JOBSHEET IV

**BRUTE FORCE DAN DIVIDE CONQUER**

Nama : Muhammad Abhinaya Zurfa  
Kelas : SIB 1 C / 20

# Tujuan Praktikum

Setelah melakukan materi praktikum ini, mahasiswa mampu:

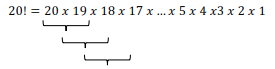
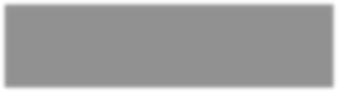
* + 1. Mahasiswa mampu membuat algoritma bruteforce dan divide-conquer
    2. Mahasiswa mampu menerapkan penggunaan algorima bruteforce dan divide-conquer

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

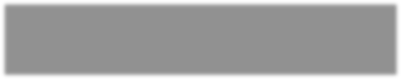
Perhatikan Diagram Class berikut ini :

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

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



Tahapan pencarian nilai faktorial dengan algoritma Divide and Conquer :

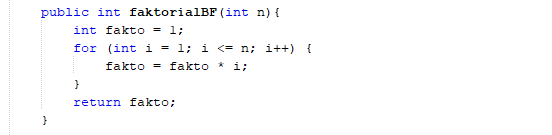
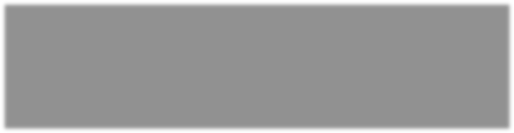


## Langkah-langkah Percobaan

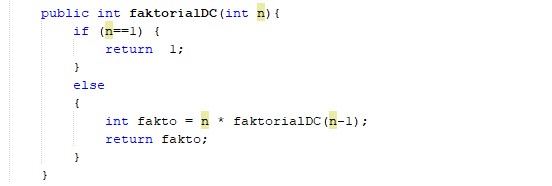
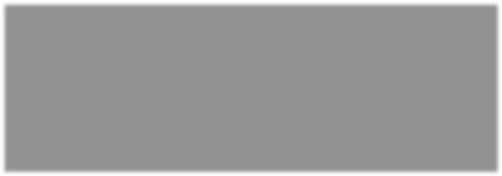
1. Buat Project baru, dengan nama “**BruteForceDivideConquer**”. Buat package dengan nama minggu5.
2. Buatlah class baru dengan nama **Faktorial**
3. Lengkapi class **Faktorial** dengan atribut dan method yang telah digambarkan di dalam diagram class di atas, sebagai berikut:
   1. Tambahkan atribut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



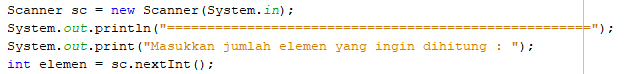
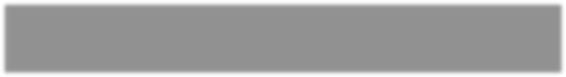
* 1. Tambahkan method faktorialBF() nilai



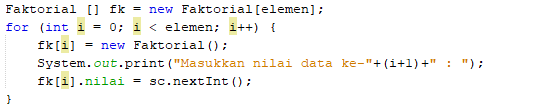
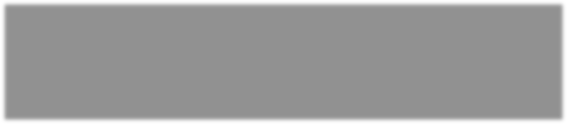
* 1. Tambahkan method faktorialDC() nilai



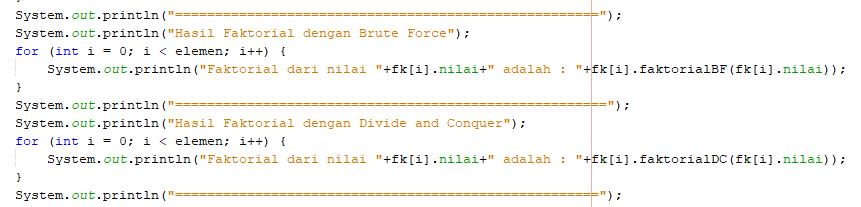
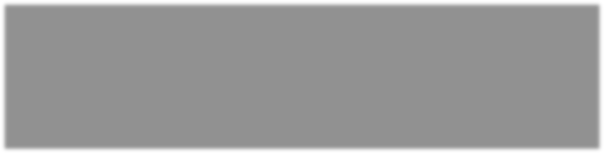
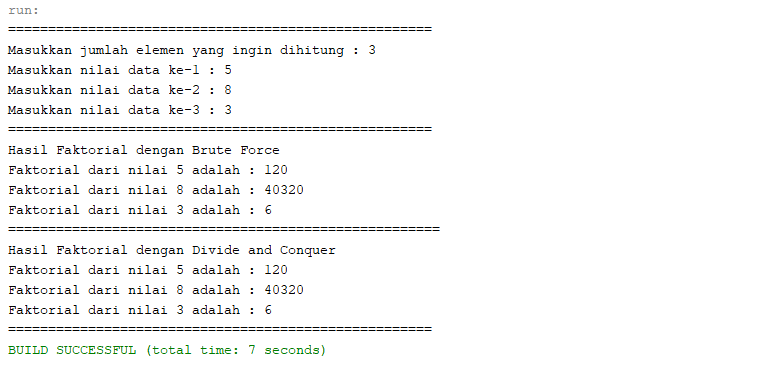
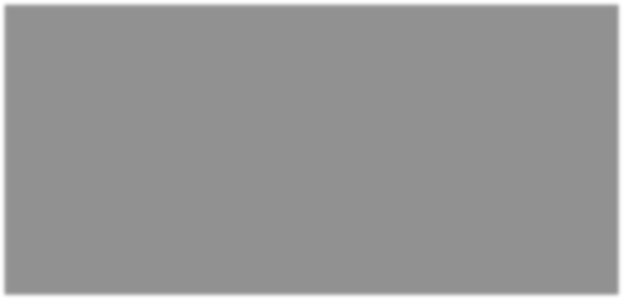
1. Coba jalankan (Run) class Faktorial dengan membuat class baru MainFaktorial.
   1. Di dalam fungsi main sediakan komunikasi dengan user untuk menginputkan jumlah angka yang akan dicari nilai faktorialnya



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



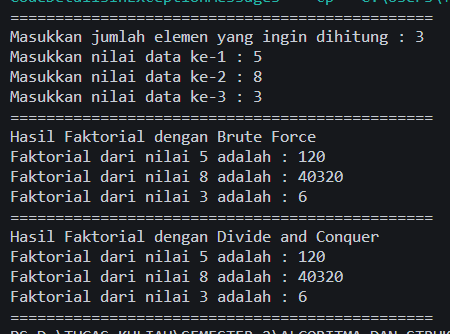
* 1. Tampilkan hasil pemanggilan method faktorialDC() dan faktorialBF()



* 1. Pastikan program sudah berjalan dengan baik!

## Verifikasi Hasil Percobaan

Cocokkan hasil compile kode program anda dengan gambar berikut ini.



## Pertanyaan

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

Pada base line DC , kasus dasarnya adalah n == 1, dimana faktorial dari 1 adalah 1 itu sendiri, lalu jika n lebih besar dari 1 maka faktorial n akan dihitung di else dimana n \* faktorialDC(n-1), dimana angka n dikali faktorial dari n-1, kemudian digabung hasilnya dan mendapatkan nilai faktorial dari n.

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

Pada implementasi kode program tersebut DC nya terdiri dari 2 tahapan, yaitu divide dimana faktorial n dibagi menjadi masalah yang lebih kecil, yaitu faktorial dari n -1, lalu conquer dimana tahap ini terjadi Ketika n == 1 dimana langsung mengembalikan nilai 1, untuk combine dalam kasus ini setiap pemanggilan rekursif sudah mengalikan n dengan hasil dari faktorialDC(n-1), sehingga penggabungan terjadi secara otomatis selama proses rekursi

1. Apakah memungkinkan perulangan pada method faktorialBF() dirubah selain menggunakan for?Buktikan! bisa dirubah menggunakan while atau do while, berikut contohnya

    public int faktorialBF(int n) {

        int fakto = 1;

        int i = 1;

        do {

            fakto \*= i;

            i++;

        } while ( i <=n );

        return fakto;

    }

1. Tambahkan pegecekan waktu eksekusi kedua jenis method tersebut!
2. Buktikan dengan inputan elemen yang di atas 20 angka, apakah ada perbedaan waktu eksekusi?

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

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

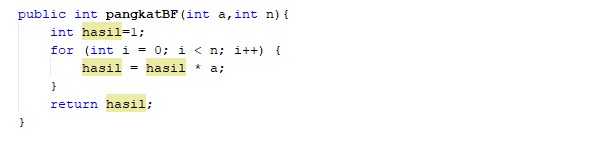
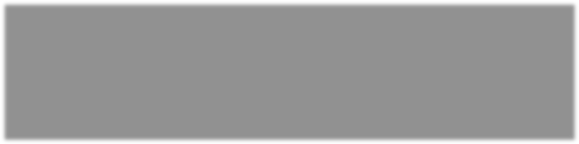
## Langkah-langkah Percobaan

1. Di dalam paket **minggu5**, buatlah class baru dengan nama **Pangkat**. Dan di dalam class **Pangkat**

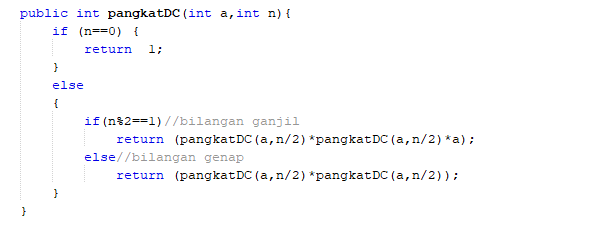
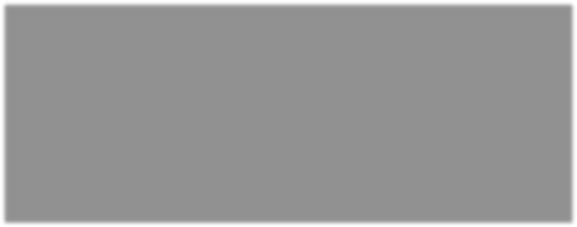
tersebut, buat atribut angka yang akan dipangkatkan sekaligus dengan angka pemangkatnya



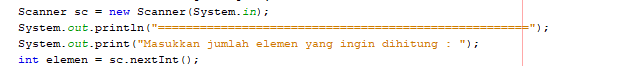
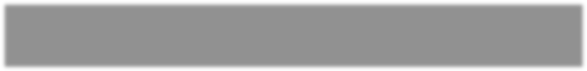
1. Pada class Pangkat tersebut, tambahkan method PangkatBF()



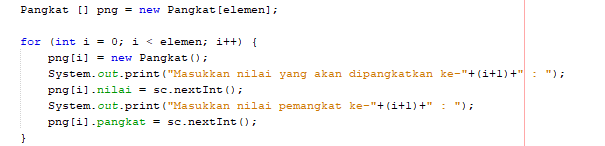
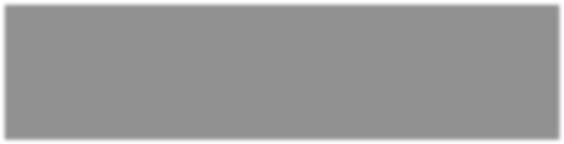
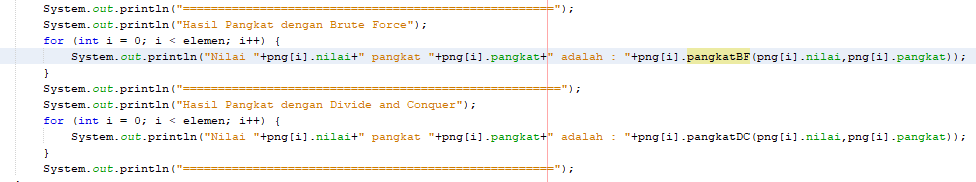
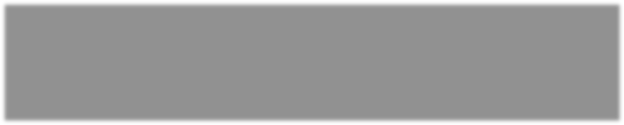
1. Pada class Pangkat juga tambahkan method PangkatDC()



1. Perhatikan apakah sudah tidak ada kesalahan yang muncul dalam pembuatan class Pangkat
2. 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.



1. 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.

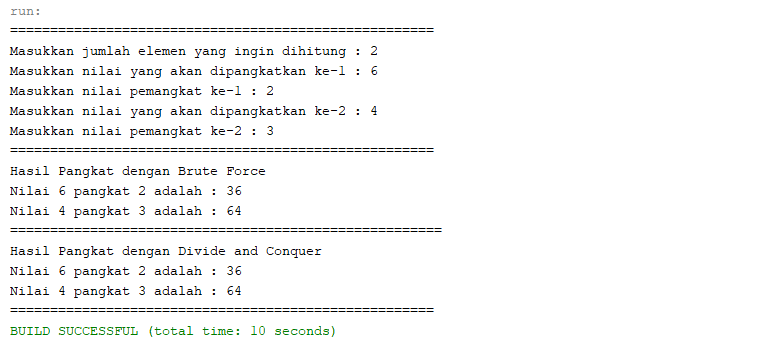
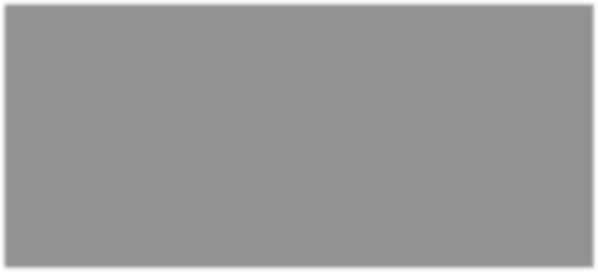


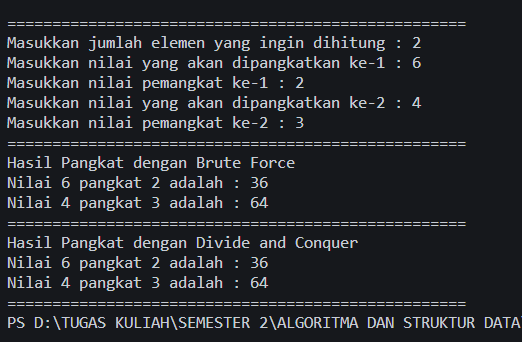
1. Kemudian, panggil hasil nya dengan mengeluarkan return value dari method PangkatBF() dan

PangkatDC().

## Verifikasi Hasil Percobaan

Pastikan output yang ditampilkan sudah benar seperti di bawah ini.





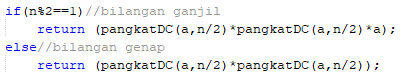
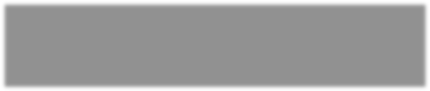
## Pertanyaan

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

Pada method PangkatBF() menggunakan perulangan for untuk menghitung hasil pangkatnya dimana mengalikan bilangan a dengan dirinya sendiri sebanyak n kali.

Sedangkan pada method PangkatDC() menggunakan rekursif, dimana jika angka n nya ganjil maka ia akan menghitung pangkat dengan mengalikan hasil dari pangkat setengah dari n dengan dirinya sendiri dan juga mengalikan dengan a sekali lagi, lalu jika angka n nya genap maka ia akan melakukan yang sama seperti ganjil tapi tidak mengalikan dengan a nya.

1. Pada method PangkatDC() terdapat potongan program sebagai berikut:



Jelaskan arti potongan kode tersebut

* Pertama pada if , jika n sisa 1 habis dibagi/ modulo 2 maka ia termasuk bilangan ganjil, lalu jika termasuk bilangan ganjil akan menjalankan return dan fungsi akan memanggil dirinya sendiri dua kali dengan parameter n dibagi 2 lalu dikalikan lagi dengan a agar mendapatkan hasil pangkat
* Kedua pada else, Ketika n nya genap akan menjalankan memanggil dirinya sendiri dua kali dengan parameter n dibagi dua tanpa perlu mengalikan dengan a lagi untuk mendapatkan nilai pangkatnya

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

Ya, Pada metode PangkatDC() tahap combine terjadi saat hasil dari dua panggilan rekursif (pangkatDC (a, n/2) dikalikan dengann dirinya sendiri kemudian dikalikan dengan a lagi (untuk ganjil) dan itu merupakan proses combine / penggabungan hasil dari sub-proses rekursif menjadi hasil akhir

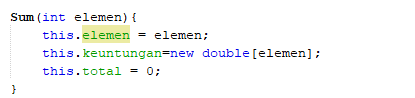
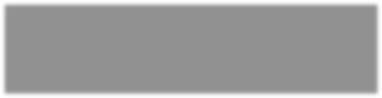
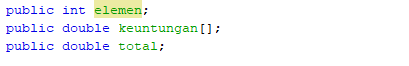
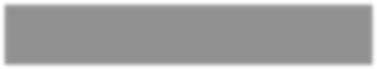
1. Modifikasi kode program tersebut, anggap proses pengisian atribut dilakukan dengan konstruktor.
2. Tambahkan menu agar salah satu method yang terpilih saja yang akan dijalankan!

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

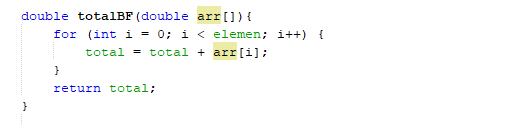
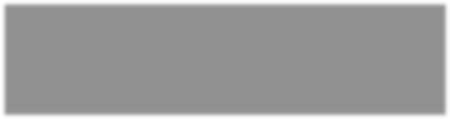
Di dalam percobaan ini, kita akan mempraktekkan bagaimana proses *divide, conquer*, dan *combine* diterapkan pada studi kasus penjumlahan keuntungan suatu perusahaan dalam beberapa bulan.

## Langkah-langkah Percobaan

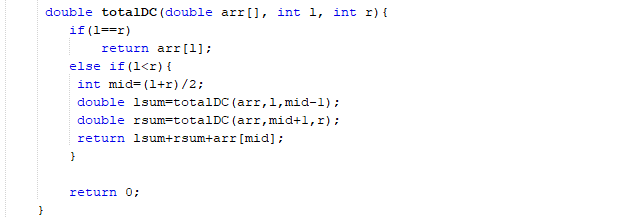
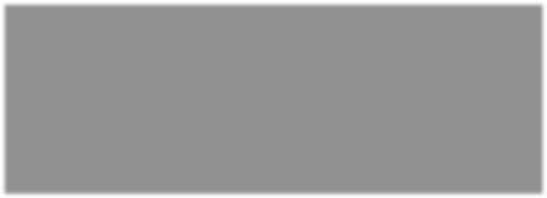
1. Pada paket minggu5. Buat class baru yaitu class Sum. DI salam class tersebut terdapat beberapa atribut jumlah elemen array, array, dan juga total. Tambahkan pula konstruktor pada class Sum.



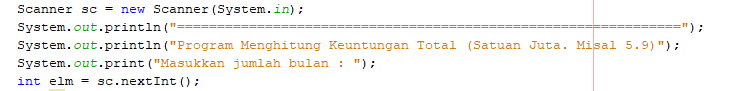
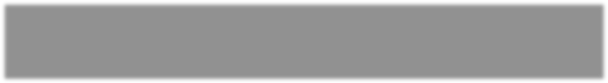
1. Tambahkan method TotalBF() yang akan menghitung total nilai array dengan cara *iterative*.



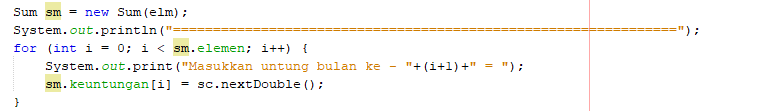
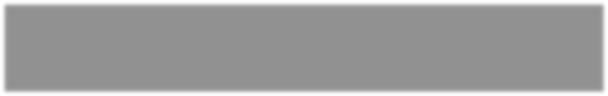
1. Tambahkan pula method TotalDC() untuk implementasi perhitungan nilai total array menggunakan algoritma Divide and Conquer



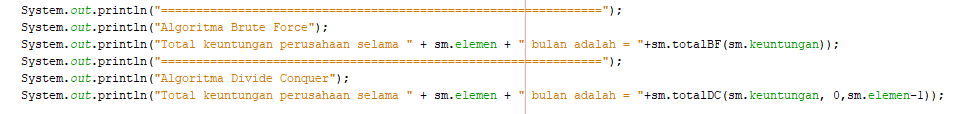
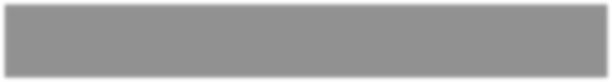
1. Buat class baru yaitu MainSum. Di dalam kelas ini terdapat method main. Pada method ini user dapat menuliskan berapa bulan keuntungan yang akan dihitung. Dalam kelas ini sekaligus dibuat instansiasi objek untuk memanggil atribut ataupun fungsi pada class Sum



1. 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.

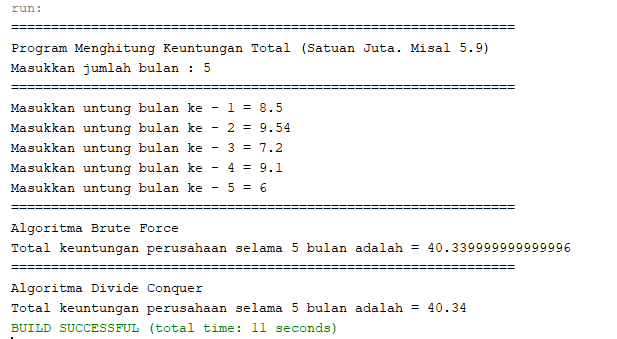
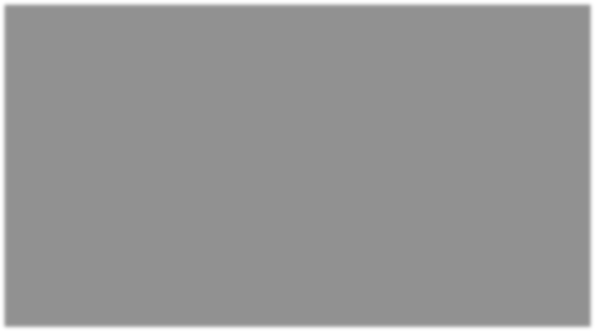


1. Tampilkan hasil perhitungan melalui objek yang telah dibuat untuk kedua cara yang ada (Brute Force dan Divide and Conquer)



## Verifikasi Hasil Percobaan

Cocokkan hasil compile kode program anda dengan gambar berikut ini.



## Pertanyaan

* + - 1. Berikan ilustrasi perbedaan perhitungan keuntungan dengan method TotalBF() ataupun

TotalDC()

* + - 1. 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.
      2. Mengapa terdapat formulasi *return value* berikut?Jelaskan!



* + - 1. Kenapa dibutuhkan variable mid pada method TotalDC()?
      2. 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!

# 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.*