

Minggu 08 Pertemuan 2

Pencarian pada Array

Algoritma Pemrograman – CII1F4

Fakultas Informatika

2021

Outline

Pencarian Nilai Ekstrim

Pencarian pada sekumpulan data yang tersusun
Acak

Latihan Soal



NILAI EKSTRIM

- a. Algoritma pencarian nilai ekstrim digunakan untuk menemukan nilai maksimum (tertinggi) atau minimum (terendah) pada kumpulan data
- b. Contoh kasus: mencari nilai ujian tertinggi/terendah, mencari penduduk dengan usia tertua/termuda, mencari data kejadian gempa dengan magnitudo terbesar/terkecil
- c. Misalkan terdapat tabel yang berisi bilangan bulat seperti di bawah ini:

1	2	3	4	5	6	7	8	9	10	11	12	13	14
9	21	4	13	29	24	5	48	33	37	4	17	20	40

Maximum

- Pencarian nilai maksimum akan menghasilkan 48 sebagai nilai terbesar pada array atau 8, yaitu indeks tabel yang menyimpan nilai terbesar.

Minimum

- Pencarian nilai minimum akan menghasilkan 4 sebagai nilai terkecil pada array atau 3, yaitu indeks tabel yang menyimpan nilai terkecil.



Pencarian Nilai Ekstrim

- Misalnya didekrasikan tipe array sebagai berikut:

Kamus

```
constant NMAX : integer = 2021  {konstanta}  
type tabInt : array [0..NMAX-1] of integer
```

- Ide pencarian nilai ekstrim

Siapkan sebuah nilai ekstrim pertama sebagai acuan

- Gunakan data pertama

Bandingkan nilai ekstrim tersebut dengan data kedua s.d. data terakhir

- Apabila nilai ekstrim tidak valid, maka update dengan yang valid

Algoritma Pencarian Nilai Minimum

- Maka diperoleh algoritma:

```
function minimum(T: tabInt, N: integer) → integer
{Diterima array T yang berisi N bilangan bulat untuk mengembalikan nilai minimum dari array T}
kamus
    min : integer   {variable untuk nilai ekstrim}
    k : integer
algoritma
    min ← T[0]       {data pertama ada di indeks ke-0}
    k ← 1            {perbandingan dilakukan dari data ke-2 hingga N-1}
    while k < N do
        if min > T[k] then           {cek apabila nilai ekstrim tidak valid}
            min ← T[k]                {update nilai ekstrim dengan yang valid}
        endif
        k ← k + 1
    endwhile
    return min           {setelah semua data dicek, maka nilai ekstrim valid}
endfunction
```

- Bagaimana dengan nilai maksimum?

Pencarian Nilai Ekstrim

- Nilai yang dicari tidak terlalu bermanfaat dibandingkan dengan lokasi dari nilai tersebut berada, yaitu **Indeks**
- **Dengan indeks, kita tahu lokasi nilai di dalam array beserta besarannya**

```
...  
kamus  
  idx_min : integer   {variable untuk indeks dari nilai ekstrim}  
  k : integer  
algoritma  
  idx_min <- 0           {indeks 0 adalah lokasi data pertama}  
  k <- 1                 {perbandingan dilakukan dari data ke-2 hingga N-1}  
  while k < N do  
    if T[idx_min] > T[k] then           {cek apabila nilai ekstrim tidak valid}  
      idx_min <- k                       {update indeks nilai ekstrim dengan yang valid}  
    endif  
    k <- k + 1  
  endwhile  
  return idx_min           {setelah semua data dicek, maka indeks nilai ekstrim valid}  
endfunction
```



Soal 1: Mahasiswa

Sebuah program digunakan untuk melaporkan data wisudawan di Universitas Telkom. Data yang disimpan adalah nama, nim, eprt, semester, dan ipk.

Masukan terdiri dari beberapa baris, yang masing-masing barisnya menyatakan nama, nim, eprt, jumlah semester dan ipk dari seorang wisudawan. Masukan berakhir apabila nim adalah “none”.

Keluaran berupa 3 bilangan yang menyatakan eprt tertinggi, ipk terendah, dan rata-rata semester lulusan.

Buatlah program dengan menggunakan subprogram

1. procedure untuk pengisian array wisudawan
2. function untuk mencari:
 - a. eprt tertinggi
 - b. ipk terendah
 - c. rata-rata semester lulusan
3. Program utamanya
4. Asumsi kapasitas dari arraynya adalah 1000.





PENCARIAN PADA KUMPULAN DATA ACAK

- a. Algoritma pencarian ini digunakan untuk menemukan nilai tertentu pada kumpulan data
- b. Contoh kasus: Kondisi yang dicari misalnya sebuah angka tertentu, seorang mahasiswa dengan NIM tertentu, seorang mahasiswa perempuan dengan nilai Algoritma A, dan sebagainya.
- c. Khas dari algoritma pencarian adalah algoritma akan berhenti ketika kondisi yang dicari sudah ditemukan.
- d. Misalkan terdapat tabel yang berisi bilangan bulat seperti di bawah ini:
 - Algoritma akan berhenti ketika kondisi yang dicari sudah ditemukan.
 - Contoh nilai yang dicari adalah 4,
 - Algoritma akan berhenti pada indeks ke-3, dimana nilai 4 pertama kali ditemukan.

1	2	3	4	5	6	7	8	9	10	11	12	13	14
9	21	4	13	29	24	5	48	33	37	4	17	20	40





PENCARIAN PADA KUMPULAN DATA ACAK

- a. Pada kasus dimana nilai yang dicari tidak ada, maka pencarian akan dilakukan sampai elemen terakhir, kemudian berhenti.
- b. Contoh bilangan yang dicari adalah 27
- c. maka pencarian akan berhenti setelah elemen terakhir di-cek.

1	2	3	4	5	6	7	8	9	10	11	12	13	14
9	21	4	13	29	24	5	48	33	37	4	17	20	40

Karena proses pencarian dilakukan secara sekuensial dari data pertama, kedua, hingga ditemukan atau data terakhir. Maka algoritma ini dikenal dengan **Sequential Search**



Sequential Search

Ide dari proses pencarian

- Cek setiap data apakah sama dengan data yang dicari.
- Apabila sama (data yang dicari ketemu) maka pencarian dihentikan
- Apabila sampai data terakhir data yang dicari tidak ditemukan maka data tidak ketemu

```
function SeqSearch(T: tabInt, N:integer, X:integer) → boolean
{Diterima array T yang berisi N bilangan bulat dan sebuah bilangan bulat X,
mengembalikan TRUE apabila X ditemukan pada elemen array T}
kamus
    ketemu : boolean
    k : integer
algoritma
    ketemu <- false      {ketemu adalah false, karena pencarian belum dilakukan}
    k <- 0
    while not ketemu and k < N do    {selama belum ketemu dan belum sampai data terakhir}
        ketemu <- T[k] == X          {update ketemu apabila X ditemukan}
        k <- k + 1
    endwhile                        {ketemu == true, atau k == N (ketemu == false) }
    return ketemu
endfunction
```

Sequential Search

Dalam pencarian: Lokasi/Indeks hasil pencarian lebih informatif dibandingkan status ketemu atau tidak

```
function SeqSearch(T: tabInt, N: integer, X: integer) → integer
{Diterima array T yang berisi N bilangan bulat dan sebuah bilangan bulat X, mengembalikan
indek dari X apabila ditemukan pada elemen array T, dan -1 apabila tidak ditemukan}
kamus
    ketemu : integer
    k : integer
algoritma
    ketemu <- -1                                {ketemu adalah -1, artinya X belum ditemukan}
    k <- 0
    while ketemu == -1 and k < N do {selama belum ketemu dan belum sampai data terakhir}
        if T[k] == X then                        {cek apabila X ditemukan pada indeks ke-k}
            ketemu <- k                          {simpan nilai k apabila X ditemukan}
        endif
        k <- k + 1
    endwhile                                     {ketemu != -1 (X ditemukan), atau k == N (ketemu == false)}
    return ketemu
endfunction
```



Contoh

1	2	3	4	5	6	7	8	9	10	11	12	13	14
9	21	4	13	29	24	5	48	33	37	4	17	20	40

- Misalnya data diberikan X berikut, X adalah 24
Ketika $k == 6$, maka $ketemu == 6$
dan while di baris 3, menjadi false,
karena $ketemu == -1$ adalah false
Artinya data ditemukan pada indeks ke-6
- Misalnya data diberikan X berikut, X adalah 42
Sampai $k == 15$, nilai $ketemu == -1$
dan while di baris 3, menjadi false,
karena $k \leq 14$ adalah false.
Artinya data tidak ditemukan.

```
1  ketemu <- -1
2  k <- 1      {indeks array dari 1}
3  while ketemu == -1 and k <= N do
4      if T[k] == X then
5          ketemu <- k
6      endif
7      k <- k + 1
8  endwhile
9  return ketemu
```



Soal 2: Bunga

Diasumsikan suatu array dengan kapasitas 100 telah berisi sejumlah N data bunga. Buatlah subprogram berikut:

- a. procedure untuk melakukan rename nama bunga tertentu.
- b. procedure delete data bunga dengan nama tertentu. (Geser elemen array untuk mengisi elemen yang kosong setelah proses delete tersebut)

Catatan:

- Tambahkan function pencarian untuk mempermudah proses rename dan delete.
- Tampilan “Bunga tidak ditemukan” apabila nama bunga yang dicari tidak ada.





TERIMA KASIH

