



NAMA : ANNISA KURNIAWATI
NIM : 2341720070
NO ABSEN : 04
KELAS : 1F
MATERI : DASAR PEMROGRAMAN

LAPORAN PRAKTIKUM ALGORITMA DAN STRUKTUR DATA

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

Input :

```
public class Faktorial04 {
    public int nilai;

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

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

public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.println("-----");
    System.out.println("Masukkan jumlah elemen: ");
    int iJml = sc.nextInt();

    Faktorial04 f4 = new Faktorial04();
    for(int i=0; i < iJml; i++){
        f4[i] = new Faktorial04();
        System.out.println("Masukkan nilai ke- " + (i+1) + ":");
        f4[i].nilai = sc.nextInt();
    }

    System.out.println("HASIL - BRUTE FORCE");
    for(int i=0; i < iJml; i++){
        System.out.println(
            "Hasil penghitungan faktorial menggunakan Brute Force adalah "
            + f4[i].faktorialBF(f4[i].nilai));
    }

    System.out.println("HASIL - DIVIDE AND CONQUER");
    for(int i=0; i < iJml; i++){
        System.out.println(
            "Hasil penghitungan faktorial menggunakan Divide and Conquer adalah "
            + f4[i].faktorialDC(f4[i].nilai));
    }
}
```

Output :

```
Masukkan jumlah elemen:
3
masukkan nilai ke-1:
5
masukkan nilai ke-2:
8
masukkan nilai ke-3:
3
HASIL - BRUTE FORCE
Hasil penghitungan faktorial menggunakan Brute Force adalah 120
Hasil penghitungan faktorial menggunakan Brute Force adalah 40320
Hasil penghitungan faktorial menggunakan Brute Force adalah 6
HASIL - DIVIDE AND CONQUER
Hasil penghitungan faktorial menggunakan Divide and Conquer adalah 120
Hasil penghitungan faktorial menggunakan Divide and Conquer adalah 40320
Hasil penghitungan faktorial menggunakan Divide and Conquer adalah 6
```

Question

1. Pada base line Algoritma Divide Conquer untuk melakukan pencarian nilai faktorial, jelaskan perbedaan bagian kode pada penggunaan if dan else!

Answer :

Pada basecase if ini untuk menghentikan looping dan mencegah terjadinya loop tak terbatas. Lalu untuk else ini fungsi rekursif untuk menjalankan pemanggilan untuk mengurangi nilai n-1 secara bertahap.

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

Answer :

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

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



NAMA : ANNISA KURNIAWATI
NIM : 2341720070
NO ABSEN : 04
KELAS : 1F
MATERI : DASAR PEMROGRAMAN

3. Jelaskan perbedaan antara fakto $*= i$; dan int fakto $= n * \text{faktorialDC}(n-1)$; !
Answer :
Perbedaanya jika fakto $*= i$; menggunakan alur brute force atau iteratif jika int fakto $= n * \text{faktorialDC}(n-1)$; menggunakan alur rekursif

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

Input :

```
public int nilai, pangkat;

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

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

```
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.println("=====");
    System.out.print("Masukkan jumlah elemen yang ingin dihitung : ");
    int elemen = sc.nextInt();

    Pangkat04[] png = new Pangkat04[elemen];
    for (int i = 0; i < elemen; i++) {
        png[i] = new Pangkat04();
        System.out.print("Masukkan nilai yang akan dipangkatkan: ");
        png[i].nilai = sc.nextInt();
        System.out.print("Masukkan nilai pemangkat: ");
        png[i].pangkat = sc.nextInt();
    }

    System.out.println("HASIL PANGKAT - BRUTE FORCE");
    for (int i = 0; i < elemen; i++) {
        System.out.println("Hasil dari "
            + png[i].nilai + " pangkat "
            + png[i].pangkat + " adalah "
            + png[i].pangkatBF(png[i].nilai, png[i].pangkat));
    }

    System.out.println("HASIL PANGKAT - DIVIDE AND CONQUER");
    for (int i = 0; i < elemen; i++) {
        System.out.println("Hasil dari "
            + png[i].nilai + " pangkat "
            + png[i].pangkat + " adalah "
            + png[i].pangkatDC(png[i].nilai, png[i].pangkat));
    }
}
```

Output :

```
=====
Masukkan jumlah elemen yang ingin dihitung : 2
Masukkan nilai yang akan dipangkatkan: 6
Masukkan nilai pemangkat: 2
Masukkan nilai yang akan dipangkatkan: 4
Masukkan nilai pemangkat: 3
HASIL PANGKAT - BRUTE FORCE
Hasil dari 6 pangkat 2 adalah 36
Hasil dari 4 pangkat 3 adalah 64
HASIL PANGKAT - DIVIDE AND CONQUER
Hasil dari 6 pangkat 2 adalah 36
Hasil dari 4 pangkat 3 adalah 64
```

Question

1. Jelaskan mengenai perbedaan 2 method yang dibuat yaitu PangkatBF() dan PangkatDC()!
Answer :
Perbedaan keduanya adalah jika PangkatBF() menggunakan alur brute force dengan iteratif menggunakan for, jika PangkatDC() menggunakan alur divide conquer dengan rekursif yaitu membagi masalah menjadi masalah yang lebih sederhana



NAMA : ANNISA KURNIAWATI
NIM : 2341720070
NO ABSEN : 04
KELAS : 1F
MATERI : DASAR PEMROGRAMAN

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

Answer :

Sudah tahap combine terjadi dalam bentuk terpisah tahap "combine" terjadi ketika hasil dari submasalah yang lebih kecil digabungkan kembali untuk membentuk hasil akhir dari masalah yang lebih besar.

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

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

Answer :

```
Pangkat04[] png = new Pangkat04[elemen];  
for (int i = 0; i < elemen; i++) {  
    System.out.print(s: "Masukkan nilai yang akan dipangkatkan: ");  
    int nilai = sc.nextInt();  
    System.out.print(s: "Masukkan nilai pemangkat: ");  
    int pangkat = sc.nextInt();  
    png[i] = new Pangkat04(nilai, pangkat);  
}
```

```
int nilai;  
int pangkat;  
  
public Pangkat04(int nilai, int pangkat) {  
    this.nilai = nilai;  
    this.pangkat = pangkat;  
}
```

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

Answer :

Input :

```
System.out.println(x: "Pilih method yang ingin dijalankan");  
System.out.println(x: "1. PANGKAT - BRUTE FORCE");  
System.out.println(x: "2. HASIL PANGKAT - DIVIDE AND CONQUER");  
System.out.print(s: "Masukkan Pilihan Anda : ");  
int pilihan = sc.nextInt();  
  
switch (pilihan) {  
    case 1:  
        System.out.println(x: "HASIL PANGKAT - BRUTE FORCE");  
        for (int i = 0; i < elemen; i++) {  
            System.out.println("Hasil dari "  
                + png[i].nilai + " pangkat "  
                + png[i].pangkat + " adalah "  
                + png[i].pangkatBF(png[i].nilai, png[i].pangkat));  
        }  
        break;  
    case 2:  
        System.out.println(x: "HASIL PANGKAT - DIVIDE AND CONQUER");  
        for (int i = 0; i < elemen; i++) {  
            System.out.println("Hasil dari "  
                + png[i].nilai + " pangkat "  
                + png[i].pangkat + " adalah "  
                + png[i].pangkatDC(png[i].nilai, png[i].pangkat));  
        }  
        break;  
    default:  
        System.out.println(x: "Pilihan tidak valid");  
}
```

Output :

```
Masukkan jumlah elemen yang ingin dihitung : 3  
Masukkan nilai yang akan dipangkatkan: 4  
Masukkan nilai pemangkat: 2  
Masukkan nilai yang akan dipangkatkan: 3  
Masukkan nilai pemangkat: 2  
Masukkan nilai yang akan dipangkatkan: 3  
Masukkan nilai pemangkat: 2  
Pilih method yang ingin dijalankan
```



NAMA : ANNISA KURNIAWATI
NIM : 2341720070
NO ABSEN : 04
KELAS : 1F
MATERI : DASAR PEMROGRAMAN

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

Input :

```
public static void main(String[] args) {  
    Scanner sc = new Scanner(System.in);  
    System.out.println(x:"=====");  
    System.out.println(x:"Program Menghitung Total (Satuan Juta. Misal 5.9)");  
    System.out.print(s:"Masukkan jumlah bulan: ");  
    int elemen = sc.nextInt();  
  
    Sum04 sm = new Sum04(elemen);  
  
    System.out.println(x:"=====");  
    for(int i = 0; i < sm.elemen; i++){  
        System.out.print("Masukkan bulan ke - " + (i+1) + ": ");  
        sm.keuntungan[i] = sc.nextDouble();  
    }  
    System.out.println(x:"=====");  
    System.out.println(x:"Algoritma Brute Force");  
    System.out.print("Total keuntungan perusahaan selama " + sm.elemen + " bulan adalah = "+sm.totalBF(sm.keuntungan));  
    System.out.println(x:"=====");  
    System.out.println(x:"Algoritma Divide Conquer");  
    System.out.print("Total keuntungan perusahaan selama " + sm.elemen + " bulan adalah = "+sm.totalDC(sm.keuntungan,1,sm.elemen-1));  
}
```

```
double keuntungan[], total;  
  
Sum04(int elemen){  
    this.elemen = elemen;  
    this.keuntungan = new double[elemen];  
    this.total = total;  
  
    this.keuntungan = keuntungan;  
}  
  
double totalBF(double arr[]){  
    double total = 0;  
    for(int i=0; i < arr.length; i++){  
        total = total + arr[i];  
    }  
    return total;  
}  
  
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;  
}
```

Output :

```
=====
Program Menghitung Total (Satuan Juta. Misal 5.9)
Masukkan jumlah bulan: 5
=====
Masukkan bulan ke - 1: 8.5
Masukkan bulan ke - 2: 9.54
Masukkan bulan ke - 3: 7.2
Masukkan bulan ke - 4: 9.1
=====
Algoritma Brute Force
Total keuntungan perusahaan selama 5 bulan adalah = 40.339999999999996
=====
Algoritma Divide Conquer
Total keuntungan perusahaan selama 5 bulan adalah = 40.34
```



NAMA : ANNISA KURNIAWATI
NIM : 2341720070
NO ABSEN : 04
KELAS : 1F
MATERI : DASAR PEMROGRAMAN
Question

1. Mengapa terdapat formulasi return value berikut?Jelaskan!

Answer :

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

Code diatas digunakan untuk memecahkan masalah yaitu jika lsum untuk hasil submasalah di sebelah kiri, rsum untuk hasil submasalah di sebelah kanan, dan arr[mid] adalah nilai dari elemen tengah array

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

Answer :

Karena untuk membantu membagi masalah menjadi submasalah yang lebih kecil saat menerapkan alur divide conquer

3. Program perhitungan keuntungan suatu perusahaan ini hanya untuk satu perusahaan saja. Bagaimana cara menghitung sekaligus keuntungan beberapa bulan untuk beberapa perusahaan.(Setiap perusahaan bisa saja memiliki jumlah bulan berbeda-beda)? Buktikan dengan program!

Answer :

Input :

```
System.out.print(s:"Masukkan jumlah perusahaan: ");
int nPerusahaan = sc.nextInt();

Sum04[] perusahaan = new Sum04[nPerusahaan];

for (int i = 0; i < nPerusahaan; i++) {
    sc.nextLine();
    System.out.println("Data Perusahaan " + (i + 1));
    System.out.print(s:"Masukkan nama perusahaan : ");
    String nama = sc.nextLine();
    System.out.print(s:"Masukkan jumlah bulan : ");
    int nbulan = sc.nextInt();

    perusahaan[i] = new Sum04(nama, nbulan);

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

double totalKeuntungan = 0;
for (Sum04 p : perusahaan) {
    totalKeuntungan += p.hitungTotalKeuntungan();
}

System.out.println(s:"=====");
System.out.println("Total keuntungan dari semua perusahaan: " + totalKeuntungan);
```

```
String nama;
int elemen;
double[] keuntungan;

public Sum04(String nama, int elemen) {
    this.nama = nama;
    this.elemen = elemen;
    keuntungan = new double[elemen];
}

public double hitungTotalKeuntungan() {
    double total = 0;
    for (double k : keuntungan) {
        total += k;
    }
    return total;
}
```

Output : Program Menghitung Keuntungan Perusahaan

```
=====
Masukkan jumlah perusahaan: 2
Data Perusahaan 1
Masukkan nama perusahaan : bri
Masukkan jumlah bulan : 1
Masukkan keuntungan bulan ke-1: 60000
Data Perusahaan 2
Masukkan nama perusahaan : bni
Masukkan jumlah bulan : 1
Masukkan keuntungan bulan ke-1: 70000
=====
Total keuntungan dari semua perusahaan: 130000.0
```



NAMA : ANNISA KURNIAWATI
NIM : 2341720070
NO ABSEN : 04
KELAS : 1F
MATERI : DASAR PEMROGRAMAN

Input :

```
public class Showroom04 {  
    String merk;  
    String tipe;  
    int tahun;  
    int topAcceleration;  
    int topPower;  
  
    public Showroom04(String merk, String tipe, int tahun, int topAcceleration, int topPower){  
        this.merk = merk;  
        this.tipe = tipe;  
        this.tahun = tahun;  
        this.topAcceleration = topAcceleration;  
        this.topPower = topPower;  
    }  
}
```

```
public static void main(String[] args) {  
    Showroom04[] mobilist = {  
        new Showroom04(merk:"BMW", tipe:"M2 Coupe", tahun:2016, topAcceleration:6816, topPower:728),  
        new Showroom04(merk:"Ford", tipe:"Fiesta ST", tahun:2014, topAcceleration:3921, topPower:575),  
        new Showroom04(merk:"Nissan", tipe:"370Z", tahun:2009, topAcceleration:4360, topPower:657),  
        new Showroom04(merk:"Subaru", tipe:"BRZ", tahun:2016, topAcceleration:4058, topPower:609),  
        new Showroom04(merk:"Subaru", tipe:"Impreza WRX STI", tahun:2013, topAcceleration:6255, topPower:703),  
        new Showroom04(merk:"Toyota", tipe:"A86 Trueno", tahun:1986, topAcceleration:3700, topPower:553),  
        new Showroom04(merk:"Toyota", tipe:"86/GT86", tahun:2014, topAcceleration:4180, topPower:609),  
        new Showroom04(merk:"Volkswagen", tipe:"Golf GTI", tahun:2014, topAcceleration:4180, topPower:631)  
    };  
  
    int topAcceleration = findMaxAcceleration(mobilist, low:0, mobilist.length - 1);  
    System.out.println("Top Acceleration Tertinggi Dengan Divide Conquer: " + topAcceleration);  
  
    int minAcceleration = findMinAcceleration(mobilist, low:0, mobilist.length - 1);  
    System.out.println("Top Acceleration Terendah Dengan Divide Conquer: " + minAcceleration);  
  
    double averagePower = calculateAveragePower(mobilist);  
    System.out.println("Rata-rata Top Power dari Seluruh Mobil Dengan Brute Force: " + averagePower);  
}  
  
static int findMaxAcceleration(Showroom04[] arr, int low, int high) {  
    if (low == high) {  
        return arr[low].topAcceleration;  
    } else if (high == low + 1) {  
        return Math.max(arr[low].topAcceleration, arr[high].topAcceleration);  
    }  
}
```

```
    } else {  
        int mid = (low + high) / 2;  
        int leftMax = findMaxAcceleration(arr, low, mid);  
        int rightMax = findMaxAcceleration(arr, mid + 1, high);  
        return Math.max(leftMax, rightMax);  
    }  
}  
  
static int findMinAcceleration(Showroom04[] arr, int low, int high) {  
    if (low == high) {  
        return arr[low].topAcceleration;  
    } else if (high == low + 1) {  
        return Math.min(arr[low].topAcceleration, arr[high].topAcceleration);  
    } else {  
        int mid = (low + high) / 2;  
        int leftMin = findMinAcceleration(arr, low, mid);  
        int rightMin = findMinAcceleration(arr, mid + 1, high);  
        return Math.min(leftMin, rightMin);  
    }  
}  
  
static double calculateAveragePower(Showroom04[] arr) {  
    int totalPower = 0;  
    for (Showroom04 mobil : arr) {  
        totalPower += mobil.topPower;  
    }  
    return (double) totalPower / arr.length;  
}
```

Output :

```
Top Acceleration Tertinggi Dengan Divide Conquer: 6816  
Top Acceleration Terendah Divide Conquer: 3700  
Rata-rata Top Power dari Seluruh Mobil: 633.125
```