

# **Laporan UAS**

## **Algoritma Struktur Data**



**Disusun oleh :**

**Nama : Mesya Angeliqa Hutagalung**

**NIM : 11321066**

**Prodi : D3 TEKNOLOGI INFORMASI**

**INSTITUT TEKNOLOGI DEL**  
**FAKULTAS VOKASI**

No. \_\_\_\_\_

Date : \_\_\_\_\_

Nama: Mesya Hutagalung

11321066

DIII - Teknologi Informasi

Kamis, 02 Juni 2003

1. Contoh penerapan linked list dalam kehidupan sehari-hari  
 ↳ Contoh penerapan linked list dalam kehidupan sehari-hari yaitu dalam presensi siswa dimana mahasiswa diurutkan berdasarkan suatu data. seperti NIM atau urut abjad dari nama mahasiswa. Selain itu pada online shop kita dapat mengurutkan barang dari harga barang tersebut

2. Alasan mengapa perlu menerapkan linked list yaitu salah satu keuntungannya adalah efektivitas kerja dalam menambah, mengurangi dan mencari sesuatu. Hal tersebut dapat terjadi karena suatu data linked list tidak tersimpan pada blok memori seperti array, melainkan terletak pada blok memori terpisah.

3. Perbedaan antara tree dan graph adalah.

Pembeda	Tree	Graph
Path	hanya satu diantara 2 simpul.	lebih dari satu jalur diizinkan.
Simpul akar	Memiliki tepat 1 simpul akar	tidak memiliki simpul root
loop	tidak ada loop yang diizinkan.	dapat memiliki loop
teknik transversal	Pre-order In-order Post-order	Pencarian luas pertama dan pencarian mendalam pertama.
tipe model	Hierarkis	lingkaran.

3. gambar linked list transversal dari potongan code tersebut adalah.

data ← First (L)

Iterate

Proses (data)

Stop next (data) = NULL

data ← Next (data)

Terminasi

4. function yang menampilkan nilai, hasil yang ditampilkan dari code tersebut.

2, 1, 0, 1, 2

6. Penerapan linked list tree graph pada proyek yang anda bangun.  
tulis minimal 150 kata.

↳ pada proyek yang kami bangun terdiri dari 3 file yaitu header.h, driver.c, dan implementasi.c

header.h untuk menampilkan data yang akan keluar

driver.c <sup>berulang-ulang</sup> untuk data-data disimpan dari struct node di header.h.

Implementasi untuk mencetak data dari header.h.

Pada driver.c kita melakukan include ke header.h, stdio.h, stdlib.h, string.h.

⑦. Root : A.  
 Parent : B, C, E, G.  
 Sibling : (BC), (DE)(FG), (HI)  
 Internal node : AB, E, G, C.  
 Eksternal node : D, H, I, F, J.

⑧. Pre-order : A, B, D, E, H, I, C, F, J, K, G, L  
 In-order : H, E, I, D, B, J, F, K, L, G, C, A  
 Post-order : H, I, E, D, B, J, K, F, L, G, C, A.

⑨.

A	0	1	0	0	0	1
B	1	0	1	1	0	0
C	0	1	0	0	1	0
D	0	1	0	0	1	0
E	0	0	1	1	0	1
F	1	0	0	0	1	0

⑩

A	0	1	0	0	0	1
B	1	0	1	1	0	0
C	0	1	0	0	1	0
D	0	1	0	0	1	0
E	0	0	0	1	0	1
F	1	0	1	0	1	0

⑪. DFS = 1, 2, 5, 4  
 BFS =