



NAMA : JIHA RAMDHAN  
NIM : 2341720043  
NO ABSEN : 15  
KELAS : TI-1C  
MATERI : JOBSHEET 5 – BRUTE FORCE & DIVIDE CONQUER

## LAPORAN PRAKTIKUM ALGORITMA DAN STRUKTUR DATA

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

```
PrakASD_1C_15 > P5 > Faktorial > Faktorial15.java > ...
1  package P5.Faktorial;
2
3  public class Faktorial15 {
4      int nilai;
5
6      int faktorialBF(int n){
7          int faktor = 1;
8          for (int i = 1; i <= n; i++) {
9              faktor *= i;
10         }
11         return faktor;
12     }
13
14     int faktorialDC(int n){
15         if (n==1) {
16             return 1;
17         }
18         else{
19             int faktor = n * faktorialDC(n-1);
20             return faktor;
21         }
22     }
23 }
```

```
PrakASD_1C_15 > P5 > Faktorial > MainFaktorial15.java > ...
1  package P5.Faktorial;
2
3  import java.util.Scanner;
4
5  public class MainFaktorial15 {
6      Run | Debug
7      public static void main(String[] args) {
8          Scanner sc15 = new Scanner(System.in);
9          System.out.println(x:"=====");
10         System.out.print(s:"Masukkan jumlah elemen: ");
11         int iJml = sc15.nextInt();
12
13         Faktorial15[] fk15 = new Faktorial15[iJml];
14         for (int i = 0; i < iJml; i++) {
15             fk15[i] = new Faktorial15();
16             System.out.print("Masukkan nilai data ke-" + (i+1) + ": ");
17             int iNilai = sc15.nextInt();
18             fk15[i].nilai = iNilai;
19         }
20
21         System.out.println(x:"HASIL - BRUTE FORCE");
22         for (int i = 0; i < iJml; i++) {
23             System.out.println(
24                 "Hasil penghitungan faktorial menggunakan Brute Force adalah " +
25                 fk15[i].faktorialBF(fk15[i].nilai));
26         }
27
28         System.out.println(x:"HASIL - DIVIDE AND CONQUER");
29         for (int i = 0; i < iJml; i++) {
30             System.out.println(
31                 "Hasil penghitungan faktorial menggunakan Divide and Conquer adalah " +
32                 fk15[i].faktorialDC(fk15[i].nilai));
33         }
34     }
35 }
```



NAMA : JIHA RAMDHAN  
NIM : 2341720043  
NO ABSEN : 15  
KELAS : TI-1C  
MATERI : JOBSHEET 5 – BRUTE FORCE & DIVIDE CONQUER

Output:

```
=====
Masukkan jumlah elemen: 3
Masukkan nilai data ke-1: 5
Masukkan nilai data ke-2: 8
Masukkan nilai data 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:

- if digunakan untuk mengecek apakah parameter  $n$  sama dengan 1. Jika  $n$  sama dengan 1, fungsi mengembalikan nilai 1.
- else digunakan ketika nilai  $n$  tidak sama dengan 1, maka nilai faktorial dihitung dengan cara memanggil kembali fungsi **faktorialDC** dengan parameter  **$n-1$**  (rekursi). Hasil dari rekursi tersebut kemudian dikalikan dengan  $n$ , sehingga menghasilkan faktorial dari nilai  $n$

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

Answer : bisa menggunakan while,

```
6      int faktorialBF(int n){
7          int faktor = 1;
8          int i = 1;
9          while (i <= n) {
10             faktor *= i;
11             i++;
12          }
13          return faktor;
14      }
```

3. Jelaskan perbedaan antara fakto \*= i; dan int fakto = n \* faktorialDC(n-1);!

Answer:

**fakto \*= i; (metode brute force):** Faktorial dihitung secara langsung dengan iterasi dari 1 hingga  $n$ , mengalikan faktor dengan setiap nilai  $i$ .



NAMA : JIHA RAMDHAN  
NIM : 2341720043  
NO ABSEN : 15  
KELAS : TI-1C  
MATERI : JOBSHEET 5 – BRUTE FORCE & DIVIDE CONQUER

**int faktor = n \* faktorialDC(n-1);** (metode Divide Conquer): Faktorial dihitung secara rekursif dengan membagi masalah menjadi bagian yang lebih kecil, mengalikan **n** dengan faktorial dari **n-1**.

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

```
PrakASD_1C_15 > P5 > Pangkat > Pangkat15.java > ...
1 package P5.Pangkat;
2
3 public class Pangkat15 {
4     public int nilai, pangkat;
5
6     int pangkatBF(int a, int n){
7         int hasil = 1;
8         for (int i = 0; i < n; i++) {
9             hasil *= a;
10        }
11        return hasil;
12    }
13
14    int pangkatDC(int a, int n){
15        if (n==0) {
16            return 1;
17        }
18        else{
19            if (n%2 == 1) {
20                return a * (pangkatDC(a, n/2) * pangkatDC(a, n/2));
21            } else {
22                return pangkatDC(a, n/2) * pangkatDC(a, n/2);
23            }
24        }
25    }
26 }
```

```
PrakASD_1C_15 > P5 > Pangkat > MainPangkat15.java > ...
1 package P5.Pangkat;
2
3 import java.util.Scanner;
4
5 public class MainPangkat15 {
6     Run | Debug
7     public static void main(String[] args) {
8         Scanner sc15 = new Scanner(System.in);
9         System.out.println(x:"=====");
10        System.out.println(x:"Masukkan jumlah elemen yang dihitung: ");
11        int elemen = sc15.nextInt();
12
13        Pangkat15[] png15 = new Pangkat15[elemen];
14        for (int i = 0; i < elemen; i++) {
15            png15[i] = new Pangkat15();
16            System.out.println(x:"Masukkan nilai yang hendak dipangkatkan: ");
17            png15[i].nilai = sc15.nextInt();
18            System.out.println(x:"Masukkan nilai pemangkat: ");
19            png15[i].pangkat = sc15.nextInt();
20        }
21
22        System.out.println(x:"HASIL PANGKAT - BRUTE FORCE");
23        for (int i = 0; i < elemen; i++) {
24            System.out.println("Hasil dari " + png15[i].nilai +
25            " pangkat " + png15[i].pangkat +
26            " adalah " + png15[i].pangkatBF(png15[i].nilai, png15[i].pangkat));
27        }
28        System.out.println(x:"HASIL PANGKAT - DIVIDE AND CONQUER");
29        for (int i = 0; i < elemen; i++) {
30            System.out.println("Hasil dari " + png15[i].nilai +
31            " pangkat " + png15[i].pangkat +
32            " adalah " + png15[i].pangkatDC(png15[i].nilai, png15[i].pangkat));
33        }
34    }
```



NAMA : JIHA RAMDHAN  
NIM : 2341720043  
NO ABSEN : 15  
KELAS : TI-1C  
MATERI : JOBSHEET 5 - BRUTE FORCE & DIVIDE CONQUER

Output :

```
=====
Masukkan jumlah elemen yang dihitung: 2
Masukkan nilai yang hendak dipangkatkan: 6
Masukkan nilai pemangkat: 2
Masukkan nilai yang hendak 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:

- **pangkatBF() (Brute Force):** Menggunakan pendekatan brute force dengan iterasi untuk menghitung pangkat.
- **pangkatDC() (Divide Conquer):** Menggunakan pendekatan Divide Conquer atau rekursif untuk menghitung pangkat dengan membagi masalah menjadi submasalah yang lebih kecil.

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

Answer : Ada, combine nya terletak pada return yang terdapat sebuah proses Conquer yang merekursif, dan ada proses divide yang membagi masalah menjadi bagian yang kecil.

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

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

Answer:

```
6      public Pangkat15(int nilai, int pangkat) {
7          this.nilai = nilai;
8          this.pangkat = pangkat;
9      }
```



NAMA : JIHA RAMDHAN  
NIM : 2341720043  
NO ABSEN : 15  
KELAS : TI-1C  
MATERI : JOBSHEET 5 - BRUTE FORCE & DIVIDE CONQUER

```
12 Pangkat15[] png15 = new Pangkat15[elemen];
13 for (int i = 0; i < elemen; i++) {
14     System.out.print(s:"Masukkan nilai yang hendak dipangkatkan: ");
15     int nilai = sc15.nextInt();
16     System.out.print(s:"Masukkan nilai pemangkat: ");
17     int pangkat = sc15.nextInt();
18     png15[i] = new Pangkat15(nilai, pangkat);
19 }
```

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

Answer:

```
21 System.out.println(x:"=====");
22 System.out.println(x:"1. Brute Force");
23 System.out.println(x:"2. Divide and Conquer");
24 System.out.print(s:"Pilih cara Hitung: (1/2) : ");
25 int input = sc15.nextInt();
26
27 switch (input) {
28     case 1:
29         System.out.println(x:"HASIL PANGKAT - BRUTE FORCE");
30         for (int i = 0; i < elemen; i++) {
31             System.out.println("Hasil dari " + png15[i].nilai +
32                 " pangkat " + png15[i].pangkat +
33                 " adalah " + png15[i].pangkatBF(png15[i].nilai, png15[i].pangkat));
34         }
35         break;
36     case 2:
37         System.out.println(x:"HASIL PANGKAT - DIVIDE AND CONQUER");
38         for (int i = 0; i < elemen; i++) {
39             System.out.println("Hasil dari " + png15[i].nilai +
40                 " pangkat " + png15[i].pangkat +
41                 " adalah " + png15[i].pangkatDC(png15[i].nilai, png15[i].pangkat));
42         }
43     default:
44         break;
45 }
46
47 }
```

```
=====
Masukkan jumlah elemen yang dihitung: 2
Masukkan nilai yang hendak dipangkatkan: 6
Masukkan nilai pemangkat: 2
Masukkan nilai yang hendak dipangkatkan: 4
Masukkan nilai pemangkat: 3
=====
```

```
1. Brute Force
2. Divide and Conquer
Pilih cara Hitung: (1/2) : 1
HASIL PANGKAT - BRUTE FORCE
Hasil dari 6 pangkat 2 adalah 36
Hasil dari 4 pangkat 3 adalah 64
```

```
=====
Masukkan jumlah elemen yang dihitung: 2
Masukkan nilai yang hendak dipangkatkan: 6
Masukkan nilai pemangkat: 2
Masukkan nilai yang hendak dipangkatkan: 4
Masukkan nilai pemangkat: 3
=====
```

```
1. Brute Force
2. Divide and Conquer
Pilih cara Hitung: (1/2) : 2
HASIL PANGKAT - DIVIDE AND CONQUER
Hasil dari 6 pangkat 2 adalah 36
Hasil dari 4 pangkat 3 adalah 64
```



NAMA : JIHA RAMDHAN  
NIM : 2341720043  
NO ABSEN : 15  
KELAS : TI-1C  
MATERI : JOBSHEET 5 – BRUTE FORCE & DIVIDE CONQUER

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

```
PrakASD_1C_15 > P5 > SumArray > MainSum15.java > ...
1  package P5.SumArray;
2
3  import java.util.Scanner;
4
5  public class MainSum15 {
6      Run | Debug
7      public static void main(String[] args) {
8          Scanner sc15 = new Scanner(System.in);
9          System.out.println("=====");
10         System.out.println("Program Menghitung Keuntungan Total (Satuan Juta, Misal 5.9)");
11         System.out.print("Masukkan Jumlah Bulan : ");
12         int elm = sc15.nextInt();
13
14         Sum15 sm15 = new Sum15(elm);
15         System.out.println("=====");
16         for (int i = 0; i < sm15.elemen; i++) {
17             System.out.print("Masukkan Keuntungan Bulan ke- " + (i+1) + " : ");
18             sm15.keuntungan[i] = sc15.nextDouble();
19         }
20
21         System.out.println("=====");
22         System.out.println("Algoritma Brute Force");
23         System.out.println("Total Keuntungan perusahaan selama " + sm15.elemen +
24             " bulan adalah " + sm15.totalBF(sm15.keuntungan));
25         System.out.println("=====");
26         System.out.println("Algoritma Divide and Conquer");
27         System.out.println("Total Keuntungan perusahaan selama " + sm15.elemen +
28             " bulan adalah " + sm15.totalDC(sm15.keuntungan, 1, 0, sm15.elemen-1));
29     }
30 }
```

```
PrakASD_1C_15 > P5 > SumArray > Sum15.java > ...
1  package P5.SumArray;
2
3  public class Sum15 {
4      int elemen;
5      double keuntungan[], total;
6
7      Sum15(int elemen){
8          this.elemen = elemen;
9          this.keuntungan = new double[elemen];
10         this.total = 0;
11     }
12
13     double totalBF(double arr[]){
14         for (int i = 0; i < arr.length; i++) {
15             total = total + arr[i];
16         }
17         return total;
18     }
19
20     double totalDC(double arr[], int l, int r){
21         if (l==r) {
22             return arr[l];
23         } else if (l<r) {
24             int mid = (l+r)/2;
25             double lsum = totalDC(arr, l, mid-1);
26             double rsum = totalDC(arr, mid+1, r);
27             return lsum + rsum + arr[mid];
28         }
29         return 0;
30     }
31 }
```



NAMA : JIHA RAMDHAN  
NIM : 2341720043  
NO ABSEN : 15  
KELAS : TI-1C  
MATERI : JOBSHEET 5 – BRUTE FORCE & DIVIDE CONQUER

Output :

```
=====
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 Divide and Conquer
Total Keuntungan perusahaan selama 5 bulan adalah 40.34
```

Question :

1. Mengapa terdapat formulasi return value berikut?Jelaskan!

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

Answer: Formulasi **return lsum + rsum + arr[mid];** digunakan dalam metode **totalDC** untuk menggabungkan hasil keuntungan dari submasalah yang lebih kecil pada algoritma Divide and Conquer.

Saat rekursi mencapai kasus dasar di mana l dan r sama, metode ini mengembalikan nilai keuntungan pada bulan tersebut. Pada tahap combine, nilai keuntungan dari kiri (**lsum**) dan kanan (**rsum**) ditambahkan, dan ditambah dengan keuntungan pada bulan tengah (**arr[mid]**)

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

Answer : Variabel **mid** digunakan dalam metode **totalDC** untuk menentukan titik tengah dari rentang array saat ini yang sedang diproses. Ini digunakan untuk membagi rentang masalah menjadi dua bagian yang lebih kecil saat memanggil rekursif metode **totalDC**





NAMA : JIHA RAMDHAN  
NIM : 2341720043  
NO ABSEN : 15  
KELAS : TI-1C  
MATERI : JOBSHEET 5 – BRUTE FORCE & 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 : Berikut Programnya:

```
PrakASD_1C_15 > P5 > SumArray > MainSum15.java > ...
You, 18 seconds ago | 1 author (You)
1 package P5.SumArray;
2
3 import java.util.Scanner;
4
You, 6 seconds ago | 1 author (You)
5 public class MainSum15 {
    Run | Debug
6     public static void main(String[] args) {
7         Scanner sc15 = new Scanner(System.in);
8         System.out.println(x:"=====");
9         System.out.println(x:"Program Menghitung Keuntungan Total (Satuan Juta, Misal 5.9)");
10        System.out.println(x:"=====");
11
12        System.out.print(s:"Masukkan Jumlah Perusahaan : ");
13        int jumlahPerusahaan = sc15.nextInt();
14
15        Sum15[] sm15 = new Sum15[jumlahPerusahaan];
16
17        for (int i = 0; i < sm15.length; i++) {
18            System.out.println(x:"=====");
19            System.out.println("Perusahaan ke-" + (i + 1));
20            System.out.print(s:"Masukkan Jumlah Bulan : ");
21            int elm = sc15.nextInt();
22
23            sm15[i] = new Sum15(elm);
24            System.out.println(x:"=====");
25            for (int j = 0; j < elm; j++) {
26                System.out.print("Masukkan Keuntungan Bulan ke-" + (j+1) + " : ");
27                sm15[i].keuntungan[j] = sc15.nextDouble();
28            }
29        }
30    }
31
32    System.out.println(x:"\n=====");
33    System.out.println(x:"HASIL PERHITUNGAN KEUNTUNGAN PERUSAHAAN");
34    for (int i = 0; i < jumlahPerusahaan; i++) {
35        System.out.println(x:"=====");
36        System.out.println("Perusahaan ke-" + (i + 1) + ":");
37        System.out.println(x:"Algoritma Brute Force");
38        System.out.println("Total Keuntungan perusahaan selama " + sm15[i].elemen +
39            " bulan adalah " + sm15[i].totalBF(sm15[i].keuntungan));
40        System.out.println(x:"-----");
41        System.out.println(x:"Algoritma Divide and Conquer");
42        System.out.println("Total Keuntungan perusahaan selama " + sm15[i].elemen +
43            " bulan adalah " + sm15[i].totalDC(sm15[i].keuntungan, 1, 0, sm15[i].elemen-1));
44    }
45 }
46
```

Dengan cara menambah input jumlah perusahaan, kemudian membuat objek Sum15 yang jumlahnya sesuai dengan jumlah perusahaan, lalu dalam loopingnya baru memasukkan jumlah bulan keuntungan, dan nantinya total keuntungan nya ditampilkan setiap perusahaannya.





NAMA : JIHA RAMDHAN  
NIM : 2341720043  
NO ABSEN : 15  
KELAS : TI-1C  
MATERI : JOBSHEET 5 – BRUTE FORCE & DIVIDE CONQUER  
Outputnya :

```
=====
Program Menghitung Keuntungan Total (Satuan Juta, Misal 5.9)
=====
```

```
Masukkan Jumlah Perusahaan : 2
=====
```

```
Perusahaan ke-1
```

```
Masukkan Jumlah Bulan : 2
=====
```

```
Masukkan Keuntungan Bulan ke-1 : 5.6
```

```
Masukkan Keuntungan Bulan ke-2 : 7.4
=====
```

```
Perusahaan ke-2
```

```
Masukkan Jumlah Bulan : 3
=====
```

```
Masukkan Keuntungan Bulan ke-1 : 9.3
```

```
Masukkan Keuntungan Bulan ke-2 : 8
```

```
Masukkan Keuntungan Bulan ke-3 : 5.6
=====
```

```
=====
HASIL PERHITUNGAN KEUNTUNGAN PERUSAHAAN
=====
```

```
Perusahaan ke-1:
```

```
Algoritma Brute Force
```

```
Total Keuntungan perusahaan selama 2 bulan adalah 13.0
-----
```

```
Algoritma Divide and Conquer
```

```
Total Keuntungan perusahaan selama 2 bulan adalah 13.0
=====
```

```
Perusahaan ke-2:
```

```
Algoritma Brute Force
```

```
Total Keuntungan perusahaan selama 3 bulan adalah 22.9
-----
```

```
Algoritma Divide and Conquer
```

```
Total Keuntungan perusahaan selama 3 bulan adalah 22.9
```



NAMA : JIHA RAMDHAN  
NIM : 2341720043  
NO ABSEN : 15  
KELAS : TI-1C  
MATERI : JOBSHEET 5 - BRUTE FORCE & DIVIDE CONQUER

#### 4.5 Latihan Praktikum

1. Sebuah showroom memiliki daftar mobil dengan data sesuai tabel di bawah ini

merk	tipe	tahun	top_acceleration	top_power
BMW	M2 Coupe	2016	6816	728
Ford	Fiesta ST	2014	3921	575
Nissan	370Z	2009	4360	657
Subaru	BRZ	2014	4058	609
Subaru	Impreza WRX STI	2013	6255	703
Toyota	AE86 Trueno	1986	3700	553
Toyota	86/GT86	2014	4180	609
Volkswagen	Golf GTI	2014	4180	631

Tentukan:

- top\_acceleration tertinggi menggunakan Divide and Conquer!
- top\_acceleration terendah menggunakan Divide and Conquer!
- Rata-rata top\_power dari seluruh mobil menggunakan Brute Force

Class Showroom :

```
PrakASD_1C_15 > P5 > LatihanPraktikum > Showroom15.java > ...
1  package P5.LatihanPraktikum;
2
3  public class Showroom15 {
4      String merk;
5      String tipe;
6      int tahun;
7      int top_acceleration;
8      int top_power;
9
10     Showroom15(String merk, String tipe, int tahun, int top_acceleration, int top_power){
11         this.merk = merk;
12         this.tipe = tipe;
13         this.tahun = tahun;
14         this.top_acceleration = top_acceleration;
15         this.top_power = top_power;
16     }
17
18     static int[] divideConquer(Showroom15[] cars, int low, int high){
19         if (low == high) {
20             return new int[]{cars[low].top_acceleration, cars[low].top_acceleration};
21         } else {
22             int mid = (low + high) / 2;
23             int[] lsum = divideConquer(cars, low, mid);
24             int[] rsum = divideConquer(cars, mid + 1, high);
25
26             return new int[]{Math.max(lsum[0], rsum[0]), Math.min(lsum[1], rsum[1])};
27         }
28     }
29
30     public static double rata2TopPower(Showroom15[] cars) {
31         int totalPower = 0;
32         for (int i = 0; i < cars.length; i++) {
33             totalPower += cars[i].top_power;
34         }
35         return (double) totalPower / cars.length;
36     }
37 }
```



NAMA : JIHA RAMDHAN  
NIM : 2341720043  
NO ABSEN : 15  
KELAS : TI-1C  
MATERI : JOBSHEET 5 - BRUTE FORCE & DIVIDE CONQUER

Main Showroom :

```
PrakASD_1C_15 > P5 > LatihanPraktikum > MainShowroom15.java > ...
1 package P5.LatihanPraktikum;
2
3 public class MainShowroom15 {
4     Run | Debug
5     public static void main(String[] args) {
6         Showroom15[] cars = {
7             new Showroom15(merk:"BMW", tipe:"M2 Coupe", tahun:2016, top_acceleration:6816, top_power:728),
8             new Showroom15(merk:"Ford", tipe:"Fiesta ST", tahun:2014, top_acceleration:3921, top_power:575),
9             new Showroom15(merk:"Nissan", tipe:"370Z", tahun:2009, top_acceleration:4360, top_power:657),
10            new Showroom15(merk:"Subaru", tipe:"BRZ", tahun:2014, top_acceleration:4058, top_power:609),
11            new Showroom15(merk:"Subaru", tipe:"Impreza WRX STI", tahun:2013, top_acceleration:6255, top_power:703),
12            new Showroom15(merk:"Toyota", tipe:"AE86 Trueno", tahun:1986, top_acceleration:3700, top_power:553),
13            new Showroom15(merk:"Toyota", tipe:"86/GT86", tahun:2014, top_acceleration:4180, top_power:609),
14            new Showroom15(merk:"Volkswagen", tipe:"Golf GTI", tahun:2014, top_acceleration:4180, top_power:631)
15        };
16
17        System.out.println(x:"-----");
18        System.out.printf(format:"| %-12s | %-15s | %-5s | %-17s | %-9s |\n",
19            ...args:"Merk", "Tipe", "Tahun", "Top Acceleration", "Top Power");
20        System.out.println(x:"-----");
21        for (Showroom15 car : cars) {
22            System.out.printf(format:"| %-12s | %-15s | %-5s | %-17s | %-9s |\n",
23                car.merk, car.tipe, car.tahun, car.top_acceleration, car.top_power);
24        }
25        System.out.println(x:"-----");
26
27        int max = Showroom15.divideConquer(cars, low:0, cars.length-1)[0];
28        int min = Showroom15.divideConquer(cars, low:0, cars.length-1)[1];
29        double avg = Showroom15.rata2TopPower(cars);
30        System.out.println(x:"== Top Acceleration ==");
31        System.out.println("Max : " + max);
32        System.out.println("Min : " + min);
33        System.out.println(x:"== Top Power ==");
34        System.out.println("Avg : " + avg);
35    }
36 }
```

Outputnya :

Merk	Tipe	Tahun	Top Acceleration	Top Power
BMW	M2 Coupe	2016	6816	728
Ford	Fiesta ST	2014	3921	575
Nissan	370Z	2009	4360	657
Subaru	BRZ	2014	4058	609
Subaru	Impreza WRX STI	2013	6255	703
Toyota	AE86 Trueno	1986	3700	553
Toyota	86/GT86	2014	4180	609
Volkswagen	Golf GTI	2014	4180	631

== Top Acceleration ==  
Max : 6816  
Min : 3700  
== Top Power ==  
Avg : 633.125