# **Daftar isi**

**[Daftar isi](#_Toc452002714)** [i](#_Toc452002714)

[**Kata Pengantar** ii](#_Toc452002715)

[**1.2 Tree** 1](#_Toc452002716)

[**1.3 Graph** 3](#_Toc452002717)

[**1.4 Perbedaan tree dan graph** 6](#_Toc452002718)

[**Daftar Pustaka** 8](#_Toc452002719)

# **Kata Pengantar**

Puji dan syukur penulis panjatkan kehadirat Allah SWT yang telah memberi rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan makalah yang berjudul **"Perbedaan Tree dan Graph"**. Makalah ini disusun bertujuan untuk memenuhi tugas mata kuliah Struktur data.

Penulis menyadari bahwa masih terdapat banyak kekurangan dan keterbatasan dalam penyusunan makalah ini. Oleh karena itu, penulis mengharapkan saran dan kritik dari pembaca, demi memperbaiki makalah ini dalam penulisan lain di kemudian hari.

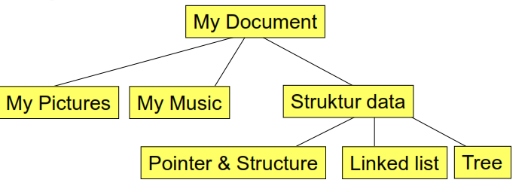
Dan semoga makalah ini dapat mendatangkan manfaat bagi kita semua.

**1.1 Latar Belakang**

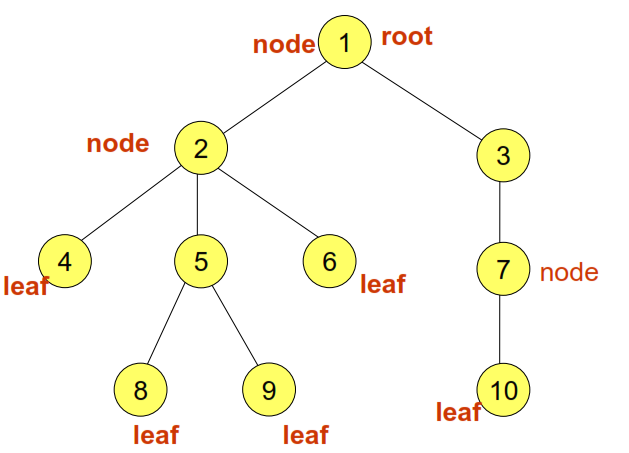
Struktur data tree dan graph sangat diperlukan untuk mengelolah data menjadi data yang terstruktur, sehingga mudah untuk diterapkan di sofware. Struktur data tree dan graph sebagai pendukung untuk komputasi komputer, dan algoritma sebagai logikanya.

# **1.2 Tree**

Penggunaan tree pada komputer contohnya yaitu direktori/folder pada windows atau linux,



Gambar Penerapan metode tree pada windows

Hubungan antar elemen: parentchild, father-son, motherdaughter Nama node: nama(angka) yang dipakai untuk membedakan sebuah node dengan node yanglain. Dalam kuliah ini adalah angka yang tertulis dalam lingkaran.

Sebuah tree didefinisikan sebagai struktur y ang dibentuk secara recursive oleh kedua rule berikut:

1. Sebuah node adalah sebuah tree. Node satu-satunyapada tree ini berfungsi sebagai root maupun leaf.
2. Dari k buah tree T1～T k , dan masing-masing memiliki root n1 ～n k

.Jika node n adalah parent dari noden 1 ～nk,

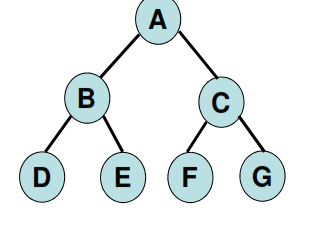
akan diperoleh sebuah tree baru T yang memiliki root n.

Dalam kondisi ini, tree T 1 ～T k menjadi sub-tree dari tree T. Root dari sub-tree n1 ～n k adalah child dari node n .

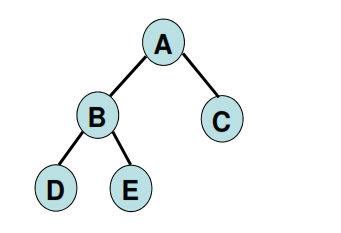
Tree adalah suatu metode dalam struktur data untuk mengurutukan data secara hirarki yang terdapat satu node yang menunjukkan hubungan bertingkat (memiliki hierarkhi). Tree merupakan salah satu bentuk struktur data tidak linear yang menggambarkan hubungan yang bersifat hirarkis (hubungan one to many) antara elemen-elemen. Tree bisa didefinisikan sebagai kumpulan simpul/node dengan satu elemen khusus yang disebut Root dan node lainnya. Tree juga adalah suatu graph yang acyclic, simple, connected yang tidak mengandung loop.

Struktur ini biasanya digunakan untuk menyajikan data yang mengandung hubungan hirarkial antara elemen-elemennya. Bentuk Pohon Berakar yang lebih mudah dikelola dalam komputer adalah Pohon Biner (Binary Tree) yang lebih

dikenal sebagai Pohon Umum (General Tree) yang dapat didefinisikan sebagai kumpulan simpul yang mungkin kosong atau mempunyai akar dan dua Subpohon yang saling terpisah yang disebut dengan Subpohon Kiri / cabang kiri (Left Subtree) dan Subpohon Kanan / cabang kanan (Right Subtree). Karakteristik Pohon Binar (Binary Tree) :

1. Setiap Simpul paling banyak hanya memiliki dua buah anak
2. Derajat Tertinggi dari setiap Simpul adalah dua.
3. Dibedakan antara Cabang Kiri dan Cabang Kanan.
4. Dimungkinkan tidak mempunyai Simpul

Pohon Biner Penuh (Full Binary Tree)Semua simpul (kecuali daun) memiliki 2 anak dan tiap cabang memiliki panjang ruas yang sama

Pohon Biner Lengkap (Complete Binary Tree) Hampir sama dengan Pohon Biner Penuh, semua simpul (kecuali daun) memiliki 2 anak tetapi tiap cabang memiliki panjang ruas berbeda

Tree dapat dibuat dengan menggunakan linked list secara rekursif, Linked list yang digunakan adalah double linked list non circular, data yang pertama kali masuk akan menjadi node root, data yang lebih kecil dari data node root akan masuk dan menempati node kiri dari node root, sedangkan jika lebih besar dari data node root, akan masuk dan menempati node di sebelah kanan node root.

# **1.3 Graph**

Graph •  Graph adalah kumpulan dari simpul dan busur yang secara matematis dinyatakan sebagai :

G = (V, E)

G = Graph.

V = Simpul atau Vertex, atau Node, atau Titik.

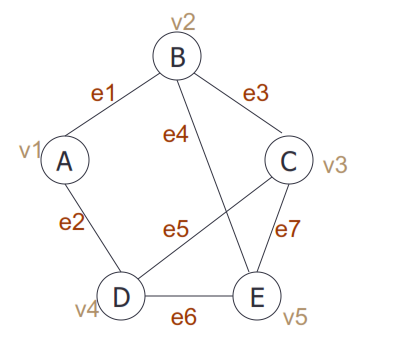
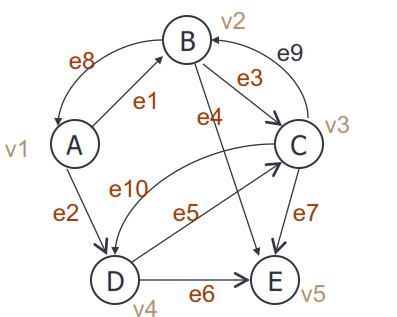
E = Busur atau Edge, atau arc.

Cara pendefinisian lain untuk graph adalah dengan menggunakan himpunan keterhubungan langsung Vx. Pada setiap verteks x terdefinisi Vx sebagai himpunan dari verteks-verteks yang adjacent

dari x. Secara formal:

Vx = {y | (x,y) -> E}

Dalam digraph didefinisikan juga terminologi-terminologi berikut ini. Predesesor dari suatu verteks x (ditulis Pred(x)) adalah himpunan semua verteks yang adjacent ke x. Suksesor dari verteks x (ditulis Succ(x)) adalah himpunan semua verteks yang adjacent dari x. Graph dapat dibedakan dari graph yang berarah dan tidak berarah, gambar graph berarah dan tidak berarah:



Grap tidak berarah Grap berarah

Masih-masing graph mempunyai keuntungan dan kelemhan.

Graph tak berarah (undirected graph atau non-directed graph) :

1. Urutan simpul dalam sebuah busur tidak dipentingkan. Mis busur

e1 dapat disebut busur AB atau BA

Graph berarah (directed graph) :

1. Urutan simpul mempunyai arti. Mis busur AB adalah e1 sedangkan busur BA adalah e8.

Graph Berbobot (Weighted Graph)

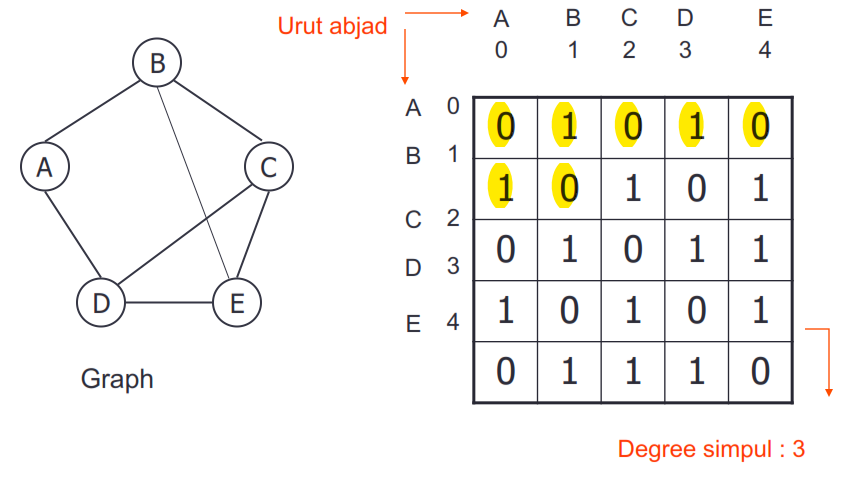
Jika setiap busur mempunyai nilai yang menyatakan hubungan antara 2 buah simpul, maka busur tersebut dinyatakan memiliki bobot, bobot sebuah busur dapat menyatakan panjang sebuah jalan dari 2 buah titik, jumlah rata-rata kendaraan perhari yang melalui sebuah jalan, dll.

1. Incident

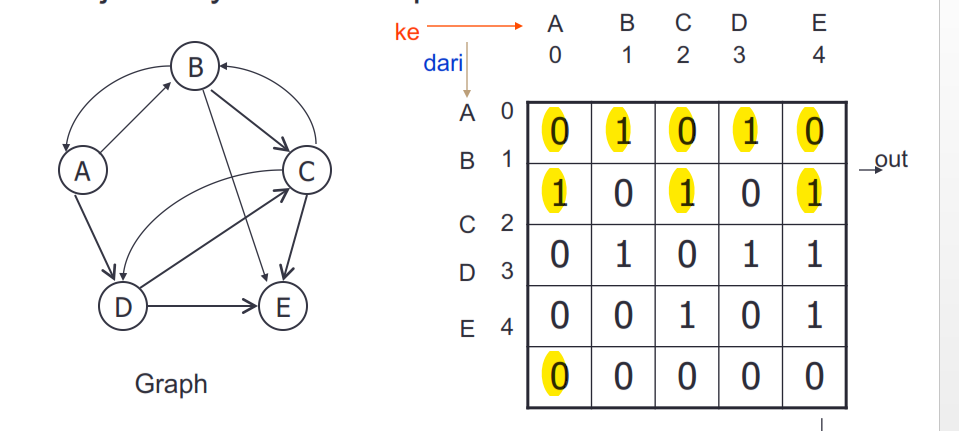
Jika e merupakan busur dengan simpul-simpulnya adalah v dan w yang ditulis e=(v,w), maka v dan w disebut “terletak” pada e, dan e disebut incident dengan v dan w.

1. Degree (derajat), indegree dan outdegree Degree sebuah simpul adalah jumlah busur yang incident dengan simpul tersebut.
2. Indegree sebuah simpul pada graph berarah adalah jumlah busur yang kepalanya incident dengan simpul tersebut, atau jumlah busur yang “masuk” atau menuju simpul tersebut.
3. Outdegree sebuah simpul pada graph berarah adalah jumlah busur yang ekornya incident dengan simpul tersebut, atau jumlah busur yang “keluar” atau berasal dari simpul tersebut.
4. Adjacent Pada graph tidah berarah, 2 buah simpul disebut adjacent bila ada busur yang menghubungkan kedua simpul tersebut. Simpul v dan w disebut adjacent. Pada graph berarah, simpul v disebut adjacent dengan simpul w bila ada busur dari w ke v.
5. Successor dan Predecessor Pada graph berarah, bila simpul v adjacent dengan simpul w, maka simpul v adalah successor simpul w, dan simpul w adalah predecessor dari simpul v.
6. Path Sebuah path adalah serangkaian simpul-simpul yang berbeda, yang adjacent secara berturut-turut dari simpul satu ke simpul berikutnya.

  Adjacency Matrix Graph tak berarah



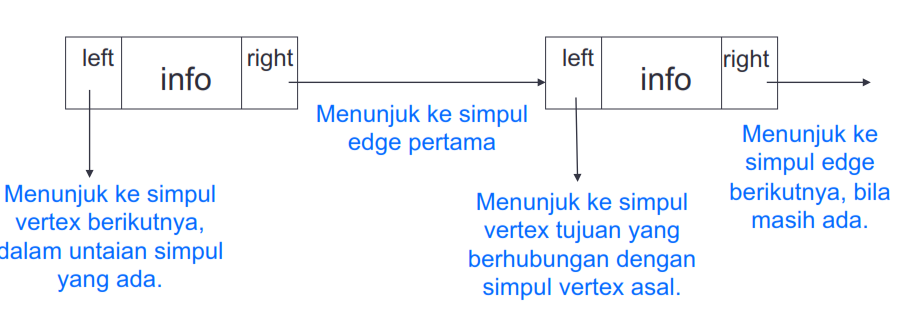
Adjacency Matrix Graph berarah



RepresentasiGraph dalam bentuk Linked List

•  Adjency List graph tak berarah

•  Digambarkan sebagai sebuah simpul yang memiliki 2 pointer.



# **1.4 Perbedaan tree dan graph**

Tree terdapat pada Pohon (Tree) selalu terdapat Path atau Jalur yang menghubungkan setiap simpul dalam dua pohont, tree adalah salah satu bentuk terhubung yang tidak mengandung sirkuit. (Tree) dapat juga didefinisikan sebagai kumpulan elemen yang salah satu elemennya disebut dengan Akar (Root) dan sisa elemen lain (Simpul) yang terpecah menjadi sejumlah himpunan yang saling tidak berhubungan yang disebut dengan Subpohon (Subtree) atau cabang Sedangkan Graph Mengandung sirkuit berarah maupun tidak berarah. Pohon mempunyai Simpul sebanyak n, maka

banyaknya ruas atau edge adalah (n-1), mempunyai Simpul Khusus yang disebut Root, jika Simpul tersebut memiliki derajat keluar >= 0, dan derajat masuk = 0. Mempunyai Simpul yang disebut sebagai Daun / Leaf, jika Simpul tersebut berderajat keluar = 0, dan berderajat masuk = 1. Tree tidak mempunya alur cycel sebaliknya Grap bisa mengandung unsur cycle.

1. Struktur Data Tree = keterhubungan hirarkis.
2. Struktur Data Graph = keterhubungan tak terbatas antara entitas data.
3. Pada Graph tidak memiliki root

Tree: setiap node kecuali root hanya memiliki sebuah parent.

Graph: dapat memiliki lebih dari sebuah parent. Graph terdapat graph berarah dan tidak berarah sedangkan tree hanya mempunyai path tidak berarah.

size,  height, serta average dari length-nya. Tree tidak boleh kosong. Pada graph Shortest Path Problem adalah problem bagaimana kita mencari sebuah jalur pada graf yang meminimalkan jumlah bobot sisi pembentuk jalur tersebut dengan algorit djikstrak.

# **Daftar Pustaka**

<http://t4urusboy08.blogspot.co.id/>

Bsi, (2009). struktur data, E-book

https://lunarphue.wordpress.com/information-technology/asd/artikel-kuliah-hahahaha-xd/