

**LAPORAN PRAKTIKUM**  
**STRUKTUR DATA DAN ALGORITMA**  
**MODUL 8**  
**ALGORITMA SEARCHING**



**Disusun Oleh**  
**Faqih Abdullah**  
**2311102048**

**Dosen Pengampu**  
**Wahyu Andi Saputra**

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

## A. DASAR TEORI

### 1 Binary search

Pengertian binary search adalah algoritma yang digunakan untuk mencari elemen tertentu dalam suatu himpunan data terurut. Metode ini bekerja dengan membagi himpunan data menjadi dua bagian, kemudian memeriksa apakah elemen yang dicari terletak di bagian kiri atau kanan. Langkah ini diulang secara berulang hingga elemen yang dicari ditemukan atau sampai bagian yang diperiksa tidak lagi dapat dibagi.

Prinsip dasar dari binary search adalah mengurangi ruang pencarian secara eksponensial setiap langkahnya. Dibandingkan dengan metode pencarian linier, binary search dapat mencapai efisiensi yang jauh lebih tinggi, terutama pada himpunan data yang besar.

Jika elemen tengah lebih besar, maka pencarian dilanjutkan di bagian sebelumnya. Sebaliknya, jika elemen tengah lebih kecil, maka pencarian dilanjutkan di bagian setelahnya. Proses ini diulang hingga elemen yang dicari ditemukan atau sampai tidak ada lagi elemen yang perlu diperiksa.

Binary search adalah algoritma yang efisien dan sering digunakan dalam berbagai konteks, seperti pengembangan perangkat lunak, basis data, dan pemrosesan data. Dengan pemahaman yang baik tentang binary search, kita dapat meningkatkan kecepatan dan efisiensi pencarian dalam aplikasi yang kita buat.



Cara kerja binary search diatas sebagai berikut:

- Terdapat sebuah array menampung 7 elemen seperti ilustrasi diatas. Nilai yang akan dicari pada array tersebut adalah 13.

- Jadi karena konsep dari Binary Search ini adalah membagi array menjadi dua bagian, maka cari nilai tengahnya dlu. Total elemen dibagi 2 yaitu  $7/2 = 4.5$  dan dibulatkan jadi 4.
- Maka elemen ke empat pada array adalah nilai tengahnya, yaitu angka 9 pada indeks ke 3.
- Kemudian cek apakah  $13 > 9$  atau  $13 < 9$ ?
- 13 lebih besar dari 9, maka kemungkinan besar angka 13 berada setelah 9 atau di sebelah kanan. Selanjutnya kita cari ke kanan dan kita dapat mengabaikan elemen yang ada di kiri.
- Setelah itu kita cari lagi nilai tengahnya, didapatlah angka 14 sebagai nilai tengah. Lalu, kita bandingkan apakah  $13 > 14$  atau  $13 < 14$ ?
- Ternyata 13 lebih kecil dari 14, maka selanjutnya kita cari ke kiri.
- Karna tersisa 1 elemen saja, maka elemen tersebut adalah nilai tengahnya. Setelah dicek ternyata elemen pada indeks ke-4 adalah elemen yang dicari, maka telah selesai proses pencariannya.

## 2 Sequential search

Sequential Search merupakan salah satu algoritma pencarian data yang biasa digunakan untuk data yang berpola acak atau belum terurut. Sequential search juga merupakan teknik pencarian data dari array yang paling mudah, dimana data dalam array dibaca satu demi satu dan diurutkan dari index terkecil ke index terbesar, maupun sebaliknya. Konsep Sequential Search yaitu:

- Membandingkan setiap elemen pada array satu per satu secara berurut.
- Proses pencarian dimulai dari indeks pertama hingga indeks terakhir.
- Proses pencarian akan berhenti apabila data ditemukan. Jika hingga akhir array data masih juga tidak ditemukan, maka proses pencarian tetap akan dihentikan.
- Proses perulangan pada pencarian akan terjadi sebanyak jumlah N elemen pada array.

Cara kerja sequential search yaitu

- $i \leftarrow 0$
- $ketemu \leftarrow false$
- Selama (tidak ketemu) dan  $(i \leq N)$  kerjakan baris 4
- Jika  $(Data[i] = x)$  maka  $ketemu \leftarrow true$ , jika tidak  $i \leftarrow i + 1$
- Jika (ketemu) maka i adalah indeks dari data yang dicari, jika tidak data tidak ditemukan.

## B. GUIDED

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

Source code

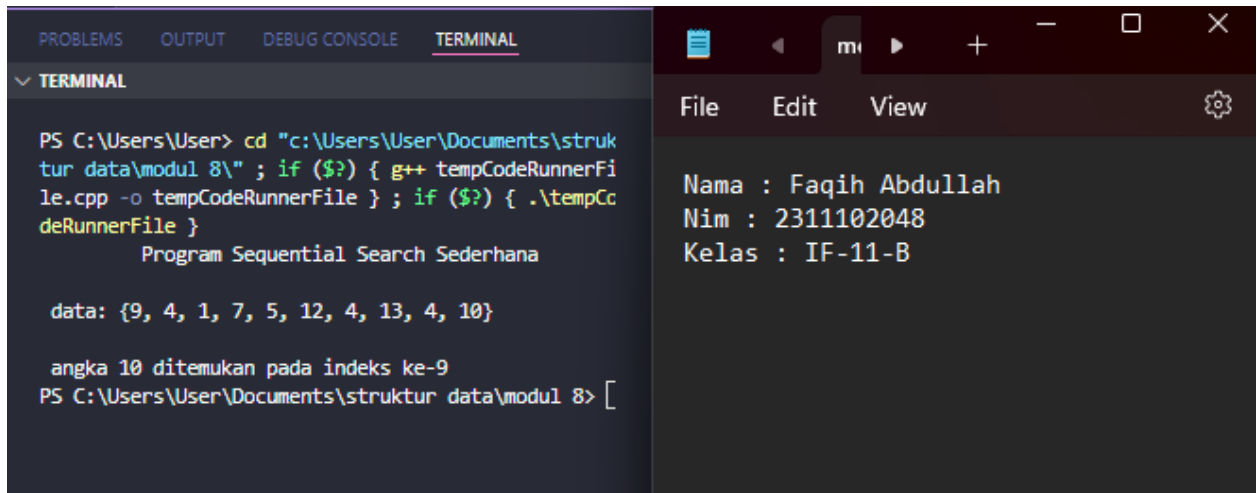
```
#include <iostream>
using namespace std;

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

    if (ketemu)
    {
        cout << "\n angka " << cari << " ditemukan pada indeks ke-" << i << endl;
    }
    else
    {
        cout << cari << " tidak dapat ditemukan pada data." << endl;
    }
    return 0;
}
```

Screenshot Output



```
PS C:\Users\User> cd "c:\Users\User\Documents\struktur data\modul 8\" ; if ($?) { g++ tempCodeRunnerFile.cpp -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
Program Sequential Search Sederhana

data: {9, 4, 1, 7, 5, 12, 4, 13, 4, 10}

angka 10 ditemukan pada indeks ke-9
PS C:\Users\User\Documents\struktur data\modul 8> [
```

Nama : Faqih Abdullah  
Nim : 2311102048  
Kelas : IF-11-B

### Deskripsi program

Program ini menggunakan array untuk menyimpan data dan menggunakan variable *n* untuk menyimpan Panjang array. Array data digunakan untuk menyimpan data yang akan dicari.

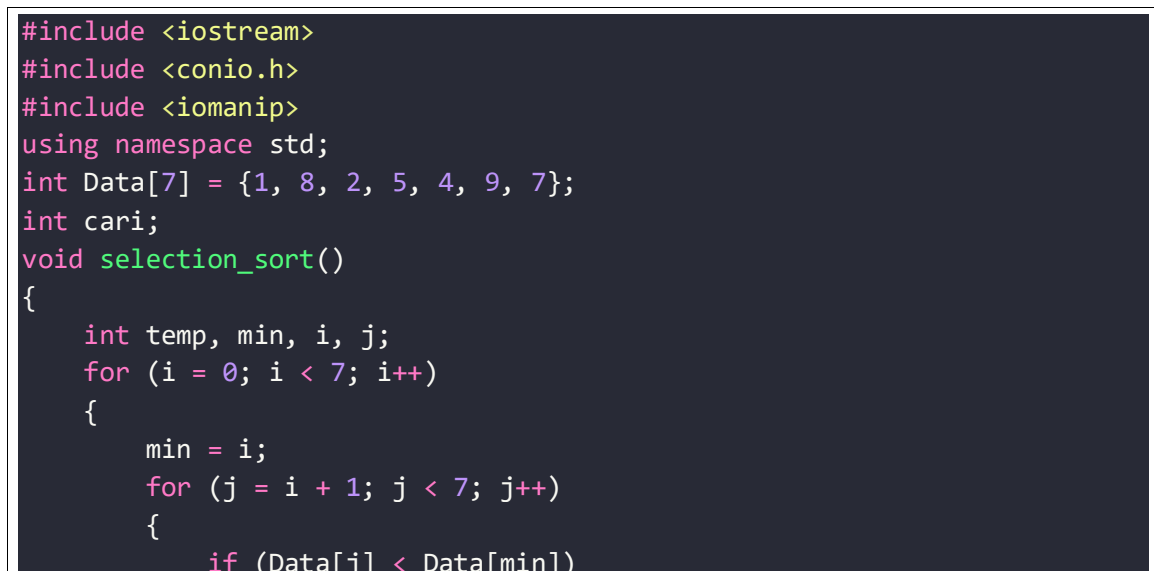
Variable *cari* digunakan untuk menyimpan angka yang ingin dicari, dan variable *ketemu* digunakan untuk menyimpan status apakah angka yang dicari ditemukan atau tidak.

Variable *i* digunakan sebagai indeks dalam loop.

Program akan melakukan loop melalui array data dan akan mengecek apakah angka pada indeks tersebut sama dengan angka yang ingin dicari. Jika sama, maka variable *ketemu* akan diubah menjadi true dan loop akan dihentikan dengan menggunakan pernyataan *break*.

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

### Source code



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

```

PS C:\Users\User\Documents\struktur data\modul 8> c
d "c:\Users\User\Documents\struktur data\modul 8\"
; if ($?) { g++ tempCodeRunnerFile.cpp -o tempCodeR
unnerFile } ; if ($?) { .\tempCodeRunnerFile }
    BINARY SEARCH

Data :  1  8  2  5  4  9  7

Masukkan data yang ingin Anda cari : 2

Data diurutkan :  1  2  4  5  7  8  9

Data ditemukan pada index ke-1
2
PS C:\Users\User\Documents\struktur data\modul 8> [

```

Nama : Faqih Abdullah  
Nim : 2311102048  
Kelas : IF-11-B

Deskripsi program

Program diatas adalah program pencarian data menggunakan binary search. Program ini menggunakan algoritma selection short untuk mengurutkan array data sebelum melakukan pencarian

program dimulai dengan mendefinisikan array data yang berisi beberapa elemen dan variabel cari yang digunakan untuk menyimpan elemen yang ingin dicari.

Fungsi `selection_sort()` digunakan untuk mengurutkan array Data menggunakan algoritma Selection Sort. Fungsi ini menggunakan teknik pemilihan secara berurutan untuk mengurutkan array.

Fungsi `binarysearch()` digunakan untuk melakukan pencarian binary. Pencarian dilakukan dengan menggunakan algoritma Binary Search. Pencarian dimulai dengan menentukan indeks awal dan indeks akhir. Selanjutnya, nilai tengah dihitung dan dibandingkan dengan elemen yang ingin dicari. Jika elemen yang dicari sama dengan elemen pada indeks tengah, maka pencarian berhasil dan indeks tengah ditampilkan. Jika tidak, maka indeks awal atau indeks akhir diperbarui tergantung pada apakah elemen pada indeks tengah lebih kecil atau lebih besar dari elemen yang ingin dicari.

### C. 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 <string>
using namespace std;

int binarySearch(string str, char c, int L, int r) {
    if (r >= L) {
        int mid = L + (r - L) / 2;
        if (str[mid] == c)
            return mid;
        if (str[mid] > c)
            return binarySearch(str, c, L, mid - 1);
        return binarySearch(str, c, mid + 1, r);
    }
    return -1;
}

int main() {
    string kalimat;
    char cari;
    cout << "Masukkan kalimat: ";
    getline(cin, kalimat);
    cout << "Masukkan huruf yang ingin dicari: ";
    cin >> cari;
    int result = binarySearch(kalimat, cari, 0, kalimat.length() - 1);
    if (result == -1)
        cout << "Huruf tidak ditemukan dalam kalimat" << endl;
    else
        cout << "Huruf ditemukan pada indeks " << result << endl;
    return 0;
}
```

Screenshot output



The image shows a C++ program being executed in a terminal window. The program prompts the user to enter a word and a character to search for. In the first run, the word 'bakso' and character 'k' are entered, and the program outputs 'Huruf ditemukan pada indeks 2'. In the second run, the word 'bakso' and character 'y' are entered, and the program outputs 'Huruf tidak ditemukan dalam kalimat'. To the right, an IDE window shows the program's output: 'Nama : Faqih Abdullah', 'Nim : 2311102048', and 'Kelas : IF-11-B'.

```
PS C:\Users\User\Documents\struktur data\modul 8> c
d "c:\Users\User\Documents\struktur data\modul 8\"
; if ($?) { g++ unguided.cpp -o unguided } ; if ($?
) { .\unguided }
Masukkan kalimat: bakso
Masukkan huruf yang ingin dicari: k
Huruf ditemukan pada indeks 2
PS C:\Users\User\Documents\struktur data\modul 8> c
d "c:\Users\User\Documents\struktur data\modul 8\"
; if ($?) { g++ unguided.cpp -o unguided } ; if ($?
) { .\unguided }
Masukkan kalimat: bakso
Masukkan huruf yang ingin dicari: y
Huruf tidak ditemukan dalam kalimat
PS C:\Users\User\Documents\struktur data\modul 8> [
```

Nama : Faqih Abdullah  
Nim : 2311102048  
Kelas : IF-11-B

Ln 3, Col 16 | 54 characters | 100% | Unix (LF) | UTF-8

### Deskripsi program

Program diatas adalah program pencarian huruf pada sebuah kalimat menggunakan Binary Search. Program ini menerima input dari pengguna yaitu kalimat dan huruf yang ingin dicari.

Program dimulaidengan mendefinisikan fungsi `binarySearch` yang menerima kalimat, huruf yang ingin dicari, indeks kiri, dan indeks kanan sebagai parameter. Fungsi ini menggunakan algoritma Binary Search untuk mencari huruf dalam kalimat. Jika huruf ditemukan, fungsi akan mengembalikan indeks huruf. Jika huruf tidak ditemukan, fungsi akan mengembalikan -1.

Pada fungsi main, program meminta pengguna untuk memasukkan kalimat dan huruf yang ingin dicari. Kemudian, program menggunakan fungsi `binarySearch` untuk mencari huruf pada kalimat. Hasil pencarian akan ditampilkan melalui konsol.

dalam screenshot output diatas saya menuliskan kata bakso. Pada output pertama saya ingin mencari huruf "k" maka program mengeluarkan pesan "huruf ditemukan pada undeks 2. Dan run yang kedua saya menuliskan kata bakso dan mencari huruf y maka program mengeluarkan pesan "huruf tidak ditemukan di dalam kalimat.

2. buatlah sebuah program yang dapat mengitung banyaknya huruf vocal dalam sebuah kalimat!

### Source code

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

```

int countVowels(string str) {
    int count = 0;
    for (char c : str) {
        if (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u') {
            count++;
        }
    }
    return count;
}

int main() {
    string kalimat;
    cout << "Masukkan kalimat: ";
    getline(cin, kalimat);
    int vowels = countVowels(kalimat);
    cout << "Jumlah huruf vokal dalam kalimat: " << vowels << endl;
    return 0;
}

```

#### Screenshot output

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
TERMINAL
PS C:\Users\User\Documents\struktur data\modul 8> c
d "c:\Users\User\Documents\struktur data\modul 8\"
; if ($?) { g++ unguided2.cpp -o unguided2 }; if (
$?) { .\unguided2 }
Masukkan kalimat: aku mau ke purbalingga
Jumlah huruf vokal dalam kalimat: 9
PS C:\Users\User\Documents\struktur data\modul 8> [

File Edit View
Nama : Faqih Abdullah
Nim : 2311102048
Kelas : IF-11-B

```

#### Deskripsi program

Program menggunakan fungsi `countVowels` untuk menghitung banyaknya huruf vokal dalam sebuah kalimat. Fungsi ini menggunakan perulangan `for` untuk memeriksa setiap karakter dalam kalimat. Jika karakter tersebut adalah huruf vokal (antara lain 'a', 'e', 'i', 'o', atau 'u'), maka variabel `count` akan ditambahkan.

Pada fungsi `main`, program meminta pengguna untuk memasukkan kalimat. Kemudian, program menggunakan fungsi `countVowels` untuk menghitung jumlah huruf vokal dalam kalimat. Hasilnya akan ditampilkan melalui konsol.

Deskripsi output : saya menginputkan kalimat “ aku mau ke purbalingga”. Didalam kalimat tersebut terdapat 9 huruf vocal maka kalimat mengeluarkan pesan jumlah huruf vocal dalam kalimat : 9.

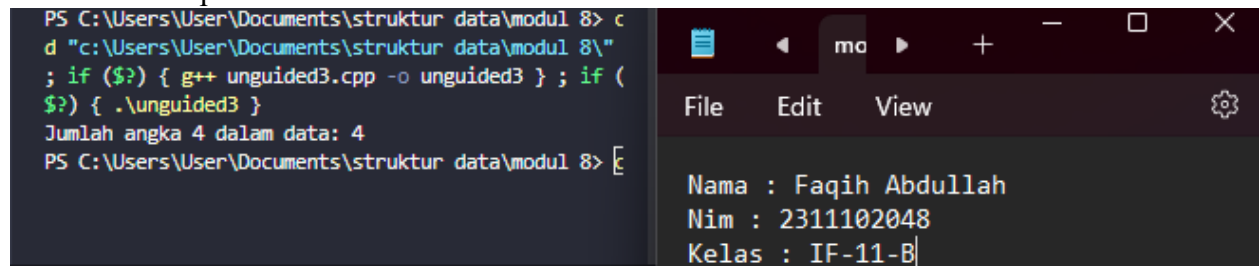
3. Diketahui data = 9, 4, 1, 4, 7, 10, 5, 4, 12, 4. Hitunglah berapa banyak angka 4 dengan menggunakan algoritma Sequential Search!

Source code

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

int main()
{
    int data[] = {9, 4, 1, 4, 7, 10, 5, 4, 12, 4};
    int size = sizeof(data) / sizeof(data[0]); //Mendapatkan ukuran array
    int target = 4;
    int count = 0; // Variabel untuk menghitung jumlah kemunculan angka 4
    for (int i = 0; i < size; i++)
    {
        if (data[i] == target)
        {
            count++;
        }
    }
    std::cout << "Jumlah angka 4 dalam data: " << count
               << std::endl;
    return 0;
}
```

Screenshot output



```
PS C:\Users\User\Documents\struktur data\modul 8> cd "c:\Users\User\Documents\struktur data\modul 8\"
; if ($?) { g++ unguided3.cpp -o unguided3 } ; if ($?) { .\unguided3 }
Jumlah angka 4 dalam data: 4
PS C:\Users\User\Documents\struktur data\modul 8>

File Edit View
Nama : Faqih Abdullah
Nim : 2311102048
Kelas : IF-11-B
```

Deskripsi program

Program diatas menggunakan array data yang berisi beberapa angka, dan mencari jumlah kemunculan angka 4 dalam array tersebut. Pada program ini, terdapat deklarasi array data yang berisi angka-angka 9, 4, 1, 4, 7, 10, 5, 4, 12, dan 4. Kemudian, program menghitung ukuran array menggunakan operasi `sizeof(data) / sizeof(data[0])` dan

*menyimpannya dalam variabel size.*

*Selanjutnya, program mencari jumlah kemunculan angka 4 dalam array dengan menggunakan perulangan for. Perulangan ini akan melakukan iterasi pada setiap elemen dalam array. Jika elemen saat ini sama dengan angka 4, maka variabel count akan ditambahkan.*

*Setelah perulangan selesai, program akan mencetak jumlah kemunculan angka 4.*

## DAFTAR PUSTAKA

sutiono s.kom., m. m. (2024, Januari 3). *DosenIT.com*. Diambil kembali dari Binary Search: pengertian, cara kerja dan kelebihan: [https://dosenit.com/ilmu-komputer/binary-search#Cara\\_Kerja\\_Binary\\_Search](https://dosenit.com/ilmu-komputer/binary-search#Cara_Kerja_Binary_Search)