LAPORAN 14 DASAR PEMROGRAMAN

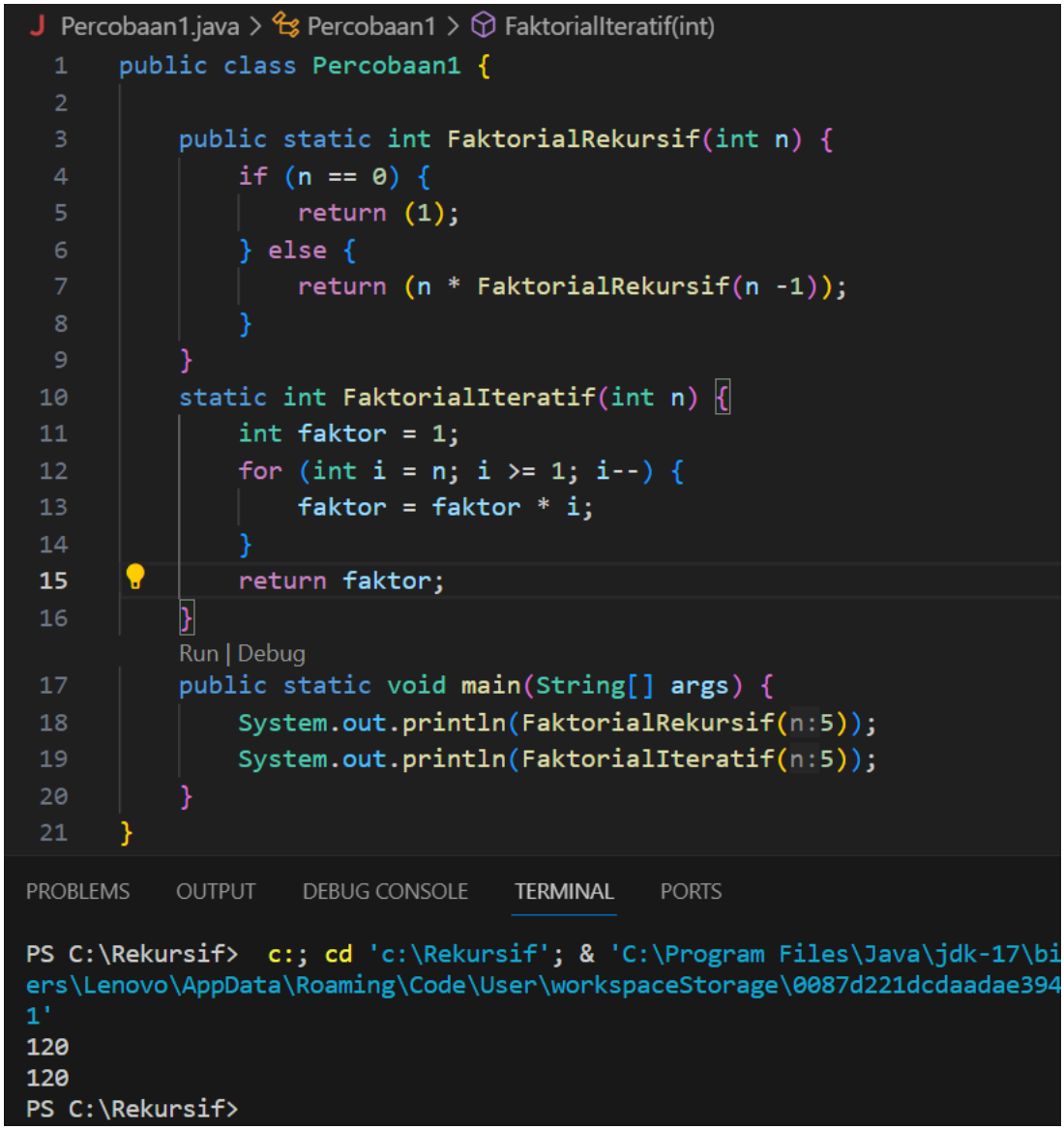
NAMA : CINDY LAILI LARASATI

NIM : 2341720038

KELAS : 1B

PRODI : D-IV TEKNIK INFORMATIKA

Percobaan 1



```
Percobaan1.java > Percobaan1 > FaktorialIteratif(int)
1   public class Percobaan1 {
2
3       public static int FaktorialRekursif(int n) {
4           if (n == 0) {
5               return (1);
6           } else {
7               return (n * FaktorialRekursif(n -1));
8           }
9       }
10      static int FaktorialIteratif(int n) {
11          int faktor = 1;
12          for (int i = n; i >= 1; i--) {
13              faktor = faktor * i;
14          }
15          return faktor;
16      }
        Run | Debug
17      public static void main(String[] args) {
18          System.out.println(FaktorialRekursif(n:5));
19          System.out.println(FaktorialIteratif(n:5));
20      }
21  }

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Rekursif>  c:; cd 'c:\Rekursif'; & 'C:\Program Files\Java\jdk-17\bi
ers\Lenovo\AppData\Roaming\Code\User\workspaceStorage\0087d221dcdaadae394
1'
120
120
PS C:\Rekursif>
```

Pertanyaan!

1. Apa yang dimaksud dengan fungsi rekursif?
   Jawab : Rekursi adalah sebuah konsep di pemrograman di mana suatu fungsi dapat memanggil dirinya sendiri selama proses eksekusi, menyelesaikan submasalah yang lebih kecil pada setiap iterasi.
2. Bagaimana contoh kasus penggunaan fungsi rekursif ?
   Jawab : Berdasarkan contoh tersebut, fungsi rekursif digunakan untuk menghitung hasil faktorial
3. 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, sama. Fungsi rekursif akan memanggil dirinya sendiri dengan parameter n-1 dan mengembalikan hasil perkalian n dengan hasil rekursif n-1. Ketika n bernilai 0, fungsi akan langsung mengembalikan nilai 1. Jika n tidak sama dengan 0, fungsi akan

melakukan pemanggilan rekursif dengan argumen n - 1 dan hasilnya dikalikan dengan n. Sebaliknya, fungsi iteratif menggunakan loop for. Pada fungsi ini, variabel faktor diinisialisasi dengan nilai 1, dan kemudian dilakukan perkalian faktor dengan setiap angka dari n hingga 1.

Percobaan 2

Percobaan2.java > Percobaan2 > hitungPangkat(int, int)

```java
import java.util.Scanner;
public class Percobaan2 {

    public static int hitungPangkat(int x, int y) {
        if (y == 0) {
            return (1);
        } else {
            return (x * hitungPangkat(x, y - 1));
        }
    }
        Run | Debug
        public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
            System.out.print(s:"Bilangan yang dihitung: ");
            int bilangan = sc.nextInt();
            System.out.print(s:"Pangkat: ");
            int pangkat = sc.nextInt();
            System.out.println(hitungPangkat(bilangan, pangkat));
        }
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
rs\Lenovo\AppData\Roaming\Code\User\workspaceStorage\0087d221dcdaadae39495d5725
bin' 'Percobaan2'
Bilangan yang dihitung: 7
Pangkat: 3
343
PS C:\Rekursif>
```

Pertanyaan !

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 : Proses pemanggilan fungsi berlanjut hingga fungsi rekursif mengembalikan nilai 1, dengan kata lain, proses ini berlangsung sampai nilai parameter (y atau pangkat) mencapai 0.

2. Tambahkan kode program untuk mencetak deret perhitungan pangkatnya. Contoh : hitungPangkat(2,5) dicetak 2x2x2x2x2x1 = 32

```
for (int i = 0; i < pangkat; i++) {
 System.out.print(bilangan);
 if (i < pangkat - 1) {
     System.out.print(s:" x ");
```

```
Bilangan yang dihitung: 7
Pangkat: 3
7 x 7 x 7 = 343
PS C:\Rekursif>
```

Percobaan 3

Percobaan3.java > Percobaan3 > main(String[])

```
1   import java.util.Scanner;
2
3   public class Percobaan3 {
4       static double hitungLaba(double saldo, int tahun) {
5           if (tahun == 0) {
6               return (saldo);
7           } else {
8               return (1.11 * hitungLaba(saldo, tahun - 1));
9           }
10      }
        Run | Debug
11      public static void main(String[] args) {
12          Scanner sc = new Scanner(System.in);
13          System.out.print(s:"Jumlah saldo awal : ");
14          double saldoAwal = sc.nextDouble();
15          System.out.print(s:"Lamanya investasi (tahun) : ");
16          int tahun = sc.nextInt();
17          System.out.print("Jumlah saldo setelah " + "tahun : ");
18          System.out.print(hitungLaba(saldoAwal, tahun));
```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
ing\Code\User\workspaceStorage\0087d221dcdaadae39495d57254cb396\redhat.java\jdt
Jumlah saldo awal : 100000
Lamanya investasi (tahun) : 3
Jumlah saldo setelah tahun : 136763.10000000003
PS C:\Rekursif>
```

Pertanyaan !

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

Base case :

```
if (tahun == 0) {
    return (saldo);
```

Base case :

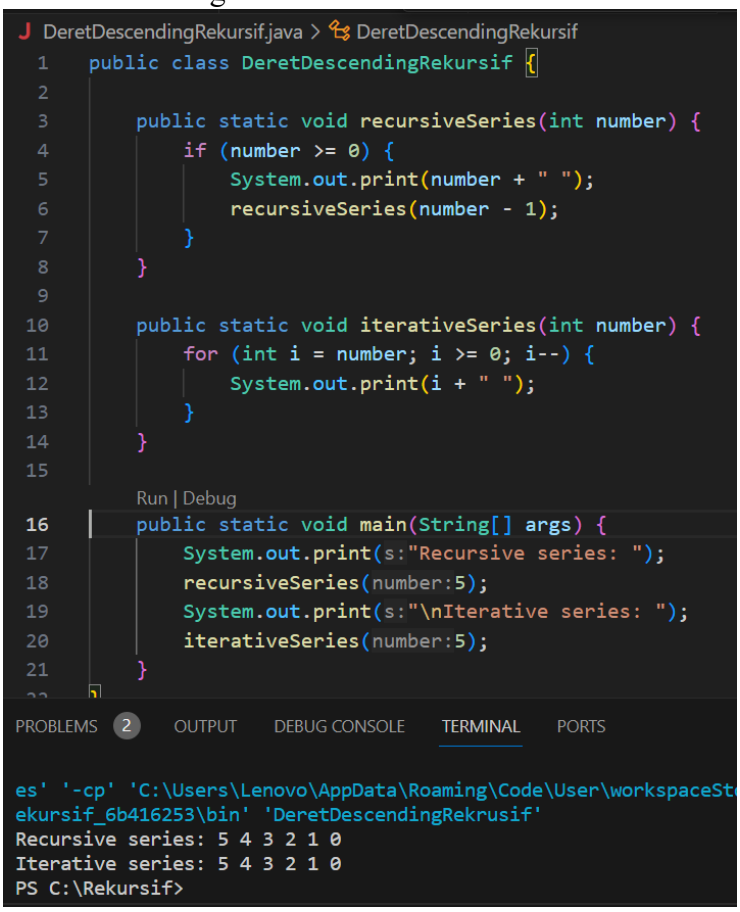
```
return (1.11 * hitungLaba(saldo, tahun - 1));
```

2. Jabarkan trace fase ekspansi dan fase subtitusi algoritma perhitungan laba di atas jika diberikan nilai hitungLaba(100000,3)
   Jawab :
   Fase ekspansi :
   Pemanggilan rekursif dipecah menjadi bentuk yang lebih sederhana. Setiap pemanggilan rekursif menghasilkan bentuk yang lebih kecil hingga mencapai kondisi dasar (tahun == 0). Contoh :
   hitungLaba(100000, 3)
   1.11 * hitungLaba(100000, 2)
   1.11 * (1.11 * hitungLaba(100000, 1))
   1.11 * (1.11 * (1.11 * hitungLaba(100000, 0)))
   1.11 * (1.11 * (1.11 * 100000))
   Fase subtitusi :
   menggantikan nilai dari pemanggilan fungsi dengan hasil perhitungan yang sesuai. Pada akhirnya, kita mendapatkan nilai akhir dari pemanggilan hitungLaba(100000, 3).
   Contoh :

   hitungLaba(100000, 3)
   = 1.11 * (1.11 * (1.11 * 100000))
   = 1.11 * (1.11 * (111000)
   = 1.11 * (122100)
   = 135531

Tugas

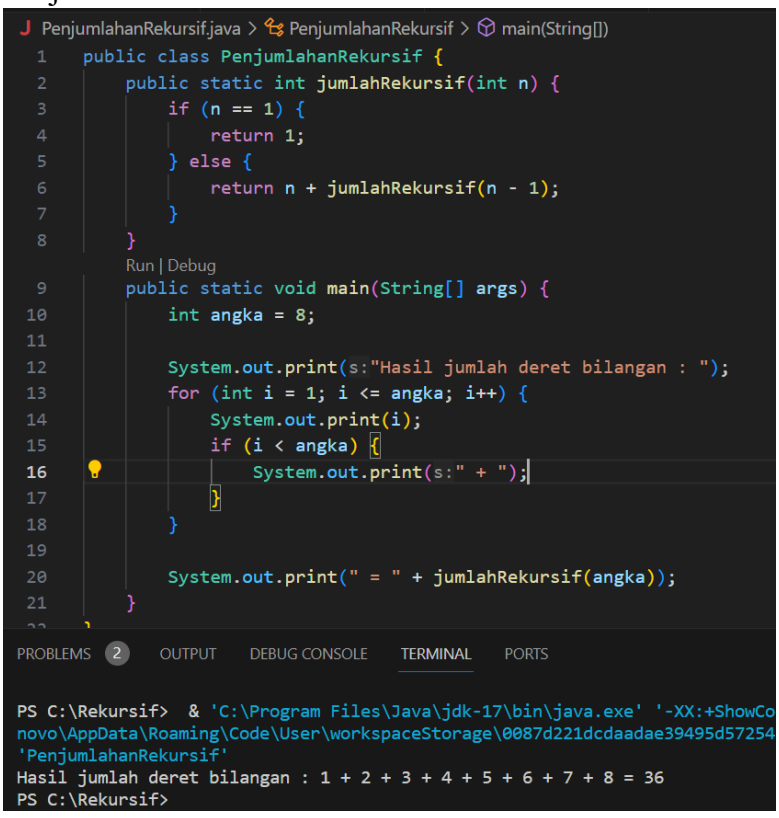
1. Deret descending rekursif



```java
public class DeretDescendingRekursif {

    public static void recursiveSeries(int number) {
        if (number >= 0) {
            System.out.print(number + " ");
            recursiveSeries(number - 1);
        }
    }

    public static void iterativeSeries(int number) {
        for (int i = number; i >= 0; i--) {
            System.out.print(i + " ");
        }
    }

    Run | Debug
    public static void main(String[] args) {
        System.out.print(s:"Recursive series: ");
        recursiveSeries(number:5);
        System.out.print(s:"\nIterative series: ");
        iterativeSeries(number:5);
    }
}
```

```
es' '-cp' 'C:\Users\Lenovo\AppData\Roaming\Code\User\workspaceSt
ekursif_6b416253\bin' 'DeretDescendingRekrusif'
Recursive series: 5 4 3 2 1 0
Iterative series: 5 4 3 2 1 0
PS C:\Rekursif>
```

2. Penjumlahan rekursif



```java
public class PenjumlahanRekursif {
    public static int jumlahRekursif(int n) {
        if (n == 1) {
            return 1;
        } else {
            return n + jumlahRekursif(n - 1);
        }
    }
    Run | Debug
    public static void main(String[] args) {
        int angka = 8;

        System.out.print(s:"Hasil jumlah deret bilangan : ");
        for (int i = 1; i <= angka; i++) {
            System.out.print(i);
            if (i < angka) {
                System.out.print(s:" + ");
            }
        }

        System.out.print(" = " + jumlahRekursif(angka));
    }
}
```

```
PS C:\Rekursif>  & 'C:\Program Files\Java\jdk-17\bin\java.exe' '-XX:+ShowCo
novo\AppData\Roaming\Code\User\workspaceStorage\0087d221dcdaadae39495d57254
'PenjumlahanRekursif'
Hasil jumlah deret bilangan : 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 = 36
PS C:\Rekursif>
```

3. Cek prima rekursif

```java
public class CekPrimaRekursif {
    public static void cekPrimaRekursif(int n, int pembagi) {
        if (n <= 1) {
            System.out.println(n + " bukan bilangan prima.");
        } else if (pembagi == 1) {
            System.out.println(n + " adalah bilangan prima.");
        } else if (n % pembagi == 0) {
            System.out.println(n + " bukan bilangan prima.");
        } else {
            // memanggil rekursif dengan pembagi yang lebih kecil
            cekPrimaRekursif(n, pembagi - 1);
        }
    }
    Run | Debug
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Masukkan sebuah bilangan: ");
        int angka = sc.nextInt();

        cekPrimaRekursif(angka, angka - 1);
    }
```

```
PROBLEMS 3  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Rekursif>  & 'C:\Program Files\Java\jdk-17\bin\java.exe' '-XX:+ShowCodeDe
novo\AppData\Roaming\Code\User\workspaceStorage\0087d221dcdaadae39495d57254cb39
'CekPrimaRekursif'
Masukkan sebuah bilangan: 5
5 adalah bilangan prima.
PS C:\Rekursif>
```

4. Fibonacci

```java
public class Fibonacci {
    Run | Debug
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Masukkan bulan ke: ");
        int bulan = sc.nextInt();

        int[] jmlMarmut = hitungPasanganMarmut(bulan);

        System.out.println("Total pasangan marmut pada bulan ke-" + bulan + " adalah " + jmlMarmut[0]);
        System.out.println("Jumlah marmut produktif pada bulan ke-" + bulan + " adalah " + jmlMarmut[1]);
        System.out.println("Jumlah marmut belum produktif pada bulan ke-" + bulan + " adalah " + jmlMarmut[2]);
    }
    public static int[] hitungPasanganMarmut(int bulan) {
        int[] hasil = new int[3];
        if (bulan <= 2) {
            hasil[0] = 1;
            hasil[1] = 0;
            hasil[2] = 1;
            return hasil;
        }
        int[] marmutSebelumnya = hitungPasanganMarmut(bulan - 1);
        int totalPasangan = marmutSebelumnya[0] + marmutSebelumnya[2];
        int jumlahProduktif = marmutSebelumnya[2];
        hasil[0] = totalPasangan;
        hasil[1] = jumlahProduktif;
        hasil[2] = totalPasangan - jumlahProduktif;
        return hasil;
    }
```

```
novo\AppData\Roaming\Code\User\workspaceStorage\0087d221dcdaada
'Fibonacci'
Masukkan bulan ke: 4
Total pasangan marmut pada bulan ke-4 adalah 3
Jumlah marmut produktif pada bulan ke-4 adalah 1
Jumlah marmut belum produktif pada bulan ke-4 adalah 2
PS C:\Rekursif>
```