

ALGORITMA PENCARIAN

MINGGU KE: 9

TUJUAN:

- Mahasiswa dapat memahami masalah pencarian.
- Mahasiswa dapat memahami algoritma pencarian beruntun.
- Mahasiswa dapat memahami algoritma pencarian beruntun Versi 1 dengan hasil pencarian boolean
- Mahasiswa dapat memahami algoritma pencarian beruntun Versi 1 dengan hasil pencarian indeks
- Mahasiswa dapat memahami algoritma pencarian beruntun Versi 2 dengan hasil pencarian boolean
- Mahasiswa dapat memahami algoritma pencarian beruntun Versi 2 dengan hasil pencarian indeks

TEORI PENGANTAR:

Bentuk pencarian berupa:

- a. Pencarian untuk memeriksa keberadaan x.
- b. Hasil pencarian adalah indeks elemen larik.
- c. Hasil pencarian berupa nilai Boolean yang menyatakan status hasil pencarian.

Definisikan tipe larik di bagian deklarasi global.

```
{kamus data global}  
DEKLARASI  
const Nmaks = 100  
type LarikInt = array [1..Nmaks] of integer
```

Algoritma Pencarian Beruntun

Algoritma pencarian beruntun adalah proses membandingkan setiap elemen larik satu per satu secara beruntun, mulai dari elemen pertama sampai elemen yang dicari ditemukan.

Versi 1 (Pembandingan elemen dilakukan sebagai kondisi pengulangan)

Hasil Pencarian: sebuah peubah boolean yang bernilai true bila x ditemukan atau bernilai false bila x tidak ditemukan

```
procedure SeqSearch1(input L:LarikInt, input n:integer, input x: integer, output ketemu: Boolean)  
{Mencari keberadaan nilai x di dalam larik L[1..n].}  
{K.Awal: x dan larik L[1..n] sudah terdefinisi nilainya.}
```

{K.Akhir: ketemu bernilai true jika x ditemukan. Jika x tidak ditemukan, ketemu bernilai false.}

DEKLARASI

i: integer

ALGORITMA:

$i \leftarrow 1$

while (i < n) and (L[i] $\neq$ x) do

$i \leftarrow i + 1$

endwhile

{i = n or L[i] = x}

if L[i] = x then

ketemu $\leftarrow$ true

else

ketemu $\leftarrow$ false

endif

Program utama:

PROGRAM Pencarian

{Program untuk mencari nilai tertentu di dalam larik}

DEKLARASI

const Nmaks = 100

type LarikInt : array[1..Nmaks] of integer

L : LarikInt

x : integer

ketemu : boolean

n : integer

procedure BacaLarik(output L: LarikInt, input n: integer)

procedure SeqSearch1(input L : LarikInt, input n : integer, input x : integer, output ketemu : Boolean)

ALGORITMA :

read(n)

BacaLarik(L, n)

read(x)

SeqSearch1(L, n, x, ketemu)

If ketemu then

write(x, ' ditemukan')

else

write(x, ' tidak ditemukan')

endif

Hasil pencarian: indeks elemen larik yang mengandung nilai x.

Setiap elemen larik L dibandingkan dengan x mulai dari elemen pertama L[1]. Aksi perbandingan dilakukan selama indeks larik i belum melebihi n dan L[i] tidak sama dengan x. Aksi perbandingan dihentikan bila L[i] = x atau i = n. Elemen terakhir, L[n] diperiksa secara khusus. Keluaran yang dihasilkan oleh prosedur pencarian adalah sebuah idx yang berisi indeks larik tempat x ditemukan. Jika x tidak ditemukan, idx diisi dengan nilai -1.

procedure SeqSearch2(input L: larik, input n: integer, input x: integer, output idx: integer)

{Mencari keberadaan nilai x di dalam larik L[1..].}

{K. Awal: x dan elemen-elemen larik L[1..n] sudah terdefinisi.}

{K.Akhir: idx berisi indeks larik L yang berisi nilai x. Jika x tidak ditemukan, maka idx diisi dengan nilai -1.}

DEKLARASI

i: integer

ALGORITMA:

i ← 1

while (i < n) and (L[i] ≠ x) do

 i ← i+1

endwhile

 {i = n or L[i] = x}

 if L[i] = x then

 idx ← i

 else

 idx ← -1

 endif

Misalkan program pemanggil prosedur SeqSearch2 bertujuan untuk menambahkan nilai x ke dalam larik, namun sebelum penambahan itu harus ditentukan apakah x sudah terdapat di dalam larik. Jika x belum terdapat di dalam larik, maka x ditambahkan pada elemen ke n+1.

PROGRAM TambahElemenLarik

{Program untuk menambahkan elemen baru ke dalam larik. Elemen baru dibaca dari piranti masukan, lalu dicari apakah sudah terdapat di dalam larik. Jika belum ada, tambahkan elemen baru setelah elemen terakhir.}

DEKLARASI

const Nmaks = 100

type LarikInt : array[1..Nmaks] of integer

L : LarikInt

n : integer

x : integer

idx : integer

procedure BacaLarik(output L: LarikInt, input n: integer)

procedure TulisLarik(output L: LarikInt, input n: integer)

procedure SeqSearch2(input L: LarikInt, input n: integer, input x: integer, output idx: integer)

ALGORITMA:

 read(n)

 BacaLarik(L,n)

 read(x)

 SeqSearch2(L,n,x,idx)

 if idx ≠ -1 then

 write (x, ' sudah terdapat di dalam larik')

 else

 n ← n+1

 L[n] ← x

 TulisLarik(L,n+1)

 endif

Versi 2 (Pembandingan elemen dilakukan di dalam badan pengulangan)

Aksi pembandingan dilakukan di dalam badan pengulangan, bukan di awal pengulangan seperti pada Versi 1. Maka diperlukan peubah Boolean yang akan berfungsi untuk menyatakan apakah x sudah

ditemukan. Misalkan peubah tersebut adalah ketemu yang pada mulanya diisi dengan nilai false. Bila x ditemukan pada elemen ke-i, maka ketemu diisi dengan nilai true.

Hasil pencarian: sebuah peubah Boolean yang bernilai true bila x ditemukan atau bernilai false bila x tidak ditemukan.

Procedure SeqSearch3(input L: LarikInt, input n: integer, input x: integer, output ketemu: Boolean)

{Mencari keberadaan nilai x di dalam larik L[1..n].}

{K.Awal: nilai x, n, dan elemen larik L[1..n] sudah terdefinisi.}

{K.Akhir: ketemu bernilai true jika x ditemukan. Jika x tidak ditemukan, ketemu bernilai false.}

DEKLARASI

i: $\leftarrow$ 1

ketemu $\leftarrow$ false

while (i $\leq$ n) and (not ketemu) do

if L[i] = x then

ketemu $\leftarrow$ true

else

i $\leftarrow$ i+1

endif

endwhile

Hasil pencarian: indeks elemen larik yang mengandung nilai x

procedure SeqSearch4(input L: LarikInt, input n: integer, input x: integer, output idx: integer)

{Mencari keberadaan nilai x di dalam larik L[1..n].}

{K.Awal: x dan elemen-elemen larik L[1..N] sudah terdefinisi.}

{K.Akhir: idx berisi indeks larik L tempat x berada. Jika x tidak ditemukan, maka idx diisi nilai -1.}

DEKLARASI

i: integer

ketemu: Boolean

ALGORITMA:

i $\leftarrow$ 1

ketemu $\leftarrow$ false

while (i $\leq$ n) and (not ketemu) do

if L[i] = x then

ketemu $\leftarrow$ true

else

i $\leftarrow$ i + 1

endif

endwhile

if ketemu then

idx $\leftarrow$ i

else

idx $\leftarrow$ -1

endif

PRAKTIKUM:

1. Buat program dalam Pascal untuk pencarian elemen larik, gunakan procedure pencarian beruntun Versi 1 dengan hasil pencarian Boolean.

2. Buat program dalam Pascal untuk penambahan elemen larik, gunakan procedure pencarian beruntun Versi 1 dengan hasil pencarian indeks.
3. Buat program dalam Pascal untuk pencarian elemen larik, gunakan procedure pencarian beruntun Versi 2 dengan hasil pencarian Boolean.
4. Buat program dalam Pascal untuk penambahan elemen larik, gunakan procedure pencarian beruntun Versi 2 dengan hasil pencarian indeks.

TUGAS:

Misalkan diberikan daftar nama. Buat program dalam Pascal untuk menambahkan elemen larik bila nama tersebut belum ada.