

**SISTEM PENERIMA BERITA MELALUI RSS PADA  
NINTENDO DS**

**SKRIPSI**

**Diajukan untuk memenuhi sebagian prasyarat  
memperoleh gelar Sarjana Teknik**



**Disusun oleh:**

**ROY MARGASA**

**NIM. 0310633087**

**DEPARTEMEN PENDIDIKAN NASIONAL**

**UNIVERSITAS BRAWIJAYA**

**FAKULTAS TEKNIK**

**MALANG**

**2009**

# SISTEM PENERIMA BERITA MELALUI RSS PADA NINTENDO DS

## SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Teknik



Disusun oleh:

**ROY MARGASA**

**NIM. 0310633087**

Disahkan dan disetujui oleh dosen pembimbing :

Pembimbing I

Pembimbing II

Ir. Heru Nurwasito, M.Kom

NIP. 131 879 033

Himawat Aryadita, ST., M.Sc.

NIP. 132 327 625

# SISTEM PENERIMA BERITA MELALUI RSS PADA NINTENDO DS

Disusun oleh :

**ROY MARGASA**

**NIM. 0310633087**

Skripsi ini telah diuji dan dinyatakan lulus pada  
tanggal 3 Agustus 2009

Majelis Penguji :

**Arief Andy S., ST., M.Kom**

**NIP. 132 231 567**

**Suprpto, ST, MT**

**NIP. 132 149 320**

**Ir. Sutrisno**

**NIP. 131 653 479**

Mengetahui :

Ketua Jurusan Teknik Elektro

**Ir. Heru Nurwarsito, M.Kom**

**NIP. 131 879 033**

## KATA PENGANTAR

Puji syukur atas kehadiran Tuhan YME karena dengan berkat rahmat dan karunia serta restu-Nya penyusunan skripsi ini dengan judul “Sistem Aplikasi Penerima Berita Melalui RSS Pada Nintendo DS” dapat diselesaikan. Penulis menyadari bahwa kajian ini tidak akan mencapai titik akhir penyelesaian tanpa bantuan berbagai pihak, karenanya penulis mengucapkan terima kasih kepada:

1. Kedua orang tuaku dan semua saudaraku untuk seluruh doa dan dukungannya yang telah diberikan hingga terselesaikannya skripsi ini.
2. Ir. Heru Nurwasito M.Kom. selaku dosen pembimbing dan Ketua Jurusan Teknik Elektro, Fakultas Teknik, Universitas Brawijaya.
3. Rudy Yuwono, ST, M.Sc. selaku Sekretaris Jurusan Teknik Elektro, Fakultas Teknik, Universitas Brawijaya.
4. Himawat Aryadito, ST., M.Sc. selaku dosen pembimbing pada penyusunan skripsi ini.
5. Bapak dan ibu dosen serta karyawan Jurusan Teknik Elektro, Fakultas Teknik, Universitas Brawijaya.
6. Krisna Shanti, Terima kasih untuk kasih sayang dan dukungannya yang sangat luar biasa.
7. Teman, sahabat, dan saudaraku yang telah bersama menjalani hari dengan suka dan duka. Terima kasih atas dukungan moral maupun ilmu yang telah diberikan hingga akhirnya titik ini bisa tercapai.
8. Serta semua pihak yang telah turut membantu baik secara langsung maupun tidak langsung dalam penyelesaian skripsi ini.

Seorang yang bijak mengerti apa yang telah dilalui dan yang akan dilalui. Tiada yang sempurna di dunia ini, tersadar bahwa skripsi ini sangat jauh dari kesempurnaan. Karenanya, segala kritik dan saran yang sifatnya membangun dari pembaca tentang skripsi ini akan diterima dengan senang hati.

Akhir kata, penulis berharap semoga skripsi ini dapat bermanfaat bagi semua pihak yang membutuhkan.

Malang, Juli 2009

Penyusun



DAFTAR ISI

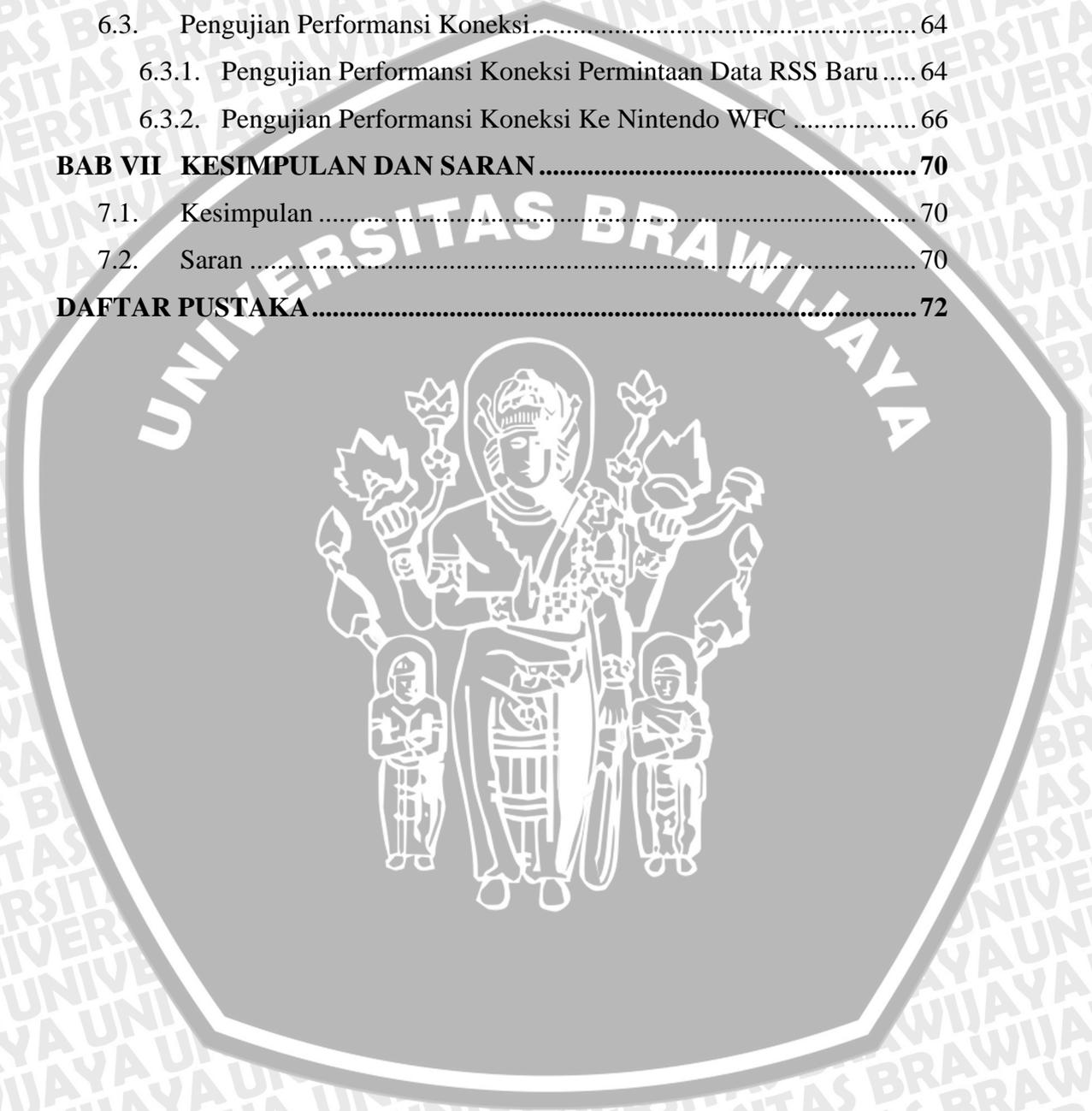
<b>HALAMAN JUDUL .....</b>	<b>i</b>
<b>LEMBAR PERSETUJUAN DOSEN PEMBIMBING .....</b>	<b>ii</b>
<b>LEMBAR PERSETUJUAN DOSEN PENGUJI.....</b>	<b>iii</b>
<b>KATA PENGANTAR.....</b>	<b>iv</b>
<b>DAFTAR ISI.....</b>	<b>vi</b>
<b>DAFTAR TABEL .....</b>	<b>x</b>
<b>DAFTAR GAMBAR.....</b>	<b>xi</b>
<b>DAFTAR ISTILAH .....</b>	<b>xiii</b>
<b>ABSTRAK .....</b>	<b>xiv</b>
<b>BAB I PENDAHULUAN .....</b>	<b>1</b>
1.1. Latar Belakang .....	1
1.2. Rumusan Masalah.....	2
1.3. Batasan Masalah.....	2
1.4. Tujuan .....	3
1.5. Manfaat .....	3
1.6. Sistematika Penulisan .....	3
<b>BAB II TINJAUAN PUSTAKA.....</b>	<b>5</b>
2.1. Teori Dasar.....	5
2.2. XML.....	5
2.3. RSS .....	5
2.4. RSS <i>Syntax</i> .....	6
2.5. RSS <i>Reader</i> .....	7
2.6. Nintendo DS.....	8
2.7. Wi-Fi.....	9
2.8. Nintendo Wi-Fi <i>Connection</i> .....	9
2.9. Nintendo DS <i>Homebrew</i> .....	9
2.10. Nintendo DS <i>Programming</i> .....	10
2.11. Libnds .....	10
2.12. Libfat.....	11
2.13. DLDI.....	11



2.14.	Dswifi.....	11
2.15.	Palib .....	12
2.16.	Bahasa Pemrograman C dan C++ .....	12
2.17.	<i>Data Flow Diagram</i> (DFD) .....	13
2.18.	<i>White Box Testing</i> .....	15
2.18.1.	Uji Coba Basis <i>Path</i> .....	16
2.18.2.	Notasi Diagram Alir.....	17
2.18.3.	<i>Cyclomatic Complexity</i> .....	18
2.19.	<i>Black Box Testing</i> .....	20
<b>BAB III</b>	<b>METODOLOGI PENELITIAN</b> .....	<b>21</b>
3.1.	Studi Literatur .....	21
3.2.	Perancangan Perangkat Lunak.....	21
3.3.	Implementasi Perangkat Lunak.....	22
3.4.	Pengujian dan Analisis Perangkat Lunak .....	22
3.5.	Pengambilan Kesimpulan dan Saran.....	22
<b>BAB IV</b>	<b>ANALISIS KEBUTUHAN DAN PERANCANGAN</b> .....	<b>23</b>
4.1.	Analisa Kebutuhan Perangkat Lunak.....	23
4.1.1.	Analisa Sistem <i>RSS Reader</i> .....	24
4.1.2.	Analisa Sistem <i>RSS</i> Aplikasi Penerima Berita Melalui <i>RSS</i> pada Nintendo DS.....	27
4.2.	Perancangan Sistem .....	28
4.2.1.	Diagram Blok Sistem .....	28
4.2.2.	Diagram Konteks .....	28
4.2.3.	<i>Data Flow Diagram Levelled</i> (DFDL) .....	29
4.2.3.1.	<i>Data Flow Diagram Level 0</i> Sistem Aplikasi Penerima Berita Melalui <i>RSS</i> pada Nintendo DS.....	29
4.2.3.2.	<i>Data Flow Diagram Level 1</i> Pengolahan Data <i>RSS</i> .....	30
4.2.3.3	<i>Data Flow Diagram Level 1</i> Administrasi Data <i>RSS</i> .....	32
4.2.4.	<i>Pseudocode</i> Sistem Aplikasi Penerima Berita Melalui <i>RSS</i> pada Nintendo DS.....	33
4.2.4.1.	<i>Pseudocode</i> Proses Permintaan Halaman [sos11].....	33
4.2.4.2.	<i>Pseudocode</i> Proses Menampilkan Berita [sos12] .....	34

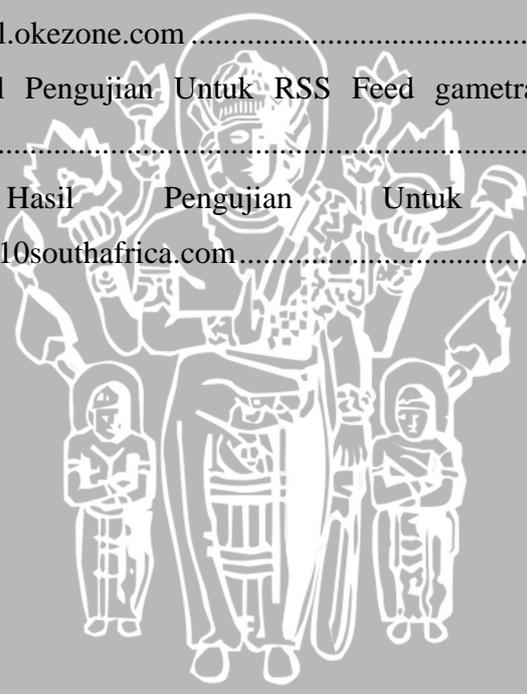
4.2.4.3.	Pseudocode Proses Permintaan Data RSS [sos13] .....	35
4.2.4.4.	Pseudocode Proses Penyimpanan Data RSS Baru [sos21].....	35
4.2.4.5.	Pseudocode Proses Permintaan Data RSS Baru [sos22].....	36
<b>BAB V IMPLEMENTASI PERANGKAT LUNAK.....</b>		<b>38</b>
5.1.	Implementasi Sistem.....	38
5.1.1.	Spesifikasi Perangkat Keras.....	39
5.1.2.	Spesifikasi Perangkat Lunak.....	39
5.1.3.	Spesifikasi Access Point .....	40
5.2.	Implementasi Algoritma .....	40
5.2.1.	Implementasi Kebutuhan Fungsional Inisialisasi .....	40
5.2.2.	Implementasi Kebutuhan Fungsional Permintaan Halaman [sos11].....	42
5.2.3.	Implementasi Kebutuhan Fungsional Permintaan Data RSS [sos13].....	44
5.2.4.	Implementasi Kebutuhan Fungsional Permintaan Data RSS Baru [sos22].....	46
5.2.5.	Implementasi Kebutuhan Fungsional Penyimpanan Data RSS Baru [sos21].....	47
5.2.6.	Implementasi Kebutuhan Fungsional Menampilkan Berita [sos12].....	48
<b>BAB VI PENGUJIAN DAN ANALISIS PERANGKAT LUNAK.....</b>		<b>50</b>
6.1.	Pengujian Unit .....	50
6.1.1.	Pengujian Unit Untuk Proses Permintaan Halaman [sos11].....	51
6.1.2.	Pengujian Unit Untuk Proses Permintaan Data RSS [sos13].....	53
6.1.3.	Pengujian Unit Untuk Proses Menampilkan Berita [sos12] .....	54
6.2.	Pengujian Implementasi Sistem Aplikasi Penerima Berita Melalui RSS pada Nintendo DS .....	55
6.2.1.	Pengujian Implementasi Analisis.....	56
6.2.2.	Pengujian Implementasi Permintaan Halaman [sos11] .....	58
6.2.3.	Pengujian Implementasi Permintaan Data RSS [sos13] .....	59

6.2.4. Pengujian Implementasi Permintaan Data RSS Baru [sos22] ....	60
6.2.5. Pengujian Implementasi Penyimpanan Data RSS Baru [sos21].....	62
6.2.6. Pengujian Implementasi Menampilkan Berita [sos12] .....	63
6.3. Pengujian Performansi Koneksi.....	64
6.3.1. Pengujian Performansi Koneksi Permintaan Data RSS Baru .....	64
6.3.2. Pengujian Performansi Koneksi Ke Nintendo WFC .....	66
<b>BAB VII KESIMPULAN DAN SARAN .....</b>	<b>70</b>
7.1. Kesimpulan .....	70
7.2. Saran .....	70
<b>DAFTAR PUSTAKA .....</b>	<b>72</b>



**DAFTAR TABEL**

Tabel 5.1. Spesifikasi Perangkat Keras Komputer .....	39
Tabel 5.2. Spesifikasi Nintendo DS .....	39
Tabel 5.3. Spesifikasi Perangkat Lunak .....	40
Tabel 5.4. Spesifikasi Access Point .....	40
Tabel 6.1. Test Case Untuk Pengujian Permintaan Halaman .....	52
Tabel 6.2. Test Case Untuk Pengujian Permintaan Data RSS .....	54
Tabel 6.3. Test Case Untuk Pengujian Menampilkan Berita .....	55
Tabel 6.4. Tabel Hasil Pengujian Untuk Koneksi Ke Nintendo WFC.....	66
Tabel 6.5. Tabel Hasil Pengujian Untuk RSS Feed international.okezone.com .....	68
Tabel 6.6. Tabel Hasil Pengujian Untuk RSS Feed gametrailers.com bagian NDS.....	68
Tabel 6.7. Tabel Hasil Pengujian Untuk RSS <i>Feed</i> worldcup2010southafrica.com.....	69



**DAFTAR GAMBAR**

Gambar 2.1.	Penulisan sintaks sebuah dokumen RSS.....	6
Gambar 2.2.	Contoh Kode Javascript Untuk Membaca RSS <i>feed</i> .....	8
Gambar 2.3.	Nintendo DS Lite .....	9
Gambar 2.4.	Struktur Kerangka Umum dari Program yang ada di Dalam C++ .....	13
Gambar 2.5.	Gambar Notasi Diagram Alir .....	16
Gambar 2.6.	Diagram Alir Contoh Diatas .....	17
Gambar 2.7.	Logika Gabungan.....	18
Gambar 2.8.	Diagram Alir Contoh .....	19
Gambar 4.1.	Diagram Pohon Analisis Kebutuhan dan Perancangan .....	23
Gambar 4.2.	Tampilan FeedDemon.....	25
Gambar 4.3.	Tampilan Google <i>Reader</i> .....	25
Gambar 4.4.	Tampilan NewsGator .....	26
Gambar 4.5.	Tampilan Google <i>Reader</i> pada PDA .....	27
Gambar 4.6.	Tampilan Google <i>Reader</i> pada Opera Mini.....	27
Gambar 4.7.	Diagram Blok Sistem Aplikasi Penerima Berita Melalui RSS Pada Nintendo DS.....	28
Gambar 4.8.	Diagram Konteks Sistem Aplikasi Penerima Berita Melalui RSS Pada Nintendo DS (S.o.S. RSS Parser).....	29
Gambar 4.9.	<i>Data Flow Diagram Level 0</i> Sistem Aplikasi Penerima Berita Melalui RSS Pada Nintendo DS .....	30
Gambar 4.10.	<i>Data Flow Diagram Level 1</i> Pengolahan Data RSS .....	31
Gambar 4.11.	<i>Data Flow Diagram Level 1</i> Administrasi Data RSS .....	32
Gambar 5.1.	Diagram Pohon Implementasi Sistem.....	38
Gambar 5.2.	Hasil Implementasi Inisialisasi .....	41
Gambar 5.3.	Hasil Implementasi Algoritma Untuk Permintaan Halaman .....	43
Gambar 5.4.	Hasil Implementasi untuk Permintaan Data RSS .....	45
Gambar 5.5.	Hasil Implementasi untuk Permintaan Data RSS Baru.....	46
Gambar 5.6.	Hasil Implementasi untuk Penyimpanan Data RSS Baru .....	47
Gambar 5.7.	Hasil Implementasi Untuk Menampilkan Berita .....	48

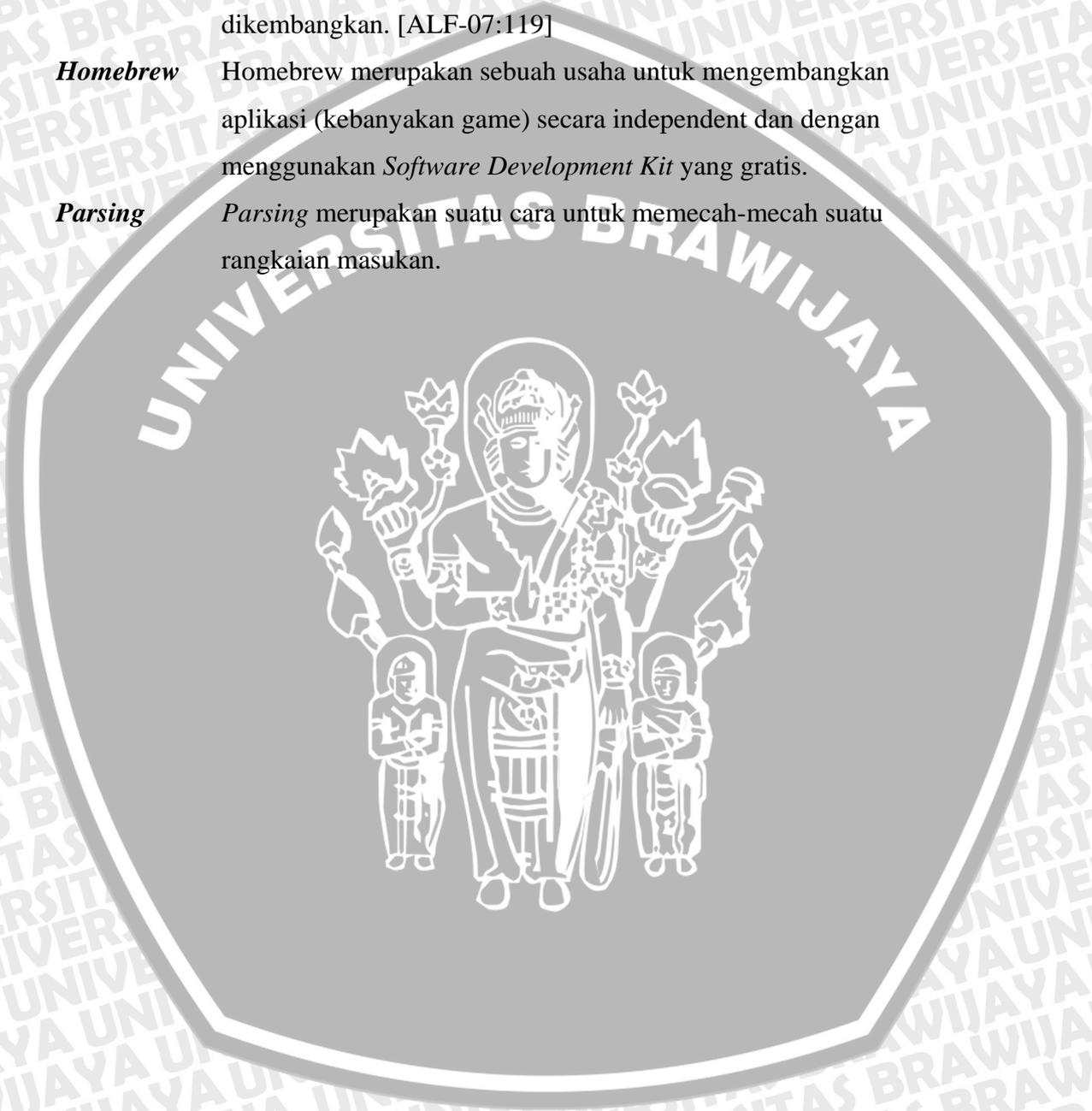


Gambar 6.1.	Diagram Pohon Pengujian dan Analisis Sistem.....	50
Gambar 6.2.	Pemodelan Algoritma Permintaan Halaman ke Dalam <i>Flow Graph</i> .....	51
Gambar 6.3.	Pemodelan Algoritma Permintaan Data RSS ke Dalam <i>Flow Graph</i> .....	53
Gambar 6.4.	Pemodelan Algoritma Menampilkan Berita ke Dalam <i>Flow Graph</i> .....	54
Gambar 6.5.	Tampilan Implementasi Inisialisasi Yang Selesai .....	57
Gambar 6.6.	Tampilan Implementasi Inisialisasi Yang Gagal .....	58
Gambar 6.7.	Tampilan Implementasi Permintaan Halaman.....	59
Gambar 6.8.	Tampilan Implementasi Permintaan Data RSS.....	60
Gambar 6.9.	Tampilan Implementasi Permintaan Data RSS Baru.....	61
Gambar 6.10.	Tampilan Implementasi Penyimpanan Data RSS Baru .....	62
Gambar 6.11.	Tampilan Data RSS Yang Telah Tersimpan.....	63
Gambar 6.12.	Tampilan Implementasi Menampilkan Berita.....	64
Gambar 6.13.	Penghitung Waktu Koneksi ke Nintendo WFC.....	65
Gambar 6.14.	Penghitung Waktu Permintaan Data RSS Baru.....	67



## DAFTAR ISTILAH

- DFD** *Data Flow Diagram*, diagram yang digunakan untuk menggambarkan proses-proses yang terjadi pada sistem yang akan dikembangkan. [ALF-07:119]
- Homebrew** Homebrew merupakan sebuah usaha untuk mengembangkan aplikasi (kebanyakan game) secara independent dan dengan menggunakan *Software Development Kit* yang gratis.
- Parsing** *Parsing* merupakan suatu cara untuk memecah-mecah suatu rangkaian masukan.



## ABSTRAK

**ROY MARGASA. 2009. : Sistem Aplikasi Penerima Berita Melalui RSS Pada Nintendo DS. Skripsi Jurusan Teknik Elektro, Fakultas Teknik, Universitas Brawijaya. Dosen Pembimbing : Ir. Heru Nurwasito, M.kom dan Himawat Aryadita, ST., M.Sc.**

Beberapa tahun terakhir ini tidak dapat dipungkiri bahwa internet telah mengalami perkembangan yang sangat pesat. Beragam informasi tersedia di internet dan dapat diakses secara luas oleh penggunanya. Akan tetapi, dengan banyaknya data baik berupa teks, gambar, maupun video yang ada di internet saat ini menyebabkan terjadinya ledakan informasi. RSS (*Really Simple Syndication*) memungkinkan *web developers* untuk menawarkan isi dari situs mereka dengan cara menampilkan *headlines* dan ringkasan isinya. RSS dibuat untuk menampilkan data yang dipilih. Tanpa RSS, user harus melihat situs-situs tertentu untuk *update* terbaru. Dengan adanya *RSS feed*, user dapat melihat ringkasan isi dari suatu situs dengan menggunakan *RSS aggregator* (sebuah situs atau program yang mengumpulkan dan menyusun *RSS feed*). Karena data RSS kecil dan cepat di-load, maka RSS dapat dengan mudah digunakan pada telepon selular ataupun PDA. [SNC-07] Nintendo DS merupakan sebuah *Portable Gaming System* yang memungkinkan user untuk menjelajahi dunia internet dengan menggunakan fitur Wi-Fi yang dimilikinya. Mengingat teknologi RSS memiliki banyak manfaat maka akan sangat berguna bagi user jika teknologi RSS diintegrasikan pada Nintendo DS. Untuk menampilkan RSS pada Nintendo DS dibutuhkan sebuah aplikasi yang mampu membaca RSS dari sebuah situs lalu kemudian menerjemahkannya ke dalam sistem operasi Nintendo. Sedangkan *browser* untuk Nintendo DS yang tersedia saat ini masih belum mampu menampilkan RSS. Untuk mengatasi permasalahan yang dihadapi maka diperlukan suatu sistem pembaca RSS. Sistem ini diharapkan dapat memudahkan *user* dalam mengakses referensi informasi dari dunia maya. Sistem ini memiliki kemampuan untuk membaca RSS yang tersedia dari situs tertentu (situs yang memiliki layanan RSS) dan kemudian menampilkannya ke dalam *user interface* pada Nintendo DS yang dapat dinikmati oleh *user*.

Perancangan dan pengimplementasian Sistem Aplikasi Penerima Berita Melalui RSS Pada Nintendo DS dilakukan dengan menggunakan bahasa pemrograman C++, dan library PALib.

Pengujian Sistem Aplikasi Penerima Berita Melalui RSS Pada Nintendo DS dilakukan pada setiap aplikasi sistem untuk mengetahui proses yang dilakukan oleh tiap-tiap aplikasi sistem tersebut. Hasil dari pengujian aplikasi sistem dapat diketahui bahwa aplikasi sistem pada Sistem Aplikasi Penerima Berita Pada Nintendo DS dapat melakukan proses sesuai dengan kegunaannya masing-masing. Pengujian performansi yang dilakukan terhadap performansi permintaan data RSS baru memiliki tingkat keberhasilan 100 %, dan untuk performansi koneksi ke Nintendo WFC memiliki tingkat keberhasilan 90%.

Kata Kunci : RSS, C++, PALib, Nintendo DS

## BAB I

### PENDAHULUAN

#### 1.1. Latar Belakang

Beberapa tahun terakhir ini tidak dapat dipungkiri bahwa internet telah mengalami perkembangan yang sangat pesat. Beragam informasi tersedia di internet dan dapat diakses secara luas oleh penggunanya. Akan tetapi, dengan banyaknya data baik berupa teks, gambar, maupun video yang ada di internet saat ini menyebabkan terjadinya ledakan informasi yang membuat user (pengguna) internet seringkali mengalami kesulitan dalam mencari informasi yang benar-benar sedang dicari atau dibutuhkan.

RSS (*Really Simple Syndication*) memungkinkan *web developers* untuk menawarkan isi dari situs mereka dengan cara menampilkan *headlines* dan ringkasan isinya. RSS dibuat untuk menampilkan data yang dipilih. Tanpa RSS, user harus melihat situs-situs tertentu untuk update terbaru. Hal ini akan sangat menyita waktu bagi kebanyakan user. Dengan adanya *RSS feed*, user dapat melihat ringkasan isi dari suatu situs dengan menggunakan *RSS aggregator* (sebuah situs atau program yang mengumpulkan dan menyusun *RSS feed*). Karena data RSS kecil dan cepat di-load, maka RSS dapat dengan mudah digunakan pada telepon selular ataupun PDA.[SNC-07]

Nintendo DS merupakan sebuah *Portable Gaming System* yang memungkinkan user untuk menjelajahi dunia internet dengan menggunakan fitur Wi-Fi yang dimilikinya. Mengingat teknologi RSS memiliki banyak manfaat maka akan sangat berguna bagi user jika teknologi RSS diintegrasikan pada Nintendo DS. Untuk menampilkan RSS pada Nintendo DS dibutuhkan sebuah aplikasi yang mampu membaca RSS dari sebuah situs lalu kemudian menerjemahkannya ke dalam sistem operasi Nintendo DS untuk ditampilkan pada LCD Nintendo DS. Sedangkan *browser* untuk Nintendo DS yang tersedia saat ini masih belum mampu menampilkan RSS.

Untuk mengatasi permasalahan yang dihadapi maka diperlukan suatu sistem pembaca RSS. Sistem ini diharapkan dapat memudahkan user dalam

mengakses referensi informasi dari dunia maya. Sistem ini memiliki kemampuan untuk membaca RSS yang tersedia dari situs tertentu (situs yang memiliki layanan RSS) dan kemudian menampilkannya ke dalam user interface pada Nintendo DS yang dapat dinikmati oleh user.

### 1.2. Rumusan Masalah

Berdasarkan permasalahan yang telah diuraikan pada latar belakang, maka dapat dirumuskan beberapa permasalahan, antara lain:

1. Menghubungkan Nintendo DS dengan jaringan internet melalui sebuah *Access Point*.
2. Membaca RSS dari situs tertentu dan menterjemahkannya ke dalam sistem operasi Nintendo DS.
3. Merancang, mengimplementasi, dan menguji aplikasi untuk menampilkan RSS pada layar *user interface* Nintendo DS.

### 1.3. Batasan Masalah

Dalam perencanaan dan pembuatan Tugas Akhir ini perlu dilakukan pembatasan masalah. Berdasarkan rumusan masalah yang telah disebutkan, maka pembahasan dibatasi pada :

1. Sistem membaca RSS menyajikan informasi RSS yang diambil dari situs-situs berikut ini :
  - a. <http://sindikasi.okezone.com/index.php/international/RSS2.0>
  - b. [http://www.gametrailers.com/rssgenerate.php?s1=&favplats\[ds\]=ds&type\[review\]=on&type\[preview\]=on&orderby=newest&limit=5](http://www.gametrailers.com/rssgenerate.php?s1=&favplats[ds]=ds&type[review]=on&type[preview]=on&orderby=newest&limit=5)
  - c. [http://www.worldcup2010southafrica.com/component/option,com\\_rss/feed,RSS2.0/no\\_html,1/](http://www.worldcup2010southafrica.com/component/option,com_rss/feed,RSS2.0/no_html,1/)
2. Sistem ini hanya membaca dan menampilkan RSS pada layar *user interface* Nintendo DS dalam format text.
3. Isi dari XML yang ditampilkan hanya dari tag *title* dan *description* yang berada didalam tag *item*.
4. Perangkat lunak aplikasi dibuat dengan menggunakan Sistem Operasi Windows XP, Dev-C++ versi 4.9.9.2, dan PALib versi 080823.

5. Komponen web selain RSS tidak dibahas lebih lanjut.
6. Perangkat keras Nintendo DS tidak dibahas lebih lanjut.

#### 1.4. Tujuan

Tujuan penelitian adalah untuk merancang dan membuat sistem RSS *reader* pada Nintendo DS menggunakan bahasa pemrograman C++.

#### 1.5. Manfaat

Manfaat penelitian ini antara lain:

1. Bagi penulis
  - a. Menerapkan ilmu yang telah diperoleh dalam Teknik Elektro konsentrasi sistem informatika dan komputer Universitas Brawijaya.
  - b. Mengembangkan aplikasi homebrew pada platform Nintendo DS.
2. Bagi pengguna
  - a. Mempercepat dan mempermudah proses penyampaian informasi berita kepada seluruh pengguna sistem ini.
  - b. Mengoptimalkan penggunaan perangkat Nintendo DS.

#### 1.6. Sistematika Penulisan

Penulisan skripsi ini akan diberikan uraian singkat dengan susunan penulisan sebagai berikut :

BAB I           Pendahuluan

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

BAB II           Tinjauan Pustaka

Berisi tentang teori dasar dan teori penunjang mengenai perancangan aplikasi untuk Nintendo DS, teori dasar pemrograman C dan C++, teori dasar parsing pada XML atau RSS.

BAB III         Metodologi Penelitian

Berisi tentang metode yang digunakan dalam pembahasan skripsi. Metode yang digunakan antara lain studi literatur, perancangan

perangkat lunak, implementasi perangkat lunak, pengujian dan analisis perangkat lunak serta pengambilan kesimpulan dan saran.

**BAB IV** Analisa Kebutuhan dan Perancangan

Membahas tentang analisa kebutuhan dari sistem dan kemudian merancang hal-hal yang berhubungan dengan analisa tersebut.

**BAB V** Implementasi Perangkat Lunak

Berisi tentang pembahasan mengenai implementasi perancangan dan implementasi antarmuka Sistem Aplikasi Penerima Berita Melalui RSS Pada Nintendo DS.

**BAB VI** Pengujian dan Analisis Perangkat Lunak

Berisi tentang pembahasan mengenai pengujian dan pengujian implementasi Sistem Aplikasi Penerima Berita Melalui RSS Pada Nintendo DS.

**BAB VII** Kesimpulan dan Saran

Berisi tentang kesimpulan dan saran-saran yang diperlukan untuk mengembangkan aplikasi selanjutnya.



## BAB II

### Tinjauan Pustaka

#### 2.1. Teori Dasar

Teori dasar yang digunakan yaitu, teori dasar bahasa pemrograman C dan C++, teori dasar XML atau RSS, dan teori dasar pembuatan aplikasi pada Nintendo DS dengan menggunakan PALib. Teori dasar bahasa pemrograman C dan C++ meliputi pembacaan dan penulisan file, serta seleksi kondisi. teori dasar XML atau RSS meliputi bagaimana cara penulisan dan pembacaan, serta cara kerja XML atau RSS. Teori dasar pembuatan aplikasi pada Nintendo DS dengan menggunakan PALib meliputi penggunaan library PALib untuk memanipulasi hardware Nintendo DS sesuai dengan keinginan user.

#### 2.2. XML

XML (*eXtensible Markup Language*) adalah sebuah format data yang dibangun dengan kemampuan melakukan transfer data antar platform [SIS-04]. XML merupakan markup language yang mirip dengan struktur HTML yang digunakan untuk mendesain dan mengembangkan page Web. Keunggulan XML jika dibandingkan dengan HTML adalah XML dapat memiliki jumlah tag yang tidak terbatas.

Dokumen XML dirangkaikan dengan XSL *stylesheets* memungkinkan isi dokumen anda dibaca pada sembarang sistem dan sembarang platform [MAR-04].

#### 2.3. RSS

RSS merupakan kependekan dari *Really Simple Syndication* (atau beberapa orang mengatakan *Rich Site Summary*). RSS merupakan format yang digunakan oleh situs-situs web untuk mendistribusikan update, judul-judul berita, rangkuman dan link untuk kembali ke halaman web aslinya. Ini berarti bahwa kita tidak perlu mengunjungi setiap situs satu persatu untuk melihat apa saja yang baru pada situs tersebut, kita hanya perlu melihat judul-judul berita dan cukup klik pada link yang disediakan untuk membaca artikel keseluruhannya. Situs-situs

yang memiliki RSS *feeds* biasanya akan menampilkan ikon "RSS" atau "XML" berwarna oranye [SNC-07].

#### 2.4. RSS Syntax

Untuk mengaplikasikan sebuah RSS dalam sebuah situs, terlebih dahulu dibuat sebuah dokumen RSS dan menyimpannya pada sebuah file berekstensi .xml. Kemudian mendaftarkan pada sebuah RSS agregator. Setiap hari *aggregator* mencari dokumen RSS dari website yang telah terdaftar dan menampilkan informasi tentang *feed*.

Contoh penulisan sintaks sebuah dokumen RSS direpresentasikan pada Gambar 2.1.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<rss version="2.0">
<channel>
<title>W3Schools Home Page</title>
<link>http://www.w3schools.com</link>
<description>Free web building tutorials</description>
<item>
<title>RSS Tutorial</title>
<link>http://www.w3schools.com/rss</link>
<description>New RSS tutorial on W3Schools</description>
</item>
<item>
<title>XML Tutorial</title>
<link>http://www.w3schools.com/xml</link>
<description>New XML tutorial on W3Schools</description>
</item>
</channel>
</rss>
```

Gambar 2.1. Penulisan sintaks sebuah dokumen RSS

Sumber : [W3C-09]

Baris pertama menunjukkan deklarasi XML, baris berikutnya merupakan deklarasi RSS yang menunjukkan bahwa RSS yang digunakan adalah versi RSS

2.0. Pada baris selanjutnya terdapat elemen `<channel>`. Elemen ini digunakan untuk menggambarkan RSS *feed*. Elemen `<channel>` memiliki anak elemen :

1. `<title>` yang mendefinisikan judul dari channel (W3Schools Home Page).
2. `<link>` yang mendefinisikan hyperlink ke channel (<http://www.w3schools.com>)
3. `<description>` yang mendefinisikan channel (*Free web building tutorials*)

Tiap elemen `<channel>` dapat memiliki satu atau lebih `<item>` elemen. Masing-masing elemen `<item>` mendefinisikan sebuah artikel atau “cerita” pada RSS feed.

Elemen `<item>` memiliki tiga anak elemen :

1. `<title>` yang mendefinisikan judul dari item (*RSS Tutorial*)
2. `<link>` yang mendefinisikan *hyperlink* ke item (<http://www.w3schools.com/rss>)
3. `<description>` yang mendefinisikan item (*New RSS tutorial on W3Schools*)

Dua baris terakhir digunakan untuk menutup elemen `<channel>` dan `<rss>` [W3C-09].

## 2.5. RSS Reader

RSS *reader* digunakan untuk membaca RSS *feed*. Sudah banyak RSS *reader* yang tersedia. Beberapa bekerja sebagai *web service*, dan beberapa terbatas pada sistem operasi (contoh: Windows, Mac, Linux, dll). Gambar 2.2 menampilkan contoh code javascript untuk membaca RSS *feed*. Secara garis besar RSS dibaca dengan cara *parsing* file RSS kemudian mengembalikan nilai entri dengan *field* yang terdapat di dalam file RSS. [WOR-09]

```

1. class FeedReader {
2.   SyndFeed feed;
3.   String url,description,title;
4.   int numEntries;
5.   FeedEntry entry[];
6.
7.   public FeedReader(String _url) {
8.     url=_url;
9.     try {
10. feed=new SyndFeedInput().build(new XmlReader(new URL(url)));

```

```
11. description=feed.getDescription();
12. title=feed.getTitle();
13.
14. java.util.List entrl=feed.getEntries();
15. Object [] o=entrl.toArray();
16. numEntries=o.length;
17.
18. entry=new FeedEntry[numEntries];
19. for(int i=0; i< numEntries; i++) {
20. entry[i]=new FeedEntry((SyndEntryImpl)o[i]);
21. println(i+": "+entry[i]);
22. }
23. }
24. catch(Exception e) {
25. println("Exception in FeedReader: "+e.toString());
26. e.printStackTrace();
27. }
28. }
29. }
```

Gambar 2.2. Contoh Kode Javascript Untuk Membaca RSS *feed*

Sumber : [WOR-09]

## 2.6. Nintendo DS

Nintendo DS adalah *portable gaming console* yang dikembangkan dan diproduksi oleh Nintendo, diluncurkan pada tahun 2004. Nintendo DS (Gambar 2.3) memiliki dua jenis layar dimana layar yang dibawah bekerja sebagai touch screen, sebuah mikrofon, dan mendukung standar wireless IEEE 802.11 (Wi-Fi). User dapat berinteraksi dengan user lainnya melalui fitur Wi-Fi dalam jarak dekat (kurang lebih 10-30m, tergantung pada kondisi), atau servis Nintendo Wi-Fi Connection. [ANO-09]



Gambar 2.3. Nintendo DS Lite

Sumber : [ANO-09]

### 2.7. Wi-Fi

Wi-Fi (*Wireless Fidelity*) yaitu sekumpulan standar yang digunakan untuk WLAN (*Wireless Local Area Networks*) yang didasari pada spesifikasi IEEE 802.11. Wi-Fi memperbolehkan beberapa device seperti PC, game console, handphone, PDA, dll. untuk terkoneksi ke internet jika berada di dalam area dari jaringan nirkabel yang terkoneksi ke internet. Daerah *coverage* dari satu atau lebih interkoneksi access point – atau biasa disebut *hotspot* – dapat membuat area kecil seperti sebuah ruangan atau daerah terbuka yang sangat luas dilingkupi oleh access point yang banyak. [ANY-09]

### 2.8. Nintendo Wi-Fi Connection

Nintendo Wi-Fi Connection (Nintendo WFC) telah dibuat untuk memudahkan konektivitas, aman untuk digunakan siapa saja, dan gratis. Permainan-permainan yang didesain untuk menggunakan kelebihan dari Nintendo Wi-Fi Connection menawarkan permainan secara online. Nintendo Wi-Fi Connection juga berguna untuk mengetahui apakah Nintendo DS sudah terkoneksi dengan internet. [ANO-09]

### 2.9. Nintendo DS Homebrew

Homebrew merupakan sebuah usaha untuk mengembangkan aplikasi (kebanyakan game) sehingga dapat dijalankan pada Nintendo DS. Komunitas homebrew tidak mendukung pembajakan atau penggunaan ilegal dari Nintendo

DS SDK oleh *non-licensed developers*. Komunitas homebrew menggunakan *unofficial development kits*, seperti devkitPro yang berisi LibNDS dan library tambahan PALib.

### 2.10. Nintendo DS Programming

Pemrograman pada devkitPro menggunakan bahasa C++. devkitPro menyediakan tool untuk *homebrew developer*. Tool-tool ini tersedia untuk Game Boy Advance, GP32, Playstation Portable, GameCube, and the Nintendo DS. Tool yang paling menarik adalah devkitARM.

devkitARM merupakan tool dari devkitPro. Tool ini memungkinkan untuk meng-*compile* ARM binari dari hampir semua komputer. Tool ini berdasar pada gcc, *gnu compiler collection*. devkitARM dilengkapi semua hal yang dibutuhkan untuk membuat software untuk Nintendo DS, GBA, dan GP32 yang akan dijalankan pada prosesor ARM.

Untuk membuat suatu aplikasi pada Nintendo DS dibutuhkan library tersendiri yaitu libnds. Nama libnds mula-mula adalah NDSLIB. NDSLIB mula-mula merupakan kumpulan lokasi memori umum pada DS. Kegunaannya adalah user dapat menggantikan referensi memori 0x06000000 dengan *statement* BG\_BMP\_RAM. Sekarang ini libnds digunakan oleh 96% komunitas *homebrew* [AME-08].

### 2.11. Libnds

Libnds, yang awalnya bernama ndslib, adalah *library* yang memberikan dukungan hampir semua fitur untuk Nintendo DS, antara lain: *touchscreen*, mikrofon, 2D atau 3D [BOU-09]. Libnds tidak bertujuan untuk memberikan komponen-komponen yang lebih maju yang akan melakukan semua pekerjaan untuk kita. Libnds hanya merupakan sebuah lapisan tipis yang akan mengabstraksi fasilitas-fasilitas perangkat keras Nintendo DS, berkat primitif-primitif tingkat rendah yang sesuai dimana akan menyembunyikan detail-detail teknis yang tidak diinginkan, seperti lokasi sebenarnya dari register-register di memori.

Libnds cukup tipis untuk tidak benar-benar mengurangi performansi bila dibandingkan dengan tingkat pemrograman yang paling rendah. Bila ternyata kita tidak menggunakan secara efektif keseluruhan libnds, ada baiknya kita memilih atau menggunakan library-library yang memiliki tingkatan lebih tinggi. [AME-08]

### 2.12. Libfat

Beberapa aplikasi-aplikasi dan permainan membutuhkan beberapa cara untuk membaca dan menulis dari dan ke media yang tetap, dan untuk mengaksesnya sebagai sistem file daripada sebuah sekuens dari blok-blok data. Libfat adalah library yang berguna untuk membaca file-file sumber (gambar, suara, dll.) dan untuk menulis score, setting, dan lain-lain.

Beberapa cartridge slot-1 (dan slot-2) sudah memiliki sistem file didalamnya (biasanya berbasis FAT, tetapi bisa juga berbasis ext2 atau lainnya), dan biasanya aplikasi-aplikasi homebrew bisa mempergunakan hal ini.

Pada kasus di Nintendo DS, diantara banya sistem file yang bisa digunakan, satu diantaranya, FAT16, adalah yang paling sering digunakan. Hal ini tergantung pada libnds, dimana sudah termasuk dari paket DevkitARM. [AME-08]

### 2.13. DLDI

DLDI adalah *Dynamically Linked Disc Interface* untuk Game Boy Advance dan Nintendo DS. DLDI dibuat oleh Michael "Chishm" Chisholm dan bekerja sebagai bagaian dari libfat untuk mengizinkan program-program homebrew membaca dan menulis pada kartu memori yang dimasukkan pada salah satu slot dari sistem. [AME-08]

### 2.14. Dswifi

Dswifi adalah sebuah libabry yang berguna untuk memberikan fitur antarmuka tingkat rendah untuk konektivitas dengan Nintendo DS Wi-Fi. *Access point*, koneksi-koneksi, alamat IP, frame, dan packet dapat diatasi oleh mudah berkat library ini. Hal ini mengizinkan untuk menggunakan konektivitas standar Wi-Fi (IEEE 802.11). [BOU-09]

### 2.15. PALib

PALib merupakan salah satu dari *library abstract* yang memiliki tingkatan lebih tinggi. PALib dibangun diatas libnds dan memiliki berbagai fitur servis-servis tingkatan yang lebih tinggi terintegrasi untuk para pembuat *homebrew*. *Library* PALib mengenkapsulasi operasi-operasi yang berhubungan dengan hardware ke tingkatan struktur dan fungsi C yang lebih tinggi, dimana memiliki ciri khas dengan awalan PA\_.

Penggunaan PALib lebih ditujukan kepada pembuat *homebrew* yang ingin memiliki metode penggunaan waktu lebih efektif untuk menulis perangkat lunak mereka. PALib didesain untuk bisa digunakan dengan mudah oleh siapa saja. Orang-orang yang mempelajari *hardware (hardware purists)* sering menganggap *library* apapun yang berada diluar libnds sebagai hal yang tidak berguna, restriktif, atau tidak optimal dalam performansinya. Secara objektif, bisa dikatakan bahwa libnds berguna untuk mengetahui bagaimana cara Nintendo DS dan hal-hal lain bekerja. Sedangkan PALib lebih memudahkan para pembuat *homebrew* dikarenakan tidak perlu memperhatikan hal-hal yang terlalu detil. [BOU-09]

### 2.16. Bahasa Pemrograman C dan C++

Pada tahun 1980, Bjarne Stroustrup, dari laboratorium Bell, memulai pengembangan dari bahasa C++. Pada tahun 1985, bahasa pemrograman C++ diluncurkan secara komersial dan juga buku panduan C++ yang pertama berjudul "*The C++ Programming Language*" oleh Bjarne Stroustrup.

Pada tahun sekitar tahun 1980, bahasa C++ terus menerus diperbaiki hingga akhirnya menjadi bahasa pemrograman yang memiliki ciri khas tersendiri [MSH-08]. Keistimewaan yang sangat berarti dari bahasa C++ ini adalah karena bahasa ini mendukung pemrograman yang berorientasi obyek (PBO atau OOP/*Object Oriented Programming*) sehingga dapat mengurangi kekompleksitasan suatu kode aplikasi.

Salah satu konsep yang penting dalam bahasa pemrograman C++ adalah struktur. Pada Gambar 2.4 dituliskan struktur kerangka umum dari program yang ada di dalam C++. [MSH-08]

```
// mengisikan daftar file header yang digunakan
#include <nama file header>
// mendeklarasikan variabel, konstanta atau fungsi yang akan digunakan
bagian_deklarasi;
// membuat fungsi utama yang mengembalikan nilai dengan tipe data int
int main ( )
{ // untuk memulai blok program
  statemen_statemen;
  return 0; // mengembalikan nilai ke sistem operasi
} // untuk mengakhiri blok program
```

Gambar 2.4. Struktur Kerangka Umum dari Program yang ada di Dalam C++

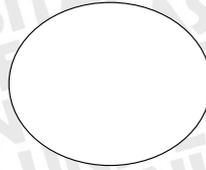
Sumber : [MSH-08]

### 2.17. *Data Flow Diagram (DFD)*

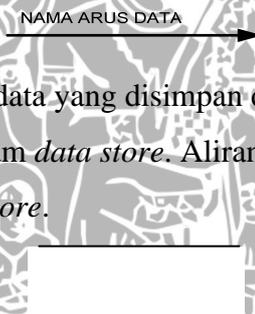
*Data Flow Diagram (DFD)* merupakan diagram yang digunakan untuk menggambarkan prose-prose yang terjadi pada sistem yang akan dikembangkan. Dengan model ini, data-data yang terlibat pada masing-masing proses dapat diidentifikasi. Pengembangan DFD menggunakan cara berjenjang. Dimulai dari context diagram, DFD level 1, level 2, dan seterusnya sesuai dengan kompleksitas dari sistem yang akan dikembangkan [ALF-07:119].

Empat elemen yang menyusun suatu *Data flow Diagram (DFD)* beserta lambang tertentu untuk membedakan satu dengan yang lain, antara lain [ALF-07:106] :

1. Proses, aktifitas atau fungsi yang dilakukan untuk alasan bisnis yang spesifik, biasa berupa manual atau terkomputerisasi.



Beberapa hal yang harus dihindari dalam menggambarkan suatu proses, yaitu:

- a. Proses yang mempunyai masukan tetapi tidak menghasilkan keluaran, kesalahan ini disebut *Black Hole*.
  - b. Proses yang menghasilkan keluaran tetapi tidak mendapatkan masukan, kesalahan ini disebut *Miracle*.
  - c. Proses yang menghasilkan keluaran lengkap tetapi mendapat masukan dengan data terbatas atau masukan kurang lengkap, kesalahan ini disebut *Grey Hole*.
2. *Data Flow*, satu data tinggal atau kumpulan logis suatu data, selalu diawali atau berakhir pada sebuah proses.
- 
3. *Data Store*, kumpulan data yang disimpan dengan cara tertentu. Data yang mengalir disimpan dalam *data store*. Aliran data di-update atau ditambahkan ke *data store*.

Beberapa hal yang perlu diperhatikan dalam menggambarkan penyimpanan data, antara lain:

- a. Hanya proses saja yang berhubungan dengan penyimpanan data karena proses yang menggunakan atau mengubah data di penyimpanan.
- b. Arus data yang menuju ke penyimpanan data dari suatu proses menunjukkan proses *update* terhadap data yang tersimpan di penyimpanan data.
- c. Arus data yang berasal dari penyimpanan data ke suatu proses menunjukkan bahwa arus tersebut menggunakan data yang ada di penyimpanan data.

4. *External Entity*, berupa orang, organisasi, atau sistem yang berada diluar sistem tetapi berinteraksi dengan sistem.

NAMA ENTITAS

Dekomposisi adalah proses untuk menggambarkan system dalam hierarki dari diagram DFD. Diagram anak menggambarkan proses yang lebih detail dibandingkan dengan diagram induk. Terdapat proses *balancing* untuk menjamin keakuratan informasi yang disajikan dalam satu level DFD dari suatu DFD secara akurat direpresentasikan pada DFD level berikutnya [ALF-07:108].

Hierarki tersebut dijelaskan sebagai berikut [ALF-07:109] :

1. *Context Diagram*

DFD pertama dalam proses bisnis. Menunjukkan konteks dimana proses bisnis berada. Menunjukkan semua proses bisnis dalam satu proses tunggal (proses 0). *Context diagram* juga menunjukkan semua entitas luar yang menerima informasi dari atau memberikan informasi ke sistem.

2. Diagram Level 0

Menunjukkan semua proses utama yang menyusun keseluruhan system. Level ini juga menunjukkan komponen internal dari proses 0 dan menunjukkan bagaimana proses-proses utama direlasikan menggunakan *data flow*. Pada level ini juga ditunjukkan bagaimana proses-proses utama terhubung dengan entitas eksternal. Pada level ini juga dilakukan penambahan data store.

3. Diagram Level 1

Diagram level 1 diciptakan dari level 0. level ini menunjukkan proses-proses internal yang menyusun setiap proses-proses utama dalam level 0, sekaligus menunjukkan bagaimana informasi berpindah dari satu proses ke proses lainnya.

4. Diagram Level 2

Menunjukkan semua proses yang menyusun sebuah proses pada level 1. bias saja penyusunan DFD tidak mencapai level 2 ini, atau mungkin harus dilanjutkan ke level berikutnya (level 3, level 4, dan seterusnya).

## 2.18. White Box Testing

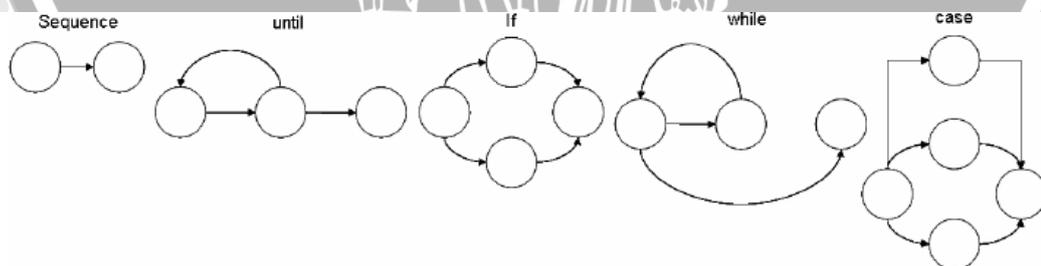
*White box testing* merupakan metode perancangan *test case* yang menggunakan struktur kontrol dari perancangan prosedural untuk mendapatkan *test case*. Dengan menggunakan metode *white box*, analisis sistem akan dapat memperoleh *test case* yang:

- Menjamin seluruh independent path di dalam modul yang dikerjakan sekurang-kurangnya sekali
- Mengerjakan seluruh keputusan logikal
- Mengerjakan seluruh loop yang sesuai dengan batasannya
- Mengerjakan seluruh struktur data internal yang menjamin validitas

### 2.18.1. Uji Coba Basis Path

Merupakan teknik uji coba *white box* yang diusulkan Tom McCabe. Metode ini memungkinkan perancang *test case* mendapatkan ukuran kekompleksan *logical* dari perancangan prosedural dan menggunakan ukuran ini sebagai petunjuk untuk mendefinisikan basis set dari jalur pengerjaan. *Test case* yang didapat digunakan untuk mengerjakan basis set yang menjamin pengerjaan setiap perintah minimal satu kali selama uji coba.

### 2.18.2. Notasi Diagram Alir



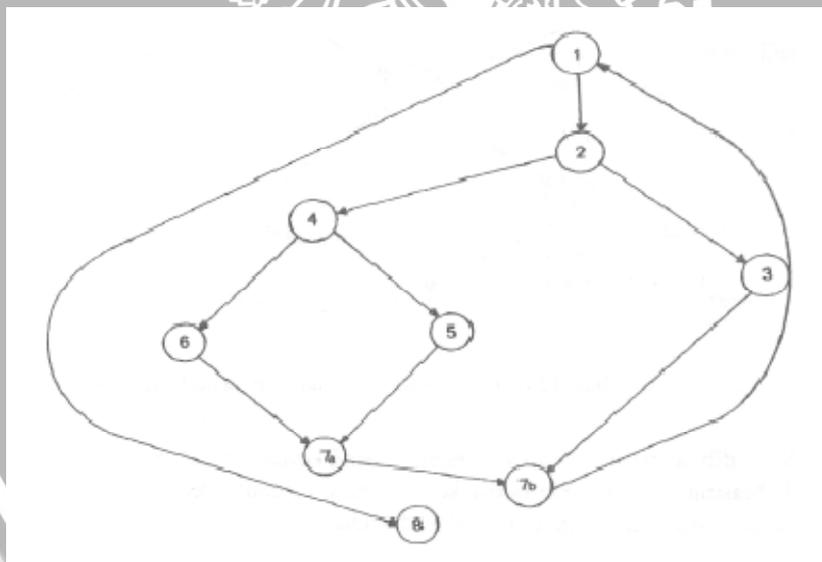
Gambar 2.5. Gambar Notasi Diagram Alir

Sumber: [SYA-08]

Lingkaran (*node*), menggambarkan satu/lebih perintah prosedural. Urutan proses dan keputusan dapat dipetakan dalam satu node. Tanda panah (*edge*),

menggambarkan aliran kontrol. Setiap node harus mempunyai tujuan *node*. *Region* adalah daerah yang dibatasi oleh *edge* dan *node*. Termasuk daerah diluar grafik alir. Contoh menterjemahkan *pseudo code* ke grafik alir

```
1 : do while record masih ada
    baca record
2 : if record ke 1 = 0
3 : then proses record
    simpan di buffer
    naikan counter
4 : else if record ke 2 = 0
5 : then riser counter
6 : proses record
    simpan pada file
7a: endif
    endif
7b: enddo
8 : end
```

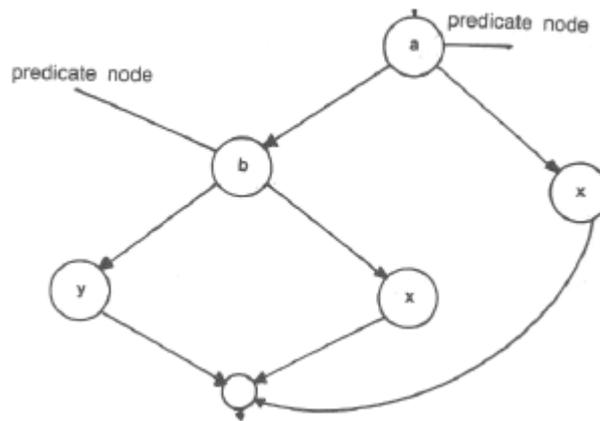


Gambar 2.6. Diagram Alir Contoh Diatas

Sumber: [SYA-08]

Nomor pada *pseudo code* berhubungan dengan nomor *node*. Apabila ditemukan kondisi majemuk (*compound condition*) pada *pseudo code* pembuatan grafik alir menjadi rumit. Kondisi majemuk mungkin terjadi pada operator Boolean (AND, OR, NAND, NOR) yang dipakai pada perintah if.

Contoh :



```

if A or B
then procedure x
else procedure y
endif
  
```

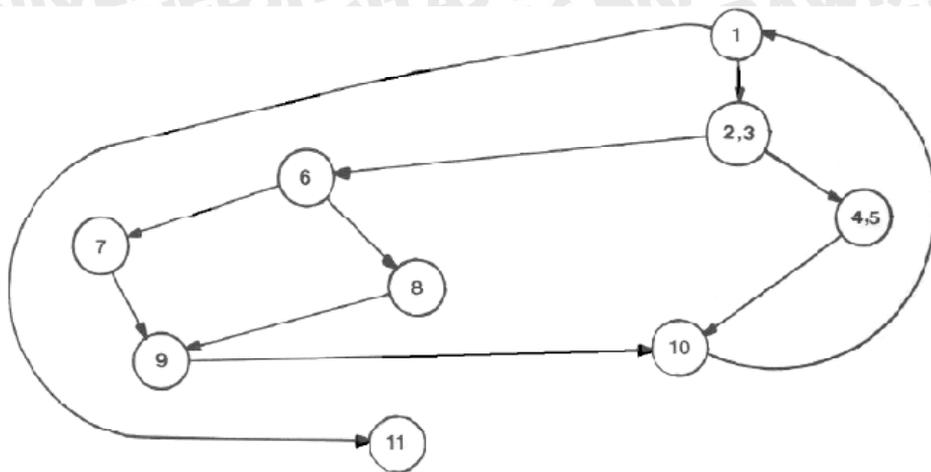
Gambar 2.7. Logika Gabungan

Sumber: [SYA-08]

*Node* dibuat terpisah untuk masing-masing kondisi A dan B dari pernyataan IF A OR B. Masing-masing node berisi kondisi yang disebut *pridicate node* dan mempunyai karakteristik dua atau lebih edge darinya.

### 2.18.3. Cyclomatic Complexity

*Cyclomatic complexity* adalah metrik software yang menyediakan ukuran kuantitatif dari kekompleksan logikal program. Apabila digunakan dalam konteks metode uji coba basis path, nilai yang dihitung untuk cyclomatic complexity menentukan jumlah jalur independen dalam basis set suatu program dan memberi batas atas untuk jumlah uji coba yang harus dikerjakan untuk menjamin bahwa seluruh perintah sekurang-kurangnya telah dikerjakan sekali. Jalur independent adalah jalur yang melintasi atau melalui program dimana sekurang-kurangnya terdapat proses perintah yang baru atau kondisi yang baru.



Gambar 2.8. Diagram Alir Contoh

Sumber: [SYA-08]

Dari gambar:

Path 1 = 1 - 11

Path 2 = 1 - 2 - 3 - 4 - 5 - 10 - 1 - 11

Path 3 = 1 - 2 - 3 - 6 - 8 - 9 ...: 10 - 1 - 11

Path 4 = 1 - 2 - 3 - 6 - 7 - 9 - 10 - 1 - 11

Path 1,2,3,4 yang telah didefinisikan diatas merupakan basis set untuk diagram alir.

Cyclomatic complexity digunakan untuk mencari jumlah path dalam satu flowgraph. Dapat dipergunakan rumusan sebagai berikut :

1. Jumlah region grafik alir sesuai dengan cyclomatic complexity.
2. Cyclomatic complexity  $V(G)$  untuk grafik alir dihitung dengan rumus:

$$V(G) = E - N + 2$$

Dimana:

E = jumlah edge pada grafik alir

N = jumlah node pada grafik alir

3. Cyclomatic complexity  $V(G)$  juga dapat dihitung dengan rumus:

$$V(G) = P + 1$$

Dimana P = jumlah predicate node pada grafik alir [SYA-08]



### 2.19. *Black Box Testing*

Pengujian *black-box* berfokus pada persyaratan fungsional perangkat lunak. Pengujian ini memungkinkan analisis sistem memperoleh kumpulan kondisi input yang akan mengerjakan seluruh keperluan fungsional program.

Tujuan metode ini mencari kesalahan pada:

- a. Fungsi yang salah atau hilang
- b. Kesalahan pada *interface*
- c. Kesalahan pada struktur data atau akses database
- d. Kesalahan performansi
- e. Kesalahan inisialisasi dan tujuan akhir

Metode ini tidak terfokus pada struktur kontrol seperti pengujian *white-box* tetapi pada domain informasi.

Pengujian dirancang untuk menjawab pertanyaan sebagai berikut:

- a. Bagaimana validitas fungsional diuji?
- b. Apa kelas input yang terbaik untuk uji coba yang baik?
- c. Apakah sistem sangat peka terhadap nilai input tertentu?
- d. Bagaimana jika kelas data yang terbatas dipisahkan?
- e. Bagaimana volume data yang dapat ditoleransi oleh sistem?
- f. Bagaimana pengaruh kombinasi data terhadap pengoperasian sistem?

[SYH-08]

## BAB III

### METODOLOGI PENELITIAN

Metodologi penelitian menjelaskan mengenai langkah-langkah yang dilakukan untuk merealisasikan aplikasi perangkat lunak yang akan dibuat. Langkah-langkah yang diperlukan antara lain studi literatur, perancangan perangkat lunak, implementasi perangkat lunak, pengujian dan analisis perangkat lunak serta pengambilan kesimpulan dan saran.

#### 3.1. Studi Literatur

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

1. Pengertian dan konsep dasar dari bahasa pemrograman C dan C++.
2. Pengertian dan konsep dasar tentang struktur bahasa XML dan RSS.
3. Pengertian dan konsep dasar library-library penunjang yang akan digunakan untuk membuat program pada Nintendo DS.

#### 3.2. Perancangan Perangkat Lunak

Perancangan perangkat lunak terdiri dari perancangan sistem, perancangan proses, dan perancangan antarmuka. Perancangan didasarkan pada teori-teori yang nantinya akan diaplikasikan pada sistem. Perancangan perangkat lunak dilakukan pada tiap-tiap blok aplikasi sistem untuk mempermudah analisis terstruktur, perancangan, dan pemrograman.

Perancangan aplikasi dilakukan untuk mempermudah implementasi, analisis, algoritma, dan pengujian. Perancangan aplikasi berdasarkan pemodelan terstruktur. Pada tahap ini perancangan dilakukan dengan membuat data-flow diagram atau flow chart.

### **3.3. Implementasi Perangkat Lunak**

Implementasi aplikasi dilakukan dengan mengacu kepada perancangan aplikasi. Implementasi dilakukan dengan mengeksekusi perangkat lunak yang telah jadi langsung pada hardware Nintendo DS.

### **3.4. Pengujian dan Analisis Perangkat Lunak**

Pengujian dan analisis perangkat lunak merupakan tahap untuk menguji sistem yang dibuat dan dibandingkan dengan teori dan perancangan. Pengujian dan analisis perangkat lunak dilakukan untuk mendapatkan kesimpulan mengenai Sistem Aplikasi Penerima Berita Melalui RSS Pada Nintendo DS yang telah dibuat.

Pengujian yang dilakukan adalah mengimplementasikan Sistem Aplikasi Penerima Berita Melalui RSS Pada Nintendo DS secara online dan dilakukan langsung pada hardware Nintendo DS. Pengujian dan analisis dilakukan terhadap permasalahan-permasalahan yang ada selama proses aplikasi dijalankan.

### **3.5. Pengambilan Kesimpulan dan Saran**

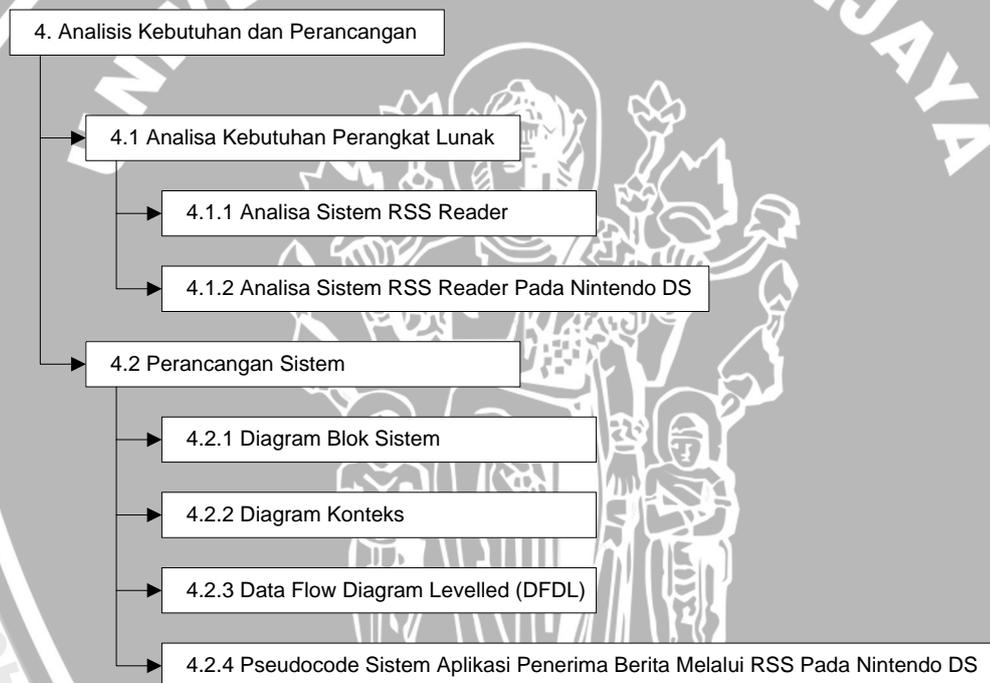
Pengambilan kesimpulan dilakukan setelah semua tahapan perancangan, implementasi dan pengujian sistem aplikasi telah selesai dilakukan dan didasarkan pada kesesuaian antara teori dan praktek. Kesimpulan diambil untuk menjawab rumusan masalah yang telah ditetapkan sebelumnya. Tahap terakhir dari penulisan adalah saran yang dimaksudkan untuk memperbaiki kesalahan-kesalahan yang terjadi dan menyempurnakan penulisan serta untuk memberikan pertimbangan atas pengembangan aplikasi selanjutnya.

## BAB IV

### ANALISIS KEBUTUHAN DAN PERANCANGAN

Bab ini membahas mengenai analisa kebutuhan dan perancangan perangkat lunak atau aplikasi Penerima Berita Melalui RSS Pada Nintendo DS. Perancangan yang dilakukan meliputi empat tahap, yaitu analisa kebutuhan perangkat lunak, perancangan sistem, dan perancangan antar muka.

Gambar 4.1. menunjukkan langkah-langkah proses analisis kebutuhan dan perancangan yang digunakan pada bab ini.



Gambar 4.1. Diagram Pohon Analisis Kebutuhan dan Perancangan

Sumber: [Pengujian]

#### 4.1. Analisa Kebutuhan Perangkat Lunak

Analisis kebutuhan perangkat lunak adalah aktifitas rekayasa perangkat lunak yang digunakan menjembatani antara kebutuhan ditingkat sistem dan perancangan perangkat lunak. Analisis kebutuhan adalah proses yang digunakan untuk mendapatkan, menganalisis, dan memvalidasi kebutuhan-kebutuhan sistem.

#### 4.1.1. Analisa Sistem RSS Reader

Secara garis besar perancangan RSS Reader pada Sistem Aplikasi Penerima Berita Melalui RSS Pada Nintendo DS memiliki sistem kerja sebagai berikut :

1. RSS Feed

RSS feed (RSS biasa disebut sebagai news feed atau RSS feed) sebuah file yang memiliki format penulisan XML. RSS masih belum memiliki standard penulisan secara resmi. Sebagian besar RSS Feed menggunakan format RSS 0.91 dan sisanya menggunakan format RSS 1.0 dan 2.0. Sehingga dari satu situs ke situs lainnya bisa saja memiliki bentuk penulisan RSS yang berbeda-beda.

2. Parsing

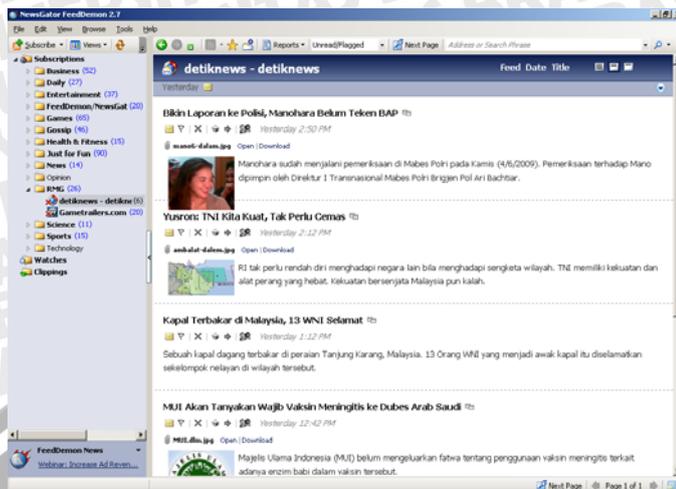
Proses parsing adalah proses membaca dari RSS feed yang diinginkan dan mengambil bagian-bagian tertentu (filtering) untuk ditampilkan.

3. Tampilan

Setelah file XML yang berisi RSS diparsing, hal selanjutnya adalah ditampilkan dalam pada user interface agar bisa dibaca oleh client.

RSS reader umumnya sering dijumpai dalam bentuk aplikasi standalone untuk PC, web-based RSS reader, e-mail client, dan pada telepon genggam atau PDA. Setiap RSS reader memiliki kelebihan dan kekurangan masing-masing.

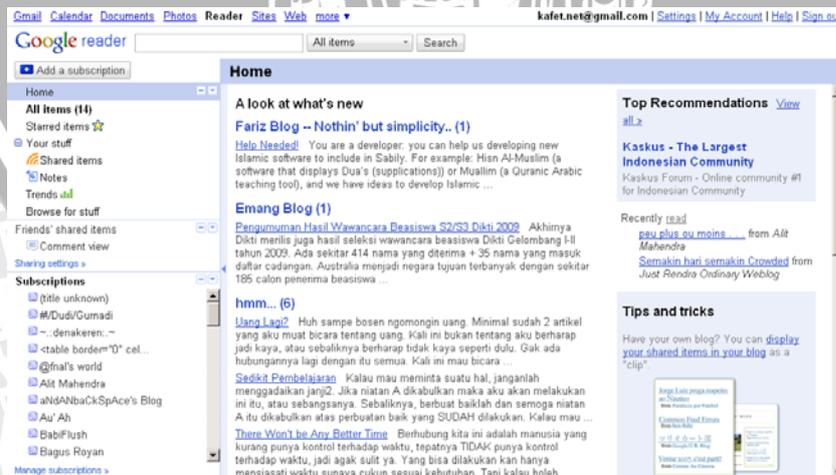
Untuk RSS reader pada aplikasi desktop PC, saya mengambil contoh FeedDemon. Kelebihan dari aplikasi desktop seperti ini adalah kita tidak perlu online untuk melihat berita, jadi sekali kita tersinkronisasi dengan internet, anda bisa membaca RSS feed yang telah tersimpan secara offline. Kelemahan yang dimiliki dari RSS reader ini adalah bila koneksi kita terputus dari internet, maka kita tidak bisa melihat konten HTML atau video yang berada di RSS. Hal ini merupakan kekurangan yang cukup signifikan yang akan menjadi penghambat bagi beberapa video-video YouTube yang dicantumkan pada berita blog tiap harinya.



Gambar 4.2. Tampilan FeedDemon

Sumber: [Analisis]

RSS reader yang berbasis web merupakan RSS feed reader yang paling sering digunakan. Tidak seperti RSS reader pada desktop, RSS reader berbasis web secara otomatis langsung merefresh konten. Mereka juga memiliki kelebihan bahwa mereka memiliki kinerja yang sama pada setiap komputer apapun sistem operasinya. Selama kita memiliki akses untuk sebuah browser, kita dapat selalu mengikuti feed yang telah kita pilih. RSS reader yang berbasis web paling umum digunakan adalah Google Reader. Google Reader sangat mudah digunakan dan memiliki banyak fitur. Kelemahan yang dimiliki Google Reader adalah tampilannya yang terlalu minimalis dan tidak memiliki fitur drag-and-drop.

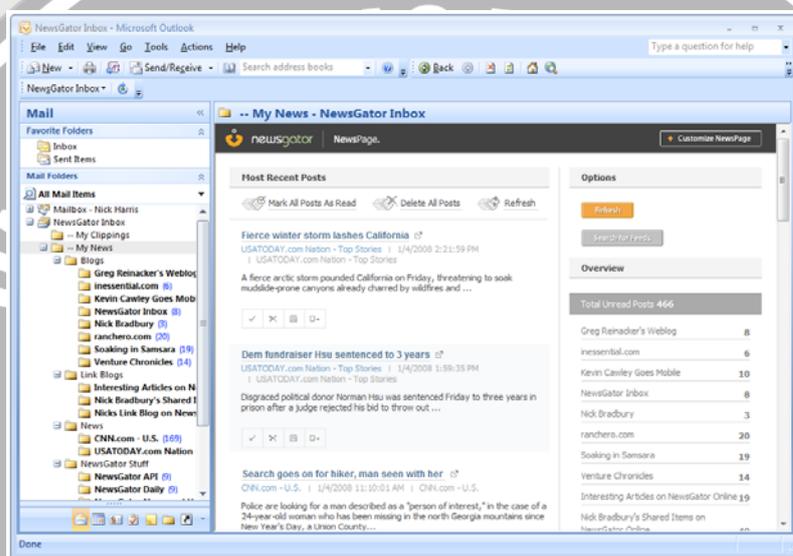


Gambar 4.3. Tampilan Google Reader

Sumber: [Analisis]



RSS reader pada e-mail juga memiliki sistem kerja yang hampir sama dengan RSS reader yang berbasis web, karena untuk membuka e-mail kita harus terhubung dengan internet dahulu. Untuk e-mail dengan pelayanan offline, user menggunakan Microsoft Outlook. RSS reader untuk Microsoft Outlook adalah NewsGator. NewsGator memiliki fitur drag-and-drop, sehingga NewsGator sangat mudah digunakan. Tetapi kelemahan yang dimilikinya antara lain : NewsGator hanya bisa digunakan untuk Microsoft Outlook saja, dan kebanyakan kesusahan dalam menginstall dan mengatur NewsGator.



Gambar 4.4. Tampilan NewsGator

Sumber: [Analisis]

RSS reader yang berfitur mobile (digunakan oleh smartphone, PDA, dan telepon genggam) secara umum berupa RSS reader yang berbasis web yang telah disesuaikan untuk fitur-fitur tertentu (ukuran layar, tampilan, dll).



Gambar 4.5. Tampilan Google Reader pada PDA

Sumber: [Analisis]



Gambar 4.6. Tampilan Google Reader pada Opera Mini

Sumber: [Analisis]

#### 4.1.2. Analisa Sistem Aplikasi Penerima Berita Melalui RSS Pada Nintendo DS

Sistem Aplikasi Penerima Berita Melalui RSS Pada Nintendo DS merupakan aplikasi homebrew yang dijalankan pada Nintendo DS dan memberikan manfaat agar user dapat mengikuti berita terkini melalui RSS feed yang telah tercantum pada menu pilihan.

Sistem Aplikasi Penerima Berita Melalui RSS Pada Nintendo DS dirancang untuk dapat melakukan:

1. User dapat membaca RSS feed yang telah diparsing pada layar Nintendo DS.
2. User dapat memilih dari RSS feed yang telah disediakan.

#### 4.2. Perancangan Sistem

Perancangan sistem merupakan tahap kedua dari perancangan perangkat lunak. Perancangan ini dilakukan untuk mengetahui aplikasi sistem yang akan dibuat secara umum. Perancangan sistem meliputi diagram blok sistem, diagram konteks dan prinsip kerja sistem.

##### 4.2.1. Diagram Blok Sistem

Sistem ini dirancang untuk dijalankan dalam jangkauan Access Point untuk terhubung ke dunia internet, seperti yang diperlihatkan pada Gambar 4.7.

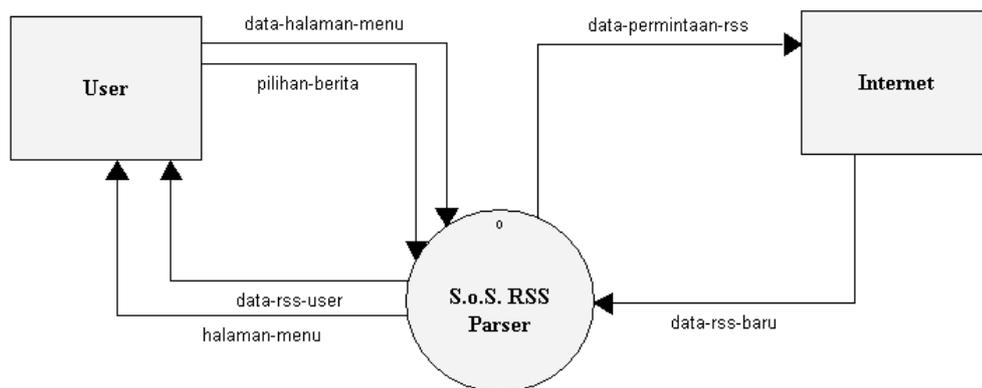


Gambar 4.7. Diagram Blok Sistem Aplikasi Penerima Berita Melalui RSS Pada Nintendo DS

Sumber: [Perancangan]

##### 4.2.2. Diagram Konteks

Diagram konteks merupakan gambaran umum dari suatu sistem yang mencakup masukan-masukan dasar, sistem umum, dan keluaran. Untuk membuat diagram ini, kita harus mengelompokkan kejadian-kejadian berdasarkan data/informasi yang masuk dan dihasilkan sistem beserta sumber/tujuannya. Diagram konteks merupakan bagian dari *Data Flow Diagram* (DFD) yang berfungsi memetakan model lingkungan, yang dipresentasikan dengan lingkaran tunggal yang mewakili keseluruhan sistem ditunjukkan dalam Gambar 4.8.



Gambar 4.8. Diagram Konteks Sistem Aplikasi Penerima Berita Melalui RSS Pada Nintendo DS (S.o.S. RSS Parser)

Sumber: [Perancangan]

#### 4.2.3. Data Flow Diagram Levelled (DFDL)

*Data Flow Diagram (DFD)* merupakan diagram yang berfungsi untuk menggambarkan sistem sebagai jaringan kerja antar fungsi yang berhubungan satu sama lain dengan aliran dan penyimpanan data. Berdasarkan kejadian-kejadian diatas dapat dibuat *data flow diagram*, yang bertujuan untuk mengetahui aliran data seluruhnya dari sistem.

##### 4.2.3.1. Data Flow Diagram Level 0 Sistem Aplikasi Penerima Berita Melalui RSS Pada Nintendo DS

DFD level 0 merupakan penjabaran dari diagram konteks yang dapat dijabarkan lebih lanjut pada level berikutnya. DFD level 0 mencantumkan entitas yang berhubungan dengan sistem. Pada DFD level 0 terdiri dari dua proses yaitu :

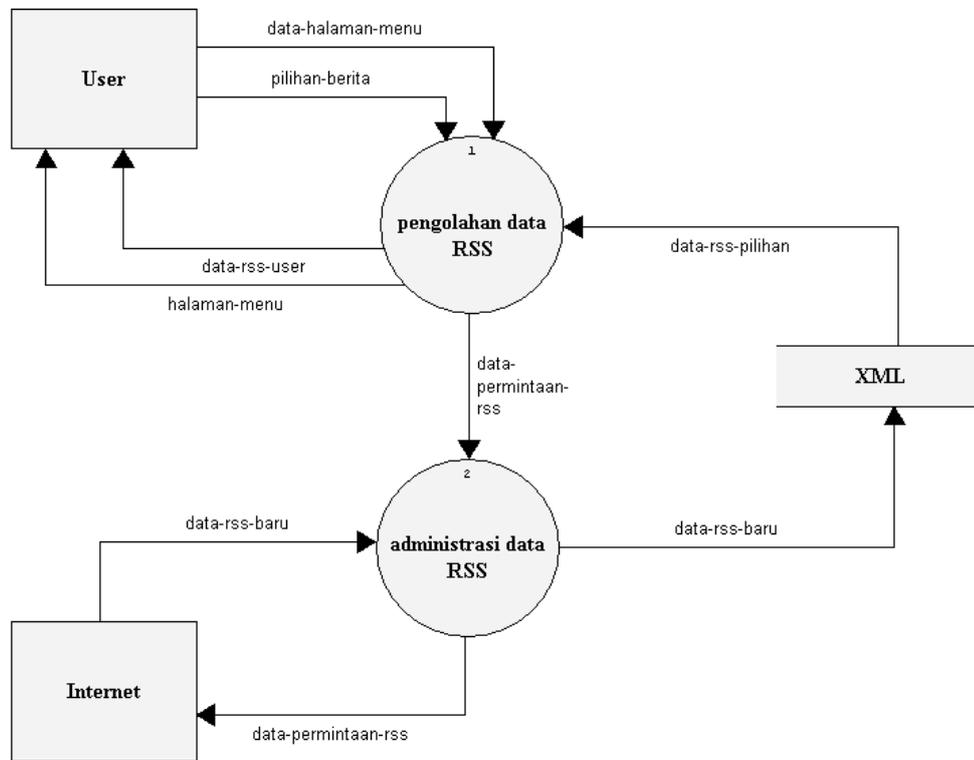
1. Proses 1 – Pengolahan Data RSS [sos10]

Proses Pengolahan data RSS merupakan proses manajemen data RSS, yaitu masukan dan keluaran user. Proses pengolahan data RSS dijelaskan pada Data Flow Diagram Level 1 terhadap proses 1.

2. Proses 2 – Administrasi data RSS [sos20]

Proses Administrasi data RSS merupakan proses manajemen data baru RSS yang diambil dari internet. Proses administrasi data RSS dijelaskan pada Data Flow Diagram Level 1 pada proses 2.

DFD Level 0 Sistem Aplikasi Penerima Berita Melalui RSS Pada Nintendo DS ditunjukkan pada Gambar 4.9.

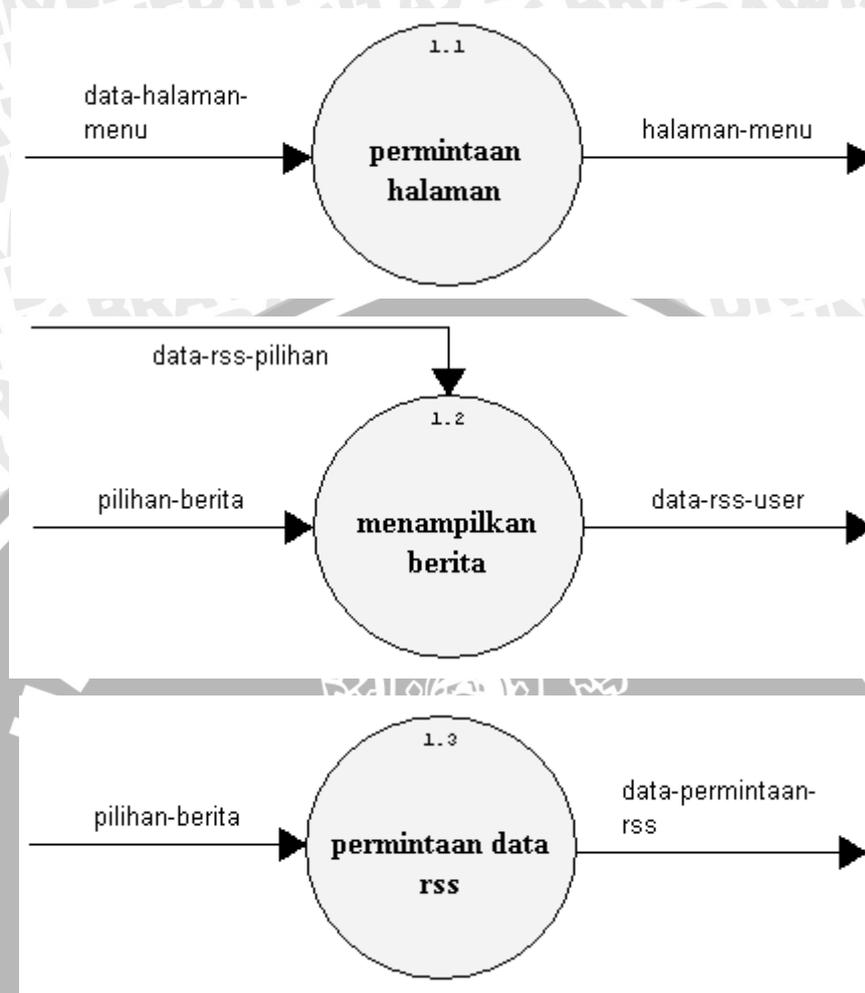


Gambar 4.9. Data Flow Diagram Level 0 Sistem Aplikasi Penerima Berita Melalui RSS Pada Nintendo DS

Sumber: [Perancangan]

#### 4.2.3.2. Data Flow Diagram Level 1 Pengolahan Data RSS

Pada pengolahan data RSS terdiri dari permintaan halaman, menampilkan berita, dan permintaan data rss. Data flow diagram level 1 pengolahan data RSS ditunjukkan dalam Gambar 4.10.



Gambar 4.10. Data Flow Diagram Level 1 Pengolahan Data RSS

Sumber: [Perancangan]

Pada pengolahan data RSS, sistem menerima masukan dari user. Pengolahan data RSS terdiri dari tiga macam proses, yaitu:

1. Proses 1.1 – Permintaan Halaman [sos11]

Proses Permintaan Halaman merupakan proses manajemen data halaman menu yang ditampilkan ke user dalam S.o.S. RSS Parser.

2. Proses 1.2 – Menampilkan Berita [sos12]

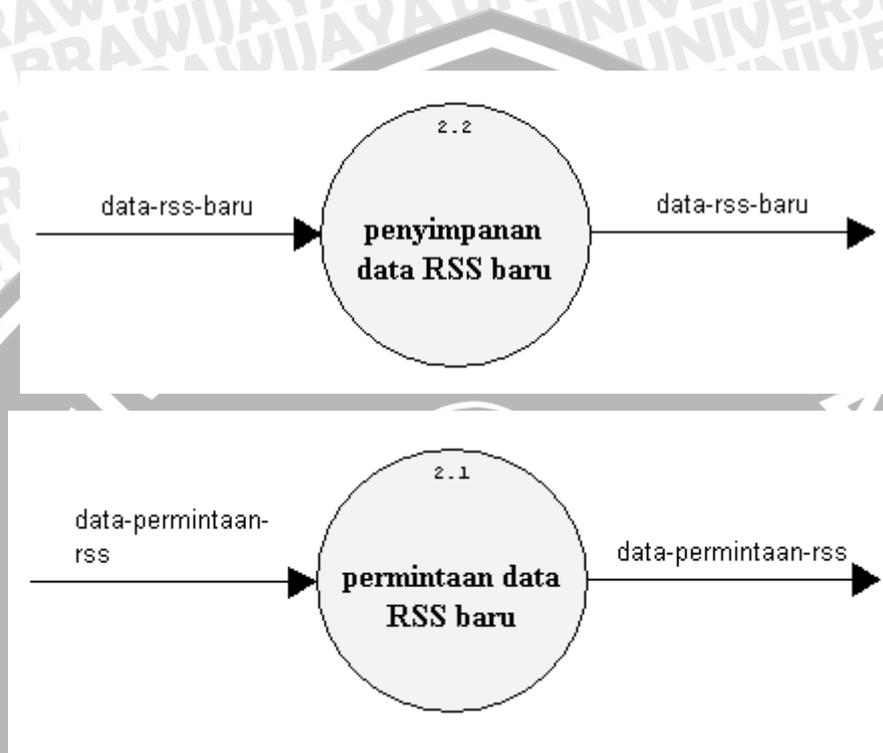
Proses Menampilkan Berita merupakan proses manajemen data RSS dan menampilkan RSS yang telah diparsing dalam S.o.S. RSS Parser.

3. Proses 1.3 – Permintaan Data RSS [sos13]

Proses Permintaan Data RSS merupakan proses manajemen data RSS yang dipilih oleh user dalam S.o.S. RSS Parser.

#### 4.2.3.3. Data Flow Diagram Level 1 Administrasi Data RSS

Pada administrasi data RSS terdiri dari penyimpanan data RSS baru, dan permintaan data RSS baru. Data flow diagram level 1 administrasi data RSS ditunjukkan dalam Gambar 4.11.



Gambar 4.11. Data Flow Diagram Level 1 Administrasi Data RSS

Sumber: [Perancangan]

Pada administrasi data RSS, sistem mengirim dan menerima RSS baru dari internet. Administrasi data RSS terdiri dari dua macam proses, yaitu:

1. Proses 2.1 – Penyimpanan Data RSS Baru [sos21]

Proses Permintaan Halaman merupakan proses manajemen data RSS baru yang telah diambil kedalam file dalam S.o.S. RSS Reader.

2. Proses 2.2 – Permintaan Data RSS Baru [sos22]

Proses Menampilkan Berita merupakan proses manajemen data RSS yang akan diambil lewat internet salam S.o.S. RSS Reader.

#### 4.2.4. Pseudocode Sistem Aplikasi Penerima Berita Melalui RSS Pada Nintendo DS

Pseudocode merupakan bahasa tingkat tinggi informal dan ringkasan dari algoritma pemrograman komputer yang menggunakan struktural dari bahasa pemrograman, dan ditujukan untuk manusia.

##### 4.2.4.1. Pseudocode Proses Permintaan Halaman [sos11]

Implementasi algoritma untuk proses permintaan halaman untuk Sistem Aplikasi Penerima Berita Melalui RSS Pada Nintendo DS ditunjukkan sebagai berikut:

1. Proses deklarasi variabel hitung dengan nilai awal 4
2. Proses deklarasi variabel dimenu dengan nilai awal true
3. Lakukan perulangan dengan kondisi selama dimenu memiliki nilai true
4. Jika tombol Down ditekan dan variabel hitung tidak bernilai 6 maka nilai variabel hitung bertambah, bila tidak beri nilai 4 pada variabel hitung
5. Jika tombol Up ditekan dan variabel hitung tidak bernilai 4 maka nilai variabel hitung berkurang, bila tidak beri nilai 6 pada variabel hitung
6. Jika tombol A ditekan, maka variabel dimenu akan bernilai false

Proses permintaan halaman dijelaskan dalam bentuk pseudocode sebagai berikut:

```

1 VARIABLE hitung=4
2 VARIABLE dimenu=true
3 PRINT ON BOTTOM SCREEN AT COORDINATE 3,21 "Tekan A untuk memilih..."
4 PRINT ON BOTTOM SCREEN AT COORDINATE 3,3 "Pilih RSS : "
5 PRINT ON BOTTOM SCREEN AT COORDINATE 3,4 "situs international.okezone"
6 PRINT ON BOTTOM SCREEN AT COORDINATE 3,5 "situs gametrailers"
7 PRINT ON BOTTOM SCREEN AT COORDINATE 3,6 "situs worldcup2010southafrica"
8 WHILE dimenu IS EQUAL TO true
9     PRINT ON BOTTOM SCREEN AT COORDINATE 0,1 " "
10    PRINT ON BOTTOM SCREEN AT COORDINATE 0,1 "-->"
11    PRINT ON BOTTOM SCREEN AT COORDINATE 0,1 " "
12    IF button Down IS PRESSED THEN
13        IF hitung IS NOT EQUAL TO 6 THEN
14            INCREMENT hitung
15        ELSE
16            hitung=4

```

```

17 PRINT ON BOTTOM SCREEN AT COORDINATE 1,6 " "
18 END IF
19 END IF
20 IF button Up IS PRESSED THEN
21 IF hitung IS NOT EQUAL TO 4 THEN
22 DECREMENT hitung
23 ELSE
24 hitung=6
25 PRINT ON BOTTOM SCREEN AT COORDINATE 1,4 " "
26 END IF
27 END IF
28 IF button A IS PRESSED THEN
29 dimenu=false
30 END IF
31 END WHILE

```

#### 4.2.4.2. Pseudocode Proses Menampilkan Berita [sos12]

Implementasi algoritma untuk proses menampilkan berita untuk Sistem Aplikasi Penerima Berita Melalui RSS Pada Nintendo DS ditunjukkan sebagai berikut:

1. Proses deklarasi variable a dengan nilai awal 0
2. Lakukan perulangan selama tombol Start belum ditekan
3. Jika tombol A ditekan dan nilai variabel a lebih kecil daripada variabel kiri yang telah dikurangi 1, maka dilakukan pemanggilan fungsi erase\_screen yang berguna untuk menghapus layar pertama dan variabel a bertambah
4. Jika tombol B ditekan dan nilai variabel a lebih besar daripada 0, maka dilakukan pemanggilan fungsi erase\_screen yang berguna untuk menghapus layar pertama dan variabel a berkurang

Proses menampilkan berita dijelaskan dalam bentuk pseudocode sebagai berikut:

```

1 VARIABLE a=0
2 WHILE button Start IS NOT PRESSED YET
3 PRINT ON BOTTOM SCREEN AT COORDINATE 1,3 "Berita ke VARIABLE a ADD 1
  VARIABLE simpan AT POSITION a"
4 IF button A IS PRESSED AND a IS SMALLER THAN kiri SUBSTRACT BY 1 THEN
5 FUNCTION erase_screen(screen=1)
6 INCREMENT a
7 END IF

```

8	IF button B IS PRESSED AND a IS GREATER THAN 0 THEN
9	FUNCTION erase_screen(screen=1)
10	DECREMENT a
11	END IF
12	END WHILE

#### 4.2.4.3. Pseudocode Proses Permintaan Data RSS [sos13]

Implementasi algoritma untuk proses permintaan data RSS untuk Sistem Aplikasi Penerima Berita Melalui RSS Pada Nintendo DS ditunjukkan sebagai berikut:

1. Bila nilai dari variabel hitung adalah 4, maka dilakukan pemanggilan proses fungsi parser untuk RSS feed international.okezone.com
2. Bila nilai dari variabel hitung adalah 5 maka dilakukan pemanggilan proses fungsi parser untuk RSS feed gametrailers.com
3. Bila nilai dari variabel hitung adalah 6, maka dilakukan pemanggilan proses fungsi parser untuk RSS feed worldcup2010southafrica.com

Proses permintaan data RSS dijelaskan dalam bentuk pseudocode sebagai berikut:

1	CASE hitung OF
2	4 : FUNCTION parser("/xml/ds_okezoneint.xml", "http://sindikasi.okezone.com/index.php/international/RSS2.0")
3	5 : FUNCTION parser ("/xml/ds_gametrailers.xml", "http://www.gametrailers.com/rssgenerate.php?sl=&favplats[ds]=ds& type[review]=on&type[preview]=on&orderby=newest&limit=5")
4	6 : FUNCTION parser("/xml/ds_worldcup2010.xml", "http://www.worldcup2010southafrica.com/component/option,com_rss/feed,R SS2.0/no_html,1/")
5	END CASE

#### 4.2.4.4. Pseudocode Proses Penyimpanan Data RSS Baru [sos21]

Implementasi algoritma untuk proses penyimpanan data RSS baru untuk Sistem Aplikasi Penerima Berita Melalui RSS Pada Nintendo DS ditunjukkan sebagai berikut:

1. Variabel down adalah pemanggilan fungsi myDownloadHTTP dengan argument nama\_xml dan alamat\_xml

Proses penyimpanan data RSS baru dijelaskan dalam bentuk pseudocode sebagai berikut:

1	PRINT ON BOTTOM SCREEN AT COORDINATE 1,3 "Unduh / simpan..."
2	PRINT ON BOTTOM SCREEN AT COORDINATE 1,3 "tunggu."
3	down = FUNCTION myDownloadHTTP(nama_xml, alamat_xml)
4	PRINT ON BOTTOM SCREEN AT COORDINATE 1,3 "selesai!"
5	PRINT ON BOTTOM SCREEN AT COORDINATE 1,3 "VARIABLE down bytes yang tersimpan..."

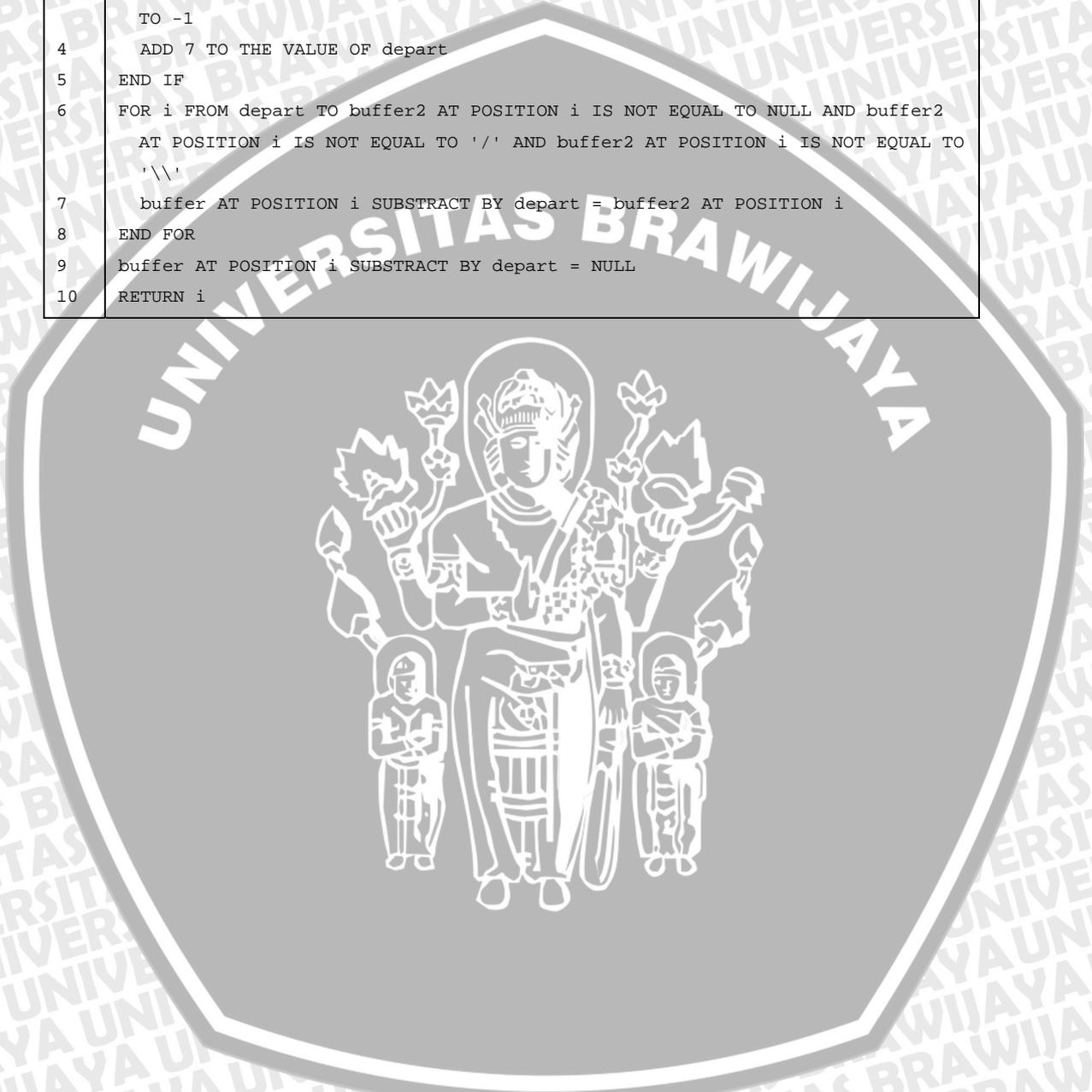
#### 4.2.4.5. Pseudocode Proses Permintaan Data RSS Baru [sos22]

Implementasi algoritma untuk proses permintaan data RSS baru untuk Sistem Aplikasi Penerima Berita Melalui RSS Pada Nintendo DS ditunjukkan sebagai berikut:

1. Proses deklarasi fungsi `get_HTTP_serveur` dengan variabel `buffer` dan `buffer2`
2. Proses deklarasi variabel `i` dan variabel `depart` dengan nilai awal 0
3. Jika proses pemanggilan fungsi `search_word` dengan variabel `mot1` bernilai `buffer2`; variabel `mot2` bernilai `"http://"`; dan variabel `depart` bernilai 0 tidak sama dengan -1, maka tambahkan nilai 7 pada variabel `depart`
4. Untuk variabel `i` bernilai dari nilai variabel `depart` sampai variabel `buffer2` dengan array ke-`i` tidak sama dengan NULL dan variabel `buffer2` dengan array ke-`i` tidak sama dengan `'/'` dan variabel `buffer2` dengan array ke-`i` tidak sama dengan `'\'`, maka variabel `buffer` dengan array ke- `i` dikurangi oleh nilai dari variabel `depart` memiliki nilai dari variabel `buffer2` array ke-`i`
5. Variabel `buffer` dengan array ke- `i` dikurangi nilai dari variabel `depart` diberi nilai NULL
6. Nilai dari variabel `i` adalah nilai kembalian untuk fungsi `get_HTTP_serveur`

Proses permintaan data RSS baru dijelaskan dalam bentuk pseudocode sebagai berikut:

```
1  FUNCTION get_HTTP_serveur(VARIABLE buffer, VARIABLE buffer2)
2  VARIABLE i, depart=0
3  IF FUNCTION search_word(mot1=buffer2, mot2="http:// ", depart=0) IS NOT EQUAL
4  TO -1
5  ADD 7 TO THE VALUE OF depart
6  END IF
7  FOR i FROM depart TO buffer2 AT POSITION i IS NOT EQUAL TO NULL AND buffer2
8  AT POSITION i IS NOT EQUAL TO '/' AND buffer2 AT POSITION i IS NOT EQUAL TO
9  '\\\
10 buffer AT POSITION i SUBSTRACT BY depart = buffer2 AT POSITION i
11 END FOR
12 buffer AT POSITION i SUBSTRACT BY depart = NULL
13 RETURN i
```

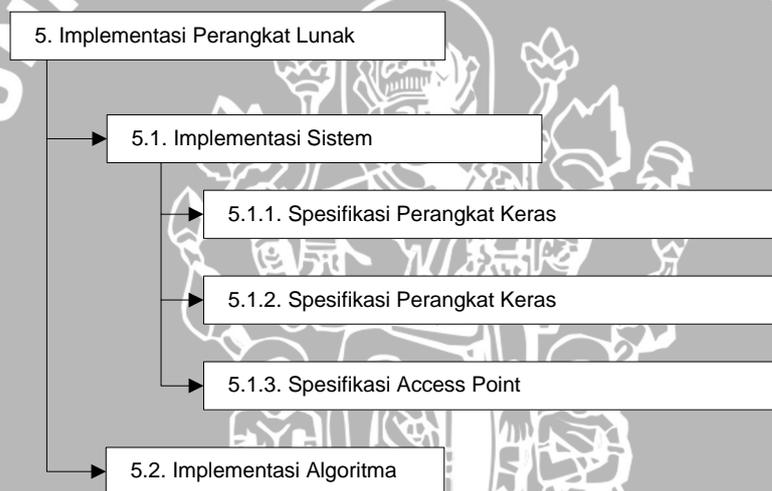


## BAB V

### IMPLEMENTASI PERANGKAT LUNAK

Bab ini membahas tentang implementasi perangkat lunak Sistem Aplikasi Penerima Berita Melalui RSS Pada Nintendo DS berdasarkan hasil yang telah didapatkan dari analisis kebutuhan dan proses perancangan perangkat lunak. Bab ini terdiri dari pembahasan implementasi sistem, batasan-batasan implementasi, implementasi algoritma dan implementasi antarmuka.

Gambar 5.1. menunjukkan langkah-langkah proses implementasi yang digunakan pada bab ini.



Gambar 5.1. Diagram Pohon Implementasi Sistem

Sumber: [Implementasi]

#### 5.1. Implementasi Sistem

Hasil analisis kebutuhan dan perancangan yang telah diuraikan pada Bab 4 diimplementasikan menjadi sebuah sistem yang nyata agar bisa berfungsi sesuai dengan kebutuhan. Sistem yang dibuat akan diimplementasikan pada perangkat keras dan perangkat lunak dengan spesifikasi tertentu serta konfigurasi jaringan tertentu.

### 5.1.1. Spesifikasi Perangkat Keras

Untuk mengembangkan Sistem Aplikasi Penerima Berita Melalui RSS Pada Nintendo DS, digunakan sebuah komputer dengan spesifikasi perangkat keras seperti terlihat pada tabel 5.1.

Tabel 5.1. Spesifikasi Perangkat Keras Komputer

Nama Komponen	Spesifikasi
<i>Processor</i>	Pentium III 833 MHz
<i>Memory</i>	SDRAM 384 MB
<i>Hard Disk</i>	20GB
<i>Motherboard</i>	Intel PW82815
<i>VGA Card</i>	Nvidia GeForce 5500 FX

Sumber: [Implementasi]

Setelah proses pengembangan sistem, maka hasil pengembangan tersebut diuji dan dijalankan pada Nintendo DS dengan spesifikasi seperti terlihat pada tabel 5.2.

Tabel 5.2. Spesifikasi Nintendo DS

Nama Komponen	Spesifikasi
<i>Jenis Nintendo DS</i>	Nintendo DS Lite
<i>DS Slot-1 Flash Cart</i>	DS Top Toy (DSTT)
<i>Media Penyimpan</i>	MicroSD 4GB

Sumber: [Implementasi]

### 5.1.2. Spesifikasi Perangkat Lunak

Untuk mengembangkan aplikasi Sistem Aplikasi Penerima Berita Melalui RSS Pada Nintendo DS, digunakan software dengan spesifikasi seperti terlihat pada Tabel 5.3.

Tabel 5.3. Spesifikasi Perangkat Lunak

Perangkat Lunak	Spesifikasi
<i>Sistem Operasi</i>	Windows XP SP2
<i>Bahasa Pemrograman</i>	C++
<i>Lingkungan Pemrograman</i>	DevkitPro 1.5.0 PALib 080823
<i>IDE (Integrated Development Environment)</i>	Dev-C++ 4.9.9.2

Sumber: [Implementasi]

### 5.1.3. Spesifikasi Access Point

Sistem Aplikasi Penerima Berita Melalui RSS Pada Nintendo DS ini diimplementasikan pada sebuah access point dengan spesifikasi terlihat pada Tabel 5.4.

Tabel 5.4. Spesifikasi Access Point

Spesifikasi Access Point	
<i>Merek</i>	Linksys WRT54GL
<i>Versi</i>	1.1
<i>CPU Speed</i>	200MHz
<i>Flash Memory</i>	4MB
<i>S/N Prefix</i>	CL7B / CL7C

Sumber: [Implementasi]

## 5.2. Implementasi Algoritma

Subbab 5.2 membahas semua implementasi algoritma yang didapat dari data flow diagram levelled pada Bab 4. Subbab ini membahas 5 implementasi algoritma yang didapat dari data flow diagram levelled pada Bab 4.

### 5.2.1. Implementasi Kebutuhan Fungsional Inisialisasi

Sebelum Sistem Aplikasi Penerima Berita Melalui RSS Pada Nintendo DS dapat digunakan, ada 3 fitur yang harus diaktifkan terlebih dahulu, yaitu:

1. fatInitDefault

Mengaktifkan fitur untuk penulisan dan pembacaan data pada media penyimpanan yang memiliki format FAT.

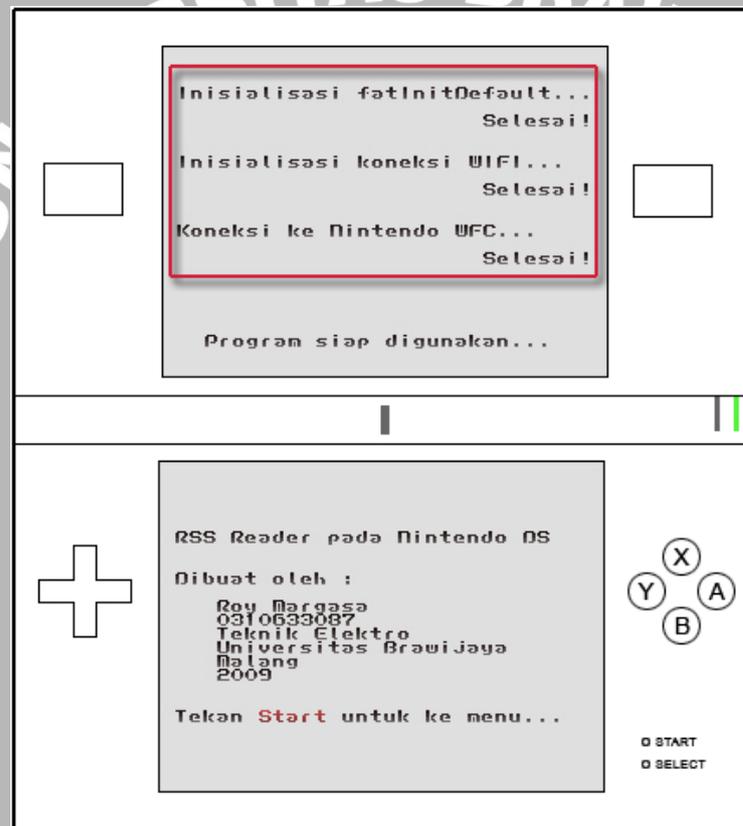
2. PA\_InitWifi

Mengaktifkan fitur wifi yang dimiliki oleh Nintendo DS.

3. PA\_ConnectWifiWFC

Mengaktifkan koneksi ke Nintendo Wi-Fi Connection Service.

Gambar 5.2. adalah implementasi kebutuhan fungsional inisialisasi.



Gambar 5.2. Hasil Implementasi Inisialisasi

Sumber: [Implementasi]

Implementasi untuk inisialisasi ditunjukkan sebagai berikut:

```

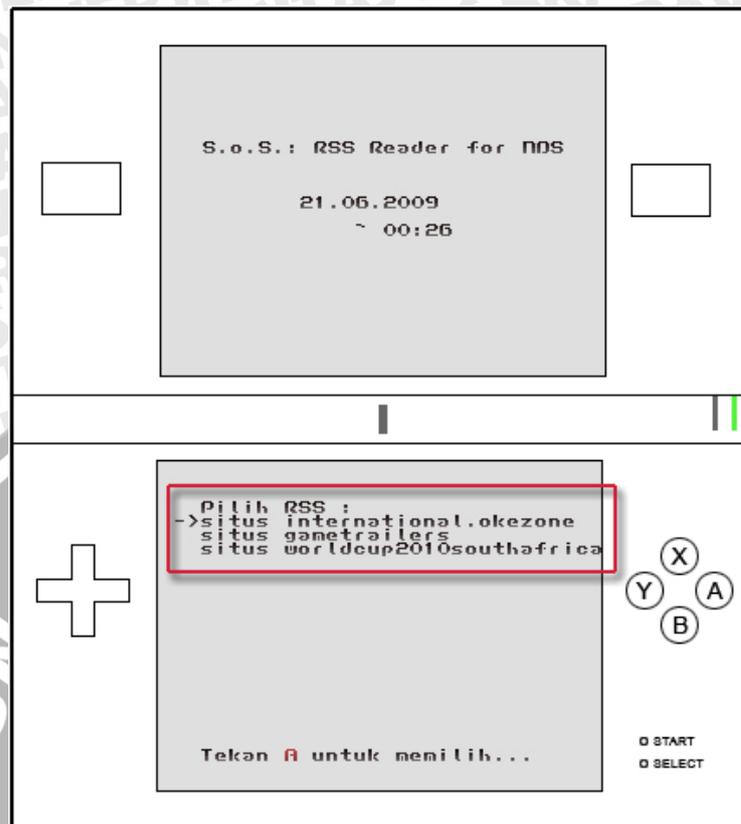
1 void inisialisasi()
2 {
3     //inisialisasi fat
4     PA_OutputSimpleText(1, 1, 3, "Inisialisasi fatInitDefault...");
5     fatInitDefault();

```

```
6 PA_OutputSimpleText(1, 23, 5, "Selesai!");
7
8 //inisialisasi WIFI Connection
9 PA_OutputSimpleText(1, 1, 8, "Enable WIFI device...");
10 PA_InitWifi();
11 PA_OutputSimpleText(1, 23, 10, "Selesai!");
12
13 //Connect to the Nintendo WFC Access Point.
14 PA_OutputSimpleText(1, 1, 13, "Inisialisasi koneksi WIFI...");
15 PA_ConnectWifiWFC();
16 if (PA_ConnectWifiWFC()==TRUE)
17 {
18     PA_OutputSimpleText(1, 23, 15, "Selesai!");
19     PA_OutputSimpleText(1, 3, 21, "Program siap digunakan...");
20 }
21 else
22 {
23     PA_OutputSimpleText(1, 23, 15, "Gagal...");
24     PA_OutputSimpleText(1, 2, 21, "Koneksi WIFI tidak tersedia");
25     while(1)
26     {
27         PA_WaitForVBL();
28     }
29 }
30 }
```

### 5.2.2. Implementasi Kebutuhan Fungsional Permintaan Halaman [sos11]

Implementasi kebutuhan fungsional permintaan halaman ditunjukkan oleh gambar 5.3. User dapat memilih RSS feed yang diinginkan pada ke-3 menu pilihan.



Gambar 5.3. Hasil Implementasi Algoritma Untuk Permintaan Halaman

Sumber: [Implementasi]

Implementasi untuk permintaan halaman ditunjukkan sebagai berikut:

```

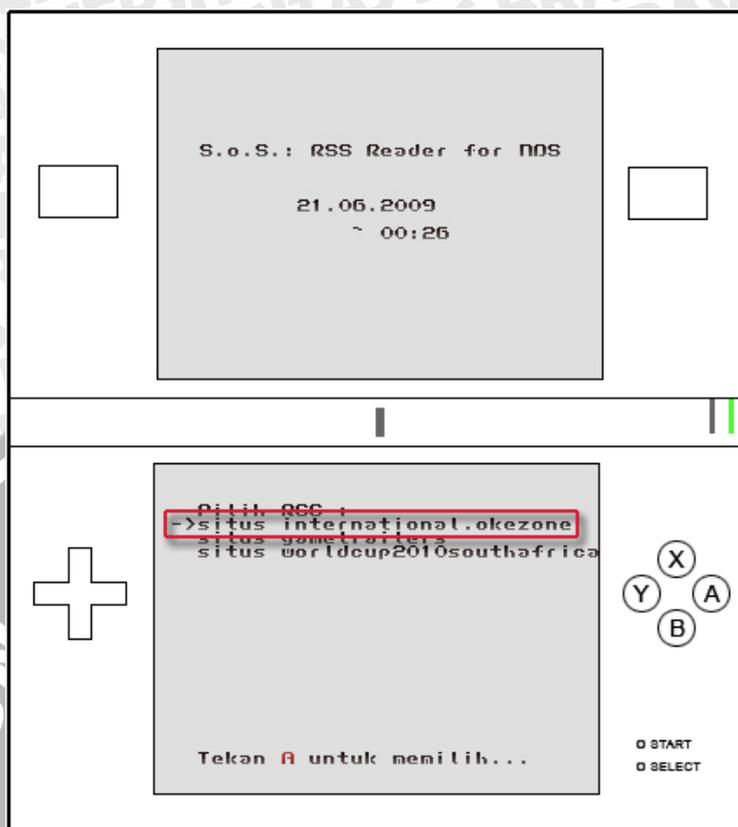
1  int hitung=4;
2  bool dimenu = true;
3
4  PA_OutputText(0, 3, 21, "Tekan %c1A%c0 untuk memilih...");
5
6  PA_OutputSimpleText(0, 3, 3, "Pilih RSS : ");
7  PA_OutputSimpleText(0, 3, 4, "situation international.okezone");
8  PA_OutputSimpleText(0, 3, 5, "situation gametrailers");
9  PA_OutputSimpleText(0, 3, 6, "situation worldcup2010southafrica");
10
11 while(dimenu==true)
12 {
13     PA_OutputSimpleText(0, 1, hitung-1, " ");
14     PA_OutputSimpleText(0, 1, hitung, "->");
15     PA_OutputSimpleText(0, 1, hitung+1, " ");
16
17     if (Pad.Newpress.Down)
18     {
19         if (hitung!=6)

```

```
20 {
21     hitung++;
22 }
23 else
24 {
25     hitung=4;
26     PA_OutputSimpleText(0, 1, 6, " ");
27 }
28 }
29
30 if (Pad.Newpress.Up)
31 {
32     if (hitung!=4)
33     {
34         hitung--;
35     }
36
37     else
38     {
39         hitung=6;
40         PA_OutputSimpleText(0, 1, 4, " ");
41     }
42
43     if (Pad.Newpress.A)
44     {
45         dimenu = false;
46     }
47
48     PA_WaitForVBL();
49 }
```

### 5.2.3. Implementasi Kebutuhan Fungsional Permintaan Data RSS [sos13]

Implementasi kebutuhan fungsional permintaan data RSS ditunjukkan oleh gambar 5.4. Digunakan untuk meminta RSS yang telah dipilih pada halaman menu[sos11] oleh user ke sistem.



Gambar 5.4. Hasil Implementasi untuk Permintaan Data RSS

Sumber: [Implementasi]

Implementasi untuk permintaan data RSS ditunjukkan sebagai berikut:

```

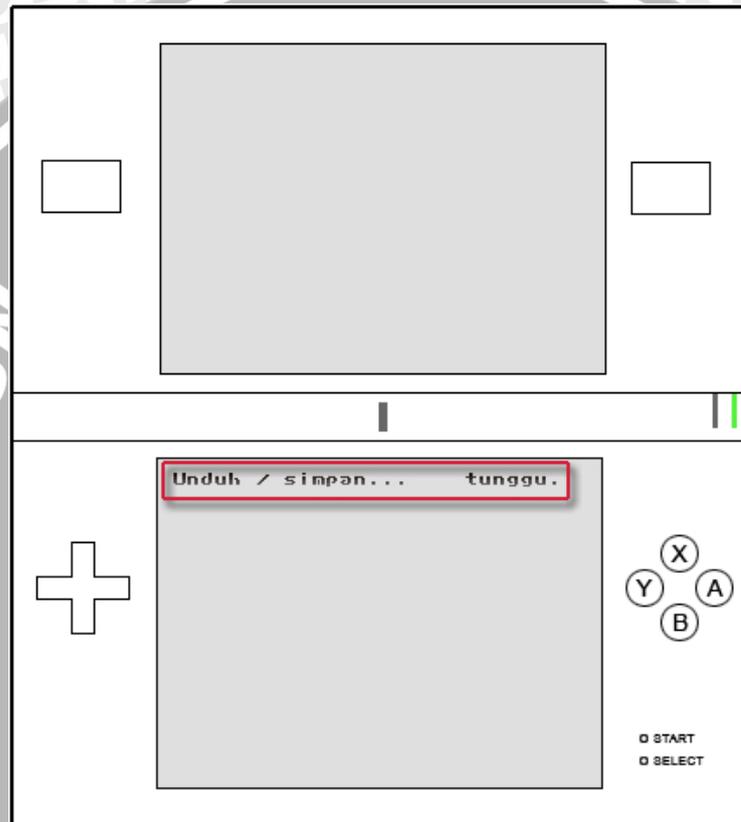
1  switch(hitung)
2  {
3      case 4 : parser("/xml/ds_okezoneint.xml",
4                  "http://sindikasi.okezone.com/index.php/international/RSS
5                  2.0"); break;
6      case 5 : parser("/xml/ds_gametrailers.xml",
7                  "http://www.gametrailers.com/rssgenerate.php?sl=&favplats
8                  [ds]=ds&
9                  type[review]=on&type[preview]=on&orderby=newest&limit=5")
10                 ; break;
11     case 6 : parser("/xml/ds_worldcup2010.xml",
12                 "http://www.worldcup2010southafrica.com/component/option,
13                 com_rss/feed,RSS2.0/no_html,1/"); break;

```

#### 5.2.4. Implementasi Kebutuhan Fungsional Permintaan Data RSS Baru

[sos22]

Implementasi kebutuhan fungsional permintaan data RSS baru ditunjukkan oleh gambar 5.5. Digunakan untuk meminta data pada RSS feed yang dipilih.



Gambar 5.5. Hasil Implementasi untuk Permintaan Data RSS Baru

Sumber: [Implementasi]

Implementasi untuk permintaan data RSS baru ditunjukkan sebagai berikut:

```

1  int get_HTTP_serveur(char *buffer, char *buffer2)
2  {
3      int i,depart=0;
4      if(search_word(buffer2,"http://",0)!=-1)
5          depart+=7;
6      for(i = depart; buffer2[i] != '\0' && buffer2[i] != '/' && buffer2[i] !=
7          '\\'; i++)
8      {
9          buffer[i-depart]=buffer2[i];
10     }

```

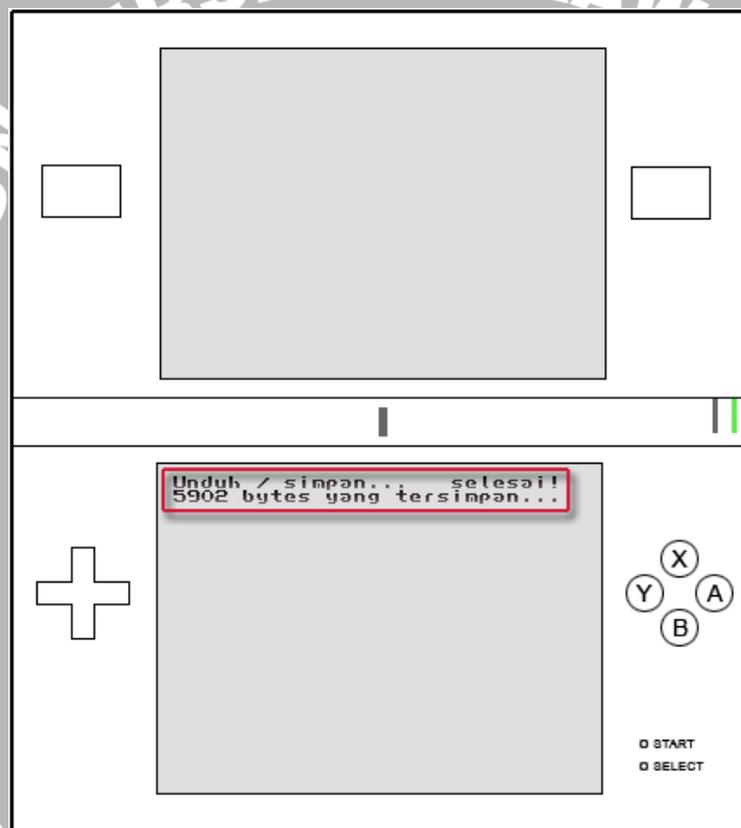
```

11     buffer[i-depart]='\0';
12     return i;}

```

### 5.2.5. Implementasi Kebutuhan Fungsional Penyimpanan Data RSS Baru [sos21]

Implementasi kebutuhan fungsional penyimpanan data RSS baru ditunjukkan oleh gambar 5.6. Digunakan untuk menyimpan RSS feed yang dipilih oleh user. RSS feed akan disimpan sebagai file dengan ekstensi XML dalam folder xml.



Gambar 5.6. Hasil Implementasi untuk Penyimpanan Data RSS Baru

Sumber: [Implementasi]

Implementasi untuk penyimpanan data RSS baru ditunjukkan sebagai berikut:

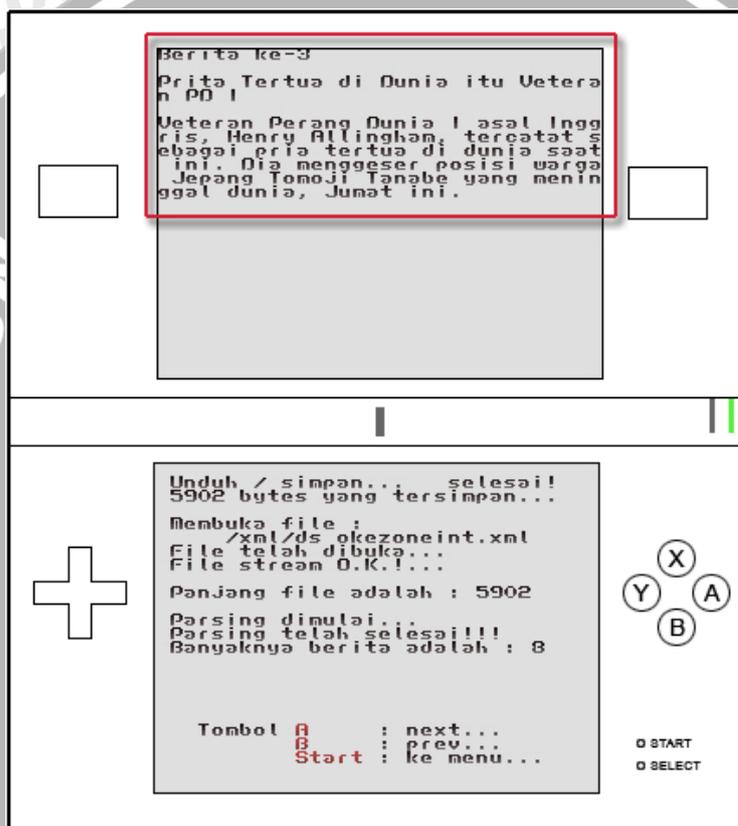
```

1     PA_OutputSimpleText(0, 1, 1, "Unduh / simpan...");
2     PA_OutputText(0, 21, 1, " tunggu.");
3     down = myDownloadHTTP(nama_xml, alamat_xml);
4     PA_OutputText(0, 21, 1, "selesai!");
5     PA_OutputText(0, 1, 2, "%d bytes yang tersimpan...", down);

```

### 5.2.6. Implementasi Kebutuhan Fungsional Menampilkan Berita [sos12]

Implementasi kebutuhan fungsional menampilkan berita ditunjukkan oleh gambar 5.7. User akan melihat berita yang telah diparsing pada layar tampilan Nintendo DS. Bila berita pada RSS feed yang telah dipilih, user dapat melihat berita selanjutnya dengan menekan tombol A, dan tombol B untuk melihat berita sebelumnya.



Gambar 5.7. Hasil Implementasi Untuk Menampilkan Berita

Sumber: [Implementasi]

Implementasi algoritma untuk menampilkan berita ditunjukkan sebagai berikut:

```

1  a=0;
2
3  while (Pad.Newpress.Start!=1)
4  {
5      PA_OutputText(1, 0, 0, "Berita ke-%d\n\n%s", a+1, simpan[a]);
6
7      if (Pad.Newpress.A && a<kiri-1)

```

```
8 {
9     erase_screen(1);
10    a++;
11 }
12
13 if (Pad.Newpress.B && a>0)
14 {
15     erase_screen(1);
16     a--;
17 }
18
19 PA_WaitForVBL();
20 }
```

UNIVERSITAS BRAWIJAYA



## BAB VI

### PENGUJIAN DAN ANALISIS PERANGKAT LUNAK

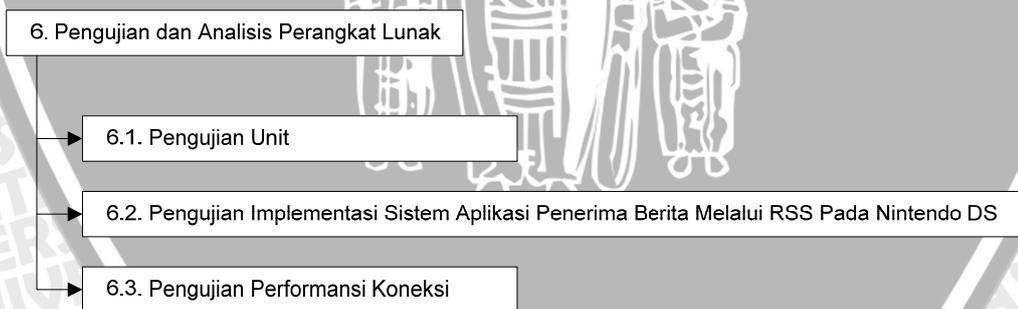
Bab ini membahas mengenai pengujian dan analisis terhadap implementasi Sistem Aplikasi Penerima Berita Melalui RSS Pada Nintendo DS. Pengujian yang dilakukan meliputi pengujian implementasi aplikasi sistem secara online.

Pengujian dilakukan untuk mengetahui apakah implementasi Sistem Aplikasi Penerima Berita Melalui RSS Pada Nintendo DS dapat dijalankan secara online. Pengujian dilakukan dengan menggunakan DS Slot-1 Flash Cart merek DSTT dengan kernel v1.17. User harus berada pada jangkuan access point yang tersedia.

Spesifikasi dan konfigurasi Sistem Aplikasi Penerima Berita Melalui RSS Pada Nintendo DS adalah sebagai berikut:

1. Nintendo DS Lite
2. DS Slot-1 Flash Cart merek DSTT
3. Micro SD

Gambar 6.1. menunjukkan langkah-langkah proses implementasi yang digunakan pada bab ini.



Gambar 6.1. Diagram Pohon Pengujian dan Analisis Sistem

Sumber: [Pengujian]

#### 6.1. Pengujian Unit

Pada pengujian unit Sistem Aplikasi Penerima Berita Melalui RSS Pada Nintendo DS ini, digunakan teknik pengujian White Box (White Box Testing) dengan teknik Basic Path Testing. Pada teknik Basic Path Testing, proses

pengujian dilakukan dengan memodelkan algoritma pada suatu flow graph, menentukan jumlah kompleksitas siklomatis (cyclomatic complexity), menentukan sebuah basis set dari jalur independen dan memberikan kasus uji (test case) pada setiap basis set yang telah ditentukan. Penulisan laporan skripsi ini hanya dicantumkan hasil pengujian unit untuk algoritma dari satu metode (operasi) saja (tidak untuk keseluruhan metode).

**6.1.1. Pengujian Unit Untuk Proses Permintaan Halaman [sos11]**



Gambar 6.2. Pemodelan Algoritma Permintaan Halaman ke Dalam Flow Graph

Sumber: [Pengujian]



Pemodelan ke dalam flow graph yang telah dilakukan terhadap permintaan halaman menghasilkan jumlah kompleksitas siklomatis (cyclomatic complexity) melalui persamaan  $V(G) = E - N + 2$ , dimana  $V(G)$  merupakan jumlah kompleksitas siklomatis,  $E$  merupakan sisi (garis penghubung antar node), dan  $N$  merupakan jumlah simpul (node).

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 17 - 14 + 2 \\ &= 5 \end{aligned}$$

Dari nilai cyclomatic complexity yang telah dihasilkan dari perhitungan yaitu ditentukan lima buah basis set dari jalur independent yaitu:

- Jalur 1 : 1-2-3-4-5-6-8-12-3
- Jalur 2 : 1-2-3-4-7-8-12-3-14
- Jalur 3 : 1-2-3-4-8-9-10-12-3-14
- Jalur 4 : 1-2-3-4-11-12-3-14
- Jalur 5 : 1-2-3-4-8-12-13-3-14

Penentuan kasus uji untuk masing-masing jalur dan hasil eksekusi untuk masing-masing kasus uji dijelaskan pada tabel 6.1.

Tabel 6.1. Test Case Untuk Pengujian Permintaan Halaman

Jalur	Kasus uji	Hasil yang diharapkan	Hasil yang didapat
1	Tombol Down ditekan	Panah berpindah ke bawah	Panah berpindah ke bawah
2	Tombol Down ditekan pada saat panah sudah pada akhir pilihan	Panah kembali ke atas	Panah kembali ke atas
3	Tombol Up ditekan	Panah berpindah ke atas	Panah berpindah ke atas
4	Tombol Up ditekan pada saat panah sudah berada pada awal pilihan	Panah kembali ke bawah	Panah kembali ke bawah
5	Tombol A ditekan	Memilih RSS feed yang ditunjuk oleh panah	Memilih RSS feed yang ditunjuk oleh panah

Sumber: [Pengujian]



### 6.1.2. Pengujian Unit Untuk Proses Permintaan Data RSS [sos13]



Gambar 6.3. Pemodelan Algoritma Permintaan Data RSS ke Dalam Flow Graph

Sumber: [Pengujian]

Pemodelan ke dalam flow graph yang telah dilakukan terhadap permintaan data RSS menghasilkan jumlah kompleksitas siklomatis (cyclomatic complexity) melalui persamaan  $V(G) = E - N + 2$ , dimana  $V(G)$  merupakan jumlah kompleksitas siklomatis,  $E$  merupakan sisi (garis penghubung antar node), dan  $N$  merupakan jumlah simpul (node).

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 6 - 5 + 2 \\ &= 3 \end{aligned}$$

Dari nilai cyclomatic complexity yang telah dihasilkan dari perhitungan yaitu ditentukan tiga buah basis set dari jalur independent yaitu:

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

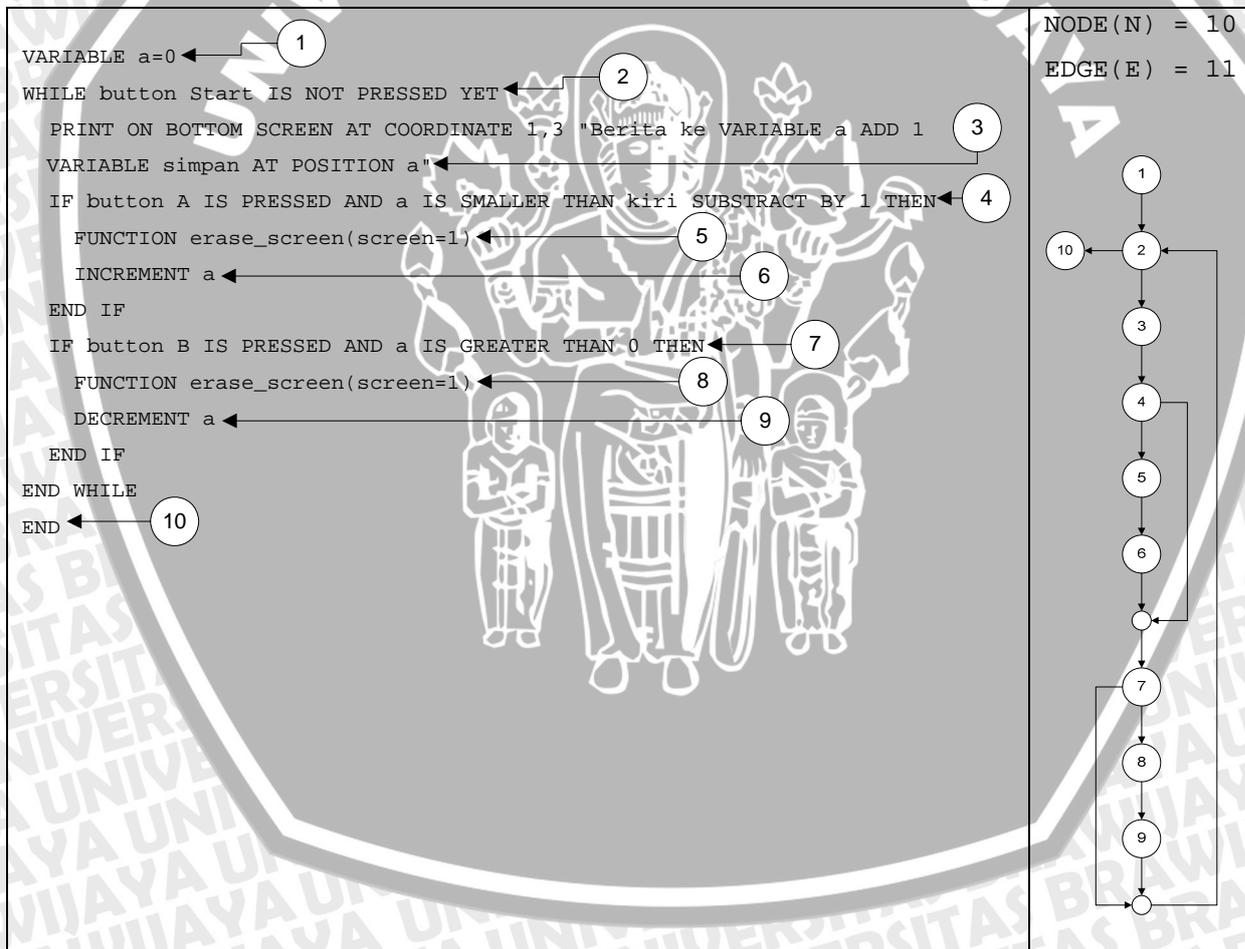
Penentuan kasus uji untuk masing-masing jalur dan hasil eksekusi untuk masing-masing kasus uji dijelaskan pada tabel 6.2.

Tabel 6.2. Test Case Untuk Pengujian Permintaan Data RSS

Jalur	Kasus uji	Hasil yang diharapkan	Hasil yang didapat
1	RSS feed international.okezone.com dipilih	Sistem mengirim permintaan RSS feed untuk international.okezone.com	Sistem mengirim permintaan RSS feed untuk international.okezone.com
2	RSS feed gametrailers.com bagian NDS dipilih	Sistem mengirim permintaan RSS feed untuk gametrailers.com bagian NDS	Sistem mengirim permintaan RSS feed untuk gametrailers.com bagian NDS
3	RSS feed worldcup2010southafrica.com dipilih	Sistem mengirim permintaan RSS feed untuk worldcup2010southafrica.com	Sistem mengirim permintaan RSS feed untuk worldcup2010southafrica.com

Sumber: [Pengujian]

6.1.3. Pengujian Unit Untuk Proses Menampilkan Berita [sos12]



Gambar 6.4. Pemodelan Algoritma Menampilkan Berita ke Dalam Flow Graph

Sumber: [Pengujian]



Pemodelan ke dalam flow graph yang telah dilakukan terhadap menampilkan berita menghasilkan jumlah kompleksitas siklomatis (cyclomatic complexity) melalui persamaan  $V(G) = E - N + 2$ , dimana  $V(G)$  merupakan jumlah kompleksitas siklomatis,  $E$  merupakan sisi (garis penghubung antar node), dan  $N$  merupakan jumlah simpul (node).

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 11 - 10 + 2 \\ &= 3 \end{aligned}$$

Dari nilai cyclomatic complexity yang telah dihasilkan dari perhitungan yaitu ditentukan lima buah basis set dari jalur independent yaitu:

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

Jalur 2 : 1-2-3-4-7-8-9-10

Jalur 3 : 1-2-3-4-7-2-10

Penentuan kasus uji untuk masing-masing jalur dan hasil eksekusi untuk masing-masing kasus uji dijelaskan pada tabel 6.3.

Tabel 6.3. Test Case Untuk Pengujian Menampilkan Berita

Jalur	Kasus uji	Hasil yang diharapkan	Hasil yang didapat
1	Tombol A ditekan	Berita berikutnya ditampilkan	Berita berikutnya ditampilkan
2	Tombol B ditekan	Berita sebelumnya ditampilkan	Berita sebelumnya ditampilkan
3	Tombol Start ditekan	Kembali ke halaman menu	Kembali ke halaman menu

Sumber: [Pengujian]

## 6.2. Pengujian Implementasi Sistem Aplikasi Penerima Berita Melalui RSS Pada Nintendo DS

Pengujian Implementasi Sistem Aplikasi Penerima Berita Melalui RSS Pada Nintendo DS untuk mengetahui apakah sistem yang dibangun sudah benar sesuai dengan yang dibutuhkan. Kebutuhan-kebutuhan yang telah dirumuskan dalam daftar kebutuhan dan merupakan hasil analisis kebutuhan akan menjadi acuan untuk melakukan pengujian implementasi. Pengujian implementasi tidak dikonsentrasikan pada alur algoritma program tetapi lebih ditekankan untuk menemukan kesesuaian antara sistem yang dibangun dengan daftar kebutuhan.

### 6.2.1. Pengujian Implementasi Inisialisasi

#### A. Tujuan

Pengujian dilakukan untuk mengetahui apakah sistem dapat menunjukkan bahwa inisialisasi sedang berjalan. User juga dapat melihat apakah inisialisasi telah berhasil atau tidak.

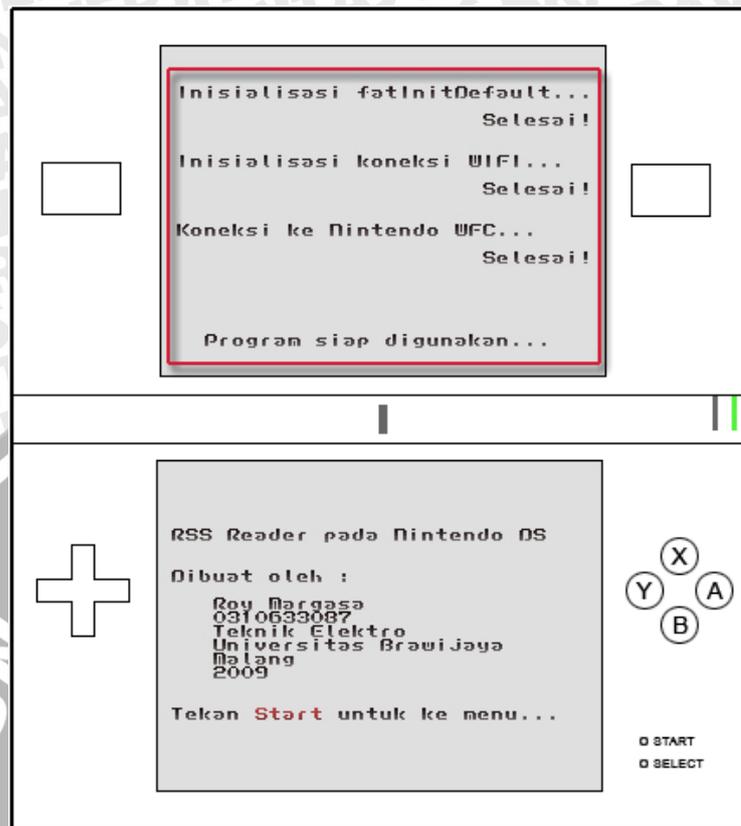
#### B. Prosedur Pengujian

1. Menjalankan program dengan tanpa mengatur konektifitas ke sebuah access point.
2. Menjalankan program dengan sudah mengatur konektifitas ke sebuah access point.

#### C. Hasil Pengujian dan Analisis

1. Tampilan implementasi inisialisasi dengan tanpa mengatur konektifitas ke sebuah access point memiliki hasil tampilan yang ditunjukkan pada gambar 6.5.

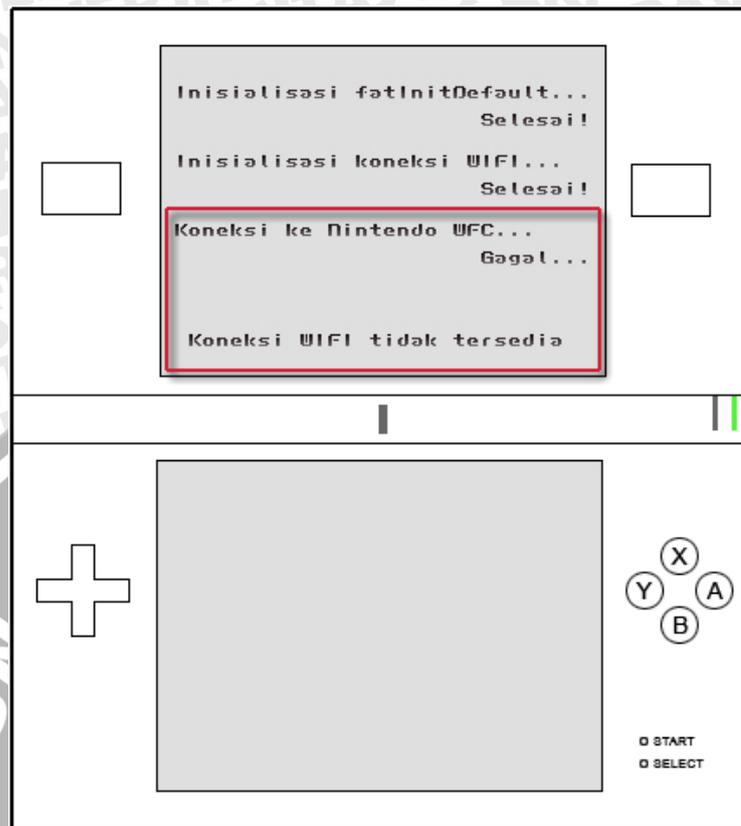




Gambar 6.5. Tampilan Implementasi Inisialisasi Yang Selesai

Sumber: [Pengujian]

2. Tampilan implementasi inisialisasi dengan sudah mengatur konektifitas ke sebuah access point memiliki hasil tampilan yang ditunjukkan pada gambar 6.6.



Gambar 6.6. Tampilan Implementasi Inisialisasi Yang Gagal

Sumber: [Pengujian]

Aplikasi inisialisasi dapat digunakan untuk menampilkan hasil dari inisialisasi baik berhasil maupun gagal.

### 6.2.2. Pengujian Implementasi Permintaan Halaman [sos11]

#### A. Tujuan

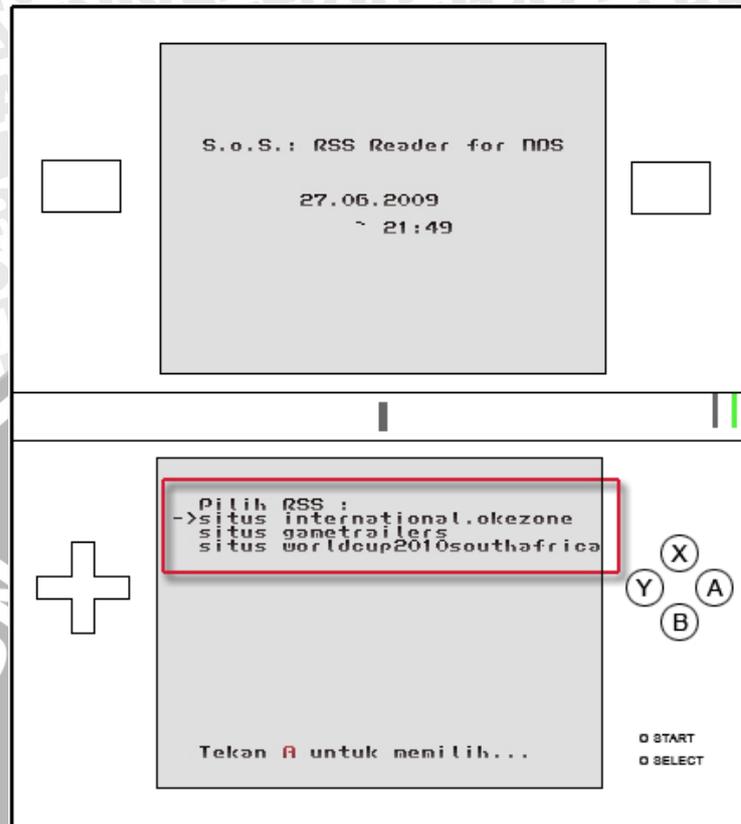
Pengujian dilakukan untuk mengetahui apakah sistem dapat menampilkan halaman menu untuk memilih RSS feed.

#### B. Prosedur Pengujian

Proses tampilan memilih menu RSS feed dilakukan setelah proses inisialisasi telah berhasil dan dilanjutkan dengan menekan tombol Start.

#### C. Hasil Pengujian dan Analisis

Tampilan halaman menu RSS feed ditampilkan pada gambar 6.7.



Gambar 6.7. Tampilan Implementasi Permintaan Halaman

Sumber: [Pengujian]

Berdasarkan pengujian, aplikasi permintaan halaman dapat digunakan untuk menampilkan halaman menu yang akan dipakai oleh user untuk memilih RSS feed yang diinginkan.

### 6.2.3. Pengujian Implementasi Permintaan Data RSS [sos13]

#### A. Tujuan

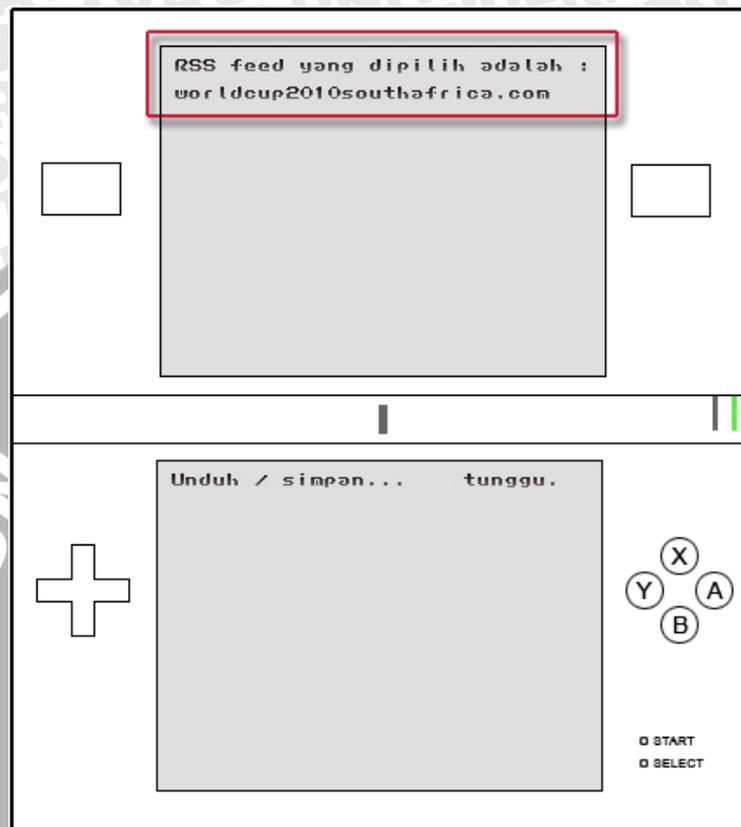
Pengujian dilakukan untuk mengetahui apakah sistem dapat menerima masukkan dari RSS feed yang dipilih oleh user.

#### B. Prosedur Pengujian

User memilih RSS feed yang diinginkan dan melakukan konfirmasi dengan menekan tombol A.

### C. Hasil Pengujian dan Analisis

Tampilan pengujian permintaan data RSS ditampilkan pada gambar 6.8.



Gambar 6.8. Tampilan Implementasi Permintaan Data RSS

Sumber: [Pengujian]

Berdasarkan pengujian, bisa disimpulkan bahwa aplikasi permintaan data RSS dapat digunakan untuk menerima masukan dari RSS feed yang dipilih oleh user.

#### 6.2.4. Pengujian Implementasi Permintaan Data RSS Baru [sos22]

##### A. Tujuan

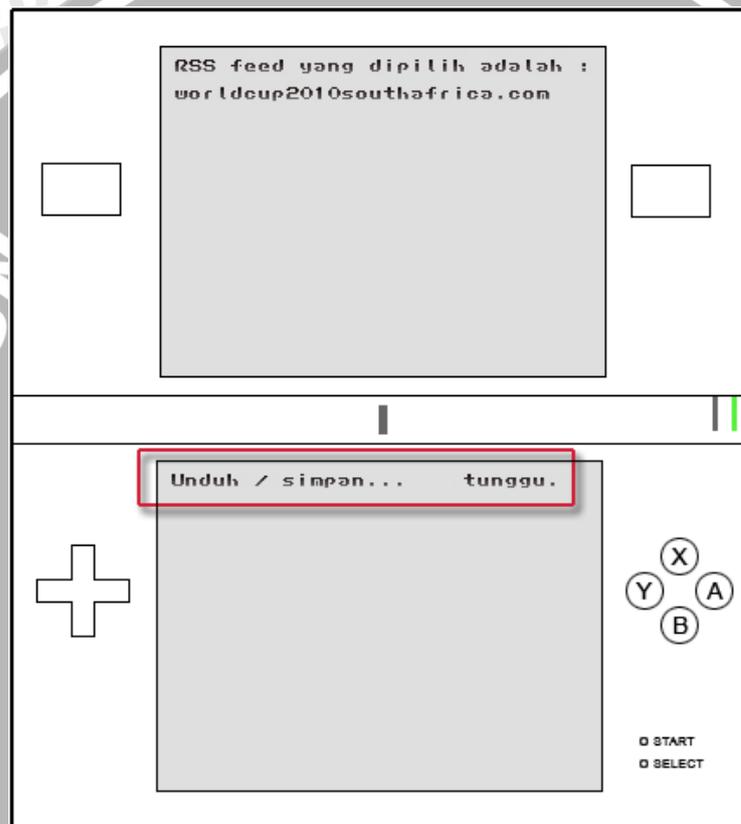
Pengujian dilakukan untuk mengetahui apakah sistem dapat meminta RSS feed yang berada pada internet.

## B. Prosedur Pengujian

Setelah user mengkonfirmasi RSS feed yang dipilih, sistem akan memunculkan tampilan bahwa RSS feed sedang diambil.

## C. Hasil Pengujian dan Analisis

Tampilan permintaan data RSS baru ditampilkan pada gambar 6.9.



Gambar 6.9. Tampilan Implementasi Permintaan Data RSS Baru

Sumber: [Pengujian]

Berdasarkan pengujian, bisa disimpulkan bahwa aplikasi permintaan data RSS baru dapat digunakan untuk meminta RSS pada RSS feed yang berada di internet.

### 6.2.5. Pengujian Implementasi Penyimpanan Data RSS Baru [sos21]

#### A. Tujuan

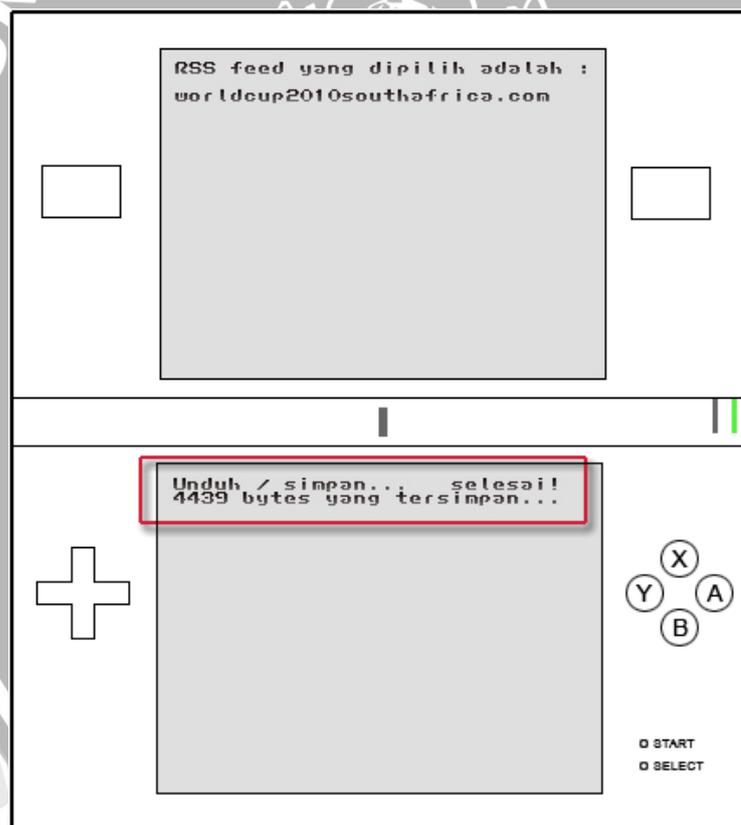
Pengujian dilakukan untuk mengetahui apakah sistem dapat menyimpan data RSS yang telah diambil dari RSS feed yang berada di internet.

#### B. Prosedur Pengujian

Setelah proses permintaan data RSS baru berjalan, sistem akan menyimpan RSS baru yang telah diambil dalam format XML.

#### C. Hasil Pengujian dan Analisis

Tampilan penyimpanan data RSS baru ditampilkan pada gambar 6.10.



Gambar 6.10. Tampilan Implementasi Penyimpanan Data RSS Baru

Sumber: [Pengujian]

Hasil data RSS yang telah disimpan dalam format XML dapat dilihat pada gambar 6.11.



File Name	Size	Type	Modified
ds_gametrainers.xml	7 KB	XML Document	6/26/2009 2:50 PM
ds_okezoneint.xml	6 KB	XML Document	6/26/2009 2:33 PM
ds_worldcup2010.xml	5 KB	XML Document	6/27/2009 9:49 PM

Gambar 6.11. Tampilan Data RSS Yang Telah Tersimpan

Sumber: [Pengujian]

Berdasarkan pengujian, bisa disimpulkan bahwa aplikasi penyimpanan data RSS baru dapat digunakan untuk menyimpan RSS yang telah diambil.

## 6.2.6. Pengujian Implementasi Menampilkan Berita [sos12]

### A. Tujuan

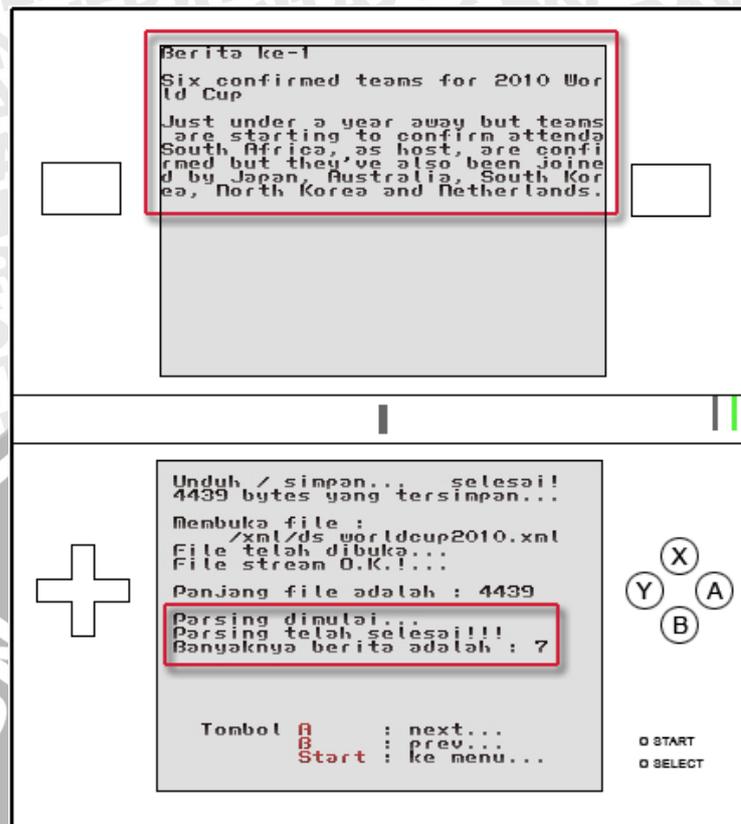
Pengujian dilakukan untuk mengetahui apakah sistem dapat menampilkan berita RSS yang telah diparsing oleh Sistem Aplikasi Penerima Berita Melalui RSS Pada Nintendo DS.

### B. Prosedur Pengujian

Setelah proses penyimpanan data RSS baru, sistem akan membaca file XML yang telah tersimpan dan memparsingnya.

### C. Hasil Pengujian dan Analisis

Tampilan menampilkan berita ditampilkan pada gambar 6.12.



Gambar 6.12. Tampilan Implementasi Menampilkan Berita  
Sumber: [Pengujian]

Berdasarkan pengujian, bisa disimpulkan bahwa aplikasi menampilkan berita dapat digunakan untuk menampilkan data RSS yang telah diparsing.

### 6.3. Pengujian Performansi Koneksi

Pengujian Performansi koneksi dilakukan untuk mengetahui performansi koneksi antara Nintendo DS dengan internet. Pengujian performansi koneksi terdiri dari pengujian koneksi permintaan data RSS baru, dan pengujian koneksi ke Nintendo WFC.

#### 6.3.1. Pengujian Performansi Koneksi Ke Nintendo WFC

##### A. Tujuan

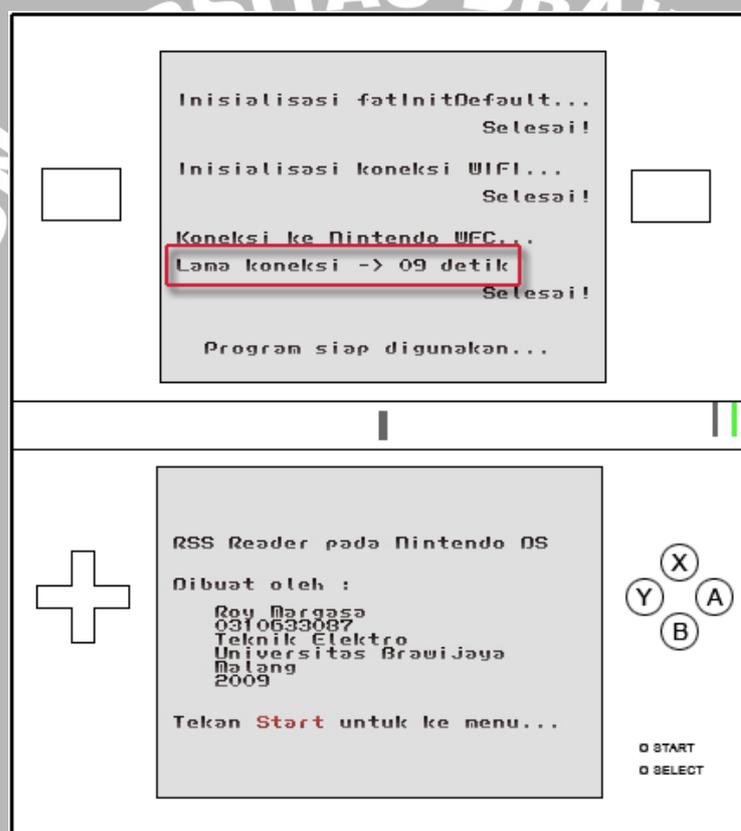
Pengujian dilakukan untuk menganalisa performansi yang dibutuhkan sistem untuk melakukan koneksi ke Nintendo WFC (Wi-Fi Connection).

## B. Prosedur Pengujian

1. Pada sistem disisipkan penghitung waktu dalam detik yang akan menghitung performansi waktu untuk koneksi ke Nintendo WFC.
2. Melakukan koneksi ke Nintendo WFC sebanyak 10 kali.

## C. Proses Pengujian

1. Penghitung waktu yang telah disisipkan pada sistem ditampilkan pada gambar 6.13.



Gambar 6.13. Penghitung Waktu Koneksi ke Nintendo WFC

Sumber: [Pengujian]

2. Hasil pengujian performansi waktu koneksi ke Nintendo WFC ditunjukkan pada tabel 6.4.

Tabel 6.4. Tabel Hasil Pengujian Untuk Koneksi Ke Nintendo WFC

Percobaan Ke	Waktu koneksi (detik) Ke Nintendo WFC	
1	sukses	12
2	sukses	9
3	sukses	8
4	sukses	7
5	gagal	-
6	sukses	9
7	sukses	7
8	sukses	8
9	sukses	9
10	sukses	8
rata-rata		8,55

Sumber: [Pengujian]

#### D. Hasil Pengujian

Sistem inialisasi telah berhasil melakukan koneksi ke Nintendo WFC dengan kondisi keberhasilan 90%, waktu terlama 12 detik, waktu tercepat 7 detik, dan dengan waktu rata-rata 8,55 detik. Kegagalan yang terjadi dikarenakan malfungsi dari hardware.

#### 6.3.2. Pengujian Performansi Koneksi Permintaan Data RSS Baru

##### A. Tujuan

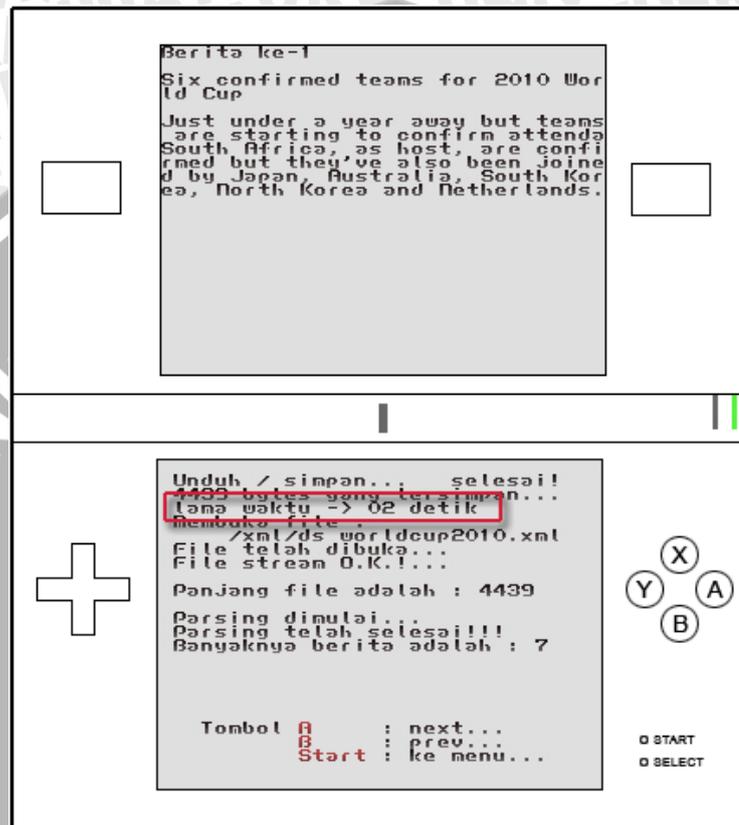
Pengujian dilakukan untuk menganalisa performansi koneksi yang dibutuhkan sistem untuk mengambil RSS feed yang berada pada internet dan menyimpannya kedalam file XML di MicroSD.

##### B. Prosedur Pengujian

1. Pada sistem disisipkan penghitung waktu dalam detik yang akan menghitung performansi waktu untuk permintaan data RSS baru.
2. Mengakses tiap RSS feed sebanyak 10 kali.

### C. Proses Pengujian

1. Penghitung waktu yang telah disisipkan pada sistem ditampilkan pada gambar 6.14.



Gambar 6.14. Penghitung Waktu Permintaan Data RSS Baru

Sumber: [Pengujian]

2. Hasil pengujian performansi waktu permintaan data RSS baru terhadap RSS feed international.okezone.com; gametrailers.com bagian NDS; dan worldcup2010southafrica.com ditunjukkan pada tabel-tabel berikut.

Tabel 6.5. Tabel Hasil Pengujian Untuk RSS Feed international.okezone.com

Percobaan Ke	Waktu unduh (detik) RSS feed okezone	
1	sukses	3
2	sukses	1
3	sukses	1
4	sukses	<1
5	sukses	1
6	sukses	1
7	sukses	1
8	sukses	2
9	sukses	1
10	sukses	1
rata-rata		1,2

Sumber: [Pengujian]

Tabel 6.6. Tabel Hasil Pengujian Untuk RSS Feed gametrailers.com bagian NDS

Percobaan Ke	Waktu unduh (detik) RSS feed gametrailers	
1	sukses	2
2	sukses	<1
3	sukses	<1
4	sukses	1
5	sukses	1
6	sukses	1
7	sukses	1
8	sukses	1
9	sukses	1
10	sukses	1
rata-rata		0,9

Sumber: [Pengujian]

Tabel 6.7. Tabel Hasil Pengujian Untuk RSS Feed worldcup2010southafrica.com

Percobaan Ke	Waktu unduh (detik) RSS feed worldcup2010	
1	sukses	4
2	sukses	1
3	sukses	2
4	sukses	2
5	sukses	2
6	sukses	1
7	sukses	2
8	sukses	2
9	sukses	2
10	sukses	1
rata-rata		1,9

Sumber: [Pengujian]

#### D. Hasil Pengujian Dan Analisis

1. Sistem permintaan data RSS baru telah berhasil mengambil RSS feed international.okezone.com dengan kondisi keberhasilan 100%, waktu terlama 3 detik , waktu tercepat dibawah 1 detik, dan dengan waktu rata-rata 1,2 detik.
2. Sistem permintaan data RSS baru telah berhasil mengambil RSS feed gametrailers.com bagian NDS dengan kondisi keberhasilan 100%, waktu terlama 2 detik , waktu tercepat dibawah 1 detik, dan dengan waktu rata-rata 0,9 detik.
3. Sistem permintaan data RSS baru telah berhasil mengambil RSS feed international.okezone.com dengan kondisi keberhasilan 100%, waktu terlama 4 detik , waktu tercepat 1 detik, dan dengan waktu rata-rata 1,9 detik.

## BAB VII

### KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan dan sara dari proses pengembangan Sistem Aplikasi Penerima Berita Melalui RSS Pada Nintendo DS.

#### 7.1. Kesimpulan

Sistem Aplikasi Penerima Berita Melalui RSS Pada Nintendo DS dapat berfungsi sesuai dengan perancangan dan pengujian yang ditunjukkan melalui:

1. Aplikasi homebrew pada Nintendo DS dengan menggunakan bahasa pemrograman C++ untuk membaca RSS feed telah berhasil dirancang sesuai dengan analisis kebutuhan dan terdiri dari beberapa aplikasi sistem antara lain: Sistem Pengolahan Data RSS dan Sistem Administrasi Data RSS.
2. Analisis kebutuhan dari sistem yang dirancang menghasilkan 5 proses pada Data Flow Diagram (DFD) yang telah divalidasi kebenarannya dengan menggunakan software Visible Analyst 7.5.
3. Pengujian performansi yang dilakukan terhadap performansi permintaan data RSS baru memiliki tingkat keberhasilan 100 %, dan untuk performansi koneksi ke Nintendo WFC memiliki tingkat keberhasilan 90%.
4. Pengujian implementasi Sistem Aplikasi Penerima Berita Melalui RSS Pada Nintendo DS secara online terhadap 5 kebutuhan implementasi berhasil dilakukan dengan hasil tanpa kesalahan sama sekali.

#### 7.2. Saran

Saran yang dapat diberikan untuk pengembangan Sistem Aplikasi Penerima Berita Melalui RSS Pada Nintendo DS antara lain:

1. Data-data RSS feed Sistem Aplikasi Penerima Berita Melalui RSS Pada Nintendo DS dapat dikembangkan menjadi lebih dari tiga sumber situs berita.

2. Sistem Aplikasi Penerima Berita Melalui RSS Pada Nintendo DS dapat dikembangkan untuk diintegrasikan dengan homebrew yang lain untuk memaksimalkan fungsi Nintendo DS.



## DAFTAR PUSTAKA

- [ALF-07] Al Fatta, Hanif. 2007. **Analisis & Perancangan Sistem Informasi**. Penerbit Andi, Yogyakarta.
- [SNC-07] St. Norbert College Technology Support Services. 2007. **Introduction to RSS**
- [SIS-04] Siswoutomo, Wiwir. 2004. **Membangun Web Service Open Source Menggunakan PHP**. Penerbit Elex Media Komputindo. Jakarta.
- [MAR-04] Martin, J., dan Tomson, B. 2004. **Belajar Sendiri ASP.Net Dalam 24 Jam**. Penerbit Andi . Yogyakarta
- [WOR-09] [http://workshop.evolutionzone.com/ Code & form » Code Read RSS feeds in Processing.htm](http://workshop.evolutionzone.com/Code%20&%20form%20»%20Code%20Read%20RSS%20feeds%20in%20Processing.htm). Diakses tanggal 30 Januari 2009.
- [AME-08] Amero, J. 2008. **Introduction to Nintendo DS Programming**. Texas, USA
- [BOU-09] Boudeville, Olivier. 2008. **A Guide To Homebrew Development For The Nintendo DS**.
- [MSH-08] Marshal, Brian. 2008. **C dan Cpp**.
- [W3C-09] [http://w3schools.com/rss/rss\\_syntax.asp](http://w3schools.com/rss/rss_syntax.asp). Diakses tanggal 30 Januari 2009.
- [SYA-08] Syahputra, Octoriano. 2008. **White Box Testing**.
- [SYH-08] Syahputra, Octoriano. 2008. **Black Box Testing**.
- [ANO-09] Anonymous. 2009. **Nintendo DS Lite**. [http://en.wikipedia.org/wiki/Nintendo\\_ds\\_lite](http://en.wikipedia.org/wiki/Nintendo_ds_lite). Diakses tanggal 15 April 2009.
- [ANY-09] Anonymous. 2009. **Wi-Fi**. <http://en.wikipedia.org/wiki/Wifi>. Diakses tanggal 15 April 2009.
- [ANN-09] Anonymous. 2009. **Nintendo DS Homebrew**. [http://en.wikipedia.org/wiki/Nintendo\\_DS\\_homebrew](http://en.wikipedia.org/wiki/Nintendo_DS_homebrew). Diakses tanggal 15 April 2009.