3 situs penjualan rumah yang sudah ditentukan sebelumnya.
4. Implementasi *alur proses* notifikasi pada aplikasi mesin pencari informasi penjualan rumah digunakan untuk memberitahu pengguna apabila terdapat informasi penjualan rumah yang baru .
5. Berdasarkan hasil pengujian validasi dengan metode *black box testing* pada sistem ini menunjukkan bahwa status validitas valid. Sistem ini sudah memenuhi kebutuhan fungsional yang telah dianalisis.
6. Berdasarkan hasil pengujian *usability* bahwa aplikasi layak diterima atau dapat digunakan dengan mudah oleh pengguna dengan indeks persentase rata-rata yang dihasilkan dari semua aspek penilaian sebesar 89,37% dengan status sangat memuaskan.

6.2 Saran

Saran yang dapat diberikan untuk pengembangan aplikasi mesin pencari informasi penjualan rumah selanjutnya antara lain :

1. Untuk pengembangan lebih lanjut, perlu dilakukan penambahan informasi seperti luas bangunan, luas tanah dan status rumah sudah terjual atau belum.

2. Untuk pengembangan lebih lanjut, perlu pengembangan *web service* / server lokal untuk efisiensi dan performa aplikasi lebih baik serta jumlah banyak informasi yang disediakan dapat lebih banyak.
3. Untuk pengembangan lebih lanjut, perlu pengembangan *library scraping* untuk menghindari kesalahan ekstrak data apabila terdapat perubahan struktur HTML pada situs sumber data.
4. Untuk pengembangan lebih lanjut, perlu penambahan fitur pengurutan untuk kemudahan penerimaan informasi oleh pengguna.



DAFTAR PUSTAKA

- [BOH-11] Böhmer, Matthias. dkk. 2011. *Falling asleep with Angry Birds, Facebook and Kindle: a large scale study on mobile app usage*. New York
- [DEW-11] Dewi Rahmatika. 2011. *Perilaku Pencarian Informasi Mahasiswa Dalam Menyusun Skripsi*. USU
- [DEV - 14] Android Developer. 2014. *Push Notification*. [online]. Tersedia di <http://developer.android.com/guide/topics/ui/notifiers/notifications.html>.
[Diakses 5 Februari 2015]
- [EVO - 14] Fitrianto, Rahmanda., 2014. *Sistem Aplikasi Mobile E-Voting Menggunakan Platform Android*. S1. Universitas Brawijaya.
- [GAR-11] Gargenta, Marco. 2011. *Learning Android*. Sebastopol, O'Reilly Media.
- [JSO-15] *Website Official JSOUP*.
URL : www.jsoup.org
[Diakses tanggal 30 Maret 2015]
- [KOM-14] Kompas "Penjualan Rumah Marak di Awal Tahun".
URL:<http://bisniskeuangan.kompas.com/read/2013/02/10/22071814/Penjualan.Rumah.Marak.di.Awal.Tahun>. [Diakses tanggal 5 Desember 2014]
- [LAM-15] Daftar Penjualan rumah situs Lamudi.
URL:<http://www.lamudi.co.id/house/buy/price:01000000000/q:malang/>.
[Diakses tanggal 1 Februari 2015]
- [LUD-12] Ludwig, Sean. 2012. *Mobile app usage grows 35%, TV & web not so much*. Akses dari <http://venturebeat.com/2012/12/05/mobile-app-usage-tv-web-2012>
[Tanggal akses 23 Februari 2015]

- [MAN-14] Bank Mandiri. URL : <http://www.bankmandiri.co.id>, diakses tanggal 5 Desember 2014.
- [MUL-10] Mulyadi 2010, *Membuat Aplikasi untuk Android*, Multimedia Center Publishing, Yogyakarta.
- [PER-12] Perez, Sarah. 2012. *comScore: In U.S. Mobile Market, Samsung, Android Top The Charts; Apps Overtake Web Browsing*. Akses dari <http://techcrunch.com/2012/07/02/comscore-in-u-s-mobile-market-samsung-android-top-the-charts-apps-overtake-web-browsing>.
[Tanggal akses 23 Februari 2015]
- [RAT-12] Ratih Dwi Puspita. "Cari engine". <http://bovis.gyuyet.ch/3dict/323Webde.html>.
[Diakses pada 9 Januari 2015]
- [RKU-15] Daftar penjualan situs Rumahku.
URL: http://www.rumahku.com/dijual/kata_kunci/malang/showadvance:1/
[Diakses tanggal 1 Februari 2015]
- [RUM-15] Daftar penjualan situs Rumah123.
URL: <http://www.rumah123.com/cari?type=rumah&category=dijual&area=malang&sold=1>.
[Diakses tanggal 1 Februari 2015]
- [SIE-08] Siegler, MG. 2008. *Analyst: There's a great future in iPhone apps*. Akses dari <http://venturebeat.com/2008/06/11/analyst-theres-a-great-future-in-iphone-apps>.
[Tanggal akses 23 Februari 2015]
- [SIN-14] Sindonews "Persentase penggunaan smartphone di Indonesia meningkat".
URL: <http://autotekno.sindonews.com/read/911875/122/persentas-e-pengguna-smartphone-di-indonesia-meningkat-1413394479>.

[Diakses tanggal 5 Desember 2014]

- [TEM-14] Tempo “59 Persen Pengguna Internet Akses Via Smartphone ”.
URL:<http://www.tempo.co/read/news/2014/11/27/072624959/59-Persen-Pengguna-Internet-Akses-Via-Smartphone>.

[Diakses tanggal 19 Januari 2015]

- [TES-11] Testiyan Wijaya. 2011. Perancangan Alat Ukur Indeks Usabilitas Pada Mesin Pencari (*Search engine*). Surakarta : Universitas Sebelas Maret.

- [TUR-10] Turland, Matthew.2010. “php architect Guide to web web scraping”. Toronto: Marco Tabini & Associates, Inc.

- [USA - 14] Rahardi Rianto, Dedi. 2014. *Pengujian Usability Sistem Menggunakan Use Questionnaire Pada Aplikasi Android*. Palembang

- [VAR-12] Vargiu, Eloisa et al.2012.”Exploiting web scraping in a collaborative filtering- based approach to web advertising”. *Artificial Intelligence recari*, 2013, Vol. 2, No.1 DOI 10.5430/airv21p44.

- [W3C-15] W3Schools Online Web Tutorials – HTTP Methods GET vs POST
URL : http://www.w3schools.com/tags/ref_httpmethods.asp

[Diakses tanggal 01 Juni 2015]

- [WID-11] Widiaseno, Zulfikar, FX Ferdinand, Ruki Harwahyu dan Reza Hadi S., “Penggunaan Teknik Ekstrasi Web Dalam Pengolahan Referensi Kepustakaan”. Dept. Teknik Elektro Universitas Indonesia.

LAMPIRAN

Lampiran 1. Lembar Kuisioner *Usability Testing*

Mesin Pencari Informasi Penjualan Rumah

Nama :

No. Identitas :

Email :

No. HP :

-FEEDBACK

Berilah tanda check (✓) pada masing-masing pertanyaan dengan jawaban yang menurut Anda paling sesuai!

1 = Sangat Tidak Setuju 3 = Netral 5 = Sangat Setuju
2 = Tidak Setuju 4 = Setuju

No	Pertanyaan	Jumlah				
		1	2	3	4	5
1	Apakah teks yang ada pada sistem jelas dan tidak menimbulkan ambigu?					
2	Apakah nama menu yang ada sudah ditulis dengan logis dan dapat anda pahami?					
3	Apakah informasi yang ditampilkan pada setiap halaman telah memungkinkan pengguna untuk dapat mengambil sebuah keputusan?					
4	Apakah setiap halaman memiliki judul yang jelas dan informatif?					
5	Apakah saat menu yang anda klik dapat menampilkan dengan cepat?					
6	Apakah saat mengetik kata pencarian, sistem memberikan pilihan daftar iklan rumah sesuai dengan masukkan awal?					
7	Apakah pilihan daftar iklan rumah yang disajikan oleh sistem pada saat pencarian, sesuai dengan yang anda maksud?					
8	Apakah sistem mudah dioperasikan?					
9	Apakah terminologi penamaan pada menu dan halaman telah sesuai dengan konten?					
10	Apakah menu dan halaman pada sistem mudah diingat?					

11	Apakah <i>icon</i> yang ada telah lazim dan sudah anda kenal?					
12	Apakah pesan kesalahan menginformasikan penyebab kesalahan terjadi?					
13	Apakah sistem sebisa mungkin mencegah anda untuk melakukan kesalahan?					
14	Apakah semua fitur yang ada telah mencakup informasi atau kebutuhan yang diharapkan?					
15	Apakah anda merasa terbantu dengan adanya aplikasi ini dalam mencari informasi penjualan rumah?					

Kritik :

.....

.....

.....

.....

.....

.....

.....

.....

Malang, Juni 2015

Saran :

.....

.....

.....

.....

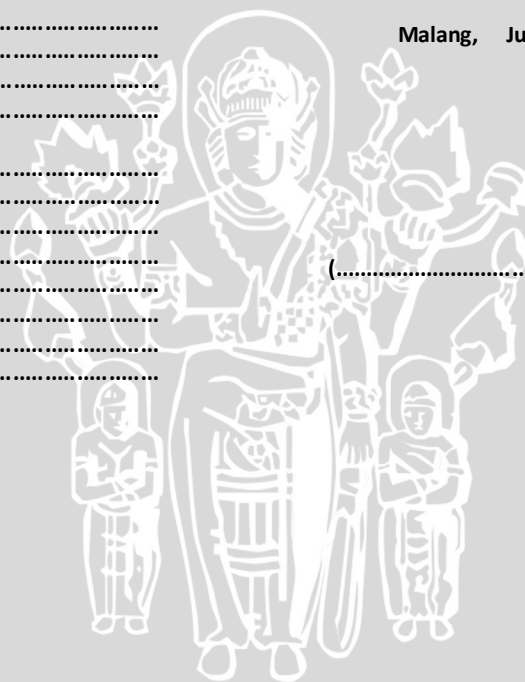
.....

.....

.....

.....

(.....)



Lampiran 2. Hasil *feedback* kuisioner *usability testing*

No	Nama	Pertanyaan														
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	Septy Kusumaning P.	5	5	5	5	5	5	5	5	5	5	5	5	4	4	5
2	Julian Sitoaji S.	4	4	4	5	4	4	4	4	4	4	3	4	3	4	4
3	Zidna Ziada E.	4	4	3	4	4	3	4	5	4	4	4	4	3	4	4
4	Luluk Setiawati H.	4	5	5	5	5	4	4	5	5	5	4	4	4	5	5
5	Rinadewi Astuti	4	5	5	5	5	5	5	5	5	5	5	4	4	5	5
6	Deni	4	5	5	5	5	5	5	5	5	5	5	4	4	5	5
7	M. Angga Huda	4	4	5	4	4	4	5	4	5	5	5	5	4	4	5
8	Fendy Gusta P.	5	4	4	5	4	4	4	5	4	5	4	5	4	5	4
9	Rasita Alfiyah R.	5	5	5	5	3	3	4	5	4	5	5	4	4	4	5
10	Marwa Burhan	5	5	5	5	5	5	5	5	5	5	5	4	4	4	4
11	Ahmad M.	2	4	3	4	4	4	4	4	5	5	4	5	5	4	4
12	Billy Astian	4	4	5	5	5	4	4	5	5	5	5	5	4	3	4
13	Syela U.	2	5	5	5	5	4	4	5	5	5	4	4	4	3	5
14	Nabila	5	5	4	5	5	4	5	5	4	3	4	5	4	5	4
15	Irma	3	3	3	3	5	5	5	5	5	5	4	3	3	4	5
16	Alfiani Fitri	5	4	3	3	4	4	3	4	4	4	4	3	4	3	4
17	Nindy D.	5	4	4	3	3	4	4	4	4	4	4	4	3	3	3
18	Rizal H.	5	5	4	5	5	5	4	5	5	5	5	5	4	5	4
19	Kholish R.	4	5	4	4	4	5	5	4	5	5	4	5	3	4	3
20	Rofiqoh P.	5	5	5	5	5	5	5	5	4	5	5	5	4	4	5
21	Krisdhamara W.	5	5	4	4	5	4	4	4	4	3	4	4	3	4	4
22	Ika Qurriati	4	4	4	4	4	5	5	5	5	4	4	4	4	5	5

23	Nadia Prevani	4	5	5	5	5	3	4	5	5	5	4	4	5	5	5
24	Fitarina A.	5	5	5	5	5	5	5	5	5	5	4	4	5	4	4
25	Fenty Susanti	4	4	4	5	4	4	4	5	4	4	5	4	4	4	5
26	Rainey Merdana	4	4	4	5	5	5	4	4	4	4	5	5	4	4	5
27	Raka Gilang	4	4	5	5	4	5	4	4	5	5	5	4	3	4	3
28	Asfarina	5	5	5	5	5	5	5	4	5	4	5	3	5	4	4
29	Claudio F.	4	4	5	5	4	4	4	4	5	5	3	3	3	3	4
30	Dika F.	5	5	5	5	5	5	4	4	5	4	5	3	4	4	5