

Modul 1

Dasar Pemrograman Java

1.1. Sekilas Pemrograman Java

Java adalah bahasa pemrograman yang multi platform dan multi device. Sekali anda menuliskan sebuah program dengan menggunakan Java, anda dapat menjalankannya hampir di semua komputer dan perangkat lain yang support Java, dengan sedikit perubahan atau tanpa perubahan sama sekali dalam kodenya. Aplikasi dengan berbasis Java ini dikompulasikan ke dalam p-code dan bisa dijalankan dengan Java Virtual Machine. Fungsionalitas dari Java ini dapat berjalan dengan platform sistem operasi yang berbeda karena sifatnya yang umum dan non-spesifik.

Java mempunyai beberapa keunggulan dibandingkan dengan bahasa pemrograman lainnya. Keunggulan bahasa pemrograman Java antara lain:

1. Berorientasi objek

Java adalah bahasa pemrograman yang berorientasi pada objek. Java membagi program menjadi objek-objek serta memodelkan sifat dan tingkah laku masing-masing dalam menyelesaikan suatu masalah.

2. Java bersifat multiplatform

Java dirancang untuk mendukung aplikasi yang dapat beroperasi di lingkungan jaringan berbeda. Untuk mengakomodasi hal tersebut, Java compiler membangkitkan bytecodes (sebuah format yang tidak tergantung pada arsitektur tertentu yang didesain untuk mengirimkan kode ke banyak platform perangkat keras dan perangkat lunak secara efisien). Java dapat dijalankan oleh banyak platform seperti Linux, Unix, Windows, Solaris, maupun Mac.

3. Java bersifat multithread

Multithreading adalah kemampuan suatu program komputer untuk mengerjakan beberapa proses dalam suatu waktu. Thread dalam Java memiliki kemampuan untuk memanfaatkan kelebihan multi prosesor apabila sistem operasi yang digunakan mendukung multi prosesor.

4. Dapat didistribusi dengan mudah

Java memiliki library rutin yang lengkap untuk dirangkai pada protocol TCP/IP (seperti HTTP dan FTP) dengan mudah. Kemampuan networking Java lebih kuat dan lebih mudah digunakan. Java memudahkan tugas pemrograman jaringan yang sulit seperti membuka dan mengakses sebuah socket koneksi. Java juga memudahkan pembuatan CGI (Common Gateway Interface).

5. Bersifat dinamis

Java dirancang untuk beradaptasi dengan lingkungan yang sedang berkembang. Java bersifat dinamis dalam tahap linking. Class yang ada dapat di link sebatas yang diperlukan, apabila diperlukan modul kode yang baru dapat di link dari beberapa sumber, bahkan dari sumber dalam jaringan Internet.

1.2. NetBeans

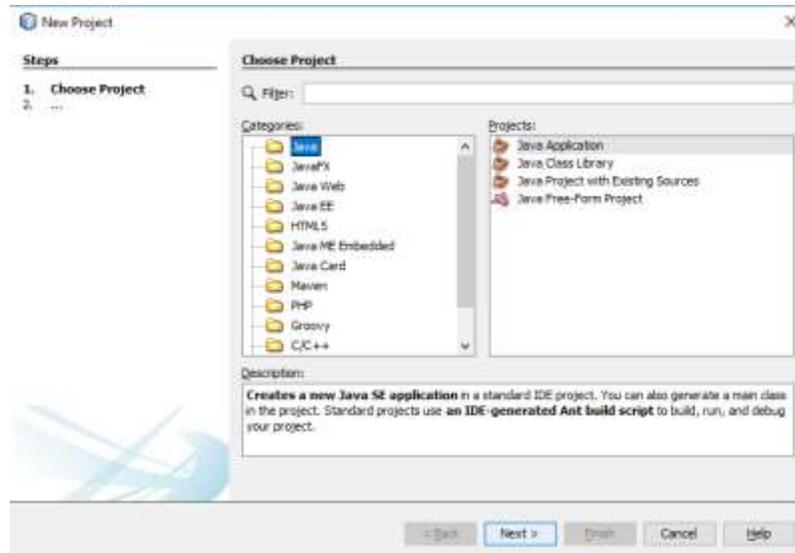
Netbeans merupakan sebuah aplikasi Integrated Development Environment (IDE) yang berbasiskan Java dari Sun Microsystems yang berjalan di atas swing. Swing merupakan sebuah teknologi Java untuk pengembangan aplikasi desktop yang dapat berjalan pada berbagai macam platform seperti windows, linux, Mac OS X dan Solaris. Sebuah IDE merupakan lingkup pemrograman yang di integrasikan ke dalam suatu aplikasi perangkat lunak yang menyediakan Graphic User Interface (GUI), suatu kode editor atau text, suatu compiler dan suatu debugger. Salah satu yang menjadi kelebihan NetBeans GUI Builder adalah yang telah disebutkan diatas, yaitu GRATIS. Selain itu NetBeans GUI Builder sangat kompetebel dengan Swing karena memang langsung dikembangkan oleh Sun yang notabenenya sebagai pengembang Swing. NetBeans hanya mensupport 1 pengembangan Java GUI, yaitu Swing, yang padahal ada Java GUI yang dikembangkan oleh eclipse yang bernama SWT dan JFace yang sudah cukup populer.

1.3 Program Java Menggunakan NetBeans

a. Membuat Project Baru

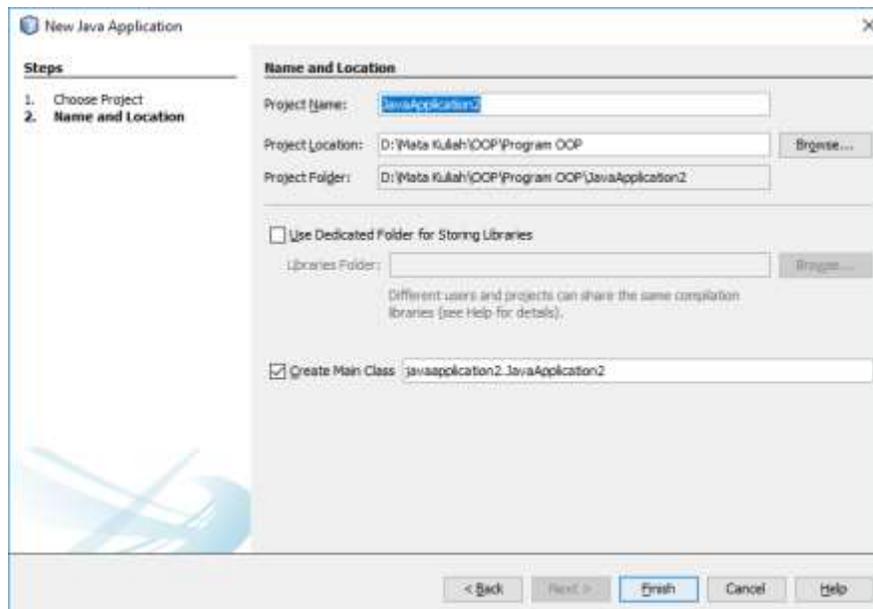
Langkah – langkah membuat project baru adalah sebagai berikut :

1. Pilih File, NewProject untuk menampilkan jendela NewProject, seperti terlihat pada gambar 1.
2. Pilih Java dalam bagian Categories dan JavaApplication dalam bagian Projects dan kemudian klik Next untuk menampilkan jendela NewJavaApplication, seperti terlihat pada gambar 2.
3. Ketik nama project demo pada kolom ProjectName dan pilih lokasi penyimpanan



Gambar 1. Tampilan New Project

4. Klik Finish untuk membuat project



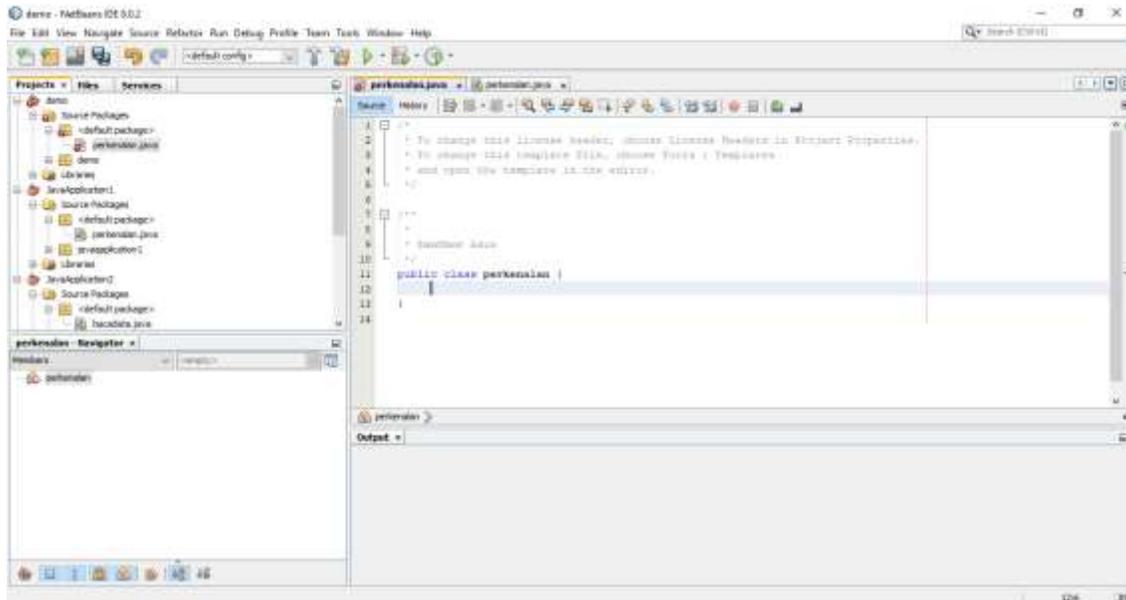
Gambar 2. Tampilan New Java Application

b. Membuat Kode Program

Langkah – langkah membuat class baru adalah sebagai berikut :

1. Klik kanan node demo pada project pane untuk menampilkan context menu.
2. Pilih New JavaClass untuk menampilkan jendela NewJavaClass
3. Ketik pengenalan pada kolom ClassName dan pilih Source Packages pada kolom Location.

4. Klik Finish untuk membuat class perkenalan.
5. Tampilan untuk class perkenalan dapat dilihat pada gambar 3



Contoh Program java

```
class Cetak_Hello
{
    public static void main (String [] arg)
    {
        System.out.println("Hello World!");
    } // akhir main
} //akhir definisi class
```

Jika program dieksekusi maka akan tercetak kata **“Hello World!”**

1.4 Implementasi Algoritma Dasar di Pemrograman Java

a. Variabel

Variabel digunakan untuk menampung data yang digunakan pada program. Variabel menampung data tergantung pada tipe data yang dideklarasikan sebelumnya.

Cara mendeklarasikan variable yaitu :

Sintaks dasar :

[tipe data] [nama variabel]

Untuk mendefinisikan variabel, suatu identifier dapat digunakan untuk menamai variabel tersebut. Identifier adalah kumpulan karakter yang dapat digunakan untuk menamai variabel, method, class, interface, dan package. Dalam pemrograman Java identifier bisa disebut sah apabila diawali dengan :

- Huruf /abjad
- Karakter Mata Uang
- Underscore(_)

Identifier dapat terdiri dari :

- Huruf / abjad
- Angka
- Underscore (_)

Identifier tidak boleh mengandung @, spasi atau diawali dengan angka serta tidak boleh menggunakan *keyword* yang telah digunakan di pemrograman java. Selain karakter, Unicode juga dapat digunakan sebagai identifier.

Java memiliki dua jenis tipe data yang dikategorikan menjadi dua yaitu tipe data primitif dan tipe data referensi.

Macam tipe data primitif diantaranya :

Integer (Bilangan bulat)

Integer merupakan tipe data numerik yang digunakan untuk mendefinisikan bilangan bulat. Tipe data numeric yang termasuk integer diantaranya :

Tipe	Deskripsi
Byte	-128 s/d +127 menempati 8 bits di memori
Short	-32768 s/d +32767 menempati 16 bits di memori
Int	-2147483648 s/d +2147483647 menempati 32 bits di memori
Long	-9223372036854775808 s/d +9223372036854775807 menempati 64 bits di memori

Floating Point (Bilangan pecahan)

Floating point digunakan untuk menangani bilangan decimal atau perhitungan yang lebih detail dibanding integer.

Tipe	Deskripsi
Float	-3.4×10^8 s/d $+3.4 \times 10^8$
Double	-1.7×10^{308} s/d $+1.7 \times 10^{308}$

Char

Char adalah karakter tunggal yang pendefinisianya di awal dan akhir menggunakan tanda petik tunggal ('). Tipe char mengikuti aturan Unicode, sehingga bisa dapat menggunakan kode untuk kemudian diikuti bilangan dari 0 sampai 65535, tetapi yang biasa digunakan adalah bilangan heksadesimal dari 0000 sampai FFFF.

Boolean

Tipe data Boolean terdiri dari dua nilai saja, yaitu *true* dan *false*. Boolean sangat penting untuk mengevaluasi suatu kondisi.

Tipe Data Referensi

Kelebihan pemrograman dengan orientasi objek adalah dapat mendefinisikan tipe data baru yang merupakan objek dari *class* tertentu. Tipe data ini digunakan untuk mereferensikan objek atau class tertentu, seperti String.

b. Operator

Operator adalah simbol khusus yang menyajikan operasi khusus pada satu, dua, atau tiga operand dan kemudian mengembalikan hasilnya. Jenis operator antara lain :

1. Operator Aritmatika

Operator ini digunakan pada operasi-operasi aritmatika seperti penjumlahan, pengurangan, pembagian dan lain-lain.

Operator	Keterangan
+	Penjumlahan
-	Pengurangan
*	Perkalian
/	Pembagian
%	Modulus (Sisa bagi)
++	Increment

Operator	Keterangan
--	Decrement

2. Operator Relasional

Untuk membandingkan 2 nilai (variabel) atau lebih digunakan operator relasional, dimana operator ini akan mengembalikan atau menghasilkan nilai *True* atau *False*.

Operator	Keterangan
<	Lebih Kecil
<=	Lebih Kecil Sama Dengan
>	Lebih Besar
>=	Lebih Besar Sama Dengan
==	Sama Dengan
!=	Tidak Sama Dengan

3. Operator Kondisional

Operator ini menghasilkan nilai yang sama dengan operator relasional, hanya saja penggunaannya lebih pada operasi – operasi Boolean.

Operator	Keterangan
&&	And
	Or
!	Not

4. Operator Shift dan Bitwise

Kedua operator ini digunakan untuk memanipulasi nilai dari bitnya, sehingga diperoleh nilai yang lain.

Operator	Keterangan
&	And

Operator	Keterangan
	Or
^	Xor
~	Not
>>	Shift Right
<<	Shift Left

5. Operator Assignment

Operator assignment digunakan untuk memberikan suatu nilai ke sebuah variabel. Operator assignment tidak hanya berupa '=', namun selain itu dalam C# terdapat beberapa shortcut assignment yang lain.

Operator	Contoh	Ekivalen dengan
+=	b += a	b = b + a
-=	b -= a	b = b - a
*=	b *= a	b = b * a
/=	b /= a	b = b / a
%=	b %= a	b = b % a
&=	b &= a	b = b & a
=	b = a	b = b a
^=	b ^= a	b = b ^ a
<<=	b <<= a	b = b << a
>>=	b >>= a	b = b >> a

Contoh

```
class contoh1
{
    public static void main (String[]args)
    {
        int a, c;
        int b = 10;
        a = 30;
        c = a + b;
        System.out.println("nilai dari c adalah " +(c));
        System.out.println("Hasil dari a*b = " +(a*b));
        System.out.println("Hasil dari a ditambah " +(a++));
    }
}
```

c. Struktur Pemilihan

Struktur pemilihan atau percabangan adalah suatu cara pemecahan masalah dengan memilih statement atau perintah tertentu berdasarkan berdasarkan suatu kondisi. Statemen atau perintah tertentu akan dijalankan apabila memenuhi ketentuan yang telah didefinisikan sebelumnya. Pernyataan percabangan memungkinkan suatu statement dieksekusi, hanya jika suatu kondisi terpenuhi atau tidak terpenuhi. Untuk menentukan suatu kondisi tertentu apakah benar atau salah, ya atau tidak; diperlukan pemahaman tentang beberapa struktur yang mendukung pengambilan keputusan. Dalam hal pemrograman, harus mengenal operator yang dipakai dalam pengambilan keputusan seperti operator relasi dan logika.

1. If

Statement If memungkinkan sebuah program untuk dapat memilih beberapa operasi untuk dieksekusi berdasarkan beberapa pilihan.

Bentuk Umum If

```
if (kondisi)
{
    Pernyataan-1;
    ...
    Pernyataan-n;
}
```

Jika kondisi benar maka pernyataan-1 sampai dengan pernyataan-n akan dijalankan, sedangkan jika kondisi salah maka program akan keluar dari struktur if.

Bentuk Umum If – Else

```
if (kondisi)
    Pernyataan-1;
else
    Pernyataan-2;
```

Jika kondisi benar maka pernyataan1 akan dijalankan, dan jika kondisi salah maka pernyataan-2 yang akan dijalankan.

Bentuk Umum Nested If

```
If (kondisi1)
    Pernyataan-1;
else if (kondisi2)
    Pernyataan-2;
else if (kondisi3)
    Pernyataan-2;
...
...
else
    pernyataan;
```

2. Switch

Switch adalah pernyataan yang digunakan untuk menjalankann salah satu pernyataan dari beberapa kemungkinan statement untuk dieksekusi, berdasarkan nilai dari sebuah ungkapan dan nilai penyeleksi. Setiap ungkapan diungkapkan dengan sebuah nilai integral konstan, seperti sebuah nilai dengan tipe byte, short, int atau char.

Bentuk Umum Switch

```
Switch (Ekspresi)
{
    case nilai1 : {
                                Statement1;
                                Statement1;
                            }
        break;
    case nilai2 : {
                                Statement1;
                                Statement1;
                            }
        break;
```

```

        default    : {
                                Statement1;
                                Statement1;
                        }
                    break;
}

```

Keterangan :

- a. Case : digunakan sebagai label yang menandai awal eksekusi deret instruksinya hingga ditemukan pernyataan break.
- b. Default : label yang digunakan atau dieksekusi apabila label case tidak ada yang memenuhi.
- c. Break : Perintah pengontrol alur program, berfungsi untuk keluar dari satu blok kondisi ataupun iterasi.

Contoh

```

class contoh2
{
    public static void main (String[]args)
    {
        int a, b, max;
        a = 20;
        b = 30;
        if ( a > b)
            System.out.println ("nilai max adalah : " +b);
        else
            System.out.println ("nilai max adalah : " +a);
    }
}

```

```

import java.util.Scanner;
class contoh3
{
    public static void main (String[] arg)
    {
        int nilai1;
        System.out.println("program mulai");
        Scanner input = new Scanner(System.in);
        System.out.print("Inputkan Nilai  : ");
        nilai1 = input.nextInt();
        if (nilai1 > 0)
            System.out.println("bilangan positif");
        else if (nilai1 < 0)
            System.out.println("bilangan negatif");
        else
            System.out.println("bilangan nol");
        System.out.println("program selesai");
    }
}

```

d. Struktur Pengulangan

Struktur pengulangan digunakan untuk mengulang Sebuah / kelompok instruksi diulang untuk jumlah pengulangan tertentu. Beberapa hal penting pada struktur pengulangan adalah adanya inisialisasi, kondisi, dan iterasi. Ada empat struktur pengulangan yang dapat digunakan pada bahasa C#, yaitu: for, while, do...while, dan foreach.

Berkaitan dengan proses pengulangan, pemrograman C# juga menyediakan pernyataan *break* (untuk mengakhiri pengulangan) , *goto* , dan *continue*(untuk melakukan pengulangan selanjutnya) .

1. Struktur For

Struktur pengulangan for biasa digunakan untuk mengulang suatu proses yang telah diketahui jumlah pengulangannya.

Bentuk Umum For :

```

for (inisialisasi;kondisi;iterasi)
{
    Pernyataan ;
}

```

Keterangan:

Inisialisasi : pernyataan untuk menyatakan keadaan awal dari variable kontrol.

Kondisi : ekspresi relasi yang menyatakan kondisi untuk keluar dari pengulangan.

Iterasi : pengatur perubahan nilai variabel kontrol.

2. Struktur while

Struktur ini digunakan bila jumlah perulangannya belum diketahui. Proses pengulangan akan terus berlanjut selama kondisinya bernilai benar (true) dan akan berhenti bila kondisinya bernilai salah.

Bentuk Umum While:

While (kondisi)

```
{  
    Pernyataan  
    ...  
}
```

3. Struktur do... while

Pada dasarnya struktur pengulangan ini sama saja dengan struktur while, hanya saja pada proses pengulangan dengan while, seleksi berada di while yang letaknya diatas, sementara pada pengulangan do ... while, seleksi berada dibawah batas pengulangan. Jadi dengan menggunakan struktur do .. while sekurang – kurangnya akan terjadi satu kali pengulangan.

Bentuk Umum do ... while

do

```
{  
    Pernyataan  
    ...  
}
```

while (kondisi)

Contoh

```
import java.util.Scanner;
class contoh4
{
    public static void main (String[] arg)
    {
        int nilai1;
        int i;
        System.out.println("program mulai");
        Scanner input = new Scanner(System.in);
        System.out.print("Inputkan Nilai 1 : ");
        nilai1 = input.nextInt();
        i = 1;
        while (i <= nilai1)
        {
            System.out.print(i);
            i++;
        }
        System.out.println();
        System.out.println("program selesai");
    }
}
```

1.5. Array

Array adalah sebuah struktur data yang terdiri atas banyak variabel dengan tipe data sama, dimana masing-masing elemen variabel mempunyai nilai indeks. Setiap elemen array mampu untuk menyimpan satu jenis data (variabel). Ukuran array bersifat tetap, larik akan mempunyai ukuran yang sama pada saat sekali dibuat. Variable array dideklarasikan dengan mencantumkan tipe dan nama variabel yang akan diikuti dengan banyaknya lokasi memori yang akan dibuat.

Contoh Array X yang terdiri dari 10 Elemen :

	X[0]	X[1]	X[2]	X[3]	X[4]	X[5]	X[6]	X[7]	X[8]	X[9]
Isi Array	50	75	60	43	70	80	65	77	78	35

Deklarasi Array

Sintaks untuk mendeklarasikan array di C # :

```
datatype[] arrayName;
```

dimana,

- datatype digunakan untuk menentukan jenis elemen dalam array.
- [] menentukan peringkat dari array. rank menentukan ukuran array.
- arrayName menentukan nama array.

Contoh :

```
int[] X;
```

Inisialisasi Array

Setelah mendeklarasikan array, maka harus dilakukan proses inisialisai agar array yang dideklarasikan dapat digunakan. Untuk menginisialisai sebuah array menggunakan kata **new**.

Sebagai contoh akan dilakukan inisialisasi untuk array X bertipe integer dengan 10 indeks/elemen, maka sintaksnya adalah :

```
int X[] = new int[10];
```

Contoh

```
import java.util.Scanner;
class contoh5
{
    public static void main (String args[])
    {
        Scanner masuk=new Scanner(System.in);
        float nilai[]=new float[5];
        System.out.println("masukkan 5 buah data
nilai");
        for(int i=0;i<5;i++)
        {
            System.out.print("Data ke"+(i+1)+" : ");
            nilai[i]=masuk.nextFloat();
        }
        System.out.println("data nilai yang
dimasukkan");
        for(int i=0;i<5;i++)
            System.out.println(nilai[i]);
    }
}
```

```
import java.util.Scanner;
class contoh6
{
    public static void main (String args[])
    {
        int[] x;
        //int nilai;
        int[] y = { 5,4,3,2,1 };
        x = new int[5];
        int i;
        System.out.print("masukkan nilai : ");
        Scanner nilai = new Scanner (System.in);
        y[2] = nilai.nextInt();
        System.out.println("data nilai yang dimasukkan");
        for (i = 0; i <5; i++)
        {
            System.out.println(y[i]);
        }
    }
}
```

```

class contoh7
{
    public static void main (String args[])
    {
        int [][] piksel;
        piksel = new int[2][3];
        // mengisi elemen tertentu
        piksel[0][0] = 70;
        piksel[0][1] = 18;
        piksel[0][2] = 45;
        piksel[1][0] = 75;
        piksel[1][1] = 66;
        piksel[1][2] = 89;
        //menampilkan elemen array
        int i,j;
        for(i=0;i<2;i++)
        {
            for (j=0; j<3;j++)

                System.out.print(piksel[i][j] + " ");
                System.out.println("");

        }
    }
}

```

Tugas praktikum

1. Buat program untuk menentukan kelulusan mahasiswa menggunakan prosedur atau fungsi.
Input Data : Nama, NIM, Nilai.
Output Data : Keterangan LULUS / GAGAL
2. Buatlah program untuk mencari nilai pada sebuah array yang bernilai integer. Jika nilai ketemu maka cetak kata "KETEMU", jika tidak maka cetak kata "TIDAK KETEMU"