LAPORAN HASIL PRAKTIKUM
ALGORITMA DAN STRUKTUR DATA
JOBSHEET 5

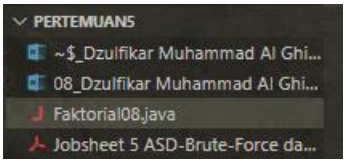
Oleh:
DZULFIKAR MUHAMMAD AL GHIFARI
NIM. 2341760071
SIB-1F / 08
D-IV SISTEM INFORMASI BISNIS
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG

PRAKTIKUM 5

4.2 PERCOBAAN

4.2.1 LANGKAH LANGKAH PERCOBAAN

1. Buatlah class baru dengan nama Faktorial



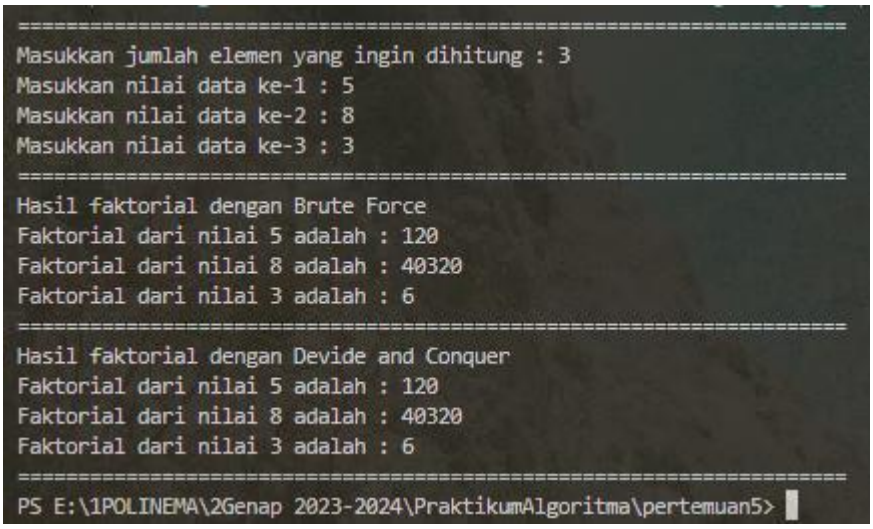
2. Lengkapi class Faktorial dengan atribut dan method

```
Codeium: Refactor | Codeium: Explain
public class Faktorial08 {
    public int nilai;

    Codeium: Refactor | Explain | Generate Javadoc | ×
    public int faktorialBF(int n) {
        int fakto = 1;
        for (int i = 1; i <= n; i++) {
            fakto = fakto * i;
        }
        return fakto;
    }

    Codeium: Refactor | Explain | Generate Javadoc | ×
    public int faktorialDC (int n) {
        if (n == 1) {
            return 1;
        }else{
            int fakto = n * faktorialDC (n-1) ;
            return fakto;
        }
    }
}
```

3. run MainFaktorial

```
import java.util.Scanner;

Codeium: Refactor | Codeium: Explain
public class MainFaktorial08 {
    Run | Debug | Codeium: Refactor | Explain | Generate Javadoc | ×
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

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

        Faktorial08 [] fk = new Faktorial08[elemen];
        for(int i=0; i<elemen; i++){
            fk[i] = new Faktorial08();
            System.out.print("Masukkan nilai data ke-"+(i+1)+ " : ");
            fk[i].nilai = sc.nextInt();
        }

        System.out.println(x:"=================================================================");
        System.out.println(x:"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(x:"=================================================================");
        System.out.println(x:"Hasil faktorial dengan Devide 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(x:"=================================================================");
    }
}
```

4.2.2 VERIFIKASI PERCOBAAN



4.2.3 PERTANYAAN

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

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

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

4. Tambahkan pegecekan waktu eksekusi kedua jenis method tersebut!

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

Jawaban

1. Yang pertama nilai yang di kirimkan akan di validasi apakah sama dengan 1 atau tidak, jika sama dengan satu maka fungsi akan mengembalikan nilai dengan value 1. Jika tidak makanilai akan dikalikan dengan pemanggilan fungsi dengan parameter -1
   N=3
   3-faktorialDC(2)
   2-faktorialDC(1)
   1
2. Didive yaitu

```
if (n == 1) {
    return 1;
```

   Conquer yaitu

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

   Combine tidak ada

3. Memungkinkan menggunakan perulangan while

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

4.

```java
System.out.println(x:"============================================================");
System.out.println(x:"Hasil faktorial dengan Brute Force");
long startTimeBF = System.nanoTime();
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.nanoTime();
System.out.println("Brute Force execution time for n=" + elemen + ": " + (endTimeBF - startTimeBF) + " nanoseconds");

System.out.println(x:"============================================================");
System.out.println(x:"Hasil faktorial dengan Devide and Conquer");

long startTimeDC = System.nanoTime();
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.nanoTime();
System.out.println("Devide and Conquer execution time for n=" + elemen + ": " + (endTimeDC - startTimeDC) + " nanoseconds");

System.out.println(x:"============================================================");
}
```

5. Didive and Conquer mengeksekusi 2x lebih cepat dibandingkan BruteForce

```
Faktorial dari nilai 37 adalah : 0
Faktorial dari nilai 39 adalah : 0
Faktorial dari nilai 41 adalah : 0
Brute Force execution time for n=20: 7160600 nanoseconds
============================================================
Faktorial dari nilai 37 adalah : 0
Faktorial dari nilai 39 adalah : 0
Faktorial dari nilai 41 adalah : 0
Devide and Conquer execution time for n=20: 3633400 nanoseconds
============================================================
PS E:\1POLINEMA\2Genap 2023-2024\PraktikumAlgoritma\pertemuan5>
```

4.3 MENGHITUNG HASIL PANGKAT DENGAN ALGORITMA BRUTE FORCE DAN DIVIDE AND CONQUER

4.3.1 LANGKAH LANGKAH PERCOBAAN

1. Buatlah class baru dengan nama Pangkat

2. Menambahkan method pangkatBF

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

3. Menambahkan method pangkatDC

```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) );
        }
    }
}
```

4. Menambahkan class MainPangkat

```java
Codeium: Refactor | Codeium: Explain
public class MainPangkat08 {
    Run | Debug | Codeium: Refactor | Explain | Generate Javadoc | ×
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

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

        Pangkat08 [] png = new Pangkat08[elemen];

        for(int i=0; i<elemen; i++){
            png[i] = new Pangkat08();
            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();
        }

    }
}
```

6. Pemanggilan method pangkatBF dan pangkatDC

```
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:"================================================================");
System.out.println(x:"Hasil Pangkat dengan Didive 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:"================================================================");
```

4.3.2 VERIFIKASI PERCOBAAN

```
================================================================
Masukkan jumlah elemen yang ingin dihitung : 2
Masukkan nilai yang akan dipangkatkan ke-1 : 2
Masukkan nilai pemangkat ke-1 : 2
Masukkan nilai yang akan dipangkatkan ke-2 : 3
Masukkan nilai pemangkat ke-2 : 2
================================================================
Hasil Pangkat dengan Brute Force
Nilai 2 pangkat 2 adalah : 4
Nilai 3 pangkat 2 adalah : 9
================================================================
================================================================
Hasil Pangkat dengan Didive Conquer
Nilai 2 pangkat 2 adalah : 4
Nilai 3 pangkat 2 adalah : 9
================================================================
```

4.2.3 PERTANYAAN

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

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

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

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

Konstruktor

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

Jawaban

1. method Brute Force menggunakan perulangan secara runtut dan di setiap literasinya akan mengkalikan hasil bilangan dengan pangkat. Method Divide Conquer akan menggunakan medote secara rekursif, dengan memberi validasi di awal apakah pangkat bernilai 0, jika ya maka akan mengembalikan 1, jika tidak maka akan memvalidasi apakah bilangan ganjil atau tidak, kemudian akan melakukan rekursif

2.di awal akan Memvalidasi bilangan apakah ganjil atau tidak, kemudian jika ganjil maka fungsi menghitung pangkat akan dikalikan dengan fungsi mengitung pangkat dengan imbuhan di akhir dikalikan dengan nilai. Jika genap maka akan menjalankan fungsi menghitung pangkat dikalikan dengan fungsi menghitung pangkat.

3.
```
return (pangkatDC(a, n/2) *pangkatDC (a,n/2) *a) ;
```

4.

```
Codeium: Refactor | Explain | Generate Javadoc | ×
public int pangkatBF(int a, int n) {
    int hasil = 1;
    for (int i = 0; i < n; i++) {
        hasil = hasil * a;
    }
    return hasil;
}
```

```
Pangkat08[] pangkatArr = new Pangkat08[elemen];

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

    pangkatArr[i] = new Pangkat08(nilai, pangkat);
}

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

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 " + pangkatArr[i].nilai + " pangkat " + pangkatArr[i].pangkat + " adalah : " + pangkatArr[i].pangkatDC(pangkatArr[i].pangkat, pangkatArr[i].nilai));
}

System.out.println(x:"==========================================================================");
}
```

```
==========================================================================
Masukkan jumlah elemen yang ingin dihitung : 2
Masukkan nilai yang akan dipangkatkan ke-1 : 2
Masukkan nilai pemangkat ke-1 : 2
Masukkan nilai yang akan dipangkatkan ke-2 : 4
Masukkan nilai pemangkat ke-2 : 4
==========================================================================
Hasil Pangkat dengan Brute Force
Nilai 2 pangkat 2 adalah : 4
Nilai 4 pangkat 4 adalah : 256
==========================================================================
Hasil Pangkat dengan Divide and Conquer
Nilai 2 pangkat 2 adalah : 4
Nilai 4 pangkat 4 adalah : 256
==========================================================================
PS E:\1POLINEMA\2Genap 2023-2024\PraktikumAlgoritma\pertemuan5>
```

5.

```
    System.out.println(x:"=================================================================");
    System.out.println(x:"Menu Perhitungan Pangkat");
    System.out.println(x:"=================================================================");

    System.out.println(x:"1. Brute Force");
    System.out.println(x:"2. Divide and Conquer");
    System.out.println(x:"3. Keluar");
    System.out.println(x:"-----------------------------------------------------------------");
    System.out.print(s:"Masukkan pilihan Anda (1/2/3): ");
    pilihan = sc.nextInt();

    // Switch case untuk memilih metode
    switch (pilihan) {
        case 1:
                System.out.println(x:"=================================================================");
                System.out.println(x:"Hasil Pangkat dengan Brute Force");
                for (int i = 0; i < elemen; i++) {
                    System.out.println("Nilai " + pangkatArr[i].nilai + " pangkat " + pangkatArr[i].pangkat + " adalah : " + pangkatArr[i].pangkatBF(pangkatArr[i].pangkat, pangkatArr[i].nilai));
                };
            break;
        case 2:
                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 " + pangkatArr[i].nilai + " pangkat " + pangkatArr[i].pangkat + " adalah : " + pangkatArr[i].pangkatDC(pangkatArr[i].pangkat, pangkatArr[i].nilai));
                };
            break;
        case 3:
            return;
        default:
            System.out.println(x:"Pilihan tidak valid!");
    }
```

```
=================================================================
Masukkan jumlah elemen yang ingin dihitung : 2
Masukkan nilai yang akan dipangkatkan ke-1 : 2
Masukkan nilai pemangkat ke-1 : 3
Masukkan nilai yang akan dipangkatkan ke-2 : 2
Masukkan nilai pemangkat ke-2 : 4
=================================================================
Menu Perhitungan Pangkat
-----------------------------------------------------------------
1. Brute Force
2. Divide and Conquer
3. Keluar
-----------------------------------------------------------------
Masukkan pilihan Anda (1/2/0): 1
=================================================================
Hasil Pangkat dengan Brute Force
Nilai 2 pangkat 3 adalah : 9
Nilai 2 pangkat 4 adalah : 16
-----------------------------------------------------------------
```

4.4 MENGHITUNG SUM ARRAY DENGAN ALGORITMA BRUTE FORCE DAN DIVIDE AND CONQUER

4.4.1 LANGKAH LANGKAH PERCOBAAN

1. Membuat class sum, menambahkan elemen, menambahkan constructor

```java
public class Sum08 {
    public int elemen;
    public double keuntungan[];
    public double total;

    Sum08(int elemen){
        this.elemen = elemen;
        this.keuntungan = new double[elemen];
        this.total = 0;
    }
}
```

2. Menambahkan method totalBF

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

3. Menambahkan method totalDC

```java
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. Menambahkan mainSum

```java
import java.util.Scanner;

Codeium: Refactor | Codeium: Explain
public class MainSum08 {
    Run | Debug | Codeium: Refactor | Explain | Generate Javadoc | ×
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.println(x:"================================================================");
        System.out.println(x:"Program Menghitung Keuntungan Total (Satuan Juta. Misal 5,9)");
        System.out.print(s:"Masukkan jumlah bulan : ");
        int elm = sc.nextInt();

    }
}
```

5. Membuat objek sum

```java
Sum08 sm = new Sum08(elm);
System.out.println(x:"================================================================");
for(int i=0; i>elm; i++){
    System.out.print("Masukkan untung bulan ke-"+ (i+1)+" : ");
    sm.keuntungan[i] = sc.nextDouble();
}
```

6. Mnampilkan hasil Brute Force dan Didice Connquer

```
System.out.println(x:"=================================================================");
System.out.println(x:"Algoritma Brute Force");
System.out.println("Total keuntungan perusahaan selama "+ sm.elemen+" bulan adalah : "+ sm.totalBF(sm.keuntungan));
System.out.println(x:"=================================================================");

System.out.println(x:"=================================================================");
System.out.println(x:"Algoritma Didive Conquer");
System.out.println("Total keuntungan perusahaan selama "+ sm.elemen+" bulan adalah : "+ sm.totalDC(sm.keuntungan, 1:0, sm.elemen-1));
System.out.println(x:"=================================================================");
```

4.4.2 VERIFIKASI PERCOBAAN

```
=================================================================
Program Menghitung Keuntungan Total (Satuan Juta. Misal 5,9)
Masukkan jumlah bulan : 3
=================================================================
Masukkan untung bulan ke-1 : 4
Masukkan untung bulan ke-2 : 5.5
Masukkan untung bulan ke-3 : 6
=================================================================
Algoritma Brute Force
Total keuntungan perusahaan selama 3 bulan adalah : 15.5
=================================================================
=================================================================
Algoritma Didive Conquer
Total keuntungan perusahaan selama 3 bulan adalah : 15.5
=================================================================
```

4.4.3 PERTANYAAN

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

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];
```

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

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!

Jawaban

1. method totalBF menggunakan perulangan dan di setiap literasi akan menambahkan total dari tiap tiap array key ke var total, sehingga setelah proses perulangan var total berisi total dari value array. Method totalDC memanfaatkan fungsi rekursif, dengan terlebih dahulu memberi validasi apakah var l sama dengan var r, jika sama maka akan mengembalikan value array key yang

sama dengan var l. Kemudian jika tidak maka akan melakukan validasi lagi apakah l < r, jika tidak maka akan mengembalikan 0, jika yam aka akan melakukan kalkulasi rekursif hingga selesai, setelah itu akan mengembalikan hasil dari kalkulasi.

2.

```
    System.out.println(x:"================================================================");
    System.out.println(x:"Algoritma Brute Force");
    System.out.println("Total keuntungan perusahaan selama "+ sm.elemen+" bulan adalah : "+ String.format(format:"%.2f",sm.totalBF(sm.keuntungan)));
    System.out.println(x:"================================================================");

    System.out.println(x:"================================================================");
    System.out.println(x:"Algoritma Didive Conquer");
    System.out.println("Total keuntungan perusahaan selama "+ sm.elemen+" bulan adalah : "+ String.format(format:"%.2f", sm.totalDC(sm.keuntungan, l:0, sm.elemen - 1)));
    System.out.println(x:"================================================================");
}
```

```
================================================================
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 : 5.7
================================================================
Algoritma Brute Force
Total keuntungan perusahaan selama 2 bulan adalah : 10.20
================================================================
================================================================
Algoritma Didive Conquer
Total keuntungan perusahaan selama 2 bulan adalah : 10.20
================================================================
PS E:\1POLINEMA\2Genap 2023-2024\PraktikumAlgoritma\pertemuan5>
```

3. untuk menghitung jumlah dari total elemen yang dihitung

4. mid digunakan untuk menampung nilai tengah dan menentukan batas rekursif

5.

```
    System.out.println(x:"================================================================");
    System.out.println(x:"Program Menghitung Keuntungan Total (Satuan Juta. Misal 5,9)");

    System.out.print(s:"Masukkan jumlah bulan : ");
    int elm = sc.nextInt();


    Sum08[] phs = new Sum08[elm];

    for (int i = 0; i < elm; i++) {
        System.out.print("Masukkan jumlah bulan untuk perusahaan ke-" + (i + 1) + " : ");
        int month = sc.nextInt();
        phs[i] = new Sum08(month);

        System.out.println("Masukkan keuntungan untuk perusahaan ke-" + (i + 1));
        for (int j = 0; j < phs[i].elemen; j++) {
            System.out.print("Masukkan untung bulan ke-" + (j + 1) + " : ");
            phs[i].keuntungan[j] = sc.nextDouble();
        }
    }

    System.out.println(x:"============================================================");
    for (int i = 0; i < elm; i++) {
        System.out.println("Perusahaan ke-" + (i + 1));
        System.out.println(x:"Algoritma Brute Force");
        System.out.println("Total keuntungan perusahaan selama " + phs[i].elemen + " bulan adalah = " + phs[i].totalBF(phs[i].keuntungan));
        System.out.println(x:"Algoritma Divide Conquer");
        System.out.println("Total keuntungan perusahaan selama " + phs[i].elemen + " bulan adalah = " + phs[i].totalDC(phs[i].keuntungan, l:0, phs[i].elemen - 1));
        System.out.println(x:"============================================================");
    }
}
```

```
Masukkan keuntungan untuk perusahaan ke-1
Masukkan untung bulan ke-1 : 2
Masukkan untung bulan ke-2 : 2
Masukkan jumlah bulan untuk perusahaan ke-2 : 2
Masukkan keuntungan untuk perusahaan ke-2
Masukkan untung bulan ke-1 : 2
Masukkan untung bulan ke-2 : 2
============================================================
Perusahaan ke-1
Algoritma Brute Force
Total keuntungan perusahaan selama 2 bulan adalah = 4.0
Algoritma Divide Conquer
Total keuntungan perusahaan selama 2 bulan adalah = 4.0
============================================================
Perusahaan ke-2
Algoritma Brute Force
Total keuntungan perusahaan selama 2 bulan adalah = 4.0
Algoritma Divide Conquer
Total keuntungan perusahaan selama 2 bulan adalah = 4.0
============================================================
```

4.4 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.

```java
import java.util.Scanner;

Codeium: Refactor | Codeium: Explain
public class MainAkar08 {
    Run | Debug | Codeium: Refactor | Explain | Generate Javadoc | X
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println(x:"==================================================");
        System.out.print(s:"Masukkan bilangan yang ingin dihitung akarnya : ");
        double num = sc.nextDouble();

        Akar08 akar = new Akar08();
        System.out.println(x:"==================================================");
        System.out.println(x:"Hasil akar dengan Brute Force");
        System.out.println("Akar dari " + num + " adalah : " + akar.akarBF(num));

        System.out.println(x:"==================================================");
        System.out.println(x:"Hasil akar dengan Divide and Conquer");
        System.out.println("Akar dari " + num + " adalah : " + akar.akarDC(num, low:0, num));
    }
}
```

```java
public class Akar08 {
    public double num;

    Codeium: Refactor | Explain | Generate Javadoc | X
    public double akarBF(double num) {
        double low = 0, high = num, mid;
        while (low <= high) {
            mid = low + (high - low) / 2;
            if (mid * mid == num) {
                return mid;
            } else if (mid * mid < num) {
                low = mid + 0.00001;
            } else {
                high = mid - 0.00001;
            }
        }
        return low;
    }

    Codeium: Refactor | Explain | Generate Javadoc | X
    public double akarDC(double num, double low, double high) {
        double mid = low + (high - low) / 2;
        if (high - low < 0.00001) {
            return mid;
        }
        if (mid * mid == num) {
            return mid;
        } else if (mid * mid < num) {
            return akarDC(num, mid, high);
        } else {
            return akarDC(num, low, mid);
        }
    }
}
```