File ini berisi padanan program yang pernah diajarkan dalam C++ menjadi program yang fungsinya sama dalam JAVA.
Program dalam bahasa JAVA ditulis dalam JDK versi 1.5
Teks biru adalah teks program dalam bahasa JAVA.
Program JAVA adalah kontribusi Yohanes Nugroho

```
// file io.cc
// Mencoba input-output
using namespace std;
#include <cstdlib>
#include <iostream>


int main () {
 int x;
 float f;
 char cc;
 cin >> x >> cc >> f ; // ketikkan : 5,     8.5
 cout << x << cc << f << endl;

return 0;
}
```

```
//padanan ada pada file IO.java
//(Di Java, nama file harus sama dengan nama kelas)

import java.io.*;
import java.util.Scanner;

class IO {


  public static void main(String args[])  {
            //Jika ingin membaca dari file, ganti menjadi
            //Scanner sc = new Scanner(new
FileInputStream("namafile"));
            Scanner sc = new Scanner(System.in);
            int x = sc.nextInt();
            float f = sc.nextFloat();
            //tidak ada next char pada kelas scanner
            //jika yang diperlukan adalah karakter,
            //maka  ambil karakter pertama dalam string
            String s= sc.next();
            System.out.printf("x = %d f = %f s = %s\n", x,
f, s);
  }

}
```

Deleted: filebacatulis.doc
Deleted: 13
Deleted: 9

```
// file iostream.cc
// Mencoba membuat sekawan ke-5
// error handling lihat stroutrup
using namespace std;
#include <cstdlib>
#include <iostream>

class Point {
public:
   Point(int X, int Y) {
     x= X; y=Y;
   }
   friend ostream& operator<< (ostream& s, const Point& P);
   friend istream& operator>> (istream& s,  Point& P);
   int GetX() { return x; }
   int GetY() { return y; }

private:
  int x;
  int y;
};

ostream& operator<< (ostream& s, const Point& P)
 {
    return s << "P= (" << P.x << "," << P.y << ")" << endl;
 }

istream& operator>> (istream& s,  Point& P)
 { // format (x,y) harus ditaati.
   // Untuk menangani error, baca Buku Strustroup 21.3.5
    int X;
    int Y;
    char cc=0;
    s >> cc >> X >> cc >> Y >> cc; // cc hanya untuk membuang karakter
    P= Point (X,Y);
    return s ;
 }
int main () {
 Point P (5,6);
 cout << P << endl;
 cin >> P;
 cout << P << endl;
return 0;
}

//Tidak ada padanan untuk IO dengan >> dan << di Java,
```

Deleted: filebacatulis.doc
Deleted: 13
Deleted: 9

```
// file : mesinkar.cc
using namespace std;
#include <cstdlib>
#include <fstream>

int main () {

ifstream MesinKar ("fin.txt"); // buat instans MesinKar dari ifstream
ofstream MesinRek ("fout.txt");

char cc;
while (MesinKar.get(cc)){
  MesinRek.put (cc);
}

MesinKar.close();
MesinRek.close();

return 0;
}
```

```
//padanan ada di MesinKar.java dan MesinKar2.java
//pada MesinKar2.java, eksepsi ditangani dengan try dan catch
import java.io.*;

class MesinKar {

  public static void main(String argv[]) throws
java.io.IOException {

    FileReader mesinkar = new FileReader("fin.txt");
    FileWriter mesinrek = new FileWriter("fout.txt");
    int cc = mesinkar.read();
    while (cc!=-1){
      mesinrek.write(cc);
      cc = mesinkar.read();

    }
    mesinkar.close();
    mesinrek.close();

  }

}
```

```
// file bacastring.cc
// string stream
using namespace std;
```

Deleted: filebacatulis.doc

Deleted: 13

Deleted: 9

```
#include <cstdlib>
#include <sstream>
#include <iostream>

void printstring (const string& s) // prints one word per line
{
   istringstream ist (s);
   string w;
   while (ist >> w) {cout << w << endl;}

}

int main () {
 printstring (" if you think C++ is difficult ... ");
 return 0;
}
```

Output

```
if
you
think
C++
is
difficult
...
```

```
//ada dua cara untuk melakukan pemecahan string
//Cara pertama dengan classs StringTokenizer (dapat digunakan di JDK
sebelum 1.5)
import java.io.*;
import java.util.*;

class BacaString1 {

  public static void main(String argv[]) {
    StringTokenizer st = new StringTokenizer("if you think
Java is difficult ... ");
    while (st.hasMoreTokens()){
      System.out.println(st.nextToken());
    }

  }

}
```

Deleted: filebacatulis.doc

Deleted: 13

Deleted: 9

```
//cara kedua dengan kelas Scanner (JDK 1.5 ke atas)
import java.io.*;
import java.util.*;

class BacaString2 {

  public static void main(String argv[]) {
    Scanner st = new Scanner(new StringReader("if you think
Java is difficult ... "));
    while (st.hasNext()){
      System.out.println(st.next());
    }

  }

}
```

Deleted: filebacatulis.doc

Deleted: 13

Deleted: 9

```
//File : mesinbaris.cc
//membaca dan menulis string dari/ke file
using namespace std;
#include<string>
#include<fstream>
#include <iostream>

int main() {
ifstream FileIn("fin.txt");

string baris;
int nbaris=0;
 while (getline (FileIn, baris )){ // baca per baris dari file
  cout << "string hasil baca= " << baris << endl;
  nbaris++; // banyaknya baris
 }
 cout << "Banyaknya baris=" << nbaris << endl;
 FileIn.close();
 return 0;
}
```

```
//MesinBaris.java
//akhir baris adalah jika kembalian readLine() sama dengan null

import java.io.*;

class MesinBaris {

  public static void main(String argv[]) throws IOException
{
    FileReader filein = new FileReader("fin.txt");
    BufferedReader br = new BufferedReader(filein);
    String s = br.readLine();
    int nbaris = 0;
    while (s!=null) {
      nbaris++;
      System.out.println("string hasil baca= "+s);
      s = br.readLine();
    }
    System.out.println("Banyaknya baris="+nbaris);
    filein.close();
  }

}
```

```
// file : string.cc
// contoh pemakaian (hanya 2 method)
// Bacalah rujukan dan coba method yang lain

#include <cstdlib>
#include <iostream>
```

Deleted: filebacatulis.doc

Deleted: 13

Deleted: 9

```
#include <list>
using namespace std;

int main ()
{
 string s1="hello";
 string s2="world";
 string s3 = s1 + "," + s2 + "endl";
 string s4= s3.substr(1,3);
 cout << s4;

 return 0;
}
```

```
//Perbedaan dengan C++: Method substr menjadi substring

class DemoString {

  public static void main(String argv[]){
    String s1="hello";
    String s2="world";
    String s3 = s1 + "," + s2 + "endl";
    String s4= s3.substring(1,3);
    System.out.println(s4);
  }

}
```

Deleted: filebacatulis.doc

Deleted: 13

Deleted: 9

```
// File : vektor.cc
#include <cstdlib>
#include <iostream>
#include <vector>
using namespace std;

int main ()
{
 vector <int> V1 ;
 V1.insert(V1.begin(),1);
 V1.insert(V1.begin(),2);
 V1.insert(V1.begin(),3);
 cout << "size = " <<  V1.size() << endl;

// Vector dipakai sebagai array dengan ukuran tetap
 vector <float> S(10);
 int i;
 for (i=0;i<S.size(); i++) { S[i] =i; cout << float(S[i])<< endl; }
 cout << "bye" << endl;

 vector <float> P(10);
 P= S; // operator = ditangani library
 for (i=0;i<P.size(); i++) {  cout << P[i]<< endl; }

return 0;
}
```

```
//di Java ukuran vektor selalu dinamik
import java.util.Vector;

class DemoVector {

  public static void main(String argv[]){
    //ukuran vektor di java selalu dinamik (tidak bisa
tetap)
    Vector<Integer> v1 = new Vector<Integer>();
    v1.add(1);
    v1.add(2);
    v1.add(3);
    System.out.println("Size = "+v1.size());

  }
}
```

Deleted: filebacatulis.doc
Deleted: 13
Deleted: 9

```
// File : vektor1.cc
#include <cstdlib>
#include <iostream>
#include <vector>
using namespace std;

void printVektor (vector<int> T ){
for (int i=0; i< T.size(); i++) {
      cout << "T[" << i << "]=" << T[i] ;
  }
 cout << endl;
}


void IsiVektor (vector<int> & T ){
 for (int i=0; i< 2; i++) {
      T[i]=i ;
  }
}
int main ()
{
 vector <int> V1 ;
 V1.insert(V1.begin(),1);
 V1.insert(V1.begin(),2);
 V1.insert(V1.begin(),3);
 V1.insert(V1.end(),10);
 V1.insert(V1.end(),30);
 cout << "size = " <<  V1.size() << endl;
 printVektor(V1);
 IsiVektor(V1);
 cout << "nilai baru = " <<  V1.size() << endl;
 printVektor(V1);
 return 0;
}
```

Deleted: int i;¶

Deleted: int i;¶

Output

```
size = 5
T[0]=3T[1]=2T[2]=1T[3]=10T[4]=30
nilai baru = 5
T[0]=0T[1]=1T[2]=1T[3]=10T[4]=30
```

```
//versi  Java memmiliki output yang sama
import java.util.Vector;

class DemoVector2 {

     void IsiVektor (Vector<Integer>  T ){
          for (int i=0; i< 2; i++) {
               T.setElementAt(i, i);
          }
     }

     void printVektor (Vector<Integer> T ){
          for (int i=0; i< T.size(); i++) {
               System.out.printf("T[%d]=%d", i, T.get(i));
          }
```

Deleted: *filebacatulis.doc*

Deleted: *13*

Deleted: *9*

```
        System.out.println();
    }

    void demo(){
        Vector <Integer> V1 = new Vector <Integer>();
        V1.add(1);
        V1.add(2);
        V1.add(3);
        V1.add(10);
        V1.add(30);
        System.out.println("size = " +V1.size());
        printVektor(V1);
        IsiVektor(V1);
        System.out.println("nilai baru = "+  V1.size());
        printVektor(V1);
    }

    public static void main(String argv[]){
            (new DemoVector2()).demo();
    }
}
```

Deleted: filebacatulis.doc

Deleted: 13

Deleted: 9

```
// File : list of Point
#include <cstdlib>
#include <iostream>
#include <list>
using namespace std;

class point {
private:
    int x;
    int y;
public:
    point(int newx, int newy)
    {
        x = newx;
        y = newy;
    }

    void mirror()
    {
        x = 0-x;
        y = 0-y;
    }

    const void printToScreen()
    {
        cout << "{" << x << ", " << y << "}" << endl;
   }
};

int main(int argc, char *argv[]) {
    list<point*> *mylist; // list of pointer to point
    mylist = new list<point*>();
    mylist->push_back(new point(1, 2));
    mylist->push_back(new point(3, 4));
    mylist->push_back(new point(-1, -2));
    mylist->push_back(new point(-5, -6));
    //print semua
    for(list<point*>::const_iterator p = mylist->begin();
         p!=mylist->end(); p++) {
        (*p)->printToScreen();
    }
    cout << endl;
    //masing-masing isi dimirror
    for (list<point*>::iterator p = mylist->begin();
           p!=mylist->end(); p++) {
        (*p)->mirror();
    }
    //print semua
    for(list<point*>::const_iterator p = mylist->begin();
           p!=mylist->end(); p++) {
        (*p)->printToScreen();;
    }
    cout << endl;
for (list < point * >::iterator p = mylist->begin();
                      p != mylist->end(); p++) {
                delete *p;
        }
```

Deleted: filebacatulis.doc

Deleted: 13

Deleted: 9

```
        delete mylist;
    getchar();
// delete

    return 0;
}

    /* OUTPUT:
    {1, 2}
    {3, 4}
    {-1, -2}
    {-5, -6}

    {-1, -2}
    {-3, -4}
    {1, 2}
    {5, 6}
    */
```

```
//kelas point dalam versi Java berada pada kelas terpisah
//perhatikan bahwa printToScreen diubah menjadi toString
//agar objek dengan mudah dicetak menggunakan System.out.println

class Point {

    private    int x;
    private   int y;
    public Point(int newx, int newy){
          x = newx;
          y = newy;
    }

    public void mirror()
    {
          x = 0-x;
          y = 0-y;
    }

    public String toString()
    {
          return  "{" + x + ", " + y + "}\n";
    }
};

//Implementasi Java tidak memerlukan penghapusan elemen
//Di Java tidak ada kelas List (list adalah interface), dan salah
//satu kelas yang mengimplementasikan interface list adalah Vector

import java.util.Vector;
import java.util.Enumeration;

class VectorPoint {
    public static void main(String argv[]) {
          Vector<Point> myvector = new Vector<Point>();
          myvector.add(new Point(1, 2));
```

Deleted: ¶

Deleted: filebacatulis.doc

Deleted: 13

Deleted: 9

```
        myvector.add(new Point(3, 4));
        myvector.add(new Point(-1, -2));
        myvector.add(new Point(-5, -6));
        //print semua
        for(Enumeration<Point> e= myvector.elements();
e.hasMoreElements();){
            System.out.print(e.nextElement());
        }
        System.out.println();

        //masing-masing isi dimirror
        for(Enumeration<Point> e= myvector.elements();
e.hasMoreElements();){
            (e.nextElement()).mirror();
        }
        //print semua
        for(Enumeration<Point> e= myvector.elements();
e.hasMoreElements();){
            System.out.print(e.nextElement());
        }
        System.out.println();

    }
}
```

Deleted: filebacatulis.doc

Deleted: 13

Deleted: 9