

LAPORAN PRAKTIKUM ALGORITMA DAN STRUKTUR DATA



2022

Praktikan

2141720183

RIDWAN CAESAR RIZQI KARISMA BIWARNI

TI 1C

Daftar Isi [harus diupdate setiap ada perubahan]

MENGHITUNG NILAI FAKTORIAL DENGAN ALGORITMA BRUTE FORCE DAN DEVIDE CONQUER.....	4
LANGKAH 1.....	4
LANGKAH 2.....	4
LANGKAH 3.....	4
LANGKAH 4.....	5
VERIFIKASI HASIL PERCOBAAN	6
<i>Pertanyaan</i>	6
<i>Jawaban</i>	7
MENGHITUNG HASIL PANGKAT DENGAN ALGORITMA BRUTE FORCE DAN DIVIDE AND CONQUER	10
LANGKAH 1.....	10
LANGKAH 2.....	10
LANGKAH 3.....	10
LANGKAH 4.....	10
LANGKAH 5.....	11
LANGKAH 6.....	11
LANGKAH 7.....	11
VERIFIKASI HASIL PERCOBAAN	11
<i>Pertanyaan</i>	12
<i>Jawaban</i>	12
MENGHITUNG SUM ARRAY DENGAN ALGORITMA BRUTE FORCE DAN DIVIDE CONQUER	14
LANGKAH 1.....	14
LANGKAH 2.....	14
LANGKAH 3.....	14
LANGKAH 4.....	15
LANGKAH 5.....	15
LANGKAH 6.....	15
VERIFIKASI HASIL PERCOBAAN	16
<i>Pertanyaan</i>	16
<i>Jawaban</i>	16
LATIHAN PRAKTIKUM	19
JAWABAN LATIHAN PRAKTIKUM	20
<i>Source code</i>	20

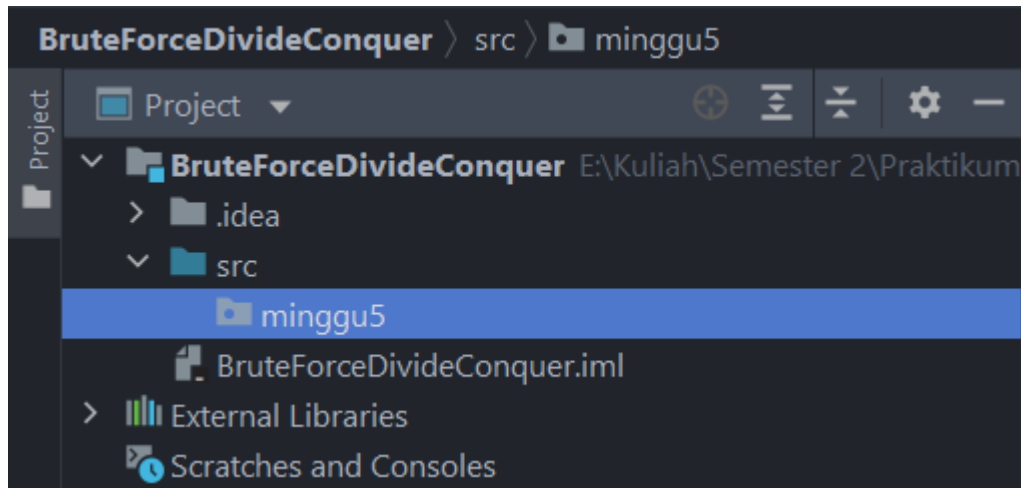


Output.....21

Menghitung Nilai Faktorial dengan Algoritma Brute Force dan Devide Conquer

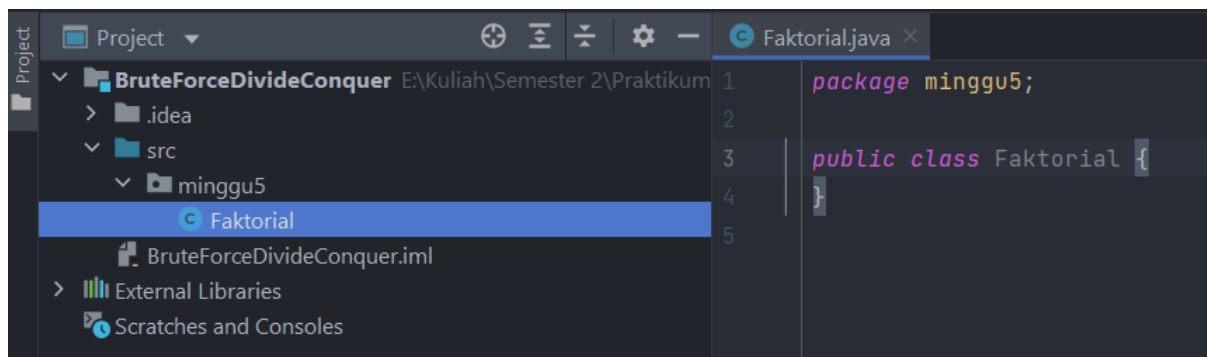
Langkah 1

Buat Project baru, dengan nama “BruteForceDivideConquer”. Buat package dengan nama minggu5



Langkah 2

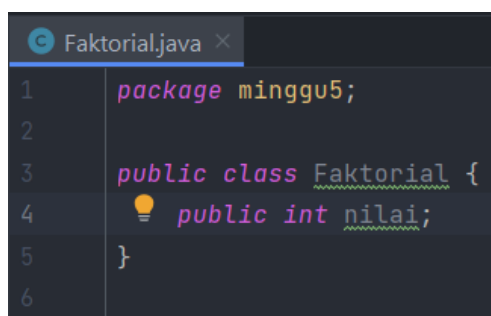
Buatlah class baru dengan nama Faktorial



Langkah 3

Lengkapi class Faktorial dengan atribut dan method yang telah digambarkan di dalam diagram class di atas, sebagai berikut:

- Tambahkan atribut nilai



- b) Tambahkan method faktorialBF() nilai

```
6      public int faktorialBF(int n) {  
7          int fakto = 1;  
8          for (int i = 1; i <= n; i++) {  
9              fakto = fakto * i;  
10         }  
11         return fakto;  
12     }
```

- c) Tambahkan method faktorialDC() nilai

```
14     public int faktorialDC(int n) {  
15         if (n == 1) {  
16             return 1;  
17         }  
18         else {  
19             int fakto = n * faktorialDC(n - 1);  
20             return fakto;  
21         }  
22     }
```

Langkah 4

Coba jalankan (Run) class Faktorial dengan membuat class baru MainFaktorial

1. Di dalam fungsi main sediakan komunikasi dengan user untuk menginputkan jumlah angka yang akan dicari nilai faktorialnya

```
5      public class MainFaktorial {  
6          public static void main(String[] args) {  
7              Scanner sc = new Scanner(System.in);  
8              System.out.println("=====");  
9              System.out.print("Masukkan jumlah elemen yang ingin dihitung : ");  
10             int elemen = sc.nextInt();  
11         }  
12     }
```

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

```
12         Faktorial[] fk = new Faktorial[elemen];  
13         for (int i = 0; i < elemen; i++) {  
14             fk[i] = new Faktorial();  
15             System.out.print("Masukkan nilai data ke-" + (i+1) + " : ");  
16             fk[i].nilai = sc.nextInt();  
17         }
```

3. Tampilkan hasil pemanggilan method faktorialDC() dan faktorialBF()

```

19 System.out.println("=====");
20 System.out.println("Hasil Faktorial dengan Brute Force");
21 for (int i = 0; i < elemen; i++) {
22     System.out.println("Faktorial dari nilai " + fk[i].nilai + " adalah : " + fk[i].faktorialBF(fk[i].nilai));
23 }
24 System.out.println("=====");
25 System.out.println("Hasil Faktorial dengan Devide and Conquer");
26 for (int i = 0; i < elemen; i++) {
27     System.out.println("Faktorial dari nilai " + fk[i].nilai + " adalah : " + fk[i].faktorialDC(fk[i].nilai));
28 }
29 System.out.println("=====");

```

4. Pastikan program sudah berjalan dengan baik!

Verifikasi hasil percobaan

Cocokkan hasil compile kode program anda dengan gambar berikut ini.

```

Run: MainFaktorial x
"C:\Users\Asus TUF DT\.jdk\openjdk-17.0.2\bin\jav
=====
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 Devide and Conquer
Faktorial dari nilai 5 adalah : 120
Faktorial dari nilai 8 adalah : 40320
Faktorial dari nilai 3 adalah : 6
=====
Process finished with exit code 0

```

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. Base line algoritma divide conquer untuk mencari nilai faktorial dari program tersebut adalah if (n==1) baris ini berfungsi untuk menghentikan perulangan rekursif dan sebagai tanda jika masalah sudah tidak bisa di pecah lagi.

2. Tahapan divide → if (n == 1) pada tahapan ini membagi masalah kedua kelompok yaitu base case dan jika n tidak sama dengan 1 maka akan dilakukan fungsi rekursif

Tahapan conquer → $\text{int fakto} = n * \text{faktorialDC}(n-)$ pada tahapan ini masalah yang dipecah akan ditemukan solusinya menggunakan fungsi rekursif.

Tahapan combine → return 1 dan return fakto. Pada tahapan ini semua solusi yang sudah diselesaikan dari pecahan masalah akan disatukan kembali.

3. Bisa menggunakan perulangan tipe lainnya

```

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. Menambahkan perintah untuk menghitung waktu eksekusi kedua method

```

20      long startTimeBF = System.currentTimeMillis();
21      System.out.println("=====");
22      System.out.println("Hasil Faktorial dengan Brute Force");
23      for (int i = 0; i < elemen; i++) {
24          System.out.println("Faktorial dari nilai " + fk[i].nilai + " adalah : " + fk[i].faktorialBF(fk[i].nilai));
25      }
26      long endTimeBF = System.currentTimeMillis();
27      long waktuEksekusiBF = endTimeBF - startTimeBF;
28      System.out.println("Waktu eksekusi metode Brute Force : " + String.valueOf(waktuEksekusiBF) + " milidetik");
29
30      long startTimeDC = System.currentTimeMillis();
31      System.out.println("=====");
32      System.out.println("Hasil Faktorial dengan Devide and Conquer");
33      for (int i = 0; i < elemen; i++) {
34          System.out.println("Faktorial dari nilai " + fk[i].nilai + " adalah : " + fk[i].faktorialDC(fk[i].nilai));
35      }
36      System.out.println("=====");
37      long endTimeDC = System.currentTimeMillis();
38      long waktuEksekusiDC = endTimeDC - startTimeDC;
39      System.out.println("Waktu eksekusi metode Brute Force : " + String.valueOf(waktuEksekusiDC) + " milidetik");

```

5. Ada perbedaan waktu eksekusi

```
Run: MainFaktorial x
"C:\Users\Asus TUF DT\.jdk\openjdk-17.0.2\bin\java.exe" "
=====
Masukkan jumlah elemen yang ingin dihitung : 20
Masukkan nilai data ke-1 : 1
Masukkan nilai data ke-2 : 2
Masukkan nilai data ke-3 : 3
Masukkan nilai data ke-4 : 4
Masukkan nilai data ke-5 : 5
Masukkan nilai data ke-6 : 6
Masukkan nilai data ke-7 : 7
Masukkan nilai data ke-8 : 8
Masukkan nilai data ke-9 : 9
Masukkan nilai data ke-10 : 10
Masukkan nilai data ke-11 : 11
Masukkan nilai data ke-12 : 12
Masukkan nilai data ke-13 : 13
Masukkan nilai data ke-14 : 14
Masukkan nilai data ke-15 : 15
Masukkan nilai data ke-16 : 16
Masukkan nilai data ke-17 : 17
Masukkan nilai data ke-18 : 18
Masukkan nilai data ke-19 : 19
Masukkan nilai data ke-20 : 20
```

```
=====
Hasil Faktorial dengan Brute Force
Faktorial dari nilai 1 adalah : 1
Faktorial dari nilai 2 adalah : 2
Faktorial dari nilai 3 adalah : 6
Faktorial dari nilai 4 adalah : 24
Faktorial dari nilai 5 adalah : 120
Faktorial dari nilai 6 adalah : 720
Faktorial dari nilai 7 adalah : 5040
Faktorial dari nilai 8 adalah : 40320
Faktorial dari nilai 9 adalah : 362880
Faktorial dari nilai 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
Waktu eksekusi metode Brute Force : 17 milidetik
=====
```



```
=====
Hasil Faktorial dengan Devide and Conquer
Faktorial dari nilai 1 adalah : 1
Faktorial dari nilai 2 adalah : 2
Faktorial dari nilai 3 adalah : 6
Faktorial dari nilai 4 adalah : 24
Faktorial dari nilai 5 adalah : 120
Faktorial dari nilai 6 adalah : 720
Faktorial dari nilai 7 adalah : 5040
Faktorial dari nilai 8 adalah : 40320
Faktorial dari nilai 9 adalah : 362880
Faktorial dari nilai 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
=====
Waktu eksekusi metode Brute Force : 11 milidetik
```

Menghitung Hasil Pangkat dengan Algoritma Brute Force dan Divide and Conquer

Langkah 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

```
Pangkat.java x
1 package minggu5;
2
3 public class Pangkat {
4     public int nilai, pangkat;
5 }
6
```

Langkah 2

Pada class Pangkat tersebut, tambahkan method PangkatBF()

```
6 public int pangkatBF(int a, int n) {
7     int hasil = 1;
8     for (int i = 0; i < n; i++) {
9         hasil = hasil * a;
10    }
11    return hasil;
12 }
```

Langkah 3

Pada class Pangkat juga tambahkan method PangkatDC()

```
14 public int pangkatDC(int a, int n) {
15     if (n == 0) {
16         return 1;
17     }
18     else {
19         if (n % 2 == 1) {
20             return (pangkatDC(a, n/2) * pangkatDC(a, n/2) * a);
21         }
22         else {
23             return (pangkatDC(a, n/2) * pangkatDC(a, n/2));
24         }
25     }
26 }
```

Langkah 4

Perhatikan apakah sudah tidak ada kesalahan yang muncul dalam pembuatan class Pangkat

Langkah 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.

```
Pangkat.java MainPangkat.java
1 package minggu5;
2
3 import java.util.Scanner;
4
5 public class MainPangkat {
6     public static void main(String[] args) {
7         Scanner sc = new Scanner(System.in);
8         System.out.println("=====");
9         System.out.print("Masukkan jumlah elemen yang ingin dihitung : ");
10        int elemen = sc.nextInt();
11    }
12 }
```

Langkah 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.

```
12        Pangkat[] png = new Pangkat[elemen];
13
14        for (int i = 0; i < elemen; i++) {
15            png[i] = new Pangkat();
16            System.out.print("Masukkan nilai yang akan dipangkatkan ke-" + (i+1) + " : ");
17            png[i].nilai = sc.nextInt();
18            System.out.print("Masukkan nilai pemangkat ke-" + (i+1) + " : ");
19            png[i].pangkat = sc.nextInt();
20        }
```

Langkah 7

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

```
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 + " adalah " + png[i].pangkatBF(png[i].nilai, png[i].pangkat));
26        }
27
28        System.out.println("=====");
29        System.out.println("Hasil Pangkat dengan Devide and Conquer");
30        for (int i = 0; i < elemen; i++) {
31            System.out.println("Nilai " + png[i].nilai + " pangkat " + png[i].pangkat + " adalah " + png[i].pangkatDC(png[i].nilai, png[i].pangkat));
32        }
33
34        System.out.println("=====");
```

Verifikasi Hasil Percobaan

Pastikan output yang ditampilkan sudah benar seperti di bawah ini.

```
Run: MainPangkat x
"C:\Users\Asus TUF DT\.jdk\openjdk-17.0.2\bin\java
=====
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 Devide and Conquer
Nilai 6 pangkat 2 adalah 36
Nilai 4 pangkat 3 adalah 64
=====
Process finished with exit code 0
```

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. Pada method pangkatBF() menggunakan algoritma brute force sementara pada method pangkatDC() menggunakan algoritma devide conquer
2. Jika nilai n modulus 2 sama dengan 1 maka statement return (pangkatDC(a, n/2) * pangkatDC(a, n/2) * a); akan dijalankan sementara jika nilai n modulus 2 tidak sama dengan satu maka statement return (pangkatDC(a, n/2) * pangkatDC(a, n/2)); akan dijalankan
3. Tahap combine terjadi pada saat return 1, return (pangkatDC(a, n/2) * pangkatDC(a, n/2) * a) dan return (pangkatDC(a, n/2) * pangkatDC(a, n/2));
4. Source code class Pangkat

```

6      Pangkat(int nilai, int pangkat) {
7          this.nilai = nilai;
8          this.pangkat = pangkat;
9      }

```

Source code class MainPangkat

```

10         int elemen = sc.nextInt();
11         int nilai, pangkat;
12         |
13         Pangkat[] png = new Pangkat[elemen];
14
15         for (int i = 0; i < elemen; i++) {
16             System.out.print("Masukkan nilai yang akan dipangkatkan ke-" + (i+1) + " : ");
17             nilai = sc.nextInt();
18             System.out.print("Masukkan nilai pemangkat ke-" + (i+1) + " : ");
19             pangkat = sc.nextInt();
20             png[i] = new Pangkat(nilai, pangkat);
21         }

```

5. Source code

```

24         System.out.println("=====");
25         System.out.println("|| [1] Metode Brute Force ||");
26         System.out.println("|| [2] Metode Devide Conquer ||");
27         System.out.println("=====");
28         System.out.print("Pilih metode yang ingin digunakan (angka 1 atau 2) : ");
29         menu = sc.nextByte();
30
31         if (menu == 1) {
32             System.out.println("=====");
33             System.out.println("Hasil Pangkat dengan Brute Force");
34             for (int i = 0; i < elemen; i++) {
35                 System.out.println("Nilai " + png[i].nilai + " pangkat " + png[i].pangkat + " adalah " + png[i].pangkatBF(png[i].nilai, png[i].pangkat));
36             }
37             System.out.println("=====");
38         } else if (menu == 2) {
39             System.out.println("=====");
40             System.out.println("Hasil Pangkat dengan Devide and Conquer");
41             for (int i = 0; i < elemen; i++) {
42                 System.out.println("Nilai " + png[i].nilai + " pangkat " + png[i].pangkat + " adalah " + png[i].pangkatDC(png[i].nilai, png[i].pangkat));
43             }
44             System.out.println("=====");
45         }

```

Output

```

Run: MainPangkat x
"C:\Users\Asus TUF DT\jdk-17.0.2\bin\java.exe"
=====
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
=====
|| [1] Metode Brute Force ||
|| [2] Metode Devide Conquer ||
=====
Pilih metode yang ingin digunakan (angka 1 atau 2) : 1
=====
Hasil Pangkat dengan Brute Force
Nilai 2 pangkat 2 adalah 4
Nilai 3 pangkat 2 adalah 9
=====

```

Menghitung Sum Array dengan Algoritma Brute Force dan Divide Conquer

Langkah 1

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

```

3      public class Sum {
4          public int elemen;
5          public double keuntungan[];
6          public double total;
7
8      Sum(int elemen) {
9          this.elemen = elemen;
10         this.keuntungan = new double[elemen];
11         this.total = 0;
12     }
13 }

```

Langkah 2

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

```

14     double totalBF(double arr[]) {
15         for (int i = 0; i < elemen; i++) {
16             total = total + arr[i];
17         }
18         return total;
19     }

```

Langkah 3

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

```

21     double totalDC(double arr[], int l, int r) {
22         if (l == r) {
23             return arr[l];
24         } else if (l < r) {
25             int mid = (l+r)/2;
26             double lsum = totalDC(arr, l, mid - 1);
27             double rsum = totalDC(arr, mid + 1, r);
28             return lsum + rsum + arr[mid];
29         }
30         return 0;
31     }

```

Langkah 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

```

6  ▶ public static void main(String[] args) {
7      Scanner sc = new Scanner(System.in);
8      System.out.println("=====");
9      System.out.println("Program Menghitung Keuntungan Total (Satuan juta. Misal 5.9)");
10     System.out.print("Masukkan jumlah bulan : ");
11     int elm = sc.nextInt();
12
13 }

```

Langkah 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

```

13     Sum sm = new Sum(elm);
14     System.out.println("=====");
15     for (int i = 0; i < sm.elemen; i++) {
16         System.out.print("Masukkan keuntungan bulan ke - " + (i+1) + " = ");
17         sm.keuntungan[i] = sc.nextDouble();
18     }

```

Langkah 6

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

```

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

```

Verifikasi Hasil Percobaan

```

Run: MainSum x
"C:\Users\Asus TUF DT\.jdk\openjdk-17.0.2\bin\java.exe" "-javaagent:E:\Dat
=====
Program Menghitung Keuntungan Total (Satuan juta. Misal 5.9)
Masukkan jumlah bulan : 5
=====
Masukkan keuntungan bulan ke - 1 = 8.5
Masukkan keuntungan bulan ke - 2 = 9.54
Masukkan keuntungan bulan ke - 3 = 7.2
Masukkan keuntungan bulan ke - 4 = 9.1
Masukkan keuntungan bulan ke - 5 = 6
=====
Algoritma Brute Force
Total keuntungan perusahaan selama 5 bulan adalah = 40.339999999999996
=====
Algoritma Devide Conquer
Total keuntungan perusahaan selama 5 bulan adalah = 40.34

```

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. Perhitungan keuntungan menggunakan method TotalBF() menjumlahkan nilai keuntungan dari kiri ke kanan sedangkan jika menggunakan method TotalDC() memecah nilai array keuntungan menjadi bagian yang lebih kecil kemudian menjumlahkannya.
2. Source code


```
Scanner sc = new Scanner(System.in);
DecimalFormat df = new DecimalFormat("###.##");
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();

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

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

Output

```
Run: MainSum x
"C:\Users\Asus TUF DT\jdk\openjdk-17.0.2\bin\java.exe" "-j
=====
Program Menghitung Keuntungan Total (Satuan juta. Misal 5.9)
Masukkan jumlah bulan : 5
=====
Masukkan keuntungan bulan ke - 1 = 8.5
Masukkan keuntungan bulan ke - 2 = 9.54
Masukkan keuntungan bulan ke - 3 = 7.2
Masukkan keuntungan bulan ke - 4 = 9.1
Masukkan keuntungan 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
```

3. Untuk menjumlahkan jumlah bagian kiri ditambah jumlah bagian kanan dan jumlah nilai tengah array.
4. Untuk mengetahui nilai median dari array
5. Source code

```
11 System.out.print("Masukkan jumlah perusahaan yang akan dihitung : ");
12 int jumPerusahaan = sc.nextInt();
13
14 for (int i = 0; i < jumPerusahaan; i++) {
15     System.out.println("=====");
16
17     System.out.println("Program Menghitung Keuntungan Total (Satuan juta. Misal 5.9)");
18     System.out.print("Masukkan jumlah bulan perusahaan ke- " + (i+1) + " : ");
19     int elm = sc.nextInt();
20
21     Sum sm = new Sum(elm);
22     System.out.println("=====");
23     for (int j = 0; j < sm.elemen; j++) {
24         System.out.print("Masukkan keuntungan bulan ke - " + (j + 1) + " = ");
25         sm.keuntungan[j] = sc.nextDouble();
26     }
27
28     System.out.println("=====");
29     System.out.println("Algoritma Brute Force");
30     System.out.println("Total keuntungan perusahaan ke- " + (i+1) + " selama " + sm.elemen + " bulan adalah = " + df.format(sm.totalBF(sm.keuntungan)));
31     System.out.println("=====");
32     System.out.println("Algoritma Divide Conquer");
33     System.out.println("Total keuntungan perusahaan ke- " + (i+1) + " selama " + sm.elemen + " bulan adalah = " + df.format(sm.totalDC(sm.keuntungan, 0, sm.elemen - 1)));
34 }
```

Output

```
"C:\Users\Asus TUF DT\jdk\openjdk-17.0.2\bin\java.exe" "-javaagent
Masukkan jumlah perusahaan yang akan dihitung : 2
=====
Program Menghitung Keuntungan Total (Satuan juta. Misal 5.9)
Masukkan jumlah bulan perusahaan ke-1 : 3
=====
Masukkan keuntungan bulan ke - 1 = 1.2
Masukkan keuntungan bulan ke - 2 = 1.3
Masukkan keuntungan bulan ke - 3 = 1.4
=====
Algoritma Brute Force
Total keuntungan perusahaan ke- 1 selama 3 bulan adalah = 3.9
=====
Algoritma Devide Conquer
Total keuntungan perusahaan ke- 1 selama 3 bulan adalah = 3.9
=====
Program Menghitung Keuntungan Total (Satuan juta. Misal 5.9)
Masukkan jumlah bulan perusahaan ke-2 : 2
=====
Masukkan keuntungan bulan ke - 1 = 2.1
Masukkan keuntungan bulan ke - 2 = 2.3
=====
Algoritma Brute Force
Total keuntungan perusahaan ke- 2 selama 2 bulan adalah = 4.4
=====
Algoritma Devide Conquer
Total keuntungan perusahaan ke- 2 selama 2 bulan adalah = 4.4
```

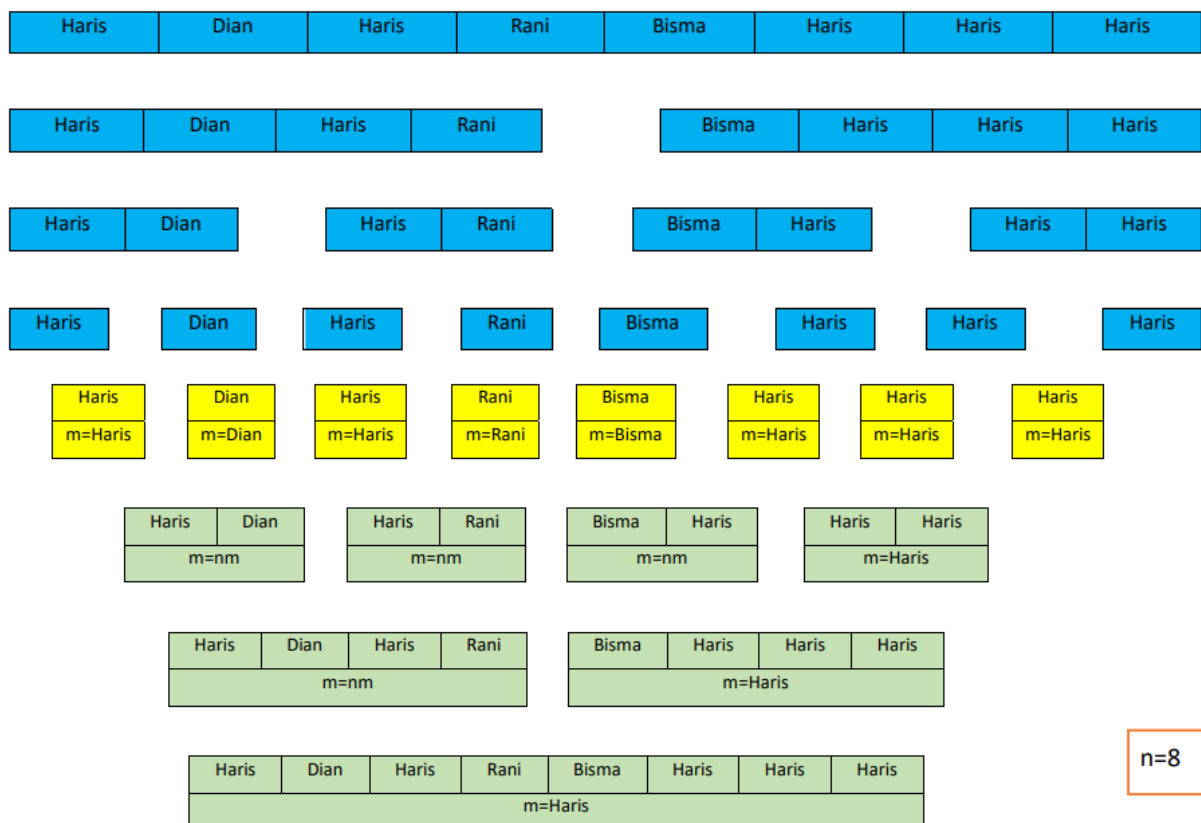
Latihan Praktikum

Suatu Perguruan Tinggi di kota Malang sedang mengadakan pemilihan suara untuk memilih ketua BEM tahun 2022. Jika jumlah suara yang terkumpul diumpamakan selalu genap. Maka dengan inputan kandidat terpilih, carilah mayoritas jumlah suara untuk masing-masing kandidat. (Jumlah elemen array dan hasil pemilihan suara merupakan inputan user).

Elemen Mayoritas : Elemen mayoritas di dalam A adalah elemen yang terdapat pada lebih dari $n/2$ posisi. Contohnya, jika $n=6$ atau $n=7$ maka nilai mayoritas paling sedikit adalah 4. Berasal dari $(7/2)+1$ atau $(6/2)+1$.

Nilai mayoritas berbeda konsep dengan menghitung total suara terbanyak kandidat terpilih !

Contoh : Hasil pemilihan suara sebagai berikut (m adalah mayoritas, nm adalah no mayoritas)



Karena $n = 8$, nilai mayoritas paling sedikit sejumlah 5 ($8/2+1$)

Keterangan : Warna Biru adalah proses divide, warna kuning dimulainya proses conquer, warna hijau dimulainya proses combine

Jawaban Latihan Praktikum

Source code

```
PemilihanSuaraRidwan.java x
1  package tugas;
2
3  import java.util.Scanner;
4
5  public class PemilihanSuaraRidwan {
6  public static void main(String[] args) {
7      Scanner input = new Scanner(System.in);
8      int elemen;
9      String[] kandidat = {"haris", "dian", "rani", "bisma"};
10     System.out.println("Nama kandidat : ");
11     for (int i = 0; i < kandidat.length; i++) {
12         System.out.println(kandidat[i]);
13     }
14     System.out.println();
15     System.out.print("Masukkan jumlah suara yang terkumpul : ");
16     elemen = input.nextInt();
17     input.nextLine();
18     String[] hasilPemilihan = new String[elemen];
19
20     for (int i = 0; i < elemen; i++) {
21         System.out.print("Masukkan nama yang di pilih ke-" + (i+1) + " : ");
22         hasilPemilihan[i] = input.nextLine();
23     }
24
25     System.out.println("Hasil pemilihan suara : ");
26     for (int i = 0; i < elemen; i++) {
27         System.out.print "[" + hasilPemilihan[i] + " ] ";
28     }
29
30     int count1 = 0;
31     int count2 = 0;
32     int count3 = 0;
```

```
33     int count4 = 0;
34
35     for (int i = 0; i < elemen; i++) {
36         switch (hasilPemilihan[i]) {
37             case "haris" -> count1++;
38             case "dian" -> count2++;
39             case "rani" -> count3++;
40             case "bisma" -> count4++;
41             default -> System.out.println("nama kandidat tidak ada");
42         }
43     }
44
45     int[] hasil = {count1, count2, count3, count4};
46     int max = 0;
47     int count = 0;
48     for (int i = 0; i < hasil.length; i++) {
49         if (max < hasil[i]) {
50             max = hasil[i];
51             count = i;
52         }
53     }
54
55     System.out.println();
56     System.out.println("Kandidat yang menjadi mayoritas : " + kandidat[count] );
57
58 }
59
60 }
```

Output

```
Run: PemilihanSuaraRidwan x
"C:\Users\Asus TUF DT\.jdk\openjdk-17.0.2\bin\java.exe" "-javaagent:E:\D
Nama kandidat :
haris
dian
rani
bisma

Masukkan jumlah suara yang terkumpul : 8
Masukkan nama yang di pilih ke-1 : haris
Masukkan nama yang di pilih ke-2 : dian
Masukkan nama yang di pilih ke-3 : haris
Masukkan nama yang di pilih ke-4 : rani
Masukkan nama yang di pilih ke-5 : bisma
Masukkan nama yang di pilih ke-6 : haris
Masukkan nama yang di pilih ke-7 : haris
Masukkan nama yang di pilih ke-8 : haris
Hasil pemilihan suara :
[haris] [dian] [haris] [rani] [bisma] [haris] [haris] [haris]
Kandidat yang menjadi mayoritas : haris

Process finished with exit code 0
```