

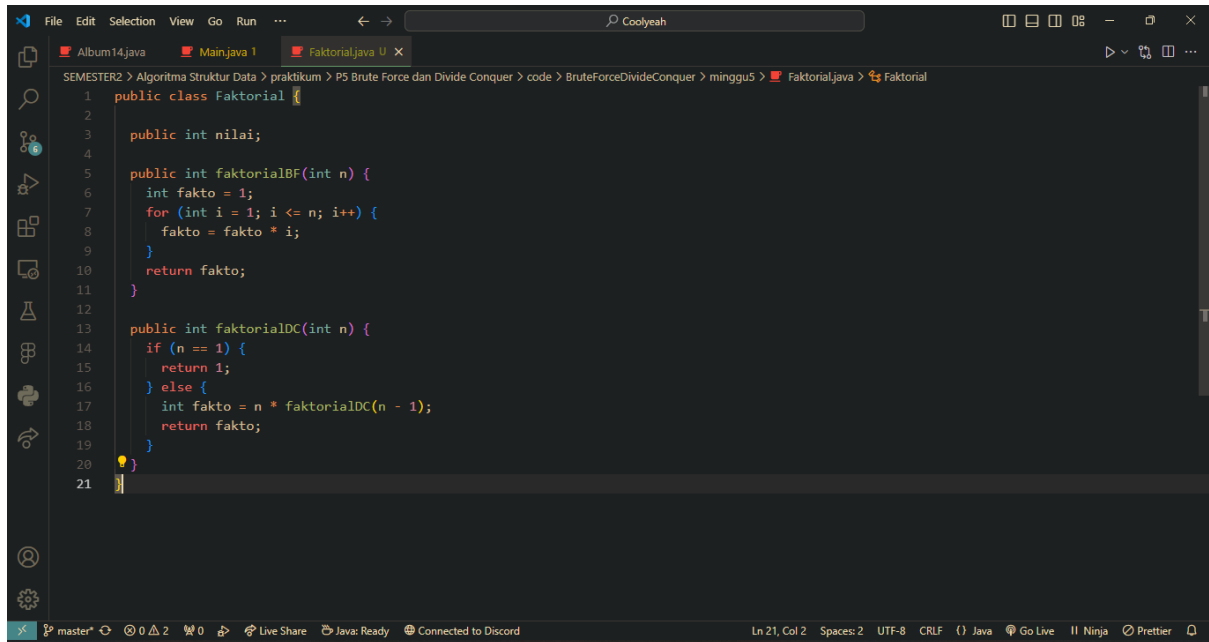
ALGORITMA STRUKTUR DATA

Praktikum – Brute Force and Divide Conquer

Lavina 2341760062

Praktikum 1

Faktorial.java



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

MainFaktorial.java

```
1 import java.util.Scanner;
2
3 public class MainFaktorial {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6
7         System.out.println("=====");
8         System.out.print("Masukkan jumlah elemen yang ingin dihitung: ");
9         int elemen = sc.nextInt();
10
11        Faktorial[] fk = new Faktorial[elemen];
12        for (int i = 0; i < elemen; i++) {
13            fk[i] = new Faktorial();
14            System.out.print("Masukkan nilai data ke-" + (1 + i) + ": ");
15            fk[i].nilai = sc.nextInt();
16        }
17
18        System.out.println("=====");
19        System.out.println("Hasil Faktorial dengan Brute Force");
20        for (int i = 0; i < elemen; i++) {
21            System.out.println("Faktorial dari nilai " + fk[i].nilai + " adalah: " + fk[i].faktorialBF(fk[i].nilai));
22        }
23    }
```

```

24     System.out.println("=====");
25     System.out.println("Hasil Faktorial dengan Divide Conquer");
26     for (int i = 0; i < elemen; i++) {
27         System.out.println("Faktorial dari nilai " + fk[i].nilai + " adalah: " + fk[i].faktorialDC(fk[i].nilai));
28     }
29 }
30 }
31

```

Output:

```

SEMESTER2 > Algoritma Struktur Data > praktikum > P5 Brute Force dan Divide Conquer > code > BruteForceDi
import java.util.Scanner;

Lavina@LAPTOP-VRURDV67 MINGW32 /d/Coolyeah (master)
$ ./usr/bin/env C:\Program Files\Java\jdk-21\bin\java.exe --enable-preview -XX:+ShowCodeDetailsInExceptionMessages -cp C:\Users\Lavina\AppData\Roaming\Code\User\workspac
eStorage\c46e7221177a4bd3a26c42b68d98275\lredhat.java\jdt_ws\Coolyeah_14a4c95\bin MainFaktorial
=====
Masukkan jumlah elemen yang ingin dihitung: 3
Masukkan nilai data ke-1: 5
Masukkan nilai data ke-2: 8
Masukkan nilai data ke-3: 3
=====
Hasil Faktorial dengan Brute Force
Faktorial dari nilai 5 adalah: 120
Faktorial dari nilai 8 adalah: 40320
Faktorial dari nilai 3 adalah: 6
=====
Hasil Faktorial dengan Divide Conquer
Faktorial dari nilai 5 adalah: 120
Faktorial dari nilai 8 adalah: 40320

Lavina@LAPTOP-VRURDV67 MINGW32 /d/Coolyeah (master)
$

```

Pertanyaan

1. Jelaskan mengenai base line Algoritma Divide Conquer untuk melakukan pencarian nilai faktorial!

Jawab: Base line dari algoritma itu ditunjukkan oleh sintaks $n == 1$ yang berarti jika n nya adalah 1 maka algoritma tidak perlu melakukan pembagian lagi sehingga program akan langsung mereturn nilai 1 karena faktorial dari 1 adalah 1.

2. Pada implementasi Algoritma Divide and Conquer Faktorial apakah lengkap terdiri dari 3 tahapan divide, conquer, combine? Jelaskan masing-masing bagiannya pada kode program!

Jawab :

Divide : if ($n == 1$) return 1;

Conquer : ada pada bagian else, $\text{int fakto} = n * \text{faktorialDC}(n-1)$; yang menggunakan rekursif.

Combine : return fakto;

3. Apakah memungkinkan perulangan pada method faktorialBF() dirubah selain menggunakan for? Buktikan!

```
public int faktorialBF(int n) {
    int fakto = 1;
    int i = 1;
    while (i <= n) {
        fakto = fakto * i;
        i++;
    }
    return fakto;
}
```

Outputnya :

```
PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS Run: M

Lavina@LAPTOP-VRURDV67 MINGW32 /d/Coolyeah (master)
$ /usr/bin/env C:\Program Files\Java\jdk-21\bin\java.exe --enable-preview -XX:+ShowCodeDetailsInExceptionMessages -cp
\Roaming\Code\User\workspaceStorage\c46e7221177a4bd3a26c42b680d90275\redhat.java\jdt_ws\Coolyeah_14a4c95\bin MainFak
=====
Masukkan jumlah elemen yang ingin dihitung: 3
Masukkan nilai data ke-1: 5
Masukkan nilai data ke-2: 8
Masukkan nilai data ke-3: 4
=====
Hasil Faktorial dengan Brute Force
Faktorial dari nilai 5 adalah: 120
Faktorial dari nilai 8 adalah: 40320
Faktorial dari nilai 4 adalah: 24
=====
Hasil Faktorial dengan Divide Conquer
Faktorial dari nilai 5 adalah: 120
Faktorial dari nilai 8 adalah: 40320
Faktorial dari nilai 4 adalah: 24
```

4. Tambahkan pengecekan waktu eksekusi kedua jenis method tersebut!

```
File Edit Selection View Go Run ... Cooleeah
Album14.java Main.java 1 Faktorial.java U MainFaktorial.java 1 U X
SEMESTER2 > Algoritma Struktur Data > praktikum > P5 Brute Force dan Divide Conquer > code > BruteForceDivideConquer > minggu5 > MainFaktorial.java > ...
3 public class MainFaktorial {
4     public static void main(String[] args) {
18         long start = System.nanoTime();
19         System.out.println(x:"=====");
20         System.out.println(x:"Hasil Faktorial dengan Brute Force");
21         for (int i = 0; i < elemen; i++) {
22             System.out.println("Faktorial dari nilai " + fk[i].nilai + " adalah: " + fk[i].faktorialBF(fk[i].nilai));
23         }
24         long end = System.nanoTime();
25         long timeElapsed1 = end - start;
26
27         start = System.nanoTime();
28         System.out.println(x:"=====");
29         System.out.println(x:"Hasil Faktorial dengan Divide Conquer");
30         for (int i = 0; i < elemen; i++) {
31             System.out.println("Faktorial dari nilai " + fk[i].nilai + " adalah: " + fk[i].faktorialDC(fk[i].nilai));
32         }
33         end = System.nanoTime();
34         long timeElapsed2 = end - start;
35
36         System.out.println("Waktu eksekusi BF: " + timeElapsed1);
37         System.out.println("Waktu eksekusi DC: " + timeElapsed2);
38     }
39 }
40
```

Output :

```
PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS

Lavina@LAPTOP-VRURDV67 MINGW32 /d/Coolyeah (master)
$ cd d:\\Coolyeah ; /usr/bin/env C:\\Program\\Files\\Java\\jdk-21\\bin\\java.exe --enable-preview -XX:+ShowCodeDetailsInException\\de\\User\\workspaceStorage\\c46e7221177a4bd3a26c42b680d90275\\redhat.java\\jdt_ws\\Coolyeah_14a4c95\\bin MainFaktorial
=====
Masukkan jumlah elemen yang ingin dihitung: 3
Masukkan nilai data ke-1: 5
Masukkan nilai data ke-2: 8
Masukkan nilai data ke-3: 4
=====
Hasil Faktorial dengan Brute Force
Faktorial dari nilai 5 adalah: 120
Faktorial dari nilai 8 adalah: 40320
Faktorial dari nilai 4 adalah: 24
=====
Hasil Faktorial dengan Divide Conquer
Faktorial dari nilai 5 adalah: 120
Faktorial dari nilai 8 adalah: 40320
Faktorial dari nilai 4 adalah: 24
Waktu eksekusi BF: 4843700
Waktu eksekusi DC: 1601800
```

5. Buktikan dengan inputan elemen yang di atas 20 angka, apakah ada perbedaan waktu eksekusi?

Jawab : Ada,

```
PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS Run: MainFaktorial +

Faktorial dari nilai 1 adalah: 1
Faktorial dari nilai 2 adalah: 2
Faktorial dari nilai 3 adalah: 6
Faktorial dari nilai 4 adalah: 24
Faktorial dari nilai 5 adalah: 120
=====
Hasil Faktorial dengan Divide Conquer
Faktorial dari nilai 1 adalah: 1
Faktorial dari nilai 2 adalah: 2
Faktorial dari nilai 3 adalah: 6
Faktorial dari nilai 4 adalah: 24
Faktorial dari nilai 5 adalah: 120
Faktorial dari nilai 6 adalah: 720
Faktorial dari nilai 7 adalah: 5040
Faktorial dari nilai 8 adalah: 40320
Faktorial dari nilai 9 adalah: 362880
Faktorial dari nilai 9 adalah: 362880
Faktorial dari nilai 8 adalah: 40320
Faktorial dari nilai 7 adalah: 5040
Faktorial dari nilai 6 adalah: 720
Faktorial dari nilai 5 adalah: 120
Faktorial dari nilai 4 adalah: 24
Faktorial dari nilai 3 adalah: 6
Faktorial dari nilai 2 adalah: 2
Faktorial dari nilai 1 adalah: 1
Faktorial dari nilai 1 adalah: 1
Faktorial dari nilai 2 adalah: 2
Faktorial dari nilai 3 adalah: 6
Faktorial dari nilai 4 adalah: 24
Faktorial dari nilai 5 adalah: 120
Waktu eksekusi BF: 15632600
Waktu eksekusi DC: 6558900
```

Praktikum 2

Pangkat.java

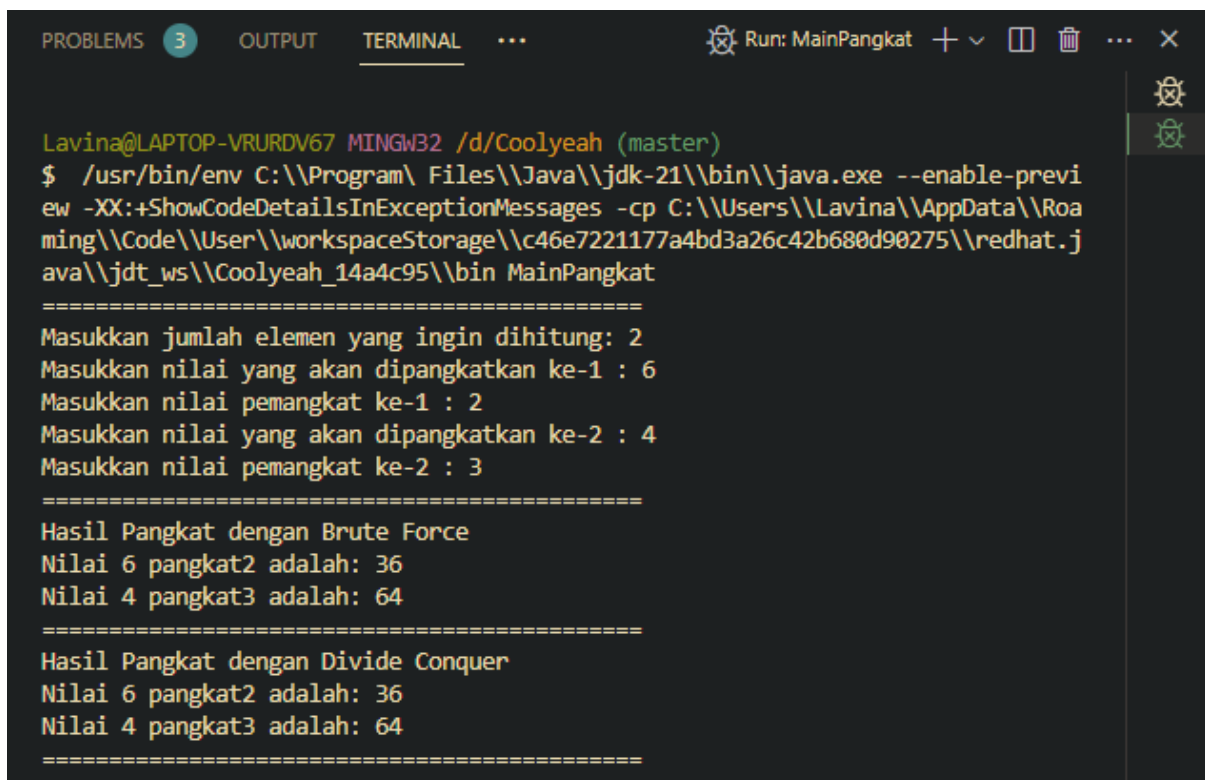
```
File Edit Selection View Go Run ... Coolyeah
Album14.java Main.java 1 Faktorial.java U MainFaktorial.java 1, U Pangkat.java X MainPangkat.java 1, U
SEMESTER2 > Algoritma Struktur Data > praktikum > P5 Brute Force dan Divide Conquer > code > BruteForceDivideConquer > minggu5 > Pangkat.java > ...
1 public class Pangkat {
2     public int nilai, pangkat;
3
4     public int pangkatBF(int a, int n) {
5         int hasil = 1;
6         for (int i = 0; i < n; i++) {
7             hasil = hasil * a;
8         }
9         return hasil;
10    }
```

```
11 public int pangkatDC(int a, int n) {
12     if (n == 0) {
13         return 1;
14     } else {
15         if (n % 2 == 1) {
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 }
24 }
```

MainPangkat.java

```
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
7         System.out.println("=====");
8         System.out.print("Masukkan jumlah elemen yang ingin dihitung: ");
9         int elemen = sc.nextInt();
10
11         Pangkat[] png = new Pangkat[elemen];
12         for (int i = 0; i < elemen; i++) {
13             png[i] = new Pangkat();
14             System.out.print("Masukkan nilai yang akan dipangkatkan ke-" + (i + 1) + " : ");
15             png[i].nilai = sc.nextInt();
16             System.out.print("Masukkan nilai pemangkat ke-" + (i + 1) + " : ");
17             png[i].pangkat = sc.nextInt();
18         }
19
20         System.out.println("=====");
21         System.out.println("Hasil Pangkat dengan Brute Force");
22         for (int i = 0; i < elemen; i++) {
23             System.out.println("Nilai " + png[i].nilai + " pangkat" + png[i].pangkat + " adalah: "
24                 + png[i].pangkatBF(png[i].nilai, png[i].pangkat));
25         }
26
27         System.out.println("=====");
28         System.out.println("Hasil Pangkat dengan Divide Conquer");
29         for (int i = 0; i < elemen; i++) {
30             System.out.println("Nilai " + png[i].nilai + " pangkat" + png[i].pangkat + " adalah: "
31                 + png[i].pangkatDC(png[i].nilai, png[i].pangkat));
32         }
33         System.out.println("=====");
34
35     }
36 }
37 }
```

Output:



```
Lavina@LAPTOP-VRURDV67 MINGW32 /d/Coolyeah (master)
$ /usr/bin/env C:\\Program\\ Files\\Java\\jdk-21\\bin\\java.exe --enable-previ
ew -XX:+ShowCodeDetailsInExceptionMessages -cp C:\\Users\\Lavina\\AppData\\Roa
ming\\Code\\User\\workspaceStorage\\c46e7221177a4bd3a26c42b680d90275\\redhat.j
ava\\jdt_ws\\Coolyeah_14a4c95\\bin MainPangkat
=====
Masukkan jumlah elemen yang ingin dihitung: 2
Masukkan nilai yang akan dipangkatkan ke-1 : 6
Masukkan nilai pemangkat ke-1 : 2
Masukkan nilai yang akan dipangkatkan ke-2 : 4
Masukkan nilai pemangkat ke-2 : 3
=====
Hasil Pangkat dengan Brute Force
Nilai 6 pangkat2 adalah: 36
Nilai 4 pangkat3 adalah: 64
=====
Hasil Pangkat dengan Divide Conquer
Nilai 6 pangkat2 adalah: 36
Nilai 4 pangkat3 adalah: 64
=====
```

Pertanyaan

1. Jelaskan mengenai perbedaan 2 method yang dibuat yaitu PangkatBF() dan PangkatDC()!

Jawab : Pada method PangkatBF() menggunakan loop untuk menghitung pangkat secara iterative sedangkan untuk method PangkatDC() yang menerapkan divide and conquer menggunakan rekursi.

2. Pada method PangkatDC() terdapat potongan program sebagai berikut:

```
if(n%2==1)//bilangan ganjil
    return (pangkatDC(a,n/2)*pangkatDC(a,n/2)*a);
else//bilangan genap
    return (pangkatDC(a,n/2)*pangkatDC(a,n/2));
```

Jelaskan arti potongan kode tersebut !

Jawab : Pada blok if mengecek apakah n bilangan ganjil, jika n adalah bilangan ganjil maka akan dilakukan perhitungan $n / 2$ dikali dengan $n / 2$ lalu dikali dengan a. Jika bilangannya genap maka dilakukan perhitungan $n / 2$ dikali dengan $n / 2$.

3. Apakah tahap combine sudah termasuk dalam kode tersebut? Tunjukkan!

Jawab : Sudah, tahap combine ditunjukkan oleh sintaks berikut :

```
return (pangkatDC(a, n / 2) * pangkatDC(a, n / 2) * a); dan
```

```
return (pangkatDC(a, n / 2) * pangkatDC(a, n / 2));
```

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

Konstruktor pada Pangkat.java

```
public Pangkat(int n, int p) {
    nilai = n;
    pangkat = p;
}
```

Pengisian atribut pada MainPangkat.java

```
Pangkat[] png = new Pangkat[elemen];
for (int i = 0; i < elemen; i++) {
    System.out.print("Masukkan nilai yang akan dipangkatkan ke-" + (i + 1) + " : ");
    nilai = sc.nextInt();
    System.out.print("Masukkan nilai pemangkat ke-" + (i + 1) + " : ");
    pangkat = sc.nextInt();

    png[i] = new Pangkat(nilai, pangkat);
}
```

Output :

```

PROBLEMS 3 OUTPUT TERMINAL ...
Lavina@LAPTOP-VRURDV67 MINGW32 /d/Coolyeh (master)
$ /usr/bin/env C:\Program Files\Java\jdk-21\bin\java.exe --enable-preview -XX:+ShowCodeDetailsInExceptionMessages -cp C:\Users\Lavina\AppData\Roaming\Code\User\workspaceStorage\c46e7221177a4bd3a26c42b680d90275\redhat.java\jdt_ws\Coolyeh_14a4c95\bin MainPangkat
=====
Masukkan jumlah elemen yang ingin dihitung: 2
Masukkan nilai yang akan dipangkatkan ke-1 : 6
Masukkan nilai pemangkat ke-1 : 2
Masukkan nilai yang akan dipangkatkan ke-2 : 4
Masukkan nilai pemangkat ke-2 : 3
=====
Hasil Pangkat dengan Brute Force
Nilai 6 pangkat2 adalah: 36
Nilai 4 pangkat3 adalah: 64
=====
Hasil Pangkat dengan Divide Conquer
Nilai 6 pangkat2 adalah: 36
Nilai 4 pangkat3 adalah: 64
=====

```

5. Tambahkan menu agar salah satu method yang terpilih saja yang akan dijalankan!

Jawab : Modifikasi pada class main,

```

1  System.out.println("=====");
2  System.out.println("Menu");
3  System.out.println("1. BF\n2. DC");
4  System.out.print("Pilih: ");
5  int pilihan = sc.nextInt();
6
7  switch (pilihan) {
8      case 1:
9          System.out.println("=====");
10         System.out.println("Hasil Pangkat dengan Brute Force");
11         for (int i = 0; i < elemen; i++) {
12             System.out.println("Nilai " + png[i].nilai + " pangkat" + png[i].pangkat + " adalah: "
13                 + png[i].pangkatBF(png[i].nilai, png[i].pangkat));
14         }
15         break;
16     case 2:
17         System.out.println("=====");
18         System.out.println("Hasil Pangkat dengan Divide Conquer");
19         for (int i = 0; i < elemen; i++) {
20             System.out.println("Nilai " + png[i].nilai + " pangkat" + png[i].pangkat + " adalah: "
21                 + png[i].pangkatDC(png[i].nilai, png[i].pangkat));
22         }
23         break;
24     default:
25         break;
26 }
27
28 System.out.println("=====");
29

```

Output :

```

Lavina@LAPTOP-VRURDV67 MINGW32 /d/Coolyeah (master)
$ /usr/bin/env C:\\Program Files\\Java\\jdk-21\\bin\\java.exe --enable-preview -XX:+ShowCodeDetails:
eStorage\\c46e7221177a4bd3a26c42b680d90275\\redhat.java\\jdt_ws\\Coolyeah_14a4c95\\bin MainPangkat
=====
Masukkan jumlah elemen yang ingin dihitung: 2
Masukkan nilai yang akan dipangkatkan ke-1 : 6
Masukkan nilai pemangkat ke-1 : 2
Masukkan nilai yang akan dipangkatkan ke-2 : 3
Masukkan nilai pemangkat ke-2 : 4
=====
Menu
1. BF
2. DC
Pilih: 1
=====
Hasil Pangkat dengan Brute Force
Nilai 6 pangkat2 adalah: 36
Nilai 3 pangkat4 adalah: 81
=====

```


Praktikum 4

Sum.java

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

MainSum.java

```
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
7         System.out.println("=====");
8         System.out.println("Program Menghitung Keuntungan Total (Satuan Juta Misal 5.9)");
9         System.out.print("Masukkan jumlah bulan: ");
10        int elm = sc.nextInt();
11
12        Sum sm = new Sum(elm);
13        System.out.println("=====");
14        for (int i = 0; i < sm.elemen; i++) {
15            System.out.print("Masukkan untung bulan ke-" + (1 + i) + " : ");
16            sm.keuntungan[i] = sc.nextDouble();
17        }
18
19        System.out.println("=====");
20        System.out.println("Algoritma Brute Force");
21        System.out
22            .println("Total keuntungan perusahaan selama " + sm.elemen + " bulan adalah : " + sm.totalBF(sm.keuntungan));
23        System.out.println("=====");
24        System.out.println("Algoritma Divide Conquer");
25        System.out.println("Total keuntungan perusahaan selama " + sm.elemen + " bulan adalah : "
26            + sm.totalDC(sm.keuntungan, 0, sm.elemen - 1));
27
28    }
29 }
30
```

Output :

```
Lavina@LAPTOP-VRURDV67 MINGW32 /d/Coolyeah (master)
$ cd d:\Coolyeah ; /usr/bin/env C:\Program Files\Java\jdk-21\bin\java.exe --enable-preview -XX:+ShowCodeDetails -Duser.dir=d:\User\workspaceStorage\c46e7221177a4bd3a26c42b680d90275\redhat.java\jdt_ws\Coolyeah_14a4c95\bin MainSum
=====
Program Menghitung Keuntungan Total (Satuan Juta Misal 5.9)
Masukkan jumlah bulan: 5
=====
Masukkan untung bulan ke-1 : 8.5
Masukkan untung bulan ke-2 : 9.54
Masukkan untung bulan ke-3 : 7.2
Masukkan untung bulan ke-4 : 9.1
Masukkan untung bulan ke-5 : 6
=====
Algoritma Brute Force
Total keuntungan perusahaan selama 5 bulan adalah : 40.339999999999996
=====
Algoritma Divide Conquer
Total keuntungan perusahaan selama 5 bulan adalah : 40.34
```

Pertanyaan

1. Berikan ilustrasi perbedaan perhitungan keuntungan dengan method TotalBF() ataupun TotalDC()

Jawab : Jika menggunakan method brute force, keuntungan akan dihitung secara berurutan dari bulan pertama hingga bulan ke-n. Misalnya, untuk perhitungan keuntungan selama 5 bulan diatas, dengan menggunakan brute force perhitungannya menjadi : $8.5 + 9.54 + 7.2 + 9.1 + 6 = 40.339999999999996$. Sedangkan jika

menggunakan divide conquer perhitungannya akan dibagi menjadi dua sub masalah yang lebih kecil dalam kode diatas dibagi menjadi sisi kiri dan kanan, sehingga perhitungannya :

$$\text{Kiri} = 8.5 + 9.54 = 18,04$$

$$\text{Kanan} = 6 + 9.1 = 15,1$$

Lalu hasil perhitungan kedua sisi tersebut disatukan (combine) juga ditambah dengan nilai tengah (mid) sehingga perhitungan akhirnya : $18,04 + 15,1 + 7,2 = 40,34$.

2. Perhatikan output dari kedua jenis algoritma tersebut bisa jadi memiliki hasil berbeda di belakang koma. Bagaimana membatasi output di belakang koma agar menjadi standar untuk kedua jenis algoritma tersebut.
3. Mengapa terdapat formulasi return value berikut?Jelaskan!

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

Jawab : Formulasi tersebut untuk menjumlahkan hasil dari perhitungan di sisi kiri, kanan dan tengah.

4. Kenapa dibutuhkan variable mid pada method TotalDC()?

Jawab : Variabel mid dibutuhkan untuk membagi array arr[] menjadi dua bagian.

5. 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!

Jawab : Modifikasi pada class main,

```

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
7          System.out.println("=====");
8          System.out.print("Masukkan Jumlah Perusahaan : ");
9          int comp = sc.nextInt();
10
11         for (int i = 0; i < comp; i++) {
12             System.out.println("=====");
13             System.out.println("Perusahaan " + (1 + i));
14             System.out.println("Program Menghitung Keuntungan Total (Satuan Juta Misal 5.9)");
15             System.out.print("Masukkan jumlah bulan: ");
16             int elm = sc.nextInt();
17
18             Sum sm = new Sum(elm);
19             System.out.println("=====");
20             for (int j = 0; j < sm.elemen; j++) {
21                 System.out.print("Masukkan untung bulan ke-" + (1 + j) + " : ");
22                 sm.keuntungan[j] = sc.nextDouble();
23             }
24
25             System.out.println("=====");
26             System.out.println("Algoritma Brute Force");
27             System.out
28                 .println("Total keuntungan perusahaan selama " + sm.elemen + " bulan adalah : " + sm.totalBF(sm.keuntungan));
29             System.out.println("=====");
30             System.out.println("Algoritma Divide Conquer");
31             System.out.println("Total keuntungan perusahaan selama " + sm.elemen + " bulan adalah : "
32                 + sm.totalDC(sm.keuntungan, 0, sm.elemen - 1));
33         }
34     }
35 }
36 }
37

```

Output :

```

=====
Masukkan Jumlah Perusahaan : 2
=====
Perusahaan 1
Program Menghitung Keuntungan Total (Satuan Juta Misal 5.9)
Masukkan jumlah bulan: 2
=====
Masukkan untung bulan ke-1 : 4.5
Masukkan untung bulan ke-2 : 4.5
=====
Algoritma Brute Force
Total keuntungan perusahaan selama 2 bulan adalah : 9.0
=====
Algoritma Divide Conquer
Total keuntungan perusahaan selama 2 bulan adalah : 9.0
=====
Perusahaan 2
Program Menghitung Keuntungan Total (Satuan Juta Misal 5.9)
Masukkan jumlah bulan: 3
=====
Masukkan untung bulan ke-1 : 2
Masukkan untung bulan ke-2 : 3.5
Masukkan untung bulan ke-3 : 2
=====
Algoritma Brute Force
Total keuntungan perusahaan selama 3 bulan adalah : 7.5
=====
Algoritma Divide Conquer
Total keuntungan perusahaan selama 3 bulan adalah : 7.5

```

Latihan

Buatlah kode program untuk menghitung nilai akar dari suatu bilangan dengan algoritma Brute Force dan Divide Conquer! Jika bilangan tersebut bukan merupakan kuadrat sempurna, bulatkan angka ke bawah

Akar.java

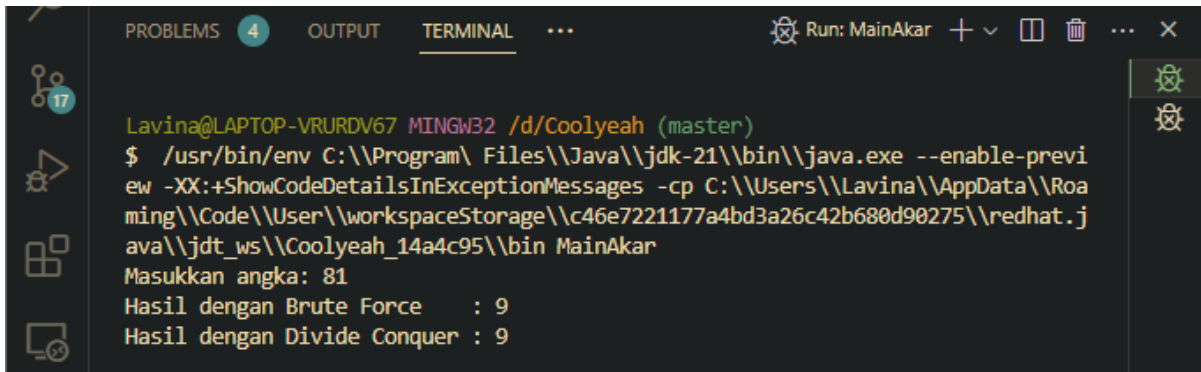
Method dengan menggunakan Brute Force:

```
1  public class Akar {
2      public int nilai;
3
4      public int akarBF(int n) {
5          for (int i = 0; i <= n; i++) {
6              if (i * i == n) {
7                  return i;
8              }
9          }
10         return -1;
11     }
```

Method dengan menggunakan Divide Conquer:

```
1    public int akarDC(int n) {
2        if (n == 0 || n == 1) {
3            return n;
4        }
5
6        int low = 1;
7        int high = n;
8
9        while (low <= high) {
10           int mid = (low + high) / 2;
11           int midAkar = mid * mid;
12
13           if (midAkar == n) {
14               return mid;
15           } else if (midAkar < n) {
16               low = mid + 1;
17           } else {
18               high = mid - 1;
19           }
20       }
21       return high;
22   }
23
24 }
```

Output:



```
Lavina@LAPTOP-VRURDV67 MINGW32 /d/Coolyeah (master)
$ /usr/bin/env C:\\Program\\ Files\\Java\\jdk-21\\bin\\java.exe --enable-previ
ew -XX:+ShowCodeDetailsInExceptionMessages -cp C:\\Users\\Lavina\\AppData\\Roa
ming\\Code\\User\\workspaceStorage\\c46e7221177a4bd3a26c42b680d90275\\redhat.j
ava\\jdt_ws\\Coolyeah_14a4c95\\bin MainAkar
Masukkan angka: 81
Hasil dengan Brute Force      : 9
Hasil dengan Divide Conquer  : 9
```