LAPORAN PRAKTIKUM BASIS DATA LANJUT

FUNGSI 2

Oleh:

MOCHAMMAD SYAIFUDDIN ZUHRI NIM. 1941720013



PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

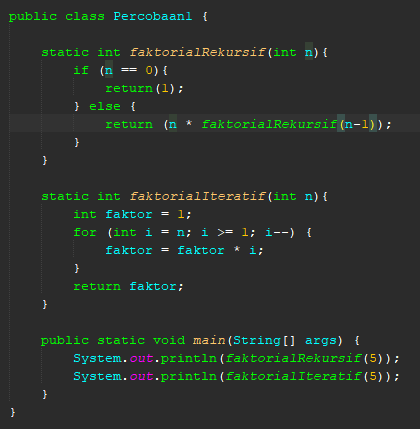
POLITEKNIK NEGERI MALANG

7 DESEMBER 2019

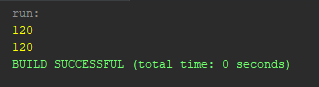
1. **PERCOBAAN 1**

* **Source Code**

|  |  |
| --- | --- |
| 1 | public class Percobaan1 { |
| 2 |  |
| 3 | static int faktorialRekursif(int n){ |
| 4 | if (n == 0){ |
| 5 | return(1); |
| 6 | } else { |
| 7 | return (n \* faktorialRekursif(n-1)); |
| 8 | } |
| 9 | } |
| 10 |  |
| 11 | static int faktorialIteratif(int n){ |
| 12 | int faktor = 1; |
| 13 | for (int i = n; i >= 1; i--) { |
| 14 | faktor = faktor \* i; |
| 15 | } |
| 16 | return faktor; |
| 17 | } |



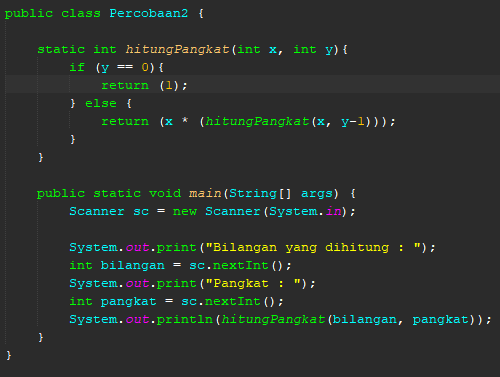
* **Capture Screen**



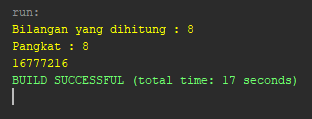
1. **PERCOBAAN 2**

* **Source Code**

|  |  |
| --- | --- |
| 1 | public class Percobaan2 { |
| 2 |  |
| 3 | static int hitungPangkat(int x, int y){ |
| 4 | if (y == 0){ |
| 5 | return (1); |
| 6 | } else { |
| 7 | return (x \* (hitungPangkat(x, y-1))); |
| 8 | } |
| 9 | } |
| 10 |  |
| 11 | public static void main(String[] args) { |
| 12 | Scanner sc = new Scanner(System.in); |
| 13 |  |
| 14 | System.out.print("Bilangan yang dihitung : "); |
| 15 | int bilangan = sc.nextInt(); |
| 16 | System.out.print("Pangkat : "); |
| 17 | int pangkat = sc.nextInt(); |
| 18 | System.out.println(hitungPangkat(bilangan, pangkat)); |
| 19 | } |
| 20 | } |



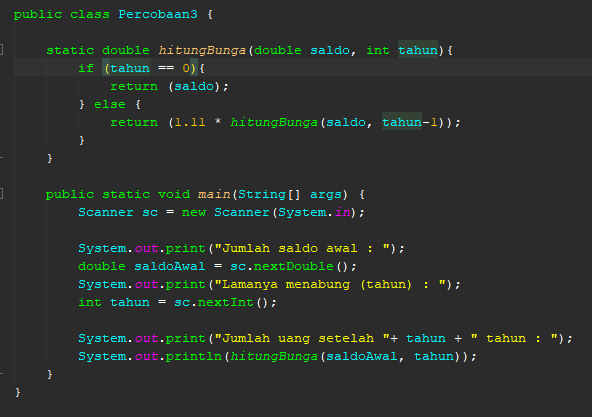
* **Capture Screen**



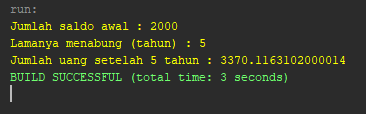
1. **PERCOBAAN 3**

* **Source Code**

|  |  |
| --- | --- |
| 1 | public class Percobaan3 { |
| 2 |  |
| 3 | static double hitungBunga(double saldo, int tahun){ |
| 4 | if (tahun == 0){ |
| 5 | return (saldo); |
| 6 | } else { |
| 7 | return (1.11 \* hitungBunga(saldo, tahun-1)); |
| 8 | } |
| 9 | } |
| 10 |  |
| 11 | public static void main(String[] args) { |
| 12 | Scanner sc = new Scanner(System.in); |
| 13 |  |
| 14 | System.out.print("Jumlah saldo awal : "); |
| 15 | double saldoAwal = sc.nextDouble(); |
| 16 | System.out.print("Lamanya menabung (tahun) : "); |
| 17 | int tahun = sc.nextInt(); |
| 18 |  |
| 19 | System.out.print("Jumlah uang setelah "+ tahun + " tahun : "); |
| 20 | System.out.println(hitungBunga(saldoAwal, tahun)); |
| 21 | } |
| 22 | } |



* **Capture Screen**



**PERTANYAAN!**

1. Apa yang dimaksud dengan fungsi rekursif?

**Jawab :**

Fungsi rekursif adalah fungsi yang memanggil dirinya sendiri. Hal ini bisa terjadi karena di dalam suatu fungsi rekursi, terdapat statement/baris program yang memanggil fungsi itu sendiri. Sehingga ketika suatu fungsi rekursif dipanggil/dijalankan, dan ketika proses eksekusi sudah sampai pada statement pemanggilan fungsi itu sendiri, maka fungsi tersebut akan dipanggil/dijalankan kembali. Dan LAGI, ketika proses eksekusi sudah sampai pada statement pemanggilan fungsi itu sendiri, maka fungsi tersebut akan dipanggil/dijalankan kembali, DEMIKIAN SETERUSNYA., hingga didapatkan suatu KONDISI AKHIR dimana proses pemanggilan fungsi tersebut tidak dilakukan lagi. Jika kondisi akhir tersebut tidak ditemukan/tidak ada, maka fungsi tersebut akan dipanggil terus menerus (infinite loop) dan ini tidak diperbolehkan.

1. Pada kasus-kasus seperti apa fungsi rekursif digunakan?

**Jawab :**

Fungsi rekursif biasanya digunakan untuk kasus yang berulang (Seperti fungsi loop) misalnya program menemukan nilai factorial sebuah bilangan.

1. Pada **Percobaan1**, apakah hasil yang diberikan fungsi **faktorialRekursif()** dan fungsi **faktorialIteratif()** sama? Jelaskan perbedaan alur jalannya program pada penggunaan fungsi rekursif dan fungsi iteratif!

**Jawab :**

Iya, Hasil dari fungsi *faktorialRekursif()* dan *faktorialIteratif()* sama.

Alur dari fungsi rekursif adalah didalam fungsi tersebut ada statement yang memanggil fungsi dirinya sendiri. Fungsi tersebut akan tersebut dilakukan atau terus dipanggil selama fungsi tersebut masih dalam kondisi yang benar. Fungsi akan berhenti jika kondisi fungsi sudah tidak ditemukan. Sedangkan fungsi iterative adalah fungsi yang menggunakan perulangan dimana dalam perulangan tersebut tidak perlu atau tidak ada pemanggilan fungsi itu sendiri. Dan statement akan terus dijalankan selama kondisi for sudah terpenuhi.

1. Pada **Percobaan2**, terdapat pemanggilan fungsi rekursif **hitungPangkat(bilangan, pangkat)** pada fungsi main, kemudian dilakukan pemanggilan fungsi **hitungPangkat()** secara berulangkali.Jelaskan sampai kapan proses pemanggilan fungsi tersebut akan dijalankan!

**Jawab :**

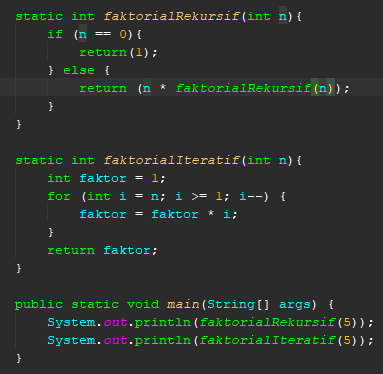
Ketika fungsi rekursif hitungPangkat(bilangan,pangkat) dipanggil/dijalankan dan kemudian eksekusi sudah sampai pada statement pemanggilan fungsi itu sendiri, maka fungsi tersebut akan dipanggil/dijalankan kembali. Dan LAGI, ketika proses eksekusi sudah sampai pada statement pemanggilan fungsi itu sendiri, maka fungsi tersebut akan dipanggil/dijalankan kembali, BEGITU SETERUSNYA hingga didapatkan suatu KONDISI AKHIR dimana proses pemanggilan fungsi tersebut tidak dilakukan lagi. Jika kondisi akhir tersebut tidak ditemukan/tidak ada, maka fungsi tersebut akan dipanggil terus menerus (infinite loop) dan ini tidak diperbolehkan.

1. Pada **Percobaan2**, jika pada saat pemanggilan fungsi **faktorialRekursif** nilai parameter yang dilewatkan diganti dari **n-1** menjadi **n** saja, sehingga sintaks program menjadi:

return n \* faktorialRekursif(n);

Apa yang terjadi? Mengapa demikian?

**Jawab :**

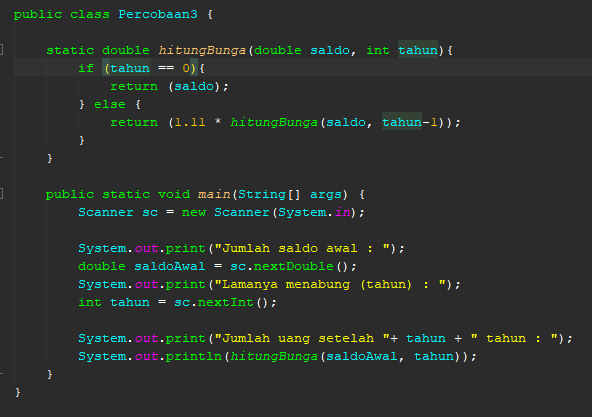




Kode program tersebut error. Karena Idenya adalah mengurangi ukuran permasalahan sampai menjadi kasus sederhana yang mempunyai penyelesaian jelas.Namun Idenya Tidak sesuai dengan nilai. Recursioncall/ Reductionstep Fungsi rekursifkonvergen (mendekat) ke arah nilai batas Biasanya mempunyai keyword retur nuntuk mengembalikan nilai ke fungsi yang memanggilnya.Untuk itu dalam memanggil fungsi itu sendiri harus sesuai dengan nilai logika untuk pembuatan nilai Faktorial.

1. Pada **Percobaan3**, sebutkan blok kode program manakah yang merupakan “base case” dan “recursion call”!

**Jawab :**

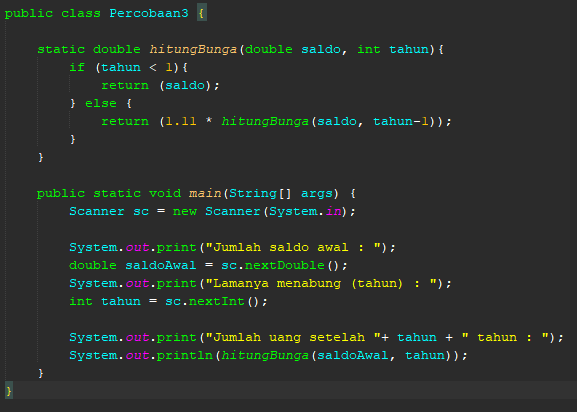


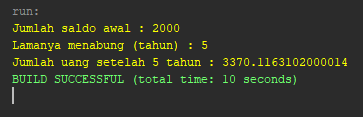
**Recursion Call**

**Base Case**

1. Pada **Percobaan3**, fungsi hitungBunga, terdapat kondisi IF (tahun == 0), ganti kode tersebut menjadi IF (tahun < 1). Apa yang terjadi? Jelaskan mengapa demikian!

**Jawab :**





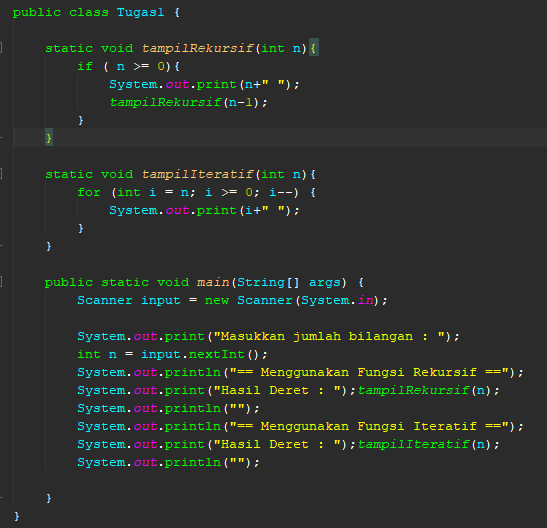
Hasilnya akan sama dengan percobaan sebelumnya. Hal itu terjadi karena adanya nilai return yang mengembalikan nilai pada fungsi itu sendiri dengan nama fungsi hitungBunga.

**TUGAS!**

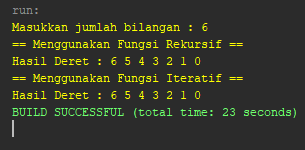
1. **Deret Descending Rekursif**

* **Source Code**

|  |  |
| --- | --- |
| 1 | public class Tugas1 { |
| 2 |  |
| 3 | static void tampilRekursif(int n){ |
| 4 | if ( n >= 0){ |
| 5 | System.out.print(n+" "); |
| 6 | tampilRekursif(n-1); |
| 7 | } |
| 8 | } |
| 9 |  |
| 10 | static void tampilIteratif(int n){ |
| 11 | for (int i = n; i >= 0; i--) { |
| 12 | System.out.print(i+" "); |
| 13 | } |
| 14 | } |
| 15 |  |
| 16 | public static void main(String[] args) { |
| 17 | Scanner input = new Scanner(System.in); |
| 18 |  |
| 19 | System.out.print("Masukkan jumlah bilangan : "); |
| 20 | int n = input.nextInt(); |
| 21 | System.out.println("== Menggunakan Fungsi Rekursif =="); |
| 22 | System.out.print("Hasil Deret : ");tampilRekursif(n); |
| 23 | System.out.println(""); |
| 24 | System.out.println("== Menggunakan Fungsi Iteratif =="); |
| 25 | System.out.print("Hasil Deret : ");tampilIteratif(n); |
| 26 | System.out.println(""); |
| 27 |  |
| 28 | } |
| 29 | } |



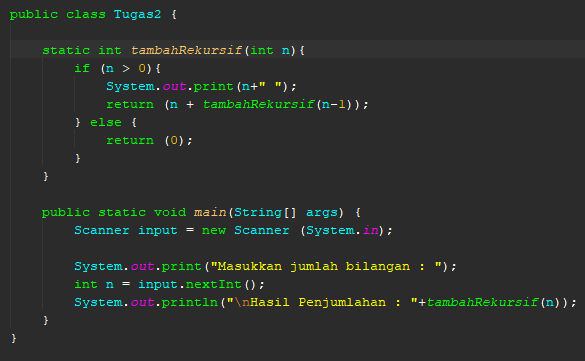
* **Capture Screen**



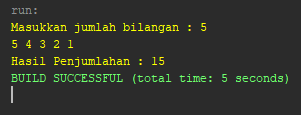
1. **Penjumlahan Rekursif**

* **Source Code**

|  |  |
| --- | --- |
| 1 | public class Tugas2 { |
| 2 |  |
| 3 | static int tambahRekursif(int n){ |
| 4 | if (n > 0){ |
| 5 | System.out.print(n+" "); |
| 6 | return (n + tambahRekursif(n-1)); |
| 7 | } else { |
| 8 | return (0); |
| 9 | } |
| 10 | } |
| 11 |  |
| 12 | public static void main(String[] args) { |
| 13 | Scanner input = new Scanner (System.in); |
| 14 |  |
| 15 | System.out.print("Masukkan jumlah bilangan : "); |
| 16 | int n = input.nextInt(); |
| 17 | System.out.println("\nHasil Penjumlahan : "+tambahRekursif(n)); |
| 18 | } |
| 19 | } |



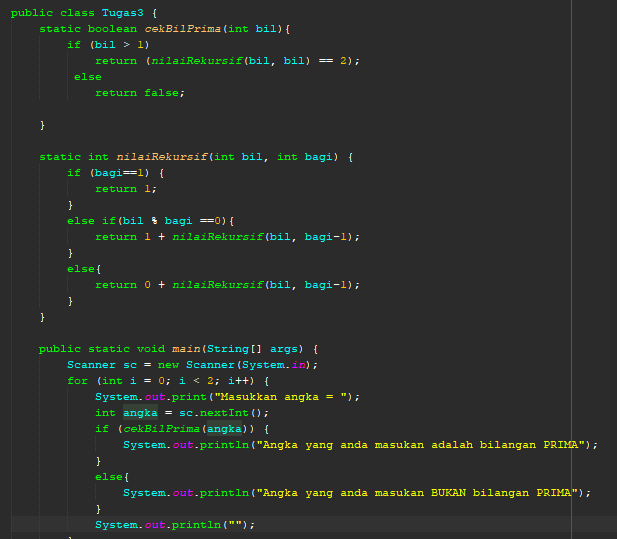
* **Capture Screen**



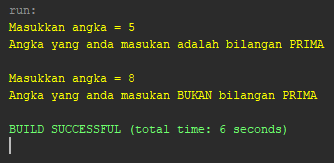
1. **Cek Prima Rekursif**

* **Source Code**

|  |  |
| --- | --- |
| 1 | public class Tugas3 { |
| 2 | static boolean cekBilPrima(int bil){ |
| 3 | if (bil > 1) |
| 4 | return (nilaiRekursif(bil, bil) == 2); |
| 5 | else |
| 6 | return false; |
| 7 |  |
| 8 | } |
| 9 |  |
| 10 | static int nilaiRekursif(int bil, int bagi) { |
| 11 | if (bagi==1) { |
| 12 | return 1; |
| 13 | } |
| 14 | else if(bil % bagi ==0){ |
| 15 | return 1 + nilaiRekursif(bil, bagi-1); |
| 16 | } |
| 17 | else{ |
| 18 | return 0 + nilaiRekursif(bil, bagi-1); |
| 19 | } |
| 20 | } |
| 21 |  |
| 22 | public static void main(String[] args) { |
| 23 | Scanner sc = new Scanner(System.in); |
| 24 | for (int i = 0; i < 2; i++) { |
| 25 | System.out.print("Masukkan angka = "); |
| 26 | int angka = sc.nextInt(); |
| 27 | if (cekBilPrima(angka)) { |
| 28 | System.out.println("Angka yang anda masukan adalah bilangan PRIMA"); |
| 29 | } |
| 30 | else{ |
| 31 | System.out.println("Angka yang anda masukan BUKAN bilangan PRIMA"); |
| 32 | } |
| 33 | System.out.println(""); |
| 34 | } |
| 35 | } |
| 36 | } |



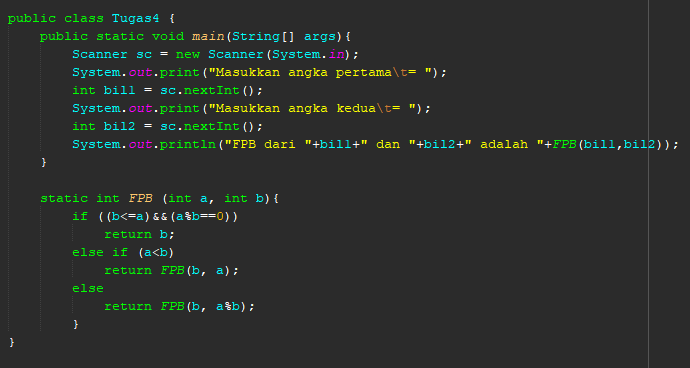
* **Capture Screen**



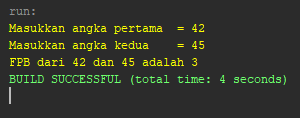
1. **FPB Euclid Rekursif**

* **Source Code**

|  |  |
| --- | --- |
| 1 | public class Tugas4 { |
| 2 | public static void main(String[] args){ |
| 3 | Scanner sc = new Scanner(System.in); |
| 4 | System.out.print("Masukkan angka pertama\t= "); |
| 5 | int bil1 = sc.nextInt(); |
| 6 | System.out.print("Masukkan angka kedua\t= "); |
| 7 | int bil2 = sc.nextInt(); |
| 8 | System.out.println("FPB dari "+bil1+" dan "+bil2+" adalah "+FPB(bil1,bil2)); |
| 9 | } |
| 10 |  |
| 11 | static int FPB (int a, int b){ |
| 12 | if ((b<=a)&&(a%b==0)) |
| 13 | return b; |
| 14 | else if (a<b) |
| 15 | return FPB(b, a); |
| 16 | else |
| 17 | return FPB(b, a%b); |
| 18 |  |
| 19 | } |



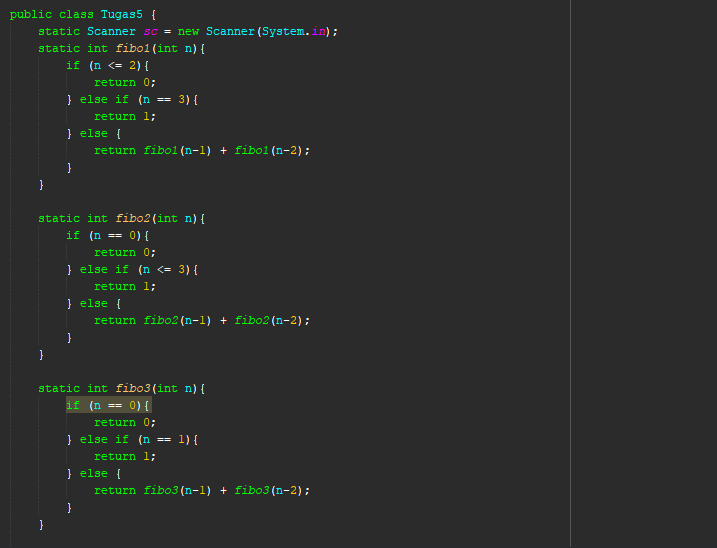
* **Capture Screen**

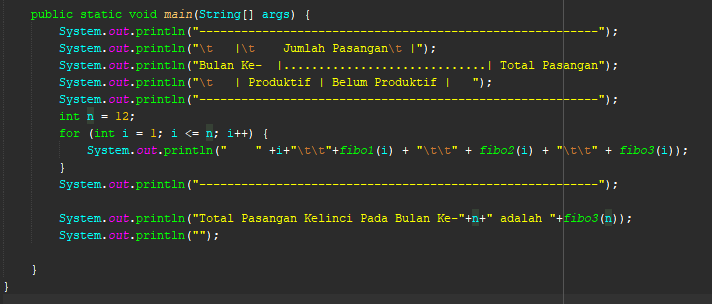


1. **Fibonacci**

* **Source Code**

|  |  |
| --- | --- |
| 1 | public class Tugas5 { |
| 2 | static Scanner sc = new Scanner(System.in); |
| 3 | static int fibo1(int n){ |
| 4 | if (n <= 2){ |
| 5 | return 0; |
| 6 | } else if (n == 3){ |
| 7 | return 1; |
| 8 | } else { |
| 9 | return fibo1(n-1) + fibo1(n-2); |
| 10 | } |
| 11 | } |
| 12 |  |
| 13 | static int fibo2(int n){ |
| 14 | if (n == 0){ |
| 15 | return 0; |
| 16 | } else if (n <= 3){ |
| 17 | return 1; |
| 18 | } else { |
| 19 | return fibo2(n-1) + fibo2(n-2); |
| 20 | } |
| 21 | } |
| 22 |  |
| 23 | static int fibo3(int n){ |
| 24 | if (n == 0){ |
| 25 | return 0; |
| 26 | } else if (n == 1){ |
| 27 | return 1; |
| 28 | } else { |
| 29 | return fibo3(n-1) + fibo3(n-2); |
| 30 | } |
| 31 | } |
| 32 |  |
| 33 | public static void main(String[] args) { |
| 34 | System.out.println("---------------------------------------------------------"); |
| 35 | System.out.println("\t |\t Jumlah Pasangan\t |"); |
| 36 | System.out.println("Bulan Ke- |.............................| Total Pasangan"); |
| 37 | System.out.println("\t | Produktif | Belum Produktif | "); |
| 38 | System.out.println("---------------------------------------------------------"); |
| 39 | int n = 12; |
| 40 | for (int i = 1; i <= n; i++) { |
| 41 | System.out.println(" " +i+"\t\t"+fibo1(i) + "\t\t" + fibo2(i) + "\t\t" + fibo3(i)); |
| 42 | } |
| 43 | System.out.println("---------------------------------------------------------"); |
| 44 |  |
| 45 | System.out.println("Total Pasangan Kelinci Pada Bulan Ke-"+n+" adalah "+fibo3(n)); |
| 46 | System.out.println(""); |
| 47 |  |
| 48 | } |
| 49 | } |





* **Capture Screen**

