

**Implementasi *Software Center* Menggunakan Metode
Evolutionary Prototyping Berbasis HTML5**

Resume Skripsi

untuk memenuhi sebagian persyaratan mencapai derajat Sarjana S-1



disusun oleh :

ESTU FARDANI

NIM. 09650004

Kepada

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS SAINS DAN TEKNOLOGI

UNIVERSITAS ISLAM NEGERI SUNAN KALIJAGA

YOGYAKARTA

2014

KATA PENGANTAR

Alhamdulillahirabbil'alamiin, Puji syukur penulis panjatkan kepada Allah SWT karena dengan restu-Nya pelaksanaan dan penyusunan skripsi yang berjudul “Implementasi *Software Center* Menggunakan Metode *Evolutionary Prototyping* Berbasis HTML5” dapat diselesaikan sebagai persyaratan menyelesaikan Sarjana Strata Satu (S1) Jurusan Teknik Informatika, Fakultas Sains dan Teknologi Universitas Islam Negeri Sunan Kalijaga Yogyakarta.

Penulisan skripsi ini tidak terlepas dari bantuan dan dukungan pelbagai pihak. Dalam kesempatan ini penulis mengucapkan terima kasih dan penghargaan yang sebesar-besarnya kepada seluruh pihak yang telah memberikan bantuan dan dukungan serta bimbingan.

Semoga Allah SWT memberikan balasan yang berlipat ganda dari segala kebaikan yang telah dilakukan. Penulis menyadari bahwa skripsi ini masih belum sempurna. Oleh karena itu, kritik dan saran yang membangun penulis harapkan sebagai masukan kearah yang lebih baik lagi.

Demikian penulis berharap agar laporan penelitian ini dapat berguna untuk kita semua.

Yogyakarta, 7 April 2014

Penulis

Implementasi *Software Center* Menggunakan Metode *Evolutionary Prototyping* Berbasis HTML5

Estu Fardani¹

09650004

¹ Universitas Islam Negeri Sunan Kalijaga

surel: estu@jogja.linux.or.id

INTISARI

Software Center adalah aplikasi wajib yang harus dimiliki oleh sistem operasi, baik sistem operasi konvensional maupun pada perangkat bergerak. *BlankOn* sebagai sistem operasi produksi dalam negeri harusnya memiliki aplikasi ini, namun dalam kenyataannya aplikasi ini belum tersedia. Trend pengembangan aplikasi mengerucut kepada teknologi HTML5 sebagai teknologi yang mudah diadaptasi hingga ke ranah desktop. Penelitian ini mencoba membuat *software center* berbasis HTML5 sebagai solusi di *BlankOn*.

Penelitian ini menggunakan metode *evolutionary prototyping* memungkinkan pengembangan aplikasi berdasarkan contoh produk yang akan dibuat. Kemudian hasil *prototyping* diimplementasikan dalam bahasa pemrograman. Setiap tahap implementasi akan dilakukan koreksi untuk mendapatkan fitur dan fungsi yang diinginkan. Proses pengembangan akan selesai jika *prototyping* yang dikerjakan telah memenuhi fitur yang diharapkan. Tahapan penelitian terbagi dalam beberapa langkah yakni analisis aplikasi sejenis, perancangan sistem, implementasi dan pengujian sistem.

Berdasarkan hasil pengujian fungsionalitas *software center*, didapatkan presentase 100% setuju sistem dapat berjalan dengan baik. Dari hasil pengujian dari segi *usability software center*, diperoleh kesimpulan bahwa responden sangat setuju sebanyak 80% dan setuju sebanyak 20%. Jadi dapat disimpulkan bahwa aplikasi *software center* dapat berjalan dengan baik dan lancar.

Kata Kunci : *Vala, Webkit, HTML5, Software Center, BlankOn*

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Linux BlankOn merupakan sebuah distribusi Linux turunan Debian yang dikembangkan oleh Yayasan Penggerak Linux Indonesia (YPLI) dan Tim Pengembang BlankOn. Distribusi ini dirancang sesuai dengan kebutuhan pengguna komputer umum di Indonesia.

Software Center memiliki fungsi untuk mempermudah manajemen perangkat lunak, yaitu menambah dan menghapus perangkat lunak yang ada, keberadaannya diperlukan dalam semua sistem operasi baik desktop maupun perangkat bergerak.

Pemasangan aplikasi di Linux BlankOn saat ini bisa dilakukan dengan bantuan terminal (konsole).

Aplikasi sejenis yang telah ada adalah *Ubuntu Software Center* dari distribusi Ubuntu. Aplikasi ini berlisensi *OpenSource* sehingga memungkinkan diterapkan pada Linux BlankOn. Namun karena aplikasi ini sudah menjadi *trademark* Ubuntu, dibutuhkan proses modifikasi lebih lanjut. Proses ini membutuhkan waktu dan tidak menjamin akan berjalan baik sehingga, muncul pemikiran bagaimana jika membuatnya mulai dari awal.

Trend aplikasi yang berkembang belakangan ini mengerucut pada penggunaan teknologi HTML5. Hal ini dikarenakan kemampuan adaptasi yang baik disegala platform, kemudahan dalam mempercantik desain dengan bantuan CSS3, tampilan yang responsif akibat dukungan javascript yang lebih baik dan kelebihan-kelebihan HTML5 lainnya.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas dapat dirumuskan permasalahan yang akan diselesaikan dalam penelitian ini adalah:

1. Bagaimana melakukan analisis dan mengembangkan *prototype Software Center* di Linux BlankOn?
2. Bagaimana mengimplementasikannya dalam bahasa pemograman HTML5?

1.3 Batasan Masalah

Berdasarkan rumusan masalah dan dengan maksud agar pembahasan dan penyusunan penelitian dapat dilakukan secara terarah dan tercapai sesuai dengan yang diharapkan maka perlu ditetapkan batasan – batasan dari masalah yang dihadapi. Adapun batasan-batasan masalah pada penelitian ini adalah sebagai berikut :

1. Aplikasi ini hanya berjalan pada Sistem Operasi Linux BlankOn 9.0 Suroboyo 64-Bit,
2. Aplikasi ini hanya menampung perangkat lunak yang tersedia pada *repository* Linux BlankOn,
3. Kategori aplikasi dibatasi hanya 4 jenis, perkantoran, Internet, audio video, aksesoris, masing-masing katergori hanya menyediakan 4 sampel aplikasi,
4. Aplikasi masih merupakan produk contoh (*prototype*).
5. Sumber *repository* hanya dibatasi tiga tautan (Arsip BlankOn, Repo UGM, Kambing UI)

1.4 Tujuan Penelitian

Tujuan penelitian ini adalah sebagai berikut :

1. Melakukan analisis dan mengembangkan aplikasi yang dapat membantu

pengguna untuk manajemen perangkat lunak yang ada di Linux BlankOn,

2. Mengimplementasikan *Software Center* di Linux BlankOn menggunakan bahasa pemrograman HTML5.

1.5 Manfaat Penelitian

Dengan adanya penelitian ini, diharapkan :

1. Memberi kemudahan pengguna Linux BlankOn untuk manajemen perangkat lunak yang ada dan terpasang.
2. Dapat memberikan solusi bagi pengembang BlankOn agar menjadi distribusi Linux yang lebih baik.

1.6 Keaslian Penelitian

Pengembangan *Software Center* sudah pernah dilakukan sebelumnya. Pada tahun 2005 penelitian dimulai oleh Matthew Paul Thomas dengan membuat prototipe "*Ubuntu Software Center*". *Software Center* ini nantinya diharapkan bisa membantu pengguna untuk memasang, membeli dan menghapus aplikasi di Ubuntu. Matthew Paul Thomas merupakan bagian tim *Canonical*, perusahaan yang mengembangkan dan penyedia dukungan komersial distribusi Linux Ubuntu. Penelitian serupa pernah dilakukan oleh Matthias Klumpp pada tahun 2012. Penelitian dilakukan dengan menerjemahkan *Ubuntu Software Center* agar mampu berjalan di distribusi *Debian* dan *Fedora*. Bahasa yang digunakan adalah python dan C.

Penulis berkeyakinan bahwa pengembangan *Software Center* di distribusi Linux BlankOn dengan penggunaan HTML5 sebagai bahasa pemrograman belum pernah dilakukan.

BAB 2

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Penelitian-penelitian yang terdahulu mengenai *Software Center* dikembangkan dengan bahasa dan konsep yang berbeda. Sebagai perbandingan dapat dilihat pada Tabel 2.1.

Tabel 2.1: Perbedaan Penelitian Terdahulu

No	Nama	Judul	Kelebihan	Kekurangan
1	Apple Inc, 2008	App Store	Integrasi dengan seluruh layanan Apple	Hanya bisa berjalan untuk sistem operasi milik Apple
2	Google Inc, 2008	Google Play	Tersedia di semua platform android, dan ponsel pintar android. Integrasi dengan akun google.	Hanya berjalan di sistem operasi android.
3	RIM, 2010	BlackBerry App World	Menyediakan aplikasi yang bisa dipasang pada blackberry	Hanya tersedia di BBOs 7 keatas, dan ponsel pintar BBOS
4	Paul Thomas, 2005	<i>Prototyping Software Center</i>	Menjelaskan secara detail apa yang harus dimiliki <i>Software Center</i>	Masih merupakan aplikasi <i>prototype</i>
5	Canonical, 2010	<i>Ubuntu Software Center</i>	Menyediakan antarmuka yang baik, kemudahan dalam penggunaan.	Terbatas hanya di bisa terpasang pada Linux Ubuntu
6	SUSE, 2010	YaST	Memudahkan pengguna untuk mengatur apapun di OpenSUSE	Hanya tersedia untuk Linux

				OpenSUSE
7	Šrain, 2006	YaST Architecture	Membahas bagaimana YaST dikembangkan	-
8	Klumpp, 2012	<i>Software Center for Fedora</i>	Bisa dijalankan pada distro lain, misal debian dan fedora	Masih versi beta, dikembangkan dengan bahasa C dan Python
9	Khusna Ismail & Nur Fajri, 2013	Aplikasi Permainan Casual Game “Becak Pursuit” Berbasis HTML 5	Mudah diakses melalui web	Masih sederhana
10	Pratami, 2013	Rancang Bangun Game Hijaiyah Match dengan HTML5	Dapat berjalan baik pada web browser	Belum dapat berjalan baik pada mobile browser

2.2 Landasan Teori

2.2.1 Mock-up

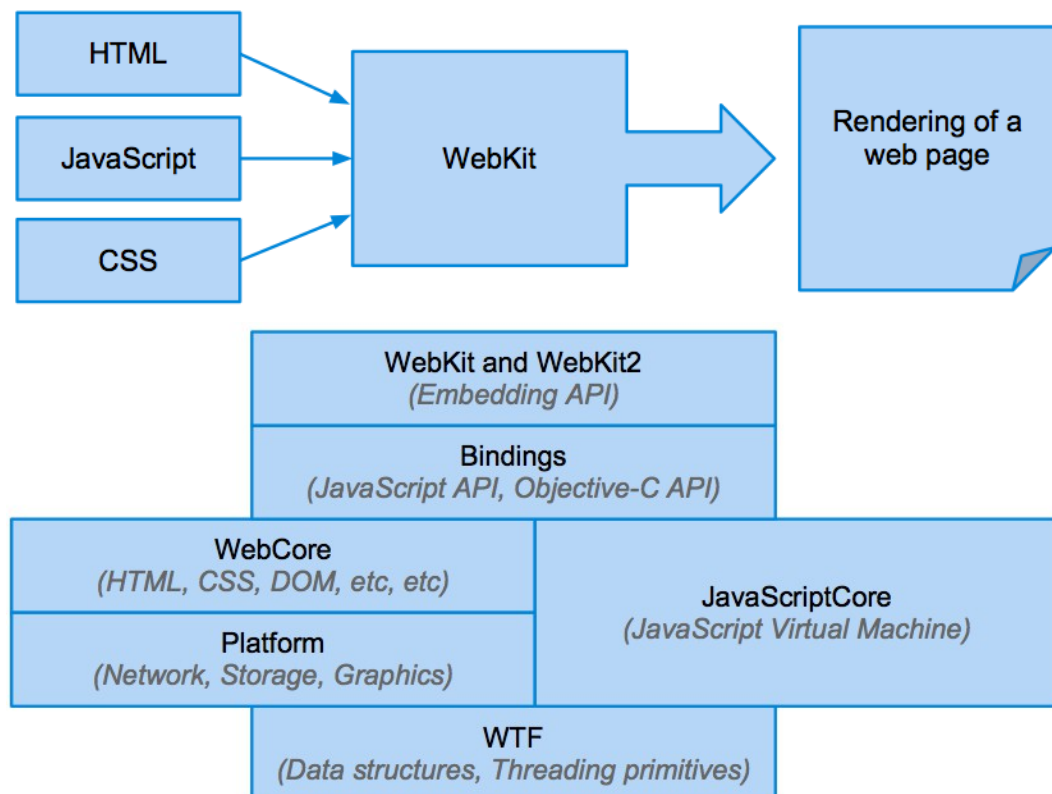
Mock-up adalah *prototype* sangat awal. Mock-up bertujuan untuk memperoleh umpan balik dari pengguna tentang desain dan ide-ide desain awal dalam proses desain (Stroik, 2010).

2.2.2 Vala

Vala adalah bahasa pemrograman baru yang bertujuan untuk membawa fitur bahasa pemrograman modern pengembang *GNOME* tanpa memaksakan persyaratan runtime tambahan dan tanpa menggunakan *ABI* yang berbeda dibandingkan dengan aplikasi dan perpustakaan yang ditulis dalam C (Anwari, 2013).

2.2.3 WebKitGTK+

WebKitGTK+ adalah platform port GNOME untuk mesin rendering WebKit. Menawarkan fungsionalitas penuh WebKit melalui satu set API berbasis *GObject*, kompetibel untuk proyek-proyek yang membutuhkan segala jenis integrasi web, dari aplikasi hybrid hibrida HTML / CSS hingga web browser penuh. Fungsionalitas WebKit berguna dalam pelbagai sistem dari komputer desktop untuk embedded system seperti telepon, tablet, dan televisi. WebKitGTK+ dibuat oleh komunitas pengembang dan desainer, untuk membawa platform web untuk semua orang (Barth, 2012).



Gambar 2.1: Cara Kerja Webkit

2.2.4 Packaging Tools

Packaging Tools (Manajer Paket) adalah kumpulan perangkat lunak untuk mengotomatisasi proses instalasi, upgrade, mengkonfigurasi dan menghapus paket perangkat lunak untuk sistem operasi komputer secara konsisten. Ini biasanya memelihara sebuah database dependensi software dan informasi versi untuk mencegah ketidakcocokan perangkat lunak dan prasyarat yang hilang.

Paket adalah distribusi perangkat lunak, aplikasi dan data. Paket juga mengandung metadata, seperti nama perangkat lunak, deskripsi tujuan, nomor versi, vendor, checksum dan daftar dependensi yang diperlukan untuk perangkat lunak untuk berjalan dengan baik. Setelah instalasi, metadata disimpan dalam database paket lokal (Hughes, 2008).

2.2.5 Evolutionary Prototyping

Evolutionary Prototyping Method merupakan salah satu dari beberapa metode yang digunakan dalam pengembangan software. Ciri khusus dari metode ini adalah melibatkan klien dalam membangun perangkat lunak (Carter Dkk, 2001).

Kelebihan Evolutionary Prototyping :

- Klien dapat melihat persyaratan sistem karena mereka ikut membicarakannya,
- Pengembang belajar dari klien,
- Produk akhir akan lebih akurat,
- Mampu menampung permintaan yang tak terduga,
- Memungkinkan fleksibilitas desain dan pengembangan,
- Proses kemajuan pengembangan dapat dipantau dengan baik

Kekurangan Evolutionary Prototyping :

- Membutuhkan keterlibatan klien lebih dekat,
- Perencanaan waktu akan terganggu,
- Proses dapat terus selamanya.

BAB 3

METODE PENGEMBANGAN SISTEM

3.1 Pengumpulan Data

Metode pengumpulan data yang digunakan dalam penelitian tugas akhir ini, yaitu:

1. Studi Kepustakaan

Pengumpulan data dengan cara mempelajari referensi-referensi dan teori-teori yang terkait dengan pengembangan *Software Center* dipelbagai platform maupun sistem operasi lain.

2. Wawancara

Wawancara dilakukan kepada pihak yang memiliki kaitan langsung dengan masalah yang diteliti, dalam hal ini adalah stakeholder atau klien yang menjadi sumber utama dalam mendapatkan informasi, data dan mencari kebutuhan sistem yang akan dibangun dalam penelitian ini. Stakeholder tersebut adalah Bapak Ahmad Haris selaku Manajer Rilis Linux BlankOn dan Mohammad Anwari selaku Manajer Proyek Linux BlankOn.

3.2 Kebutuhan Sistem

Perangkat keras yang dibutuhkan untuk mendukung pengembangan aplikasi *Software Center* dengan spesifikasi sebagai berikut :

1. Prosesor : CPU Intel Core 2 Duo Processor (2CPUs), ~2.4GHz.
2. Memory : 4GB.
3. Display : VGA NVIDIA GeForce 320M 256 MB
4. Hardisk : 128 GB

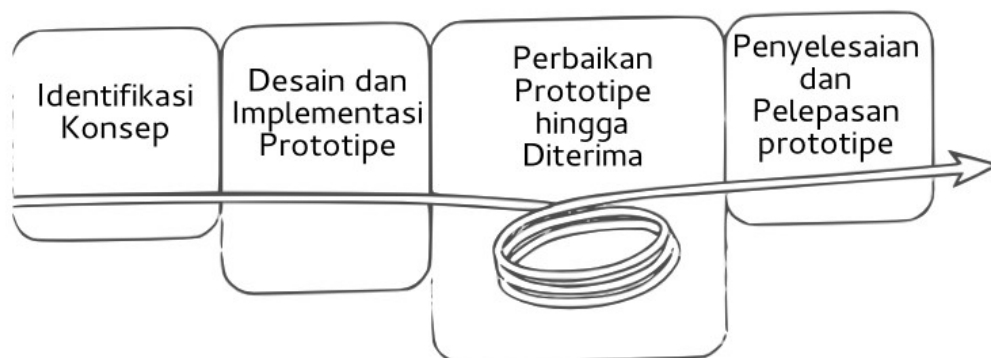
Sedangkan perangkat lunak yang digunakan :

1. Sistem Operasi : Linux BlankOn Suroboyo (develop Branch)
2. Sketsa Editor : Evolus Pencil 2.0.5

3. Text Editor : Sublime Text 3 Stable Build 3059
4. Back-end Programing : Vala 0.20 dan Vala Compiler 0.2037
5. Front-end Programing : HTML5
6. Rendering HTML5 : WebKitGtk Stable 2.0
7. Subversioning : Git (Version Control System)
8. Web Browser : Google Chrome Ver 27.0.1453.110

3.3 Metodologi Pengembangan

Metode pengembangan sistem yang digunakan dalam penelitian ini adalah dengan metode *Evolutionary Prototyping*. Metode evolutionary prototyping mempunyai tahapan-tahapan seperti pada Gambar 3.1.



Gambar 3.1: *Evolutionary Prototyping*

3.4 Metodologi Pengujian Sistem

Model uji validasi dan uji verifikasi yang digunakan adalah pengujian *Black Box*. Pengujian *Black Box* adalah pengujian yang dilakukan dengan mengabaikan bagaimana sistem berjalan. Pengujian ini fokus kepada output yang dikeluarkan tanpa peduli bagaimana input itu diproses pada sistem hingga

mengeluarkan output.

3.4.1 *Alpha Testing*

Uji validasi *Alpha Testing* terfokus pada fungsionalitas sistem. Pengujian mulai dilakukan ketika proses pembuatan sedang berlangsung. Berikut akan dijelaskan rencana pengujian dalam Tabel 3.1.

Tabel 3.1: Rancangan Item Uji Alpha Testing

Item Uji	Detail Uji
Memasang Aplikasi	Aplikasi berhasil dipasang
Menghapus Aplikasi	Aplikasi berhasil dihapus
Mengubah Sumber <i>Repository</i>	Sumber <i>repository</i> berhasil diubah
Mencari Aplikasi berdasarkan kategori	Memilik list aplikasi berdasarkan kategori
Mencari Aplikasi berdasarkan kata kunci	Memasukkan key pencarian di kolom pencarian

3.4.2 *Beta Testing*

Dalam perencanaan pengujian *Beta Testing* ini, akan ditetapkan beberapa parameter untuk mengetahui sejauh mana sistem dapat diterima oleh para pengguna. Adapun parameter-parameter tersebut adalah:

1. Tingkat efektifitas *Software Center* sebelum dan sesudah diimplementasikan di Linux BlankOn,
2. Kemudahan pengguna dalam memahami dan menggunakan sistem yang dibangun.

BAB 4

ANALISIS DAN DESAIN SISTEM

4.1 Analisis Sistem

4.1.1 Analisis Masalah

Analisis masalah dilakukan untuk memperoleh definisi permasalahan dan penggambaran yang tepat dari apa yang akan dilakukan oleh aplikasi, yang bertujuan untuk mengetahui seluk beluk aplikasi yang akan diteliti.

4.1.1.1 Analisis Aplikasi Sejenis

Analisis *Software Center* sejenis dilakukan sebagai pembanding dalam merancang *Software Center* dengan HTML5. *Software Center* yang dibahas dan menjadi acuan disajikan dalam Tabel 4.1:

Tabel 4.1: Perbandingan Aplikasi Software Center Sejenis

Item Banding	Kemudahan Akses	Update	Kategorisasi
Ubuntu Software Center	Ubuntu Software Center dapat diakses melalui desktop sistem operasi ubuntu bisa juga diakses melalui website https://apps.ubuntu.com/	Untuk mengupdate suatu aplikasi sistem tidak melakukan download aplikasi secara keseluruhan tetapi hanya melengkapi paket apa yang belum ada pada aplikasi versi sebelumnya.	Aplikasi dibagi menjadi beberapa kategori yaitu Provided by Ubuntu, Canonical Partners, dan For Purchase.
App Store	Dalam mengakses App Store diperlukan gadget atau setidaknya sebuah PC dengan iTunes yang terinstall untuk	Untuk mengupdate suatu aplikasi, pengguna harus mendownload keseluruhan aplikasi tersebut.	Memiliki pembagian aplikasi yang rapi dan terorganisir. Kategori terbagi atas aplikasi yang terbaru, aplikasi yang

	download aplikasi		terpopuler, aplikasi berbayar, dan aplikasi gratis.
Google Play	Google Play mudah untuk diakses karena dapat diakses tanpa memerlukan gadget dan bisa diakses melalui web browser.	Sama halnya dengan app store, untuk mengupdate suatu aplikasi, pengguna harus mendownload keseluruhan aplikasi tersebut.	Pembagian aplikasi pada google play dibagi menjadi 3 yaitu apps, games, dan books. Pembagian kategori apps dan games berdasarkan Top Paid, Top Free, Top Grossing, Top New Paid, dan Top New Free.
WarSi BlankOn	Tampilan berbasis HTML5 dan interaktif.	Untuk mengupdate pengguna hanya cukup menekan tombol update.	Ada 4 kateogori utama; Internet, Aksesoris, Office, Audio. Semua aplikasi gratis.

4.1.2 Analisa Dan Kebutuhan Fungsional

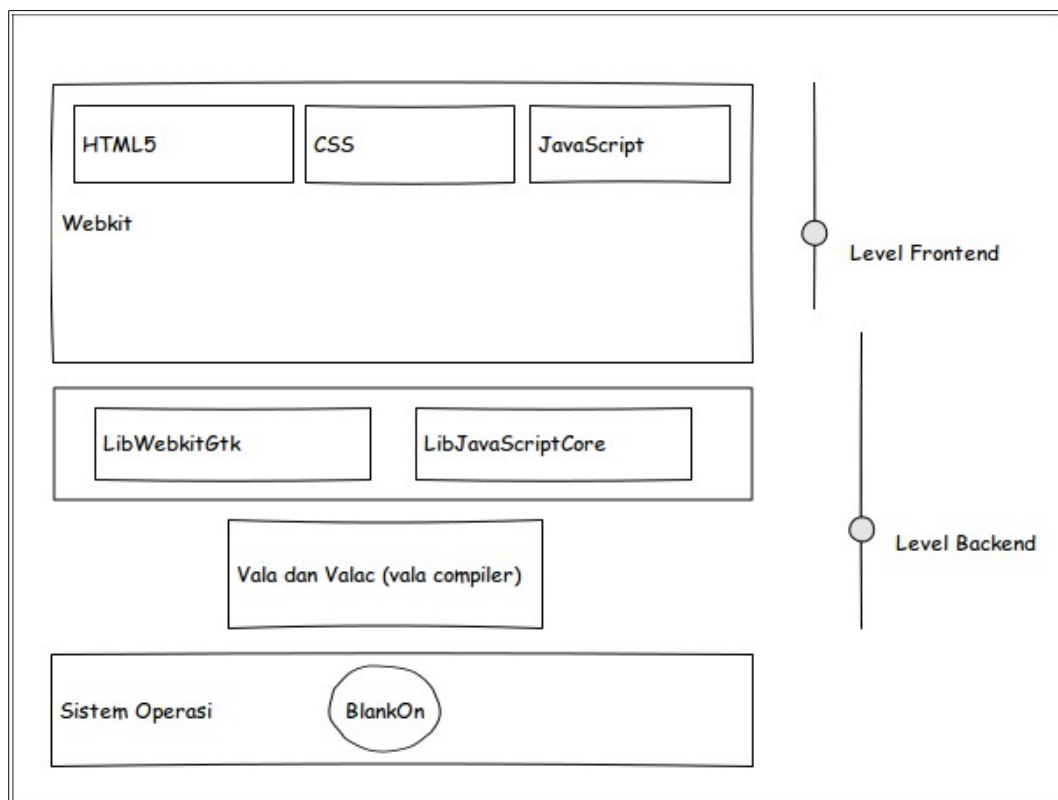
Hasil dari wawancara dengan stakeholder didapatkan beberapa analisis utama dalam pengembangan sistem ini yaitu mengenai pengguna dan fitur apa saja yang akan dikembangkan. Berikut merupakan pembagian kategori pengguna dalam sistem ini:

1. Pengguna BlankOn, merupakan pengguna utama. Pengguna ini dapat melakukan apapun pada *Software Center* seperti melihat software terbaru, melihat software yang terpasang, memasang software baru, menghapus software, mengubah sumber *repository*.
2. Pemaket, merupakan pengguna secara tidak langsung, bertugas merawat konten *Software Center*, sehingga interaksi langsung tidak dilakukan.

4.2 Perancangan Sistem

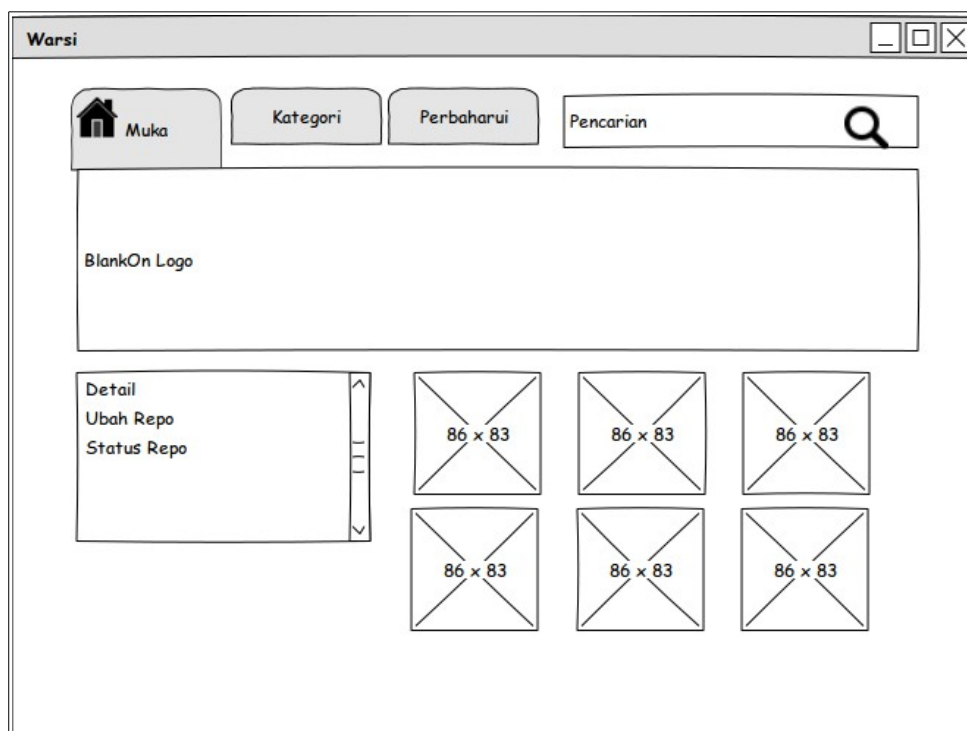
Hasil dari analisis kebutuhan fungsional sebelumnya kemudian dirancang dengan menggunakan UML (Unified Modelling Language).

4.2.1 Desain Sistem

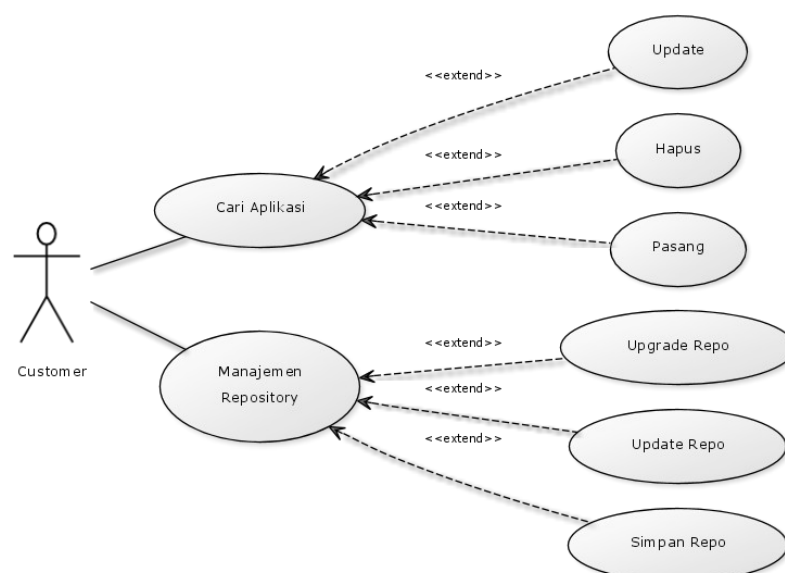


Gambar 4.1: Desain Sistem

4.2.2 Rancangan Antarmuka



Gambar 4.2: Rancangan Antarmuka Halaman Depan



Gambar 4.3: Diagram Use case

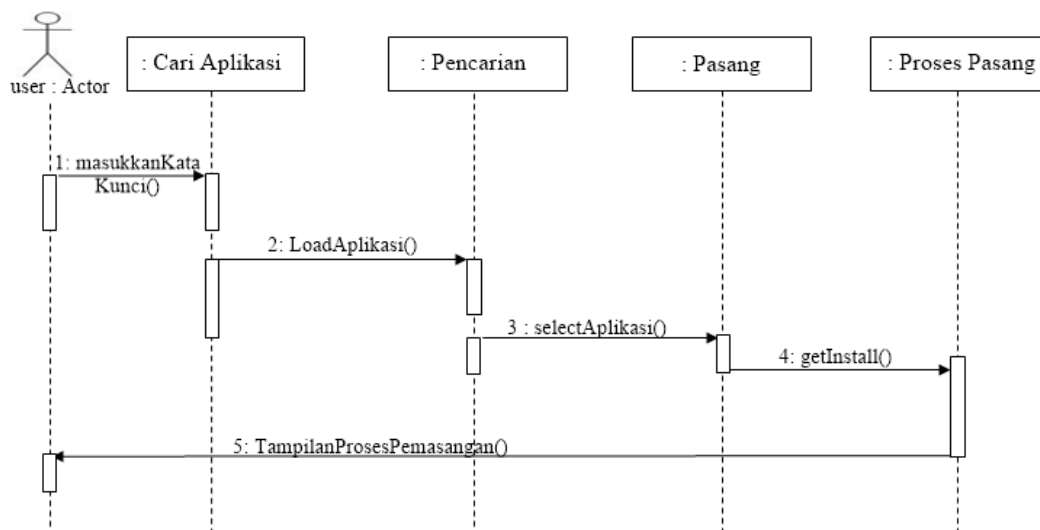
4.2.3 Use Case Diagram

Use case Diagram merupakan konstruksi untuk mendeskripsikan hubungan-hubungan yang terjadi antar aktor dengan aktivitas yang terdapat pada sistem. Sasaran pemodelan *use case* diantaranya adalah mendefinisikan kebutuhan fungsional dan operasional sistem dengan mendefinisikan skenario penggunaan sistem yang akan dibangun. Dari hasil analisis aplikasi yang ada maka *use case diagram* untuk aplikasi *Software Center* dapat di lihat pada Gambar 4.10.

4.2.4 Sequence Diagram

Sequence Diagram menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan pada sistem sebagai respon dari sebuah *event* untuk menghasilkan *output* tertentu.

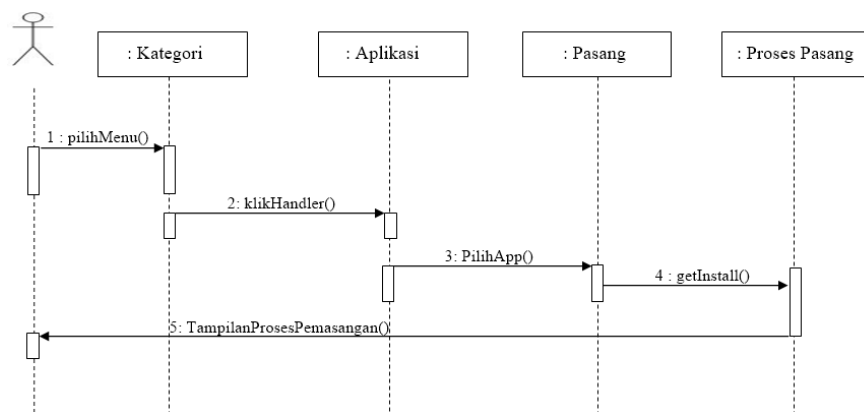
4.2.4.1 Sequence Diagram Cari Aplikasi Berdasarkan Nama



Gambar 4.4: Sequence Diagram Cari Aplikasi

Gambar 4.14 merupakan *Sequence Diagram* cari aplikasi. Diagram ini menangani proses untuk menampilkan halaman pencarian.

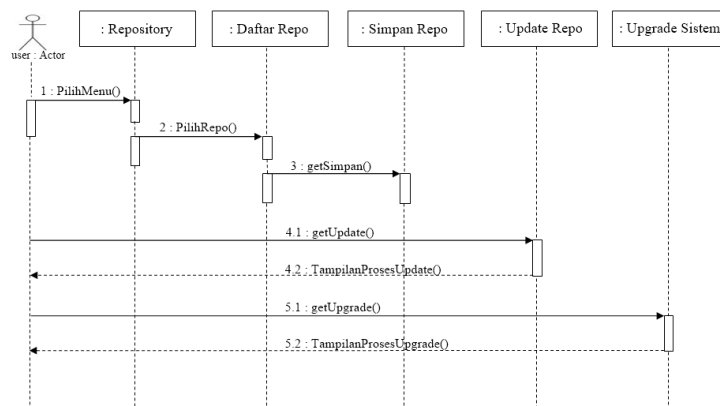
4.2.4.2 *Sequence Diagram* Cari Aplikasi Berdasarkan Kategori



Gambar 4.5: *Sequence Diagram* Pilih Kategori

Sequence diagram pilih aplikasi berdasarkan kategori dapat dilihat pada Gambar 4.15. Diagram ini menangani proses untuk menampilkan halaman pilihan kategori aplikasi.

4.2.4.3 *Sequence Diagram* Manajemen Repository



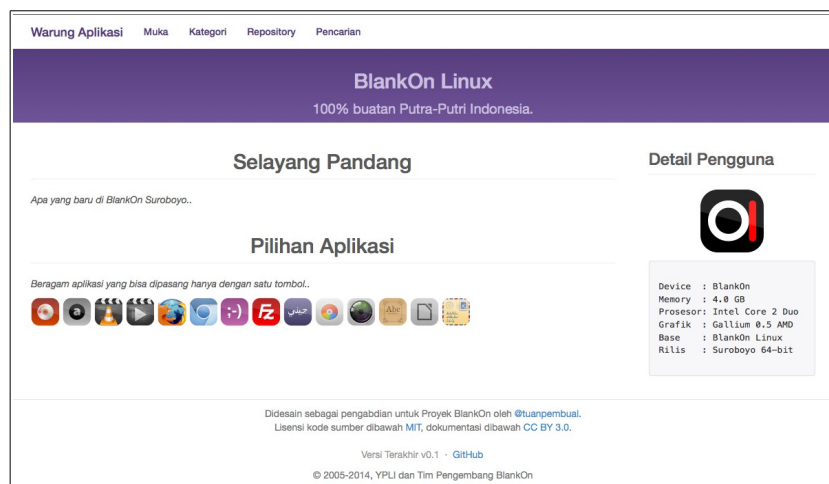
Gambar 4.6: *Sequence Diagram* Manajemen Repository

BAB 5

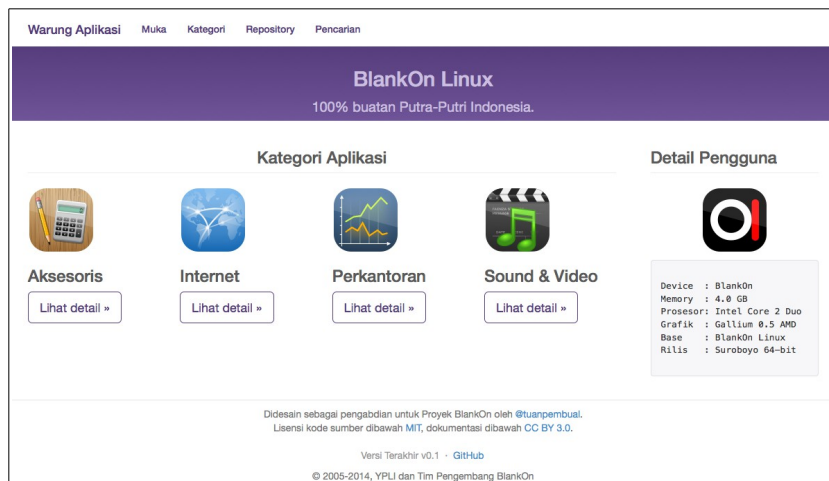
IMPLEMENTASI DAN PENGUJIAN SISTEM

5.1 Implementasi

5.1.1 Implementasi Antarmuka



Gambar 5.1: Implementasi Halaman Awal



Gambar 5.2: Implementasi Halaman Kategori

5.2 Pengujian Sistem

5.2.1 Pengujian Alpha

Alpha Testing meliputi bagaimana pengujian dilakukan pada sistem oleh pakar dengan item-item tertentu. Pengujian dilakukan sebelum aplikasi layak diuji oleh pengguna umum. Pengujian alpha dilakukan pada lingkungan pengembang software untuk memastikan program yang dibuat dapat berjalan dengan baik. Rencana pengujian yang akan dilakukan dapat dilihat pada tabel 5.2.

Tabel 5.1: Rencana Pengujian

Masukan	Hasil yang diharapkan	Pengamatan	Hasil Uji
Menekan tombol menu muka	Menampilkan halaman muka	Dapat menampilkan halaman muka	<input type="checkbox"/> Berhasil <input type="checkbox"/> Tidak berhasil
Menekan tombol menu kategori	Menampilkan halaman kategori aplikasi	Dapat menampilkan halaman kategori	<input type="checkbox"/> Berhasil <input type="checkbox"/> Tidak berhasil
Memasukkan kata kunci dikolom pencari	Menampilkan halaman hasil pencarian	Dapat menampilkan halaman hasil pencarian	<input type="checkbox"/> Berhasil <input type="checkbox"/> Tidak berhasil
Menekan tombol menu kategori Aksesoris	Menampilkan halaman aplikasi dengan kategori Aksesoris	Dapat menampilkan halaman kategori Aksesoris	<input type="checkbox"/> Berhasil <input type="checkbox"/> Tidak berhasil
Menekan tombol menu kategori Internet	Menampilkan halaman aplikasi dengan kategori Internet	Dapat menampilkan halaman kategori Internet	<input type="checkbox"/> Berhasil <input type="checkbox"/> Tidak berhasil
Menekan tombol menu kategory Audio Video	Menampilkan halaman aplikasi dengan kategori Audio Video	Dapat menampilkan halaman kategori Audio Video	<input type="checkbox"/> Berhasil <input type="checkbox"/> Tidak berhasil
Menekan tombol	Menampilkan	Dapat	<input type="checkbox"/> Berhasil

menu kategori Office	halaman aplikasi dengan kategori Office	menampilkan halaman kategori Office	[] Tidak berhasil
Menekan ikon aplikasi	Menampilkan keterangan aplikasi	Dapat menampilkan halaman aplikasi terpilih.	[] Berhasil [] Tidak berhasil
Menekan ikon pasang/hapus aplikasi	Menampilkan halaman persetujuan pemasangan/penghapusan aplikasi	Dapat memasang/menghapus aplikasi	[] Berhasil [] Tidak berhasil
Menekan tombol menu perbaharui	Menampilkan halaman menu perbaharui	Dapat menampilkan halaman perbaharui	[] Berhasil [] Tidak berhasil
Menekan tombol ubah <i>repository</i>	Menampilkan halaman pengubah <i>repository</i>	Dapat mengubah repostori dan menyimpannya	[] Berhasil [] Tidak berhasil

5.2.2 Pengujian Beta

Pengujian beta dilakukan pada lingkungan pengguna aplikasi. Pengujian beta merupakan pengujian yang dilakukan secara objektif dimana diuji secara langsung ke lapangan, dengan menggunakan kuesioner mengenai tanggapan pengguna terhadap aplikasi yang telah dibuat.

Tabel 5.2: Pengujian Usabilitas Aplikasi

No	Pertanyaan	Penilaian				
		SS	S	N	TS	STS
1	Saya memahami cara menggunakan aplikasi <i>software center</i>					
2	Saya merasa mudah memasang dan menghapus aplikasi di Linux BlankOn setelah menggunakan <i>software center</i>					
3	Saya setuju <i>software center</i> ini memiliki					

	tampilan yang menarik					
4	Saya berkeinginan menggunakan <i>software center</i> ini kembali					

Keterangan

SS : Sangat Setuju

TS : Tidak Setuju

S : Setuju

STS : Sangat Tidak Setuju

N : Netral

Tabel 5.3: Pengujian Fungsionalitas Aplikasi

No.	Pertanyaan	Penilaian	
		Ya	Tidak
1	Software Center dapat menampilkan halaman muka		
2	Software Center dapat mencari aplikasi berdasarkan kata kunci		
3	Software Center dapat menampilkan halaman kategori		
4	Software Center dapat menampilkan halaman kategori aksesoris		
5	Software Center dapat menampilkan halaman kategori office		
6	Software Center dapat menampilkan halaman kategori Internet		
7	Software Center dapat menampilkan halaman kategori audio video		
8	Software Center dapat menampilkan halaman manajemen <i>repository</i>		
9	Software Center dapat memasang aplikasi terpilih		
10	Software Center dapat menghapus aplikasi terpilih		
11	Software Center dapat mengubah sumber <i>repository</i>		

BAB 6

HASIL DAN PEMBAHASAN

6.1 Proses Pengembangan *Software Center*

Pengembangan aplikasi *Software Center* ini menggunakan metode *Evolutionary Prototyping*. Dengan metode ini, pengembangan berawal dari desain sederhana (mockup), kemudian desain tadi dipresentasikan kepada klien untuk diminta saran dan koreksi sebelum dibuat rancang dalam bentuk kode pemograman.

6.1.1 Analisis

Pada tahap awal pengembangan, informasi didapat dengan melakukan wawancara dengan klien (stakeholder) dalam hal ini bapak Ahmad Harus selaku manajer rilis BlankOn Suroboyo dan bapak Muhammad Anwari selaku manajer Proyek BlankOn. Hasil wawancara adalah penulis mengetahui fitur – fitur apa saja yang akan dikerjakan pada *Software Center*. Kemudian bagaimana pengembangan *Software Center* sejalan dengan pengembangan desktop BlankOn kedepan.

Setelah mendapatkan fitur-fitur dari aplikasi, pengembang melakukan analisis lebih lanjut yaitu merancang mockup antarmuka aplikasi sebagai modal awal pengembangan. Mockup ini kemudian dipresentasikan kepada stakeholder untuk mendapatkan saran dan koreksi.

6.1.2 Implementasi

Dari hasil mockup yang telah dikoreksi, pengembang melanjutkan tahap

implementasi. Pada tahap ini dilakukan penerjemahan kode menggunakan bahasa pemrograman HTML5, CSS dan Javascript. Penulis memilih menggunakan platform ini berdasarkan petunjuk Manajer Proyek BlankOn agar sesuai dengan pengembangan desktop BlankOn. Setelah pengkodean tampilan (*front-end*) selesai, proses selanjutnya adalah mengerjakan pemrograman di balik layar (*back-end*). Pengembang memilih menggunakan bahasa Vala karena bahasa ini mampu menjembatani HTML5 dengan sistem operasi BlankOn.

Antarmuka didesain dengan bantuan bootstrap css agar tampilan menarik dan mudah dimodifikasi. Bootstrap mendukung penggunaan elemen yang dinamis sesuai dengan aturan HTML5. Interaksi dinamis sangat dibutuhkan agar nilai user experience (UX) semakin baik. Javascript dan JQuery mampu menjawab tantangan ini.

Penulis menggunakan AngularJS sebagai framework javascript karena kemudahan teknologi dan hasil user experience (UX) yang baik dari framework ini. Namun dalam perjalanan pengembangan ini penulis menemukan hambatan yang cukup besar. AngularJS membutuhkan akses file secara langsung dan template halaman html yang sangat dinamis. Sayangnya proses ini terbentur dengan aturan webkit yaitu pembatasan akses langsung terhadap file (MDN, 2014). Masalah ini biasa dikenal sebagai *HTTP access control* atau *Cross-Origin Resource Sharing (CORS)*. Akibatnya akses langsung terhadap file html dan json terganggu.

Penulis memutuskan tetap menggunakan angularJS dengan melakukan beberapa perubahan, mencari alternatif untuk memanipulasi aturan *CORS*. Jawabannya dengan memasukkan file yang akan diakses langsung ke dalam halaman sebelumnya. Misal saat mengakses halaman detail aplikasi, halaman ini penulis masukkan ke dalam file index.html sebagai script kategori template. File index.html bisa dilihat pada bagian lampiran.

File json sebagai objek yang akan diakses penulis masukkan ke dalam file

jasvasript secara langsung agar tetap bisa diakses oleh angularJS. Contoh file yang berisi data json dapat dilihat pada lampiran AppJson.js.

Setelah desain antarmuka dapat berjalan baik, pengembangan menuju bagian sistem (*back-end*). Pengerjaan dimulai dari mencari cara agar Vala mampu berkomunikasi dengan halaman html. Komunikasi ini berupa saling kirim pengembalian setiap aksi yang dilakukan masing-masing bagian. Penulis menemukan kendala lain akibat kurangnya publikasi program kombinasi Vala dan webkit.

Satu persatu kendala terselesaikan kemudian dipresentasikan kepada pengembang BlankOn untuk dimintai koreksi kemudian dilakukan perbaikan sesuai saran yang diberikan.

Koreksi dan perbaikan ini mencakup desain antarmuka maupun proses *back-end* yang dilakukan demi mencapai aplikasi yang lebih baik. Perbaikan antarmuka pertama adalah anjuran menggunakan bootstarp agar sesuai dengan pengembangan destop BlankOn. Perbaikan antarmuka kedua adalah migrasi bootstrap agar menggunakan versi 3(tiga). Perbaikan ketiga adalah agar mampu menampung perubahan halaman yang lebih dinamis dengan memanfaatkan fitur template dari angularJS.

Perbaikan dari sisi *back-end* juga tidak luput. Perbaikan pertama adalah saran untuk menggunakan pustaka API Vala yang lebih terstruktur sesuai pustaka yang telah disediakan oleh Vala. Perbaikan kedua adalah penggunaan JSCore yang mengacu pada standar pengkodean sesuai dengan dokumentasi pengembangan aplikasi Gnome. Perbaikan ketiga adalah perbaikan bentuk program angularJS yaitu app.js dan controller.js dibuat terpisah sesuai manual pemograman angularJS. Perbaikan keempat meski belum sempurna meliputi komunikasi antara JSCore dengan angular agar lebih dinamis dan mudah dirawat oleh pengembang lainnya.

Setelah tiga kali revisi antarmuka dan empat kali revisi model

pengkodean *back-end* akhirnya *prototype* aplikasi *software center* dinyatakan layak digunakan oleh klien (dalam hal ini manajer rilis dan manajer proyek BlankOn) sehingga status pengembangan selesai. Dilanjutkan menuju state selanjutnya yaitu proses implementasi secara langsung pada BlankOn untuk dilakukan pengujian selanjutnya.

6.2 Hasil Pengujian Sistem

6.2.1 Hasil Pengujian Alpha Testing

Pengujian *Alpha testing* dilakukan dengan mengajukan kuesione berisi 5 poin penting mengenai sistem kepada responden untuk disini, tentunya setelah sistem didemokan di depan para responden. Daftar penguji ahli dapat dilihat pada Tabel 6.1.

Tabel 6.1. Daftar Penguji Ahli

No	Nama	Pekerjaan
1	Ahmad Haris	Manajer Rilis BlankOn
2	M. Anwari	Manajer Proyek BlankOn
3	Mahyudin Idam Ramli	Pemaket BlankOn
4	Kukuh Syafaat	Kesenian BlankOn

Adapun lembar kuesioner dan hasil dari pengujian *Alpha testing* terhadap sistem dapat dilihat pada Tabel 6.2.

Tabel 6.2. Hasil Pengujian Alpha Testing

KASUS DAN HASIL UJI				
Parameter Uji	Yang diharapkan	Pengamatan	Ya	Tidak
Menekan tombol menu muka	Menampilkan halaman muka	Dapat menampilkan halaman muka	4	0
Menekan tombol menu kategori	Menampilkan halaman kategori aplikasi	Dapat menampilkan halaman kategori	4	0
Memasukkan kata kunci dikolom pencari	Menampilkan halaman hasil pencarian	Dapat menampilkan halaman hasil pencarian	4	0
Menekan tombol menu kategori Aksesoris	Menampilkan halaman aplikasi dengan kategori Aksesoris	Dapat menampilkan halaman kategori Aksesoris	4	0
Menekan tombol menu kategori Internet	Menampilkan halaman aplikasi dengan kategori Internet	Dapat menampilkan halaman kategori Internet	4	0
Menekan tombol menu kategory Audio Video	Menampilkan halaman aplikasi dengan kategori Audio Video	Dapat menampilkan halaman kategori Audio Video	4	0
Menekan tombol menu kategori Office	Menampilkan halaman aplikasi dengan kategori Office	Dapat menampilkan halaman kategori Office	4	0
Menekan ikon apikasi	Menampikan keterangan aplikasi	Dapat menampilkan halaman aplikasi terpilih.	4	0
Menekan ikon pasang/hapus aplikasi	Menampilkan halaman persetujuan pemasangan/penghapusan aplikasi	Dapat memasang/menghapus aplikasi	4	0
Menekan tombol menu perbaharui	Menampilkan halaman menu perbaharui	Dapat menampilkan halaman perbaharui	4	0
Menekan tombol ubah <i>repository</i>	Menampilan halaman pengubah <i>repository</i>	Dapat mengubah repostori dan menyimpannya	4	0
Jumlah			44	0
Keterangan : B = Berhasil TB= Tidak Berhasil				

6.2.2 Hasil Pengujian Beta Testing

Pengujian *Beta testing* dilakukan dengan mengajukan kuesioner berisi 5 poin penting mengenai sistem kepada responden untuk disini, tentunya setelah sistem didemokan di depan para responden. Daftar penguji *Beta testing* dapat dilihat pada Tabel 6.3.

Tabel 6.3. Daftar Penguji Beta

No	Nama	Pekerjaan
1	Muhammad Khayat	Staff IT Infest Yogyakarta
2	Muhammad Amirudin Latief	Heldesk Jogja Goes Open Source
3	Yudi Setiyadi	Blogger Pengguna Linux
4	Cindy Prastica Adhi	Mahasiswa STIE YKPN Yogyakarta
5	Jauharoh Pratami	Programer Web
6	Wahyu Bimo S	IT Instruktur, Sys Admin
7	Dedy Hariyadi	Mahasiswa Magister T. Informatika UII
8	Jauhari Ramadhan	Siswa Kelas X SMK Muh 1 Yogyakarta
9	Rizqa Gardha M	Guru TIK SMA Muh 3 Yogyakarta
10	Aziz Ardiansyah W.	Java Developer

Adapun lembar kuesioner dan hasil dari pengujian *Beta testing* terhadap sistem dapat dilihat pada Tabel 6.4. dan Tabel 6.5

Tabel 6.4: Pengujian Usabilitas Aplikasi

No	Pertanyaan	Penilaian				
		SS	S	N	TS	STS
1	Saya memahami cara menggunakan aplikasi <i>software center</i>	10	0	0	0	0
2	Saya merasa mudah memasang dan menghapus aplikasi di Linux BlankOn setelah menggunakan <i>software center</i>	8	2	0	0	0

3	Saya setuju <i>software center</i> ini memiliki tampilan yang menarik	8	2	0	0	0
4	Saya berkeinginan menggunakan <i>software center</i> ini kembali	6	4	0	0	0
Jumlah		32	8	0	0	0

Tabel 6.5: Pengujian Fungsionalitas Aplikasi

No.	Pertanyaan	Penilaian	
		Ya	Tidak
1	Software Center dapat menampilkan halaman muka	10	0
2	Software Center dapat mencari aplikasi berdasarkan kata kunci	10	0
3	Software Center dapat menampilkan halaman kategori	10	0
4	Software Center dapat menampilkan halaman kategori aksesoris	10	0
5	Software Center dapat menampilkan halaman kategori office	10	0
6	Software Center dapat menampilkan halaman kategori Internet	10	0
7	Software Center dapat menampilkan halaman kategori audio video	10	0
8	Software Center dapat menampilkan halaman perbaharui	10	0
9	Software Center dapat memasang aplikasi terpilih	10	0
10	Software Center dapat menghapus aplikasi terpilih	10	0
11	Software Center dapat mengubah sumber <i>repository</i>	10	0
Jumlah		110	0

6.2.3 Perhitungan Hasil Beta Testing

Presentase hasil pengujian *usability* sistem :

- Sangat setuju : $(32/40) * 100 = 80\%$
- Setuju : $(8/40) * 100 = 20\%$

- Netral : $(0/40) * 100 = 0\%$
- Tidak setuju : $(0/40) * 100 = 0\%$
- Sangat tidak setuju : $(0/40) * 100 = 0\%$

Presentase hasil pengujian fungsionalitas sistem :

- Menjawab YA : $(110 / 110) * 100 \% = 100\%$
- Menjawab TIDAK : $(0/ 110) * 100 \% = 0\%$

Berdasarkan hasil pengujian yang melibatkan 10 responden atau pengguna di atas, dapat diketahui bahwa sebagian besar pengguna menyatakan penilaian yang baik terhadap aplikasi *software center* yang telah dibuat, maka didapat hasil pengujian yang menunjukkan bawa 100% pengguna menyatakan fungsionalitas sistem telah berjalan dengan baik dan 0% responden menyatakan fungsional sistem tidak berjalan dengan baik.

Berdasarkan pengujian dari segi *usability software center*, diperoleh kesimpulan bahwa sebagian besar responden puas dengan aplikasi yang dibuat. Data hasil pengujian *usability* bahwa responden sangat setuju sebanyak 80%, setuju sebanyak 20%, netral 0%, yang menyatakan tidak setuju sebanyak 0% dan yang menyatakan sangat tidak setuju sebanyak 0%.

Berdasarkan hasil pengujian tersebut, dapat disimpulkan bahwa *software center* yang telah dibuat ini layak untuk digunakan. Akan tetapi perlu adanya pengembangan aplikasi yang lebih lanjut untuk mendapatkan aplikasi yang optimal.

BAB 7

KESIMPULAN

7.1 Kesimpulan

Berdasarkan kegiatan yang telah dilakukan selama pengembangan *Software Center* menggunakan metode *Evolutionary Prototyping* berbasis HTML5, maka dapat diambil kesimpulan sebagai berikut :

1. Analisis dan pengembangan *Software Center* yang dapat membantu pengguna untuk manajemen aplikasi di Linux BlankOn sudah berhasil dilakukan,
2. Dalam penelitian ini bahasa pemrograman HTML5 mampu digunakan untuk pengembangan aplikasi desktop,
3. Berdasarkan hasil pengujian menunjukkan bahwa *Software Center* dapat berjalan di Linux BlankOn.

7.2 Saran

Software Center ini tentunya tidak terlepas dari kekurangan dan kelemahan. Oleh karena itu, untuk kebaikan pengembangan sistem selanjutnya, maka penulis menyarankan beberapa hal, diantaranya:

1. Perlu adanya penambahan data daftar aplikasi beserta informasi detail lainnya. Sebagai alternatif bisa dimulai oleh tim pemaket aplikasi *repository* untuk mulai melakukan pendataan aplikasi,
2. Perlu adanya metode komunikasi antara JSCore dan HTML5 yang lebih baik agar kode *Software Center* bisa menjadi sangat dinamis,
3. Kendala awal dalam pengembangan ini adalah adanya keterbatasan CORS. Perlu solusi yang lebih baik agar akses file langsung tidak menemukan kendala,
4. Perlu adanya otomatisasi untuk memperbaharui daftar aplikasi beserta versi terbaru tiap aplikasi,
5. Perlu adanya perubahan lebih lanjut agar *Software Center* ini bisa diimplementasikan pada distribusi Linux lainnya,
6. Penambahan status bar progres pemasangan aplikasi agar *Software Center* lebih mudah digunakan.

DAFTAR PUSTAKA

- Anwari, Muhammad. (2013). *GNOME 3 Application Development Beginner*. Birmingham: Packt Publishing Ltd.
- Barth, Adam. (2012). *How WebKit Works*. WebKit.
- Carter, Ryan A., Antón, Annie I., Dagnino, Aldo., Williams, Laurie. (2001). *Evolving Beyond Requirements Creep: A Risk-Based Evolutionary Prototyping Model*. North Carolina State University, .
- Dharwiyanti, S., & Wahono, R. S. (2003). *Pengantar Unified Modeling Language (UML)*. Jakarta: IlmuKomputer.
- Hendarto, Handri. (2007). *GUI: Pengembangan Graphical User Interface (GUI) Untuk Operator Training Simulator (OTS)*. Bandung: Institut Teknologi Bandung.
- Hildebrand, Nikolaus & Bode, Nicolai . (2013). *The Apple App Store - Mechanism of Technology Adoption*. Munchen: Parel To.
- Hughes, Ricard. (2008, Februari 24). *PackageKit: Package Abstraction Framework*. Mei 24, 2013. <http://hughsie.com/public/introduction-to-packagekit.pdf>.
- Ismail, Adhi Kusna & Nur Fajri, Aufar. (2013). *Aplikasi Permainan Casual Game Black Pursuit Berbasis HTML 5*. Jakarta: Universitas Bina Nusantara.
- Kadir, Abdul. (2009). *Mastering Ajax dan PHP*. Yogyakarta: Andi Yogyakarta.
- Kadir, Abdul. (2013). *From Zero To A Pro HTML5*. Yogyakarta: Andi Publisher.
- Klumpp, Matthias. (2012, Agustus 30). *GSOC Appstream Final Report*. April 17, 2013. <http://blog.tenstral.net/2012/08/gsoc-appstream-final-report.html>.
- Mac-Vicar P., Duncan . (2008). *What you should know about YaST*. Novell, Inc.
- Mozilla Developer Network. (2014, April 18). *HTTP access control (CORS)*. April 20, 2014. https://developer.mozilla.org/en-US/docs/Web/HTTP/Access_control_CORS.
- Paul Thomas, Mathew. (2005, Agustus 29). *Ubuntu Software Center*. April 18, 2013. <http://wiki.ubuntu.com/SoftwareCenter>.
- Pratami, Jauharoh. (2013). *Rancang Bangun Game Hijaiyah Match dengan HTML5*. Yogyakarta: Universitas Islam Negeri Sunan Kalijaga.
- Purnomo, Agus Dkk. (2012). *Panduan BlankOn Rote*. Jakarta: YLPI.
- Raharja, R. Anthon dkk. (2001). *Modul Pelatihan Pengenalan Linux*. Bandung: Telemactics Indonesia.

- Research In Motion. (2010). *BlackBerry App World Storefront Version: 2.0 User Guide*. Research In Motion Limited.
- Salim, Hendri. (2013). *App Store vs Google Play, Ronde 1 Fight!* . Desember, 18 2013. <http://id.gamesinasia.com/app-store-vs-google-play-ronde-1-fight/>.
- Stroik, Brian . (2010). *Why a Mock Up, Because the Owner Expects it Done Right* . Oscar J. Boldt Construction.
- Wahlin, Dan. (2013). *AngularJS in 60 Minutes*. Arizona: Wahlin Consulting.
- Wheeler, David A.. (2000). *Why Open Source Software / Free Software (OSS/FS, FLOSS, or FOSS)? Look at the Numbers!*. California: Self-Publisher.
- Šrain, Jiří. (2006). *YaST Architecture*. :Novel, Inc.