kelompok II

Teknik Pencarian Data

02

Pencarian Data

Proses pencarian adalah menemukan nilai (data) tertentu di dalam sekumpulan data yang bertipe sama (baik bertipe dasar atau bertipe bentukan).

02

Teknik Pencarian Data

Sequential Search

Merupakan suatu teknik pencarian data dalam array (1 dimensi), dimana data-data tidak perlu diurutkan terlebih dahulu.

Binary Search

Merupakan suatu teknik pencarian data yang hanya dapat dilakukan pada kumpulan data yang sudah diurutkan terlebih dahulu.

Interpolation Search

Merupakan teknik pencarian data yang menyerupai teknik binary search, akan tetapi berbeda pada pembagian datanya.

Sequential Search

- Pencarian sekuensial (sequential searching) atau pencarian berurutan sering disebut pencarian linier merupakan metode pencarian yang paling sederhana. Pencarian beruntun adalah proses membandingkan seriap elemen larik satu per satu secara beruntun, mulai dari elemen pertama sampai elemen yang dicari ditemukan atau seluruh elemen sudah diperiksa.

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

int main() {
    int angka[] = {10, 3, 6, 2, 1, 7, 8};
    int lenght = sizeof(angka)/sizeof(*angka);
    int dtCari, hasilCari;
    cout<<"Data Tersedia : "<<endl;
    for(int a=0; a<lenght; a++) {
        cout<<"Index ke-"<<a<<" : "<<angka[a]<<endl;
    }
    cout<<"Data yang Dicari : ";
    cin>>dtCari;
    for(int b=0; b<lenght; b++) {
        if(dtCari==angka[b]) {
            hasilCari++;
        }
    }
    if(hasilCari==0) {
        cout<<"Data tidak ditemukan"<<endl;
    }
    else {
        cout<<"Data "<<dtCari<<" ditemukan di : "<<endl;
        for(int c=0; c<lenght; c++) {
            if(dtCari==angka[c]) {
                cout<<"Index ke-"<<c<<endl;
            }
        }
    }
}
```

05

```
1) Read n
2) i<—0, Ketemu <—0
3) For i=0; i<n; i++
        Input Data[i]
4) Endfor
5) Read x
6) For i=0;  i<n;  i++
        If Data[i] = x then
             Ketemu<—1
             If Ketemu = 1 then
                      Write "Data Ditemukan!"
                          Endif
                  Endif
 7) Endfor
 8) If Ketemu =0 then
           Write "Data Tidak Ditemukan !"
 9) Endif
```

Algoritma
Sequential Search

05

BINARY SEARCH & INTERPOLATION SEARCH

06

BINARY SEARCH

Pembagian data dengan rumus :
tengah = (awal + akhir)/2

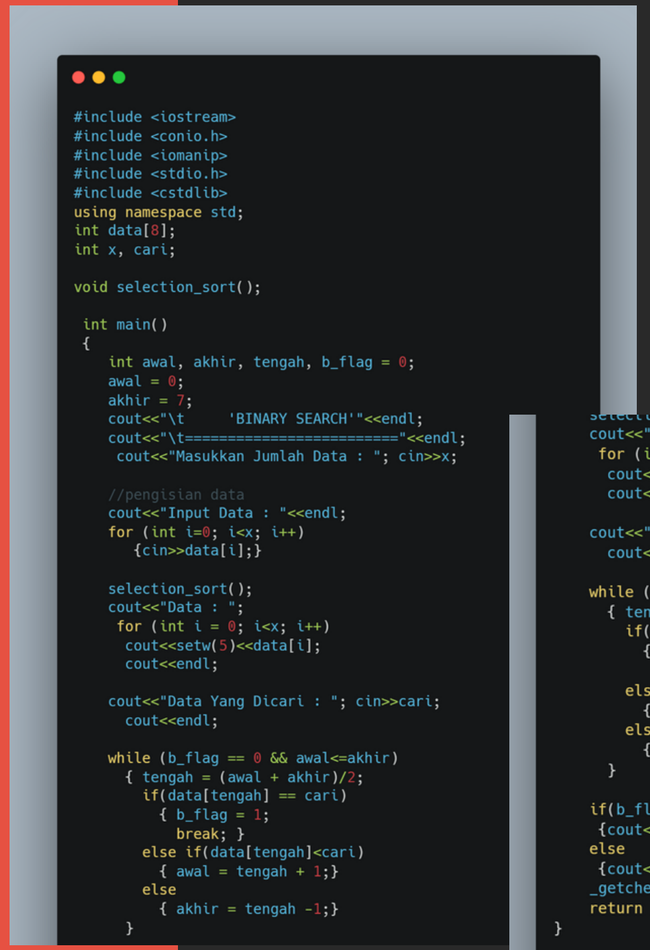
INTERPOLATION SEARCH

Pembagian data denga rumus :
posisi=(kunci-data[awal]/data[akhir]-data[awal])*(akhir-awal)+awal

07

Binary Search

Terdapat metode pencarian pada data terurut yang paling efficient, yaitu metode pencarian bagidua atau pencarian binary (binary search). Metode ini digunakan untuk kebutuhan pencarian dengan waktu yang cepat. Prinsip pencarian dengan membagi data atas dua bagian. Data yang disimpan di dalam larik harus sudah terurut.



```
#include <iostream>
#include <conio.h>
#include <iomanip>
#include <stdio.h>
#include <cstdlib>
using namespace std;
int data[8];
int x, cari;

void selection_sort();

 int main()
 {
    int awal, akhir, tengah, b_flag = 0;
    awal = 0;
    akhir = 7;
    cout<<"\t     'BINARY SEARCH'"<<endl;
    cout<<"\t======================="<<endl;
     cout<<"Masukkan Jumlah Data : "; cin>>x;

    //pengisian data
    cout<<"Input Data : "<<endl;
    for (int i=0; i<x; i++)
       {cin>>data[i];}

    selection_sort();
    cout<<"Data : ";
     for (int i = 0; i<x; i++)
      cout<<setw(5)<<data[i];
      cout<<endl;

    cout<<"Data Yang Dicari : "; cin>>cari;
      cout<<endl;

    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;}
      }
```

```
    cout<<"Data : ";
     for (int i = 0; i<x; i++)
      cout<<setw(5)<<data[i];
      cout<<endl;

    cout<<"Data Yang Dicari : "; cin>>cari;
      cout<<endl;

    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<<"\nData ditemukan pada index ke-"<<tengah<<endl;}
    else
     {cout<<"\nData tidak ditemukan\n";}
    _getche();
    return EXIT_SUCCESS;
}

void selection_sort()
{
    int temp, min, i, j;
    for(i=0; i<x; i++)
      { min = i;
        for(j= i+1; j<x; j++)
          {if(data[j]<data[min])
            {min=j;}
          }

        temp = data[i];
        data[i] = data[min];
        data[min] = temp;
       }
}
```

07

08

```
1) Read n
2) i<—0, Ketemu<—0
3) For i=0; i<n; i++
       Input Data[i]
    {Data Masukan Berikutnya
     Harus Lebih besar}
4) Endfor
5) Read X
6) Awal 1<—0, Akhir<— n-1
7) While (Awal<=Akhir &&
    Ketemu=0)
      Tengah = (Awal+Akhir)/2
      Write Data[Tengah]
      If Data[Tengah] = x then
          Ketemu<—1
   Else
     If (x < Data[Tengah])
       Write "Cari di Kiri"
       Akhir<—Tengah-1
     Else
       Write "Cari di Kanan"
       Awal<—Tengah+1
   Endif
8) Endwhile
9) If (Ketemu=1) then
      Write "Data Ditemukan"
   Else
      Write "Data Tidak Ditemukan"
   Endif
```

Algoritma
Binary Search

08

08

INTERPOLATION Search

- Proses pencarian data ini hampir sama dengan proses pencarian binary search, pencarian ini juga dilakukan pada kumpulan data yang sudah terurut. Akan tetapi pada interpolation search kita membagi data menerut rumus :
- (kunci-data[awal]/data[akhir-data[akhir])*(akhir-awal)-awal
- Singkatnya proses pencarian interpolation search hampir mirip dengan proses pencarian kata dikamus, yaitu kita mencari data yang dimaksud dengan cara memperkirakan letak data.

```
#include <iostream>
#include <conio.h>
#include <iomanip>
using namespace std;
int x;
int data[10];

void selection_sort();

int main()
{
    int cari_data, posisi, awal, akhir, proses;
    bool berhenti = false;

    cout<<"\t 'INTERPOLATION SEARCH'"<<endl;
    cout<<"\t======================="<<endl;
    cout<<"Masukkan Jumlah Data : "; cin>>x;

    //pengisian data
    cout<<"Input Data : "<<endl;
    for (int i=0; i<x; i++)
       {cin>>data[i];}

    selection_sort();
    cout<<"Data : ";
     for (int i = 0; i<x; i++)
      cout<<setw(5)<<data[i];
      cout<<endl;

    cout<<"Data Yang Dicari : "; cin>>cari_data;

    awal=0;
    akhir=x-1;
    proses=0;

    while(berhenti!=true)
        { proses++;
          posisi = (((cari_data-data[awal])*(akhir-awal))/(data[akhir]-data[awal])+awal);

          if(data[posisi]==cari_data)
            { cout<<"Data "<<cari_data<<" Pada Posisi Indeks Ke-"<<posisi<<endl;
              cout<<"Proses Pencarian Sebanyak "<<proses<<endl;
              berhenti=true;
            }
          else if(data[posisi]<cari_data)
            { awal=posisi+1;}
          else {
            cout<<"Data "<<cari_data<<" Tidak Ditemukan.\n";
            berhenti=true;
              }
          }

    return 0;
}

void selection_sort()
{
    int temp, min, i, j;
    for(i=0; i<x; i++)
      { min = i;
```

```
int main()
{
    int cari_data, posisi, awal, akhir, proses;
    bool berhenti = false;

    cout<<"\t 'INTERPOLATION SEARCH'"<<endl;
    cout<<"\t======================="<<endl;
    cout<<"Masukkan Jumlah Data : "; cin>>x;

    //pengisian data
    cout<<"Input Data : "<<endl;
    for (int i=0; i<x; i++)
       {cin>>data[i];}

    selection_sort();
    cout<<"Data : ";
     for (int i = 0; i<x; i++)
      cout<<setw(5)<<data[i];
      cout<<endl;

    cout<<"Data Yang Dicari : "; cin>>cari_data;

    awal=0;
    akhir=x-1;
    proses=0;

    while(berhenti!=true)
        { proses++;
          posisi = (((cari_data-data[awal])*(akhir-awal))/(data[akhir]-data[awal])+awal);

          if(data[posisi]==cari_data)
            { cout<<"Data "<<cari_data<<" Pada Posisi Indeks Ke-"<<posisi<<endl;
              cout<<"Proses Pencarian Sebanyak "<<proses<<endl;
              berhenti=true;
            }
          else if(data[posisi]<cari_data)
            { awal=posisi+1;}
          else {
            cout<<"Data "<<cari_data<<" Tidak Ditemukan.\n";
            berhenti=true;
              }
          }

    return 0;
}

void selection_sort()
{
    int temp, min, i, j;
    for(i=0; i<x; i++)
      { min = i;
        for(j= i+1; j<x; j++)
          {if(data[j]<data[min])
            {min=j;}
          }

        temp = data[i];
        data[i] = data[min];
        data[min] = temp;
      }
}
```

10

Algoritma Interpolation Search

```
1) Read n
2) i<—0, Ketemu<—0
3) For i=0; i<n; i++
       Input Data[i]
    {Data Masukan Berikutnya
     Harus Lebih besar}
4) Endfor
5) Read X
6) Awal 1<—0, Akhir<— n-1
7) While (Awal<=Akhir &&
    Ketemu=0)
      posisi=(kunci-data[awal]
         /data[akhir]-data[awal])
         *(akhir-awal)+awal
      Write Data[posis]
      If Data[posisi] = x then
          Ketemu<—1
      Else
        If (x < Data[psisi])
          Write "Cari di Kiri"
          Akhir<—posisi-1
        Else
          Write "Cari di Kanan"
          Awal<—posisi+1
      Endif
8) Endwhile
9) If (Ketemu=1) then
       Write "Data Ditemukan"
    Else
       Write "Data Tidak Ditemukan"
    Endif
```

Members of the group

- RANI FADHILAH
  222330
- MUH DAFFA HIDAYATULLA
  222343
- GLORYA S.P LINGAN
  222352
- TOUFIQ ARHAM
  222351
- ANUGRAH LUDEN
  222339
- RAHMAT RAMADHAN
  222337
- ANDI AIDIL FATRA.R
  222331

THANK YOU

Sekian Presentasi Dari Kelompok II