

PERAN APLIKASI EASY SPLITTER PADA KEMUDAHAN PEMECAHAN DAN PENGGABUNGAN FILE

ABSTRAK

Teknologi komputer memaksa kita untuk menggunakan data-data yang harus di simpan dalam media penyimpanan baik berupa floppy disk, flashdisk bahkan hardisk baik yang internal maupun yang external. Besarnya kapasitas data terkadang membuat menyedia penyimpanan tidak cukup untuk menampungnya, untuk itu diperlukan suatu teknologi guna memecahkan masalah tersebut. Untuk itu di buat suatu aplikasi Easy Spitter yang dapat memecahkan suatu file sehingga kapasitasnya menjadi lebih kecil dan dapat di tampung oleh media penyimpanan dan kemudian menggabungkan kembali file tersebut tanpa harus ada data yang hilang seperti file asli atau tanpa merubah bentuk dan struktur file yang ada. Program aplikasi ini dibuat dengan menggunakan bahasa pemrograman Java J2SDK.

Kata kunci : Easy, Splitter, Memecah, Menggabung, Aplikasi

Nani Mintarsih

Jurusan Manajemen Informatika
Fakultas Ilmu Komputer
(nanim@staff.gunadarma.ac.id)

ABSTRACT

Computer technology is forcing us to use these data to be stored in a storage medium in the form of floppy disks, hard drives and even flash both the internal and the external. The amount of data capacity to provide a storage sometimes is not enough to contain it, for it needed a technology to solve the problem. For an application was made ??Easy Spitter can solve a file so that the capacity becomes smaller and can be at capacity by the storage medium and then merge the files again without having any missing data as the original file or without changing the shape and structure of an existing file. This application program is created using the Java programming language J2SDK

Pendahuluan

Dunia teknologi pada masa sekarang ini telah mengalami kemajuan yang sangat pesat di berbagai bidang, tidak terkecuali di bidang komputer khususnya teknologi perangkat lunak (software). Saat ini perangkat lunak yang banyak digunakan seperti aplikasi audio, video, game dan lain-lain mempunyai ukuran file yang cukup besar. Untuk menyimpan file-file tersebut media penyimpanan memiliki peran yang sangat penting. Pada sebuah komputer media penyimpanan sangat dibutuhkan untuk mem-backup file dan sebagai tempat penyimpanan sementara.

Kapasitas file yang cukup besar terkadang menjadi kendala pada saat kita akan menyimpan file tersebut pada media penyimpan. Untuk itu dibuat suatu teknologi pemecah file agar dapat disimpan pada media yang tidak terlalu besar serta menggabungkan kembali nanti tanpa ada perubahan jenis dan struktur data yang hilang. Sehingga masalah penyimpanan data dapat diatasi.

Tinjauan Pustaka

Konsep File

File adalah sebuah blok informasi yang terbentuk dari beberapa byte disimpan secara bersamaan dalam sebuah media penyimpanan dalam komputer. Sebuah file dapat berupa program, dokumen, database atau apapun yang merupakan kumpulan byte.

Adapun beberapa pengertian lain mengenai file adalah sebagai berikut:

- 1 Sekumpulan data atau informasi yang saling berhubungan sesuai dengan

tujuan pembuatnya. Data pada file bisa berupa numerik, alpha numerik, binary atau text. Setiap file memiliki nama dan pengacuan terhadap suatu file menggunakan nama file tersebut.

- 1 Suatu barisan data dalam format (record tertentu) yang terekam pada suatu medium storage (disk, tape, CD, dll.).
- 1 Sebuah representasi non-fisik data yang ada pada sebuah sistem operasi. Sebuah file adalah sebutan untuk sekumpulan byte. File bisa berisi sederetan karakter atau kode ASCII yang membentuk sebuah dokumen, atau bisa juga berupa rangkaian instruksi untuk software aplikasi untuk dijalankan oleh komputer. Ringkasnya, file adalah rangkaian pola-pola byte. Saat sebuah program dijalankan atas perintah komputer, harddisk akan membaca rangkaian byte itu dan mengirimkannya ke CPU.
- 1 Kumpulan informasi yang berhubungan dan tersimpan dalam secondary storage.

File merupakan bentuk logika dari data yang disimpan dalam disk (secondary storage). Sistem operasi memetakan setiap file kedalam bentuk fisiknya dalam disk. Setiap data hanya dapat disimpan pada secondary storage dalam bentuk file. File dapat menyimpan data dalam bentuk apa saja, dalam bentuk teks, gambar, suara, dan sebagainya.

File juga merupakan unit penyimpanan dasar dalam sebuah komputer. Sebuah file memungkinkan sistem operasi Windows untuk memisahkan sekumpulan informasi satu dari yang lainnya. Sebagai contoh : Saat

kita menggunakan program untuk menulis surat, maka kita akan menyimpan surat tersebut ke dalam nama dan tipe file yang unik. Saat kita menyimpan data gambar, maka kita juga akan menyimpan ke dalam nama dan tipe file yang berbeda. Setiap file mempunyai nama dan tipe yang unik dan disesuaikan dengan data yang tersimpan.

Berbagai jenis file :

- 1 File program adalah file yang dapat di-execute (dijalankan), artinya dengan menekan enter pada file yang dipilih maka program akan dijalankan. Ada dua tipe /ekstensi file yang dapat dijalankan yaitu : .COM dan .EXE.
- 1 Batch file adalah file yang digunakan untuk menjalankan beberapa perintah atau program, atau menjalankan urutan perintah dengan hanya menjalankan satu perintah. Cara menjalankan batch file ini sama dengan menjalankan file program. Ekstensi batch file adalah: .BAT.
- 1 File teks adalah file yang hanya berisi teks. Ekstensi-nya adalah: .TXT. File sistem adalah file yang berisi informasi tentang hardware. File- file ini berekstensi: .SYS dan .DRV. Sekelompok file yang mempunyai persamaan sifat atau tujuan dapat disimpan dalam lokasi yang berbeda/folder yang berbeda. File Sistem adalah suatu struktur yang digunakan sistem operasi untuk menyimpan dan membaca data dari hard disk. File Sistem menyediakan mekanisme untuk penyimpanan data dan program yang dimiliki oleh sistem operasi serta seluruh pengguna dari

sistem komputer File System terdiri dari dua bagian:

- 1 Kumpulan file yang masing-masingnya menyimpan data-data yang berhubungan.
- 1 Struktur direktori yang mengorganisasi dan menyediakan informasi mengenai seluruh file dalam sistem.

Adapun contoh-contoh format file system sebagai berikut : FAT (File Allocation Table), FAT32 (File Allocation Table 32), NTFS (New Technology File System) (Ketiga variasi ini umum digunakan untuk platform

Windows), Ext, Ext2, Ext3 (Ketiga variasi ini umum digunakan untuk platform Linux), OS/2, HPFS, Reiser dll.

File dipergunakan untuk menyimpan data secara permanen. Secara umum, operasi yang dapat dilakukan terhadap file adalah:

- 1 Membaca
- 1 menulis.

Terdapat 2 jenis file dari bentuk datanya:

- 1 teks
- 1 biner

Biasanya langkah untuk operasi file:

1. Mendefinisikan file
2. Membuka file (untuk dibaca, ditulis, atau ditambahi)
3. Membaca atau menulis data
4. Menutup file

Sistem operasi membutuhkan struktur file tertentu untuk menjalankan/mengakses suatu file. Semua sistem operasi diharuskan mampu mengenal sedikitnya satu jenis struktur file. Jika sistem operasi mengenal semakin banyak struktur file, maka semakin luas aplikasi yang dapat dijalankan namun ukuran sistem operasi semakin membengkak. Sebaliknya, jika semakin sedikit struktur file, maka sistem operasi hanya dapat menjalankan aplikasi dalam jumlah yang sedikit pula.

Sebuah file terdiri dari kepala dan badan file. Bagian kepala berisi informasi mengenai jenis file, panjang file, dan status file. Sedangkan bagian badan berisi data-data dari file tersebut. Data-data tersebut biasanya memiliki pola-pola tertentu yang nantinya akan bermanfaat dalam proses pengembalian file (file recovery).

Contoh atribut yang dimiliki oleh sebuah file :

- 1 Nama
- 1 Tipe
- 1 Lokasi
- 1 Ukuran
- 1 Waktu pembuatan dan identitas pembuat
- 1 Proteksi
- 1 Informasi lain tentang file, dll

Contoh operasi pada file yang sering di lakukan :

- 1 Membuat
- 1 Menulis
- 1 Membaca
- 1 Menghapus
- 1 Mencari
- 1 Membuka
- 1 Menutup

Pemecah dan Penggabungan File

Teknologi komputer yang terus berkembang mendorong perkembangan yang cukup pesat di bidang perangkat lunak (software). Bermunculannya beraneka ragam software-software baru yang mampu memikat para pengguna (user) komputer untuk menggunakannya dan tak jarang program-program tersebut menjadi favorit bagi mereka. Namun, pada umumnya software-software itu menghasilkan file-file yang berukuran besar sehingga menimbulkan suatu permasalahan yang terletak pada keterbatasan media penyimpanan. Terutama bagi para pengguna komputer bermedia penyimpanan terbatas.

Misalnya seorang pengguna komputer memiliki sebuah komputer dengan media penyimpanan atau biasa disebut harddisk yang telah dipartisi menjadi dua drive yaitu C dan D. Pada suatu ketika pengguna tersebut (user) hendak menduplikat sebuah file berukuran 600 MB ke dalam komputernya. Sedangkan sisa free space di drive C dan drive D masing-masing 300 MB dan 400 MB. Di lain kesempatan, seorang pengguna komputer hendak menduplikat file berukuran 1,5 MB ke sebuah disket, sedangkan kapasitas penyimpanan pada sebuah disket hanya 1,44 MB. Maka file itu tidak muat untuk di duplikasikan pada disket.

Pengompresan data pun kadang-kadang tidak berguna jika format file yang diduplikat adalah format terkompresi seperti *.jpg, *.mp3, *.zip dan sebagainya. Masalah-masalah tersebut sering dihadapi pengguna komputer, dan bagi beberapa orang, masalah ini sangat membingungkan. Saat ini, CD memang sudah menjadi media penyimpanan yang umum, tetapi jika menduplikat ke CD akan terasa mubazir apabila diisi hanya sedikit sekali. Alternatif lain adalah menggunakan flash disk, sebuah media penyimpanan yang berukuran kecil namun memiliki kapasitas relatif besar. Namun sayangnya perangkat yang satu ini tidak murah harganya.

Untuk mengatasi masalah itu, seorang pengguna komputer dapat menggunakan Aplikasi Pemecah dan Penggabungan File (Split and Combine File Application). Dengan aplikasi semacam ini, file bisa dibagi menjadi beberapa bagian, sehingga dapat menduplikat sebuah file yang berukuran 1,5 MB itu ke dalam 2 buah disket, dan menduplikat sebuah file yang berukuran 600 MB tersebut ke drive C dan drive D. Proses pemecahan file dapat dilakukan dengan memisahkan byte-byte file ke dalam beberapa file, sehingga akan menghasilkan beberapa file yang berukuran lebih kecil dari file awal. Pemecahan file ini dapat dioperasikan untuk hampir semua jenis file. File dapat dipecah menjadi beberapa bagian, user yang menentukan ukuran file hasil pemecah. Pilihan besarnya 100 KB, 360 KB, 720 KB, 1,38 MB atau ditentukan sendiri. Setelah proses selesai, akan terbentuk file berformat namafile.000, namafile.001, namafile.002 dan seterusnya.

Metode Penulisan

Tahapan yang digunakan dalam penulisan ilmiah ini terdiri dari :

- a. Studi Pustaka
Yaitu dengan melakukan mencari materi berdasarkan referensi dari berbagai pembahasan dan literatur yang berhubungan dengan pembuatan aplikasi easy splitter berbasis java.
- b. Studi Lapangan
Metode pengumpulan data, disini penulis mendapatkan materi dari beberapa artikel dan mendownload dari internet dengan menggunakan media search engine.
[Http://www.ilmukomputer.com/berseri/eko-java/index.php](http://www.ilmukomputer.com/berseri/eko-java/index.php), Juli 2011
- c. *Melakukan analisis terhadap sistem yang berjalan.
- d. *Merancang sistem usulan dari hasil analisis sistem berjalan.
- e. Implementasi rancangan sistem usulan.
- f. Uji coba hasil implementasi sistem usulan.
- g. Evaluasi sistem yang sudah di uji coba.

Pembahasan

Pada bagian ini akan dijelaskan cara pembuatan aplikasi Easy Splitter sebagai aplikasi untuk mengatasi keterbatasan kapasitas media penyimpanan data maupun memaksimalkan kapasitas dari media penyimpanan data.

Perencanaan Aplikasi Easy Splitter 1.0

Pada pembahasan ini akan dibuat suatu aplikasi yang digunakan untuk membagi-bagi ke dalam beberapa bagian, hal ini digunakan untuk bertujuan memaksimalkan isi dari media penyimpanan data. Pada aplikasi terdapat 3 bagian utama yang dapat digunakan oleh user. Bagian pertama yaitu bagian split, bagian ini digunakan user untuk membagi-bagi file atau men-split file ke dalam beberapa bagian.

Bagian kedua yaitu bagian merge, bagian ini berfungsi untuk menggabungkan kembali file-file yang telah di-split sebelumnya. Sedangkan bagian ketiga adalah bagian option, bagian ini berfungsi untuk menentukan folder yang akan ditempatkan file split dan file merge dan juga menentukan ukuran awal atau default size yang akan di-split

Struktur Navigasi Program

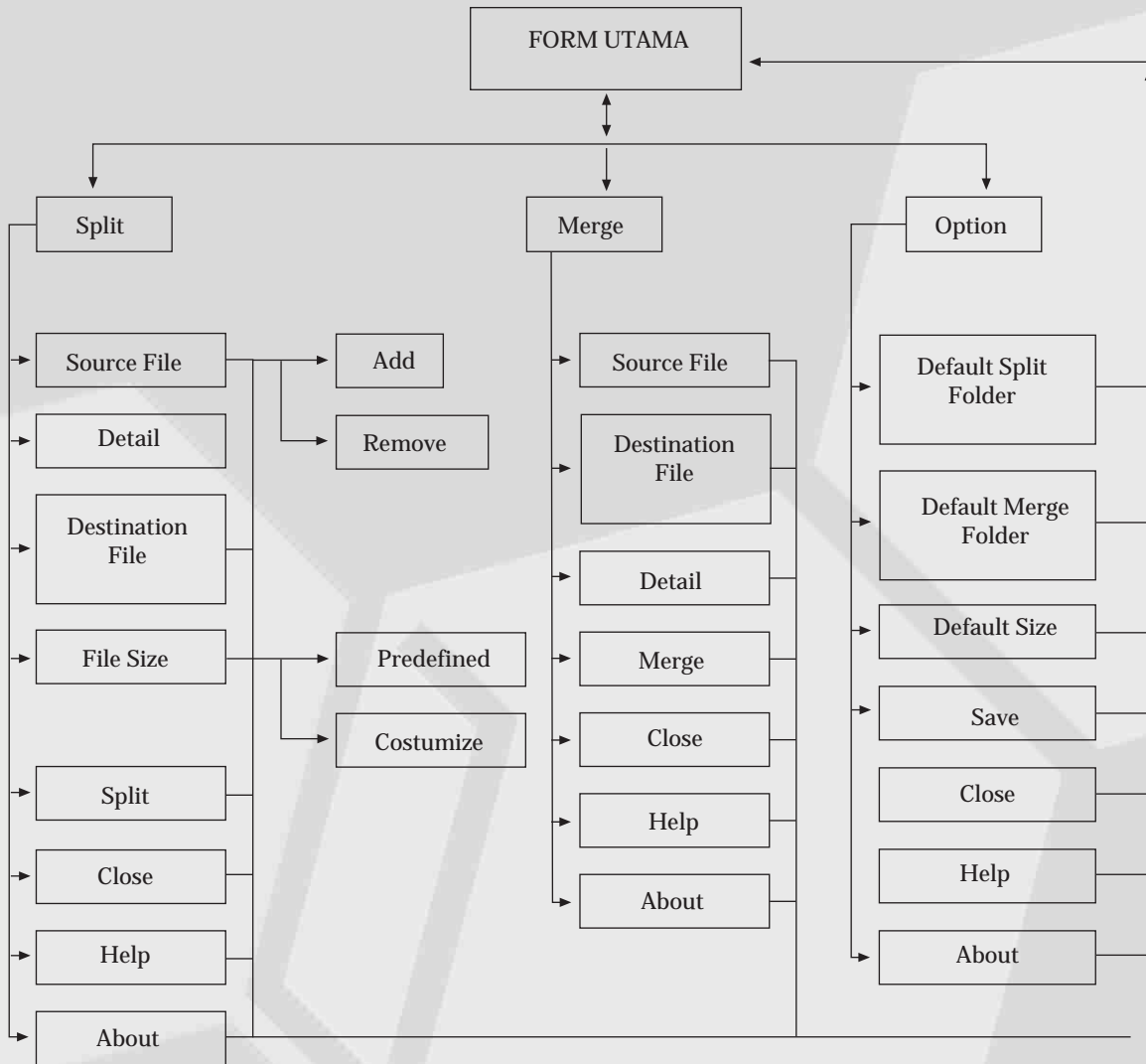
Pembuatan struktur navigasi dirancang agar jalannya program dapat diikuti dan dipahami serta digunakan untuk menandakan adanya hubungan antar menu. Struktur navigasi yang digunakan pada aplikasi ini adalah struktur navigasi hirarki. Struktur navigasi hirarki merupakan suatu struktur yang menggunakan percabangan untuk menampilkan form berdasarkan alur pemilihan dari suatu menu. Struktur navigasi aplikasi Easy Splitter ini terdiri

dari sebuah menu utama yang disebut sebagai form utama, form utama ini akan mempunyai pilihan-pilihan menu yaitu split, merge, option. Selanjutnya dari setiap menu tersebut, masing masing memiliki beberapa sub menu yang memiliki fungsi yang berbeda. Yang lebih lengkapnya dapat dilihat pada gambar 1

```
filechooser.FileSystemView;
import java.util.*;
import java.text.*;
import java.net.*;
```

b. Membuat pernyataan pendeklarasian variabel-variabel yang digunakan dalam aplikasi

```
JButton badd,bremove,brmvall,
bbrowse,bspilt,bclose,bhelp,bbrows
e2,bbrowse3,bmerge,bclose2,bgetc
mnd,bview,bhelp2,bbrowse4,bbro
wse5,bbrowse6,bsave,bclose3,bhelp
3,babout;
JButton bsp;
JButton abdetail,ababout,ababout2;
JTextField tf1,tf2,tf3,tf4,tf5,
tf6,tf7,tf8,tf9;
JTextArea ta1,ta2;
JCheckBox cdelete,cwipe,cbatch,
cdltsplit;
JRadioButton rdefined,rcustomized,
rsplit,reachsplit,rbytes,rkb,rmb;
JTabbedPane tp;
JScrollPane sp1,sp2,sp3;
ButtonGroup bg1,bg2,bg3;
JComboBox cb;
JPanel p1,p2,p3,p4,p5,p6,p7,
p8,p9,p10,p11,p12,p13,p14,p15,p16,
p17,p18,p19,p20,p21,p22,p23,p24,p
25,p26,p27;
JPanel x1,x2,x3,x4,x5,x6,
x7,x8,x9,x10;
JList lil;
JFileChooser fc1,fc2,fc3;
String[] str=new String[100];
int ctr=0;
String cmf=null;
int csize;
int ind;
String[] ps={"10 KB","100 KB","120
KB","500 KB","720 KB","1.2
MB","1.44 MB","100 MB","250
MB","650 MB"};
int[] pos={10240,102400,
122880,512000,737280,1213952,14
57644,103809020,26004085,67475
8630};
int attr[]=new int[3];
int[] size=new int[1000];
int[] no=new int[1000];
int[] lsize=new int[1000];
char[] mod=new char[1000];
int gin[]=new int[1000];
File src;
```



Gambar 1. Struktur Navigasi Program Easy Splitter

Langkah-langkah Pembuatan Program

Pembuatan Source Code Program

Setelah software pendukung telah diinstal, langkah selanjutnya adalah mengetik listing program. Hal pertama yang dilakukan adalah membuka teks editor yang ada pada komputer, lalu memberikan nama file yaitu EasySplitter.java, dan file tersebut ditempatkan atau disimpan pada folder j2sdk1.4.2_17 lalu pilih folder bin dan tempatkan file tersebut di folder bin. Seperti terlihat pada gambar 2.

Setelah membuat file Easy Splitter.java, selanjutnya dilakukan pengetikan listing program. Listing program EasySplitter sebagai berikut :

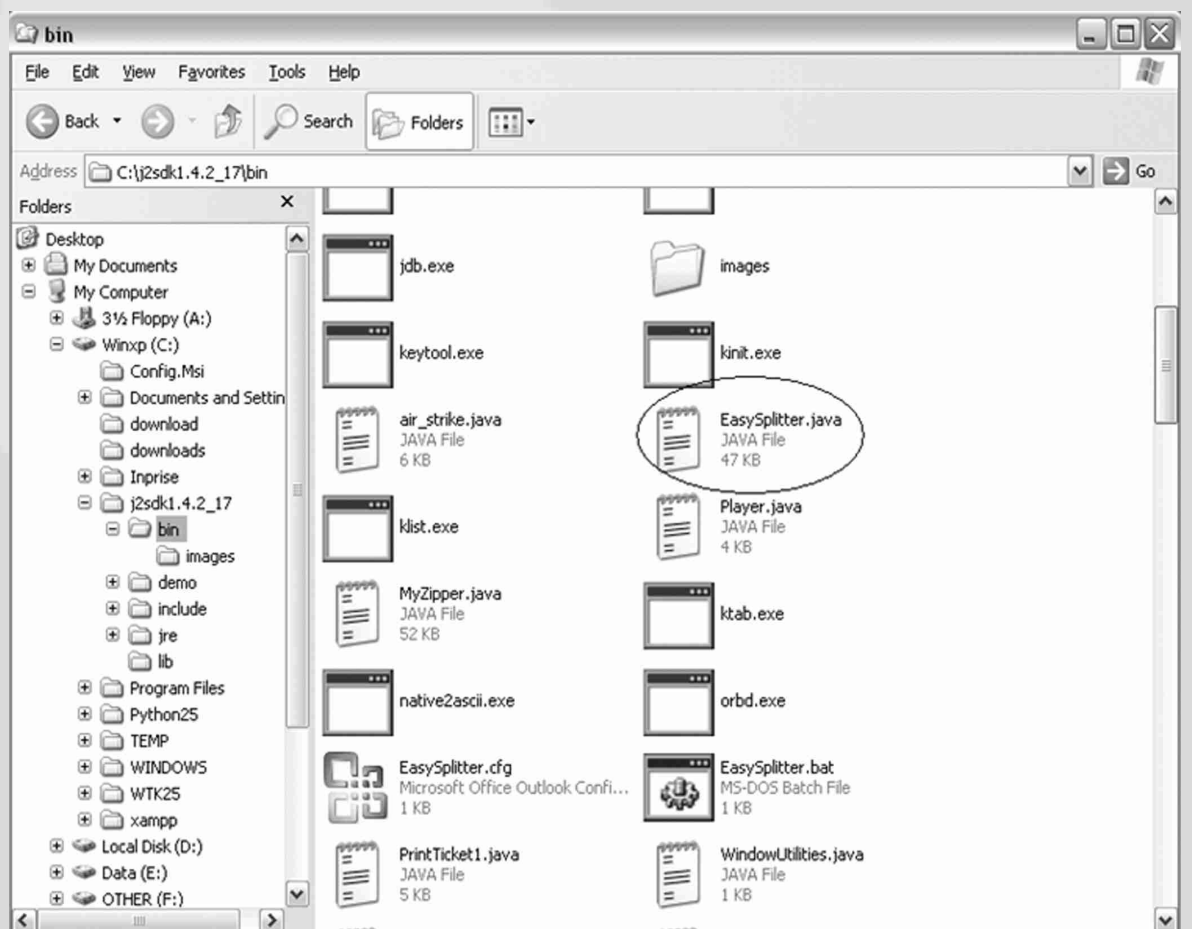
1. File EasySplitter.java

- a. Membuat pernyataan import. Import digunakan untuk mengakses semua kelas yang dibutuhkan dalam pemrograman java.


```
import java.awt.*;
import java.awt.event.*;
import java.io.*;
import javax.swing.*;
import javax.swing.event.*;
import javax.swing.border.*;
import javax.swing.
```

EasySplitter.


```
JLabel lsource,ldestination,
lcomment,lfiles,lsplit,ldestination2,lco
mment2,lorigfile,lorigsize,lnosplit,leachsp
lit,llastsplit,lhvcmd,lhvbtc,ldfitsplit,l
dfitmerge,ldfitsize,lorigfile2,lorigsize2;
```



Gambar 2. File EasySplitter

- c. Pada aplikasi EasySplitter ini dibagi menjadi 3 bagian tab yaitu *split*, *merge*, *option*. Bagian pertama yaitu tab *split*, berikut ini adalah potongan bagian listing program dari tab *split*.

```
public EasySplitter()
{
    super("EasySplitter 1.0 - Powered by
    Java");
    Container c=getContentPane();
    tp=new JTabbedPane();

    tp.setTabPlacement(JTabbedPane.
    TOP);

    lsource=new JLabel("Source
    Files:(Double click each to configure
    splits separately)");
    lsource.setToolTipText("Double click
    on each item to set split configuration
    separately and then press Set button
    to save it");
    x1=new JPanel(new FlowLayout
    (FlowLayout.LEFT));
    x1.add(lsource);
    ldestination=new JLabel
    ("Destination File:");
    x2=new JPanel(new FlowLayout
    (FlowLayout.LEFT));
    x2.add(ldestination);
    lcomment=new JLabel
    ("Comment:");
    x3=new JPanel(new FlowLayout
    (FlowLayout.LEFT));
    x3.add(lcomment);
    li1=new JList();
    int md=ListSelectionModel.
    MULTIPLE_INTERVAL_SELECTI
    ON;
    li1.setSelectionMode(md);
    li1.setVisibleRowCount(6);
    li1.addMouseListener(new
    MouseAdapter()
```

- d. Bagian ini merupakan bagian tab *merge*, berikut adalah potongan bagian listing program untuk tab *merge*.

```
//second tab

lsplit=new JLabel("Split File:");
x5=new JPanel(new FlowLayout
(FlowLayout.LEFT));
x5.add(lsplit);
ldestination2=new JLabel
("Destination Folder:");
x6=new JPanel(new
FlowLayout(FlowLayout.LEFT));
x6.add(ldestination2);
lcomment2=new JLabel
("Comment:");
x7=new JPanel(new
FlowLayout(FlowLayout.LEFT));
x7.add(lcomment2);

tf4=new JTextField(25);
tf5=new JTextField(25);
ta2=new JTextArea(30,4);
ta2.setLineWrap(true);
ta2.setWrapStyleWord(true);
sp3=new JScrollPane(ta2);

bbrowse2=new JButton("Browse");
bbrowse3=new JButton("Browse");
bmerge=new JButton("Merge");
bclose2=new JButton("Close");
bgetcmd=new JButton("Get
```

```
Comment");
bview=new JButton("View Detail");
bhelp2=new JButton("Help");
ababout=new JButton("About");
```

```
bbrowse2.addActionListener(this);
bbrowse3.addActionListener(this);
bmerge.addActionListener(this);
bclose2.addActionListener(this);
bgetcmd.addActionListener(this);
bview.addActionListener(this);
bhelp2.addActionListener(this);
ababout.addActionListener(this);
```

- e. Bagian ini merupakan bagian dari tab ke 3 yaitu tab *option*. Berikut adalah potongan bagian listing program untuk tab *option*.

```
//third tab

tf6=new JTextField(25);
tf7=new JTextField(25);
tf8=new JTextField(25);

ldfltsplit=new JLabel("Default Split
Folder:");
x8=new JPanel(new FlowLayout
(FlowLayout.LEFT));
x8.add(ldfltsplit);
ldfltmerge=new JLabel("Default
Merge Folder:");
x9=new JPanel(new FlowLayout
(FlowLayout.LEFT));
x9.add(ldfltmerge);
ldfltsize=new JLabel("Default
Size:");
x10=new JPanel(new FlowLayout
(FlowLayout.LEFT));
x10.add(ldfltsize);
```

```
bbrowse4=new JButton("Browse");
bbrowse5=new JButton("Browse");
bbrowse6=new JButton("Browse");
bbrowse4.addActionListener(this);
bbrowse5.addActionListener(this);
bbrowse6.addActionListener(this);
```

- f. Bagian ini merupakan bagian utama dari listing program ini. Berikut adalah listing programnya.

```
public static void main(String as[])
{
    EasySplitter js=new EasySplitter();
    js.setLocation(50,50);
    js.setSize(700,470);
    js.show();
}
```

- g. bagian ini merupakan bagian untuk memproses file secara *split* atau membagi-bagi *file*. Berikut adalah potongan program.

```
public int split(String source,String
dest,String cmt,int no,int size,int
lsize,boolean del,
boolean wipe,boolean batch)
{
    String name,entry;
    long fsize,mtime;
    boolean ronly;
    File fl=new File(source);
    if(fl.exists())
    {
        name=fl.getName();
        fsize=fl.length();
        mtime=fl.lastModified();
```

```
ronly=fl.canWrite();
if(ronly)
ronly=false;
else
ronly=true;
}
else
{
    JOptionPane.showMessageDialog(
(Container)null,"File not
found","EasySplitter 1.0",JOption
Pane.OK_OPTION);
return 0;
```

- h. Bagian ini merupakan bagian untuk memproses file secara *merge* atau menggabungkan *file*. Berikut adalah potongan program.

```
public int merge(String split,String
dest,boolean del)
{
    try
    {
        File fm=new File(split);
        FileReader fr=new FileReader(fm);
        BufferedReader br=new
BufferedReader(fr);
        String ll=br.readLine();
        StringTokenizer st1=new
StringTokenizer(ll,"");
        String beg=st1.nextToken();
        if(!(beg.equals("jfs")))
        {
            JOptionPane.showMessageDialog(
(Container)null,"Not a valid
split file!","EasySplitter
1.0",JOptionPane.OK_OPTION);
            return 1;
        }
```

- i. Bagian ini merupakan *class filter1*, *class* ini berfungsi untuk membuat nama tipe data pada hasil *file* yang telah displit dengan tipe *jfs*. berikut ini adalah listing programnya. *class filter1 extends javax.swing.filechooser.FileFilter*

```
{
    public boolean accept(File fileobj)
    {
        String extension = "";
        if(fileobj.getPath().lastIndexOf('.') >
0)
            extension = fileobj.getPath
().substring(
fileobj.getPath().lastIndexOf('.')
+ 1).toLowerCase();
        if(extension != "")
            return extension.equals("jfs");
        else
            return fileobj.isDirectory();
    }
    public String getDescription()
    {
        return "EasySplitter Files (*.jfs)";
    }
}
```

- j. Bagian ini merupakan bagian dari *class help*. *Class help* digunakan untuk menampilkan bantuan bagi *user* untuk menggunakan aplikasi EasySplitter ini. Pertama harus dibuat *file* yang berisi tata cara penggunaan menggunakan teks editor lalu disimpan dengan tipe *.txt*, setelah itu

dalam program dipanggil file tersebut untuk ditampilkan. Berikut adalah listing programnya.

```
class help extends
javax.swing.JFrame
{
JEditorPane ep;
JScrollPane sp;
JButton b;
URL urls[],temp[];
JPanel p;
help()
{
super("Help - EasySplitter 1.0");
ep=new JEditorPane();
ep.setEditable(false);
ep.addHyperlinkListener(new
HyperlinkListener()
{
public void hyperlinkUpdate
(HyperlinkEvent he)
{
if(he.getEventType()==HyperlinkE
vent.EventType.ACTIVATED)
{
try
{
openpage(he.getURL());
}
catch(Exception e){}
}
});
sp=new JScrollPane(ep,Scroll
PaneConstants.VERTICAL_SCROL
LBAR_ALWAYS,ScrollPaneConsta
nts.HORIZONTAL_SCROLLBAR_
ALWAYS);
b=new JButton("OK");
b.addActionListener(new
ActionListener()
{
public void actionPerformed
(ActionEvent ae)
{
setVisible(false);
}
});
p=new JPanel(new FlowLayout());
p.add(b);
getContentPane().setLayout(new
BorderLayout());
getContentPane().add(p,Border
Layout.SOUTH);
getContentPane().add(sp, Border
Layout.CENTER);
String url1 = "file:" + System.get
Property("user.dir") +
System.getProperty("file.separator")
+
"help.txt";
```

k. Bagian ini merupakan bagian dari class about. Pada class about ini digunakan untuk menampilkan info tentang pembuat aplikasi ini. Berikut adalah listing programnya.

```
class about extends
javax.swing.JFrame
{
JLabel la1,la2,la3,la4,la5;
JTextArea ta;
JScrollPane sp;
JPanel p1,p2,p3,p4,p5;
JButton b;
JLabel img;
```

```
about()
{
super("About EasySplitter 1.0");
la1=new JLabel(" Name
: EasySplitter(File Splitter)");
la2=new JLabel(" Version : 1.0");
la3=new JLabel(" Created by: Wisnu
Nugroho");
la4 = new JLabel("
Universitas Gunadarma");
la5=new JLabel(" 2009 ");
img=new JLabel(new ImageIcon
("images/wisnu.jpg"));
ta=new JTextArea(5,30);
ta.setText(" Easy Splitter 1.0 \n"+
" Free to use and distribute to
others. \n"+
" wisnu nugroho mail:
inoe_getoo88@yahoo.co.id");
sp=new JScrollPane(ta);
b=new JButton("OK");
b.addActionListener(new
ActionListener()
{
public void actionPerformed
(ActionEvent ae)
{
setVisible(false);
}
});
```

Setelah file tersebut dibuat dan disimpan pada folder j2sdk1.4.2_17/bin dengan nama EasySplitter.java.

Setiap file sourcecode hasil kompilasi dari program java, akan diubah ke dalam bentuk yang dapat dimengerti oleh interpreter java. Interpreter java digunakan untuk mengeksekusi/ menjalankan berkas bytecode (.class). Dalam pembuatan aplikasi EasySplitter ini, jika sudah dikompilasi, maka file akan terbagi menjadi lima class. Fungsi dari masing-masing class akan dijelaskan satu persatu seperti berikut:

1. **Class Easysplitter**
Class EasySplitter merupakan kelas utama yang berfungsi untuk menu input-output (form utama). Pada class ini terjadi proses pembuatan menu seperti label, teks field, skin, proses split dan merge serta mengatur ukuran frame serta peletakkan frame saat program aplikasi dijalankan dan membuat metode penanganan kejadian. Penanganan kejadian berfungsi untuk menangani suatu kejadian ketika komponen menu dipilih sehingga bisa melakukan tugasnya.
2. **Class About**
Class about merupakan sebuah kelas yang berfungsi untuk menampilkan informasi tentang pembuat aplikasi ini. Class ini akan mengikuti kelas utamanya yaitu class easysplitter.
3. **Class help**
Class help merupakan sebuah kelas yang digunakan untuk membantu user dalam menggunakan aplikasi EasySplitter ini. Dimana didalamnya terdapat tata cara penggunaan aplikasi ini secara jelas.

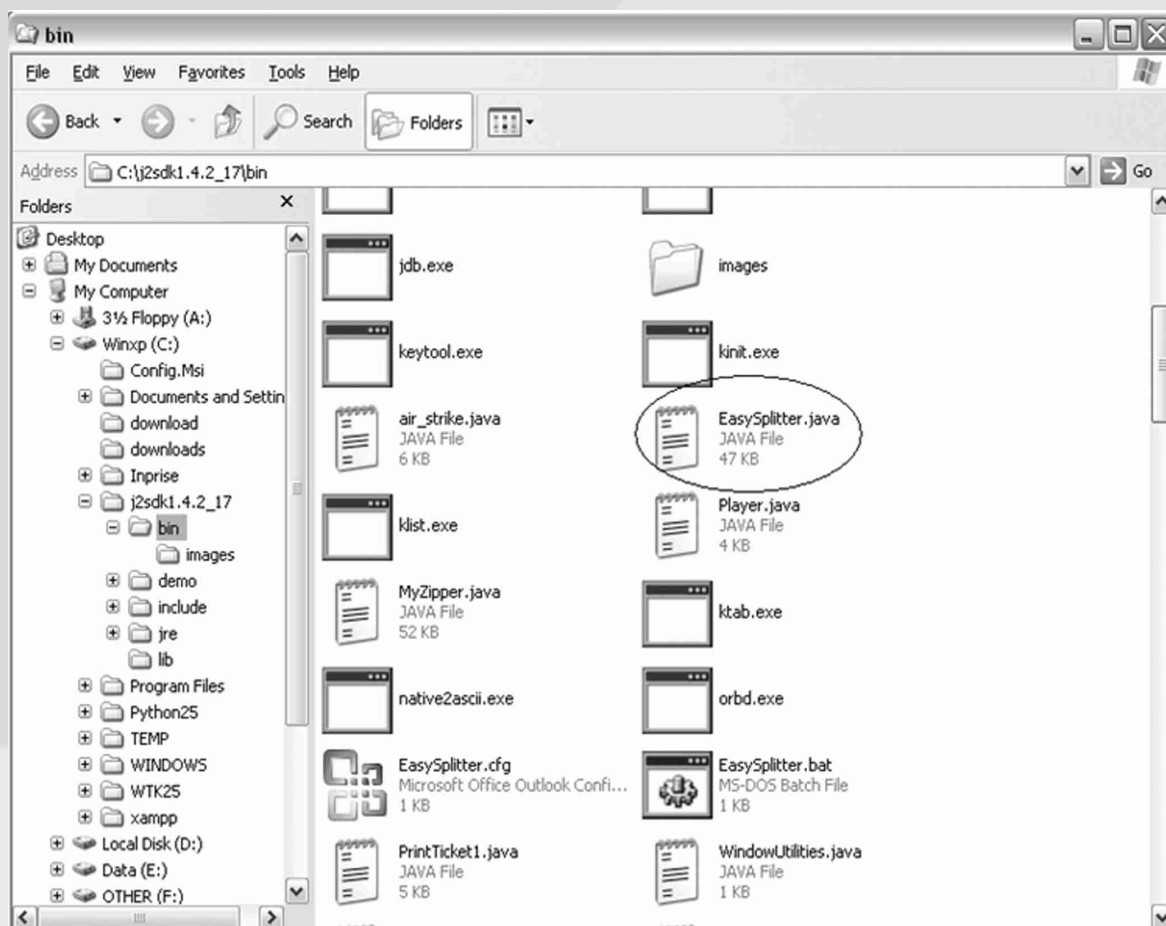
4. **Class filter1**
Class filter1 ini merupakan sebuah kelas yang digunakan untuk menyaring nama tipe data sebuah file yang akan diproses secara split. Dimana nantinya nama tipe data dari file tersebut berubah menjadi .jfs.

5. **Class sizeset**
Class sizeset ini merupakan sebuah kelas yang berfungsi untuk memodifikasi ukuran file yang akan diproses secara split. Dimana nantinya ukuran tersebut akan menjadi ukuran standar yang diterapkan selanjutnya.

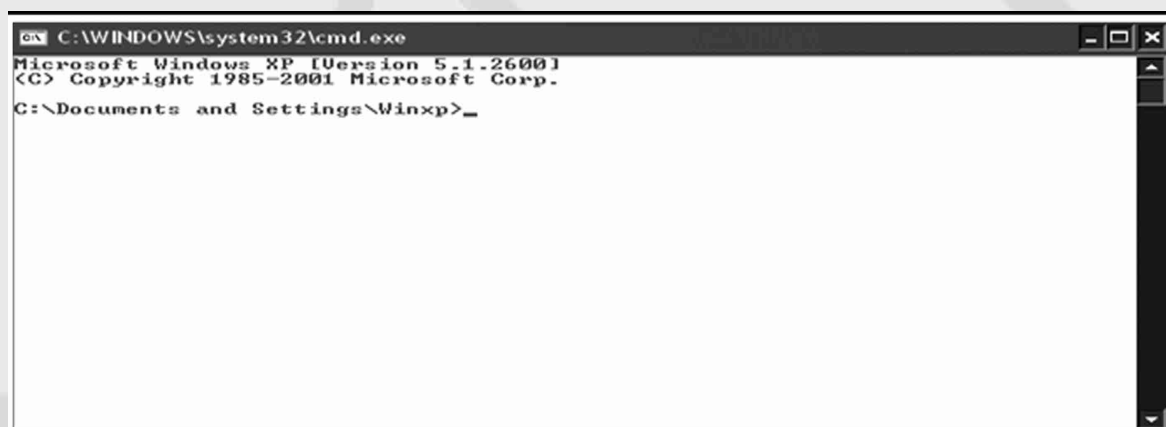
Cara Menjalankan dan Penggunaan Program Aplikasi

Langkah-langkah untuk menjalankan aplikasi ini adalah sebagai berikut :

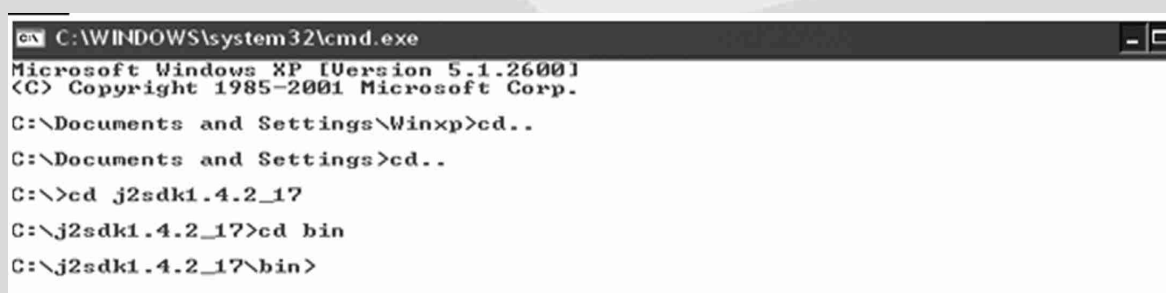
1. Tempatkan file EasySplitter.java pada folder C:\j2sdk\bin, seperti pada gambar 3.
2. Buka Command Prompt dengan mengklik Start->Run->lalu ketikkan cmd. Lalu setelah itu akan muncul command prompt seperti pada gambar 4.
3. Selanjutnya ketikkan cd.. untuk masuk ke bagian utama partisi setelah itu lanjutkan dengan mengetikkan cd j2sdk1.4.2_17 dan cd bin. Seperti pada gambar 5.
4. Selanjutnya mengetikkan javac EasySplitter.java untuk mengkompilasi file EasySplitter dan dilanjutkan dengan mengetikkan java EasySplitter. Seperti pada gambar 6.
5. Selanjutnya akan keluar menu utama EasySplitter. Selanjutnya akan dilakukan proses split atau membagi-bagi file dengan cara mengambil file yang akan diproses dengan cara menekan tombol add setelah itu akan muncul form untuk mengambil file seperti pada gambar 7 dan 8. dan juga tentukan folder tujuan (destination folder).
6. Setelah memilih file yang akan diproses, selanjutnya tekan tombol split. Berikut ini merupakan hasil split file yang telah dilakukan. Tipe dari file tersebut berubah dari .doc menjadi .jfs. File tersebut terdapat pada folder split seperti pada gambar 9.
7. Selanjutnya adalah melakukan proses merge atau menggabungkan kembali file-file yang sebelumnya terbagi-bagi. Pertama tentukan file yang akan di-merge dengan menekan tombol add setelah itu pilih file seperti pada gambar 10.. setelah itu menekan tombol give detail untuk mengetahui detail file dan dilanjutkan dengan menekan tombol merge. Selanjutnya hasil file yang sebelumnya bertipe .jfs akan kembali ke tipe asalnya yaitu .doc



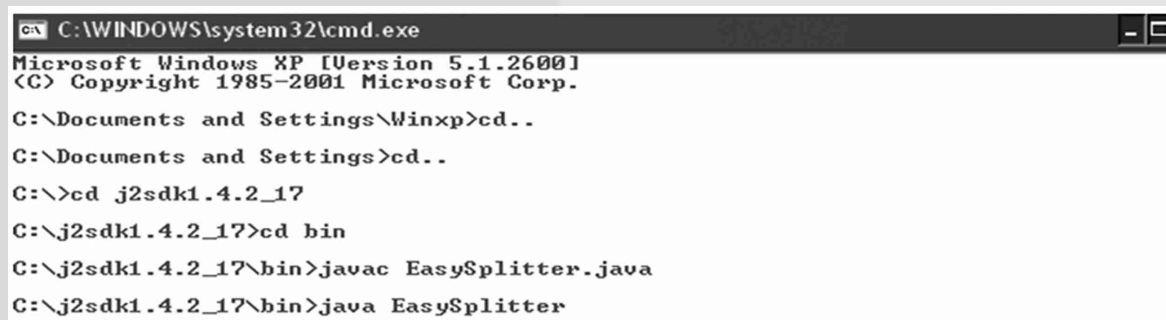
Gambar 3. File EasySplitter.java



Gambar 4. Command Prompt



Gambar 5. Command Prompt



Gambar 6. Compile EasySplitter



Gambar 7. Mengambil File

seperti pada gambar 10 dan hasil dari *merge* tersebut dapat dilihat pada folder *mergedfiles* seperti pada gambar 11.

8. Selanjutnya untuk menentukan *folder split* dan *merge* tujuan dan juga menentukan ukuran *file* yang akan di *split* diatur pada bagian tab *option* seperti pada gambar 13.
9. Untuk melihat tata cara penggunaan aplikasi ini dapat dilihat dengan cara menekan tombol *help* maka akan tampil form seperti gambar 14.
10. Untuk melihat informasi tentang profil pembuat aplikasi dapat dilihat dengan menekan tombol *about*. Selanjutnya akan tampil form seperti pada gambar 15.

Spesifikasi *Hardware* dan *Software*

Dalam pembuatan program aplikasi Easy Splitter ini, dibutuhkan dukungan *hardware* yang memadai agar *software* yang digunakan nantinya dapat berjalan dengan optimal. Adapun spesifikasi *hardware* dan *software* yang diperlukan, yaitu :

Kesimpulan

Dengan adanya aplikasi Easy Splitter ini, maka pemecahan dan penggabungan file dapat lebih mudah dilakukan tanpa mengubah bentuk dan struktur dari file itu sendiri. Sehingga dapat mengatasi masalah kekurangan kapasitas penyimpanan. Selain itu program ini dapat berjalan di semua sistem operasi dan dapat dilakukan pada hampir semua jenis file (misalnya :file teks, gambar, audio dan video,.....) sehingga merupakan kemudahan bagi setiap pengguna komputer.

Daftar Pustaka

Ady Wicaksono, *Dasar-Dasar Pemrograman Java 2*, PT Elex Media Komputindo, Jakarta, 2002.

Indrajani, *Pemrograman Berorientasi Objek dengan Java*, PT Elex Media Komputindo, Jakarta, 2004.

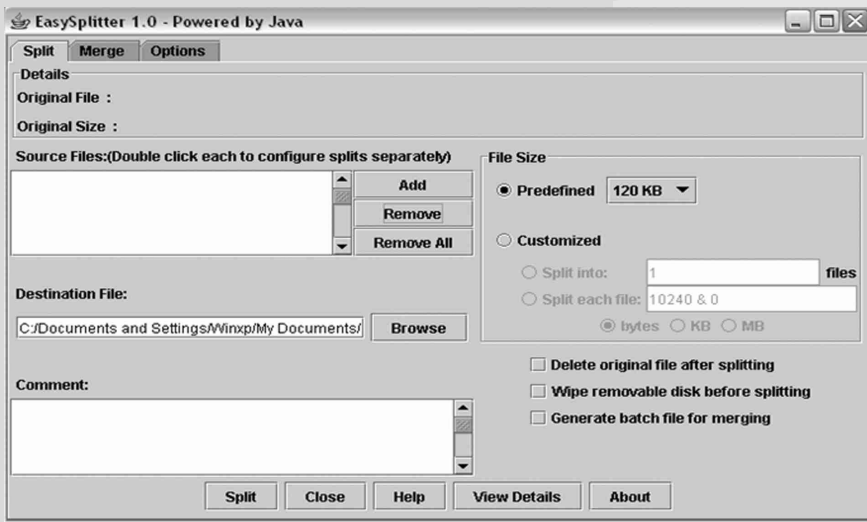
Rijanto Tosin, *Flowchart Untuk Siswa dan Mahasiswa*, DINASTINDO, Jakarta, 1994.

Rangsang Purnama, *Tuntunan Pemograman Java*, Prestasi Pustaka, Jakarta, 2005.

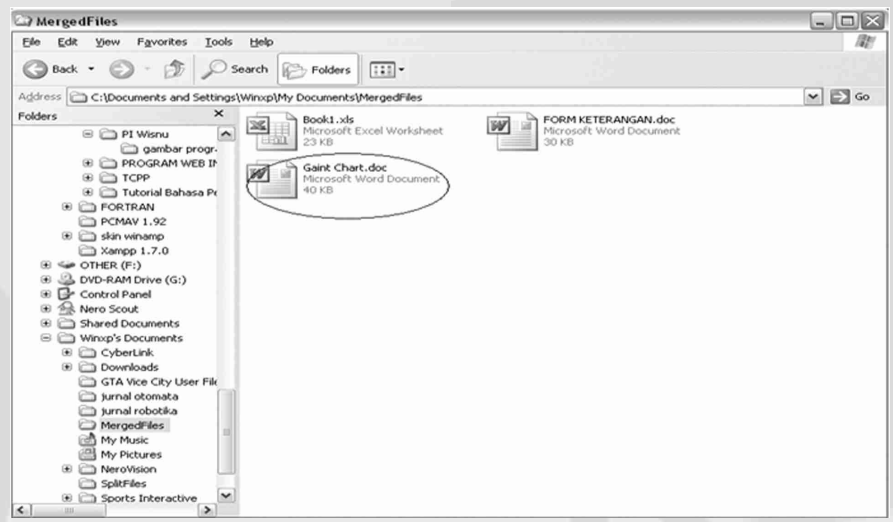
Wisnu Nugroho, *Aplikasi Easy Splitter Dengan Java*, 2010

www.onno.vlsm.org, Akhmad Daniel Sembiring, *Sekilas Java*, Jakarta, 6 Juni 2006.

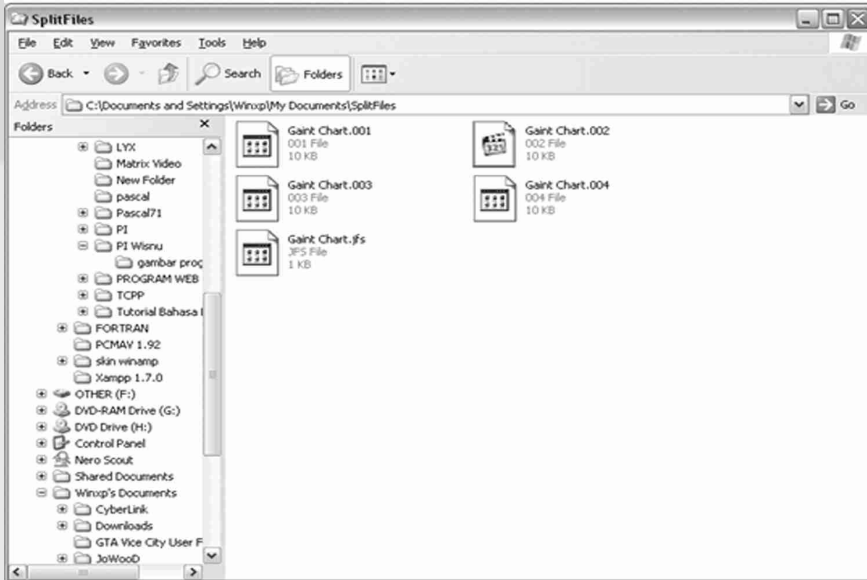
[Http://www.ilmukomputer.com/berse/ri/eko-java/index.php](http://www.ilmukomputer.com/berse/ri/eko-java/index.php), Mei 2006. *hp*, Mei 2006



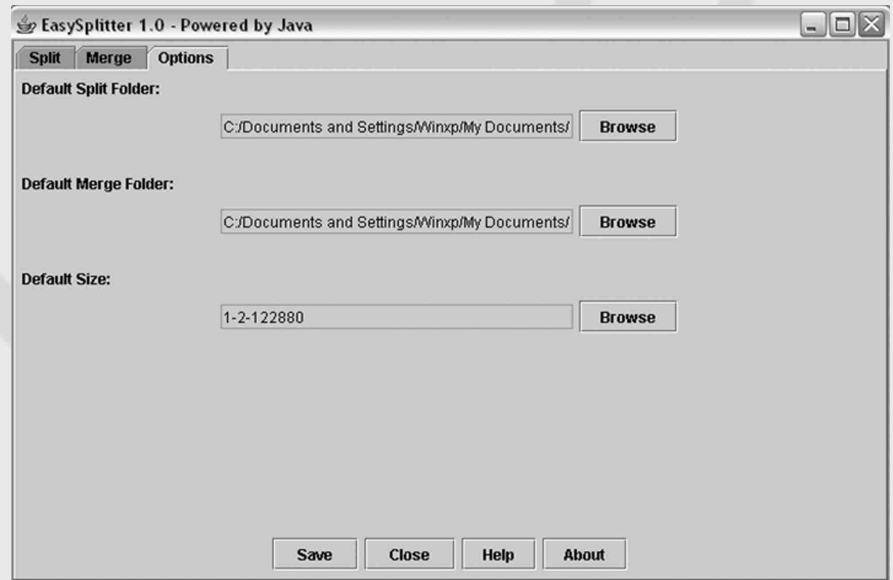
Gambar 8. Form Utama Split



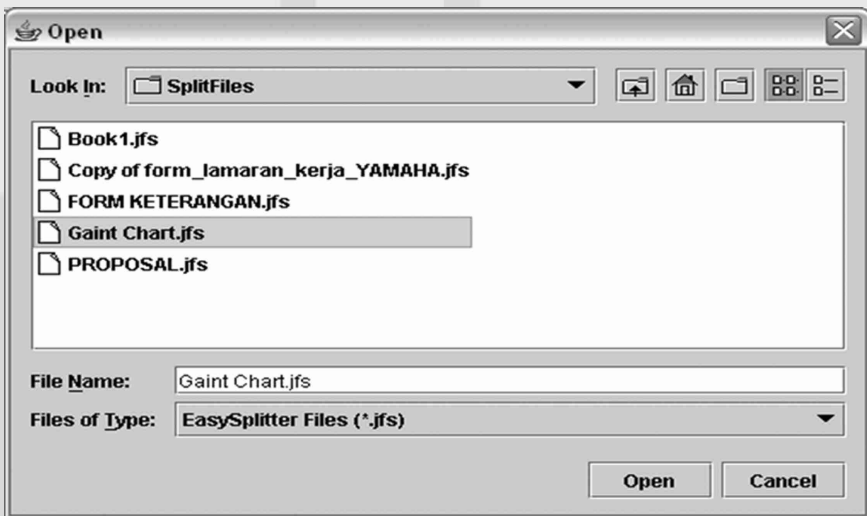
Gambar 12. Hasil Proses Merge



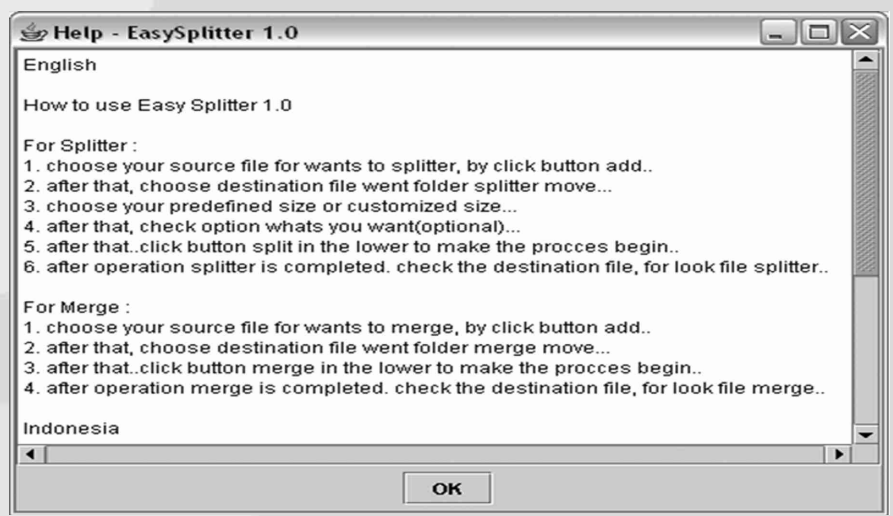
Gambar 9. Hasil Proses Splitter pada file Gaintchart



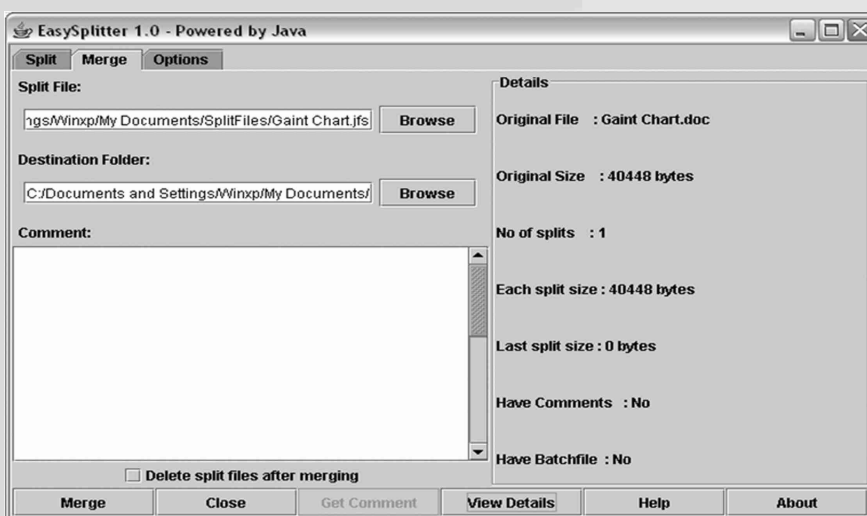
Gambar 13. Tab Option



Gambar 10. Mencari File .jfs



Gambar 14. Form Help



Gambar 11. Form Tab Merge



Gambar 15. Form About

Tabel 1. Kebutuhan Minimum Hardware

Kebutuhan	Minimum
Processor	Pentium III 800 MHz keatas
RAM	256 MB
Ruang Hardisk yang kosong	900 MB
VGA	64 MB

Tabel 2. Kebutuhan Minimum Software

Kebutuhan	Software yang digunakan
Software	- Microsoft Windows NT / ME / 2000 / XP - Java JDK 1.6.0