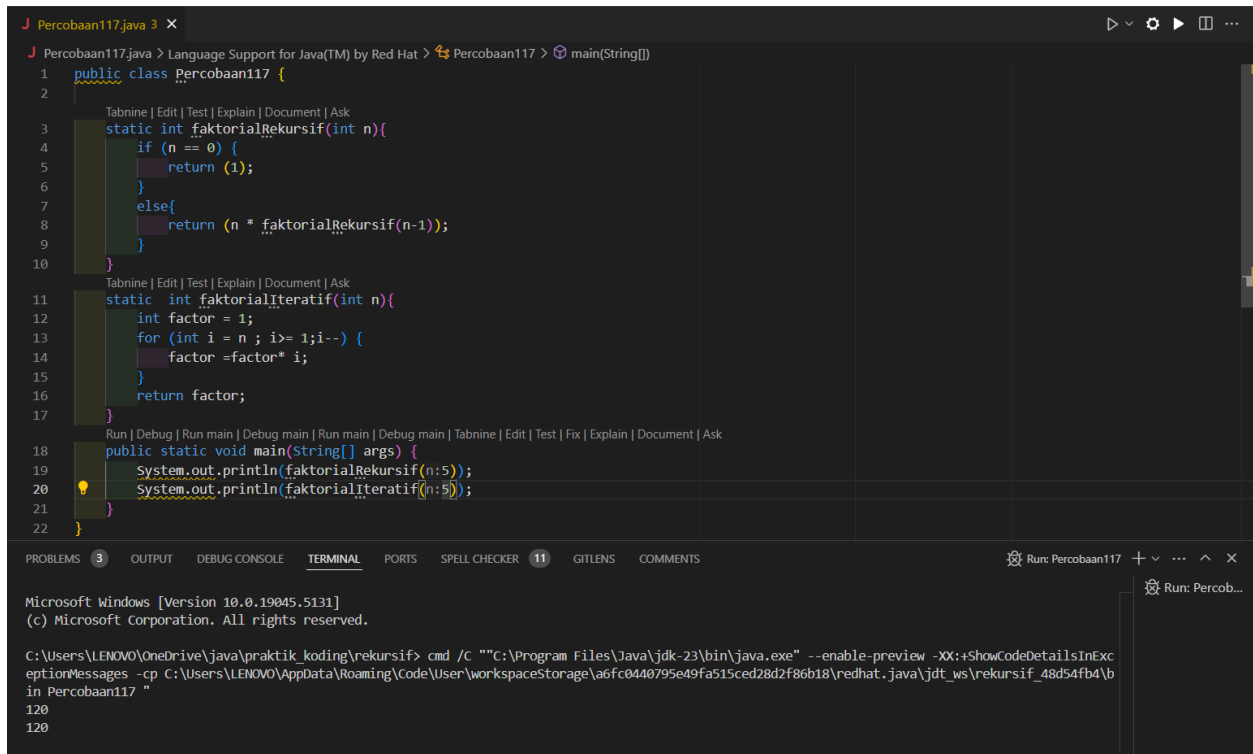


Nama : Muhammad Hasbi Ashiddiqi

Kelas : 1D

Absen : 17

Percobaan 1



```
J Percobaan117.java 3 x
Percobaan117.java > Language Support for Java(TM) by Red Hat > Percobaan117 > main(String[])
1 public class Percobaan117 {
2
3     Tabnine | Edit | Test | Explain | Document | Ask
4     static int faktorialRekursif(int n){
5         if (n == 0) {
6             return 1;
7         }
8         else{
9             return (n * faktorialRekursif(n-1));
10        }
11    }
12
13    Tabnine | Edit | Test | Explain | Document | Ask
14    static int faktorialIteratif(int n){
15        int factor = 1;
16        for (int i = n ; i>= 1;i--) {
17            factor =factor* i;
18        }
19        return factor;
20    }
21
22    Run | Debug | Run main | Debug main | Run main | Debug main | Tabnine | Edit | Test | Fix | Explain | Document | Ask
23    public static void main(String[] args) {
24        System.out.println(faktorialRekursif(5));
25        System.out.println(faktorialIteratif(5));
26    }
27 }
```

PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL PORTS SPELL CHECKER 11 GITLENS COMMENTS

Microsoft Windows [Version 10.0.19045.5131]
(c) Microsoft Corporation. All rights reserved.

C:\Users\LENOVO\OneDrive\java\praktik_koding\rekursif> cmd /c ""C:\Program Files\Java\jdk-23\bin\java.exe" --enable-preview -XX:+ShowCodeDetailsInExceptionMessages -cp C:\Users\LENOVO\AppData\Roaming\Code\User\workspaceStorage\af6c0440795e49fa515ced28d2f86b18\redhat.java\jdt_ws\rekursif_48d54fb4\b in Percobaan117 ""

120

120

Jawaban

1. fungsi rekursif adalah suatu bentuk perulangan yang tidak melibatkan iterasi
2. encarian Elemen dalam Struktur Data
3. Hasilnya sama karena kedua metode benar secara logik.

Perbedaannya terletak pada cara kerjanya:

Rekursif: Cocok untuk masalah yang dapat dipecah menjadi sub-masalah kecil, tetapi bisa lebih lambat dan rentan terhadap stack overflow jika nilai n terlalu besar.

Iteratif: Lebih efisien dalam penggunaan memori dan cocok untuk penghitungan besar tanpa risiko stack overflow.

Percobaan 2

```
percobaan217.java > Language Server for Java by Apache NetBeans > percobaan217 > main(String[] args)
1
2 import java.util.Scanner;
3
4 public class percobaan217 {
5     static int hitungPangkat(int x, int y) {
6         if (y == 0) {
7             return 1;
8         } else {
9             return (x * hitungPangkat(x, y - 1));
10        }
11    }
12    public static void main(String[] args) {
13        Scanner sc = new Scanner (System.in);
14        int bilangan , pangkat;
15
16        System.out.print(s:"Bilangan yang dihitung: ");
17        bilangan = sc.nextInt();
18
19        System.out.print(s:"pangkat: ");
20        pangkat = sc.nextInt();
21
22        System.out.println("Hasil: " + hitungPangkat(bilangan, pangkat));
23    }
24 }
25 }
```

PROBLEMS 10 OUTPUT DEBUG CONSOLE TERMINAL PORTS SPELL CHECKER 28 GITLENS COMMENTS

```
ers\LENOVO\OneDrive\java\praktik_koding\rekursif> cmd /c ""C:\Program Files\Java\jdk-23\bin\java.exe" --enable-preview -XX:+ShowCodeDetailsInExceptionMessages -cp C:\Users\LENOVO\AppData\Roaming\Code\User\workspaceStorage\a6fc0440795e49fa515ced28d2f86b18\redhat.java\jdt_ws\rekursif_48d54fb4\bin pe
rcobaan217 "
Bilangan yang dihitung: 2
Pangkat: 1
Hasil: 2
C:\Users\LENOVO\OneDrive\java\praktik_koding\rekursif>
```

Jawaban

1. Pemanggilan fungsi rekursif akan berhenti setelah nilai pangkat (y) mencapai 0, sesuai dengan basis kasus.

```

import java.util.Scanner;

public class percobaan217 {
    Tabnine | Edit | Test | Fix | Explain | Document | Ask
    static int hitungPangkat(int x, int y) {
        if (y == 0) {
            System.out.print(s:"1");
            return 1;
        } else {
            System.out.print(x + "x");
            return (x * hitungPangkat(x, y - 1));
        }
    }

    Run | Debug | Run main | Debug main | Run main | Debug main | Tabnine | Edit | Test | Fix | Explain | Document | Ask
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int bilangan, pangkat;

        System.out.print(s:"Bilangan yang dihitung: ");
        bilangan = sc.nextInt();

        System.out.print(s:"Pangkat: ");
        pangkat = sc.nextInt();

        System.out.print(s:"Deret perhitungan: ");
        int hasil = hitungPangkat(bilangan, pangkat);
        System.out.println(" = " + hasil);
    }
}

```

2.

Percobaan 3

```

1
2 import java.util.Scanner;
3
4 public class Percobaan317 {
5     Tabnine | Edit | Test | Explain | Document | Ask
6     static double hitungLaba(double saldo, int tahun) {
7         if (tahun == 0) {
8             return saldo;
9         } else {
10             return 1.11 * hitungLaba(saldo, tahun - 1);
11         }
12     }
13     Run | Debug | Run main | Debug main | Run main | Debug main | Tabnine | Edit | Test | Fix | Explain | Document | Ask
14     public static void main(String[] args) {
15         Scanner sc = new Scanner(System.in);
16         int tahun;
17         double saldoAwal;
18         System.out.print(s:"Jumlah saldo awal: ");
19         saldoAwal = sc.nextDouble();
20         System.out.print(s:"Lamanya investasi (tahun): ");
21         tahun = sc.nextInt();
22         System.out.print("Jumlah saldo setelah " + tahun + " tahun: ");
23         System.out.print(hitungLaba(saldoAwal, tahun));
24     }
25 }

```

PROBLEMS 19 OUTPUT DEBUG CONSOLE TERMINAL PORTS SPELL CHECKER 64 GITLENS COMMENTS

(c) Microsoft Corporation. All rights reserved.

```

C:\Users\LENOVO\OneDrive\java\praktik koding\rekursif> cmd /c ""C:\Program Files\Java\jdk-23\bin\java.exe" --enable-preview -XX:+ShowCodeDetailsInExceptionMessages -cp C:\Users\LENOVO\AppData\Roaming\Code\User\workspaceStorage\af6c0440795e49fa515ced28d2f86b18\redhat.java\jdt_ws\rekursif_48d54fb4\b
in Percobaan317 "
Jumlah saldo awal: 200
Lamanya investasi (tahun): 10
Jumlah saldo setelah 10 tahun: 567.8841972138036

```

Jawaban

1. base case :

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

recursion call:

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

2. fase ekspansi

Pemanggilan awal: hitungLaba(100000, 3)

-Karena tahun != 0, rekursi dipanggil:

1.11 * hitungLaba(100000, 2)

Pemanggilan kedua: hitungLaba(100000, 2)

-Karena tahun != 0, rekursi dipanggil lagi:

1.11 * hitungLaba(100000, 1)

Pemanggilan ketiga: hitungLaba(100000, 1)

-Karena tahun != 0, rekursi dipanggil lagi:

1.11 * hitungLaba(100000, 0)

Pemanggilan keempat: hitungLaba(100000, 0)

-Karena tahun == 0, base case tercapai.

Fungsi mengembalikan nilai 100000.

Fase substitusi

-Hasil dari hitungLaba(100000, 0) adalah 100000.

Substitusi ke pemanggilan sebelumnya:

hitungLaba(100000, 1) = 1.11 * 100000 = 111000

-Hasil dari hitungLaba(100000, 1) adalah 111000.

Substitusi ke pemanggilan sebelumnya:

hitungLaba(100000, 2) = 1.11 * 111000 = 123210

-Hasil dari hitungLaba(100000, 2) adalah 123210.

Substitusi ke pemanggilan sebelumnya:

hitungLaba(100000, 3) = 1.11 * 123210 = 136761

Tugas

1.

```
public class tugasNo117 {
    Tabnine | Edit | Test | Fix | Explain | Document | Ask
    public static void deretDescendingRekursif(int n) {
        if (n < 0) {
            return;
        }
        System.out.print(n + " ");
        deretDescendingRekursif(n - 1);
    }

    Tabnine | Edit | Test | Fix | Explain | Document | Ask
    public static void deretDescendingIteratif(int n) {
        for (int i = n; i >= 0; i--) {
            System.out.print(i + " ");
        }
    }

    Run | Debug | Run main | Debug main | Run main | Debug main | Tabnine | Edit | Test | Fix | Explain | Document | Ask
    public static void main(String[] args) {
        int n = 10;
        System.out.println(x:"Deret dengan rekursif:");
        deretDescendingRekursif(n);
        System.out.println();
        System.out.println(x:"Deret dengan iteratif:");
        deretDescendingIteratif(n);
    }
}
```

2.

```
TugasNo217.java > Language Support for Java(TM) by Red Hat > TugasNo217 > main(String[])
1 public class TugasNo217 {
    Tabnine | Edit | Test | Explain | Document | Ask
2     public static int hitungPenjumlahan(int n) {
3         if (n == 1) {
4             return 1;
5         } else {
6             return n + hitungPenjumlahan(n - 1);
7         }
8     }

    Run | Debug | Run main | Debug main | Run main | Debug main | Tabnine | Edit | Test | Explain | Document | Ask
9     public static void main(String[] args) {
10         int f = 8;
11         int hasil = hitungPenjumlahan(f);
12         System.out.println("Penjumlahan dari 1 hingga " + f + " adalah " + hasil);
13     }
14 }
15 }
```

3.

```
1 public class TugasNo317 {  
    Tabnine | Edit | Test | Explain | Document | Ask  
2     public static int hitungPasangan(int bulan) {  
3         if (bulan == 1 || bulan == 2) {  
4             return 1;  
5         } else {  
6             return hitungPasangan(bulan - 1) + hitungPasangan(bulan - 2);  
7         }  
8     }  
9  
    Run main | Debug main | Run main | Debug main | Tabnine | Edit | Test | Explain | Document | Ask  
10    public static void main(String[] args) {  
11        int bulanKe = 12;  
12        int totalPasangan = hitungPasangan(bulanKe);  
13        System.out.println("Jumlah total pasangan marmut pada bulan ke-" + bulanKe + " adalah: " + totalPasangan);  
14    }  
15 }
```