**4.2.3 Pertanyaan**

1. Jelaskan mengenai base line Algoritma Divide Conquer untuk melakukan pencarian nilai faktorial! Jawaban: ➢ Membagi menjadi 2 upa-masalah. Pada kode program di atas berada pada baris if (n==1) dan else. ➢ Menyelesaikan masalah (solusi) secara rekursif. Pada kode program di atas berada pada baris int fakto = n \* faktorialDC (n - 1). ➢ Menggabungkan hasil solusi dari kedua upa-masalah. Pada kode program di atas berada pada baris return 1 dan return fakto.
2. Pada implementasi Algoritma Divide and Conquer Faktorial apakah lengkap terdiri dari 3 tahapan divide, conquer, combine? Jelaskan masing-masing bagiannya pada kode program! Jawaban: Iya lengkap. ➢ Divide : if (n==1) dan else. Terdapat pada baris 16. ➢ Conquer : int fakto = n \* faktorialDC(n-1). Terdapat pada baris 20. ➢ Combine : return 1 dan return fakto. Terdapat pada baris 17 dan 21.
3. Apakah memungkinkan perulangan pada method faktorialBF() diubah selain menggunakan for? Buktikan! Jawaban: Iya memungkinkan dapat diubah menggunakan while.
4. Buktikan dengan inputan elemen yang di atas 20 angka, apakah ada perbedaan waktu eksekusi? Jawaban: Terdapat perbedaan waktu.

**4.3.3 Pertanyaan**

1. Jelaskan mengenai perbedaan 2 method yang dibuat yaitu PangkatBF() dan PangkatDC()! Jawaban: PangkatBF menggunakan iteratifsedangkan PangkatDC menggunakan rekursif.
2. Pada method PangkatDC() terdapat potongan program sebagai berikut: Jelaskan arti potongan kode tersebut! Jawaban: Jika (n%2==1) maka merupakan bilangan ganjil. Jadi, perhitungan pangkat dikalikan dengan a lagi (pangkatDC(a, n/2) \* pangkatDC(a, n/2) \* a), dan jika selain itu maka merupakan bilangan genap jadi perhitungannya hanya (pangkatDC(a, n/2) \* pangkatDC(a, n/2)).
3. Apakah tahap combine sudah termasuk dalam kode tersebut? Tunjukkan! Jawaban: Belum, Tahapan combine belum termasuk dalam kode tersebut karena hanya terdapat 2 solusi dari 2 upa-masalah, sedangkan jumlah semua upa-masalah dan solusi dari upa-masalah tersebut adalah 3. Jadi, belum bisa di combine.
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!

**4.4.3 Pertanyaan**

1. Berikan ilustrasi perbedaan perhitungan keuntungan dengan method TotalBF() ataupun TotalDC() Jawaban: Keuntungan TotalBF yaitu: ➢ Dapat digunakan untuk memecahkan hampir sebagian besar masalah, sederhana, mudah dimengerti, lebih cocok dalam permasalahan yg berukuran kecil seperti penjumlahan/perkalian n buahbilangan, menentukan elemen minimum/maksimum. Keuntungan TotalDC yaitu: ➢ Dapat digunakan untuk memecahkan masalah yang sulit, memiliki efisiensi algoritma yang tinggi seperti sorting, bekerja secara parallel.
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! Jawaban: Karena digunakan untuk memenuhi solusi pada upa-masalah ke-2, menggunakan cara rekursif dengan menjumlahkan hasil hitungan lsum, rsum, dan arr[mid].
4. Kenapa dibutuhkan variable mid pada method TotalDC()? Jawaban: Karena variabel mid digunakan sebagai tempat dari rumus (l+r)/2 dengan bertipe data integer. Baris kode variabel mid dituliskan terlebih dahulu agar dapat dipanggil di baris kode yang lain, agar tidak menuliskan rumus yang sama, dan hanya perlu memanggil variabel mid.
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!