BAB VIII
Pencarian(Searching)

Tujuan

1. Menunjukkan beberapa algoritma dalam Pencarian
2. Menunjukkan bahwa pencarian merupakan suatu persoalan yang bisa diselesaikan dengan beberapa algoritma yang berbeda
3. Dapat memilih algoritma yang paling sesuai untuk menyelesaikan suatu permasalahan pemrograman

---

Dalam kehidupan sehari-hari sebenarnya kita sering melakukan pencarian data. Sebagai contoh, jika kita menggunakan Kamus untuk mencari kata-kata dalam Bahasa Inggris yang belum diketahui terjemahannya dalam Bahasa Indonesia. Contoh lain saat kita menggunakan buku telepon untuk mencari nomor telepon teman atau kenalan dan masih banyak contoh yang lain.

Pencarian data sering juga disebut *table look-up* atau *storage and retrieval information* adalah suatu proses untuk mengumpulkan sejumlah informasi di dalam pengingat komputer dan kemudian mencari kembali informasi yang diperlukan secepat mungkin.

Algoritma pencarian (*searching algorithm*) adalah algoritma yang menerima sebuah argumen kunci dan dengan langkah-langkah tertentu akan mencari rekaman dengan kunci tersebut. Setelah proses pencarian dilaksanakan, akan diperoleh salah satu dari dua kemungkinan, yaitu data yang dicari ditemukan (*successful*) atau tidak ditemukan (*unsuccessful*).

Metode pencarian data dapat dilakukan dengan dua cara yaitu **pencarian internal** (*internal searching*) dan **pencarian eksternal** (*external searching*). Pada pencarian internal, semua rekaman yang diketahui berada dalam pengingat komputer sedangakan pada pencarian eksternal, tidak semua rekaman yang diketahui berada dalam pengingat komputer, tetapi ada sejumlah rekaman yang tersimpan dalam penyimpan luar misalnya pita atau cakram magnetis.

Selain itu metode pencarian data juga dapat dikelompokka menjadi **pencarian statis** (*static searching*) dan **pencarian dinamis** (*dynamic searching*). Pada pencarian statis, banyaknya rekaman yang diketahui dianggap tetap, pada pencarian dinamis, banyaknya rekaman yang diketahui bisa berubah-ubah yang disebabkan oleh penambahan atau penghapusan suatu rekaman.

Ada dua macam teknik pencarian yaitu pencarian sekuensial dan pencarian biner. Perbedaan dari dua teknik ini terletak pada keadaan data. Pencarian sekuensial digunakan apabila data dalam keadaan acak atau tidak terurut. Sebaliknya, pencarian biner digunakan pada data yang sudah dalam keadaan urut.

8.1 Pencarian Berurutan (*Sequential Searching*)

Pencarian berurutan sering disebut pencarian linear merupakan metode pencarian yang paling sederhana. Pencarian berurutan menggunakan prinsip sebagai berikut : data yang ada dibandingkan satu per satu secara berurutan dengan yang dicari sampai data tersebut ditemukan atau tidak ditemukan.

Pada dasarnya, pencarian ini hanya melakukan pengulangan dari 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 data yang sama, berarti data tidak ada. Pada kasus yang paling buruk, untuk N elemen data harus dilakukan pencarian sebanyak N kali pula.

Algoritma pencarian berurutan dapat dituliskan sebagai berikut :

1 i ← 0
2 ketemu ← false
3 Selama (tidak ketemu) dan (i <= N) kerjakan baris 4
4 Jika (Data[i] = x) maka ketemu ← true, jika tidak i ← i + 1
5 Jika (ketemu) maka i adalah indeks dari data yang dicari, jika tidak data tidak ditemukan

Di bawah ini merupakan fungsi untuk mencari data menggunakan pencarian sekuensial.

```
int SequentialSearch(int x)
{
   int i = 0;
   bool ketemu = false;

   while ((!ketemu) && (i < Max)){
```

```
      if(Data[i] == x)
         ketemu = true;
      else
         i++;
   }
   if(ketemu)
      return i;
   else
      return -1;
}
```

Program 8.1 Fungsi untuk Mencari Data dengan Metode Sekuensial

Fungsi diatas akan mengembalikan indeks dari data yang dicari. Apabila data tidak ditemukan maka fungsi diatas akan mengembalikan nilai –1.

8.2 Pencarian Biner (Binary Search)

Salah satu syarat agar pencarian biner dapat dilakukan adalah data sudah dalam keadaan urut. Dengan kata lain, apabila data belum dalam keadaan urut, pencarian biner tidak dapat dilakukan. Dalam kehidupan sehari-hari, sebenarnya kita juga sering menggunakan pencarian biner. Misalnya saat ingin mencari suatu kata dalam kamus

Prinsip dari pencarian biner dapat dijelaskan sebagai berikut : mula-mula diambil posisi awal 0 dan posisi akhir = N - 1, kemudian dicari posisi data tengah dengan rumus (posisi awal + posisi akhir) / 2. Kemudian data yang dicari dibandingkan dengan data tengah. Jika lebih kecil, proses dilakukan kembali tetapi posisi akhir dianggap sama dengan posisi tengah –1. Jika lebih besar, porses dilakukan kembali tetapi posisi awal dianggap sama dengan posisi tengah + 1. Demikian seterusnya sampai data tengah sama dengan yang dicari.

Untuk lebih jelasnya perhatikan contoh berikut. Misalnya ingin mencari data 17 pada sekumpulan data berikut :

3	9	11	12	15	17	20	23	31	35
awal				tengah					akhir

Mula-mula dicari data tengah, dengan rumus (0 + 9) / 2 = 4. Berarti data tengah adalah data ke-4, yaitu 15. Data yang dicari, yaitu 17, dibandingkan dengan data tengah

ini. Karena 17 > 15, berarti proses dilanjutkan tetapi kali ini posisi awal dianggap sama dengan posisi tengah + 1 atau 5.

3	9	11	12	15	17	20	23	31	35
					awal		tengah		akhir

Data tengah yang baru didapat dengan rumus (5 + 9) / 2 = 7. Berarti data tengah yang baru adalah data ke-7, yaitu 23. Data yang dicari yaitu 17 dibandingkan dengan data tenah ini. Karena 17 < 23, berarti proses dilanjukkan tetapi kali ini posisi akhir dianggap sama dengan posisi tengah – 1 atau 6.

3	9	11	12	15	17	20	23	31	35
					awal=tengah	akhir			

Data tengah yang baru didapat dengan rumus (5 + 6) / 2 = 5. Berarti data tengah yang baru adalah data ke-5, yaitu 17. data yang dicari dibandingkan dengan data tengah ini dan ternyata sama. Jadi data ditemukan pada indeks ke-5.

Pencarian biner ini akan berakhir jika data ditemukan atau posisi awal lebih besar daripada posisi akhir. Jika posisi sudah lebih besar daripada posisi akhir berarti data tidak ditemukan.

Untuk lebih jelasnya perhatikan contoh pencarian data 16 pada data diatas. Prosesnya hampir sama dengan pencarian data 17. Tetapi setelah posisi awal 5 dan posisi akhir 6, data tidak ditemukan dan 16 < 17, maka posisi akhir menjadi posisi tengah – 1 atau = 4 sedangkan posisi awal = 5.

3	9	11	12	15	17	20	23	31	35
				akhir	awal				

Disini dapat dilihat bahwa posisi awal lebih besar daripada posisi akhir, yang artinya data tidak ditemukan.

Algoritma pencarian biner dapat dituliskan sebagai berikut :

1 L ← 0

2 R ← N - 1

3 ketemu ← false

4 Selama (L <= R) dan (tidak ketemu) kerjakan baris 5 sampai dengan 8

5 m ← (L + R) / 2

6 Jika (Data[m] = x) maka ketemu ← true

7 Jika (x < Data[m]) maka R ← m – 1

8 Jika (x > Data[m]) maka L ← m + 1

9 Jika (ketemu) maka m adalah indeks dari data yang dicari, jika tidak data tidak ditemukan

Di bawah ini merupakan fungsi untuk mencari data menggunakan pencarian biner.

```
int BinarySearch(int x)
{
   int L = 0, R = Max-1, m;
   bool ketemu = false;

   while((L <= R) && (!ketemu))
   {
      m = (L + R) / 2;
      if(Data[m] == x)
         ketemu = true;
      else if (x < data[m])
         R = m - 1;
      else
         L = m + 1;
   }
   if(ketemu)
      return m;
   else
      return -1;
}
```

Program 8.2 Fungsi Pencarian Data dengan Metode Biner

Fungsi diatas akan mengembalikan indeks dari data yang dicari. Apabila data tidak ditemukan maka fungsi diatas akan mengembalikan nilai –1.

Jumlah pembandingan minimum pada pencarian biner adalah 1 kali, yaitu apabila data yang dicari tepat berada di tengah-tengah. Jumlah pembandingan maksimum yang dilakukan dengan pencarian biner dapat dicari menggunakan rumus logaritma, yaitu :

$C = {}^{2}\log(N)$

8.3 Kesimpulan

1. Algoritma pencarian berurutan digunakan untuk mencari data pada sekumpulan data atau rekaman yang masih acak
2. Algoritma pencarian biner digunakan untuk mencari data pada sekumpulan data atau rekaman yang sudah dalam keadaan terurut.

8.4 Latihan

1. Buatlah fungsi algoritma pencarian berurutan dengan menggunakan struktur data single linked list. Sebutkan keuntungan dari penggunaan struktur data single linked list ini dibandingkan dengan menggunakan array
2. Buatlah fungsi untuk menyisipkan (*insert*) data pada algoritma pencarian berurutan apabila data tersebut tidak ditemukan. Data baru diletakkan pada posisi terakhir. Implementasikan fungsi ini dengan menggunakan struktur data array dan single linked list.
3. Buatlah fungsi algoritma pencarian biner dengan menggunakan struktur data single linked list
4. Buatlah fungsi untuk menyisipkan (*insert*) data pada algoritma pencarian biner apabila data tersebut tidak ditemukan. Data baru disisipkan pada posisi yang tepat sehingga kumpulan data tersebut tetap terurut. Implementasikan fungsi ini dengan menggunakan struktur data array dan single linked list.