MODUL PEMOGRAMAN BERORIENTASI

OBJEK 2

I/O STREAM PADA JAVA

Oleh: Rudhi Wahyudi Febrianto

Universitas Teknologi Bandung

BAB I

PENDAHULUAN

1.1. Latar Belakang

Saat ini tidak dapat dipungkiri bahwa perkembangan zaman yang mengglobal dalam berbagai bidang, hampir seleruh hal telah berkembang termasuk didalamnya adalah bidang teknologi. Dalam teknologi ini kita mengenal pemrograman komputer yang didalamnya terdapat bahasa pemrograman, salah satu dari bahasa tersebut adalah bahasa Java.

Java adalah bahasa pemrograman yang dapat membuat seluruh bentuk aplikasi, desktop, web, mobile dan lain sebagainya, sebagaimana dibuat dengan menggunakan bahasa pemrograman lainnya. tidak heran jika banyak masyarakt dunia menggunakan fasilitas teknologi informasi dan komunikasi tidak asing mendengar bahasa pemrograman ini, dikarenakan hampir sebagian besar produk informasi dan telekomunikasi bertaraf modern bermain dipasar teknologi dengan membawa java sebagai bahasa pemrograman mereka.

Pemrograman Java awalnya dibuat oleh James Gosling pada tahun 1995 sebagai bagian dari Sun Microsystem Java Platform. Sintaks java banyak diturunkan dari bahasa C dan C++ tetapi lebih sederhana, ketat dan memiliki akses ke Sistem Operasi yang lebih terbatas. Hal ini dekarenakan Java ditujukan untuk bahasa pemrograman yang cukup sederhana dipelajari dan mudah dibaca.

Java merupakan bahasa tingkat tinggi (*High Level Language*), maksud dari bahasa tingkat tinggi adalah suatu bahasa pemrograman yang dibuat tidak untuk dapat dimengerti oleh mesin atau *assembler*, namun juga dapat dipahami oleh manusia. Didalam penggunaan bahasa pemrograman Java dikenal beberapa operator antara lain *Class, Object, Constructor, Inheritance* dan lain sebagainya.

1.2.Tujuan

Adapun tujuan dari praktikum ini adalah:

1. Mampu memahami konsep *I/O Stream*.
2. Mampu membuat program yang mengalirkan data melalui stream.

BAB II

DASAR TEORI

2.1. Pengertian *Stream*

Stream merupakan keadaan dari sebuah *file* atau sebuah *device* yang memungkinkan rangkaian item dapat dibaca maupun ditulis. Secara umum, *stream* dalam java dibagi menjadi dua, yaitu *byte stream* dan *character stream. Byte stream* digunakan untuk operasi I/O yang menggunakan data biner (byte), sedangkan *character byte* digunakan untuk menangani operasi I/O yang menggunakan *character.*

2.1.1. *InputStream* dan *OutStream*

Terdapat 2 class abstrak yang dirancang sebagai kelas induk untuk kelas-kelas yang termasuk kategori stream byte, yaitu class InputStream dan OutputStream.Class InputStream dan OutputStream merupakan class induk yang digunakan untuk menangani operasi input/output (I/O). Kedua class ini adalah abstract class sehingga tidak dapat digunakan secara langsung, tetapi dapat digunakan class class turunan dari kedua class ini.

Class-class turunan (subclass-subclass) dari class InputStream antara lain:

- ByteArrayInputStream
- FileInputStream
- FilterInputStream
- ObjectInputStream
- PipedInputStream

Class-class turunan (subclass-subclass) dari class OutputStream antara lain:

- ByteArrayOutputStream
- FileOutputStream
- FilterOutputStream
- ObjectOutputStream
- PipedOutputStream.

BAB III
METODOLOGI PERCOBAAN

3.1. Alat dan Bahan

1. PC/Laptop
2. NetBeans IDE
3. Java

3.2. Algoritma

3.2.1. Algoritma Percobaan Menulis Data ke *File Binary*

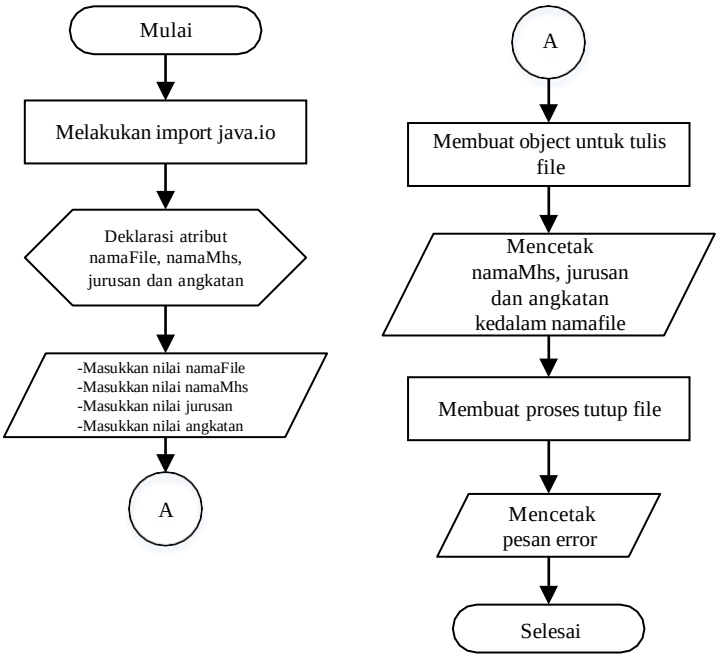
1. Mulai.
2. Lakukan perintah pemanggilan java.io
3. Lakukan proses tulis data
4. Lakukan proses tutup file
5. Lakukan pembuatan pesan error jika terjadi kesalahan.
6. Selesai.

3.2.2. Algoritma Percobaan Membaca Data dari *File Binary*

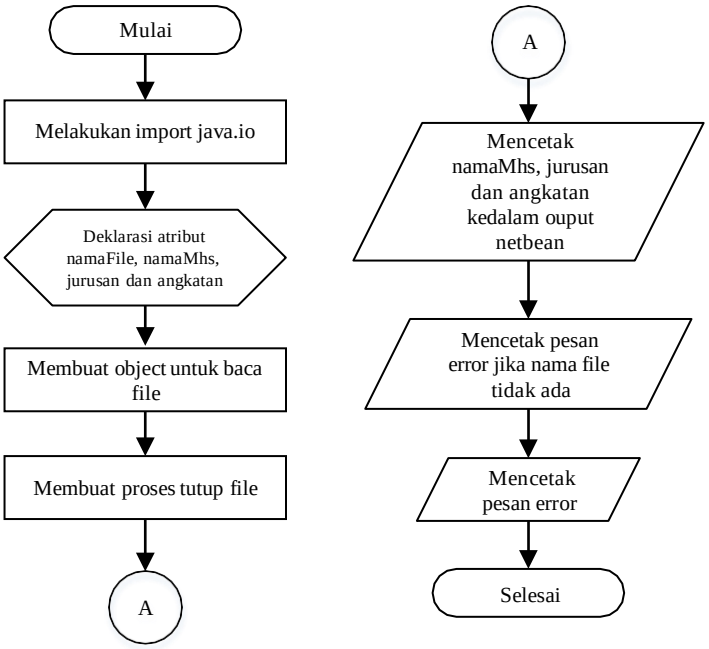
1. Mulai
2. Lakukan perintah pemanggilan java.io
3. Lakukan proses baca data
4. Lakukan proses tutup file
5. Lakukan pembuatan pesan error jika tidak terdapat file
6. Lakukan pembuatan pesan error jika terjadi kesalahan
7. Selesai

3.3. Flowchart (bagan alir)

3.3.1. Flowchart Percobaan Menulis Data ke *File Binary*



3.3.2. Flowchart Percobaan Membaca Data dari *File Binary*



BAB IV

HASIL DAN ANALISA

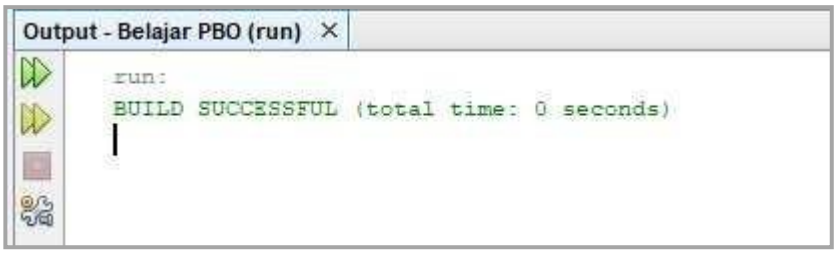
4.1. Hasil Percobaan

4.1.1. Percobaan Menulis Data ke *File Binary*

4.1.1.1. List Program Percobaan Menulis Data ke *File Binary*

```
package praktikum4;
import java.io.*;
public class Praktikum4_1 {
    public static void main (String[]
args)
    throws IOException {
        String namaFile = "PBO.txt";
        String namaMhs = "Adam Yordan";
        String jurusan = "Teknik
Informatika";
        int angkatan = 2017;
        FileOutputStream outFile = new
            FileOutputStream(namaFile);
        try {
            DataOutputStream outStream =
new

DataOutputStream(outFile);

            outStream.writeUTF(namaMhs);
            outStream.writeUTF(jurusan);
            outStream.writeInt(angkatan);
            outStream.close();
        }
        catch (IOException e){
            System.out.println("IOERROR:"
+ e.getMessage() + "\n");
        }
    }
}
```

4.1.1.2. Hasil Program Percobaan Menulis Data ke *File Binary*



Gambar 4.1. Hasil Program Percobaan Menulis Data ke *File Binary*

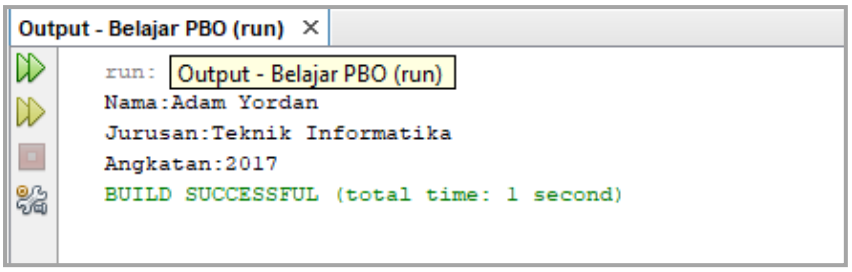
4.1.2. Percobaan Membaca Data dari *File Binary*

4.1.2.1. List Program Percobaan Membaca Data dari *File Binary*

```
package praktikum4;
import java.io.*;

public class Praktikum4_2 {
public static void main(String[]
args)throws IOException{
    String namaFile = "PBO.txt";
    String  namaMhs,  jurusan;
    int angkatan;
    try{
        FileInputStream inFile = new
            FileInputStream (namaFile);
        DataInputStream inStream = new
            DataInputStream(inFile);
        namaMhs = inStream.readUTF();

        jurusan = inStream.readUTF();
        angkatan = inStream.readInt();
        inStream.close();

System.out.println("Nama:"+namaMhs+"\nJuru
san:"+jurusan+"\nAngkatan:"+angkatan);
    }
    catch (FileNotFoundException e){

System.out.println("File"+namaFile+"TidakA
da !\n");
    }
```

```
        catch (IOException ex){

    System.out.println("IOERROR:"+ex.getMessag
    e()+"\n");
        }
    }
    }
```

4.1.2.2. Hasil Program Percobaan Membaca Data dari *File Binary*



Gambar 4.2. Hasil Program Percobaan Membaca Data dari *File Binary*

4.2. Analisa

4.2.1. Analisa Percobaan Menulis Data ke File Binary

Pada percobaan menulis data ke file binary dapat di analisa bahwa hasil tersebut didapatkan dari penggunaan import java.io yang dimana akan memungkinkan untuk melakukan input dan output file. Pada program tersebut terdapat perintah throws dimana perintah ini digunakan untuk menampung atribut dan nilai dari namaFile, namaMhs, jurusan dan angkatan, namaFile merupakan wadah nantinya yang digunakan untuk menampung perintah untuk tulis data dengan ekstensi .txt. pada program ini terdapat perintah try dimana didalam perintah ini digunakan untuk menulis nilai yang telah dimasukkan atribut kedalam PBO.txt dimana menggunakan sintaks writeUTF yang telah diakses oleh object file tersebut. Perintah yang terakhir merupakan perintah pasangan dari try yaitu perintah catch dimana perintah ini digunakan untuk mencetak pesan

error jika terdapat kesalahan dalam penulisan, dan pada hasil yang telah dicoba pada saat proses menjalankan file, file berhasil dijalankan menandakan nilai dari atribut telah berhasil ditulis kedalam file PBO.txt.

4.2.2. Analisa Percobaan Membaca Data Data dari File Binary

Pada percobaan membaca data dari file binary dapat dianalsisa bahwa hasil tersebut didapatkan dari penggunaan import java.io yang dimana akan memungkinkan untuk melakukan input dan output file. Pada program tersebut terdapat perintah throws dimana perintah ini digunakan untuk mendeklarasikan atribut namaFile, namaMhs, jurusan dan angkatan. pada program tersebut terdapat perintah try dimana digunakan untuk membuat object yang akan membaca atribut namaFile dan terdapat perintah lainnya untuk membaca namaMhs, jurusan, angkatan dan melakukan proses tutup file, kemudian terdapat perintah untuk mencetak hasil dari namaMhs, jurusan dan angkatan. yang terakhir adalah terdapat perintah catch yaitu pasangan dari perintah try dimana perintah ini digunakan untuk mecetak pesan error jika file tidak ada dan jika terjadi sebuah kesalahan dalam penulisan.

4.3. Latihan

4.3.1. List Program

File class manusia

```
package Latihan;
public class manusia extends mahasiswa {
 private String nim;
 private String program_studi;
 private String jurusan;
 manusia(String nama, String alamat,String
jenis_kelamin){
 super(nama,alamat,jenis_kelamin); }
```

```
 manusia(String nim,String nama,String
alamat,String jurusan,String
program_studi,String jenis_kelamin ){
 super.setNama(nama);
 super.setAlamat(alamat);
 super.setJK(jenis_kelamin);
 this.nim=nim;
 this.program_studi=program_studi;
 this.jurusan=jurusan; }
 public String getNama(){
 return super.getNama();}
 public String getAlamat(){
 return super.getAlamat();}
 public String getJK(){
 return super.getJK();}
 public String getNim(){
 return nim;}
 public String getProgram_studi(){
 return program_studi;}
 public String getJurusan(){
 return jurusan;
     }
}
```

File class mahasiswa

```
  package Latihan;
  public class mahasiswa {
  private String nama, alamat, jenis_kelamin;
  mahasiswa(){}
```

```
      mahasiswa(String nama,String alamat,String
      jenis_kelamin){
       this.nama=nama;
       this.alamat=alamat;
       this.jenis_kelamin=jenis_kelamin;}
      public void setNama(String n){
      nama=n;}
      public void setAlamat(String a){
       alamat=a;}
      public void setJK(String j){
       jenis_kelamin=j;}
      public String getNama(){
       return nama;}
      public String getAlamat(){
       return alamat;}
      public String getJK(){
       return jenis_kelamin;
          }
}
```

File class utama

```
      package Latihan;
      import java.io.*;
      import java.util.ArrayList;
      public class utama {
       static ArrayList listDataMahasiswa = new
      ArrayList();
       public static void main(String[] args)throws
      Exception{
       BufferedReader br = new BufferedReader(new
```

```
InputStreamReader(System.in));
 manusia mhs1= new manusia("","","");
 manusia mhs2= new manusia("","","");
 while(true){
 System.out.println("===========Menu---------
--");
 System.out.println("1. Input Data Mahasiswa
1");
 System.out.println("2. Input Data Mahasiswa
2");
 System.out.println("3. Lihat Data
Mahasiswa");
 System.out.println("4. Keluar Program");
System.out.println("************************
");
 System.out.print("Pilihan Anda : ");
 int pilih = Integer.parseInt(br.readLine());
 System.out.println();
 switch(pilih){
 case 1:
 System.out.print("Nama : ");
 String nama = br.readLine();
 System.out.print("Alamat : ");
 String alamat = br.readLine();
 System.out.print("Jenis Kelamin (L/P) : ");
 String jenis_kelamin = br.readLine();
 mhs1 = new manusia(nama, alamat,
jenis_kelamin);
 System.out.println();
 break;
```

```
        case 2:
        System.out.print("Nama : ");
        nama = br.readLine();
        System.out.print("Alamat : ");
        alamat = br.readLine();
        System.out.print("Jenis Kelamin (L/P) : ");
        jenis_kelamin = br.readLine();
        mhs2    =    new    manusia(nama,    alamat,
      jenis_kelamin);
        System.out.println();
        break;
        case 3:
        System.out.println("Nama         :         "+
      mhs1.getNama());
        System.out.println("Alamat        :        "+
      mhs1.getAlamat());
        System.out.println("Jenis   Kelamin   :   "+
      mhs1.getJK());
        System.out.println();
        System.out.println("Nama         :         "+
      mhs2.getNama());
        System.out.println("Alamat        :        "+
      mhs2.getAlamat());
        System.out.println("Jenis   Kelamin   :   "+
      mhs2.getJK());
        break;
        case 4 :
        System.exit(0);
                  }
              }
```

```
            }

}
```

4.3.2. Hasil

```
run:
==========Menu----------
1. Input Data Mahasiswa 1
2. Input Data Mahasiswa 2
3. Lihat Data Mahasiswa
4. Keluar Program
*************************
Pilihan Anda : 1

Nama : Adam Y
Alamat : Kampung Jawa Belakang II
Jenis Kelamin (L/P) : L

==========Menu----------
1. Input Data Mahasiswa 1
2. Input Data Mahasiswa 2
3. Lihat Data Mahasiswa
4. Keluar Program
*************************
Pilihan Anda : 2

Nama : Botak
Alamat : Kampung Jawa Belakang II
Jenis Kelamin (L/P) : L
```

Gambar 4.3. Hasil Latihan

```
==========Menu----------
1. Input Data Mahasiswa 1
2. Input Data Mahasiswa 2
3. Lihat Data Mahasiswa
4. Keluar Program
*************************
Pilihan Anda : 3

Nama : Adam Y
Alamat : Kampung Jawa Belakang II
Jenis Kelamin : L

Nama : Botak
Alamat : Kampung Jawa Belakang II
Jenis Kelamin : L
==========Menu----------
1. Input Data Mahasiswa 1
2. Input Data Mahasiswa 2
3. Lihat Data Mahasiswa
4. Keluar Program
*************************
Pilihan Anda : 4

BUILD SUCCESSFUL (total time: 2 minutes 9 seconds)
```

KESIMPULAN

5.1. Kesimpulan

1. Pada percobaan pembuatan *i/o stream* langkah awal yang harus dilakukan adalah memanggil/*import* java.io.
2. Throws merupakan perintah yang digunakan untuk mendeklarasikan atribut yang dibutuhan baik beserta nilainya maupun tidak.
3. Perintah try merupakan perintah yang digunakan untuk melakukan proses tulis maupun baca.
4. Perintah catch merupakan perintah yang digunakan untuk membuat pesan error jika terdapat kesahalan dalam menulis ataupun baca.
5. Jika menggunakan perintah try, pengguna juga harus menggunakan perintah catch, dikarenakan jika tidak menggunakan perintah tersebut akan terjadi error.
6. Method writeUTF yang dipakai pada percobaan menulis ke file binary merupakan method yang digunakan untuk menulis string kedalam file.
7. Method writeInt yang digunakan pada percobaan menulis ke file binary merupakan method yang digunakan untuk menulis atribut dengan tipe data integer ke dalam file.
8. Method readUTF yang digunakan pada percobaan membaca dari file binary merupakan method yang digunakan untuk membaca atribut dengan tipe data string yang terdapat didalam file yang telah dibuat.
9. Method readInt yang digunakan pada percobaan membaca dari file binary merupakan method yang digunakan untuk membaca atribut dengan tipe data integer yang terdapat didalam file yang telah dibuat.
10. Method close yang terdapat pada percobaan menulis dan membaca file binary adalah method yang digunakan untuk menutup file.

5.2. Saran

Pada saat praktikum sebaiknya diberikan banyak contoh dalam pembuatan program dalam bahasa pemrograman java dan diberikan tantangan bagaimana menyelesaikan suatu permasalahan.