

LAPORAN PRAKTIKUM 1
ALGORITMA PEMOGRAMAN DAN
KOMPUTER 2 “LOOPING”



Disusun oleh

Jiryan Farokhi/5002221102

Asisten Laboratorium

Muhammad Andhika Reswara/5002211086 dan Komang
Ryaandhi Suandita/5002211109

Dosen Pengampu

Dr. Budi Setiyono, S.Si, MT / 19720207 199702 1 001

DEPARTEMEN
MATEMATIKA FAKULTAS
SAINS DAN ANALITIKA DATA
INSTITUT TEKNOLOGI SEPULUH
NOPEMBER SURABAYA
2022

BAB 1 DESKRIPSI

1. Diberikan sebuah input dengan bertipe integer atau bilangan bulat. Di sini programmer diminta untuk menampilkan sebuah deret fibonacci sebanyak n bilangan sesuai dengan inputan yang diberikan. Alur logika dalam pengerjaan soal ini adalah dengan membuat sebuah looping untuk menentukan nilai berikutnya dengan logika angka selanjutnya merupakan penjumlahan dari dua angka sebelumnya. Namun harus diberikan kondisi jika nilai $i = 1$ dan 2 maka akan dilakukan proses *continue* yang mana hal ini akan membuat proses looping akan ter-*skip* dan dilakukan proses looping selanjutnya, Kita bisa memanfaatkan nilai angka ketiga untuk melakukan hal ini misalkan dengan mendefinisikan sebuah integer dengan nama angka berikutnya dan dua angka yang dimanfaatkan sebagai looping yaitu $t1$ dan $t2$. Nilai $t1$ dan $t2$ akan berubah tiap looping dengan menggunakan operasi pendefinisian kembali di dalam looping.
2. Diberikan sebuah input berupa bilangan bertipe integer X dan batas bilangan. Di sini programmer diminta untuk menampilkan semua bilangan yang berada pada rentang sampai batas bilangan, tetapi bilangan-bilangan tersebut tidak ada yang habis dibagi bilangan X . Alur logika dalam pengerjaan soal ini adalah dengan memanfaatkan *looping* dengan kondisi awal $i = 1$ dan sampai batas $i \leq 10$ dengan *increment* $i++$. Kemudian pada operasi looping ditambahkan sebuah kondisi *if-else* yang mana jika i dimodulo-kan dengan X bernilai nol maka *looping* akan terlewat atau tidak ditampilkan dalam barisan. Dalam hal ini dimanfaatkan fungsi *continue* setelah kondisi tersebut. Maka dari itu, barisan akan menampilkan bilangan antara 1 sampai dengan 10 tetapi tidak menampilkan bilangan yang habis dibagi dengan X .
3. Diberikan sebuah input berupa bilangan bertipe integer n . Di sini programmer diminta untuk menampilkan semua barisan bintang yang membentuk segitiga dengan jumlah bintang yang terus bertambah tiap barisnya sesuai dengan jumlah inputan n yang diberikan. Pertambahan bintang setiap barisnya sesuai dengan jumlah baris yang diinginkan. Alur logika dalam pengerjaan soal ini adalah dengan memanfaatkan *looping* bersarang (*nested loop*) yang mana dideklarasikan integer $i = 1$ dengan batas $i \leq n$ input dan *increment* $i++$ kemudian pada *loop* kedua int $j = 1$ dengan batas $j \leq i$ karena setiap pengulangan akan naik sebanyak 1 angka untuk setiap pengulangan i . Kemudian untuk menampilkan tampilan $*$ yang lebih rapi akan ditambahkan spasi setelah bintang pada outputnya yang berada pada operasi j (" $*$ ") dan ditambahkan sebuah perintah untuk menampilkan outputan yang terurut ke bawah pada operasi i (" \n "). Variabel n ini akan menjadi penentu untuk jumlah baris i yang akan ditampilkan dan banyaknya bintang pada baris i dengan *looping* pada kolom ke- j .
4. Salah satu contoh pemrograman untuk menerapkan konsep *break* dan *continue* adalah ketika dalam barisan angka 1 – 10 programmer ingin menampilkan barisan dalam rentang 1 sampai 8 saja tetapi bilangan 5 tidak ditampilkan. Kegunaan *break* dan *continue* adalah dalam program ini *continue* berfungsi untuk tidak menampilkan bilangan 5 dalam *looping* menampilkan barisan angka 1-10 dan akan melanjutkan *looping* berikutnya karena kondisi yang diberikan *continue* akan dilewati. Kemudian *break* di sini berfungsi untuk memberhentikan *looping* ketika menampilkan angka 9 pada proses *looping* tersebut dalam hal ini *looping* bertambahnya i akan berhenti jika menjadi angka 9 sehingga barisan angka 1-10 yang hanya ingin menampilkan rentang 1 – 9 saja dan angka 5 tidak ditampilkan dapat

dijalankan.

BAB 2

SOURCE CODE

Soal 1

```
import java.util.Scanner;

public class soal_1 {

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Masukkan panjang deret fibonacci yang diinginkan :
" );

        int n = input.nextInt();
        int n1 = 1, n2 = 1;
        int nextnumber = 0;
        System.out.println();
        for (int i = 1; i <= n ; i++){
            if(i==1){
                System.out.print(n1 + " ");
                continue;
            }
            if(i==2){
                System.out.print(n2 + " ");
                continue;
            }
            nextnumber = n1+n2;
            n1 = n2;
            n2 = nextnumber;

            System.out.print(nextnumber + " ");

        }

    }

}
```

Soal 2

```
import java.util.Scanner;

public class soal_2 {

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Masukkan batas bilangan : ");
        int batas_bilangan = input.nextInt();
        System.out.print("Masukkan bilangan X : ");
        int x = input.nextInt();

        for(int i = 1; i<= batas_bilangan; i++){
            if (i % x == 0){
```

```

        continue;
    }
    System.out.print(i + " ");
}

}

}

```

Soal 3

```

import java.util.Scanner;

public class soal_3 {

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Masukkan panjang tinggi segitiga bintang : ");
        int n = input.nextInt();

        for(int i=1; i<=n; i++) {
            for(int j=1; j<=i; j++) {
                System.out.print(" *");
            }
            System.out.println();
        }
    }
}

```

Soal 4

```

import java.util.Scanner;

public class soal_4 {

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Masukkan panjang barisan yang diingkan : ");
        int n = input.nextInt();
        System.out.println("Hanya ingin menampilkan barisan 1-9 dan 5 tidak
ditampilkan : ");

        for(int i=1; i<=n; i++) {
            if (i == 5){
                continue;
            }else if(i == 9){
                break;
            }
            System.out.print(i + " ");
        }
    }
}

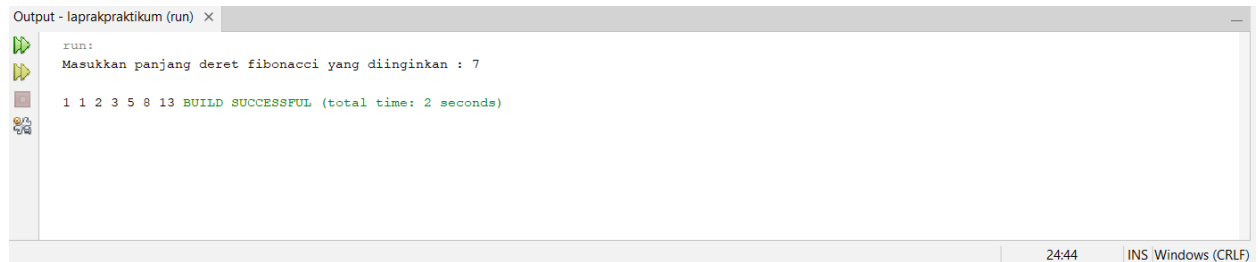
```

}

BAB 3

OUTPUT PROGRAM

Soal 1

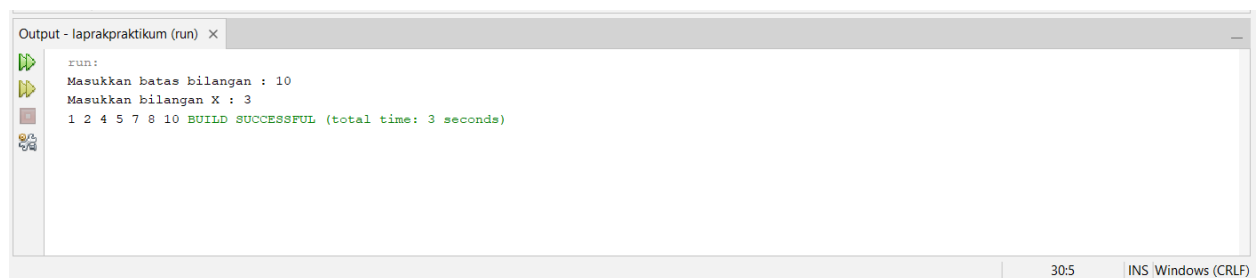


The screenshot shows a Java IDE output window titled "Output - laprakpraktikum (run)". The output text is as follows:

```
run:
Masukkan panjang deret fibonacci yang diinginkan : 7
1 1 2 3 5 8 13 BUILD SUCCESSFUL (total time: 2 seconds)
```

At the bottom right of the window, the time "24:44" and the text "INS Windows (CRLF)" are visible.

Soal 2

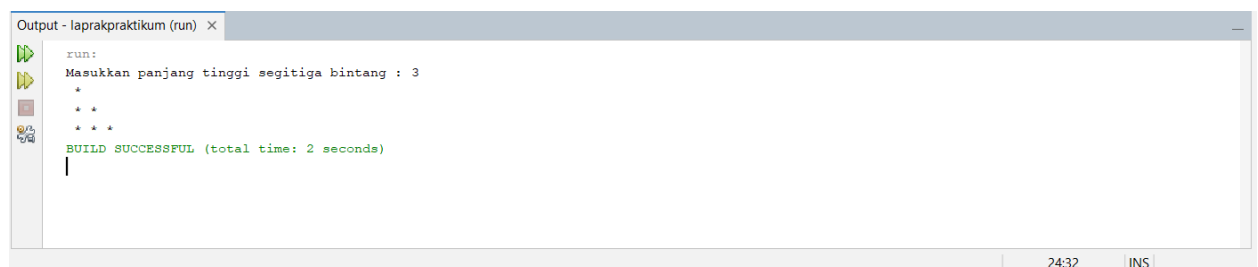


The screenshot shows a Java IDE output window titled "Output - laprakpraktikum (run)". The output text is as follows:

```
run:
Masukkan batas bilangan : 10
Masukkan bilangan X : 3
1 2 4 5 7 8 10 BUILD SUCCESSFUL (total time: 3 seconds)
```

At the bottom right of the window, the time "30:5" and the text "INS Windows (CRLF)" are visible.

Soal 3







The screenshot shows a Java IDE output window titled "Output - laprakpraktikum (run)". The output text is as follows:

```
run:
Masukkan panjang tinggi segitiga bintang : 3
*
* *
* * *
BUILD SUCCESSFUL (total time: 2 seconds)
```

At the bottom right of the window, the time "24:32" and the text "INS" are visible.

Soal 4

Output - laprakpraktikum (run) ×



```
run:
Masukkan panjang barisan yang diinginkan : 10
Hanya ingin menampilkan barisan 1-9 dan 5 tidak ditampilkan :
1 2 3 4 6 7 8 BUILD SUCCESSFUL (total time: 4 seconds)
|
```

24:28INS

