

Jawaban Jobsheet 4

Satria Wiguna / Ti_1D / Absen 26

Percobaan 1:

Class:

```
1  public class Faktorial {
2      public int nilai;
3
4      int faktorialBF(int n){
5          int fakto = 1;
6          for (int i=1; i<=n;i++){
7              fakto = fakto * i ;
8          }
9          return fakto;
10     }
11
12     int faktorialDC(int n){
13         if (n==1){
14             return 1;
15         }
16         else{
17             int fakto = n*faktorialDC(n-1);
18             return fakto;
19         }
20     }
21 }
22
```

Main :

```
2  public class BruteForceDivideConquer {
3      public static void main(String[] args) {
4
5          Faktorial [] fk = new Faktorial [ijml];
6          for (int i = 0; i <ijml ;i++){
7              fk [i] = new Faktorial ();
8              System.out.println("Masukkan Nilai data ke - "+ (i+1) + ":");
9              int nilai = sc.nextInt();
10             fk[i].nilai = nilai;
11         }
12         System.out.println(x:"HASIL - BRUTE FORCE");
13         for(int i=0 ; i<ijml;i++){
14             System.out.println
15             ("Hasil Perhitungan Faktorial Menggunakan Brute Force Adalah " + fk[i].faktorialBF(fk[i].nilai));
16         }
17         System.out.println(x:"HASIL - DIVIDE AND ConQUER");
18         for(int i=0 ; i<ijml;i++){
19             System.out.println
20             ("Hasil Perhitungan Faktorial Menggunakan Brute Force Adalah " + fk[i].faktorialDC(fk[i].nilai));
21         }
22         sc.close();
23     }
24 }
25
26
27
28
29
30
```

Pertanyaan

1. Pada base line Algoritma Divide Conquer untuk melakukan pencarian nilai faktorial, jelaskan perbedaan bagian kode pada penggunaan if dan else!
perbedaan antara bagian if dan else terletak pada pemisahan antara kasus basis dan langkah rekursif. Jika nilai input adalah 1, maka fungsi langsung mengembalikan nilai 1. Jika tidak, fungsi akan melakukan langkah rekursif dengan mengalikan nilai input dengan hasil faktorial dari nilai input dikurangi 1.
2. Apakah memungkinkan perulangan pada method `faktorialBF()` dirubah selain menggunakan for?Buktikan!

Bisa

```
int faktorialBF(int n) {
    if (n == 0 || n == 1) {
        return 1;
    } else {
        return n * faktorialBF(n - 1);
    }
}
```

3. Jelaskan perbedaan antara **fakto *= i;** dan **int fakto = n * faktorialDC(n-1);** !
 - `fakto *= i;` adalah bentuk singkat dari `fakto = fakto * i`
 - `faktorialDC(n-1);` merupakan implementasi rekursif dari perhitungan faktorial yang memanfaatkan konsep pemanggilan fungsi diri sendiri dengan parameter yang semakin kecil hingga mencapai kondisi basis

Percobaan 2:

Main:

```
1  import java.util.Scanner;
2
3  public class MainPangkat {
4      public static void main(String[] args) {
5          Scanner sc = new Scanner(System.in);
6          System.out.println("=====");
7          System.out.print("Masukkan Jumlah Elemen Yang Dihitung : ");
8          int elemen = sc.nextInt();
9
10         Pangkat[] png = new Pangkat[elemen];
11         for (int i = 0; i < elemen; i++) {
12             png[i] = new Pangkat();
13             System.out.println("Masukkan Nilai Yang Hendak Dipangkatkan: ");
14             int nilai = sc.nextInt();
15             png[i].nilai = nilai;
16             System.out.println("Masukkan Nilai Pemangkat: ");
17             int pangkat = sc.nextInt();
18             png[i].pangkat = pangkat;
19         }
20         System.out.println("HASIL PANGKAT - BRUTE FORCE");
21         for (int i=0; i<elemen; i++) {
22             System.out.println("Hasil Dari " + png[i].nilai + " pangkat " + png[i].pangkat + " adalah " + png[i].PangkatBF(png[i].nilai,png[i].pangkat));
23         }
24         System.out.println("HASIL PANGKAT - DIVIDE AND CONQUER");
25         for (int i=0; i<elemen; i++) {
26             System.out.println("Hasil Dari " + png[i].nilai + " pangkat " + png[i].pangkat + " adalah " + png[i].PangkatDC(png[i].nilai,png[i].pangkat));
27         }
28     }
29 }
```

Class:

```
1 public class Pangkat {
2     public int nilai,pangkat;
3     int PangkatBF(int a, int n){
4         int hasil = 1;
5         for ( int i= 1 ; i <= n ; i++){
6             hasil *= a;
7         }
8         return hasil;
9     }
10    int PangkatDC(int a,int n){
11        if (n==0){
12            return 1;
13        } else{
14            if(n%2==1)
15            {
16                return (PangkatDC(a,n/2)*PangkatDC(a,n/2)*a);
17            }else{
18                return (PangkatDC(a,n/2)*PangkatDC(a,n/2));
19            }
20        }
21    }
22 }
23
```

Pertanyaan

1. Jelaskan mengenai perbedaan 2 method yang dibuat yaitu `PangkatBF()` dan `PangkatDC()` !
 - `PangkatBF()` menggunakan pendekatan brute force, di mana nilai pangkat langsung digunakan untuk melakukan iterasi dengan mengalikan nilai `a` sebanyak `n` kali.
 - `PangkatDC()` menggunakan pendekatan divide and conquer, di mana perhitungan pangkat dibagi menjadi dua bagian dan hasilnya dikombinasikan kembali. Pendekatan ini menggunakan rekursi untuk membagi permasalahan menjadi submasalah yang lebih kecil.
2. Apakah tahap *combine* sudah termasuk dalam kode tersebut? Tunjukkan!

Sudah

```
int PangkatDC(int a,int n){
    if (n==0){
        return 1;
    } else{
        if(n%2==1)
        {
            return (PangkatDC(a,n/2)*PangkatDC(a,n/2)*a);
        }else{
            return (PangkatDC(a,n/2)*PangkatDC(a,n/2));
        }
    }
}
```

3. Modifikasi kode program tersebut, anggap proses pengisian atribut dilakukan dengan konstruktor.

```
public class Pangkat {
    public int nilai, pangkat;
    public Pangkat(int nilai, int pangkat){
        this.nilai = nilai;
        this.pangkat = pangkat;
    }
}
```

4. Tambahkan menu agar salah satu method yang terpilih saja yang akan dijalankan menggunakan switch-case!

```
1  import java.util.Scanner;
2
3  public class MainPangkat {
4      public static void main(String[] args) {
5          Scanner sc = new Scanner(System.in);
6          System.out.println("=====");
7          System.out.print("Masukkan Jumlah Elemen Yang Dihitung : ");
8          int elemen = sc.nextInt();
9
10         Pangkat[] png = new Pangkat[elemen];
11         for (int i = 0; i < elemen; i++) {
12             System.out.println("Masukkan Nilai Yang Hendak Dipangkatkan: ");
13             int nilai = sc.nextInt();
14             System.out.println("Masukkan Nilai Pemangkat: ");
15             int pangkat = sc.nextInt();
16             png[i] = new Pangkat(nilai, pangkat);
17         }
18
19         System.out.println("Pilih metode perhitungan pangkat:");
20         System.out.println("1. Brute Force");
21         System.out.println("2. Divide and Conquer");
22         int choice = sc.nextInt();
23
24         switch (choice) {
25             case 1:
26                 System.out.println("HASIL PANGKAT - BRUTE FORCE");
27                 for (int i = 0; i < elemen; i++) {
28                     System.out.println("Hasil Dari " + png[i].nilai + " pangkat " + png[i].pangkat + " adalah " + png[i].PangkatBF(png[i].nilai, png[i].pangkat));
29                 }
30                 break;
31             case 2:
32                 System.out.println("HASIL PANGKAT - DIVIDE AND CONQUER");
33                 for (int i = 0; i < elemen; i++) {
34                     System.out.println("Hasil Dari " + png[i].nilai + " pangkat " + png[i].pangkat + " adalah " + png[i].PangkatDC(png[i].nilai, png[i].pangkat));
35                 }
36                 break;
37             default:
38                 System.out.println("Pilihan tidak valid");
39                 break;
40         }
41         sc.close();
42     }
43 }
```

Percobaan 3 :

```
1  import java.util.Scanner;
2
3  public class MainSum {
4      Run | Debug
5      public static void main(String[] args) {
6          Scanner sc = new Scanner(System.in);
7          System.out.println(x:"=====");
8          System.out.println(x:"Program Menghitung Keuntungan Total (Satuan Juta. Misal 5.9)");
9          System.out.print(s:"Masukkan jumlah bulan : ");
10         int elm = sc.nextInt();
11
12         Sum sm = new Sum(elm);
13         System.out.println(x:"=====");
14         for (int i = 0; i < sm.elemen; i++) {
15             System.out.print("Masukkan untung bulan ke-" + (i+1) + " = ");
16             sm.keuntungan[i] = sc.nextDouble();
17         }
18
19         System.out.println(x:"=====");
20         System.out.println(x:"Algoritma Brute Force");
21         System.out.println("Total keuntungan perusahaan selama " + sm.elemen + " bulan adalah = " + sm.totalBF(sm.keuntungan));
22         System.out.println(x:"=====");
23         System.out.println(x:"Algoritma Divide and Conquer");
24         System.out.println("Total keuntungan perusahaan selama " + sm.elemen + " bulan adalah = " + sm.totalDC(sm.keuntungan, 1:0, sm.elemen-1));
25     }
26 }
```

```
1  public class Sum {
2      public int elemen;
3      public double keuntungan[], total;
4
5      Sum (int elemen){
6          this.elemen = elemen;
7          this.keuntungan = new double[elemen];
8          this.total = 0;
9      }
10
11      double totalBF(double arr[]){
12          for(int i = 0; i < elemen; i++) {
13              total = total + arr[i];
14          }
15          return total;
16      }
17
18      double totalDC(double arr[], int l, int r) {
19          if(l == r) {
20              return arr[l];
21          } else if(l < r) {
22              int mid = (l + r) / 2;
23              double lsum = totalDC(arr, l, mid-1);
24              double rsum = totalDC(arr, mid + 1, r);
25              return lsum + rsum + arr[mid];
26          }
27          return 0;
28      }
29  }
30 }
```

Pertanyaan :

1. Mengapa terdapat formulasi *return value* berikut?Jelaskan!

```
return lsum+rsum+arr[mid];
```

Pada baris `return lsum + rsum + arr[mid];`, `lsum` adalah hasil dari rekursi pada subarray sebelah kiri, `rsum` adalah hasil dari rekursi pada subarray sebelah kanan, dan `arr[mid]` adalah elemen tengah dari subarray saat ini.

2. Kenapa dibutuhkan variable `mid` pada method `TotalDC()` ?
Variable `mid` dibutuhkan dalam metode `totalDC()` karena metode ini menggunakan pendekatan Divide and Conquer untuk menghitung total dari array.
3. Program perhitungan keuntungan suatu perusahaan ini hanya untuk satu perusahaan saja. Bagaimana cara menghitung sekaligus keuntungan beberapa bulan untuk beberapa perusahaan.(Setiap perusahaan bisa saja memiliki jumlah bulan berbeda-beda)? Buktikan dengan program!

```
1  import java.util.Scanner;
2
3  public class MainSum {
4      public static void main(String[] args) {
5          Scanner sc = new Scanner(System.in);
6          System.out.println("=====");
7          System.out.println("Program Menghitung Keuntungan Total (Satuan Juta. Misal 5.9)");
8          System.out.print("Masukkan jumlah perusahaan: ");
9          int numCompanies = sc.nextInt();
10         System.out.print("Masukkan jumlah bulan: ");
11         int numMonths = sc.nextInt();
12
13         Sum[] companies = new Sum[numCompanies];
14
15
16         for (int i = 0; i < numCompanies; i++) {
17             System.out.println("=====");
18             System.out.println("Masukkan data keuntungan untuk Perusahaan " + (i + 1));
19             companies[i] = new Sum(numMonths);
20             for (int j = 0; j < numMonths; j++) {
21                 System.out.print("Masukkan untung bulan ke-" + (j + 1) + " = ");
22                 companies[i].keuntungan[j] = sc.nextDouble();
23             }
24         }
25
26
27         System.out.println("=====");
28         System.out.println("Algoritma Brute Force");
29         for (int i = 0; i < numCompanies; i++) {
30             System.out.println("Total keuntungan perusahaan " + (i + 1) + " selama " + numMonths + " bulan adalah = " +
31                 companies[i].totalBF(companies[i].keuntungan));
32         }
33
34         System.out.println("=====");
35         System.out.println("Algoritma Divide and Conquer");
36         for (int i = 0; i < numCompanies; i++) {
37             System.out.println("Total keuntungan perusahaan " + (i + 1) + " selama " + numMonths + " bulan adalah = " +
38                 companies[i].totalDC(companies[i].keuntungan, 0, numMonths - 1));
39         }
40     }
41 }
```

```

=====
Program Menghitung Keuntungan Total (Satuan Juta. Misal 5.9)
Masukkan jumlah perusahaan: 2
Masukkan jumlah bulan: 4
=====

```

```

Masukkan data keuntungan untuk Perusahaan 1

```

```

Masukkan untung bulan ke-1 = 34

```

```

Masukkan untung bulan ke-2 = 23

```

```

Masukkan untung bulan ke-3 = 22

```

```

Masukkan untung bulan ke-4 = 34
=====

```

```

Masukkan data keuntungan untuk Perusahaan 2

```

```

Masukkan untung bulan ke-1 = 123

```

```

Masukkan untung bulan ke-2 = 34

```

```

Masukkan untung bulan ke-3 = 345

```

```

Masukkan untung bulan ke-4 = 234
=====

```

```

Algoritma Brute Force

```

```

Total keuntungan perusahaan 1 selama 4 bulan adalah = 113.0

```

```

Total keuntungan perusahaan 2 selama 4 bulan adalah = 736.0
=====

```

```

Algoritma Divide and Conquer

```

```

Total keuntungan perusahaan 1 selama 4 bulan adalah = 113.0

```

```

Total keuntungan perusahaan 2 selama 4 bulan adalah = 736.0

```

4.5 Latihan Praktikum

1. Sebuah showroom memiliki daftar mobil dengan data sesuai tabel di bawah ini

merk	tipe	tahun	top_acceleration	top_power
BMW	M2 Coupe	2016	6816	728
Ford	Fiesta ST	2014	3921	575
Nissan	370Z	2009	4360	657
Subaru	BRZ	2014	4058	609
Subaru	Impreza WRX STI	2013	6255	703
Toyota	AE86 Trueno	1986	3700	553
Toyota	86/GT86	2014	4180	609
Volkswagen	Golf GTI	2014	4180	631

Tentukan:

- top_acceleration tertinggi menggunakan Divide and Conquer!
 - top_acceleration terendah menggunakan Divide and Conquer!
 - Rata-rata top_power dari seluruh mobil menggunakan Brute Force!
- Mobil Class :

```

public class mobil26 {
    String merk;
    String tipe;
    int tahun;
    int topAcceleration;
    int topPower;

    public mobil26(String merk, String tipe, int tahun, int topAcceleration, int topPower) {
        this.merk = merk;
        this.tipe = tipe;
        this.tahun = tahun;
        this.topAcceleration = topAcceleration;
        this.topPower = topPower;
    }

    public String toString(){
        return "Merk: " + merk + ", Tipe: " + tipe + ", Tahun: " + tahun + ", Top Acceleration: " + topAcceleration + ", Top Power: " + topPower;
    }
}

```

Mobil Main

```

1  public class mobilmain26 {
2      public static int maxAcceleration(mobil26[] car, int left, int right) {
3          if (left == right)
4              return car[left].topAcceleration;
5
6          int mid = (left + right) / 2;
7
8          int maxLeft = maxAcceleration(car, left, mid);
9          int maxRight = maxAcceleration(car, mid + 1, right);
10
11         return Math.max(maxLeft, maxRight);
12     }
13
14     public static int minAcceleration(mobil26[] cars, int left, int right) {
15         if (left == right)
16             return cars[left].topAcceleration;
17
18         int mid = (left + right) / 2;
19
20         int minLeft = minAcceleration(cars, left, mid);
21         int minRight = minAcceleration(cars, mid + 1, right);
22
23         return Math.min(minLeft, minRight);
24     }
25
26     public static double averageTopPower(mobil26[] cars) {
27         int sum = 0;
28         for (mobil26 car : cars) {

```



```

29         sum += car.topPower;
30     }
31     return (double) sum / cars.length;
32 }
33
34
35 Run | Debug
36 public static void main(String[] args) {
37     mobil26[] cars = {
38         new mobil26(merk:"BMW", tipe:"M2 Coupe", tahun:2016, topAcceleration:6816, topPower:728),
39         new mobil26(merk:"Ford", tipe:"Fiesta ST", tahun:2014, topAcceleration:3921, topPower:575),
40         new mobil26(merk:"Nissan", tipe:"370Z", tahun:2009, topAcceleration:4360, topPower:657),
41         new mobil26(merk:"Subaru", tipe:"BRZ", tahun:2014, topAcceleration:4058, topPower:609),
42         new mobil26(merk:"Subaru", tipe:"Impreza WRX STI", tahun:2013, topAcceleration:6255, topPower:703),
43         new mobil26(merk:"Toyota", tipe:"AE86 Trueno", tahun:1986, topAcceleration:3700, topPower:553),
44         new mobil26(merk:"Toyota", tipe:"86/GT86", tahun:2014, topAcceleration:4180, topPower:609),
45         new mobil26(merk:"Volkswagen", tipe:"Golf GTI", tahun:2014, topAcceleration:4180, topPower:631)
46     };
47
48     System.out.println(x:"Data Mobil :");
49     System.out.println(x:"=====");
50     for(mobil26 car : cars){
51         System.out.println(car);
52     }
53     System.out.println(x:"=====");

```

```

53
54     int maxAcc = maxAcceleration(cars, left:0, cars.length - 1);
55     int minAcc = minAcceleration(cars, left:0, cars.length - 1);
56     double avgTopPower = averageTopPower(cars);
57
58     System.out.println("Top Acceleration Tertinggi: " + maxAcc);
59     System.out.println("Top Acceleration Terendah: " + minAcc);
60     System.out.println("Rata - rata Top Power: " + avgTopPower);
61 }
62 }

```