

OPERASI FILE READ - WRITE

Yuliana Setiowati
Politeknik Elektronika Negeri Surabaya

Membaca File

- Untuk membaca file, diperlukan langkah-langkah:
 - Buat object File dengan memasukkan parameter nama file atau nama file lengkap dengan path.
 - Periksa apakah file tersebut ada dan tipenya adalah file (bukan direktori).
 - Bungkus object File dengan object FileReader, selanjutnya bungkus lagi dengan object BufferedReader (BufferedReader digunakan untuk mempercepat proses pembacaan)
 - Baca file dengan method `readLine()`, proses pembacaan satu baris, jika tidak ada lagi baris yang dibaca akan mengembalikan null.

Membaca File

- Buatlah sebuah class Read untuk membaca file *.txt. Simpanlah file *.txt diluar folder src (didalam project yang ada)

```
import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.io.IOException;

public class Read {
    public static void main(String args[]){
        //proses pembacaan
        File file = new File("Data21.txt" );
        if (!file.exists() && !file.isFile()){
            System.out.println("File tidak ditemukan");
            System.exit(-1);
        }
        try{
            BufferedReader fileReader = new BufferedReader(new FileReader(file));
            while(true){
                String line = fileReader.readLine();
                if (line == null)
                    break ;
                System.out.println(line);
            }
        }catch(IOException io){
        }
    }
}
```

Membaca File

- Output program
- Jika file tidak ditemukan

```
File tidak ditemukan  
Java Result: -1  
BUILD SUCCESSFUL (total time: 0 seconds)
```

- File ditemukan

```
Modul Pemrograman Berbasis Object 2  
Politeknik Elektronika Negeri Surabaya  
BUILD SUCCESSFUL (total time: 0 seconds)
```

Menulis File

- Untuk menulis file, diperlukan langkah-langkah:
 - Buat object File dengan memasukkan parameter nama file atau nama file lengkap dengan path.
 - Periksa apakah file tersebut ada dan tipenya adalah file (bukan direktori).
 - Bungkus object File dengan object FileWriter, selanjutnya bungkus lagi dengan object PrintWriter.
 - Inputkan kalimat yang akan dituliskan kedalam file(bisa menggunakan JOptionPane).
 - Gunakan method flush() untuk memaksa sistem mentransfer isi buffer di memori ke file.
 - Gunakan method close() untuk menutup object PrintWriter.

Menulis File

```
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.io.PrintWriter;
import javax.swing.JOptionPane;

public class Write {
    public static void main(String args[]){
        PrintWriter fout = null ;
        File file = new File("Data22.txt");

        if (!file.exists() && !file.isFile()){
            System.out.println("File tidak ditemukan");
            System.exit(-1);
        }

        try{
            fout = new PrintWriter(new FileWriter(file));
            String line = JOptionPane.showInputDialog("Tulis Kalimat");
            fout.println(line);
            fout.flush();
        } catch (IOException e){
            fout.close();
        }
    }
}
```

Menulis File

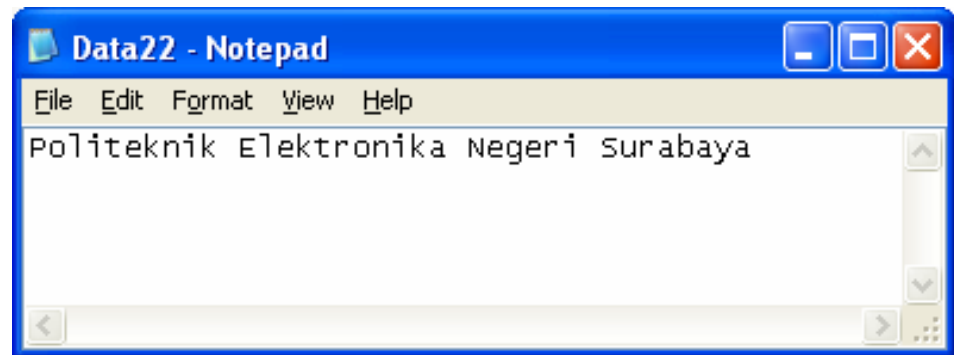
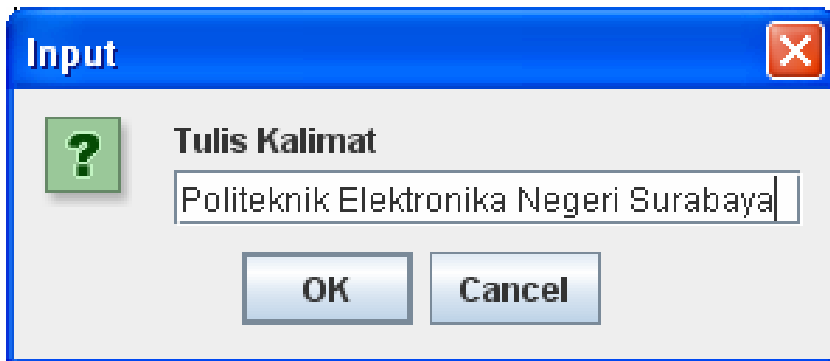
- Jika file tidak ditemukan

```
File tidak ditemukan
```

```
Java Result: -1
```

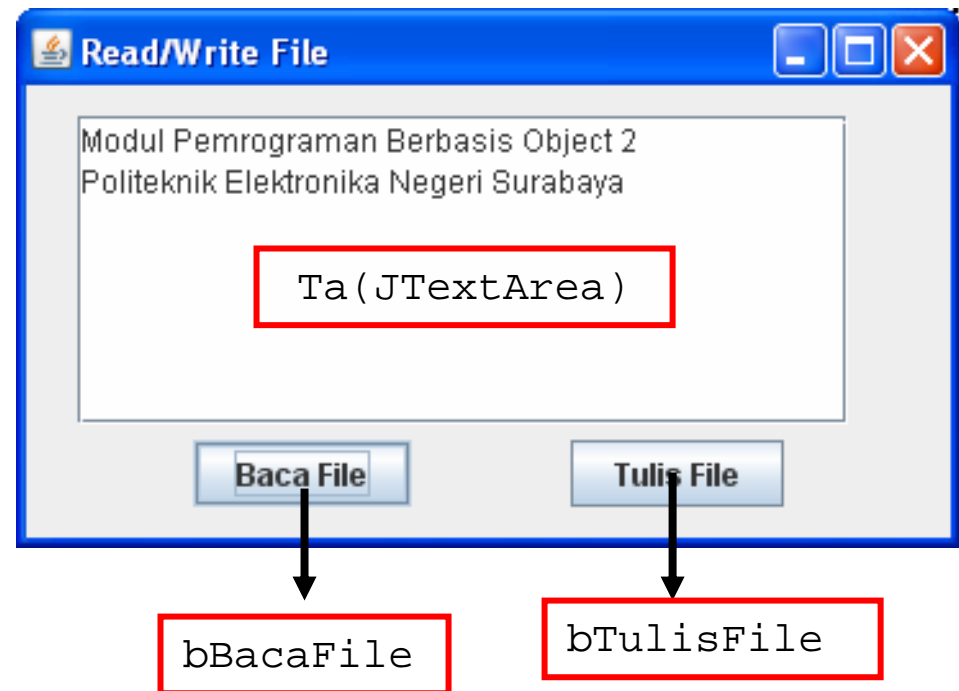
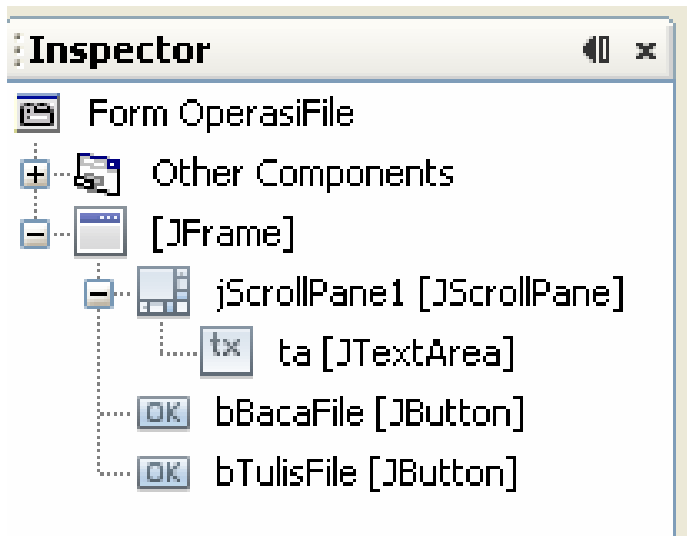
```
BUILD SUCCESSFUL (total time: 0 seconds)
```

- Jika file ditemukan
- Inputkan kalimat yang akan ditulis ke file menggunakan JOptionPane. Lihat hasil pada file txt



Aplikasi Read – Write File

- Buatlah aplikasi GUI sederhana untuk membaca dan menulis ke file.



Aplikasi Read – Write File

- Buatlah fungsi sendiri BacaFile()

```
void BacaFile(){
    //proses pembacaan
    File file = new File("Data21.txt");
    if (!file.exists() && !file.isFile()){
        ta.append("File tidak ditemukan");
        System.exit(-1);
    }
    try{
        BufferedReader fileReader = new BufferedReader(new FileReader(file));
        while(true){
            String line = fileReader.readLine();
            if (line == null)
                break ;
            ta.append(line+"\n");
        }
    }catch(IOException io){
    }
}
```

Aplikasi Read – Write File

- Buatlah fungsi sendiri TulisFile()

```
void TulisFile(){
    PrintWriter fout = null ;
    File file = new File("Data21.txt");
    if (!file.exists() && !file.isFile()){
        ta.append("File tidak ditemukan");
        System.exit(-1);
    }

    try{
        fout = new PrintWriter(new FileWriter(file));
        String line = ta.getText() ;
        fout.println(line);
        fout.flush();

    } catch(IOException e){
        fout.close();
    }
}
```

Aplikasi Read – Write File

- Double klik pada button bBacaFile

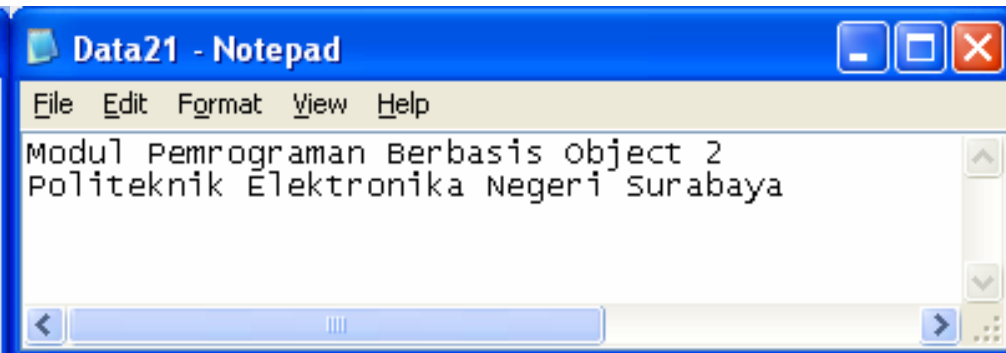
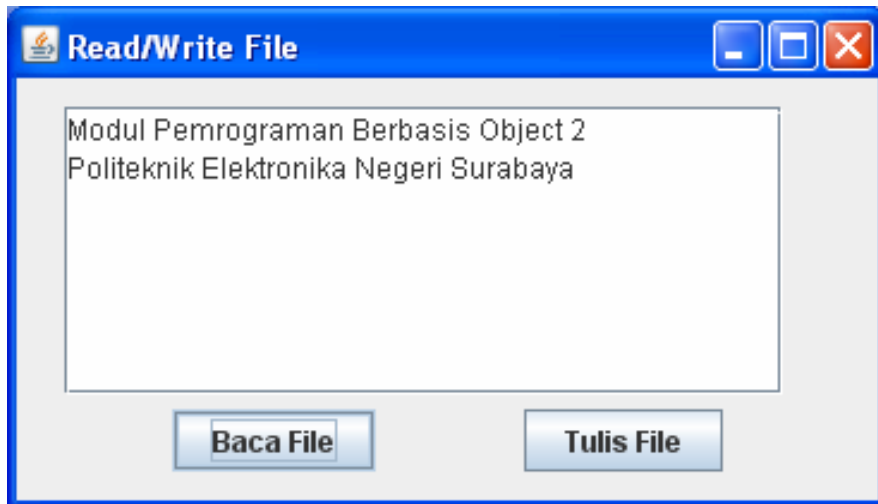
```
private void bBacaFileActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    this.BacaFile();  
}
```

- Double klik pada button bTulisFile

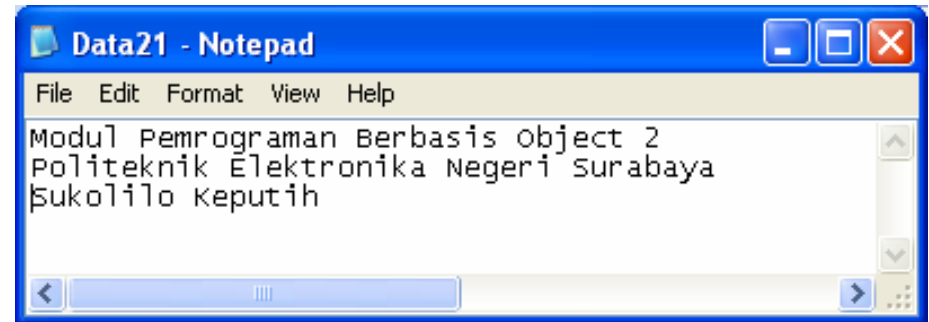
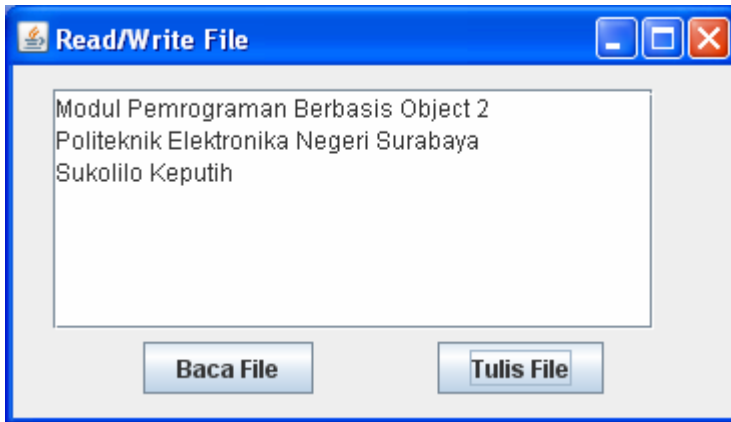
```
private void bTulisFileActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    TulisFile();  
}
```

Aplikasi Read – Write File

- Tekan button Baca File, maka pada textarea akan menampilkan text yang terdapat pada file txt
- Selanjutnya tambahkan text sembarang, tekan button Tulis File (text yang terdapat pada textarea akan tersimpan dalam file txt).

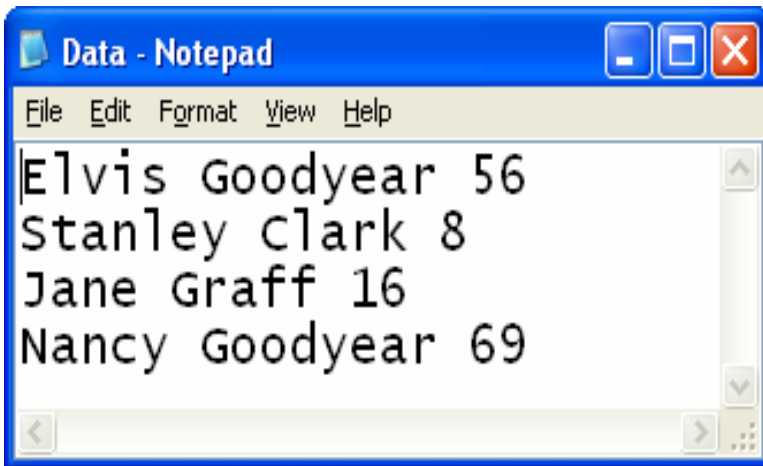


Aplikasi Read – Write File



Memparsing Data

- Terdapat file *.txt, setiap baris menyatakan sebuah object dari class Person, selanjutnya object-object ini akan disimpan dalam object Vector.



```
Data - Notepad
File Edit Format View Help
Elvis Goodyear 56
Stanley Clark 8
Jane Graff 16
Nancy Goodyear 69
```

firstname	lastname	Age
Elvis	Goodyear	56
Stanley	Clark	8
Jane	Graff	16
Nancy	Goodyear	69

Object dari class Person,
semua object tersimpan
dalam object Vector

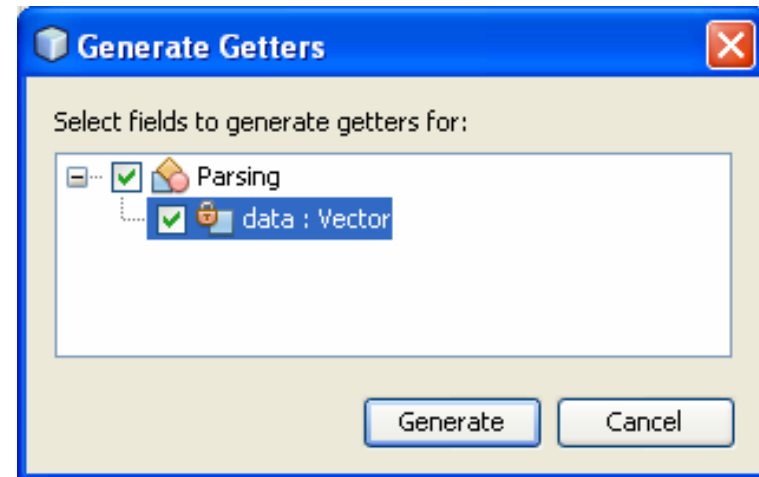
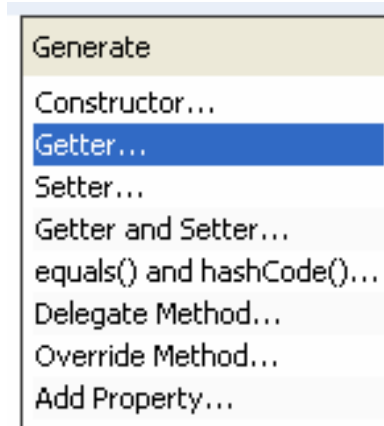
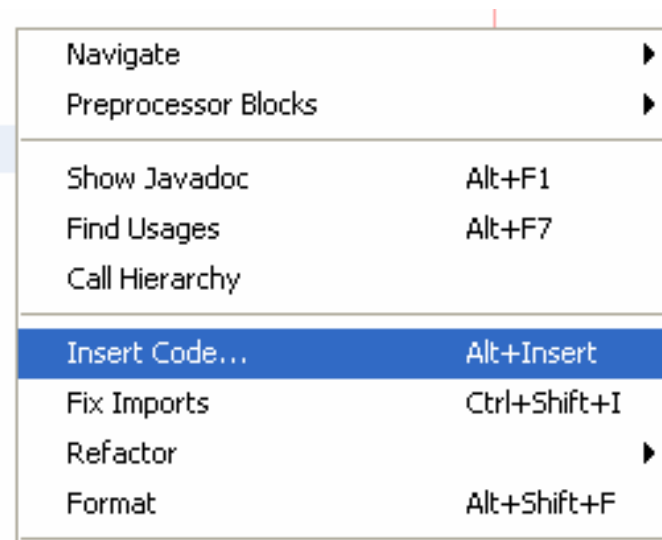
Memparsing Data

- Tambahkan pada fungsi BacaFile() pada hal 9, untuk memarsing data

```
//membaca file dan melakukan parsing
void BacaFile(){
    //proses pembacaan
    File file = new File("Data.txt" );
    if (!file.exists() && !file.isFile()){
        System.out.println("File tidak ditemukan");
        System.exit(-1);
    }
    try{
        BufferedReader fileReader = new BufferedReader(new FileReader(file));
        while(true){
            String line = fileReader.readLine();
            if (line == null)
                break ;
            String h[] = line.split(" ");
            Person p = new Person();
            p.setFirstName(h[0]);
            p.setLastName(h[1]);
            p.setAge(Integer.parseInt(h[2]));
            data.add(p);
        }
    }catch(IOException io){
    }
}
```

Memparsing Data

- Tambahkan pada class Parsing object Vector data dan fungsi getData() dengan cara klik kanan → insert code → Getter → pilih data:Vector



Memparsing Data

```
public class Parsing {  
    private Vector data ;  
  
    public Parsing() {  
        data = new Vector();  
    }  
    public Vector getData() {  
        return data;  
    }  
}
```

Memparsing Data (Cara I)

- Pembacaan file txt dilakukan perbaris (disimpan dalam var line bertipe String)
- String line dipecah berdasarkan spasi " " (tergantung parameter inputan), hasil disimpan dalam array String h.
- Buat object Person p, dan set firstName dengan h[0], set lastName dengan h[1], sedangkan untuk mengeset age, ubah dulu String h[3] menjadi int.

```
while(true){  
    String line = fileReader.readLine();  
    if (line == null)  
        break ;  
    String h[] = line.split(" ");  
    Person p = new Person();  
    p.setFirstName(h[0]);  
    p.setLastName(h[1]);  
    p.setAge(Integer.parseInt(h[2]));  
    data.add(p);  
}
```

Memparsing Data (Cara II)

- Pembacaan file txt dilakukan perbaris (disimpan dalam var line bertipe String).
- Untuk memecah String gunakan object StringTokenizer dengan pemisah spasi (" "), misal st.
- Buat object Person p.
- Set firstName, lastName dan age dengan membaca kata/token berikutnya menggunakan st.nextToken().

```
while(true){
    String line = fileReader.readLine();
    if (line == null)
        break ;
    StringTokenizer st = new StringTokenizer(line, " ");
    while(st.hasMoreTokens()){
        Person p = new Person();
        p.setFirstName(st.nextToken());
        p.setLastName(st.nextToken());
        p.setAge(Integer.parseInt(st.nextToken()));
        data.add(p);
    }
}
```

Memparsing Data

- Pada fungsi utama panggil fungsi BacaFile(), hasil dari pembacaan file berupa object Vector d, Vector d berisi object Person p yang ditampilkan dengan cara p.toString().
- **Jangan lupa** tambahkan fungsi toString pada class Person. Fungsi ini digunakan untuk mengubah object menjadi String.

```
public static void main(String args[]){  
    Parsing parse = new Parsing();  
    parse.BacaFile();  
    Vector d = parse.getData() ;  
  
    for(int i=0;i<d.size();i++){  
        Person p = (Person) (d.get(i));  
        System.out.println(p.toString());  
    }  
}
```

```
@Override  
public String toString() {  
    return firstName+" "+lastName+" "+age ;  
}
```

Memparsing Data

- Output program

```
run:  
Elvis Goodyear 56  
Stanley Clark 8  
Jane Craff 16  
Nancy Goodyear 69
```