

# Penerapan Program *Divide and Conquer* pada Pemilihan Harga Tiket Kereta Api

Muhammad Duta Faturrahman 120140065

Program Studi Teknik Informatika

Jurusan Teknologi Produksi dan Industri

Institut Teknologi Sumatera, Jl. Terusan Ryacudu, Way Huwi, Kec. Jati Agung,

Kabupaten Lampung Selatan, Lampung

muhammad.120140065@student.itera.ac.id

**Abstraksi-**Tiket Kereta Api adalah sebuah kertas yang digunakan untuk melakukan perjalanan kereta api dengan tujuan yang sudah kita tetapkan sebelum membeli tiket tersebut. Fungsi dari tiket kereta api ini adalah sebagai alat bukti bahwa penumpang yang membeli tiket telah mempunyai hak nya untuk menaiki kereta tersebut karena telah membeli tiket sebelumnya. Harga tiket kereta api ini diuntuk berdasarkan jarak tempuh yang dijalani oleh Kereta Api yang akan berjalan nantinya. Algoritma Divide and Conquer dapat digunakan untuk mencari harga tiket kereta api dari yang paling murah sampai yang termahal.

**Keywords**—tiket kereta api; Divide and Conquer; mencari; murah; mahal;

## I. PENDAHULUAN

Kereta api merupakan salah satu sistem transportasi umum yang banyak digunakan masyarakat Indonesia untuk menjalani kehidupan sehari-hari. Stasiun merupakan tempat transit kereta api dan digunakan untuk penumpang yang naik dan turun serta menunggu kereta datang. Selain itu, stasiun merupakan tempat jual beli tiket, penumpang dapat mengisi form pemesanan untuk pemesanan di hari lain, dan untuk pembelian tiket Go Show (keberangkatan langsung), penumpang hanya akan menunjukkan identitasnya saja. Dengan perkembangannya, membeli tiket kereta api tidak harus datang ke stasiun dan bisa dibeli secara online. Namun, pembelian online hanya berlaku untuk tiket kereta bisnis dan eksekutif.

*Divide and Conquer* terbagi menjadi dua kata yaitu Divide dan Conquer. *Divide* berarti memecah masalah menjadi beberapa sub-pertanyaan yang mirip dengan masalah asli tetapi

lebih kecil (idealnya berukuran sama), dan *Conquer* (memecahkan) berarti memecahkan setiap pertanyaan. Berarti (kecil jika langsung, rekursif jika kecil) besar), dan join berarti menggabungkan solusi dari setiap masalah untuk membentuk solusi dari masalah aslinya. Objek masalah dibagi menjadi input atau instance masalah berukuran  $n$ , seperti tabel (array), matriks, eksponensial, polinomial, dll., tergantung pada masalahnya. Setiap kuesioner memiliki karakteristik yang sama (sama tipe) dengan karakteristik pertanyaan aslinya, sehingga metode divide-and-conquer lebih natural diekspresikan dalam skema rekursif.

## II. TEORI DASAR

### A. Algoritma

Kata algoritma menurut latinisasi nama seseorang pakar matematika yaitu Uzbekistan Al Khawārizmi (hidup sekitar abad ke-9), sebagaimana tercantum dalam terjemahan karyanya pada bahasa latin menurut abad ke-12 “Algorithmi de numero Indorum”.

Pada awalnya istilah algoritma merupakan kata yang merujuk pada aturan-aturan aritmetis untuk merampungkan masalah menggunakan sapta numerik arab. Pada abad ke-18, kata ini berkembang sebagai prosedur pemecahan, yang meliputi seluruh mekanisme atau urutan langkah yang kentara & diharapkan untuk merampungkan suatu permasalahan.

Masalah ada dalam ketika akan menuangkan bagaimana proses yang wajib dilewati pada suatu/sebuah sistem (program) bagi personal komputer sebagai akibatnya dalam

ketika eksekusinya, personal komputer bisa bekerja misalnya yang diharapkan. Programmer personal komputer akan lebih nyaman menuangkan mekanisme komputasinya atau urutan langkah proses menggunakan gambaran (diagram alur) diatas kertas.

Algoritma adalah formasi perintah yang sangat berguna bagi kehidupan manusia. Algoritma umumnya dipakai menjadi formasi perintah untuk merampungkan suatu perkara. Algoritma ini mempunyai pelaksanaan yang beragam pada setiap perkara yang ada. Contohnya saja merupakan prosedur pemecahan dengan cara merampungkan suatu aritmatika yang rumit, prosedur pemecahan untuk menghitung luas penampang berdasarkan suatu kabel, atau bahkan untuk menghitung bayaran parkir pada setiap mal.

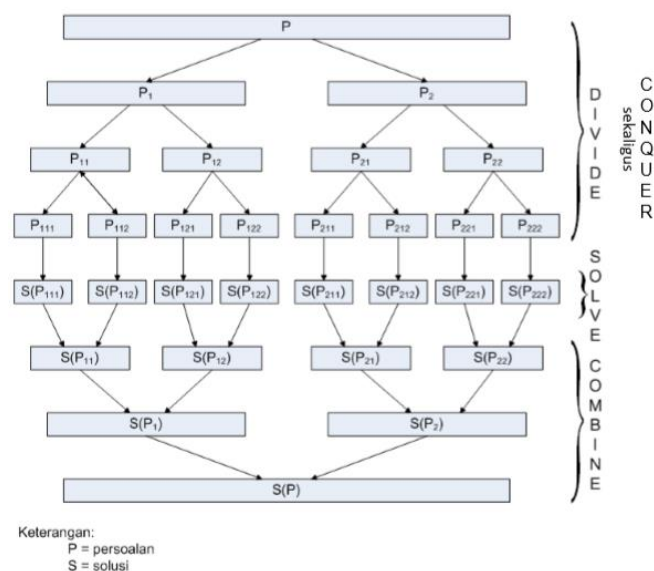
Salah satu pelaksanaan bentuk pemrograman ini merupakan pada bahasa pemrograman yang diklaim bahasa C. Dimana bahasa C ini mempunyai suatu aturan-aturan eksklusif yang sangat krusial sebagai akibatnya pada penggunaanya kita wajib memperhatikan cara memakai aturan tersebut.

## B. Divide and Conquer

Algoritma divide-and-conquer adalah algoritma untuk menyelesaikan masalah komputasi dengan membagi masalah besar menjadi bagian-bagian yang lebih kecil sehingga dapat dengan mudah diselesaikan. Secara konseptual, kata "membagi" dalam nama algoritma ini mengacu pada fase membagi masalah menjadi beberapa subpertanyaan. Sub-isu ini mirip dengan edisi asli, tetapi lebih kecil. Istilah penaklukan di sini mengacu pada tahap pemecahan masalah mencoba memecahkan setiap pertanyaan tersebut sesuai dengan ukuran.

Jika kuesioner masih cukup besar dan tidak dapat diselesaikan secara langsung, kuesioner akan diselesaikan secara rekursif. Ini, tentu saja, juga memengaruhi pemecahan pertanyaan menjadi bagian-bagian yang lebih kecil. Jika tidak, ukurannya dianggap sangat kecil dan masalah kertas diselesaikan secara langsung. Selain kedua tahap tersebut, algoritma ini membutuhkan tahap akhir yaitu tahap combine. Setelah semua pertanyaan kertas berhasil diproses, solusi untuk

setiap pertanyaan perlu digabungkan kembali. Oleh karena itu, kita memiliki solusi lengkap untuk masalah awal.



**Gambar 1.** Ilustrasi Konsep Algoritma *Divide and Conquer*

(sumber :

[https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Divide-and-Conquer-\(2021\)-Bagian1.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Divide-and-Conquer-(2021)-Bagian1.pdf) )

Objek persoalan algoritma ini yang bisa dibagi misalnya tabel atau larik, eksponen, polinom, dan matriks yang merupakan kumpulan tabel. Dalam hal ini, tiap upa-persoalan yang dibagi memiliki karakteristik yang sama dengan karakteristik persoalan semula. Dengan demikian, maka algoritma ini lebih natural dan cocok diungkapkan dalam skema rekursif.

```

procedure DIVIDEandCONQUER(input P : problem, n : integer)
{ Menyelesaikan persoalan P dengan algoritma divide and conquer
  Masukan: masukan persoalan P berukuran n
  Luaran: solusi dari persoalan semula }
Deklarasi
  r : integer

Algoritma
if n ≤ n0 then {ukuran persoalan P sudah cukup kecil }
  SOLVE persoalan P yang berukuran n ini
else
  DIVIDE menjadi r upa-persoalan, P1, P2, ..., Pr, yang masing-masing berukuran n1, n2, ..., nr
  for masing-masing P1, P2, ..., Pr, do
    DIVIDEandCONQUER(Pi, ni)
  endfor
  COMBINE solusi dari P1, P2, ..., Pr menjadi solusi persoalan semula
endif

```

**Gambar 2.** Skema Umum Algoritma *Divide and Conquer*

(sumber :

[https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/20202021/Algoritma-Divide-and-Conquer-\(2021\)-Bagian1.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/20202021/Algoritma-Divide-and-Conquer-(2021)-Bagian1.pdf) )

Untuk skema yang demikian, maka kompleksitas waktu algoritma Divide and Conquer yang demikian dapat dituliskan sebagai berikut.

$$T(n) = \begin{cases} g(n), & n \leq n_0 \\ T(n_1) + T(n_2) \dots + T(n_r) + f(n), & n > n_0 \end{cases}$$

Dalam kompleksitas waktu algoritma  $T(n)$ ,  $g(n)$  merupakan kompleksitas waktu yang diperlukan jika ternyata persoalan yang diberikan merupakan basis rekursi dan bisa dipecahkan secara langsung (solve). Adapun  $T(n_i)$  merupakan kompleksitas waktu yang dimiliki oleh setiap upa-persoalan  $i$  dan  $f(n)$  merupakan kompleksitas waktu penggabungan semua solusi upa-persoalan yang ada. Jika semua upa-persoalan dibagi menjadi dua bagian yang berukuran sama, maka skema rekursif yang dimiliki oleh algoritma ini akan berubah, begitu juga dengan pembagian ke dalam  $n$  bagian yang sama. Pembagian yang demikian akan menghasilkan kompleksitas waktu dan skema rekursif sebagai berikut.

```

procedure DIVIDEandCONQUER(input  $P$  : problem,  $n$  : integer)
{ Menyelesaikan persoalan dengan algoritma divide and conquer
  Masukan: masukan yang berukuran  $n$ 
  Luaran: solusi dari persoalan semula
}
Deklarasi
 $r$  : integer

Algoritma
if  $n \leq n_0$  then {ukuran persoalan sudah cukup kecil}
  SOLVE persoalan  $P$  yang berukuran  $n$  ini
else
  DIVIDE menjadi 2 upa-persoalan,  $P_1$  dan  $P_2$ , masing-masing berukuran  $n/2$ 
  DIVIDEandCONQUER( $P_1$ ,  $n/2$ )
  DIVIDEandCONQUER( $P_2$ ,  $n/2$ )
  COMBINE solusi dari  $P_1$  dan  $P_2$ 
endif

```

**Gambar 3.** Skema Konsep Rekursif yang telah Dimodifikasi

(Sumber :

[https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/20202021/Algoritma-Divide-and-Conquer-\(2021\)-Bagian1.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/20202021/Algoritma-Divide-and-Conquer-(2021)-Bagian1.pdf) )

$$T(n) = \begin{cases} g(n), & n \leq n_0 \\ 2T(n/2) + f(n), & n > n_0 \end{cases}$$

Beberapa permasalahan yang bisa dipecahkan dengan menggunakan algoritma Divide and Conquer antara lain adalah persoalan mencari nilai maksimum dan minimum, menghitung perpangkatan, algoritma pengurutan seperti Quicksort dan

Mergesort, Closest Pair Problem, Convex Hull, perkalian matriks, bilangan bulat besar, atau perkalian dua buah polinom.

Selain algoritma Divide and Conquer, ada algoritma lain dengan konsep yang mirip, yaitu Algoritma Decrease and Conquer. Decrease and Conquer adalah metode perancangan algoritma yang dilakukan dengan mereduksi persoalan menjadi dua upa-persoalan yang lebih kecil, tetapi selanjutnya hanya akan melakukan pemrosesan terhadap sebuah sub-persoalan saja. Dengan kemiripan ini, algoritma ini sering digabungkan dan kemudian dipisahkan karena perbedaan mendasar tersebut.

### C. Kereta Api

Kereta api adalah sarana transportasi berupa kendaraan dengan tenaga gerak, baik berjalan sendiri maupun dirangkaikan dengan kendaraan lainnya, yang akan ataupun sedang bergerak di rel. Kereta api merupakan alat transportasi massal yang umumnya terdiri dari lokomotif (kendaraan dengan tenaga gerak yang berjalan sendiri) dan rangkaian kereta atau gerbong (dirangkaikan dengan kendaraan lainnya). memanfaatkannya secara maksimal sebagai alat transportasi utama angkutan darat baik di dalam kota, antarkota, maupun antarnegara.

## III. PEMBAHASAN

Untuk dapat menggambarkan implementasi dari *Divide and Conquer* pada program tiket kereta api, kita gunakan beberapa contoh harga yang dimasukkan secara acak dan kemudian program *Divide and Conquer* akan melakukan pencarian harga yang termurah sampai yang termahal. Misal disini saya menggunakan 20 data harga yang dimasukkan secara acak.

1. 160.000
2. 153.000
3. 159.000
4. 161.000
5. 152.000
6. 151.000
7. 154.000
8. 156.000

9. 158.000
10. 155.000
11. 165.000
12. 168.000
13. 157.000
14. 162.000
15. 166.000
16. 163.000
17. 165.000
18. 169.000
19. 164.000
20. 170.000

#### A. Penjelasan Algoritma Divide and Conquer

Penyelesaian permasalahan ini akan diselesaikan dengan menggunakan algoritma *Divide and Conquer*. Maka dari itu, hal pertama yang harus dilakukan yaitu membagi persoalan menjadi beberapa upa-persoalan yang mempunyai kemiripan dengan persoalan semula namun berukuran lebih kecil.

Selanjutnya kita selesaikan masing-masing upa-persoalan (secara langsung jika sudah berukuran kecil atau secara rekursif jika masih berukuran besar). Yang terakhir, gabungkan masing-masing upa-persoalan sehingga membentuk solusi persoalan semula.

#### B. Penyelesaian Permasalahan

Untuk dapat menentukan permasalahan yang diminta yaitu mencari harga tiket mulai dari yang termurah sampai harga tiket yang termahal. Maka dari itu, saya telah membuat program C++ untuk mencari solusi tersebut, contoh program C++ nya yaitu sebagai berikut.

```
#include<stdio.h>
#include<conio.h>
using namespace std;

// Deklarasi diluar main program
int a[100];
int max, min;
```

```
// Sub-program untuk mencari harga termurah dan
// harga termahal
void maxmin(int i, int j){
    int max1, min1, mid;
    if (i==j){           // larik berukuran 1 elemen
        max=min=a[i];
    } else if(i==j-1) { // larik berukuran 2 elemen
        if(a[i]>a[j]) {
            max=a[i];
            min=a[j];
        } else {
            max=a[j];
            min=a[i];
        }
    } else { // larik berukuran lebih dari 2 elemen
        mid=(i+j)/2; // membagi larik menjadi 2
        // bagian pada posisi mid
        maxmin(i, mid);
        max1=max;
        min1=min;
        maxmin(mid+1,j);
        if(max<min1) {
            max=max1;
        }
        if (min>min1) {
            min=min1;
        }
    }
}

int main(){
    int i, num;
    printf("\n== Program Pencarian Harga Tiket
    Termurah dan Termahal ==\n");

    printf("Masukkan Banyak Tiket : ");
    scanf("%i", &num);
    printf("\n Masukkan Harga-harganya : ");
    for (i=0; i<num;i++){
        scanf("%i",&a[i]);
    }
    max=a[0];
```

```

min=a[0];
maxmin(0, num-1);
printf("Harga Tiket Termahal : %i\n",max);
printf("Harga Tiket Termurah : %i\n",min);
return 0;
}

```

Maka keluaran programnya seperti berikut ini.

```

== Program Pencarian Harga Tiket Termurah dan Termahal ==
Masukkan Banyak Tiket : 20

Masukkan Harga-harganya : 160000 153000 159000 161000 15
2000 151000 154000 156000 158000 155000 165000 168000 157
000 162000 166000 163000 165000 169000 164000 170000
Harga Tiket Termahal : 170000
Harga Tiket Termurah : 151000
PS D:\Kuliah\C++\00 - Template>

```

Saat kita memasukkan jumlah tiket misalnya 20 tiket, maka program akan meminta masukkan kembali harga-harga tiketnya untuk dicek manakah harga tiket termurah dan harga tiket yang termahal tanpa harus mengurutkan datanya terlebih dahulu.

Saat harga-harga tiket sudah dimasukkan, program secara otomatis membagi jumlah tiket menjadi 2 bagian dengan mencari nilai tengah (mid) sebagai pemisah kedua bagian tersebut. Kemudian program akan mengecek kembali apakah jumlah tiket yang dibagi tersebut masih bisa dibagi lagi atau tidak, jika iya maka program akan melakukan pencarian nilai tengah (mid) untuk membagi kembali jumlah tiket tersebut menjadi 1 elemen atau 2 elemen saja yang tersisa diakhir pembagian jumlah tiket tersebut.

Ketika program selesai melakukan pembagian jumlah tiket sampai menjadi 1 elemen atau 2 elemen kemudian program akan menentukan nilai minimal dan maksimal yang ada didalam elemen tersebut dan kemudian menggabungkan elemen-elemen yang sudah dipisah untuk membentuk solusi yang utuh.

Setelah dilakukan beberapa kali nilai perbandingan miniman dan maksimal, diakhir program akan terlihat nilai mana yang benar-benar minimal dan nilai mana yang benar-

benar maksimal setelah dilakukan penggabungan elemen-elemen tersebut.

### C. Hasil Pencarian

Berdasarkan program yang sudah diuntuk, sebenarnya ada beberapa proses pencarian yang dilakukan oleh program mulai dari membagi beberapa persoalan menjadi kemungkinan yang tidak dapat dibagi lagi yang artinya sudah menjadi solusi akhir yang kemudian akan digabungkan kembali menjadi sebuah solusi yang utuh.

Pada studi kasus kali ini, jumlah tiket 20 dibagi 2 yaitu menjadi 10 tiket dan 10 tiket yang kemudian dibagi lagi menjadi 2 bagian yaitu 5 tiket dan 5 tiket dan kemudian dibagi 2 lagi menjadi 2 bagian dan 3 bagian dan yang terakhir untuk 3 bagian akan dipecah kembali menjadi 1 bagian dan 2 bagian. Untuk yang 1 bagian nilai minimal dan maksimalnya sudah jelas terlihat, tetapi apabila elemen lebih dari satu maka akan dilakukan perbandingan untuk menentukan nilai minimal dan maksimalnya.

Setelah itu bagian-bagian tiket yang sudah dipecah dan dicari nilai miniman dan maksimalnya maka akan digabungkan kembali dengan elemen lainnya dan dilakukan perbandingan kembali terkait nilai minimal dan maksimalnya. Untuk ilustrasi gambar penyelesaian permasalahan dapat dilihat pada link berikut ini (<https://drive.google.com/file/d/1ryHGueSfUYixCtCZFn7OpAJrawDsolt/view?usp=sharing> )

## IV. KESIMPULAN

Berdasarkan pemaparan yang telah dilakukan, dapat ditarik beberapa kesimpulan yaitu algoritma *Divide and Conquer* ini sangat efektif untuk mencari suatu nilai minimal dan maksimal tanpa harus mengurutkan nilai-nilai tersebut, algortima *Divide and Conquer* juga sangat berguna untuk mencari nilai minimal dan maksimal apabila datanya banyak sekali, jadi program akan langsung menemukan solusinya tanpa harus memperhatikan urutan data yang ada.

#### VIDEO LINK AT YOUTUBE

<https://youtu.be/x-MH7F1IPTI>

#### UCAPAN TERIMA KASIH

Atas berkat rahmat Allah SWT Tuhan Yang Maha Esa penulis dapat menyelesaikan karyanya yang berjudul “Penerapan Program *Divide and Conquer* pada Pemilihan Harga Tiket Kereta Api”. Penulis juga mengucapkan terima kasih terhadap kedua orang tua yang selalu mendukung dan menyanyangi penulis, teman-teman penulis yang juga mendukung penulis, dan Bapak Dosen Pengampu Strategi dan Algoritma Kelas RA yang telah membimbing mahasiswanya.

#### REFERENCES

- [1] Munir, Rinaldi. “Algoritma Divide and Conquer (2021) Bagian 1” [https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Divide-and-Conquer-\(2021\)-Bagian1.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Divide-and-Conquer-(2021)-Bagian1.pdf) (Diakses 28 Maret 2022).
- [2] Andikafisma.” Algoritma Divide and Conquer “ <https://andikafisma.wordpress.com/algoritma-divide-and-conquer/> (Diakses 28 Maret 2022)
- [3] creatormedia.my.id. “Rangkuman Algoritma Divide and Conquer” <https://creatormedia.my.id/rangkuman-algoritma-divide-and-conquer/#:~:text=dan%20penelitian%20operasi,-.Divide%20and%20Conquer%20adalah%20metode%20pemecahan%20masalah%20yang%20bekerja%20dengan,menjadi%20solusi%20dari%20masalah%20semula>. (Diakses 28 Maret 2022).
- [4] Wahyup. “Implementasi Algoritma Divide and Conquer” <https://www.awonapa.com/2020/12/implementasi-algoritma-divide-and.html> (Diakses 28 Maret 2022).

#### PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Lampung, 29 Maret 2022



Muhammad Duta Faturrahman 120140065