

Laporan Praktikum Algoritma dan Struktur Data
Jobsheet 5 – Brute Force dan Divide Conquer

Dosen Pengampu : Ibu Triana Fatmawati, S.T., M.T



Nama : Annisa
Nim : 2341760032
Prodi : D-IV Sistem Informasi Bisnis
Absen : 04

JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
2023/2024

4.2 Menghitung Nilai Faktorial dengan Algoritma Brute Force dan Divide and Conquer

4.2.1 Langkah-langkah Percobaan

Berikut merupakan kode program dari class Faktorial

```
1  /**
2   * Faktorial_04
3   */
4  public class Faktorial_04 {
5
6      public int nilai;
7      public int faktorialBF (int n) {
8          int fakto04 = 1;
9          for (int i = 1; i <= n; i++) {
10             fakto04 = fakto04 * i;
11         }
12         return fakto04;
13     }
14
15     public int faktorialDC(int n) {
16         if (n==1) {
17             return 1;
18         }
19         else
20         {
21             int fakto04 = n * faktorialDC (n-1);
22             return fakto04;
23         }
24     }
25 }
26 }
```

Berikut merupakan kode program dari class MainFaktorial

```
1  import java.util.Scanner;
2  /**
3   * MainFaktorial_04
4   */
5  public class MainFaktorial_04 {
6
7      public static void main(String[] args) {
8          Scanner sc04 = new Scanner(System.in);
9
10         System.out.println("=====");
11         System.out.print("Masukkan jumlah elemen yang dihitung : ");
12         int elemen = sc04.nextInt ();
13
14         Faktorial_04 [] fk04 = new Faktorial_04 [elemen];
15         for (int i = 0; i < elemen; i++) {
16             fk04 [i] = new Faktorial_04();
17             System.out.print("Masukkan nilai data ke-" + (i+1) + " : ");
18             fk04[i].nilai = sc04.nextInt();
19         }
20         System.out.println("=====");
21         System.out.println("Hasil Faktorial dengan Brute Force");
22         for (int i = 0; i < elemen; i++) {
23             System.out.println("Faktorial dari nilai " + fk04[i].nilai + " adalah : " + fk04[i].faktorialBF(fk04[i].nilai));
24         }
25         System.out.println("=====");
26         System.out.println("Hasil Faktorial dengan Divide and conquer");
27
28         for (int i = 0; i < elemen; i++){
29             System.out.println("Faktorial dari nilai " + fk04[i].nilai + " adalah : " + fk04[i].faktorialDC(fk04[i].nilai));
30         }
31         System.out.println("=====");
32     }
33 }
```

4.2.2 Verifikasi Hasil Percobaan

```
C:\Users\user> cmd /C "C:\Users\user\.jdk\openjdk-21.0.2\bin\java.exe --enable-preview
-cp C:\Users\user\AppData\Local\Temp\vscodesws_f86b0\jdt_ws\jdt.ls-java-project\bin Main
=====
Masukkan jumlah elemen yang 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 and conquer
Faktorial dari nilai 5 adalah : 120
Faktorial dari nilai 8 adalah : 40320
Faktorial dari nilai 3 adalah : 6
=====
C:\Users\user>
```

4.2.3 Pertanyaan

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

Base line pada algoritma Divide Conquer pada pencarian nilai factorial berada pada line 15 di class Faktorial. Karena pada line tersebut merupakan sebuah tanda akhir dari pemanggilan fungsi rekursif.

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

Lengkap. Dimana pembagiannya :

- Divide ; pada if else, dimana setiap case akan dibagi
- Conquer : untuk conquer pada sintask “faktorialDC(n-1)”
- Combine : pada sintaks “int fakto = n * faktorialDC(n-1)”

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

Bisa, berikut bukti screenshot dengan perulangan while

```
1  /**
2   * Faktorial_04
3   */
4  public class Faktorial_04 {
5
6      public int nilai;
7      public int faktorialBF (int n) {
8          int fakto04 = 1;
9          int i = 1;
10         while (i <= n) {
11             fakto04 = fakto04*i;
12             i++ ;
13         }
14         return fakto04;
15     }
16     public int faktorialDC(int n) {
17         if (n==1) {
18             return 1;
19         }
20         else
21         {
22             int fakto04 = n * faktorialDC (n-1);
23             return fakto04;
24         }
25     }
26 }
27 }
```

4. Tambahkan pengecekan waktu eksekusi kedua jenis method tersebut!

Berikut merupakan kode program yang sudah ditambah pengecekan waktu eksekusi

```
1  import java.util.Scanner;
2  /**
3   * MainFaktorial_04
4   */
5  public class MainFaktorial_04 {
6
7      public static void main(String[] args) {
8          Scanner sc04 = new Scanner(System.in);
9
10         System.out.println("=====");
11         System.out.print("Masukkan jumlah elemen yang dihitung : ");
12         int elemen = sc04.nextInt ();
13
14         Faktorial_04 [] fk04 = new Faktorial_04 [elemen];
15         for (int i = 0; i < elemen; i++) {
16             fk04 [i] = new Faktorial_04();
17             System.out.print("Masukkan nilai data ke- "+(i+1)+" : ");
18             fk04[i].nilai = sc04.nextInt();
19         }
20         System.out.println("=====");
21         System.out.println("Hasil Faktorial dengan Brute Force");
22         long startTime = System.nanoTime();
23         for(int i = 0; i < elemen; i++){
24             System.out.println("Faktorial dari nilai " + fk04[i].nilai + " adalah : " + fk04[i].faktorialBF(fk04[i].nilai));
25         }
26         long endTime = System.nanoTime();
27         double timeTakenBF = (endTime-startTime)/1_000_000_000.0;
28         System.out.println("Waktu eksekusi Brute Force: " + (timeTakenBF)+ " seconds");
29         startTime = System.nanoTime();
30         System.out.println("=====");
31         System.out.println("Hasil Faktorial dengan Divide and Conquer");
32         for (int i = 0; i < elemen; i++) {
33             System.out.println("Faktorial dari nilai "+fk04[i].nilai+" adalah : "+fk04[i].faktorialDC(fk04[i].nilai));
34         }
35         endTime = System.nanoTime();
36         double timeTakenDC = (endTime - startTime)/1_000_000_000.0;
37         System.out.println("Waktu eksekusi Brute Force: " + (timeTakenDC)+ "seconds");
38         System.out.println("=====");
39     }
40 }
```

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

```
Faktorial dari nilai 20 adalah : 1
Waktu eksekusi Brute Force: 0.0116637seconds
=====
C:\Users\user>
```

4.3 Menghitung Hasil Pangkat dengan Algoritma Brute Force dan Divide and Conquer

4.3.1 Langkah-langkah Percobaan

```
1  /**
2   * pangkat_04
3   */
4  public class pangkat_04 {
5
6      public int nilai, pangkat_04;
7
8      public int pangkatBF (int a, int n) {
9          int hasil = 1;
10         for (int i = 0; i < n; i++) {
11             hasil = hasil * a ;
12         }
13         return hasil;
14     }
15
16     public int pangkatDC(int a, int n) {
17         if (n==0) {
18             return 1;
19         } else {
20             if (n%2==1) //bilangan ganjil
21                 return (pangkatDC(a, n/2) * pangkatDC(a, n/2)*a);
22             else //bilangan genap
23                 return (pangkatDC(a, n/2) * pangkatDC(a, n/2));
24         }
25     }
26 }
```

```
1  import java.util.Scanner;
2  /**
3   * MainPangkat_04
4   */
5  public class MainPangkat_04 {
6
7      public static void main(String[] args) {
8          Scanner sc04 = new Scanner(System.in);
9          System.out.println("=====");
10         System.out.print("Masukkan jumlah elemen yang ingin dihitung : ");
11         int elemen = sc04.nextInt();
12         pangkat_04 [] png = new pangkat_04[elemen];
13
14         for (int i = 0; i < elemen; i++) {
15             png[i] = new pangkat_04();
16             System.out.print("Masukkan nilai yang akan dipangkatkan ke-" + (i+1) + " : ");
17             png[i].nilai = sc04.nextInt();
18             System.out.print("Masukkan nilai pemangkat ke-" + (i+1) + " : ");
19             png[i].pangkat_04 = sc04.nextInt();
20         }
21
22         System.out.println("=====");
23         System.out.println("Hasil Pangkat dengan Brute Force ");
24         for (int i = 0; i < elemen; i++) {
25             System.out.println("Nilai "+png[i].nilai+" pangkat "+png[i].pangkat_04+" adalah : "+png[i].pangkatBF(png[i].nilai, png[i].pangkat_04));
26         }
27
28         System.out.println("=====");
29         System.out.println("Hasil Pangkat dengan Divide dan Conquer");
30         for (int i = 0; i < elemen; i++) {
31             System.out.println("Nilai "+png[i].nilai+" pangkat "+png[i].pangkat_04+" adalah : "+png[i].pangkatDC(png[i].nilai, png[i].pangkat_04));
32         }
33         System.out.println("=====");
34     }
35 }
36 }
```

4.3.2 Verifikasi Hasil Percobaan

```
=====
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 pangkat 2 adalah : 36
Nilai 4 pangkat 3 adalah : 64
=====
Hasil Pangkat dengan Divide dan Conquer
Nilai 6 pangkat 2 adalah :36
Nilai 4 pangkat 3 adalah :64
=====
C:\Users\user\Documents\ASD Jobsheet 5>
```

4.3.3 Pertanyaan

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

Method PangkatBF() merupakan method menggunakan perulangan kalkulasinya, sehingga ketika ada data yang lebih banyak durasi akan lebih lama.

Method PangkatDC() merupakan method yang menggunakan fungsi rekursif dengan melakukan pembagian case hingga menemukan solusi lalu dicombine sehingga mendapatkan solusi akhir yang lebih cepat dibandingkan dengan method PangkatBF().

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

Potongan kode tersebut akan dilakukan sebuah pembagian case dengan menggunakan if else ketika kondisi $n \% 2 == 1$ bernilai true maka program akan menjalankan statement $\text{pangkatDC}(a, n/2) * \text{pangkatDC}(a, n/2) * a$. Dimana statement tersebut merupakan solusi pertama ketika kondisi awal bernilai true, lalu akan dilakukan pemanggilan fungsi lagi jika kondisi n bernilai false dan menjalankan statement $\text{pangkatDC}(a, n/2) * \text{pangkatDC}(a, n/2)$ sebagai solusi ke-2 hingga terakhir. Pada n yang dimaksud pada pemanggilan fungsi tersebut terdapat perbedaan nilai yang kiri dan kanan dimana n kiri akan dicari dulu lalu ke n kanan dengan nilai yang berbeda.

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

Sudah, tahap combine terdapat pada kode:

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

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

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

Berikut kode program hasil modifikasi

```
1  /**
2   * pangkat_04
3   */
4  public class pangkat_04 {
5
6      public int nilai, pangkat_04;
7
8      public int pangkatBF (int a, int n) {
9          int hasil = 1;
10         for (int i = 0; i < n; i++) {
11             hasil = hasil * a ;
12         }
13         return hasil;
14     }
15
16     public int pangkatDC(int a, int n) {
17         if (n==0) {
18             return 1;
19         } else {
20             if (n%2==1) //bilangan ganjil
21                 return (pangkatDC(a, n/2) * pangkatDC(a, n/2)*a);
22             else //bilangan genap
23                 return (pangkatDC(a, n/2) * pangkatDC(a, n/2));
24         }
25     }
26
27     public pangkat_04(){
28         this.nilai = nilai;
29         this.pangkat_04 = pangkat_04;
30     }
31     public pangkat_04 (int nilai2, int pangkat2) {
32     }
33 }
```

```
1  import java.util.Scanner;
2  /**
3   * MainPangkat_04
4   */
5  public class MainPangkat_04 {
6
7      public static void main(String[] args) {
8          Scanner sc04 = new Scanner(System.in);
9          System.out.println("=====");
10         System.out.print("Masukkan jumlah elemen yang ingin dihitung : ");
11         int elemen = sc04.nextInt();
12
13         pangkat_04 [] png = new pangkat_04[elemen];
14
15         for (int i = 0; i < elemen; i++) {
16             png[i] = new pangkat_04();
17             System.out.print("Masukkan nilai yang akan dipangkatkan ke-" + (i+1) + " : ");
18             png[i].nilai = sc04.nextInt ();
19             System.out.print("Masukkan nilai pangkat ke-" + (i+1) + " : ");
20             png[i].pangkat_04 = sc04.nextInt();
21         }
22
23         System.out.println("=====");
24         System.out.println("Pilih metode yang ingin dijalankan:");
25         System.out.println("1. Brute Force");
26         System.out.println("2. Divide and Conquer");
27         System.out.print("Masukkan pilihan anda: ");
28         int pilih = sc04.nextInt();
29
30         while (true) {
31             switch (pilih) {
32                 case 1:
33                     System.out.println("=====");
34                     System.out.println("Hasil pangkat dengan Brute Force");
35                     for (int i = 0; i < elemen; i++){
36                         System.out.println("Nilai " + png[i].nilai + " pangkat " + png[i].pangkat_04 + "adalah : " + png[i].pangkatBF(png[i].nilai, png[i].pangkat_04));
37                     }
38                     break;
39                 case 2:
40                     System.out.println("=====");
41                     System.out.println("Hasil pangkat dengan Divide and Conquer");
42                     for (int i = 0; i < elemen; i++) {
43                         System.out.println("Nilai " + png[i].nilai + " pangkat " + png[i].pangkat_04 + "adalah : " + png[i].pangkatDC(png[i].nilai, png[i].pangkat_04));
44                     } break;
45                 default:
46                     System.out.println("=====");
47                     System.out.println("=====");
48                     System.out.println("Pilih metode yang ingin dijalankan");
49                     System.out.println("1. Brute Force");
50                     System.out.println("2. Divide and Conquer");
51                     System.out.print("Masukkan pilihan anda: ");
52                     pilih = sc04.nextInt();
53                     if (pilih == 1 || pilih == 2) {
54                         break;
55                     }
56             }
57         }
58     }
59 }
60 }
```

Berikut merupakan hasil running dari kode program di atas

```
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
=====
Pilih metode yang ingin dijalankan:
1. Brute Force
2. Divide and Conquer
Masukkan pilihan anda: 1
=====
Hasil pangkat dengan Brute Force
Nilai 6 pangkat 2 adalah : 36
Nilai 4 pangkat 3 adalah : 64

C:\Users\User\Documents\ASD Jobsheet 5>
```

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

```
1 import java.util.Scanner;
2
3 //
4 *
5 *
6 public class MainPangkat_M4 {
7
8     public static void main(String[] args) {
9         Scanner scM = new Scanner(System.in);
10         System.out.println("=====");
11         System.out.print("Masukkan jumlah elemen yang ingin dihitung : ");
12         int elemen = scM.nextInt();
13
14         pangkat_M4 [] pg = new pangkat_M4[elemen];
15
16         for (int i = 0; i < elemen; i++) {
17             pg[i] = new pangkat_M4();
18             System.out.print("Masukkan nilai yang akan dipangkatkan ke- " + (i+1) + " : ");
19             pg[i].nilai = scM.nextInt();
20             System.out.print("Masukkan nilai pemangkat ke- " + (i+1) + " : ");
21             pg[i].pangkat_M4 = scM.nextInt();
22         }
23
24         System.out.println("=====");
25         System.out.print("Pilih metode yang ingin dijalankan:");
26         System.out.println("1. Brute Force");
27         System.out.println("2. Divide and Conquer");
28         System.out.print("Masukkan pilihan anda: ");
29         int pilih = scM.nextInt();
30
31         while (true) {
32             switch (pilih) {
33                 case 1:
34                     System.out.println("=====");
35                     System.out.print("Hasil pangkat dengan Brute Force");
36                     for (int i = 0; i < elemen; i++) {
37                         System.out.print("Nilai " + pg[i].nilai + " pangkat " + pg[i].pangkat_M4 + " adalah : " + pg[i].pangkatM(pg[i].nilai, pg[i].pangkat_M4));
38                     }
39                     break;
40                 case 2:
41                     System.out.println("=====");
42                     System.out.print("Hasil pangkat dengan Divide and Conquer");
43                     for (int i = 0; i < elemen; i++) {
44                         System.out.print("Nilai " + pg[i].nilai + " pangkat " + pg[i].pangkat_M4 + " adalah : " + pg[i].pangkatDK(pg[i].nilai, pg[i].pangkat_M4));
45                     }
46                     break;
47             }
48             System.out.println("=====");
49             System.out.print("Pilih metode yang ingin dijalankan:");
50             System.out.println("1. Brute Force");
51             System.out.println("2. Divide and Conquer");
52             System.out.print("Masukkan pilihan anda: ");
53             pilih = scM.nextInt();
54
55             if (pilih == 1 || pilih == 2) {
56                 break;
57             }
58         }
59     }
60 }
```

```
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
=====
Pilih metode yang ingin dijalankan:
1. Brute Force
2. Divide and Conquer
Masukkan pilihan anda: 1
=====
Hasil pangkat dengan Brute Force
Nilai 6 pangkat 2 adalah : 36
Nilai 4 pangkat 3 adalah : 64

C:\Users\User\Documents\ASD Jobsheet 5>
```


4.4 Menghitung Sum Array dengan Algoritma Brute Force dan Divide and Conquer

4.4.1 Langkah-langkah Percobaan

```
1  /**
2   * Sum_04
3   */
4  public class Sum_04 {
5
6      public int elemen;
7      public double keuntungan [];
8      public double total;
9
10     Sum_04 (int elemen) {
11         this.elemen = elemen;
12         this.keuntungan = new double[elemen];
13         this.total = 0;
14     }
15
16     double totalBF(double arr[]) {
17         for (int i = 0; i < elemen; i++){
18             total = total + arr [i];
19         }
20         return total;
21     }
22
23     double totalDC(double arr[], int l, int r) {
24         if (l == r)
25             return arr[l];
26         else if (l < r) {
27             int mid = (l+r)/2;
28             double lsum = totalDC(arr,l,mid-1);
29             double rsum = totalDC(arr,mid+1,r);
30             return lsum+rsum+arr[mid];
31         }
32
33         return 0;
34     }
35 }
```

```
1  import java.util.Scanner;
2  /**
3   * MainSum_04
4   */
5  public class MainSum_04 {
6
7      public static void main(String[] args) {
8          Scanner sc04 = new Scanner(System.in);
9          System.out.println("=====");
10         System.out.println("Program Menghitung Keuntungan Total (Satuan Juta.Misal 5.9)");
11         System.out.print("Masukkan Jumlah Bulan : ");
12         int elm = sc04.nextInt();
13
14         Sum_04 sm = new Sum_04(elm);
15         System.out.println("=====");
16         for (int i = 0; i < sm.elemen; i++){
17             System.out.print("Masukkan untung bulan ke - " + (i+1) + " = ");
18             sm.keuntungan[i] = sc04.nextDouble();
19         }
20
21         System.out.println("=====");
22         System.out.println("Algoritma Brute Force");
23         System.out.println("Total keuntungan perusahaan selama " + sm.elemen + " bulan adalah = " + sm.totalBF(sm.keuntungan));
24         System.out.println("=====");
25         System.out.println("Algoritma Divide Conquer");
26         System.out.println("Total keuntungan perusahaan selama " + sm.elemen+ " bulan adalah = " + sm.totalDC(sm.keuntungan, 0, sm.elemen-1));
27
28     }
29 }
```

4.4.2 Verifikasi Hasil Percobaan

```
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
C:\Users\user\Documents\ASD Jobsheet 5>
```

4.4.3 Pertanyaan

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

Keuntungan menghitung dengan menggunakan method TotalBF() adalah lebih mudah untuk menghitung kalkulasi dan dapat mempermudah membuat program karena tidak terlalu sulit untuk merancang programnya. Sedangkan method TotalDC() lebih cepat dalam proses kalkulasi ketika dijalankan.

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.

```
1  /**
2   * Sum_04
3   */
4  public class Sum_04 {
5
6      public int elemen;
7      public double keuntungan [];
8
9      public Sum_04 (int elemen) {
10         this.elemen = elemen;
11         this.keuntungan = new double[elemen];
12     }
13
14     public double totalBF(double[] arr) {
15         double total = 0;
16         for (int i = 0; i < elemen; i++){
17             total += arr[i];
18         }
19         return total;
20     }
21
22     public double totalDC(double[] arr, int start, int end) {
23         if (start == end) {
24             return arr[start];
25         } else {
26             int mid = (start + end) / 2;
27             double leftTotal = totalDC(arr, start, mid);
28             double rightTotal = totalDC(arr, mid + 1, end);
29             return leftTotal + rightTotal;
30         }
31     }
32 }
```

```

1  import java.util.Scanner;
2  /**
3   * MainSum_04
4   */
5  public class MainSum_04 {
6
7      public static void main(String[] args) {
8          Scanner sc04 = new Scanner(System.in);
9          System.out.println(x:"=====");
10         System.out.println(x:"Program Menghitung Keuntungan Total (Satuan Juta, Misal 5.9)");
11         System.out.print(s:"Masukkan Jumlah Bulan : ");
12         int elm = sc04.nextInt();
13         int elm = MainSum_04.main(String[])
14
15         Sum_04 sm = new Sum_04(elm);
16         System.out.println(x:"=====");
17         for (int i = 0; i < sm.elemen; i++){
18             System.out.print("Masukkan untung bulan ke - " + (i+1) + " = ");
19             sm.keuntungan[i] = sc04.nextDouble();
20         }
21
22         System.out.println(x:"=====");
23         System.out.println(x:"Algoritma Brute Force");
24         System.out.printf("Total keuntungan perusahaan selama " + "Kd bulan adalah %10.2f\n", sm.elemen, sm.totalBF(sm.keuntungan));
25         System.out.println(x:"Algoritma Divide Conquer");
26         System.out.printf("Total keuntungan perusahaan selama " + "Kd bulan adalah %10.2f\n", sm.elemen, sm.totalDC(sm.keuntungan, start:0, sm.elemen-1));
27     }
28 }

```

(Maaf bu untuk screenshot nya tidak sama karena codesnap nya tiba-tiba tidak bisa 🙏)

```

=====
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      40.34
=====
Algoritma Divide Conquer
Total keuntungan perusahaan selama 5 bulan adalah      40.34
C:\Users\user\Documents\ASD Jobsheet 5>

```

3. Mengapa terdapat formulasi return value berikut?Jelaskan!

```

return lsum+rsum+arr[mid];

```

Formulasi return tersebut merupakan combine dari Divide and Conquer dimana pada program tersebut mengkalkulasikan semua hasil dari proses sebelumnya untuk mendapatkan hasil akhir yang diinginkan

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

Karena variable mid merupakan pembatas pada saat pembagian atau divide proses, sehingga proses perhitungan akan dilakukan sesuai dengan per bagian yang telah dibagi.

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!

Dengan menambahkan input untuk memasukkan jumlah perusahaan dan menambahkan array objek untuk perusahaan yang berbeda

```

1  /**
2   * Sum_04
3   */
4  public class Sum_04 {
5
6      public int elemen;
7      public double keuntungan [];
8
9      public Sum_04 (int elemen) {
10         this.elemen = elemen;
11         this.keuntungan = new double[elemen];
12     }
13
14     public double totalBF(double[] arr) {
15         double total = 0;
16         for (int i = 0; i < arr.length; i++){
17             total += arr[i];
18         }
19         return total;
20     }
21
22     public double totalDC(double[] arr, int start, int end) {
23         if (start == end) {
24             return arr[start];
25         } else {
26             int mid = (start + end) / 2;
27             double leftTotal = totalDC(arr, start, mid);
28             double rightTotal = totalDC(arr, mid + 1, end);
29             return leftTotal + rightTotal;
30         }
31     }
32 }

```

```

1  import java.util.Scanner;
2  /**
3   * MainSum_04
4   */
5  public class MainSum_04 {
6
7      public static void main(String[] args) {
8          Scanner sc04 = new Scanner(System.in);
9          System.out.println("=====");
10         System.out.println("Program Menghitung Keuntungan Total (Satuan Juta, Misal 5.9)");
11         System.out.print("Masukkan Jumlah Perusahaan : ");
12         int jumlahPerusahaan = sc04.nextInt();
13         Sum_04 sm[] = new Sum_04[jumlahPerusahaan];
14         for (int i = 0; i < jumlahPerusahaan; i++){
15             System.out.print("Masukkan jumlah bulan perusahaan ke-" + (i+1) + " = ");
16             int elm = sc04.nextInt();
17             sm[i] = new Sum_04(elm);
18         }
19         System.out.println("=====");
20         for (int i = 0; i < sm.length; i++){
21             System.out.println("Perusahaan ke-" + (i+1));
22             for (int j = 0; j < sm[i].elemen; j++){
23                 System.out.print("Masukkan untung bulan ke - " + (j+1) + " = ");
24                 sm[i].keuntungan[j] = sc04.nextDouble();
25             }
26         }
27         for (int i = 0; i < sm.length; i++) {
28             System.out.println("\nPerusahaan ke-" + (i+1));
29             System.out.println("=====");
30             System.out.println("Algoritma Brute Force");
31             System.out.printf("Total keuntungan perusahaan selama " + "%d bulan adalah %1.2f\n", sm[i].elemen, sm[i].totalBF(sm[i].keuntungan));
32             System.out.println("=====");
33             System.out.println("Algoritma Divide Conquer");
34             System.out.printf("Total keuntungan perusahaan selama " + "%d bulan adalah %1.2f\n", sm[i].elemen, sm[i].totalDC(sm[i].keuntungan, 0, sm[i].elemen-1));
35             System.out.println("=====");
36         }
37     }
38 }
39

```

Berikut merupakan hasil running dari kode program di atas

```
=====
Program Menghitung Keuntungan Total (Satuan Juta, Misal 5.9)
Masukkan Jumlah Perusahaan : 2
Masukkan jumlah bulan perusahaan ke-1 = 3
Masukkan jumlah bulan perusahaan ke-2 = 2
=====
Perusahaan ke-1
Masukkan untung bulan ke - 1 = 8.5
Masukkan untung bulan ke - 2 = 9.54
Masukkan untung bulan ke - 3 = 7.2
Perusahaan ke-2
Masukkan untung bulan ke - 1 = 9.1
Masukkan untung bulan ke - 2 = 6
=====
Perusahaan ke-1
=====
Algoritma Brute Force
Total keuntungan perusahaan selama 3 bulan adalah 25.24
=====
Algoritma Divide Conquer
Total keuntungan perusahaan selama 3 bulan adalah 25.24
=====
Perusahaan ke-2
=====
Algoritma Brute Force
Total keuntungan perusahaan selama 2 bulan adalah 15.10
=====
Algoritma Divide Conquer
Total keuntungan perusahaan selama 2 bulan adalah 15.10
=====
C:\Users\user\Documents\ASD Jobsheet 5>
```

4.5 Latihan Praktikum

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.

Berikut merupakan kode program untuk menghitung nilai akar dengan BF dan DC dengan menginputkan angka kuadrat sempurna dan tidak sempurna.

```
1  /**
2   * NilaiAkar_04
3   */
4  public class NilaiAkar_04 {
5      public int nilai;
6
7      public int akarPangkatDuaBF(int bilangan) {
8          int akar = 0;
9          while (akar * akar <= bilangan) {
10             akar++;
11         }
12         return akar - 1;
13     }
14
15     public int akarPangkatDuaDC(int bilangan, int min, int max) {
16         if (max < min) {
17             return min - 1;
18         }
19         int mid = (min + max) / 2;
20         if (mid * mid == bilangan) {
21             return mid;
22         } else if (mid * mid < bilangan) {
23             return akarPangkatDuaDC(bilangan, mid + 1, max);
24         } else {
25             return akarPangkatDuaDC(bilangan, min, mid - 1);
26         }
27     }
28 }
```

```

1 import java.util.Scanner;
2 /**
3  * MainNilaiAkar_04
4  */
5 public class MainNilaiAkar_04 {
6
7     public static void main(String[] args) {
8         Scanner sc04 = new Scanner(System.in);
9         System.out.println("=====");
10        System.out.println("\t\tNILAI AKAR PANGKAT");
11        System.out.println("=====");
12        System.out.print("Masukkan jumlah bilangan yang ingin dihitung akarnya: ");
13        int jumlahBilangan = sc04.nextInt();
14
15        NilaiAkar_04[] akar = new NilaiAkar_04[jumlahBilangan];
16        for (int i = 0; i < jumlahBilangan; i++) {
17            akar[i] = new NilaiAkar_04();
18            System.out.print("Masukkan bilangan ke-" + (i + 1) + ": ");
19            akar[i].nilai = sc04.nextInt();
20        }
21
22        System.out.println("=====");
23        System.out.println("Hasil Akar Pangkat Dua dengan Brute Force");
24        for (int i = 0; i < jumlahBilangan; i++) {
25            System.out.println("Akar pangkat dua dari " + akar[i].nilai + " adalah: " + akar[i].akarPangkatDuaBF(akar[i].nilai));
26        }
27
28        System.out.println("=====");
29        System.out.println("Hasil Akar Pangkat Dua dengan Divide and Conquer");
30        for (int i = 0; i < jumlahBilangan; i++) {
31            System.out.println("Akar pangkat dua dari " + akar[i].nilai + " adalah: " + akar[i].akarPangkatDuaDC(akar[i].nilai, 0, akar[i].nilai));
32        }
33        System.out.println("=====");
34    }
35 }

```

Berikut merupakan hasil running dari kode program di atas

```

=====
                        NILAI AKAR PANGKAT
=====
Masukkan jumlah bilangan yang ingin dihitung akarnya: 5
Masukkan bilangan ke-1: 2
Masukkan bilangan ke-2: 7
Masukkan bilangan ke-3: 11
Masukkan bilangan ke-4: 8
Masukkan bilangan ke-5: 10
=====
Hasil Akar Pangkat Dua dengan Brute Force
Akar pangkat dua dari 2 adalah: 1
Akar pangkat dua dari 7 adalah: 2
Akar pangkat dua dari 11 adalah: 3
Akar pangkat dua dari 8 adalah: 2
Akar pangkat dua dari 10 adalah: 3
=====
Hasil Akar Pangkat Dua dengan Divide and Conquer
Akar pangkat dua dari 2 adalah: 1
Akar pangkat dua dari 7 adalah: 2
Akar pangkat dua dari 11 adalah: 3
Akar pangkat dua dari 8 adalah: 2
Akar pangkat dua dari 10 adalah: 3
=====

```

C:\Users\user\Documents\ASD Jobsheet 5>