

Nama : Bilhaq Avi Dewantara
NIM : 120140141
Kelas : RB

Pertemuan ke-1 | Soal 1

Link repl.it: <https://replit.com/@I20I40I4IBILHAQ/P1SOAL1PRAK-ASD#main.cpp>

Info tambahan (opsional) : Program ini merupakan pengimplementasikan fungsi sub-program metode *Binary Searching* dengan menggunakan sub-program biasa, perintah yang diminta ialah mengubah fungsi tersebut menjadi rekursif dengan nilai dan angka yang sudah diketahui.

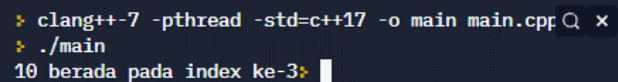
Penjelasan code :

Tujuan dari program ini adalah mencari suatu angka dengan metode *Binary searching* menggunakan fungsi rekursif.

```
#include <iostream>
using namespace std;
int cari(int arr[], int left, int right, int num) { // Deklarasi
fungsi "cari" dengan 4 parameter variabel
    if (left <= right) { // Apabila left kurang dari sama dengan
right maka akan menginisialisasi percabangan
        int mid = (left + right)/2; // kondisi dimana untuk mencari
nilai tengah dari suatu array
        if (arr[mid] == num)
            return mid; // Apabila nilai tengah array sama
dengan angka yang dicari maka akan return mid
        if (arr[mid] > num)
            return cari(arr, left, mid-1, num); // Apabila nilai
tengah lebih dari angka yang dicari maka nilai tengah
array dikurangi 1
        if (arr[mid] < num)
            return cari(arr, mid+1, right, num); // Apabila nilai
tengah kurang dari angka yang dicari maka nilai tengah
array ditambah 1
    }
    return -1; // Kondisi dimana semua kondisi percabangan tidak
terpenuhi
}
```

```
int main(){
    int arr[] = { 2, 3, 4, 10, 40 };
    int x = 10; // x adalah bilangan yg dicari (target)
    // n adalah panjang dari array
    int n = sizeof(arr) / sizeof(arr[0]);
    int result = cari(arr, 0, n-1, x);
    (result == -1) ? cout << x << " tidak ada di array"
    : cout << x << " berada pada index ke-" << result;

    return 0;
}
```



```
❖ clang++-7 -pthread -std=c++17 -o main main.cpp
❖ ./main
10 berada pada index ke-3
```

Fungsi dengan nama “cari” ini memiliki 4 variabel yaitu arr[], left, right, dan num yang mana variabel arr sebagai penampung array yang terdapat pada main(), variabel right sebagai penampung besar/panjang array (arr) tadi, variabel left sebagai pembantu kondisi percabangan agar dapat terpenuhi dengan memberikan nilainya yaitu 0, dan variabel num sebagai penampung angka yang akan dicari entah itu di dalam array atau tidak. Isi dari fungsi “cari”. Isi dari fungsi cari yang pertama adalah percabangan left <= right dimana hal tersebut digunakan untuk apakah semua isi array sudah di cek dengan angka yang dicari. Kemudian deklarasi mid dengan menambahkan variabel (left+right)/2. Selanjutnya kondisi apabila nilai tengah array lebih dari angka yang dicari maka program akan memanggil/menjalankan kembali fungsi “cari” dengan mengurangi nilai tengah array sebanyak 1, Apabila sebaliknya, maka program akan memanggil/menjalankan kembali fungsi tersebut dengan menambahkan nilai tengah sebesar 1 satuan. Apabila terdapat tidak adanya angka yang dicari dalam array maka program akan me-return -1.

Sedangkan pada main() terdapat deklarasi int arr[] dengan 5 angka yang berbeda dapat dilihat dari gambar. Variabel x=10 menandakan bahwa kita akan mencari angka 10 pada arr[] apakah ada atau tidak, kemudian variabel n berguna dalam menganalisis berapa panjang array dari variabel arr[]. Adapun juga variabel result berguna dalam penampung hasil dari fungsi “cari” tadi. Apabila x terdapat di array maka akan keluar output yang tertera pada gambar, apabila tidak ada akan mengeluarkan “tidak ada di array”.

Nama : Bilhaq Avi Dewantara
NIM : 120140141
Kelas : RB

Pertemuan ke-1 | Soal 2

Link repl.it: <https://replit.com/@I20I40I4IBILHAQ/P1SOAL2-PRAK-ASD>

Info tambahan (opsional) : Program ini berfungsi dalam mempersingkat main() dengan cara menggunakan fungsi Sub-Program

Penjelasan code :

Tujuan dibuatnya program ini adalah membantu merevisi program yang telah dibuat oleh Andi karena mempunyai banyak perulangan, untuk itu perintah yang diberikan adalah mempermudah dan memangkas baris” yang ada di main().

```
4 float Rparalel(int n){ // Fungsi mencari Hambatan
   Paralel
5     float input, jumlah = 0;
6     for (int i = 0; i < n; i++)
7     {
8         cout << "Masukkan nilai ke-" << i+1 << " : ";
9         cin >> input;
10        jumlah += 1.0/input; // Rumus Paralel 1/Hambatan
           (1/R1+1/R2+1/R3+...Rn)
11    }
12    jumlah = 1.0/jumlah; // Dibalikkan agar memudahkan
           dalam menghitung Kuat Arus
13    return jumlah;
14 }
15
16 float Rseri(int n){ // Rumus mencari Hambatan Seri
17     float input, jumlah = 0;
18     for (int i = 0; i < n; i++)
19     {
20         cout << "Masukkan nilai ke-" << i+1 << " : ";
21         cin >> input;
22         jumlah += input; // Rumus Seri (R1+R2+R3+...Rn)
23     }
24     return jumlah;
25 }
```

```
27 float Baterai(int n){ // Rumus mencari Tegangan
28     float input, jumlah = 0;
29     for (int i = 0; i < n; i++)
30     {
31         cout << "Masukkan nilai ke-" << i+1 << " : ";
32         cin >> input;
33         jumlah += input; // Menjumlahkan tegangan di
           setiap baterai
34     }
35     return jumlah;
36 }
37
38 float KuatArus(float V, float Rs, float Rp){ // Rumus
   mencari Kuat Arus
39     float I = 0;
40     I = V / (Rs+Rp);
41     return I;
42 }
```

```
> clang++-7 -pthread -std=c++17 -o main main.cpp
> ./main
Masukkan jumlah hambatan tersusun paralel : 3
Masukkan nilai ke-1 : 1
Masukkan nilai ke-2 : 3
Masukkan nilai ke-3 : 5
Jumlah nilai hambatan paralel : 0.652174
Masukkan jumlah hambatan tersusun seri : 3
Masukkan nilai ke-1 : 5
Masukkan nilai ke-2 : 3
Masukkan nilai ke-3 : 1
Jumlah nilai hambatan seri : 9
Masukkan jumlah baterai yang tersusun seri : 3
Masukkan nilai ke-1 : 2
Masukkan nilai ke-2 : 4
Masukkan nilai ke-3 : 6
Jumlah nilai tegangan listrik : 12
Nilai kuat arus pada Amperemeter adalah : 1.24324
```

Untuk mempersingkat main() dibuatlah Fungsi Sub-Program mencari Paralel, Seri, dan Baterai/Tegangan dengan meng-input-kan suatu nilai menggunakan perulangan. Pada fungsi Rparalel digunakan rumus $1.0/\text{input}$ dimana itu merupakan rumus umum mencari paralel. Sedangkan fungsi Rseri dan Baterai hanya dengan menjumlahkan setiap nilai input di perulangannya saja. Kita pindah ke sisi main(), disana dapat dilihat bahwa terdapat deklarasi variabel yang kemudian pengguna diminta diberikan batasan-batasan di setiap hambatan/nilai yang mana batasan tersebut akan menjalankan fungsi Rparalel, Rseri, dan

Baterai. Hasil return setiap fungsi tersebut akan ditampung ke beberapa variabel seperti Rp, Rs, dan V. Kemudian dari hasil ketiga variabel tersebut dicarilah nilai Kuat Arus dengan menjalankan Fungsi KuatArus pada Sub-Program dengan rumus $I=V/(Rs+Rp)$ dengan I sebagai variabelnya.