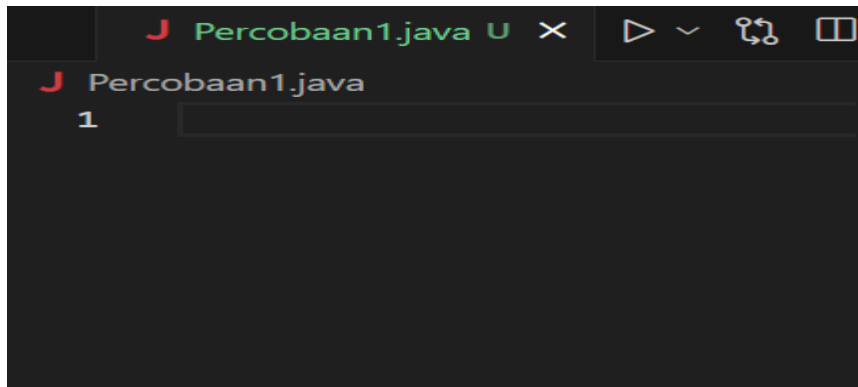


Nama : Adellia Salsa Al Barra  
Nim : 244107020222  
Kelas : TI 1B

## PERCOBAAN 1

1. Buat file baru



2. Buat fungsi static dengan nama faktorialRekursif(), dengan tipe data kembalian fungsi int dan memiliki 1 parameter dengan tipe data int berupa bilangan yang akan dihitung nilai faktorialnya

```
1 public class Percobaan1 {  
2     static int faktorialRekursif(int n) {  
3         if (n== 0) {  
4             return (1);  
5         } else {  
6             return (n * faktorialRekursif(n-1));  
7         }  
8     }  
}
```

3. Buat lagi fungsi static dengan nama faktorialIteratif(), dengan tipe data kembalian fungsi int dan memiliki 1 parameter dengan tipe data int berupa bilangan yang akan dihitung nilai faktorialnya

```
9     static int faktorialIteratif(int n) {  
10         int faktor = 1;  
11         for (int i = n; i >= 1; i--) {  
12             faktor = faktor * i;  
13         }  
14         return faktor;  
15     }  
}
```

4. Buatlah fungsi main dan lakukan pemanggilan terhadap kedua fungsi yang telah dibuat sebelumnya, dan tampilkan hasil yang didapatkan

```

Run | Debug
16      public static void main(String[] args) {
17          system.out.println(faktorialRekursif(n:5));
18          System.out.println(faktorialIteratif(n:5));
19      }
20  }

```

##### 5. Menjalankan program

```

PS C:\vscode\Rekursif> cd "c:\vscode\Rekursif\" ; if ($?) { javac Percobaan1.java } ; if ($?) { java Percobaan1 }
120
120
PS C:\vscode\Rekursif>

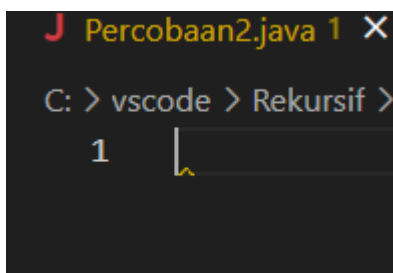
```

### PERTANYAAN

1. Fungsi rekursif adalah fungsi yang memanggil dirinya sendiri untuk menyelesaikan masalah. Ini berguna untuk masalah yang dapat dipecah menjadi sub-masalah yang lebih kecil
2.
  - a) Penghitungan faktorial
  - b) Pencarian dalam pohon biner atau graf
  - c) Menelusuri folder dalam file system yang bersifat hierarkis
3. Hasil yang diberikan sama karena metode rekursif maupun iteratif, keduanya menghitung nilai faktorial berdasarkan logika matematika yang sama
  - a) Alur Rekursif  
Memanggil dirinya sendiri hingga mencapai base case. Ini menggunakan lebih banyak memori karena setiap panggilan menambah lapisan di tumpukan
  - b) Alur Iteratif  
Menggunakan loop untuk menghitung hasil secara langsung tanpa memanggil dirinya sendiri, sehingga lebih efisien dalam penggunaan memori

### PERCOBAAN 2

1. Buat file baru



The screenshot shows a new file named 'Percobaan2.java' being created in the VS Code editor. The file is located in the directory 'C:\> vscode > Rekursif >'. The file is currently empty, with a cursor at the beginning of the first line.

2. Buat fungsi static dengan nama hitungPangkat(), dengan tipe data kembalian fungsi int dan memiliki 2 parameter dengan tipe data int berupa bilangan yang akan dihitung pangkatnya dan bilangan pangkatnya

```

3
4  static int hitungPangkat (int x, int y) {
5      if (y == 0) {
6          return (1);
7      } else {
8          return (x * hitungPangkat(x, y - 1));
9      }
10 }

```

3. Buatlah fungsi main dan deklarasikan Scanner dengan nama sc

```

11 public static void main(String[] args) {
12     Scanner sc = new Scanner(System.in);

```

4. Buatlah dua buah variabel bertipe int dengan nama bilangan dan pangkat

```

    int bilangan, pangkat;

```

5. Tambahkan kode berikut ini untuk menerima input dari keyboard

```

15     System.out.print(s:"Bilangan yang dihitung: ");
16     bilangan = sc.nextInt();
17     System.out.print(s:"pangkat: ");
18     pangkat = sc.nextInt();

```

6. Lakukan pemanggilan fungsi hitungPangkat yang telah dibuat sebelumnya dengan mengirimkan dua nilai parameter.

```

    System.out.println(hitungPangkat(bilangan, pangkat));
}

```

7. Menjalankan program

```

java Percobaan2 }
Bilangan yang dihitung: 2
pangkat: 2
4
PS C:\vscode\Rekursif> cd "c:\vscode\Rekursif\" ; if ($?) {
    java Percobaan2 }
Bilangan yang dihitung: 5
pangkat: 3
125
PS C:\vscode\Rekursif>

```

## PERTANYAAN

1. Fungsi rekursif akan berhenti saat pangkat bernilai 0. Ini disebut base case. Pada titik itu, fungsi mengembalikan 1 dan mulai menyelesaikan hasil dari setiap panggilan sebelumnya

```

1  import java.util.Scanner;
2
3  public class Percobaan2 {
4
5      static int hitungPangkat (int x, int y) {
6          if (y == 0) {
7              System.out.print(s:"1");
8              return (1);
9          } else {
10             System.out.print(x + (y > 1 ? "x" : ""));
11             return (x * hitungPangkat(x, y - 1));
12         }
13     }
14
15     Run | Debug
16     public static void main(String[] args) {
17         Scanner sc = new Scanner(System.in);
18         int bilangan, pangkat;
19
20         System.out.print(s:"Bilangan yang dihitung: ");
21         bilangan = sc.nextInt();
22         System.out.print(s:"pangkat: ");
23         pangkat = sc.nextInt();
24
25         System.out.print(s:"Perhitungan: ");
26         int hasil = hitungPangkat(bilangan, pangkat);
27         System.out.println(" = " + hasil);
28     }
29 }

```

2.

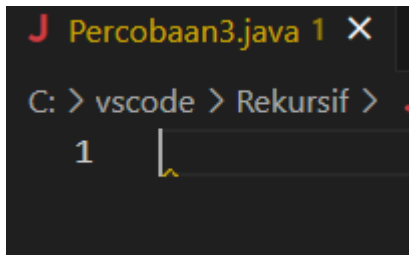
```

PS C:\vscode\Rekursif> cd "c:\vscode\Rekursif"
java Percobaan2 }
Bilangan yang dihitung: 2
pangkat: 5
Perhitungan: 2x2x2x2x21 = 32

```

## PERCOBAAN 3

1. Buat file baru



2. Buat fungsi static dengan nama `hitungLaba()`, dengan tipe data kembalian fungsi `double` dan memiliki 2 parameter dengan tipe data `int` berupa saldo investor dan lamanya investasi. Pada kasus ini dianggap laba yang ditentukan adalah 11% per tahun. Karena perhitungan laba adalah  $\text{laba} * \text{saldo}$ , sehingga untuk menghitung besarnya uang setelah ditambah laba adalah  $\text{saldo} + \text{laba} * \text{saldo}$ . Dalam hal ini, besarnya laba adalah  $0.11 * \text{saldo}$ , dan saldo dianggap  $1 * \text{saldo}$ , sehingga  $1 * \text{saldo} + 0.11 * \text{saldo}$  dapat diringkas menjadi  $1.11 * \text{saldo}$  untuk perhitungan saldo setelah ditambah laba (dalam setahun)

```
1  import java.util.Scanner;
2  public class Percobaan3 {
3
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 }
```

3. Buatlah fungsi main dan deklarasikan Scanner dengan nama `sc`

```
Run | Debug
12      public static void main(String[] args) {
13          Scanner sc = new Scanner(System.in);
```

4. Buatlah sebuah variabel bertipe `double` dengan nama `saldoAwal` dan sebuah variabel bertipe `int` bernama `tahun`

```
double saldoAwal;
int tahun;
```

5. Tambahkan kode berikut ini untuk menerima input dari keyboard

```
17      System.out.print(s:"Jumlah saldo awal: ");
18      saldoAwal = sc.nextDouble();
19      System.out.print(s:"Lamanya investasi (tahun): ");
20      tahun = sc.nextInt();
```

6. Lakukan pemanggilan fungsi `hitungLaba` yang telah dibuat sebelumnya dengan mengirimkan dua nilai parameter

```

22     System.out.print("Jumlah saldo setelah " + tahun + " tahun : ");
23     System.out.print(hitungLaba(saldoAwal, tahun));
24 }
25 }

```

## 7. Menjalankan program

```

PS C:\vscode\Rekursif> cd "c:\vscode\Rekursif\" ; if ($?) { javac Percobaan3.java
java Percobaan3 }
Jumlah saldo awal: 1000000
Lamanya investasi (tahun): 5
Jumlah saldo setelah 5 tahun : 1685058.1551000006
PS C:\vscode\Rekursif>

```

## PERTANYAAN

### 1. Base case :

```

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

```

Recursion call:

```

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

```

### 2. Fase ekspansi:

$$\begin{aligned}
 &\text{hitungLaba}(100000, 3) \\
 &= 1.11 * \text{hitungLaba}(100000, 2) \\
 &= 1.11 * (1.11 * \text{hitungLaba}(100000, 1)) \\
 &= 1.11 * (1.11 * (1.11 * \text{hitungLaba}(100000, 0))) \\
 &= 1.11 * (1.11 * (1.11 * 100000))
 \end{aligned}$$

Fase substitusi:

$$\begin{aligned}
 \text{hitungLaba}(100000, 0) &= 100000 \\
 \text{hitungLaba}(100000, 1) &= 1.11 * 100000 = 111000 \\
 \text{hitungLaba}(100000, 2) &= 1.11 * 111000 = 123210 \\
 \text{hitungLaba}(100000, 3) &= 1.11 * 123210 = 136743
 \end{aligned}$$

## TUGAS

```

1  import java.util.Scanner;
2  public class TugasNo1 {
3
4      static void deretDescendingRekursif(int n) {
5          if (n < 0) {
6              return;
7          }
8          System.out.print(n + " ");
9          deretDescendingRekursif(n - 1);
10     }
11
12     static void deretDescendingIteratif(int n) {
13         for (int i = n; i >= 0; i--) {
14             System.out.print(i + " ");
15         }
16     }
17
18     public static void main(String[] args) {
19         Scanner sc = new Scanner(System.in);
20
21         System.out.print(s:"Masukkan nilai n: ");
22         int n = sc.nextInt();
23
24         System.out.println(x:"Deret menggunakan fungsi Rekursif:");
25         deretDescendingRekursif(n);
26
27         System.out.println();
28
29         System.out.println(x:"Deret menggunakan fungsi Iteratif:");
30         deretDescendingIteratif(n);
31
32     }
33 }
34

```

1.

```

1  import java.util.Scanner;
2  public class TugasNo2 {
3
4      static int penjumlahan(int f) {
5          if (f == 0) {
6              return 0;
7          }
8          return f + penjumlahan(f - 1);
9      }
10
11     static void tampilkanPerhitungan(int f) {
12         if (f == 1) {
13             System.out.print(s:"1");
14             return;
15         }
16         tampilkanPerhitungan(f - 1);
17         System.out.print(" + " + f);
18     }
19
20     public static void main(String[] args) {
21         Scanner sc = new Scanner(System.in);
22         int f, hasil = 0;
23         System.out.print(s:"Masukkan nilai f= ");
24         f = sc.nextInt();
25
26         System.out.print(s:"Perhitungan= ");
27         tampilkanPerhitungan(f);
28
29         System.out.println();
30
31         hasil = penjumlahan(f);
32         System.out.println("Penjumlahan dari 1 sampai " + f + " adalah= " + hasil);
33     }
34 }
35

```

2.

```

1  import java.util.Scanner;
2  public class TugasNo3 {
3
4      public static int hitungPasanganMarmut(int bulan) {
5          if (bulan <= 2) {
6              return 1;
7          } else {
8              return hitungPasanganMarmut(bulan - 1) + hitungPasanganMarmut(bulan - 2);
9          }
10     }
11
12     public static int hitungPasanganProduktif(int bulan) {
13         if (bulan <= 2) {
14             return 0;
15         } else {
16             return hitungPasanganMarmut(bulan - 2);
17         }
18     }
19
20     public static void main(String[] args) {
21         Scanner sc = new Scanner(System.in);
22         int bulan, pasanganProduktif, pasanganBelumProduktif, totalPasangan;
23
24         System.out.print(s:"Masukkan bulan ke- (1-12): ");
25         bulan = sc.nextInt();
26
27         totalPasangan = hitungPasanganMarmut(bulan);
28         pasanganProduktif = hitungPasanganProduktif(bulan);
29         pasanganBelumProduktif = totalPasangan - pasanganProduktif;
30
31         System.out.println("Bulan ke-" + bulan + ":");
32         System.out.println("Jumlah Pasangan Produktif: " + pasanganProduktif);
33         System.out.println("Jumlah Pasangan Belum Produktif: " + pasanganBelumProduktif);
34         System.out.println("Total Pasangan Marmut: " + totalPasangan);
35     }
36 }

```

3.