UNIVERSITAS Dinamika

ALGORITMA PEMROGRAMAN
Pertemuan XI

SEARCHING

Oleh
Achmad Arrosyidi

TUJUAN PEMBELAJARAN

Umum:

- ✓ Mahasiswa dapat membuat algoritma pencarian (*searching*) dalam bentuk *flowchart*.

Khusus:

- ✓ Mahasiswa dapat menerapkan algoritma percarian (*searching*) dalam bentuk *flowchart*.

MATERI KULIAH

1. Pengantar Searching
2. Sequential Search
3. Binary Search
4. Latihan

1. PENGANTAR SEARCHING

1. PENGANTAR SEARCHING (1)

- Didalam sebuah program, adakalanya dibutuhkan sebuah proses pencarian data.
- Pencarian data atau searching, awalnya hanya menggunakan metode sequential.
- Dalam perkembangannya banyak metode yang telah ditemukan dan ditemukan.
- Acuan dari kualitas algoritma searching selain memastikan bahwa jika data terdapat dalam database maka harus ditemukan atau sebaliknya (efektivitas). Namun juga kualitasnya ditunjukkan dari kecepatan algoritma tersebut bekerja dalam menemukan data (efisien waktu).
- Diantara metode searching yang sudah ada Sequential Search, Probability Search. Binary Search Tree, dan Balanced Search Tree.

1. PENGANTAR SEARCHING (2)

- Key adalah sebuah subset dari isi sebuah data yang digunakan untuk perbandingan selama proses pencarian.
- Big-O Notation adalah notasi yang digunakan untuk mengindikasikan kenaikan (*Order of growth*) unjuk kerja dari sebuah algoritma *searching*.

2. SEQUENTIAL SEARCH

2. SEQUENTIAL SEARCH (1)

- Sebuah algoritma pencarian sederhana yang digunakan untuk mencari sebuah item yang khusus didalam list.
- Algoritma ini menjalankan perulangan dalam setiap elemen O(n) *list* sampai dengan terjadi kesamaan antara yang dicari dengan item yang berada dalam list atau sampai dengan pencarian list terakhir.

2. SEQUENTIAL SEARCH (2)

```
1) algorithm SequentialSearch(list, item)
2)     Pre:   list ≠ ∅
3)     Post:  return index of item if found, otherwise −1
4)     index ← 0
5)     while index < list.Count and list[index] ≠ item
6)         index ← index + 1
7)     end while
8)     if index < list.Count and list[index] = item
9)         return index
10)    end if
11)    return −1
12) end SequentialSearch
```

2. SEQUENTIAL SEARCH (3)

- ILUSTRASI SEQUENTIAL SEARCH – DATA DITEMUKAN

Data dalam Daftar (list)

NO INDEKS	DATA DALAM LIST
0	Agus
1	Bagus
2	Cagus
3	Dagus
4	Eagus
5	Fagus
6	Gagus

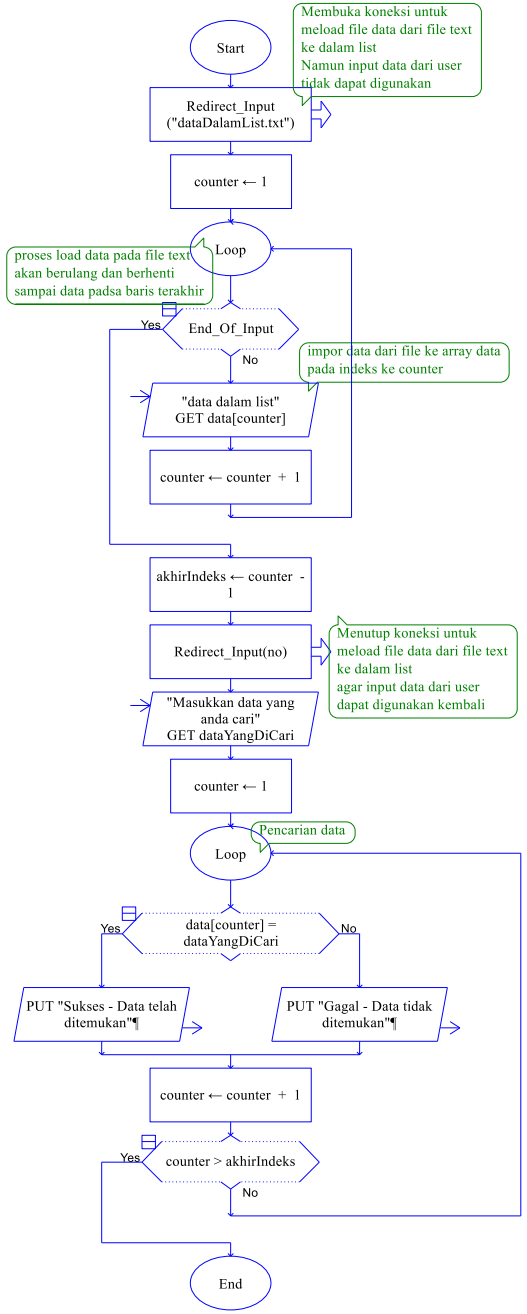
Data yang dicari (Target Key)

Fagus

Bandingkan data yang dicari dengan:
List[0] Agus
List[1] Bagus
List[2] Gagus
List[3] Dagus
List[4] Eagus
List[5] Fagus → Sukses (data ditemukan)

Membutuhkan 6 kali perbandingan

2. SEQUENTIAL SEARCH (4)



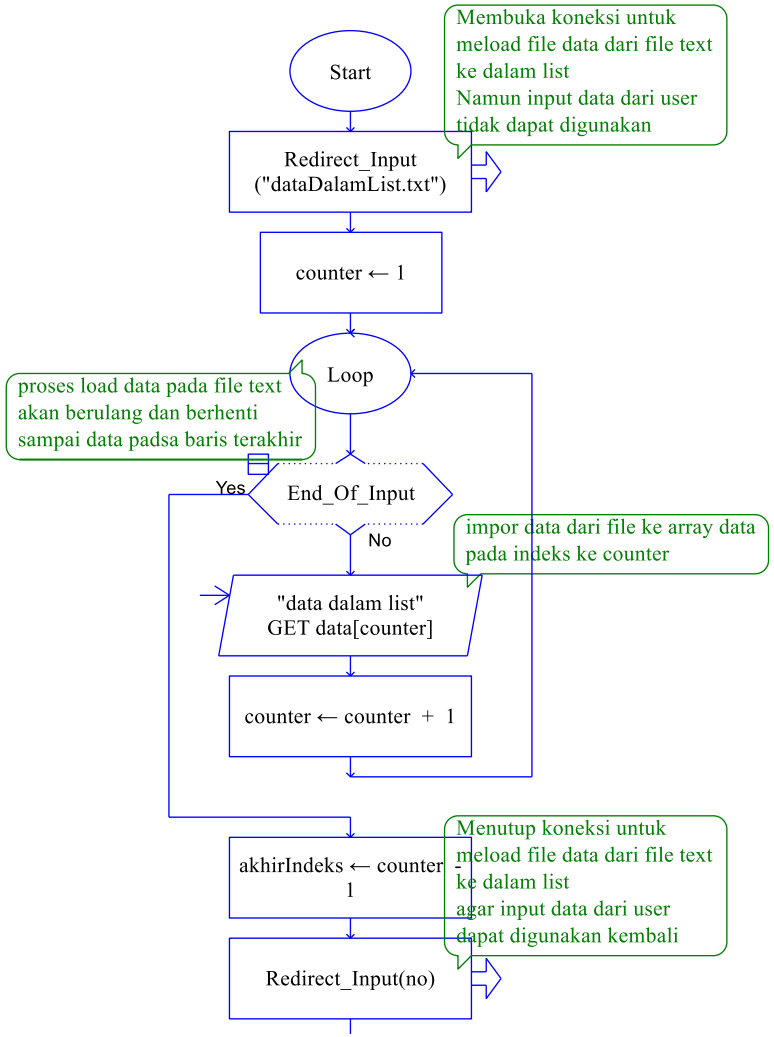
Data yang dicari (Target Key)

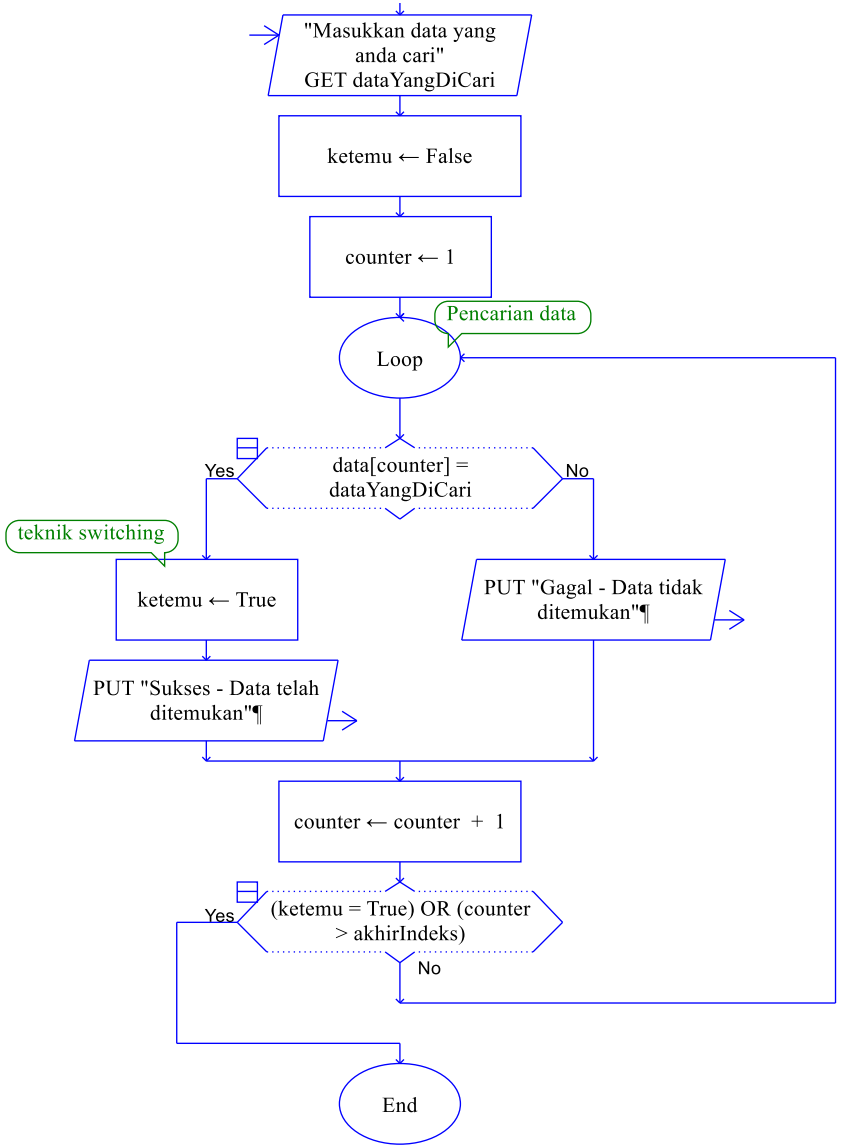
Fagus

Bandingkan data yang dicari dengan:

List[1] Agus → Gagal – Data tidak ditemukan
List[2] Bagus → Gagal – Data tidak ditemukan
List[3] Cagus → Gagal – Data tidak ditemukan
List[4] Dagus → Gagal – Data tidak ditemukan
List[5] Eagus → Gagal – Data tidak ditemukan
List[6] Fagus → Sukses (data ditemukan)
List[7] Gagus → Gagal – Data tidak ditemukan

2. SEQUENTIAL SEARCH (5)





Data yang dicari (Target Key)

Fagus

Bandingkan data yang dicari dengan:

List[1] Agus → Gagal – Data tidak ditemukan
List[2] Bagus → Gagal – Data tidak ditemukan
List[3] Cagus → Gagal – Data tidak ditemukan
List[4] Dagus → Gagal – Data tidak ditemukan
List[5] Eagus → Gagal – Data tidak ditemukan
List[6] Fagus → Sukses (data ditemukan)

Membutuhkan 6 kali perbandingan

2. SEQUENTIAL SEARCH (6)

- ILUSTRASI SEQUENTIAL SEARCH – DATA TIDAK DITEMUKAN

Data dalam Daftar (list)

NO INDEKS	DATA DALAM LIST
0	Agus
1	Bagus
2	Cagus
3	Dagus
4	Eagus
5	Fagus
6	Gagus

Data yang dicari (Target Key)

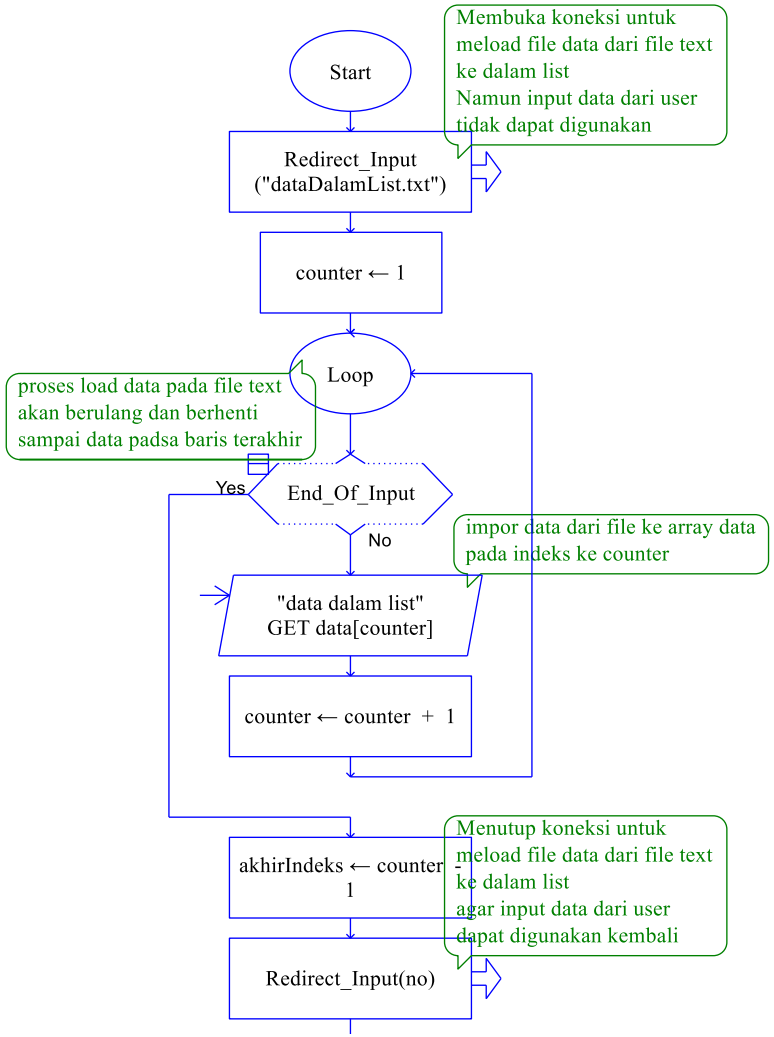
Zagus

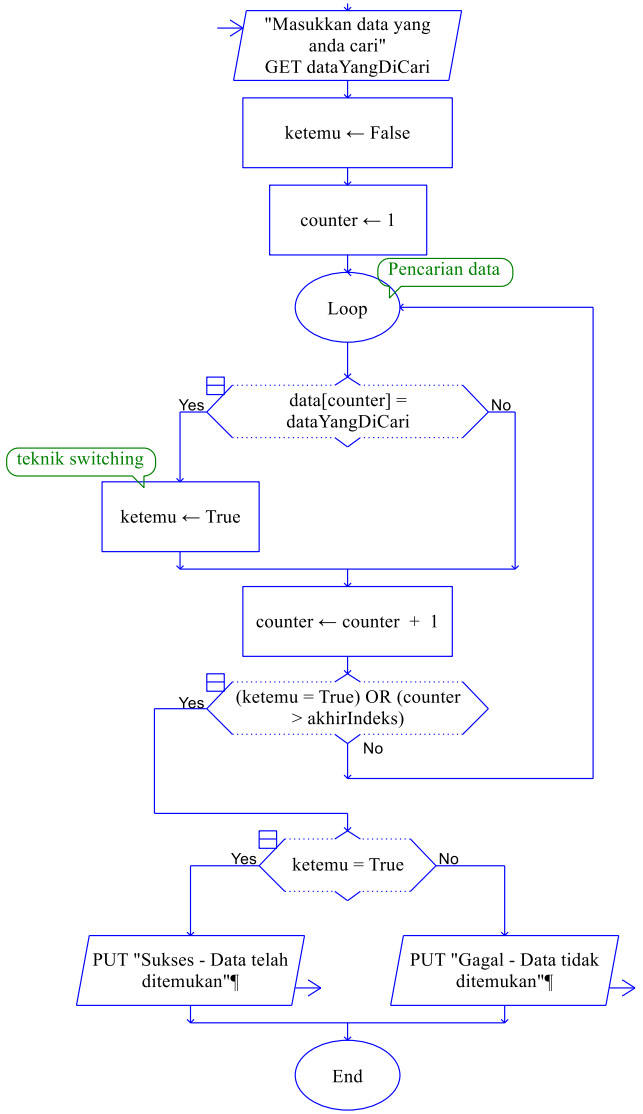
Bandingkan data yang dicari dengan:

List[0] Agus
List[1] Bagus
List[2] Cagus
List[3] Dagus
List[4] Eagus
List[5] Fagus
List[6] Gagus → Gagal (data tidak ditemukan)

Membutuhkan 7 kali perbandingan

2. SEQUENTIAL SEARCH (7)





Data yang dicari (Target Key)

Fagus

Bandingkan data yang dicari dengan:
List[6] Fagus → Sukses (data ditemukan)

Membutuhkan 6 kali perbandingan

Data yang dicari (Target Key)

Zagus

Bandingkan data yang dicari dengan:
List[6] Gagus → Gagal – Data tidak ditemukan

Membutuhkan 7 kali perbandingan

UNIVERSITAS Dinamika

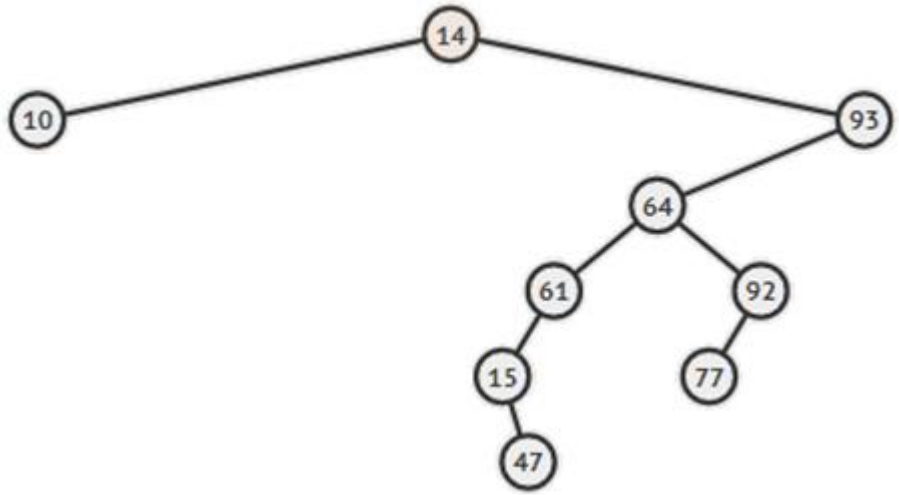
3. BINARY SEARCH TREE

2. BINARY SEARCH TREE (1)

- Sebuah Pohon Biner Terurut (PBT) atau dalam Bahasa Inggris biasa disebut Binary Search Tree (BST)
- BST adalah sebuah pohon biner di mana setiap simpul hanya memiliki tidak lebih dari 2 anak yang memenuhi properti BST.
- Semua simpul-simpul di sub-pohon kiri dari sebuah simpul harus memiliki nilai lebih kecil dibandingkan daripada simpul itu dan semua simpul-simpul di sub-pohon kanan dari sebuah simpul harus memiliki nilai lebih besar daripada simpul itu.
- Sebagai contoh sederhana semua nilai yang dimasukkan pada visualisasi ini adalah bilangan-bilangan bulat yang unik dan perubahan kecil diperlukan untuk mendukung nilai yang tidak unik.

2. BINARY SEARCH TREE (2)

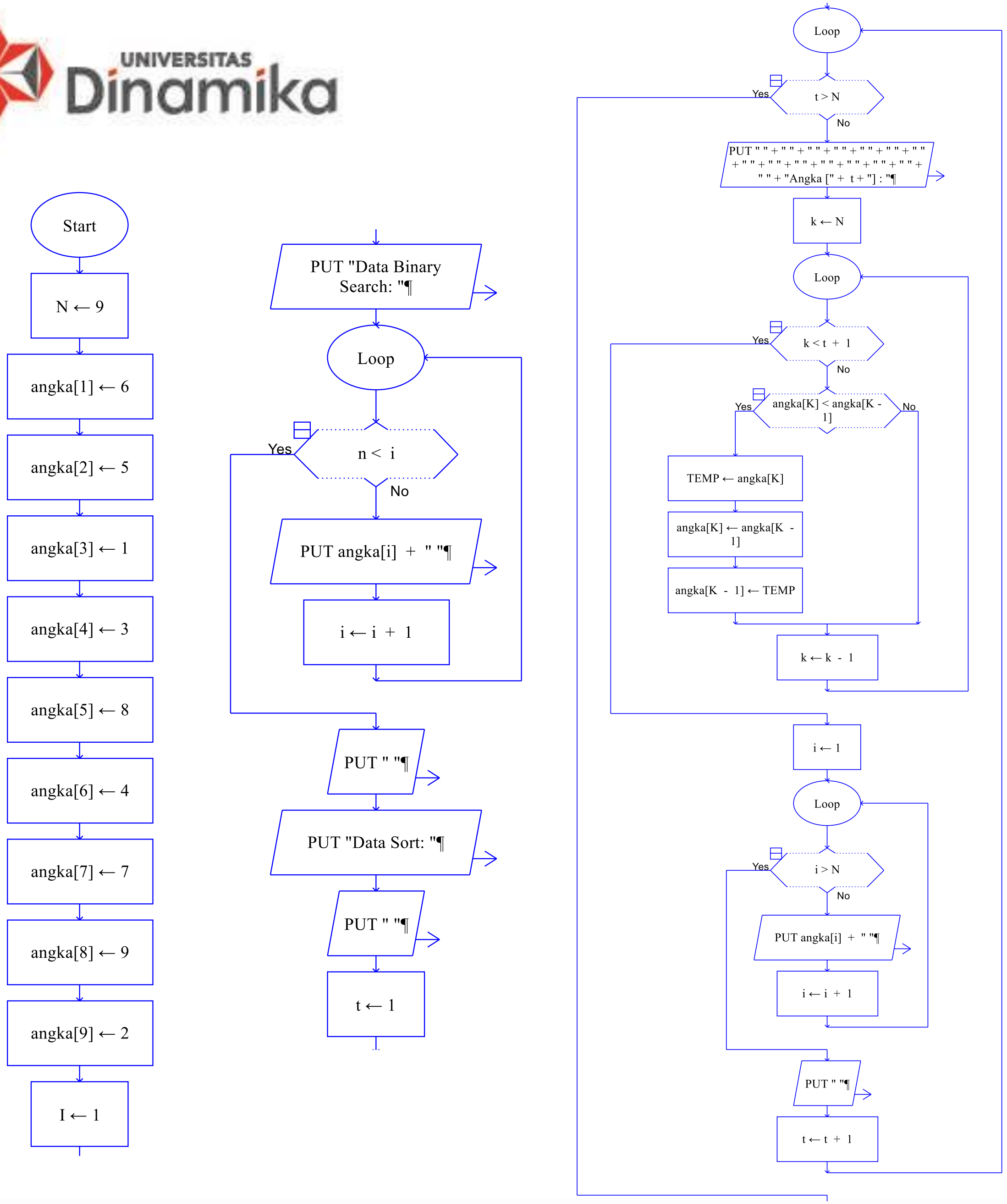
- Sebuah pohon Adelson-Velskii Landis (AVL) adalah BST yang dapat menyeimbangkan diri sendiri di mana tinggi pohon tersebut dapat selalu dibatasi oleh O(log N) dimana N adalah jumlah simpul-simpul didalam pohon AVL tersebut.

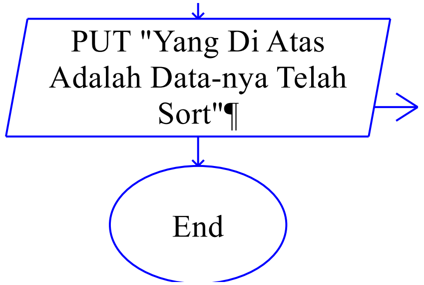


- Untuk simulasi silahkan buka link berikut:
- https://visualgo.net/id/bst?slide=1

UNIVERSITAS Dinamika

2. BINARY SEARCH TREE (3)





- Untuk simulasi silahkan buka link berikut:
- https://visualgo.net/id/bst?slide=1

3. LATIHAN

3. LATIHAN

1. Pilihlah sebuah variabel array 1 Dimensi dalam tugas / UTS untuk digunakan sebagai data pencarian, dan tambahkan proses pencarian menggunakan sequential search, dengan fitur:
   a. Load file dari file text
   b. Teknik switching!
2. Tandai proses pencarian pada nomor 1 dengan komentar pada:
   a. Input data yang dicari!
   b. Lokasi data yang dicari!
3. Pilihlah sebuah variabel array 2 Dimensi dalam tugas / UTS untuk digunakan sebagai data pencarian, dan tambahkan proses pencarian menggunakan sequential search, dengan fitur:
   a. Load file dari file text
   b. Teknik switching!
4. Tandai proses pencarian pada nomor 3 dengan komentar!

REFERENSI

Arrosyidi, A. (2019). Buku Ajar Algoritma Pemrograman dengan Raptor. Surabaya: PT REVKA PETRA MEDIA.

Halim, S., & Halim, F. (2020, 11 29). visualising data structures and algorithms through animation. Retrieved from VisuAlgo.net/en: https://visualgo.net/en

Institut Bisnis dan Informatika Stikom Surabaya. (2010). Logika dan Algoritma. Surabaya: Sekolah Tinggi Manajemen Informatika & Teknik Komputer Surabaya (STIKOM).

UNIVERSITAS Dinamika

SELESAI
TERIMA KASIH