POLITEKNIK NEGERI MALANG

TEKNOLOGI INFORMASI

TEKNIK INFORMATIKA

Nama: Abdul Rahman Hanif Darmawan

NIM: 244107020232

Kelas: TI-1A

Prodi: D4-TEKNIK INFORMATIKA

JOBSHEET V
BRUTE FORCE DAN DIVIDE CONQUER

Menghitung Nilai Faktorial dengan Algoritma Brute Force dan Divide and Conquer

5.2.2 Verifikasi Hasil Percobaan

```
Masukkan nilai: 5
Nilai faktorial 5 menggunakan BF: 120
Nilai faktorial 5 menggunakan DC: 120
```

5.2.3 Pertanyaan

1. Pada base line Algoritma Divide Conquer untuk melakukan pencarian nilai faktorial, jelaskan perbedaan bagian kode pada penggunaan if dan else!
   - Pada if (n == 1) return 1; menggunakan base case, menghentikan rekursi dan pada else return n * faktorialDC(n - 1); memanggil fungsi dengan n-2 hingga mencapai base case
2. Apakah memungkinkan perulangan pada method faktorialBF() diubah selain menggunakan for? Buktikan!
   -
```
public class Faktorial {
    Qodo Gen: Options | Test this method
    int faktorialBF(int n) {
      int fakto = 1, i = 1;
      while (i <= n) {
        fakto *= i;
        i++;
      }
    return fakto;
    }
```
3. Jelaskan perbedaan antara fakto *= i; dan int fakto = n * faktorialDC(n-1);!
   - fakto *=I memperbarui nilai dalam loop
   - int fakto = n * faktorialDC(n-1); memanggil fungsi untuk menyelesaikan perhitungan
4. Buat Kesimpulan tentang perbedaan cara kerja method faktorialBF() dan faktorialDC()!
   - faktorialBF() iterasi lebih efisien dan simpel, faktorialDC menggunakan rekursi

Menghitung Hasil Pangkat dengan Algoritma Brute Force dan Divide and Conquer

5.3.2 Verifikasi Hasil Percobaan

```
Masukkan jumlah elemen : 3
Masukan nilai basis elemen ke-1: 2
Masukan nilai pangkat elemen ke-1: 3
Masukan nilai basis elemen ke-2: 4
Masukan nilai pangkat elemen ke-2: 5
Masukan nilai basis elemen ke-3: 6
Masukan nilai pangkat elemen ke-3: 7
HASIL PANGKAT BRUTEFORCE:
2^3: 8
4^5: 1024
6^7: 279936
HASIL PANGKAT DIVIDE AND CONQUER:
2^3: 8
4^5: 1024
6^7: 279936
```

5.3.3 Pertanyaan

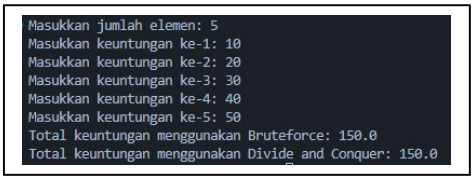
1. Jelaskan mengenai perbedaan 2 method yang dibuat yaitu pangkatBF() dan pangkatDC()!
   - pangkatBF() menggunakan iterasi for loop untuk mengalikan a sebanyak n sedangkan pangkatDC() menggunakan rekursi dengan Divide and Conquer membagi pangkat menjadi dua bagian.
2. Apakah tahap combine sudah termasuk dalam kode tersebut?Tunjukkan!
   -
   ```
   return (pangkatDC(a, n/2) * pangkatDC(a, n/2) * a);
   }else{
   return (pangkatDC(a, n/2) * pangkatDC(a, n/2));
   ```
3. Pada method pangkatBF()terdapat parameter untuk melewatkan nilai yang akan dipangkatkan dan pangkat berapa, padahal di sisi lain di class Pangkat telah ada atribut nilai dan pangkat, apakah menurut Anda method tersebut tetap relevan untuk memiliki parameter? Apakah bisa jika method tersebut dibuat dengan tanpa parameter? Jika bisa, seperti apa method pangkatBF() yang tanpa parameter?
   - Kurang efektif karena atribut nilai dan pangkat sudah ada di kelas, dan bisa dibuat tanpa parameter dengan mengambil langsung atribut nilai dan pangkat dari objek.

   ```
   Qodo Gen: Options | Test this method
   int pangkatBF() {
       int hasil = 1;
       for (int i = 0; i < pangkat; i++) {
           hasil *= nilai;
       }
       return hasil;
   }
   ```
4. Tarik tentang cara kerja method pangkatBF() dan pangkatDC()!

- pangkatBF() mengalikan nilai berulang kali, pangkatDC() membagi pangkat menjadi bagian lebih kecil, dan lebih cepat daripada pangkatBF().

Menghitung Sum Array dengan Algoritma Brute Force dan Divide and Conquer

5.4.2 Verifikasi Hasil Percobaan



5.4.3 Pertanyaan

1. Kenapa dibutuhkan variable mid pada method TotalDC()?
   - Mid digunakan untuk membagi array menjadi dua bagian dalam metode Divide and Conquer agar setiap bagian dapat dihitung totalnya secara rekursif
2. Untuk apakah statement di bawah ini dilakukan dalam TotalDC()?
   ```
   double lsum = totalDC(arr, l, mid);
   double rsum = totalDC(arr, mid+1, r);
   ```
   - Untuk membagi masalah menjadi dua submasalah yaitu menghitung total bagian kiri lsum dan bagian kanan rsum dari array secara rekursif
3. Kenapadiperlukan penjumlahan hasil lsum dan rsum seperti di bawah ini?
   ```
   return lsum+rsum;
   ```
   - Setelah membagi array, hasil dari dua bagian harus dijumlahkan untuk mendapatkan total keseluruhan array
4. Apakah base case dari totalDC()?
   - Base case terjadi saat hanya ada satu elemen dalam array sehingga fungsi langsung mengembalikan nilai elemen tersebut
5. Tarik Kesimpulan tentang cara kerja totalDC()
   - totalDC() menggunakan divide and conquer dengan membagi array menjadi dua bagian, menghitung total setiap bagian secara rekursif, lalu menggabungkan hasilnya

4.5 Latihan Praktikum

NilaiMahasiswa.java

```
package Tugas;

public class NilaiMahasiswa {
    static int findMax(int[] arr, int left, int right) {
        if (left == right) {
            return arr[left];
        }
        int mid = (left + right) / 2;
        int maxLeft = findMax(arr, left, mid);
        int maxRight = findMax(arr, mid + 1, right);
        return Math.max(maxLeft, maxRight);
    }

    static int findMin(int[] arr, int left, int right) {
        if (left == right) {
            return arr[left];
        }
        int mid = (left + right) / 2;
        int minLeft = findMin(arr, left, mid);
        int minRight = findMin(arr, mid + 1, right);
        return Math.min(minLeft, minRight);
    }

    static double calculateAverage(int[] arr) {
        int sum = 0;
        for (int value : arr) {
            sum += value;
        }
        return (double) sum / arr.length;
    }
}
```

MainNilaiMahasiswa.java

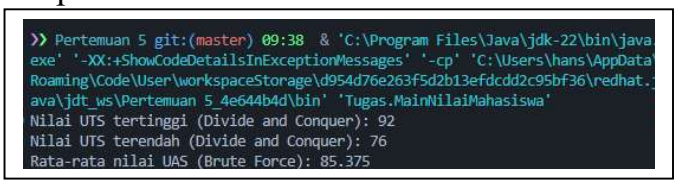
```
package Tugas;

public class MainNilaiMahasiswa {
    public static void main(String[] args) {
        int[] nilaiUTS = {78, 85, 90, 76, 92, 88, 80, 82};
        int[] nilaiUAS = {82, 88, 87, 79, 95, 85, 83, 84};

        int maxUTS = NilaiMahasiswa.findMax(nilaiUTS, 0,
nilaiUTS.length - 1);
        int minUTS = NilaiMahasiswa.findMin(nilaiUTS, 0,
nilaiUTS.length - 1);
        double avgUAS = NilaiMahasiswa.calculateAverage(nilaiUAS);

        System.out.println("Nilai UTS tertinggi (Divide and
Conquer): " + maxUTS);
        System.out.println("Nilai UTS terendah (Divide and Conquer):
" + minUTS);
        System.out.println("Rata-rata nilai UAS (Brute Force): " +
avgUAS);
    }
}
```

Output



Link Github :