

**LAPORAN PRAKTIKUM STRUKTUR  
DATA DAN ALGORITMA**

**MODUL 8  
“ALGORITMA SEARCHING”**



**DISUSUN OLEH :  
RIZKY PERLINTA SEMBIRNG  
2311102061**

**DOSEN  
WAHYU ANDI SAPUTRA, S.PD., M.PD.**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA  
FAKULTAS INFORMATIKA  
INSTITUT TEKNOLOGI TELKOM PURWOKERTO  
2024**

## **A. DASAR TEORI**

### **PENGERTIAN SEARCHING**

Searching merupakan kegiatan mencari data yang akan dibutuhkan. Searching dalam pemrograman dapat dilakukan untuk mencari data yang berada pada memory komputer. Dalam kehidupan sehari-hari kita sering melakukan Searching seperti pada saat mencari data maupun informasi yang ada pada internet. Terdapat macam-macam metode yang dapat digunakan dalam Searching, antara lain:

- Sequential Search
- Binary Search

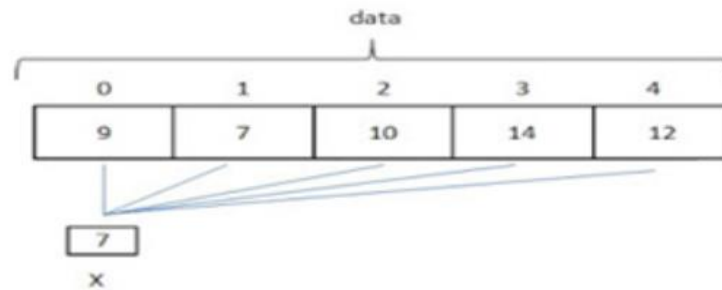
#### **1. Sequential Search**

Sequential Search adalah sebuah metode pencarian data dalam array dengan cara membandingkan data yang akan dicari dengan data yang terdapat dalam array secara berurutan. Pencarian data dengan menggunakan metode Sequential Search lebih efektif untuk mencari data dalam posisi tidak terurut atau acak.

Proses Sequential Search dapat dijelaskan sebagai berikut:

1. Menentukan data yang akan dicari.
2. Membaca data array satu per satu secara sekuensial
3. Membaca data mulai dari data pertama sampai dengan data terakhir, kemudian data yang dicari akan dibandingkan dengan masing-masing data yang terdapat dalam array.
  - a. Apabila data yang dicari ditemukan maka kita dapat membuat pernyataan bahwa data telah ditemukan.
  - b. Jika data yang dicari tidak ditemukan maka kita dapat membuat pernyataan bahwa data tidak ditemukan.

### Ilustrasi dari Proses Sequential Search:



Data yang dicari yaitu 7, lalu disimpan dalam variabel x, kemudian akan dibandingkan satu per satu secara kuensial terhadap data yang terdapat dalam array. Jika ditemukan data pada array yang sama dengan data yang dicari maka data ditemukan.

Kelebihan dan Kekurangan Sequential Searching yaitu:

- Kelebihan Sequential Searching adalah Sequential Searching lebih mudah dalam proses implementasi dalam pemrograman.
- Kekurangan Sequential Searching adalah data yang terdapat dalam suatu array jumlahnya sangat banyak, maka waktu yang diperlukan dalam membandingkan data yang dicari dengan jumlah data dalam suatu array membutuhkan waktu yang lebih lama.

## 2. Binary Search

Metode Binary Search adalah suatu metode pencarian data dengan cara mengelompokkan array menjadi bagian-bagian. Binary Search hanya dapat diterapkan pada data yang telah terurut baik ascending maupun descending dalam suatu array.

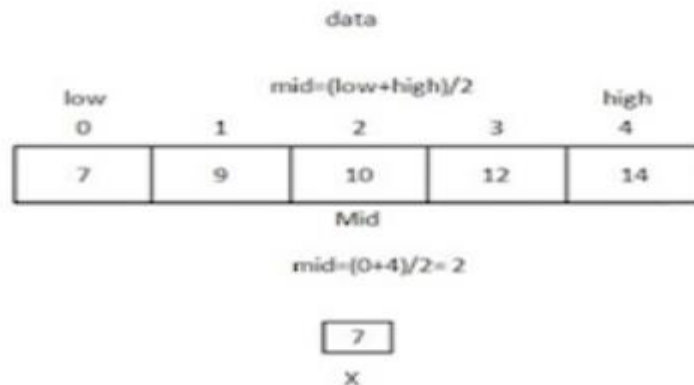
Proses Binary Search dengan data ascending dan telah urut:

1. Membuat perulangan kemudian menentukan posisi low yaitu posisi yang menandakan index paling rendah kemudian menentukan posisi high, setelah itu mencari posisi mid atau tengah =  $(high + low)/2$
2. Membandingkan data yang dicari dengan nilai yang berada di posisi mid

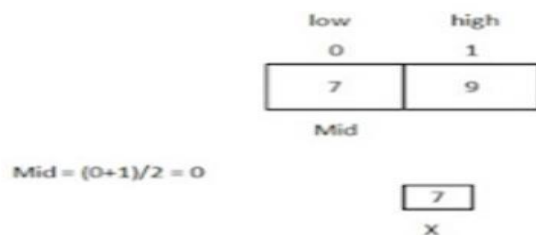
atau tengah.

3. Jika data yang dicari sama dengan nilai yang ada di posisi mid atau tengah berarti data ditemukan.
4. Jika data yang dicari lebih kecil dari nilai yang terdapat pada mid maka pencarian data akan dilakukan dibagian kiri mid dengan melakukan pembandingan. Dengan kondisiposisi high berubah yaitu (mid - 1) dan posisi low tetap.
5. Jika data yang dicari lebih besar dari nilai yang terdapat pada mid maka pencarian data akan dilakukan di bagian kanan dari mid dengan posisi low yang berubah yaitu (mid + 1) dan posisi high tetap.

#### Ilustrasi dari Proses Binary Search:



Setelah menentukan low dan high lalu menentukan mid. perhitungan  $\text{mid} = (\text{low} + \text{high}) / 2$  jadi  $\text{mid} = (0 + 4) / 2$ , artinya mid berada pada data (2). Data yang dicari kemudian ditampung di variabel x lalu dibandingkan dengan data atau nilai yang berada di mid. data yang dicari adalah  $7 < 10$  kemudian pencarian data dilakukan di bagian kiri mid.



Pencarian di bagian kiri mid masih dalam perulangan yang sama, namun indeksnya dipersempit. Karena data yang dicari lebih kecil dari data mid yang sebelumnya maka posisi high akan berubah yaitu  $(mid - 1)$  yang sebelumnya dan low tetap pada posisi yang sama. Kemudian menentukan  $mid = (0+1)/2 = 0$  artinya mid sekarang terletak di data (0). lalu data yang dicari dibandingkan dengan mid.  $7=7$  artinya data telah ditemukan.

#### Kelebihan dan Kekurangan Binary Search:

1. Kelebihan Binary Search adalah tidak perlu membandingkan data yang dicari dengan seluruh data array yang ada, hanya cukup melalui titik tengah kemudian kita bias menentukan ke mana selanjutnya mencari data yang akan dicari.
2. Kekurangan Binary Search adalah penerapannya agak sedikit lebih rumit dikarenakan tidak dapat digunakan pada data array yang masih acak, sehingga harus melakukan sorting terlebih dahulu dalam penerapannya.

## B. Guided

### Guided 1.

Buatlah sebuah project dengan menggunakan sequential search sederhana untuk melakukan pencarian data

Source Code:

```
#include <iostream>
using namespace std;

int main(){
    int n = 10;
    int data[n] = {9, 4, 1, 7, 5, 12, 4, 13, 4, 10};
    int cari = 10;
    bool ketemu = false;
    int i;

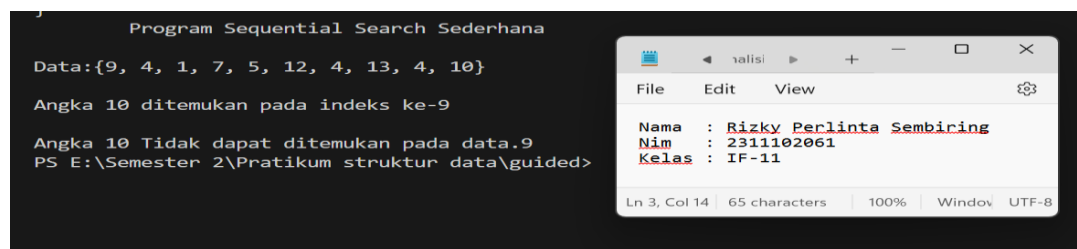
    //algoritma sequential search
    for (i = 0; i < n; i++){
        if (data[i] == cari) {
            ketemu = true;
            break;
        }
    }

    cout << "\tProgram Sequential Search Sederhana\n " << endl;
    cout << "Data:{9, 4, 1, 7, 5, 12, 4, 13, 4, 10}" << endl;

    if (ketemu) {
        cout << "\nAngka " << cari << " ditemukan pada indeks ke-" <<
        i << endl;
        cout << "\nAngka " << cari << " Tidak dapat ditemukan pada
        data." << i << endl;
    }

    return 0;
}
```

Screenshot Output:



**Deskripsi Program:**Program diatas adalah implementasi dari algoritma searching. Pada program diatas menggunakan searching jenis sequential search. Program memiliki sebanyak 10 data, dan setelah itu user menginputkan data yang ingin dicari. Kemudian, program menggunakan loop for untuk memeriksa setiap elemen dalam array. Jika angka yang dicari ditemukan, program mencetak indeks tempat angka tersebut berada.

Guided 2.

Buatlah sebuah project untuk melakukan pencarian data dengan menggunakan Binary Search.

Source Code:

```
#include <iostream>
using namespace std;
#include <conio.h>
#include <iomanip>
int DATA[7] = {1, 8, 2, 5, 4, 9, 7};
int cari;
void selection_sort()
{
    int temp, min, i, j;
    for (i = 0; i < 7; i++)
    {
        min = i;
        for (j = i + 1; j < 7; j++)
        {
            if (DATA[j] < DATA[min])
            {
                min = j;
            }
        }
        temp = DATA[i];
        DATA[i] = DATA[min];
        DATA[min] = temp;
    }
}
void binarysearch()
{
    // searching
```

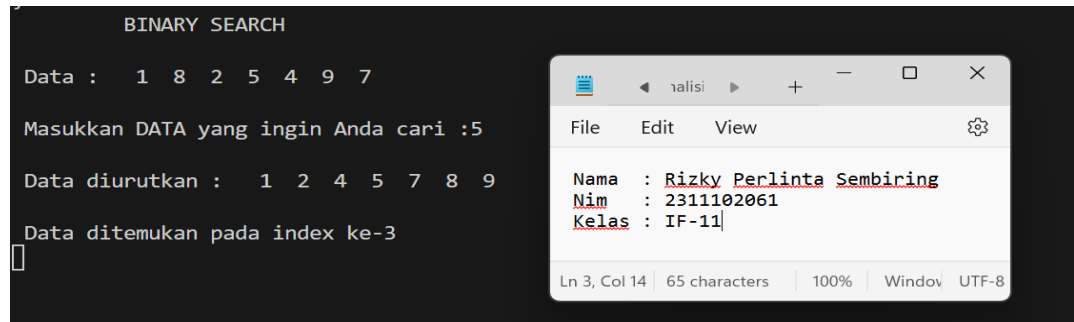
```

int awal, akhir, tengah, b_flag = 0;
awal = 0;
akhir = 7;
while (b_flag == 0 && awal <= akhir)
{
    tengah = (awal + akhir) / 2;
    if (DATA[tengah] == cari)
    {
        b_flag = 1;
        break;
    }
    else if (DATA[tengah] < cari)
        awal = tengah + 1;
    else
        akhir = tengah - 1;
}
if (b_flag == 1)
    cout << "\n Data ditemukan pada index ke-"<<tengah<<endl;
else cout << "\n Data tidak ditemukan\n";
}
int main()
{
    cout << "\t BINARY SEARCH " << endl;
    cout << "\n Data : ";
    // tampilkan DATA awal
    for (int x = 0; x < 7; x++)
        cout << setw(3) << DATA[x];
    cout << endl;
    cout << "\n Masukkan DATA yang ingin Anda cari :";
    cin >> cari;
    cout << "\n Data diurutkan : ";
    // urutkan DATA dengan selection sort
    selection_sort();
    // tampilkan DATA setelah diurutkan
    for (int x = 0; x < 7; x++)
        cout << setw(3) << DATA[x];
    cout << endl;
    binarysearch();
    _getche();
    return EXIT_SUCCESS;
}

```



Screenshot Output:



The screenshot shows two windows. The left window is a terminal titled "BINARY SEARCH" with a dark background. It displays the following text: "Data : 1 8 2 5 4 9 7", "Masukkan DATA yang ingin Anda cari :5", "Data diurutkan : 1 2 4 5 7 8 9", and "Data ditemukan pada index ke-3". The right window is a text editor titled "halisi" with a light background. It contains the following text: "Nama : Rizky Perlinta Sembiring", "Nim : 2311102061", and "Kelas : IF-11". The text editor window also shows a menu bar with "File", "Edit", and "View", and a status bar at the bottom indicating "Ln 3, Col 14", "65 characters", "100%", "Window", and "UTF-8".

**Deskripsi Program:** Deskripsi Program: Program diatas adalah implementasi dari algoritma searching. Pada program diatas menggunakan searching jenis sequential search. Pada program tersebut memiliki 7 data diurutkan menggunakan fungsi `selection\_sort`. Fungsi `binarysearch` mencari angka dalam array yang telah diurutkan dengan membagi array menjadi dua bagian hingga angka ditemukan atau semua elemen diperiksa. Pada output akan menampilkan data sebanyak 7 data dan user disuruh memasukkan data yang ingin dicari. Contoh ketika user menginputkan nilai 5 maka data terlebih dahulu diurutkan setelah itu maka program akan mencari data tersebut ada di uruta/indeks ke berapa.

### C. UNGUIDED

**Unguided 1:** Buatlah sebuah program untuk mencari sebuah huruf pada sebuah kalimat yang sudah di input dengan menggunakan Binary Search

**Source Code:**

```
#include <iostream>
#include <conio.h>
#include <iomanip>

using namespace std;

string kalimat;
char c;

void toLower() {
    string temp;
    for (int i = 0; i < kalimat.length(); i++) {
        temp += tolower(kalimat[i]);
    }
    kalimat = temp;
}

void selection_sort()
{
    int min, i, j;
    char temp;
    toLower();
    for (i = 0; i < kalimat.length(); i++)
    {
        min = i;
        for (j = i + 1; j < kalimat.length(); j++)
        {
            if (kalimat[j] < kalimat[min])
            {
                min = j;
            }
        }
        temp = kalimat[i];
        kalimat[i] = kalimat[min];
        kalimat[min] = temp;
    }
}

void binarysearch()
{
    int awal, akhir, tengah, b_flag = 0;
    awal = 0;
    akhir = kalimat.length();
```

```

while (b_flag == 0 && awal <= akhir)
{
    tengah = (awal + akhir) / 2;
    if (kalimat[tengah] == c)
    {
        b_flag = 1;
        break;
    }
    else if (kalimat[tengah] < c)
        awal = tengah + 1;
    else
        akhir = tengah - 1;
}
if (b_flag == 1)
    cout << "\n Karakter '" << c << "' ditemukan pada index ke - " << tengah << endl;
else
    cout << "\nData tidak ditemukan\n";
}
int main()
{
    cout << "\t BINARY SEARCH " << endl;
    cout << "Masukkan kalimat : ";
    getline(cin, kalimat);
    cout << "\nMasukkan karakter yang ingin Anda cari : ";
    cin >> c;
    c = tolower(c);
    cout << "\nKalimat yang diurutkan berdasarkan karakter : ";
    selection_sort();
    for (int x = 0; x < kalimat.length(); x++)
        cout << kalimat[x];
    cout << endl;
    binarysearch();
    return EXIT_SUCCESS;
}

```

#### Screenshot Output:

```

        BINARY SEARCH
Masukkan kalimat : rizky perlinta sembiring

Masukkan karakter yang ingin Anda cari : k

Kalimat yang diurutkan berdasarkan karakter :  abeegiiiiklmnprrrstyz

Karakter 'k' ditemukan pada index ke - 11

```

**Deskripsi Program:** program diatas adalah searching jenis binary search untuk huruf berada pada indeks ke berapa. Contoh saya menginputkan kalimat rizky perlinta sembiring dan saya akan mencari huruf k berada pada indeks ke berapa. Program tersebut terlebih dahulu akan mengurutkan huruf a samapi huruf z. Dan setelah itu maka akan dicari huruf yang ingin dicari oleh user.

**Unguided 2:** Buatlah sebuah program yang dapat menghitung banyaknya huruf vocal dalam sebuah kalimat!

**Source Code:**

```
#include <iostream>
using namespace std;
int main()
{
    string kalimat;
    int jumlah = 0;

    cout << "Jumlah Huruf Vokal" << endl;
    cout << "Masukkan kalimat : ";
    getline(cin, kalimat);

    for (int i = 0; i < kalimat.length(); i++) {
        char c = tolower(kalimat[i]);
        if (c == 'a' || c == 'i' || c == 'u' || c == 'e' || c ==
'o') jumlah++;
    }

    cout << "\nKalimat yang dimasukkan memiliki " << jumlah << "
huruf vokal.";
    return 0;
}
```

**Screenshot Output:**

```
Jumlah Huruf Vokal
Masukkan kalimat : rizky perlinta sembiring

Kalimat yang dimasukkan memiliki 7 huruf vokal.
```

**Deskripsi Program:** program diatas adalah program untuk mencari jumlah huruf vokal. User disuruh untuk menginputkan kalimat yang berisi bebas setelah diinputkan maka program akan menjumlahkan huruf vokal yang berada pada kalimat tersebut.

**Unguided 3:** Diketahui data = 9, 4, 1, 4, 7, 10, 5, 4, 12, 4. Hitunglah berapa banyak angka 4 dengan menggunakan algoritma Sequential Search!

**Source Program:**

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

int DATA[10]={ 9, 4, 1, 4, 7, 10, 5, 4, 12, 4};

void sequentialsearch(int angka){
    int temp=0;
    for (int i = 0; i < 10; i++)
    {
        if (DATA[i] == angka)
        {
            temp=temp+1;
        }
    }
}

cout<<" angka "<<angka<<" berjumlah "<< temp<<" dalam array data";
}

int main(){
    int angka=4;
    cout<<"Program Sequential Search Sederhana"<<endl;
    cout<<"Data: {9, 4, 1, 4, 7, 10, 5, 4, 12, 4} "<<endl;
    cout<<"Mencari Angka 4 di dalam data"<<endl;
    sequentialsearch(angka);
}
```

**Screenshot Output:**

```
Program Sequential Search Sederhana
Data: {9, 4, 1, 4, 7, 10, 5, 4, 12, 4}
Mencari Angka 4 di dalam data
    angka 4 berjumlah 4 dalam array data
```

**Deskripsi Program:** Kode ini merupakan implementasi sederhana dari algoritma pencarian sekuensial. Kode ini mendefinisikan sebuah array global bernama ``DATA`` yang berisi 10 elemen. Fungsi ``sequentialsearch`` bertugas menghitung berapa kali angka tertentu muncul dalam array tersebut dengan menggunakan loop untuk memeriksa setiap elemen. Fungsi ini kemudian menampilkan jumlah kemunculan angka tersebut di konsol. Pada fungsi ``main``, angka yang akan dicari (angka 4) dideklarasikan, informasi awal ditampilkan, dan ``sequentialsearch`` dipanggil untuk mencari angka 4 dalam array serta menampilkan hasilnya.

#### **D. Kesimpulan**

Searching adalah operasi penting dalam pemrograman C++, dengan berbagai teknik yang sesuai untuk berbagai jenis data dan kebutuhan. Linear Search sederhana dan efektif untuk dataset kecil atau yang tidak terurut, sementara Binary Search menawarkan kinerja cepat untuk data yang terurut. Standard Library C++ menyediakan fungsi yang memudahkan seperti `std::find` dan `std::binary_search`, dan hashing memungkinkan pencarian waktu konstan untuk pasangan key-value. Dengan memahami dan menggunakan teknik-teknik ini, pengembang dapat mengelola dan mengambil data secara efisien dalam aplikasi mereka.

#### **E. Referensi**

[1] Asisten Pratikum “Modul 8 Algoritma Searching”, Learning Management System, 2024.

[2] BOCAH ALPRO, “Searching dalam Bahasa Pemrograman C++”. (2018), diakses pada 29 mei, dari

<https://sinualpro.blogspot.com/2018/12/searching-dalam-bahasa-pemrograman-c.html>