Algoritma dan Pemrograman Lanjut

Pertemuan Ke-6
Pencarian *(Searching)* 1

Disusun Oleh :
Wilis Kaswidjanti, S.Si.,M.Kom.

Jurusan Teknik Informatika
Fakultas Teknologi Industri
Universitas Pembangunan Nasional "Veteran"
Yogyakarta

Algoritma dan Pemrograman Lanjut

Judul Materi : Pencarian (*Searching)* 1

Deskripsi Materi : Materi ini membahas metode-metode searching sequensial search menggunakan tipe data array untuk data urut maupun tidak urut

Tujuan Instruksional Khusus :

1. Memahami dan membandingkan metode searching menggunakan tipe data array
2. Mendeskripsikan berbagai metode searching
3. Mengilustrasikan berbagai metode searching menggunakan tipe data array

Referensi :

- Buku Teks

  Munir, Rinaldi (2005), *Algoritma dan Pemrograman dalam Bahasa Pascal dan C*, Buku 2, Edisi Ketiga, Penerbit Informatika Bandung, Bab 1, hal 1-34.

  Charibaldi, N. (2004), *Modul Kuliah Algoritma Pemrograman II*, Edisi Kedua, Yogyakarta

- Buku Acuan/Referensi

  Brassard, Gilles (1999), *Fundamentals of algorithma*, PrinteceHall.

  Jarne, Stroustrup B. (1997), *C++ Programming language*, AT &T.

  Kristanto, Andri (2003), *Algoritma pemrograman C++*, Graha Ilmu.

  Schildt,Herbert (2000), *The Complete Reference C++*, McGraw-Hill.

  Sedgewick, R. (2000), *Algoritma Third edition In C part 5*, Addison Wesley.

PENCARIAN (*SEARCHING*) 1

PENDAHULUAN

Pencarian di perlukan untuk mencari informasi khusus dari tabel pada saat lokasi yang pasti dari informasi tersebut sebelumnya tidak diketahui. Pencarian selalu dinyatakan dengan referensi pada adanya sekolompok data yang tersimpan secara terorganisasi, kelompok data tersebut kita sebut tabel.

Array memungkinkan untuk menyimpan nilai yang bertipe sama. Operasi yang umum dalam array adalah Sequential Search dan Binary search. Perbedaan dari kedua teknik ini terletak pada keadaan data.

ISI

Pencarian Sequential (Sequential Search)

Pencarian Sequential digunakan apabila data dalam keadaan acak atau tidak terurut. Pencarian Sequential atau sering disebut Pencarian Linear menggunakan prinsip sebagai berikut data yang ada dibandingkan satu persatu secara berurutan dengan data yang dicari.

Pencarian Sequential (Sequential Search) :

1. Sequential Search pada Array yang elemen datanya Belum Terurut
    a. Metode tanpa Sentinel
    b. Metode dengan sentinel
2. Sequential Search pada Array yang elemen datanya Sudah Terurut
    c. Metode Sequential Tanpa Sentinel
    d. Metode Sequential Search Dengan Sentinel

A. Proses pencarian sequential data belum terurut tanpa sentinel :

- pada dasarnya pencarian ini hanya melakukan pengulangan dari elemen ke-1 sampai dengan jumlah data.
- pada setiap pengulangan, dibandingkan data ke-i dengan yang dicari,
- apabila sama berarti data telah ditemukan,

- sebaliknya apabila sampai akhir pengulangan tidak ada yang sama, berarti data tidak ada.

Contoh Program *SeqSearch_BelumUrut_nonSentinel*

```
/* SeqSearch_BelumUrut_nonSentinel
   diasumsikan Array X sudah ada dan berisi data yang belum
   terurut, nilai yang dicari adalah y dan hanya ada satu */

#include <iostream.h>
typedef enum boolean {false=0,true=1};
main()
{
  int X[10]={20,50,10,30,90,60,70,80,40,100};
  boolean found;
  int i,y;
  cout << "nilai yang dicari = ";
  cin >> y;
  found=false;
  i=0;
  while ((i<10) & (!found))
  {
    if (X[i]==y)
         found = true;
    else
         i = i + 1;
  }
  if(found)
     cout << y <<" ditemukan pada index array ke-" <<i;
  else
     cout << y <<" tidak ada dalam Array tersebut";
}
```

Cara lain untuk Sequential Search pada data Array X[n], adalah dengan menyediakan satu tempat setelah elemen terakhir, yaitu pada X[n+1] dan menyimpan harga yang dicari (misal y) pada posisi tersebut. Nilai yang dicari pada posisi elemen terakhir tersebut dinamakan sentinel.

B. Proses pencarian sequential data belum terurut dengan sentinel :

- pada dasarnya pencarian ini sama dengan proses pencarian sequential data belum terurut tanpa sentinel yaitu melakukan pengulangan dari elemen ke-1 sampai dengan jumlah data.
- pada setiap pengulangan, dibandingkan data ke-i dengan yang dicari,
- apabila sama berarti data telah ditemukan,

- perbedaannya dengan yang tanpa sentinel adalah ketika data ditemukan tapi data tersebut adalah sentinel berarti data tidak ada.

Contoh Program *SeqSearch_BelumUrut_Sentinel {cara1}*

```
/* SeqSearch_BelumUrut_Sentinel{cara1}
  diasumsikan Array X[0..10] sudah ada dan indeks ke 0..9 telah
  berisi data yang belum urut, nilai yang dicari adalah y dan
  hanya ada satu, y diletakkan di index ke-10 */

#include <iostream.h>
main()
{
 int X[11]={20,50,10,30,90,60,70,80,40,100};
 int i,y;
 cout << "nilai yang dicari = ";
 cin >> y;
 X[10]=y;
 i=0;
 while (X[i]!=y)
       i=i+1;
 if (i>9)
       cout << "tidak ada " << y << " dalam Array";
 else
       cout << y << " ditemukan dalam Array pada index ke-" << i;
}
```

Contoh Program *SeqSearch_BelumUrut_Sentinel {cara2}*

```
/* SeqSearch_BelumUrut_Sentinel {cara2}
diasumsikan Array X[0..10] sudah ada dan indeks ke 0..9 telah
berisi data yang belum terurut,nilai yang dicari adalah y dan
hanya ada satu, y diletakkan di index ke-10 */

#include <iostream.h>
typedef enum boolean {false=0,true=1};
main()
{
 int X[11]={20,50,10,30,90,60,70,80,40,100};
 int i,y;
 boolean found;
 cout << "nilai yang dicari = ";
 cin >> y;
 X[10]=y;
found=false;
 i=0;
 while (!found)
 {  if (X[i]==y)  found=true;
    else  i=i+1;
 }
 if (i==10)
```

```
      cout << "tidak ada " << y << " dalam Array";
 else
      cout << y << " ditemukan dalam Array pada index ke-" << i;
}
```

C. Proses pencarian sequential data sudah terurut tanpa sentinel :

Dimulai dari elemen pertama pada Array, dilakukan pembandingan dengan elemen yang dicari. Jika elemen dalam Array masih lebih kecil dari elemen yang dicari maka pencarian diteruskan. Jika sudah lebih besar, pencarian dihentikan, dan bisa dipastikan bahwa elemen yang dicari memang tidak ada.

Contoh Program *SeqSearch_Urut_NonSentinel*

```
/* SeqSearch_SudahUrut_NonSentinel
diasumsikan Array X sudah ada dan berisi data yang sudah terurut,
nilai yang dicri adalah y dan hanya ada satu */

#include <iostream.h>
typedef enum boolean {false=0,true=1};
main()
{
 int X[10]={10,20,30,40,50,60,70,80,90,100};
 int i,y;
 boolean found;
 cout << "nilai yang dicari = ";
 cin >> y;
 found=false;
 i=0;
 while ((i<9) & (!found) & (y>=X[i]))
 {    if (X[i]==y)
           found=true;
      else
           i=i+1;
 }

if (found)
      cout << y << " ditemukan dalam Array pada index ke-" << i;
 else
      cout << "tidak ada " << y << " dalam Array";
}
```

D. Proses pencarian sequential data sudah terurut dengan sentinel :

Jika digunakan cara pencarian dengan sentinel (elemen yang dicari disisipkan di index setelah data terakhir), dan elemen yang dicari lebih besar dari data terakhir yang

ada di Array sehingga data yang dicari sama dengan data sentinel maka dapat disimpulkan bahwa data tidak ditemukan.

Contoh Program *SeqSearch_Urut_Sentinel {cara1}*

```
/* SeqSearch_SudahUrut_Sentinel {cara 1}
diasumsikan Array X[1..nmax] sudah ada dan indeks 1..n telah
berisi data yang sudah terurut,nilai yang dicari adalah y dan
hanya ada satu */

#include <iostream.h>
typedef enum boolean {false=0,true=1};
main()
{
 int X[11]={10,20,30,40,50,60,70,80,90,100};
 int i,y;
 boolean found;
 cout << "nilai yang dicari = "; cin >> y;
 X[10]=y;
 found=false;
 i=0;
 while ((!found) & (X[i]<y))
     i=i+1;  //tidak ada pengecekkan ketemu atau tidak
 if (i>9)
    cout << "tidak ada " << y << " dalam Array";
 else
    if (X[i]==y)
         cout << y <<" ditemukan dalam Array pada index ke-"<< i;
    else
         cout << "tidak ada " << y << " dalam Array";
}
```

Contoh Program *SeqSearch_Urut_Sentinel {cara2}*

```
/* SeqSearch_SudahUrut_Sentinel {cara 2}
diasumsikan Array X [0..10] sudah ada dan indeks 1..9 telah
berisi data yang sudah terurut,nilai yang dicri adalah y dan
hanya ada satu */

#include <iostream.h>
typedef enum boolean {false=0,true=1};
main()
{
 int X[11]={10,20,30,40,50,60,70,80,90,100};
 int i,y;
 boolean found;
 cout << "nilai yang dicari = "; cin >> y;
 X[10]=y;
 found=false;
 i=0;
 while ((!found) & (X[i]<=y))
```

```
 {    if (X[i]==y)
        found=true;
      else
        i=i+1;
 }
 if (i==10)
   cout << "tidak ada " << y << " dalam Array";
 else
   if (found)
     cout <<  y << " ditemukan dalam Array pada index ke-" << i;
   else
      cout << "tidak ada " << y << " dalam Array";
}
```

PENUTUP

Sequensial Search mempunyai empat macam metode, dan metode pencarian ini dapat dipakai untuk kelompok data yang urut maupun yang tidak urut.

SOAL-SOAL

Tulislah program untuk mencari data bertipe rekaman menggunakan salah satu metode pencarian. Banyaknya data 10. Tipe rekaman mempunyai empat field, yaitu:

- nomorinduk, bertipe bilangan bulat
- nama, bertipe string
- alamat, bertipe string
- golongan, bertipe char (dapat bernilai ‘A’, …’Z’)

Data dicari melalui nomorinduknya. Ketika data yang dicari ditemukan, tampilkan tiga field lainnya.