Name : Azaria Cindy Sahasika
Number Id : 2341760169 / 06
Class : 1G – Business Information System
Lesson : Algorithm and Data Structure
Material : Jobsheet 4
Github Link : https://github.com/azariacindy/algorithm-ds

JOBSHEET IV
BRUTE FORCE DAN DIVIDE CONQUER

4.1 Tujuan Praktikum

Setelah melakukan materi praktikum ini, mahasiswa mampu:

1. Mahasiswa mampu membuat algoritma bruteforce dan divide-conquer
2. Mahasiswa mampu menerapkan penggunaan algorima bruteforce dan divide-conquer

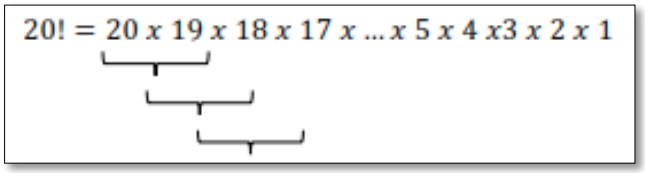
4.2 Menghitung Nilai Faktorial dengan Algoritma Brute Force dan Divide and Conquer

Perhatikan Diagram Class berikut ini :

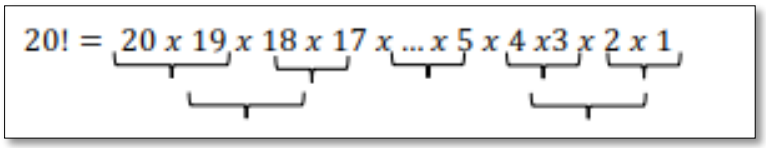
Faktorial
nilai: int
faktorialBF(): int faktorialDC(): int

Berdasarkan diagram class di atas, akan dibuat program class dalam Java. Untuk menghitung nilai faktorial suatu angka menggunakan 2 jenis algoritma, Brute Force dan Divide and Conquer. Jika digambarkan terdapat perbedaan proses perhitungan 2 jenis algoritma tersebut sebagai berikut :

Tahapan pencarian nilai faktorial dengan algoritma Brute Force :



Tahapan pencarian nilai faktorial dengan algoritma Divide and Conquer :



4.2.1 Langkah-langkah Percobaan

1. Buat Project baru, dengan nama "**BruteForceDivideConquer**". Buat package dengan nama minggu5.
2. Buatlah class baru dengan nama **Faktorial**
3. Lengkapi class **Faktorial** dengan atribut dan method yang telah digambarkan di dalam diagram class di atas, sebagai berikut:
   a) Tambahkan atribut nilai

```
public int nilai;
```

   b) Tambahkan method `faktorialBF()` nilai

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

   c) Tambahkan method `faktorialDC()` nilai

```
public int faktorialDC(int n){
    if (n==1) {
        return  1;
    }
    else
    {
        int fakto = n * faktorialDC(n-1);
        return fakto;
    }
}
```

4. Coba jalankan (Run) class `Faktorial` dengan membuat class baru `MainFaktorial`.
   a) Di dalam fungsi main sediakan komunikasi dengan user untuk menginputkan jumlah angka yang akan dicari nilai faktorialnya

```
Scanner sc = new Scanner(System.in);
System.out.println("=====================================================");
System.out.print("Masukkan jumlah elemen yang ingin dihitung : ");
int elemen = sc.nextInt();
```

   b) Buat Array of Objek pada fungsi main, kemudian inputkan beberapa nilai yang akan dihitung faktorialnya

```
Faktorial [] fk = new Faktorial[elemen];
for (int i = 0; i < elemen; i++) {
    fk[i] = new Faktorial();
    System.out.print("Masukkan nilai data ke-"+(i+1)+" : ");
    fk[i].nilai = sc.nextInt();
}
```

c) Tampilkan hasil pemanggilan method `faktorialDC()` dan `faktorialBF()`

```
System.out.println("=====================================================");
System.out.println("Hasil Faktorial dengan Brute Force");
for (int i = 0; i < elemen; i++) {
    System.out.println("Faktorial dari nilai "+fk[i].nilai+" adalah : "+fk[i].faktorialBF(fk[i].nilai));
}
System.out.println("=====================================================");
System.out.println("Hasil Faktorial dengan Divide and Conquer");
for (int i = 0; i < elemen; i++) {
    System.out.println("Faktorial dari nilai "+fk[i].nilai+" adalah : "+fk[i].faktorialDC(fk[i].nilai));
}
System.out.println("=====================================================");
```

d) Pastikan program sudah berjalan dengan baik!

```
week5 > faktorial06.java > ...
Click here to ask Blackbox to help you code faster |
1   package week5;
2
    Comment Code | Improve Code
3   public class faktorial06 {
4       public int nilai;
5
6       public int faktorialBF(int n) {
7           int fakto = 1;
8           for (int i = 1; i <= n; i++) {
9               fakto *= i;
10          }
11          return fakto;
12      }
13
14      public int faktorialDC(int n) {
15          if (n==1) {
16              return 1;
17          } else {
18              int fakto = n * faktorialDC(n-1);
19              return fakto;
20          }
21      }
22  }
```

```
=============================================
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 and Conquer
Faktorial dari nilai 5 adalah: 120
Faktorial dari nilai 8 adalah: 40320
Faktorial dari nilai 3 adalah: 6
=============================================
```

```java
week5 > J faktorialMain06.java > ...
Click here to ask Blackbox to help you code faster |
package week5;

import java.util.Scanner;

Comment Code | Improve Code
public class faktorialMain06 {
    Run | Debug
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print(s:"================================================");
        System.out.println(x:"\nMasukkan jumlah elemen yang ingin dihitung: ");
        int elemen = sc.nextInt();

        faktorial06[] fk = new faktorial06[elemen];
        for (int i = 0; i < elemen; i++) {
            fk[i] = new faktorial06();
            System.out.println("Masukkan nilai data ke-"+ (i+1) +": ");
            fk[i].nilai = sc.nextInt();
        }

        System.out.print(s:"================================================");
        System.out.println(x:"\nHasil faktorial dengan Brute Force");
        for (int i = 0; i < elemen; i++) {
            System.out.printf("Faktorial dari nilai "+ fk[i].nilai +" adalah: "+ fk[i].faktorialBF(fk[i].nilai));
        }

        System.out.print(s:"================================================");
        System.out.println(x:"\nHasil faktorial dengan Divide and conquer");
        for (int i = 0; i < elemen; i++) {
            System.out.println("Faktorial dari nilai "+ fk[i].nilai +" adalah: "+ fk[i].faktorialDC(fk[i].nilai));
        }
        System.out.print(s:"================================================");

        sc.close();
    }
}
```

4.2.2 Verifikasi Hasil Percobaan

Cocokkan hasil compile kode program anda dengan gambar berikut ini.

```
run:
=====================================================
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 and Conquer
Faktorial dari nilai 5 adalah : 120
Faktorial dari nilai 8 adalah : 40320
Faktorial dari nilai 3 adalah : 6
=====================================================
BUILD SUCCESSFUL (total time: 7 seconds)
```

4.2.3 Pertanyaa

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

   ➔ The baseline in the Divide and Conquer algorithm for calculating factorials is the condition where the problem becomes simple enough to solve directly. Here, the baseline is reached when the value of n (input) is equal to 1. Then, the value of the factorial is 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!

   ➔ Yes, the implementation of the Factorial Divide and Conquer algorithm in the complete program code consists of 3 stages:

   a. Divide (pembagian)

      Division is performed on the initial condition (n > 1). The program divides the problem into two smaller subproblems, which is to calculate the factorial n-1 and multiply it by n.

   b. Conquer (penaklukan)

      The conquest is done in a recursive manner. The program calls back the factorial functionDC(n-1) to calculate the value of factorial n-1.

   c. Combine (penggabungan)

      The combine is done by multiplying the result of subproblem n * factorialDC(n-1). The result of this multiplication is the value of factorial n.

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

   ➔ use while loop:

```
week5 > J faktorial06.java > ...
  3     public class faktorial06 {
 13
 14         public int faktorialBF(int n) {
 15             int fakto = 1;
 16             int i = 1;
 17             while (i <= n) {
 18                 fakto *= i;
 19                 i++;
 20             }
 21             return fakto;
 22         }
```

4. Tambahkan pegecekan waktu eksekusi kedua jenis method tersebut!

   ➔ Execution time of the Divide and Conquer method is much faster than the Brute Force method.

```java
Comment Code | Improve Code
public class faktorialMain06 {
    Run | Debug
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print(s:"================================================");
        System.out.println(x:"\nMasukkan jumlah elemen yang ingin dihitung: ");
        int elemen = sc.nextInt();

        faktorial06[] fk = new faktorial06[elemen];
        for (int i = 0; i < elemen; i++) {
            fk[i] = new faktorial06();
            System.out.println("Masukkan nilai data ke-" + (i + 1) + ": ");
            fk[i].nilai = sc.nextInt();
        }

        System.out.println(x:"\n================================================");
        System.out.println(x:"Hasil faktorial dengan Brute Force");
        long startTimeBF = System.currentTimeMillis();
        for (int i = 0; i < elemen; i++) {
            System.out.println("Faktorial dari nilai " + fk[i].nilai + " adalah: " + fk[i].faktorialBF(fk[i].nilai));
        }
        long endTimeBF = System.currentTimeMillis();
        long elapsedTimeBF = endTimeBF - startTimeBF;
        System.out.println("Waktu eksekusi Brute Force: " + elapsedTimeBF + " milliseconds");

        System.out.println(x:"\n================================================");
        System.out.println(x:"Hasil faktorial dengan Divide and Conquer");
        long startTimeDC = System.currentTimeMillis();
        for (int i = 0; i < elemen; i++) {
            System.out.println("Faktorial dari nilai " + fk[i].nilai + " adalah: " + fk[i].faktorialDC(fk[i].nilai));
        }
        long endTimeDC = System.currentTimeMillis();
        long elapsedTimeDC = endTimeDC - startTimeDC;
        System.out.println("Waktu eksekusi Divide and Conquer: " + elapsedTimeDC + " milliseconds");

        System.out.print(s:"================================================");
```

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

```
================================================
Hasil faktorial dengan Brute Force
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 10 adalah: 3628800
Faktorial dari nilai 11 adalah: 39916800
Faktorial dari nilai 12 adalah: 479001600
Faktorial dari nilai 13 adalah: 1932053504
Faktorial dari nilai 14 adalah: 1278945280
Faktorial dari nilai 15 adalah: 2004310016
Faktorial dari nilai 16 adalah: 2004189184
Faktorial dari nilai 17 adalah: -288522240
Faktorial dari nilai 18 adalah: -898433024
Faktorial dari nilai 19 adalah: 109641728
Faktorial dari nilai 20 adalah: -2102132736
Faktorial dari nilai 21 adalah: -1195114496
Faktorial dari nilai 23 adalah: 862453760
Faktorial dari nilai 24 adalah: -775946240
Faktorial dari nilai 25 adalah: 2076180480
Waktu eksekusi Brute Force: 276 milliseconds
================================================
Hasil faktorial dengan Divide and Conquer
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 10 adalah: 3628800
Faktorial dari nilai 11 adalah: 39916800
Faktorial dari nilai 12 adalah: 479001600
Faktorial dari nilai 13 adalah: 1932053504
Faktorial dari nilai 14 adalah: 1278945280
Faktorial dari nilai 15 adalah: 2004310016
Faktorial dari nilai 16 adalah: 2004189184
Faktorial dari nilai 17 adalah: -288522240
Faktorial dari nilai 18 adalah: -898433024
Faktorial dari nilai 19 adalah: 109641728
Faktorial dari nilai 20 adalah: -2102132736
Faktorial dari nilai 21 adalah: -1195114496
Faktorial dari nilai 23 adalah: 862453760
Faktorial dari nilai 24 adalah: -775946240
Faktorial dari nilai 25 adalah: 2076180480
Waktu eksekusi Divide and Conquer: 126 milliseconds
================================================
```

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

Pada praktikum ini kita akan membuat program class dalam Java. Untuk menghitung nilai pangkat suatu angka menggunakan 2 jenis algoritma, Brute Force dan Divide and Conquer.

4.3.1 Langkah-langkah Percobaan

1. Di dalam paket **`minggu5`**, buatlah class baru dengan nama **`Pangkat`**. Dan di dalam class **`Pangkat`** tersebut, buat atribut angka yang akan dipangkatkan sekaligus dengan angka pemangkatnya

```
    public int nilai,pangkat;
```

2. Pada class Pangkat tersebut, tambahkan method `PangkatBF()`

```
public int pangkatBF(int a,int n){
    int hasil=1;
    for (int i = 0; i < n; i++) {
        hasil = hasil * a;
    }
    return hasil;
}
```

3. Pada class Pangkat juga tambahkan method `PangkatDC()`

```
public int pangkatDC(int a,int n){
    if (n==0) {
        return  1;
    }
    else
    {
        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));
    }
}
```

4. Perhatikan apakah sudah tidak ada kesalahan yang muncul dalam pembuatan class Pangkat
5. Selanjutnya buat class baru yang di dalamnya terdapat method main. Class tersebut dapat dinamakan `MainPangkat`. Tambahkan kode pada class main untuk menginputkan jumlah nilai yang akan dihitung pangkatnya.

```
Scanner sc = new Scanner(System.in);
System.out.println("=======================================================");
System.out.print("Masukkan jumlah elemen yang ingin dihitung : ");
int elemen = sc.nextInt();
```

6. Nilai pada tahap 5 selanjutnya digunakan untuk instansiasi array of objek. Di dalam Kode berikut ditambahkan proses pengisian beberapa nilai yang akan dipangkatkan sekaligus dengan pemangkatnya.

```
Pangkat [] png = new Pangkat[elemen];

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

7. Kemudian, panggil hasil nya dengan mengeluarkan return value dari method `PangkatBF()` dan `PangkatDC()`.

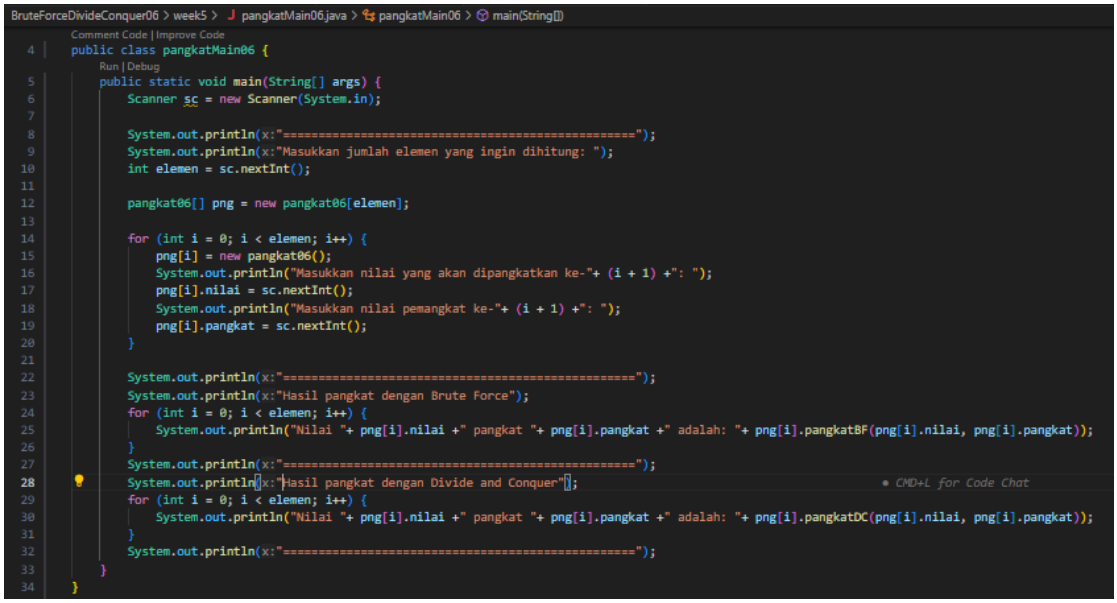
```
        System.out.println("=====================================================");
        System.out.println("Hasil Pangkat dengan Brute Force");
        for (int i = 0; i < elemen; i++) {
            System.out.println("Nilai "+png[i].nilai+" pangkat "+png[i].pangkat+" adalah : "+png[i].pangkatBF(png[i].nilai,png[i].pangkat));
        }
        System.out.println("=====================================================");
        System.out.println("Hasil Pangkat dengan Divide and Conquer");
        for (int i = 0; i < elemen; i++) {
            System.out.println("Nilai "+png[i].nilai+" pangkat "+png[i].pangkat+" adalah : "+png[i].pangkatDC(png[i].nilai,png[i].pangkat));
        }
        System.out.println("=====================================================");
```

4.3.2 Verifikasi Hasil Percobaan

Pastikan output yang ditampilkan sudah benar seperti di bawah ini.

```
run:
=====================================================
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 and Conquer
Nilai 6 pangkat 2 adalah : 36
Nilai 4 pangkat 3 adalah : 64
=====================================================
BUILD SUCCESSFUL (total time: 10 seconds)
```

```
=================================================
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 and Conquer
Nilai 6 pangkat 2 adalah: 36
Nilai 4 pangkat 3 adalah: 64
=================================================
```



```java
BruteForceDivideConquer06 > week5 > pangkatMain06.java > pangkatMain06 > main(String[])
Comment Code | Improve Code
public class pangkatMain06 {
    Run | Debug
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.println(x:"================================================");
        System.out.println(x:"Masukkan jumlah elemen yang ingin dihitung: ");
        int elemen = sc.nextInt();

        pangkat06[] png = new pangkat06[elemen];

        for (int i = 0; i < elemen; i++) {
            png[i] = new pangkat06();
            System.out.println("Masukkan nilai yang akan dipangkatkan ke-"+ (i + 1) +": ");
            png[i].nilai = sc.nextInt();
            System.out.println("Masukkan nilai pemangkat ke-"+ (i + 1) +": ");
            png[i].pangkat = sc.nextInt();
        }

        System.out.println(x:"================================================");
        System.out.println(x:"Hasil pangkat dengan Brute Force");
        for (int i = 0; i < elemen; i++) {
            System.out.println("Nilai "+ png[i].nilai +" pangkat "+ png[i].pangkat +" adalah: "+ png[i].pangkatBF(png[i].nilai, png[i].pangkat));
        }
        System.out.println(x:"================================================");
        System.out.println(x:"Hasil pangkat dengan Divide and Conquer");
        for (int i = 0; i < elemen; i++) {
            System.out.println("Nilai "+ png[i].nilai +" pangkat "+ png[i].pangkat +" adalah: "+ png[i].pangkatDC(png[i].nilai, png[i].pangkat));
        }
        System.out.println(x:"================================================");
    }
}
```

4.3.3 Pertanyaan

1. Jelaskan mengenai perbedaan 2 method yang dibuat yaitu `PangkatBF()` dan `PangkatDC()`!
   - ➔ PangkatBF(): uses the Brute Force method to calculate the power result by multiplying the value to be multiplied by itself as many times as the power value.
   - ➔ PangkatDC(): used the Divide and Conquer method to calculate the power result by dividing the problem into smaller subproblems, then calculating the power of half the power value, and then combining the results.
2. Pada method `PangkatDC()` terdapat potongan program sebagai berikut:

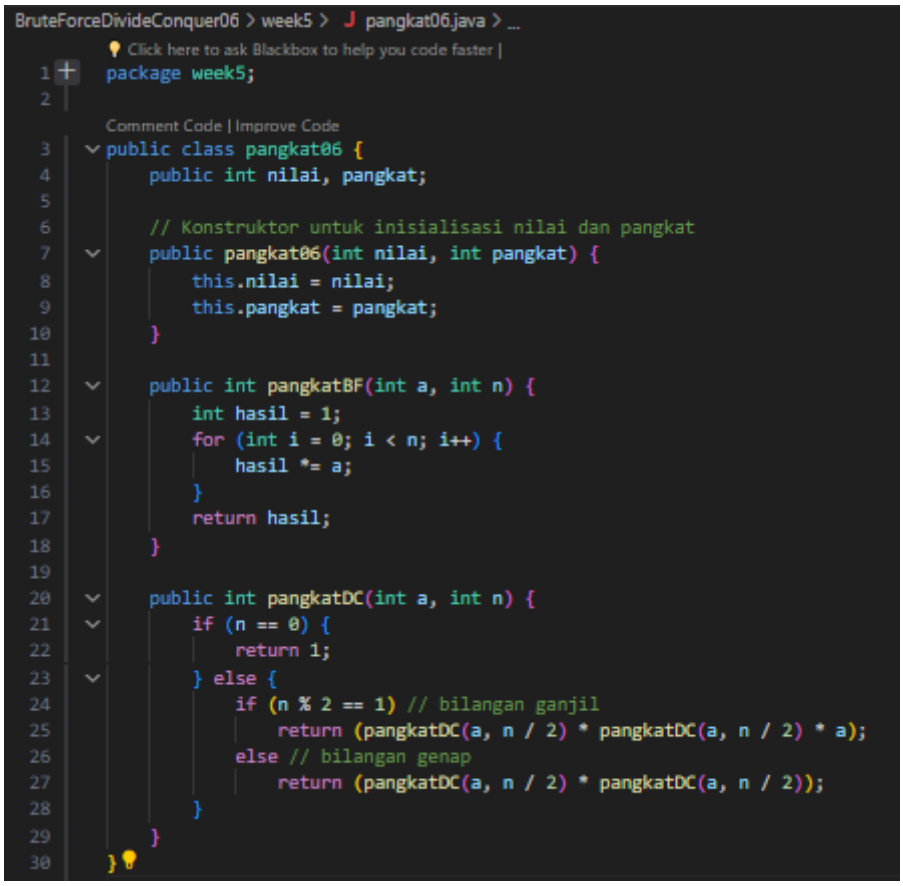
```java
if(n%2==1)//bilangan ganjil
    return (pangkatDC(a,n/2)*pangkatDC(a,n/2)*a);
else//bilangan genap
    return (pangkatDC(a,n/2)*pangkatDC(a,n/2));
```

   Jelaskan arti potongan kode tersebut
   - ➔ 'if (n == 0)': if the power value is 0, then the result is 1 or true.
   - ➔ 'if (n%2 == 1)': if odd, then calculate the result by dividing the power by two and multiply the result by itself and value(a). if odd, then calculate and divide the power by two then multiply the result by itself without value(a).
3. Apakah tahap *combine* sudah termasuk dalam kode tersebut?Tunjukkan!

```java
public int pangkatDC (int a, int n) {
    if (n == 0) {
        return 1;
    } else {
        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));
    }
```

➔ The combine stage is usually not explicitly visible in the code, it usually occurs when combining results from smaller sub-problems to form the final result. The combine stage is implicit in the looping process of the 'pangkatDC()' method.

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



```
BruteForceDivideConquer06 > week5 > pangkat06.java > ...
    Click here to ask Blackbox to help you code faster |
1   package week5;
2
    Comment Code | Improve Code
3   public class pangkat06 {
4       public int nilai, pangkat;
5
6       // Konstruktor untuk inisialisasi nilai dan pangkat
7       public pangkat06(int nilai, int pangkat) {
8           this.nilai = nilai;
9           this.pangkat = pangkat;
10      }
11
12      public int pangkatBF(int a, int n) {
13          int hasil = 1;
14          for (int i = 0; i < n; i++) {
15              hasil *= a;
16          }
17          return hasil;
18      }
19
20      public int pangkatDC(int a, int n) {
21          if (n == 0) {
22              return 1;
23          } else {
24              if (n % 2 == 1) // bilangan ganjil
25                  return (pangkatDC(a, n / 2) * pangkatDC(a, n / 2) * a);
26              else // bilangan genap
27                  return (pangkatDC(a, n / 2) * pangkatDC(a, n / 2));
28          }
29      }
30  }
```

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

BruteForceDivideConquer06 > week5 > pangkatMain06.java > {} week5

```java
public class pangkatMain06 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.println("================================================");
        System.out.println("Masukkan jumlah elemen yang ingin dihitung: ");
        int elemen = sc.nextInt();

        pangkat06[] png = new pangkat06[elemen];

        for (int i = 0; i < elemen; i++) {
            System.out.println("Masukkan nilai yang akan dipangkatkan ke-" + (i + 1) + ": ");
            int nilai = sc.nextInt();
            System.out.println("Masukkan nilai pemangkat ke-" + (i + 1) + ": ");
            int pangkat = sc.nextInt();
            png[i] = new pangkat06(nilai, pangkat);
        }

        System.out.println("================================================");
        System.out.println("Pilih Metode:");
        System.out.println("1. Pangkat dengan Brute Force");
        System.out.println("2. Pangkat dengan Divide and Conquer");
        System.out.print("Pilihan Anda: ");
        int pilihan = sc.nextInt();

        switch (pilihan) {
            case 1:
                System.out.println("Hasil pangkat dengan Brute Force");
                for (int i = 0; i < elemen; i++) {
                    System.out.println("Nilai " + png[i].nilai + " pangkat " + png[i].pangkat + " adalah: "
                            + png[i].pangkatBF(png[i].nilai, png[i].pangkat));
                }
                break;
            case 2:
                System.out.println("Hasil pangkat dengan Divide and Conquer");
                for (int i = 0; i < elemen; i++) {
                    System.out.println("Nilai " + png[i].nilai + " pangkat " + png[i].pangkat + " adalah: "
                            + png[i].pangkatDC(png[i].nilai, png[i].pangkat));
                }
                break;
            default:
                System.out.println("Pilihan tidak valid!");
                break;
        }
```

```
================================================
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:
1. Pangkat dengan Brute Force
2. Pangkat dengan Divide and Conquer
Pilihan Anda: 1
Hasil pangkat dengan Brute Force
Nilai 6 pangkat 2 adalah: 36
Nilai 4 pangkat 3 adalah: 64
================================================
```

```
================================================
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:
1. Pangkat dengan Brute Force
2. Pangkat dengan Divide and Conquer
Pilihan Anda: 2
Hasil pangkat dengan Divide and Conquer
Nilai 6 pangkat 2 adalah: 36
Nilai 4 pangkat 3 adalah: 64
================================================
```

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

Di dalam percobaan ini, kita akan mempraktekkan bagaimana proses *divide, conquer*, dan *combine* diterapkan pada studi kasus penjumlahan keuntungan suatu perusahaan dalam beberapa bulan.

4.4.1 Langkah-langkah Percobaan

1. Pada paket minggu5. Buat class baru yaitu class `Sum`. DI salam class tersebut terdapat beberapa atribut jumlah elemen array, array, dan juga total. Tambahkan pula konstruktor pada class `Sum`.

```
public int elemen;
public double keuntungan[];
public double total;
```

```
Sum(int elemen){
    this.elemen = elemen;
    this.keuntungan=new double[elemen];
    this.total = 0;
}
```

2. Tambahkan method `TotalBF()` yang akan menghitung total nilai array dengan cara *iterative*.

```
double totalBF(double arr[]){
    for (int i = 0; i < elemen; i++) {
        total = total + arr[i];
    }
    return total;
}
```

3. Tambahkan pula method `TotalDC()` untuk implementasi perhitungan nilai total array menggunakan algoritma Divide and Conquer

```
double totalDC(double arr[], int l, int r){
    if(l==r)
        return arr[l];
    else if(l<r){
     int mid=(l+r)/2;
     double lsum=totalDC(arr,l,mid-1);
     double rsum=totalDC(arr,mid+1,r);
     return lsum+rsum+arr[mid];
    }

    return 0;
}
```

4. Buat class baru yaitu `MainSum`. Di dalam kelas ini terdapat method main. Pada method ini user dapat menuliskan berapa bulan keuntungan yang akan dihitung. Dalam kelas ini sekaligus dibuat instansiasi objek untuk memanggil atribut ataupun fungsi pada class `Sum`

```
Scanner sc = new Scanner(System.in);
System.out.println("================================================================");
System.out.println("Program Menghitung Keuntungan Total (Satuan Juta. Misal 5.9)");
System.out.print("Masukkan jumlah bulan : ");
int elm = sc.nextInt();
```

5. Karena yang akan dihitung adalah total nilai keuntungan, maka ditambahkan pula pada method main mana array yang akan dihitung. Array tersebut merupakan atribut yang terdapat di class `Sum`, maka dari itu dibutuhkan pembuatan objek Sum terlebih dahulu.

```
Sum sm = new Sum(elm);
System.out.println("================================================================");
for (int i = 0; i < sm.elemen; i++) {
    System.out.print("Masukkan untung bulan ke - "+(i+1)+" = ");
    sm.keuntungan[i] = sc.nextDouble();
}
```

6. Tampilkan hasil perhitungan melalui objek yang telah dibuat untuk kedua cara yang ada (Brute Force dan Divide and Conquer)

```
System.out.println("============================================================");
System.out.println("Algoritma Brute Force");
System.out.println("Total keuntungan perusahaan selama " + sm.elemen + " bulan adalah = "+sm.totalBF(sm.keuntungan));
System.out.println("============================================================");
System.out.println("Algoritma Divide Conquer");
System.out.println("Total keuntungan perusahaan selama " + sm.elemen + " bulan adalah = "+sm.totalDC(sm.keuntungan, 0,sm.elemen-1));
```

4.4.2 Verifikasi Hasil Percobaan

Cocokkan hasil compile kode program anda dengan gambar berikut ini.

```
run:
================================================================
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
BUILD SUCCESSFUL (total time: 11 seconds)
```

```
================================================
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 perubahan selama 5 bulan adalah: 74.68
```

4.4.3 Pertanyaan

1. Berikan ilustrasi perbedaan perhitungan keuntungan dengan method `TotalBF()` ataupun `TotalDC()`
   - `TotalBF()`: to perform monthly profit increments one by one using looping.
   - `TotalDC()`: to split the profit array into two and calculate the total profit of each part recursively, then combine both parts as well as the profit of the middle element.
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.
   - To limit the output behind a comma, you can use 'String.format'.
3. Mengapa terdapat formulasi *return value* berikut?Jelaskan!

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

   - Used to combine the total gain of the two parts of the array as well as the gain of the center element or combine stage.
4. Kenapa dibutuhkan variable `mid` pada method `TotalDC()`?
   - is used to mark the midpoint when dividing the profit array into two parts in the 'totalDC()' method, in order to divide the subproblems into smaller ones.
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!

BruteForceDivideConquer06 > week5 > companyMain06.java > ...

```java
public class companyMain06 {
    public static void main(String[] args) {

        System.out.println(x:"================================================");
        System.out.print(s:"Masukkan jumlah perusahaan: ");
        int jumlahPerusahaan = sc.nextInt();

        company06[] perusahaan = new company06[jumlahPerusahaan];

        for (int i = 0; i < jumlahPerusahaan; i++) {
            System.out.println(x:"================================================");
            System.out.println("Perusahaan ke-" + (i + 1));
            System.out.print(s:"Nama perusahaan: ");
            String nama = sc.next();
            System.out.print(s:"Jumlah bulan: ");
            int jumlahBulan = sc.nextInt();
            perusahaan[i] = new company06(nama, jumlahBulan);
            for (int j = 0; j < jumlahBulan; j++) {
                System.out.print("Masukkan keuntungan perusahaan untuk bulan ke-" + (j + 1) + ": ");
                double keuntungan = sc.nextDouble();
                perusahaan[i].keuntungan[j] = keuntungan;
            }
        }

        System.out.println(x:"================================================");
        for (company06 p : perusahaan) {
            System.out.println("Total keuntungan perusahaan " + p.nama + " adalah: " + p.totalKeuntungan());
        }
        System.out.println(x:"================================================");
    }
}
```

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.*

```
================================================
Masukkan bilangan: 25
================================================
Akar (Brute Force): 5.0
Akar (Divide and Conquer): 4.0
================================================
```

BruteForceDivideConquer06 > week5 > akar06.java > akar06

```java
public class akar06 {
    public double akarBruteForce(int n) {
        for (int i = 0; i <= n; i++) {
            if (i * i == n) {
                return i;
            }
        }
        return Math.floor(Math.sqrt(n)); // Bulatkan ke bawah jika bukan kuadrat sempurna
    }
                                        • CMD+L for Code Chat
    public double akarDivideConquer(int n) {
        return akarDC(n, start:0, n);
    }

    private double akarDC(int n, double start, double end) {
        if (start <= end) {
            double mid = (start + end) / 2;
            double midSqr = mid * mid;
            if (midSqr == n) {
                return mid;
            } else if (midSqr < n) {
                return akarDC(n, mid + 1, end);
            } else {
                return akarDC(n, start, mid - 1);
            }
        }
        return Math.floor(end); // Bulatkan ke bawah jika bukan kuadrat sempurna
    }
}
```

BruteForceDivideConquer06 > week5 > akarMain06.java > _

Click here to ask Blackbox to help you code faster |

```java
package week5;
import java.util.Scanner;

Comment Code | Improve Code
public class akarMain06 {
    Run | Debug
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        akar06 akar = new akar06();

        System.out.println(x:"================================================");
        System.out.print(s:"Masukkan bilangan: ");
        int bilangan = sc.nextInt();

        System.out.println(x:"================================================");
        System.out.println("Akar (Brute Force): " + akar.akarBruteForce(bilangan));
        System.out.println("Akar (Divide and Conquer): " + akar.akarDivideConquer(bilangan));
        System.out.println(x:"================================================");
    }
}
```