

Penggunaan Metode *Multi Threads* untuk Pengelolaan Proses *Download* di Internet

Zen Munawar, S.T., S.Kom., M.Kom

Program Studi Manajemen Informatika

Politeknik LP3I Bandung

Jl.Pahlawan No. 59 Bandung, Indonesia

E-mail:munawarzen@gmail.com

Abstrak : Penelitian ini bertujuan merancang aplikasi *download manager* yang dapat mengatur beberapa proses *download* sekaligus dengan metode *multi-thread* pada sebuah aplikasi *download* yang efisien dan membandingkan aplikasi yang dibuat dengan aplikasi yang sudah ada. Hasil penelitian ini diharapkan dapat memberikan kegunaan bagi pengguna internet dalam melakukan proses *download* file dengan sebuah aplikasi yang dapat menangani beberapa proses *download* serta memberikan pilihan thread (eksekusi *download* terhadap file) dengan metode *multi-thread*. Hasil pengujian pada penelitian ini, aplikasi yang dibuat setara dengan aplikasi berlisensi yang sudah ada.

Kata kunci : *download, download manager, thread, multi-thread*

1. Pendahuluan

Informasi dapat diperoleh dari internet, untuk memperoleh informasi tersebut tersedia fasilitas yang disebut *download*.

Download merupakan proses transfer data dari *server* ke *client*. Agar data dapat sampai ke *client*, maka harus melalui lalu lintas jaringan. karena lalu lintas jaringan internet semakin padat, terkadang user yang melakukan *download* sering mengalami kegagalan. Hasilnya data tidak sampai atau data sampai ke pengguna namun dalam keadaan rusak atau koneksinya terputus ditengah proses sehingga pengguna harus mengulangi proses *download* dari awal.

Masalah di atas dapat diperbaiki dengan memperbaiki kinerja *download*, antara lain dengan membuat *download manager* yang dapat mengatur proses *download* tersebut. Jika saat proses *download* koneksinya tiba-tiba terputus maka pengguna dapat melanjutkan proses tersebut tanpa harus mengulanginya dari awal. Konsep *multi-thread* ini diterapkan pada perangkat lunak.

Tujuan dari penelitian ini adalah merancang aplikasi *download manager* yang dapat mengatur beberapa proses *download* sekaligus, menerapkan metoda *multi-thread* pada sebuah aplikasi *download* yang efisien, membandingkan aplikasi yang dibuat dengan aplikasi berlisensi yang sudah ada.

Setelah implementasi dilakukan pengujian untuk mengetahui kinerja aplikasi dengan membandingkan dengan aplikasi yang sudah ada.

Hasil penelitian ini diharapkan dapat memberikan aplikasi alternatif bagi pengguna dalam proses *download file* dengan sebuah aplikasi yang dapat menangani beberapa proses *download* serta memberikan pilihan *thread* dengan metode *multi-thread*.

2. Metode Penelitian

Studi pustaka, yaitu dengan membaca dan mempelajari literatur berupa teori-teori seperti : jurnal, buku, dan artikel-artikel yang berhubungan dengan penelitian yang dilakukan.

Pengumpulan Data, Tahap mengumpulkan fakta dan informasi untuk mendapatkan informasi.

Desain dan Implementasi, dilakukan dengan cara melakukan perancangan dan implementasi sistem untuk membuat gambaran yang jelas dari masalah yang dihadapi dengan pembuatan program aplikasi.

Pengujian Fungsional, metode yang digunakan adalah *black box testing* untuk membantu kesalahan pada perangkat lunak dan sistem dapat berjalan sesuai dengan kebutuhan, serta membandingkan hasil dengan aplikasi yang sudah ada.

3. Landasan Teori

Download merupakan sebuah proses dalam pentransferan dari satu komputer ke komputer lain dalam suatu jaringan, tipikalnya dari komputer *host*, atau sebuah *network server* ke sebuah *Personal Computer (PC)* atau sebuah *Workstation*.

Thread merupakan abstraksi yang mendefinisikan sebuah alur eksekusi. Dari definisi tersebut maka *multi-thread* dapat disebut sebagai abstraksi yang memiliki alur eksekusi lebih dari satu.

Multi-thread merupakan perluasan dari paradigma dari *multitasking*. *Multitasking* adalah kemampuan komputer untuk melakukan banyak tugas secara konkuren. Pada *multitasking*, konkurensi melibatkan banyak proses. Perbedaannya dengan *multi-thread* adalah *multi-thread* lebih menggunakan banyak benang (*thread*) kendali dalam satu proses. (Bambang Hariyanto, 2003)

Java adalah bahasa pemrograman kelanjutan dari bahasa pemrograman C dan C++. Terdapat beberapa kemiripan antara Java dengan C atau C++ namun Java memiliki ciri khas tersendiri yang membedakannya dengan kedua bahasa tersebut. antara lain : *Platform independence*, *Berorientasi Objek*, *Java API*, *Internet*

Unified Modeling Language (UML) adalah satu set notasi dan konvensi yang digunakan untuk menggambarkan dan memodelkan sebuah aplikasi. Tetapi, UML. tidak menspesifikasikan sebuah metodologi atau langkah apa yang perlu diambil dalam pengembangan perangkat lunak. hal tersebut merupakan tugas dari pendekatan unified (Unified Approach-UA) (Sholih : 2006).

UML memiliki 3 aspek utama (Sinan Si Alhir : 2003) yaitu Bahasa, Model dan Unified.

Untuk sistem yang rumit, pendekatan terbaik adalah dengan memodelkan, sehingga tidak ada satu pun titik pandang yang akan kekurangan. Setiap model mungkin diekspresikan pada tingkatan ketelitian yang berbeda (Sholih : 2006). Model yang terbaik adalah model yang berhubungan dengan kenyataan. Pada Penelitian ini digunakan grafik diagram : Use-case Diagram, Activity Diagram, Package Diagram.

Booth menerangkan bahwa menggambarkan sebuah kegiatan secara sistemasi dalam sebuah media yang statis seperti di atas selembar kertas sangatlah sulit, tetapi masalah tersebut dihadapi oleh kebanyakan disiplin ilmu. Pada pengembangan berorientasi objek, anda dapat mengekspresikan

“Penguujian perangkat lunak adalah elemen kritis dari jaminan kualitas perangkat lunak dan merepresentasikan kajian pokok dari analisis , desain dan pengkodean” . (Roger S. Pressman, 1997 : 525).

Semua produk yang direkayasa dapat diuji dengan satu atau dua cara yaitu black box testing dan white box testing (Roger S. Pressman, 1997 : 532):

4. Analisis

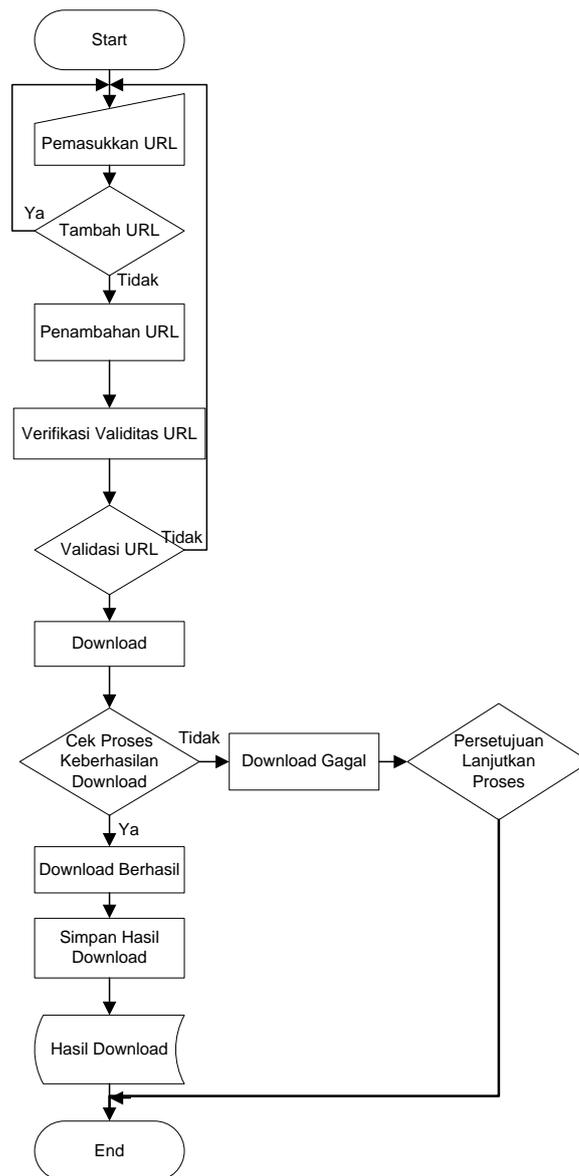
Analisis sistem dapat didefinisikan sebagai penguraian dari suatu sistem yang utuh kedalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya. (Jogiyanto HM, 2002)

Sebuah *download manager* mengidentitikasi alamat file bukan alamat website dari file tersebut. Oleh karena itu dibutuhkan sebuah inputan alamat lengkap yang akurat.

Analisis Output Aplikasi, *Output* dari sebuah aplikasi *download manager* adalah laporan dari proses *download* yang meliputi kecepatan transfer *file*, waktu *download* dan sukses tidaknya sebuah proses *download*.

Analisis terhadap sistem yang telah berjalan penulis menggunakan perangkat lunak pembanding. *Download manager* adalah perangkat lunak yang dapat menangani beberapa proses *download* sekaligus. Disini penulis menggunakan aplikasi *download* komersial *Download Accelerator Plus* (DAP) sebagai perangkat lunak pembanding. Namun disini penulis menggunakan versi *Trial* dari perangkat lunak tersebut.

Flow Chart :



Gambar *Flow Chart* dari Aplikasi DAP

Analisis Perangkat Lunak, Sebuah aplikasi *download manager* harus memiliki kemampuan yaitu memiliki field, fasilitas, fungsi, informasi, fungsi konfigurasi. Untuk membangun perangkat lunak ini dibutuhkan beberapa perangkat lunak pendukung, yaitu : Sistem Operasi MS Windows XP SP2, Browser MS IE atau Mozilla Firefox, Java Netbeans 6.9, Apache Server.

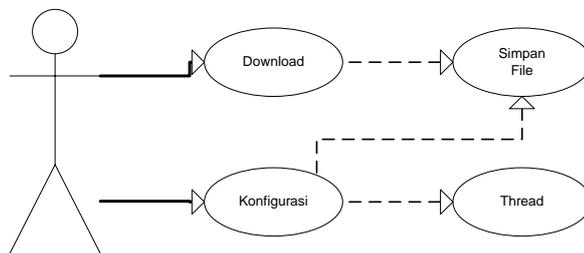
Analisis Kebutuhan Perangkat Keras, Untuk membuat aplikasi download manager ini dibutuhkan sebuah perangkat lunak dengan spesifikasi berikut CPU Pentium IV 2.4 GHz, Ram 1 Gb, Hard Disk ATA 80 MB 5400 rpm, Modem Terintegrasi, Monitor Standar, Keyboard dan Mouse.

Analisis Brainware, Analisis *brainware* dilakukan untuk mengidentifikasi kelayakan atau kemampuan seseorang yang akan menggunakan aplikasi yang dibuat.

5. Perancangan dan Pengujian

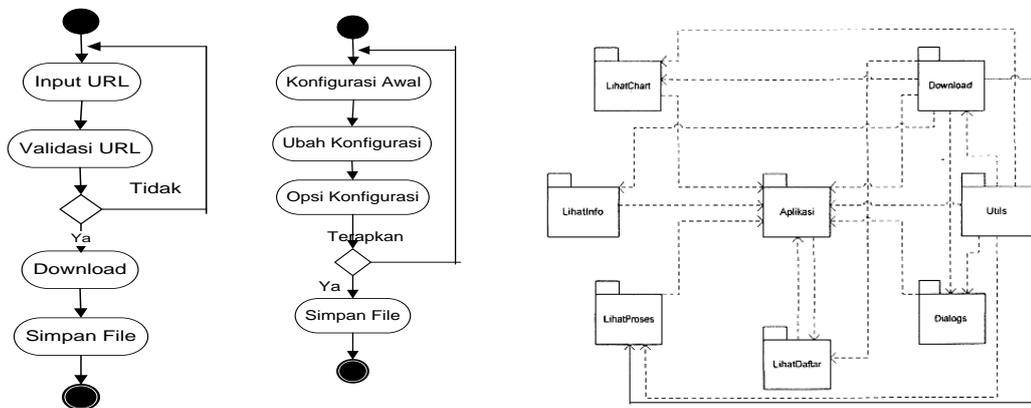
Perancangan Sistem, Pemodelan aplikasi menggunakan UML (*Unified Modeling Language*). Dalam perancangannya digunakan pemodelan dengan diagram sebagai berikut *Use Case Diagram, Activity Diagram, Class Diagram*

Use Case Diagram, *Use case diagram* menggambarkan fungsionalitas sistem atau persyaratan-persyaratan yang harus dipenuhi sistem dari pandangan pemakai.



Gambar *Use Case Diagram* aplikasi *download manager*

Activity Diagram (Diagram Aktivitas), digunakan untuk memodelkan aliran kerja (*workflow*) dalam perangkat lunak.



Gambar Diagram aktivitas proses *download*, proses Konfigurasi, kelas

Diagram Aktivitas Proses konfigurasi, Berikut ini dijabarkan aliran kerja pada proses konfigurasi *download* dari aplikasi.

Rancangan Class Diagram (Diagram Kelas), digunakan untuk menampilkan kelas-kelas atau paket-paket di dalam suatu sistem dan relasi antar kelas-kelas tersebut. Diagram kelas atau paket memberikan gambaran

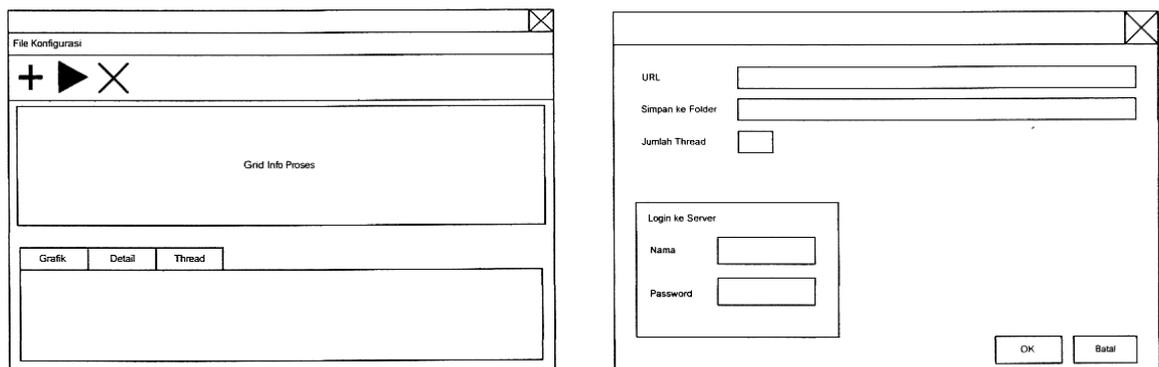
secara static yang digunakan untuk membangun sumber kode pemhangunan aplikasi yang dirancang.

Berikut adalah gambaran rancangan paket-paket dan relasinya yang digunakan dalam aplikasi *download manager* :

Gambar tersebut menjelaskan bahwa tiap-tiap kelas dari aplikasi memiliki *dependency* (ketergantungan) satu sama lainnya. Kelas-kelas ini mempunyai fungsi masing-masing dimana hasil dari fungsi dari kelas tersebut selanjutnya akan digunakan oleh kelas lain untuk memproses fungsinya.

Perancangan Antar Muka (*Graphical User Interface/GUI*), dibutuhkan agar pengguna aplikasi dapat mengoperasikan aplikasi dengan nyaman dan mudah. Hal ini menjadi salah satu aspek penting dalam keberhasilan aplikasi tersebut. Perancangan berikut dilakukan untuk menggambarkan desain antar muka aplikasi yang akan dibangun.

Perancangan *Form* Utama Aplikasi, memuat tombol-tombol untuk *download*, label informasi proses *download* dan tab untuk melakukan konfigurasi.

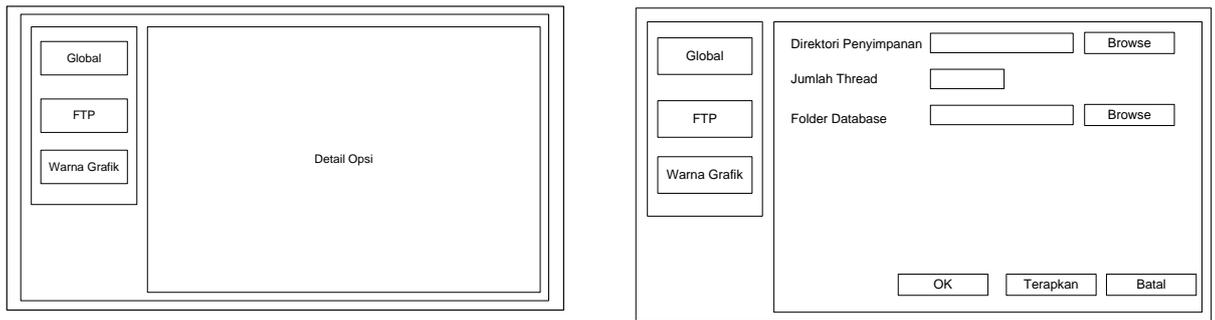


Gambar 5.3 Rancangan form Utama Aplikasi dan Rancangan form Tambah download

Perancangan *Form* Tambah *Download*, dibangun untuk melakukan proses *download*, terdapat *field* untuk penginputan *URL*, menentukan di *folder* mana *file* akan disimpan, jumlah *thread* serta fasilitas *login* ke *website server* tujuan.

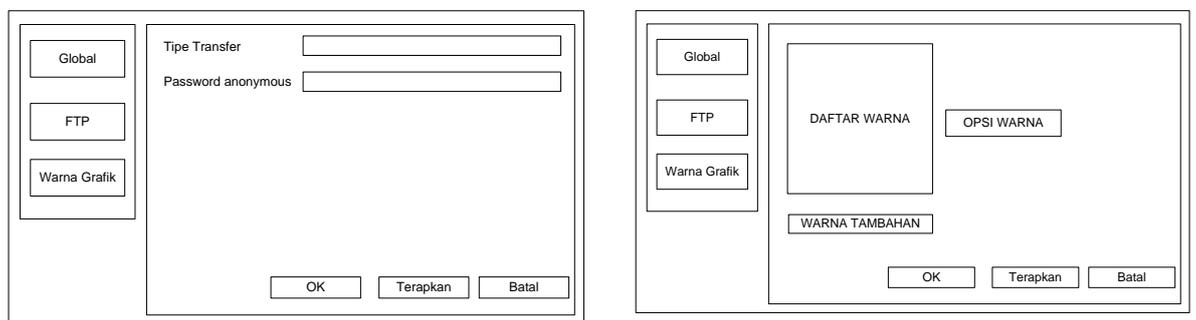
Perancangan *Form* Menu Konfigurasi, berisi *setting-setting* untuk proses *download*, disini akan tersedia berbagai konfigurasi dari aplikasi yang dibuat.

Perancangan *Form* Konfigurasi Global, berisi *setting* folder, jumlah *thread* dan dimana database aplikasi akan dibuat.



Gambar 5.6 Rancangan Menu form Konfigurasi dan form Konfigurasi Global

Perancangan Form Konfigurasi FTP dan Form Konfigurasi Warna Grafik



Gambar 5.7 Rancangan form konfigurasi FTP dan Form Konfigurasi Warna Grafik

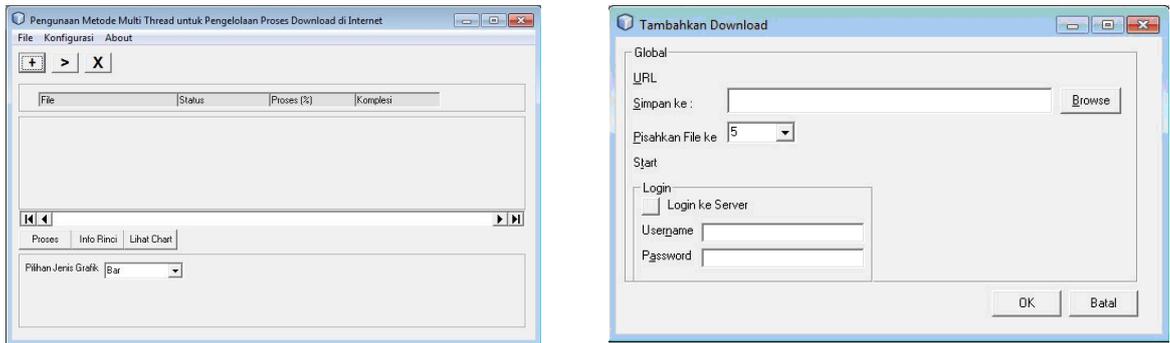
Pengujian, merupakan suatu elemen kritis dari jaminan sistem yang telah dibuat dan merepresentasikan kajian pokok dari spesifikasi, desain dan pengkodean.

Pengujian Tampilan, Pengujian tampilan dilakukan untuk melihat apakah aplikasi yang telah dibuat dapat diakses dengan baik dalam sistem komputer.

Tabel 5.1 Pengujian Tampilan

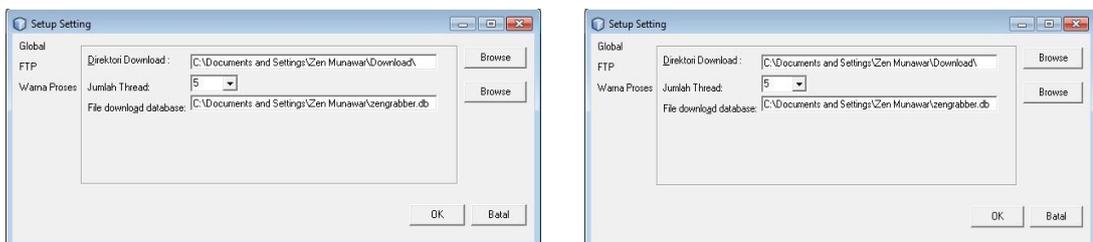
No.	Modul	Berhasil	
		Ya	Tidak
1	Form Utama	X	
2	Form Tambah Download	X	
3	Form Menu Konfigurasi	X	
4	Form Menu Konfigurasi Global	X	
5	Form Menu Konfigurasi FTP	X	
6	Form Menu Konfigurasi Warna Grafik	X	

Tampilan *Form* Utama dan *Form* Tambah Download



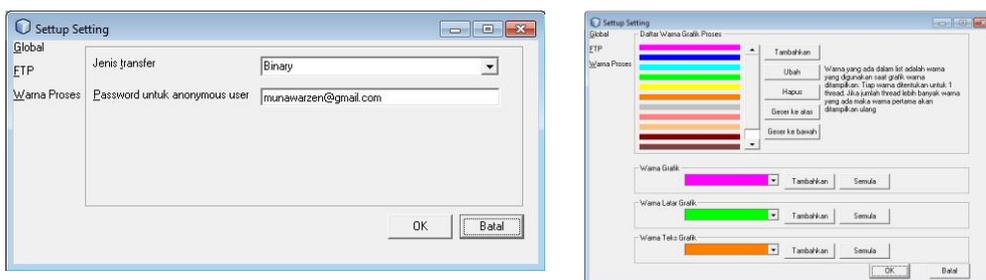
Gambar 5.8 Tampilan *Form* Utama dan *Form* Tambah Download

Tampilan *Form* Menu Konfigurasi



Gambar 5.9 Tampilan *Form* Menu Konfigurasi dan *Form* Menu Konfigurasi Global

Tampilan *Form* Konfigurasi FTP dan *Form* Konfigurasi Warna Grafik



Gambar 5.10 Tampilan *Form* Konfigurasi *Form* FTP dan Konfigurasi Warna Grafik

Pengujian Fungsional, Pengujian dilakukan dengan menguji fungsi dari aplikasi seperti pengujian pen-download-an beberapa file, pengujian *download* dengan jumlah *thread* yang berbeda dan pengujian penyimpanan file pada *folder*.

Pengujian Pengeksekusian Beberapa *download*. pada pengujian ini aplikasi diuji dengan mengeksekusi beberapa proses *download* sekaligus. Berikut ini tabel hasil pengujiannya :

Tabel 5.2 Pengujian Eksekusi Beberapa Download

No	Nama File	Ukuran	Tipe File	Berhasil	
				Ya	Tidak
1	Backup4all-User-Manual.pdf	1.88 Mb	File PDF	x	
2	Granule-1.3.0-3-win32.rar	19 Mb	Arsip (Rar)	X	
3	Speedygnfs11-ch.zip	126 Kb	Arsip (Zip)	X	

Dari hasil pengujian diatas dapat dilihat bahwa aplikasi dapat menangani beberapa proses download sekaligus.

Pengujian penyimpanan file pada folder, pengujian ini akan diuji apakah aplikasi dapat menyimpan file path folder yang ditentukan oleh user. File beserta folder tujuannya sebagai berikut : Speedygnfs-ch-zip disimpan di folder “C:\download”, Backup4all-User-Manual.pdf, disimpan di folder “C:\Zen Munawar Files\Download”, Granule-1-3.3.0-3-win32.rar, disimpan di folder “C:\Document and Settings\Zen Munawar”

Tabel 5.3 Hasil Pengujian Penyimpanan File

No	Nama File	Hasil Download	Sesuai	
			Ya	Tidak
1	Speedygnfs11-ch.zip	C:\download	X	
2	Backup4all-User-Manual.pdf	C:\Zen Munawar Files\Download	x	
3	Granule-1.3.0.3-win32.rar	C:\Document and Settings\Zen Munawar	x	

Dari pengujian diatas dapat dilihat aplikasi dapat menyimpan file hasil *download* di folder yang telah ditentukan oleh *user*.

Pengujian Akselerasi Waktu, dilakukan untuk mengetahui waktu pen-download-an dengan *men-download* 2 file yang berbeda tipe ukuran. Dikarenakan *traffic* Internet yang selalu berubah-ubah maka pengujian dilakukan sebanyak 5 kali kemudian diambil waktu rata-ratanya. Thread maksimum pengujian adalah 5. Berikut ini hasil penguiannya: File 1 Nama File : speedygnfs11-ch.zip, Ukuran 126 Kb,

Tabel 5.4 Akselerasi waktu download tiap thread file 1

Jumlah Thread	Waktu (detik) di Percobaan Ke-					Rata-rata Waktu
	1	2	3	4	5	
1	36	30	32	38	30	33,2
2	25	30	30	32	31	29,6
3	27	25	27	31	33	28,6
4	26	27	26	29	32	28
5	24	26	27	30	29	27,2

File 2 , Nama File : Backup4all-User-Manual.pdf, Ukuran 1,88 Mb

Tabel 5.5 Akselerasi waktu download tiap thread file 2

Jumlah Thread	Waktu (detik) di Percobaan Ke-					Rata-rata Waktu
	1	2	3	4	5	
1	386	370	395	383	387	384,2
2	387	383	390	377	375	382,4
3	389	388	371	380	370	379,6
4	383	378	381	378	377	379,4
5	385	379	380	381	369	378,8

Dari pengujian *download 2 file diatas* dapat dilihat bahwa tiap peningkatan thread akan memberikan tambahan akselerasi waktu. meskipun tidak telalu signifikan.

Pengujian dengan perbandingan, Sebagai pembandingan penulis menggunakan aplikasi komersil *Download Accelerator Plus Trial* .Pengujian dilakukan membandingkan fasilitas dan akselerti waktu kedua aplikasi.

Tabel 5.6 Uji Banding Fitur

No	Kriteria pembandingan	Aplikasi yang dibuat	DAP Versi Trial
1	Proses download >= 1	Ya	Ya
2	Meneruskan proses download setelah mengalami kegagalan	Ya	Ya
3	Opsi Multi-thread	Ya	Tidak
4	Koneksi ke FTP	Ya	Ya
5	Fungsi Upload	Tidak	Ya
6	Opsi folder penyimpanan	Ya	Ya
7	Laporan proses	Ya	Ya
8	Laporan tiap thread	Ya	Tidak

Uji Akselerasi Waktu Apiikasi Pembandingan, DAP Plus Trial diuji akselerasi waktunya dahulu sebelum dibandingkan dengan aplikasi yang dibuat.

File 1 Nama File : *speedygnfsl I-ch.zip*, Ukuran : 126 Kb

Tabel 5.7 Waktu download file I

Jumlah Thread	Waktu (detik) di percobaan ke					Rata-rata waktu
	1	2	3	4	5	
	34	24	26	27	25	27,2

File 2 Nama File *Backup4all-User-Manual.pdf*, Ukuran : 1,88 Mb

Tabel 5.8 Waktu download file 2

Jumlah Thread	Waktu (detik) di percobaan ke					Rata-rata waktu
	1	2	3	4	5	
	373	379	377	387	379	27,2

Perbandingan Akselerasi Waktu

Berikut ini adalah perbandingan akselerasi waktu antara aplikasi yang dibuat dengan aplikasi pembanding. File 1 Nama File : speedygnfs11-ch-zip, Ukuran 126 Kb

Tabel 5.9 Perbandingan Akselerasi Waktu pada Download File 1

	Aplikasi yang dibuat Thread					DAP Trial
	1	2	3	4	5	
Rata-rata Waktu	33,2	29,6	28,6	28	27,2	27,2

File 2 Nama File Backup4all-User-Manual.pdf, Ukuran 1,88 Mb

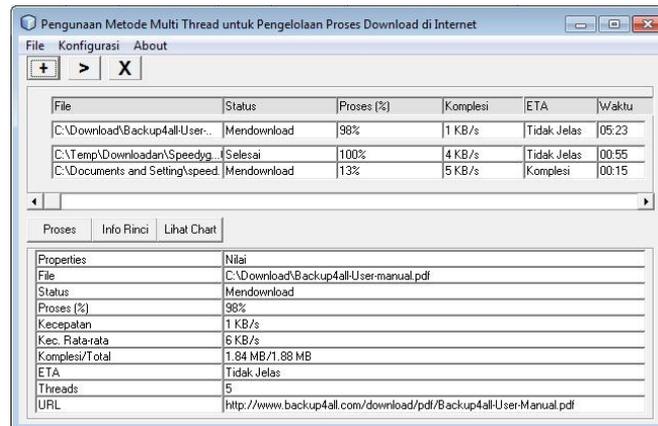
Tabel 5.10 Perbandingan Akselerasi Waktu pada Download File 2

	Aplikasi yang dibuat Thread					DAP 8.5 Trial
	1	2	3	4	5	
Rata-rata Waktu	384,2	382,4	379,6	379,4	378,8	379

Dilihat dari kedua hasil pengujian diatas, aplikasi yang dibuat bisa dikatakan setara dengan aplikasi pembanding Download Accelerator Pro.

6. Implementasi Sistem

Implementasi Sistem merupakan langkah lanjut dari tahap perancangan dan pengujian. Implementasi sistem mencakup segala kebutuhan aplikasi yang dibuat agar aplikasi dapat berjalan dengan lancar. Berikut ini adalah hasil tampilan aplikasi download manager yang telah dibuat.



Gambar 6.1 Hasil Implementasi

7. Kesimpulan

Terdapat beberapa hal yang dapat disimpulkan sebagai berikut :

1. Aplikasi yang dibuat dapat menangani proses download lebih dari satu. Proses-proses download yang sedang berjalan dapat dilihat dalam label info pada aplikasi, lengkap dengan kompleksi proses, estimasi waktu, waktu transfer, kecepatan transfer dan terdapat diagram (chart) agar mempermudah user memperoleh informasi mengenai proses download yang sedang berlangsung.
2. Metoda multi-thread dapat diterapkan pada aplikasi, pentransferan file dapat dieksekusi per bagian.
3. File hasil download bisa disimpan pada folder khusus, baik itu folder hasil penyetaan user maupun folder default dari aplikasi.

Daftar Pustaka

- [1] Chonoles, Michael Jesse, A Schardt James, 2003, *Unified Modelling Language 2*, Wiley Publishing, Inc., New York
- [2] Eddings, Joshua, 2003, *How Internet Works*, California
- [3] Hariyanto, Bambang, 2003, *Esensi-esensi Bahasa Pemrograman Java*, Informatika, Bandung
- [4] Jogiyanto HM, 2001., *Analisis & Desain Sistem Informasi : Pendekatan Terstruktur Teori dan Praktek Aplikasi Bisnis*, Andi Offset, Yogyakarta.
- [5] Lowe, Doug, 2005, *Java All-In-One Desk Reference*, Wiley Publishinh, Inc, New Jersey
- [6] Oaks, Scott., Wong, Henry, 2001, *Java Threads 2nd*, O'Really, California
- [7] Sholiq, 2006, *Pemodelan Sistem Informasi Berorientasi Objek dengan UML*, Graha Ilmu, Yogyakarta.
- [8] SI Alhir, Sinan, 2003, *Learning UML*, O'Really, California
- [9] Westnet Learning Techonlogies and Digital Creator, 1998, *Networking Fundamental*, WestNet, Washington D.C.