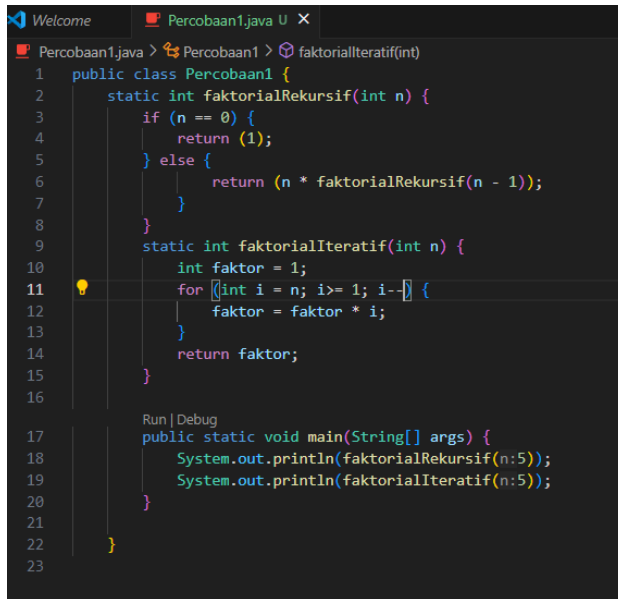


Nama : Bima Adiwijaya

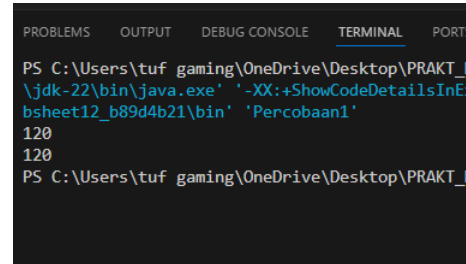
NIM : 244107020022

Kelas : TI-1D

1. Percobaan 1



```
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     }
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    }
16
17    public static void main(String[] args) {
18        System.out.println(faktorialRekursif(5));
19        System.out.println(faktorialIteratif(5));
20    }
21
22 }
23
```

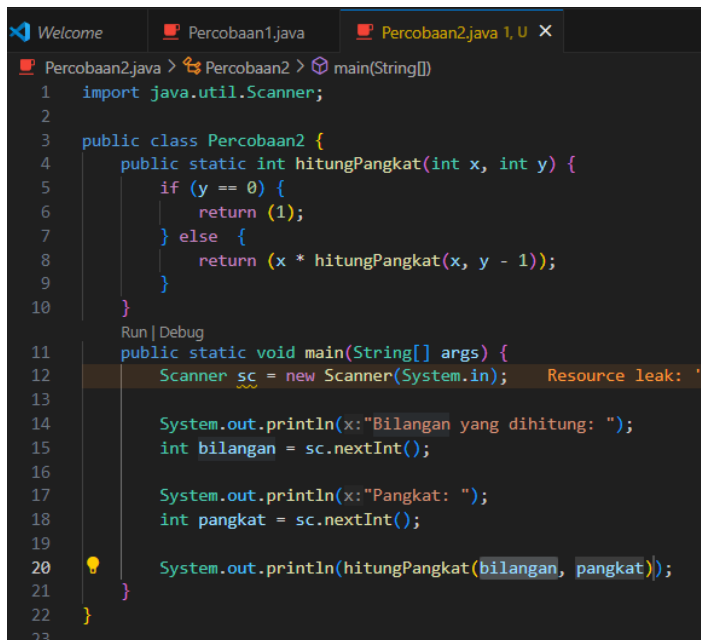


```
PS C:\Users\tuf gaming\OneDrive\Desktop\PRAKT_I
\jdk-22\bin\java.exe' '-XX:+ShowCodeDetailsInE
bsheet12_b89d4b21\bin' 'Percobaan1'
120
120
PS C:\Users\tuf gaming\OneDrive\Desktop\PRAKT_I
```

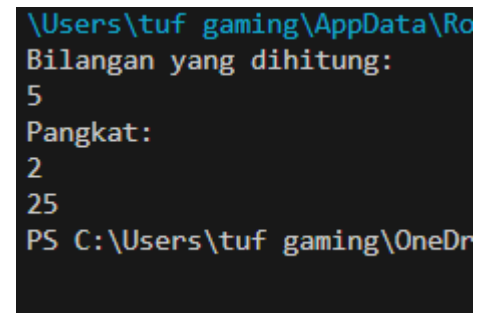
Pertanyaan percobaan 1

1. Apa yang dimaksud dengan fungsi rekursif?
Jawab: fungsi rekursif adalah fungsi yang memanggil dirinya sendiri selama proses eksekusi untuk menyelesaikan masalah dengan membaginya menjadi submasalah yang lebih kecil dan serupa
2. Bagaimana contoh kasus penggunaan fungsi rekursif?
Jawab: digunakan untuk menghitung factorial
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: Hasilnya sama. Alur Rekursif memanggil dirinya sendiri dan menggunakan stack. Alur Iteratif menggunakan loop untuk menghitung factorial secara langsung.

2. Percobaan kedua



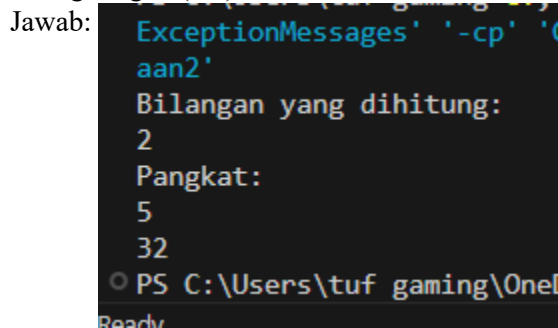
```
1 import java.util.Scanner;
2
3 public class Percobaan2 {
4     public static int hitungPangkat(int x, int y) {
5         if (y == 0) {
6             return (1);
7         } else {
8             return (x * hitungPangkat(x, y - 1));
9         }
10    }
11
12    public static void main(String[] args) {
13        Scanner sc = new Scanner(System.in);
14
15        System.out.println(x: "Bilangan yang dihitung: ");
16        int bilangan = sc.nextInt();
17
18        System.out.println(x: "Pangkat: ");
19        int pangkat = sc.nextInt();
20
21        System.out.println(hitungPangkat(bilangan, pangkat));
22    }
23 }
```



```
\Users\tuf gaming\AppData\Ro
Bilangan yang dihitung:
5
Pangkat:
2
25
PS C:\Users\tuf gaming\OneDr
```

Pertanyaan Percobaan Kedua

1. Pada Percobaan2, terdapat pemanggilan fungsi rekursif hitungPangkat(bilangan, pangkat) pada fungsi main, kemudian dilakukan pemanggilan fungsi hitungPangkat() secara berulang kali. Jelaskan sampai kapan proses pemanggilan fungsi tersebut akan dijalankan!
Jawab: Proses pemanggilan fungsi rekursif hitungPangkat(bilangan, pangkat) akan dijalankan secara berulang hingga kondisi basis terpenuhi.
2. Tambahkan kode program untuk mencetak deret perhitungan pangkatnya. Contoh : hitungPangkat(2,5) dicetak $2 \times 2 \times 2 \times 2 \times 2 \times 1 = 32$



```
ExceptionMessages' '-cp' '0
aan2'
Bilangan yang dihitung:
2
Pangkat:
5
32
PS C:\Users\tuf gaming\OneD
Ready
```

3. Percobaan Ketiga

```
Percobaan3.java > Percobaan3
1  import java.util.Scanner;
2
3  public class Percobaan3 {
4      public 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     }
11
12     Run | Debug
13     public static void main(String[] args) {
14         Scanner sc = new Scanner(System.in);
15         System.out.println(x:"Jumlah saldo awal: ");
16         double saldoAwal = sc.nextDouble();
17
18         System.out.println(x:"Lamanya investasi (tahun): ");
19         int tahun = sc.nextInt();
20
21         System.out.println("Jumlah saldo setelah " + tahun + " tahun: ");
22         System.out.println(hitungLaba(saldoAwal, tahun));
23     }
24 }
```

```
\f78578d179a20d5faba7dbd09cc577ee\
Jumlah saldo awal:
50000
Lamanya investasi (tahun):
5
Jumlah saldo setelah 5 tahun:
84252.90775500004
PS C:\Users\tuf gaming\OneDrive\De
```

Pertanyaan percobaan ketiga

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

Jawab: Base case = `if (tahun == 0) { return saldo; }`

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

2. Jabarkan trace fase ekspansi dan fase substitusi algoritma perhitungan laba di atas jika diberikan nilai `hitungLaba(100000,3)`

Jawab:

Fase ekspansi:

- Fungsi dipanggil dengan `hitungLaba(100000, 3)`:
Masuk ke else, panggil `hitungLaba(100000, 2)`.
- Fungsi dipanggil dengan `hitungLaba(100000, 2)`:
Masuk ke else, panggil `hitungLaba(100000, 1)`.
- Fungsi dipanggil dengan `hitungLaba(100000, 1)`:
Masuk ke else, panggil `hitungLaba(100000, 0)`.
- Fungsi dipanggil dengan `hitungLaba(100000, 0)`:
Masuk ke **base case**, kembalikan saldo = 100000.

Fase Substitusi:

- Dari `hitungLaba(100000, 0)`:
Kembalikan **100000** ke pemanggil sebelumnya (`hitungLaba(100000, 1)`).
- Dari `hitungLaba(100000, 1)`:
 $\text{Hitung } 1.11 \times 100000 = 111000$
Kembalikan **111000** ke pemanggil sebelumnya (`hitungLaba(100000, 2)`).
- Dari `hitungLaba(100000, 2)`:
 $\text{Hitung } 1.11 \times 111000 = 123210$
Kembalikan **123210** ke pemanggil sebelumnya (`hitungLaba(100000, 3)`).
- Dari `hitungLaba(100000, 3)`:
 $\text{Hitung } 1.11 \times 123210 = 136764.1$
Kembalikan **136764.1** sebagai hasil akhir.

Tugas

1.

```
Tugas1.java > Tugas1 > DeretDescendingRekursif(int)
1  import java.util.Scanner;
2
3  public class Tugas1 {
4      public static int DeretDescendingRekursif(int n) {
5          if (n == 0) {
6              return 0;
7          } else {
8              System.out.println(n + " ");
9              return DeretDescendingRekursif(n - 1);
10         }
11     }
12
13     Run | Debug
14     public static void main(String[] args) {
15         Scanner input = new Scanner(System.in);
16
17         System.out.println(x:"Masukkan angka: ");
18         int angka = input.nextInt();
19
20         System.out.println(DeretDescendingRekursif(angka));
21     }
22 }
```

```
7
6
5
4
3
2
1
0
```

2.

```
Tugas2.java > Tugas2
1  import java.util.Scanner;
2
3  public class Tugas2 {
4      public static int PenjumlahanRekursif(int n) {
5          if (n == 0) {
6              return 0;
7          } else {
8              return n + PenjumlahanRekursif(n - 1);
9          }
10     }
11
12     Run | Debug
13     public static void main(String[] args) {
14         Scanner input = new Scanner(System.in);
15
16         System.out.println(x:"Masukkan angka: ");
17         int angka = input.nextInt();
18
19         System.out.println(PenjumlahanRekursif(angka));
20     }
21 }
```

```
Masukkan angka:
8
36
```

3.

```
Tugas3.java > Tugas3 > main(String[])
1  public class Tugas3 {
2      public static int Fibonacci(int bulan) {
3          if (bulan == 1 || bulan == 2) {
4              return 1;
5          } else {
6              return Fibonacci(bulan - 1) + Fibonacci(bulan - 2);
7          }
8      }
9
10     Run | Debug
11     public static void main(String[] args) {
12         int bulan = 12;
13         int pasanganMarmut = Fibonacci(bulan);
14         System.out.println("Jumlah pasangan marmut pada bulan ke-" + bulan + " adalah: " + pasanganMarmut);
15     }
16 }
```

144

```
PS C:\Users\tuf gaming c:; cd 'c:\Users\tuf gaming\OneDrive\Desktop\PRAKT_DASPRO'
ExceptionMessages' -cp' 'C:\Users\tuf gaming\AppData\Local\Temp\
Jumlah pasangan marmut pada bulan ke-12 adalah: 144
PS C:\Users\tuf gaming\OneDrive\Desktop\PRAKT_DASPRO
```