**POKOK BAHASAN 2**

**OPERATOR , CONTROL FLOW dan STRING**

**PENDAHULUAN**

Pada pokok bahasan ini akan dibahas mengenai pendeklarasian variable, baik instance variable, class variable, maupun local variable, serta membahas fungsi dari masing-masing simbol operator dan control flow. Sehingga setelah mempelajari bab ini diharapkan mahasiswa mampu :

1. Mendeklarasikan variable dengan berbagai macam tipe data.
2. Mengakses instance variable, class variable, maupun local variable.
3. Memahami fungsi dari masing-masing simbol operator.
4. Memahami control flow.
5. Mahasiswa mampu membuat program yang berkaitan dengan String.
6. Mahasiswa mampu memahami prinsip penggunaan String.

**PENYAJIAN (TUTORIAL)**

Operator dan Control Flow adalah jantung dari program Control Flow mempunyai kemampuan untuk mengendalikan jalan dari program. Tanpa Operator dan Control Flow, program tidak akan mampu melakukan banyak hal. Ada beberapa jenis Operator dan Control Flow dalam Java, yaitu:

**1. Aritmatika Operator**

Java mempunyai operator aritmatika seperti + (penjumlahan), - (pengurangan),\* (perkalian), / (pembagian) dan % (modulo). Operator – operator ini berlaku seperti dalam operasi aritmatika. Dalam Java juga dimungkinkan untuk penyederhanaan penulisan operator seperti yang terlihat di bawah ini;

Int a, b, c, d, e,

a = 15;b =10; //=80;d=100;e=20;

a+ = 5; //samadengan a = a+5

b- =e; //samadengan b = b-e

c\* =b; //samadengan c = c\*b

d/ =4; //samadengan d = d/4

e%=3; //samadengan e = e % 3, hasilnya = 2

d++; //samadengan d = d+1 (increment )

c-- ; //samadengan c = c-1 (decrement)

**2.Logika Operator**

Ada tigalogika Operator dalam java yaitu:

&& Logika /nd

|| Logika*Or*

| Logika*Not*

Ketiga operator tersebut dioperasikan terhadap Boolean data tipe yang menghasilkan bolean data type juga (true atau false)

**3. Bit per bit operator**

Variabel bertipe long, int, short, byte dan char dapat mempunyai operator yang dioperasikan langsung kesusunan binernya (bit atau bit).

Tabel 2.1 Tabel bit per bit Operator

|  |  |  |
| --- | --- | --- |
| Operator | Nama | Keterangan |
| ~ | NOT | Mengubah 0 menjadi 1 dan 1 menjadi 0 |
| & | AND | 1 & 1 = 1 selainituhasilnyaselalu 0 |
| | | OR | 0|0 = 0 selainituhasilnyaselalu 1 |
| ^ | XOR | 0^0 = 0, 1^1 = 0 selainituhasilnya 1 |
| >> | SHIFT RIGT | Dorong 1 bit kekanantermasuktanda minus |
| >>> | SHIFT RIGT | Dorong 1 bit ke kana nisi bit paling kiridengan 0 |
| << | SHIFT LEFT | Dorong 1 bit kekiri |

Sepertihalnya aritmatika operator, Bit per bit operator juga bias dilakukan

Penyederhanaan penulisannya.

**4. Relasional**

Operator relasional digunakan untuk membandingkan dua nilai.

Hasil dari perbandingan tersebut adalah data tipe Boolean (true atau false).

Tabel 2.2 Tabel operator Relasional

|  |  |
| --- | --- |
| Operator | Keterangan |
| > | LebihBesar |
| < | Lebihkecil |
| >= | Lebihbesaratausamadengan |
| <= | Lebihkecilatausamadengan |
| != | Tidaksamadengan |
| = = | Samadengan |

Tipe variable di java seperti short, int, long, float, double, Boolean, dan character dapat dibandingkan satu dengan yang lainnya .

Contoh :

Char c = ‘\*’;

Inti = 0x2a;

Boolean b;

b = c = = i;

**5. Prioritas**

Setiap operator mempunyai prioritas sendiri – sendiri, sehingga perhitungan matematika dilakukan sesuai prioritasnya. Lihat contoh dibawah ini :

C = a + b\*d;

Akan dijalankan sesuai prioritas yaitu \*,+,=

Dibawah ini diberikan table prioritas dari operator, urutan prioritas dari atas kebawah

Operator yang sama prioritasnya akan dioperasikan dari kanan kekiri.

Tabel 2.3 Tabel operator Prioritas

|  |
| --- |
| Operator |
| ( ) [ ] |
| ++ -- ~! |
| \*/ % |
| + - |
| >>>>><< |
| >>= <<= |
| == ! = |
| & |
| ^ |
| | |
| && |
| || |
| ?: |
| = |

**6.Pernyataan*if***

Pernyataan ***if*** adalah control flow yang paling sederhana. Pernyataan ini mengevaluasi suatu ekspresi. Jika ekspresi benar (true) maka blok pernyataan akan dijalankan. Sebaliknya computer akan melompati blok pernyataan tersebut. Sintaks dari pernyataan ***if*** :

If (ekspresi)

Pernyataan;

**7. Pernyataan *if-else***

Pernyataan ***if-else*** mengevaluasi suatu ekspresi. Jika ekspresi benar (true) maka blok pernyataan dijalankan. Sebaliknya computer akan menjalankan blok pernyataan yang lain. Sintaks dari penyataan ***if-else:***

If(ekspresi )

Pernyataan\_1;

Else

Pernyataan\_2;

**8. Pernyataan switch, case dan break**

Switch digunakan untuk percabangan (seleksi) lebih dari dua. Ini digunakan untuk menghindari pergunaan if-else yang berlapis-lapis.

**9. Pengulangan (Loop)**

**Pengulangandengan*while***

Alur program blok while akan dilakukan berulang-berulang selama ekspresi benar (true)

Sintaks dari pengulangan while adalah sbb:

While (ekspresi)

Pernyataan;

**Pengulangandengan*do-while***

Pengulangan dengan while pemeriksaan ekspresi dilakukan sebelum pengulangan. Ada kalanya pengulangan pemeriksaan ekspresi dilakukan setelah pengulangan. Jadi minimal satu kali pengulangan dilakukan. Pengulangan ini menggunakan do-while, sintaksnya.

do

pernyataan;

while (ekspresi);

**Pengulangandengan*for***

Pengulangan dengan for menyerupai while, namun dilengkapi dengan inialisasi, test kondisidan update counter. Jadi sintaks pengulangan dengan for adalah :

for (inialisasi\_counter ; test\_kondisi ; update\_counter) {

instruksi;

--------

};

Inialisasi\_counter dilakukan sekali pada saat akan dilakukan pengulangan, test\_kondisi diperiksa setiap kali sebelum pengulangan dilakukan, sedangkan update\_counter dilakukan setiap kali selesai pengulangan.

**Break, continue**

Dalam pengulangan break digunakan untuk mematahkan (mengakhiri) pengulangan secara paksa. Sedangkan continue digunakan untuk melanjutkan pengulangan berikutnya secara paksa dengan mengabaikan perintah dibawah pernyataan continue.

**String**

String adalah obyek yang terdiri atas deretan karakter. Tidak seperti array dari karakter, String tidak perlu ditentukan jumlah panjang karakternya. String dapat langsung di inialisasi tanpa kata kunci new, contoh :

String str = “ Hello “ ;

Str = str + “ Java Programmer” ;

String dapat juga diciptakan melalui cara berikut :

char c\_arr [ ] = { ‘h’, ‘e’, ‘1’, ‘0’ };

String s1 = new String (c\_arr) ;

String s2 = new String (c\_arr, 2, 2);

// offset ke 2 sebanyak 2 karakter

Ada beberapa method dalam String yang perlu kita ketahui antara lain:

*Int length ( )* memberikan nilai balik (return value) panjang dari aktual string. *char charAt (int index)* memberikan nilai balik berupa karakter pada posisi index dalam aktual string.

public class TestString {

public static void main(String argv[]){

String tabel[]={"amir","bambang","hasan"};

String str="Ini adalah # String yang diproses";

boolean found=false;

int I;

for (I=0;I<str.length();I++){

if (str.charAt(I)=='#'){

found=true;

break;

}

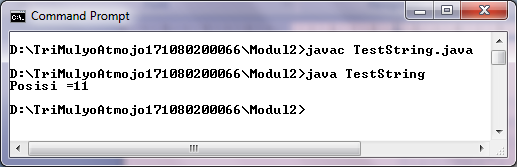
}

if (found)

System.out.println("Posisi ="+I);

}

}



*int indexOf(char ch)*

Memberikan nilai balik posisi pada String di mana karakter ch berada

*int indexOf(String s)*

Memberikan nilai balik posisi index pada String dimana string s ditemukan

*int indexOf(char ch, int starPos)*

*int indexOf(String s, int startPos)*

Sama dengan indexOf sebelumnya, hanya pencarian tidak dimulai dari index 0, melainkan dari index = startPos

*Int lastIndexOf(char c)*

Int lastIndexOf(char c, int startPos)

Int lastIndexOf(String s);

Int lastIndexOf(string s, int startPos)

Method lastIndexOf( ) mencari karakter/String pada String yang diberikan, hanya pencarian dimulai dri kanan ke kiri.

Boolean equals (String s)

Membandingkan aktual string dengan string s dengan mengabaikan huruf besar maupun huruf kecil. Jika sama maka akan memberikan nilai balik true, sebaliknya akan memberikan nilai balik false.

String substring (int offset)

Memberikan niali balik isi aktual string dari posisi offset sampai akhir string

String substring(int offset, int endIndex)

Memberikan nilai balik si aktual string dari posisi offset sampai posisi endIndex

String toLowerCase( )

Mengubah string menjadi huruf kecil semua

String toUpperCase( )

Mengubah spasi yang berdekatan dalam string

Char[ ] getCahrs(int startPos, int endPos, char buffer[ ], int offset)

Mengambil karakter dan startPos sampai endPos, dan menyimpanannya ke dalam buffer dimulai dari index = offset.

Int compareTo(String str)

Membandingkan String dalam UTF (Unicode Transfer Format) dengan nilai balik 0 bila sam, negative bila kecil, dan positif bila lebih besar.

**StringTokenizer**

Class StringTokanizer berfungsi untuk membagi string berdasarkan token tertentu. Class ini sangat membantu untuk menginterprestasikan suatu string ke suatu maksud tertentu. Contoh yang paling populer adalah membagi kalimat menjadi beberapa kata. Kita juga membaca data dari file kemudian data tersebut kita uraikan berdasarkan token tertentu.

**LEMBAR KERJA DAN TUGAS**

1. Menggunakan Bit Operator

public class OprTest {

public static void main (String[] args) {

byte b=0x8;

byte c, d, e, f, g, h, i;

c=(byte)~b;

c &=0xf;

d= (byte) (b | c);

e= (byte) (c>>2);

f= (byte) (e+3);

g= (byte) (c & f);

h= c;

h ^=0x3;

i= (byte) (h >>>1);

i <<=2;

System.out.println(" c = " + c);

System.out.println(" d = " + d);

System.out.println(" e = " + e);

System.out.println(" f = " + f);

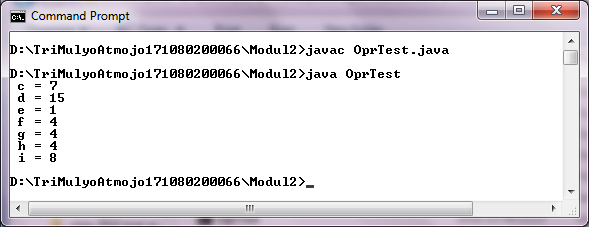
System.out.println(" g = " + g);

System.out.println(" h = " + h);

System.out.println(" i = " + i);

}

}



1. Menggunakan pernyataan if

public class TestIf {

public static void main (String argv []) {

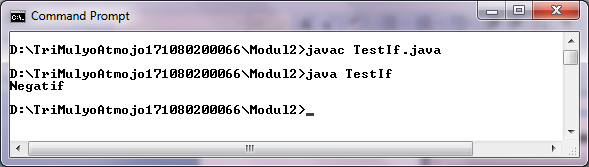
int x = -3;

if ( x < 0 )

System.out.println("Negatif");

}

}



1. Menggunakan pernyataan if-else

public class TestIfElse {

public static void main (String args []) {

int a=10, b=13, x=0, y=0;

if (a>b)

x=a;

else {

x=b;

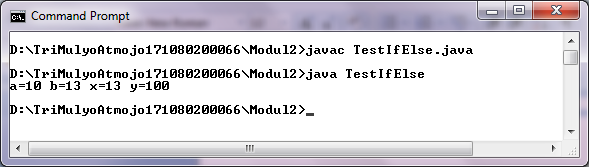
y=100;

}

System.out.println ("a="+a+ " b="+b+" x="+x+" y="+y);

}

}



1. Menggunakan pernyataan switch case, dan break

public class TestSwitch {

public static void main (String argv[]) {

char c='\_';

switch (c) {

case '+':

System.out.println("Operasipenjumlahan");

break;

case '\_':

System.out.println("Operasipengurangan");

break;

case '/':

System.out.println("Operasipembagian");

break;

case '\*':

System.out.println("Operasiperkalian");

break;

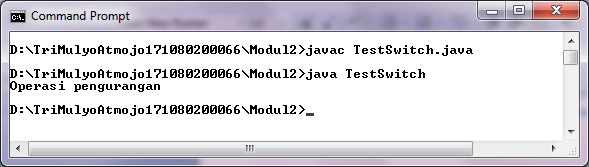
default:

System.out.println("Operasitidakdikenal");

}

}

}



1. Menggunakan Pengulangan dengan while

public class TestWhile {

public static void main (String argv []) {

int m;

m = 5;

while (m > 0) {

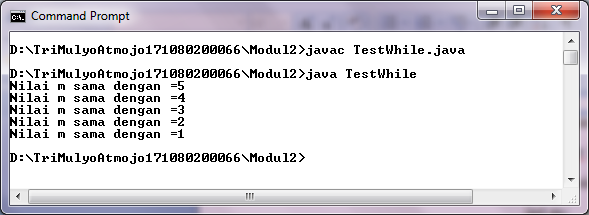
System.out.println ("Nilai m samadengan ="+m);

m--;

}

}

}



1. Menggunakan Pengulangan do-while

public class TestDoWhile {

public static void main (String argv []) {

int m = 5;

do {

System.out.println(" m ="+m);

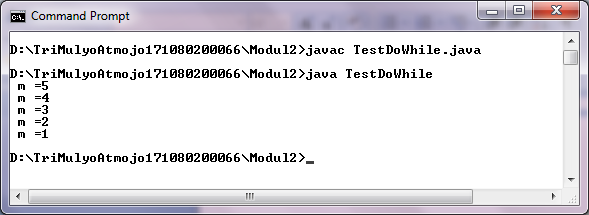
m--;

}

while (m > 0);

}

}



1. Menggunakan Pengulangan dengan for

public class TestFor {

public static void main (String argv []) {

int m;

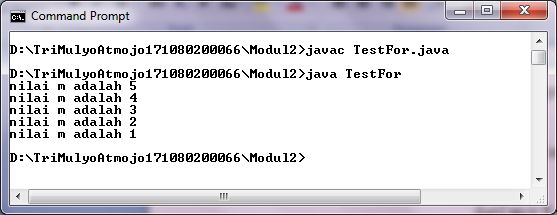
for ( m = 5 ; m > 0 ; m--) {

System.out.println("nilai m adalah "+m);

}

}

}



1. Menggunakan Break

public class TestBreak {

public static void main (String argv[]) {

int i, j;

j = 50;

for (i=5; i>0; i--) {

if (j>100)

break;

j += 50;

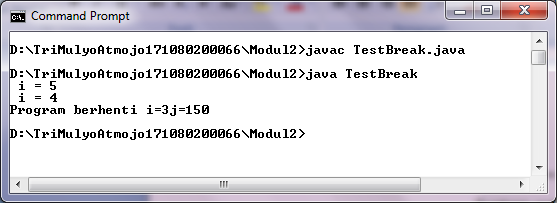
System.out.println(" i = " +i);

}

System.out.println("Program berhenti i="+i +"j="+j);

}

}



1. Menggunakan continue

public class TestContinue {

public static void main (String arvg []) {

int i, j;

j = 50;

for (i=5; i>0; i--){

if (j>100)

continue;

j += 50;

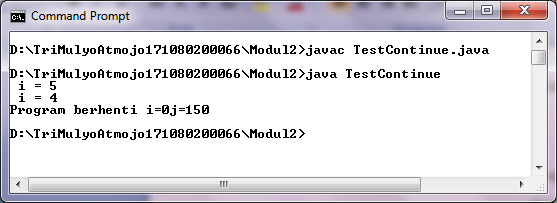
System.out.println(" i = " +i);

}

System.out.println("Program berhenti i="+i +"j="+j);

}

}



1. Menggunakan Boolean equals (string s)

import java.io.\*;

public class Compare {

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

String tabel[]={"ari", "gali", "ika", "rozi"};

DataInputStream dis=new DataInputStream(System.in);

System.out.println("Berikannama :");

System.out.flush();

String nama=dis.readLine();

for (int i=0;i<tabel.length;i++) {

if (tabel[i].equals(nama)) {

System.out.println(nama+" terdaftar dalam tabel");

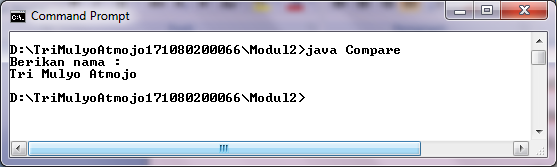
break;

}

}

}

}



1. Menggunakan Int ComapareTo (string str)

public class StringOperasi {

public static void main (String argv[]) {

String salam="TerimaKasih, AndaBelajar Java Programming";

String bagian=salam.substring(5,16);

String satu="Satu", dua="Dua", tiga="Tiga";

System.out.println(bagian);

System.out.println(salam.toUpperCase());

System.out.println(salam.toLowerCase());

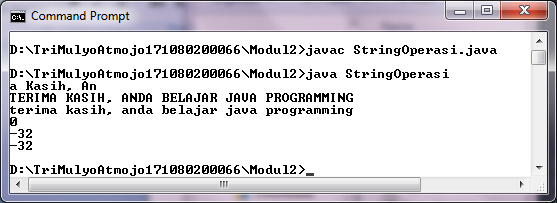
System.out.println(satu.compareTo("Satu"));

System.out.println(dua.compareTo("dua"));

System.out.println(tiga.compareTo("tiga"));

}

}



1. Membagi String berdasarkan token tertentu

importjava.util.\*;

public class TestToken {

public static void main (String argv[]) {

String kalimat= "Di siniPencerahanBersemi";

StringTokenizerst =new StringTokenizer(kalimat," ");

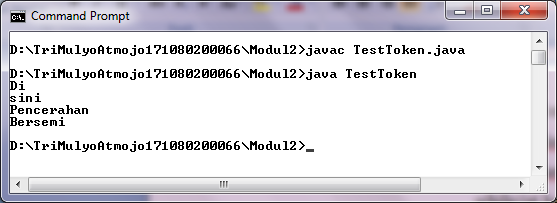
while (st.hasMoreTokens()) {

System.out.println(st.nextToken());

}

}

}



1. Membaca data dari file kemudian menguraikan data tersebut berdasarkan token tertentu

* Membuat file customer.txt dengan struktur sbb

Nama\_depan : nama\_belakang : kota : propinsi

* Dari file customer.txt tersebut bias diuraikan data-data customer

dengan menggunakan class dibawah ini

import java.io.\*;

importjava.util.\*;

public class TokenFile {

public static void main (String argv[]) {

TokenFilett=new TokenFile();

tt.dbTest();

}

public void dbTest() {

DataInputStream dis=null;

String dbRecord=null;

try {

File f=new File ("customer.txt");

FileInputStreamfis=new FileInputStream(f);

BufferedInputStreambis=new BufferedInputStream(fis);

dis=new DataInputStream(fis);

//read db

while((dbRecord=dis.readLine())!=null) {

StringTokenizerst= new StringTokenizer(dbRecord, ":");

String namaDepan =st.nextToken();

String namaBelakang =st.nextToken();

String kota=st.nextToken();

String propinsi=st.nextToken();

System.out.println("NamaDepan :"+namaDepan);

System.out.println("NamaBelakang :"+namaBelakang);

System.out.println("Kota :"+kota);

System.out.println("Propinsi :"+propinsi);

}

}

catch (IOException e) {

//Catch error

System.out.println("IOExceprion error!"+e.getMessage());

}

finally {

if(dis!=null){

try {

dis.close();

}

catch (IOExceptionioe) {

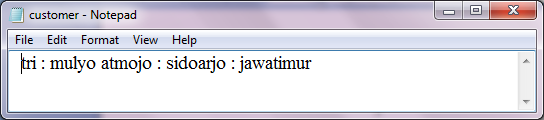
}

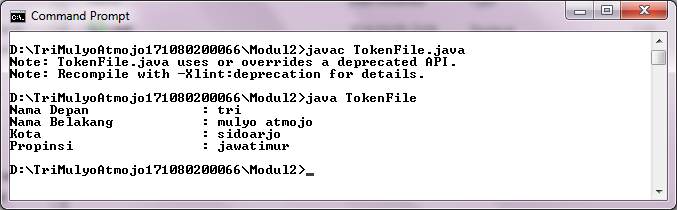
}

}

}

}





**TUGAS**

1. Buatlah program untuk menghitung jumlah bilangan 1 s/d 100 (1+2+3+4…..+100). Tampilkan hasilnya pada layar monitor.

public class Tugas1 {

public static void main (String argv[]) {

int m,n;

n = 1;

m = 0;

while ( n <= 100 ) {

m = m + n;

n++;

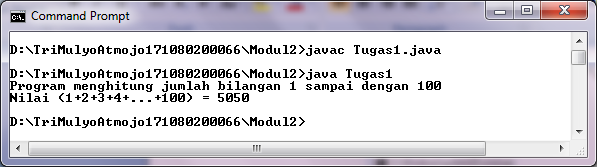
}

System.out.println ("Program menghitungjumlahbilangan 1 sampaidengan 100");

System.out.println ("Nilai (1+2+3+4+...+100) = " + m);

}

}



1. Buatlah progam untuk menampilkan semua bilangan dari 1 s/d 100 yang habis dibagi 5 atau habis dibagi 9

public class Tugas2 {

public static void main (String argv[]) {

int a, b, c ;

a = 1;

b = 5;

c = 9;

while (a<=100) {

if (((a%b)==0) || ((a%c)==0))

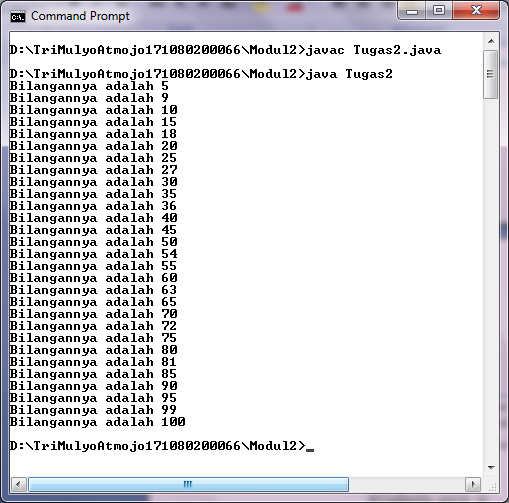
System.out.println ("Bilangannyaadalah " + a);

a++;

}

}

}



1. Buatlah kode program dengan menggunakan statement if. Yang

menampilkan nilai yang dimasukkan dan memberikan keterangan terhadap nilai tersebut dengan ketentuan sebagai berikut:

Nilai >80 mendapat nilai = A, keterangan : Lulus – memuaskan

Nilai > 70 <=80 mendapat nilai = B, keterangan Lulus – Baik

Nilai > 60 <= 70 mendapat nilai C keterangan Lulus – Cukup

Nilai >50 <= 60 mendapat nilai D keterangan Tidak Lulus – Tidak memuaskan

importjava.util.Scanner;

public class Tugas3 {

public static void main (String argv[]) {

Scanner input = new Scanner (System.in);

intnilai;

System.out.print("Masukkannilaiujian [ 0 - 100 ]: ");

nilai = input.nextInt();

if(nilai>80 &&nilai<=100)

System.out.println("Nilai = A , Lulus - Memuaskan");

else if(nilai>70 &&nilai<=80)

System.out.println("Nilai = B, Lulus - Baik");

else if(nilai>60 &&nilai<=70)

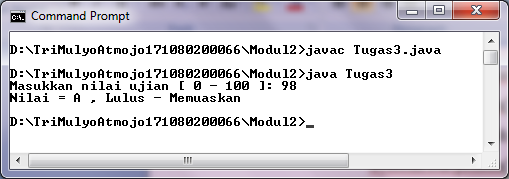
System.out.println("Nilaiujian = C, Lulus - Cukup");

else

System.out.println("Nilaiujian = D, Tidak Lulus - TidakMemuaskan");

}

}



1. Buatlah program untuk menampilkan susunan bilangan biner dari suatu

bilangan bulat, dengan susunan terbalik misalnya biner dari 12 akan ditulis 0011.

import java.io.\*;

public class Tugas4 {

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

String hasil;

InputStreamReaderisr=new InputStreamReader(System.in);

BufferedReaderbr = new BufferedReader(isr);

System.out.print("Masukkannilai :");

String nil=br.readLine();

hasil="";

int n = Integer.parseInt(nil);

int v = 1;

while (v <= n/2){

v = v\*2;}

while (v>0){

if (n < v){

hasil +="0";

}

else{

hasil +="1";

n = n-v;

}

v=v/2;

}

System.out.println("Nilaibinerdari "+nil+" adalah:"+hasil);

System.out.print("Jika di balikhasilnya :");

intnhasil;

char data[]=hasil.toCharArray();

for (nhasil=hasil.length(); nhasil>0; nhasil--) {

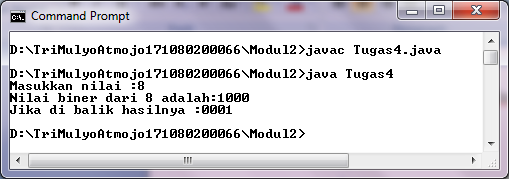
System.out.print(data[nhasil-1]);

}

System.out.println();

}

}



1. Buatlah class untuk membalik urutan kata dalam suatu kalimat.

import java.io.\*;

public class Tugas5 {

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

String KATA, HASIL;

BufferedReaderbr = new BufferedReader(new InputStreamReader(System.in));

System.out.print("Masukkan KATA : "); KATA= br.readLine();

System.out.println("");

System.out.print("HasilnyaAdalah :");

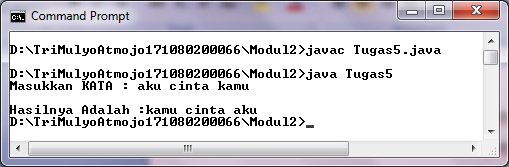
String [] result = KATA.split("\\s");

for (int x=result.length-1; x>=0; x--)

System.out.print(result[x]+" ");

}

}



1. Input sebuah nim, dan ganti semua digit ‘1’ menjadi ‘\*’.

Contoh layout:

Input NIM Anda : 0111500382

NIM Anda menjadi : 0\*\*\*500382

Input NIM Anda : 0244500016

NIM Anda menjadi : 02445000\*6

import java.io.\*;

class Tugas6 {

public static void main(String[] args) throws IOException {

String NIM, HASIL;

BufferedReaderbr = new BufferedReader(new InputStreamReader(System.in));

System.out.print("Input NIM : "); NIM= br.readLine();

HASIL=NIM.replace('1','\*');

System.out.println("NIM Menjadi : "+HASIL);

}

}

