

Teori bahasa dan automata

Pertemuan 1

Mahasiswa mampu menjelaskan Kedudukan Teori Bahasa Automata pada ilmu Komputer.

Materi

- Pengenalan: bahasa dan Automata
- Himpunan
- Fungsi
- Relasi
- Graph
- Hirarki Chomsky
- ϵ -NFA

- ***Konsep Bahasa dan Automata***

Teori bahasa membicarakan bahasa formal (*formal language*), terutama untuk Bahasa. **Bahasa** di dalam kamus adalah suatu sistem yang meliputi pengekspresian gagasan, fakta, konsep, termasuk sekumpulan simbol-simbol dan aturan untuk melakukan manipulasinya. Bahasa bisa juga disebut sebagai rangkaian simbol- simbol yang mempunyai makna.

- **Otomata** merupakan suatu sistem yang terdiri atas sejumlah berhingga *state*, di mana *state* menyatakan informasi mengenai *input*. Otomata juga dianggap sebagai mesin otomatis (bukan mesin fisik) yang merupakan suatu model matematika dari suatu sistem yang menerima input dan menghasilkan output, serta terdiri dari sejumlah berhingga state.

Teori bahasa dan automata merupakan bagian ilmu komputer berupa model dan gagasan yang mendasari mengenali komputasi. Automata adalah mesin abstrak yang dapat mengenali (*recognize*), menerima (*accept*), atau membangkitkan (*generate*) sebuah kalimat dalam bahasa tertentu.

Beberapa Pengertian Dasar

- Simbol adalah sebuah entitas abstrak (seperti halnya pengertian *titik* dalam geometri). Sebuah huruf atau sebuah angka adalah contoh simbol.
- String adalah deretan terbatas (*finite*) simbol-simbol. Sebagai contoh, jika a , b , dan c adalah tiga buah simbol maka $abcb$ adalah sebuah string yang dibangun dari ketiga simbol tersebut.
- Jika w adalah sebuah string maka panjang string dinyatakan sebagai $|w|$ dan didefinisikan sebagai cacahan (banyaknya) simbol yang menyusun string tersebut. Sebagai contoh, jika $w = abcb$ maka $|w| = 4$.
- String hampa adalah sebuah string dengan nol buah simbol. String hampa dinyatakan dengan simbol ε (atau \wedge) sehingga $|\varepsilon| = 0$. String hampa dapat dipandang sebagai simbol hampa karena keduanya tersusun dari nol buah simbol.
- Alfabet adalah himpunan hingga (*finite set*) simbol-simbol

Operasi Dasar String

- Prefik string w adalah string yang dihasilkan dari string w dengan menghilangkan *no*/ atau lebih simbol-simbol paling belakang dari string w tersebut.

Contoh : abc , ab , a , dan ε adalah semua $\text{Prefix}(x)$

- ProperPrefix string w adalah string yang dihasilkan dari string w dengan menghilangkan *satu* atau lebih simbol-simbol paling belakang dari string w tersebut.

Contoh : ab , a , dan ε adalah semua $\text{ProperPrefix}(x)$

- Postfix (atau Sufix) string w adalah string yang dihasilkan dari string w dengan menghilangkan *no*/ atau lebih simbol-simbol paling depan dari string w tersebut.

Contoh : abc , bc , c , dan ε adalah semua $\text{Postfix}(x)$

Operasi Dasar String (Lanjutan)

- ProperPostfix (atau PoperSufix) string w adalah string yang dihasilkan dari string w dengan menghilangkan *satu* atau lebih simbol-simbol paling depan dari string w tersebut.

Contoh : bc , c , dan ε adalah semua ProperPostfix(x)

- Head string w adalah simbol paling depan dari string w .

Contoh : a adalah Head(x)

- Tail string w adalah string yang dihasilkan dari string w dengan menghilangkan simbol paling depan dari string w tersebut.

Contoh : bc adalah Tail(x)

Operasi Dasar String (Lanjutan)

- Substring string w adalah string yang dihasilkan dari string w dengan menghilangkan *nol* atau lebih simbol-simbol paling depan dan/atau simbol-simbol paling belakang dari string w tersebut.

Contoh : $abc, ab, bc, a, b, c,$ dan ε adalah semua Substring(x)

- ProperSubstring string w adalah string yang dihasilkan dari string w dengan menghilangkan *satu* atau lebih simbol-simbol paling depan dan/atau simbol-simbol paling belakang dari string w tersebut.

Contoh : $ab, bc, a, b, c,$ dan ε adalah semua Substring(x)

- Subsequence string w adalah string yang dihasilkan dari string w dengan menghilangkan *nol* atau lebih simbol-simbol dari string w tersebut.

Contoh : $abc, ab, bc, ac, a, b, c,$ dan ε adalah semua Subsequence(x)

Operasi Dasar String (Lanjutan)

- Proper Subsequence string w adalah string yang dihasilkan dari string w dengan menghilangkan *satu* atau lebih simbol-simbol dari string w tersebut.

Contoh : ab, bc, ac, a, b, c , dan ε adalah semua Subsequence(x)

- Concatenation adalah penyambungan dua buah string. Operator concatenation adalah *concat* atau tanpa lambang apapun.

Contoh : $\text{concat}(xy) = xy = abc123$

- Alternation adalah pilihan satu di antara dua buah string. Operator alternation adalah *alternate* atau $|$.

Contoh : $\text{alternate}(xy) = x | y = abc \text{ atau } 123$

- Kleene Closure : $x^* = \varepsilon | x | xx | xxx | \dots = \varepsilon | x | x | x | \dots$
- Positive Closure : $x^+ = x | xx | xxx | \dots = x | x | x | \dots$

Konsep Dasar

- Anggota alfabet dinamakan simbol terminal.
- Kalimat adalah deretan hingga simbol-simbol terminal.
- Bahasa adalah himpunan kalimat-kalimat. Anggota bahasa bisa tak hingga kalimat.
- Simbol-simbol berikut adalah simbol terminal :
 - ❑ huruf kecil, misalnya : a, b, c
 - ❑ simbol operator, misalnya : +, /, dan *
 - ❑ simbol tanda baca, misalnya : (,), dan ;
 - ❑ simbol tanda baca, misalnya : (,), dan ;
 - ❑ string yang tercetak tebal, misalnya : **if**, **then**, dan **else**.
- Simbol-simbol berikut adalah simbol non terminal /Variabel :
 - ❑ huruf besar, misalnya : A, B, C
 - ❑ huruf S sebagai simbol awal
 - ❑ string yang tercetak miring, misalnya : *expr*

TEORI HIMPUNAN

- Pengertian Himpunan merupakan kumpulan dari elemen- elemen. Dan notasi yang digunakan untuk elemen adalah ϵ ,

Contoh :

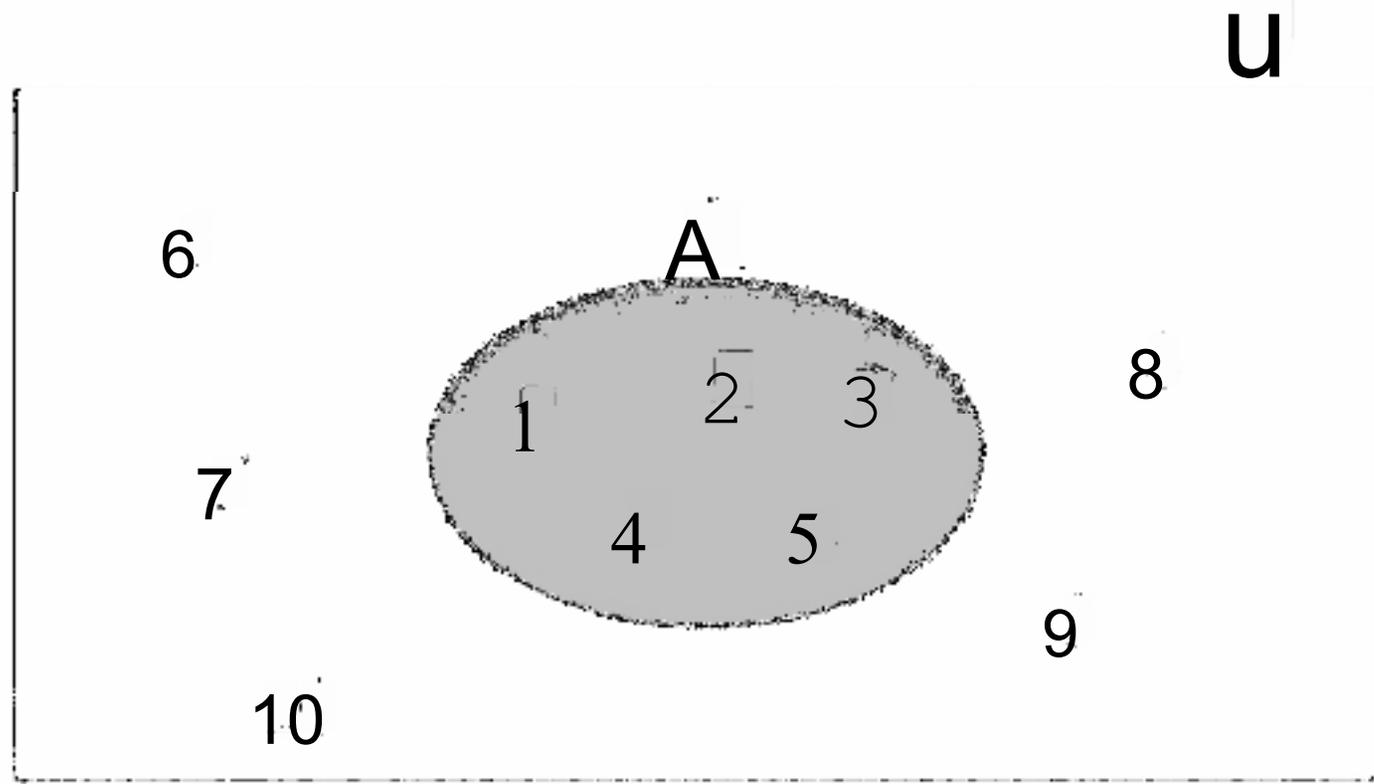
$$a \in X$$

Maka dibaca “a merupakan elemen himpunan X”

- **$b \notin X$**

Maka dibaca “b bukan elemen dari himpunan X”

$$A = \{ 1, 2, 3, 4, 5 \}$$



Himpunan Universal : semua element yang mungkin

$$U = \{ 1, \dots, 10 \}$$

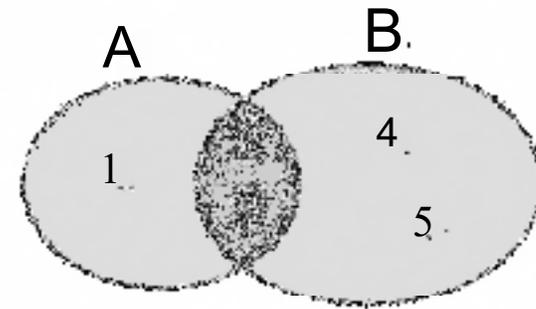
Operasi himpunan

$$A = \{1, 2, 3\}$$

$$B = \{2, 3, 4, 5\}$$

- Union

$$A \cup B = \{1, 2, 3, 4, 5\}$$



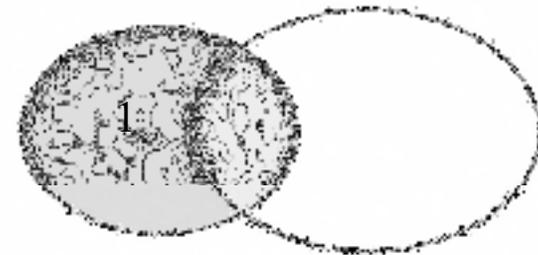
- Intersection

$$A \cap B = \{2, 3\}$$

- Difference

$$A - B = \{1\}$$

$$B - A = \{4, 5\}$$



Venn diagrams

Operasi Himpunan

- Gabungan (Union)

Gabungan himpunan A dan B ditulis dengan $A \cup B$ adalah suatu himpunan yang anggotanya berada di A atau berada di B.

Jadi $A \cup B = \{ x \mid x \in A \text{ atau } x \in B \}$

- Contoh:

$A = \{1, 2, 3\}$ dan $B = \{2, 3, 4, 5\}$. Maka $A \cup B = \{1, 2, 3, 4, 5\}$

Operasi Himpunan

- Irisan (Intersection)

Irisan himpunan A dan B ditulis dengan $A \cap B$ adalah suatu himpunan yang anggotanya berada di A dan juga berada di B.

Jadi $A \cap B = \{x \mid x \in A \text{ dan } x \in B\}$

Contoh:

$A = \{1, 2, 3\}$ dan $B = \{2, 3, 4, 5\}$. Maka $A \cap B = \{2, 3\}$

Operasi Himpunan

- Selisih (Difference)

Selisih himpunan antara himpunan A dan himpunan B ditulisdengan $A-B$, dimana himpunan yang terdapat pada himpunan A tetapi tidak terdapat pada himpunan B.

Jadi $A-B = \{x \mid x \in A \text{ atau } x \notin B\}$

Contoh :

$A = \{1,2,3\}$ dan $B = \{2,3,4,5\}$. Maka $A-B = \{1\}$

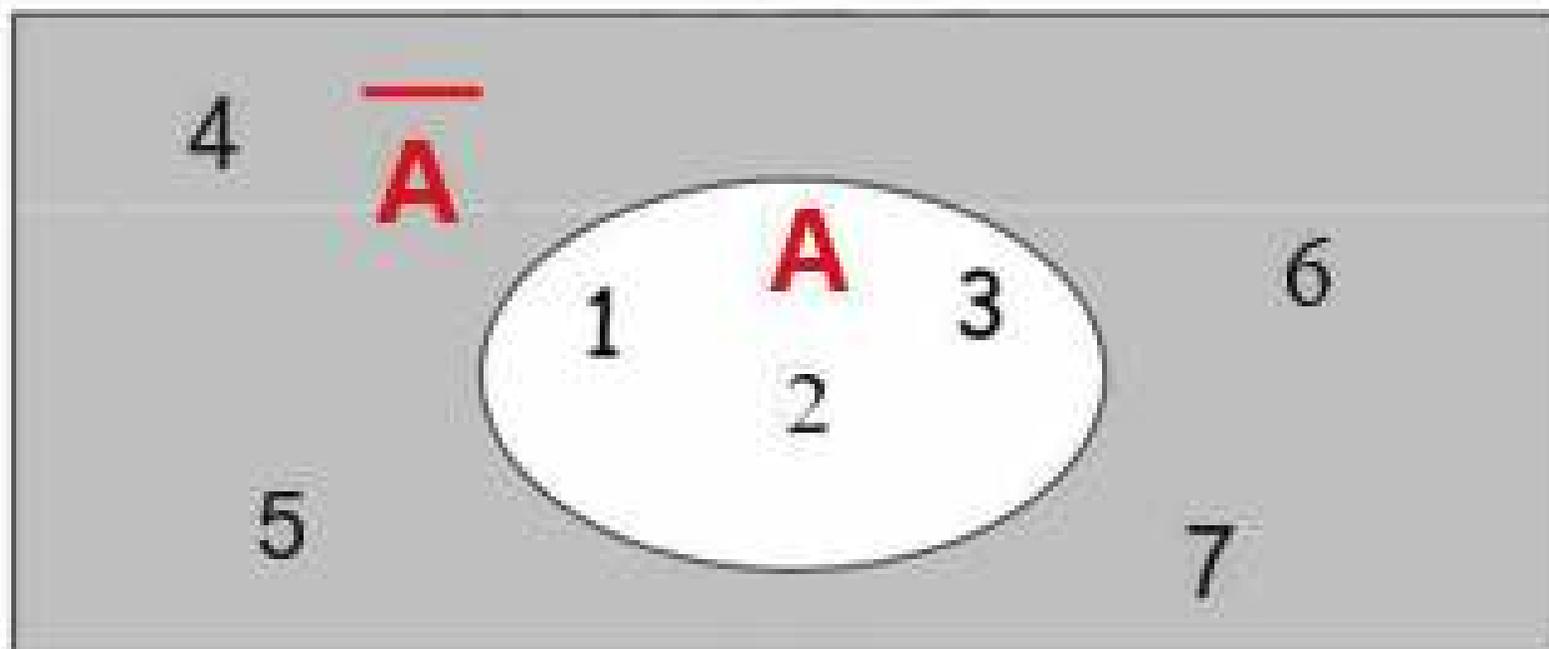
Operasi Himpunan

- Komplemen

Komplemen dari A ditulis dengan " \overline{A} " adalah himpunan yang anggotanya berada dalam himpunan semesta tetapi bukan berada di A .

Jadi $\overline{A} = \{x \mid x \in S, x \notin A\}$

$U = \{1, 2, 3, \dots, 7\}$. Jika $A = \{1, 2, 3\}$ maka $\overline{A} = \{4, 5, 6, 7\}$



Definisi Relasi

Aksi menghubungkan dua objek, satu objek dengan objek lainnya

Contoh relasi dalam kehidupan sehari-hari

- ❖ Relasi orangtua antara bapak dengan anak
- ❖ Relasi memperkerjakan antara majikan dan pegawai

Contoh relasi pada aritmatika

- ❖ Kurang dari
- ❖ Lebih besar dari

Contoh dalam geometric

- ❖ Relasi antara luas bujur sangkar dengan panjang sisinya

Definisi Relasi (lanjutan)

Suatu “relasi R” terdiri dari

a..Sebuah himpunan A

b.Sebuah himpunan B

c.Suatu kalimat terbuka $P(x,y)$ dimana $P(a,b)$ adalah benar atau salah untuk sembarang pasangan terurut (a,b) yang termasuk dalam $A \times B$

Maka dapat disebut R adalah suatu relasi dari A ke B dan menyatakan dengan

$$R = (A, B, P(x,y))$$

Selanjutnya, jika $P(a,b)$ adalah benar ditulis aRb yang berarti “a berhubungan dengan b”

“ $A \times B$ ” berarti A cross B, yang didefinisikan sebagai $\{ \langle a, b \rangle \mid a \in A \text{ dan } b \in B \}$

Via: aku senang permen dan coklat

Andre: aku senang coklat dan es krim

Ita: aku suka es krim

Dari contoh di atas dapat dibuat dua himpunan, yaitu :

-Himpunan A adalah himpunan nama orang

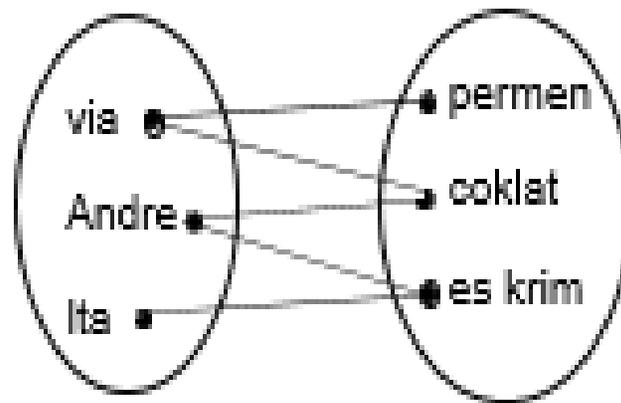
$A = \{ \text{Via, Andre, Ita} \}$

-Himpunan B adalah himpunan makanan kesukaan

$B = \{ \text{es krim, coklat, permen} \}$

Relasi dari himpunan A ke himpunan B adalah "makanan kesukaan" dan dapat dinyatakan dengan 3 cara

a. Diagram Relasi

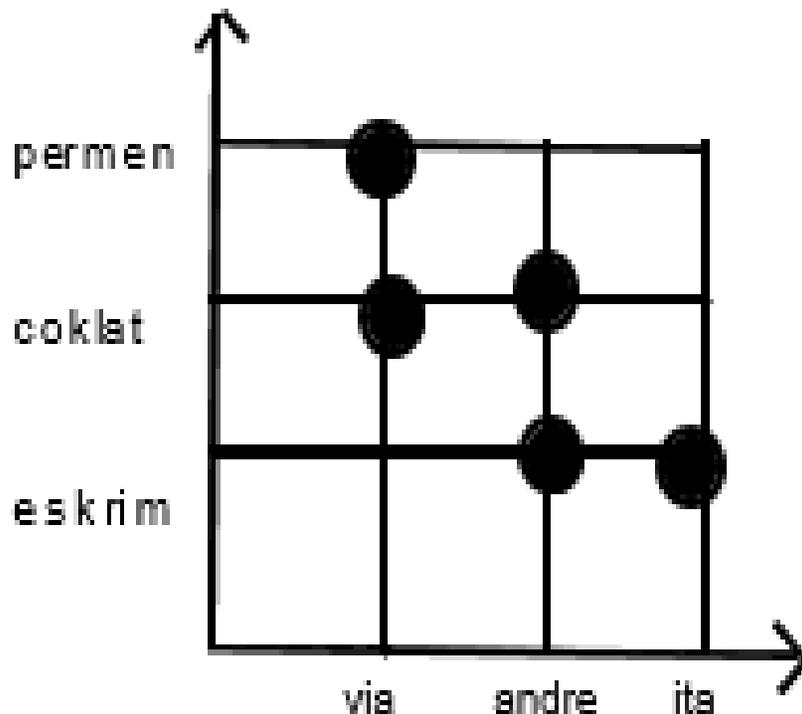


b. Himpunan pasangan berurutan (Relasi biner)

$\{ (Via, permen) , (Via, coklat) , (Andre, coklat) , (Andre, es krim) , (Ita, es krim) \}$

$$D(R) = \{a \mid (\exists b)(a,b) \in S\}$$

c. Diagram Cartesius



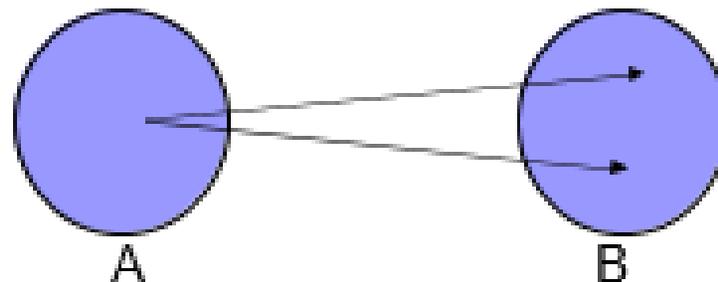
$$A \quad B \quad a,b \mid a \quad A,b \quad B$$

- Dapat dilihat bahwa $R \subseteq (A \times B)$,
- A adalah daerah asal R , dan B adalah daerah hasil R .
- $(Via, permen) \in R$ atau $Via R permen$
- $(Via, es krim) \notin R$ atau $Via R es krim$.

Fungsi

PENGERTIAN FUNGSI

- Definisi : Misalkan A dan B dua himpunan takkosong. Fungsi dari A ke B adalah aturan yang mengaitkan setiap anggota A dengan tepat satu anggota B.
- ATURAN :
 - setiap anggota A harus habis terpasang dengan anggota B.
 - tidak boleh membentuk cabang seperti ini.



Fungsi

ILUSTRASI FUNGSI

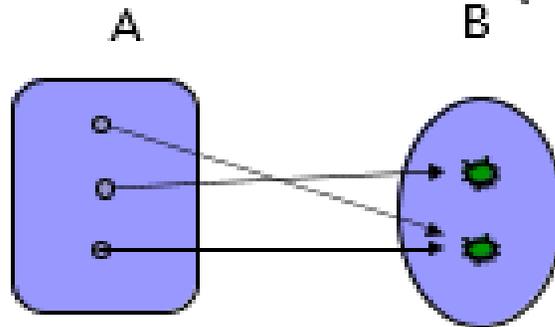


Ditulis $f : A \rightarrow B$, dibaca f adalah fungsi dari A ke B . A disebut domain, B disebut kodomain. Elemen $a \in A$ disebut argumen dan $f(a) \in B$ disebut bayangan (image) dari a .

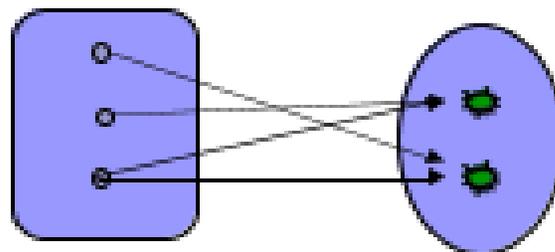
Himpunan $R_f := \{ y \in B : y = f(x) \text{ untuk suatu } x \in A \}$ disebut daerah jelajah (range) fungsi f dalam B . Bila $S \subset A$ maka himpunan $f(S) := \{ f(s) : s \in S \}$ disebut bayangan (image) himp S oleh fungsi f .

Fungsi

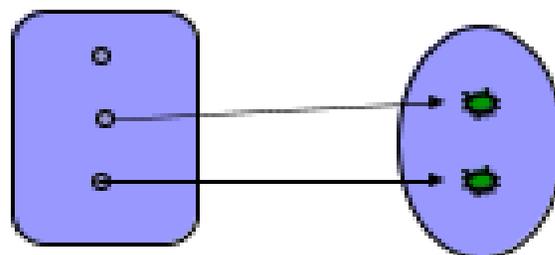
ILUSTRASI FUNGSI (LANJ)



Fungsi



Bukan fungsi, sebab ada elemen A yang mempunyai 2 kawan.



Bukan fungsi, sebab ada elemen A yang tidak mempunyai kawan.

GRAPH

Graph adalah Kumpulan simpul-simpul dan sisi- sisi (edge) yang menghubungkan vertex tersebut.

Istilah dm Graph adalah $G = (V,E)$

Vertex (V) = himpunan tidak kosong dari simpul simpul

Edge (E) = himpunan sisi (edges) yang menghubungkan sepasang simpul

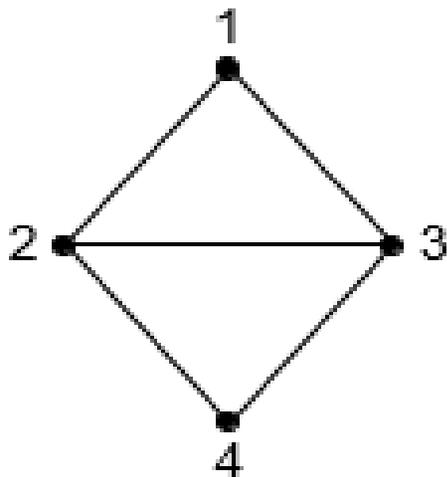
Path: Gandingan beberapa sisi/ edge dan simpul **Tree** (Pohon)

❖ Jenis – Jenis Graph

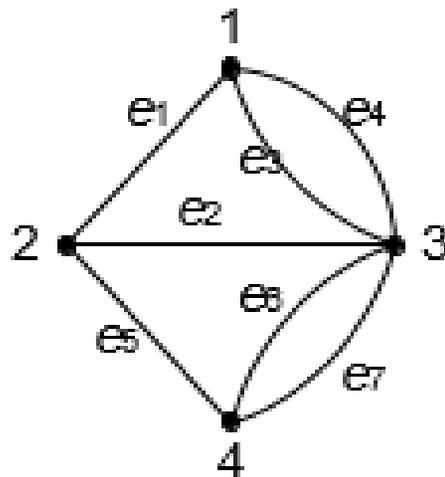
Berdasarkan ada tidaknya gelang atau sisi ganda pada suatu graph, maka graph dibagi menjadi :

1. Graph sederhana (simple graph)
2. Graph Tak Sederhana(unsimple graph)

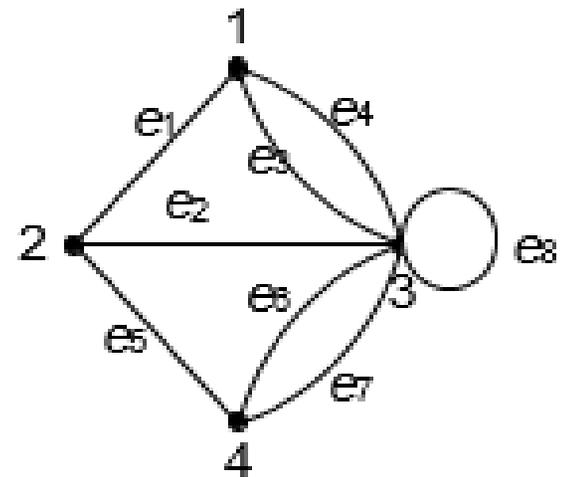
Jenis- Jenis Graph



G1



G2

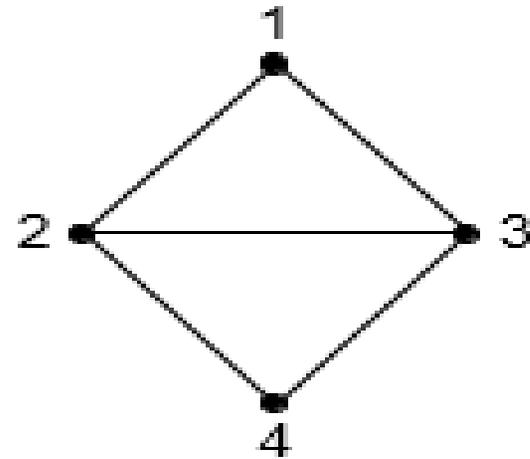


G3

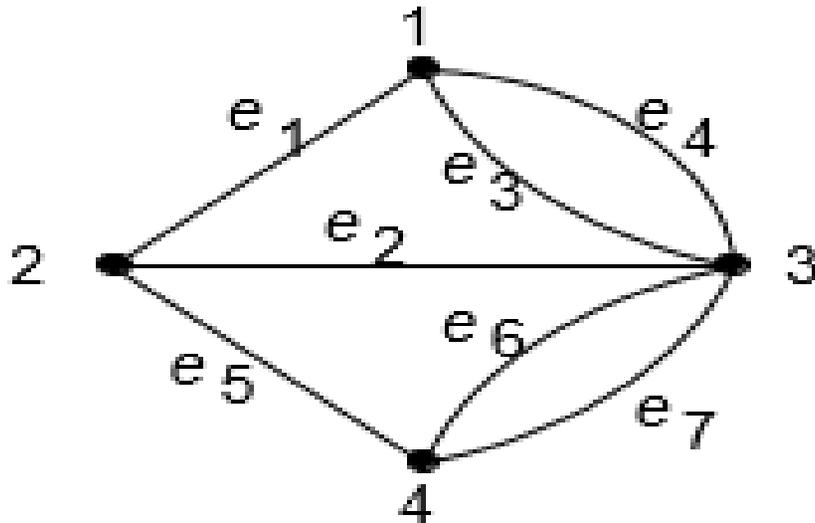
G_1 adalah graph dengan

$$V = \{ 1, 2, 3, 4 \}$$

$$E = \{ (1, 2), (1, 3), (2, 3), (2, 4), (3, 4) \}$$



■ Graph G_2



G_2 adalah graph dengan

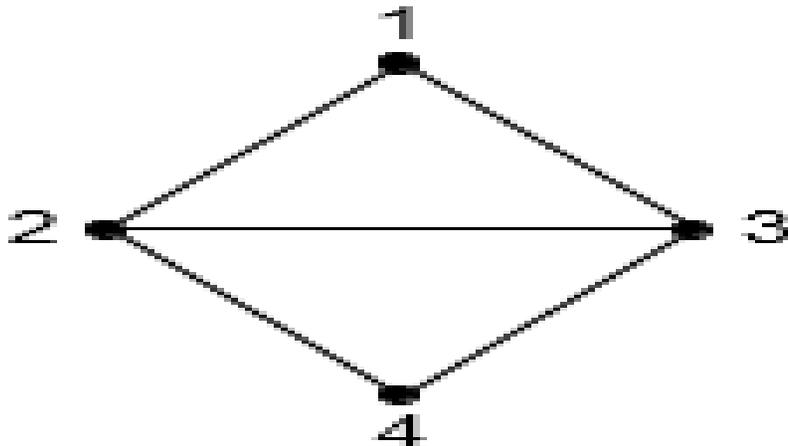
$$V = \{ 1, 2, 3, 4 \}$$

$$E = \{ (1, 2), (2, 3), (1, 3), (1, 3), (2, 4), (3, 4), (3, 4) \}$$

$$= \{ e_1, e_2, e_3, e_4, e_5, e_6, e_7 \}$$

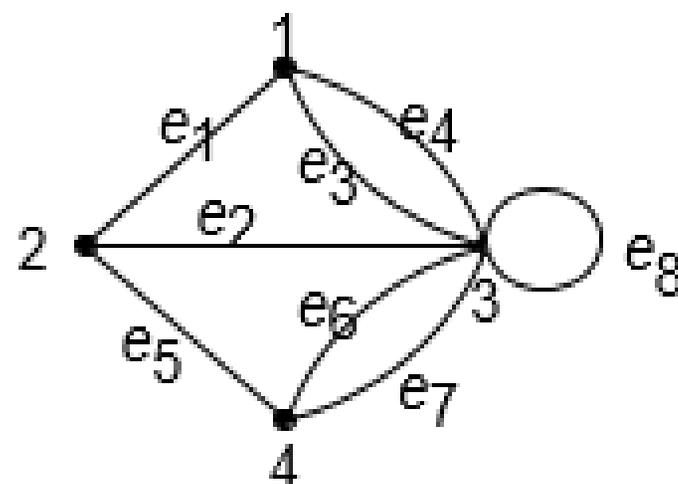
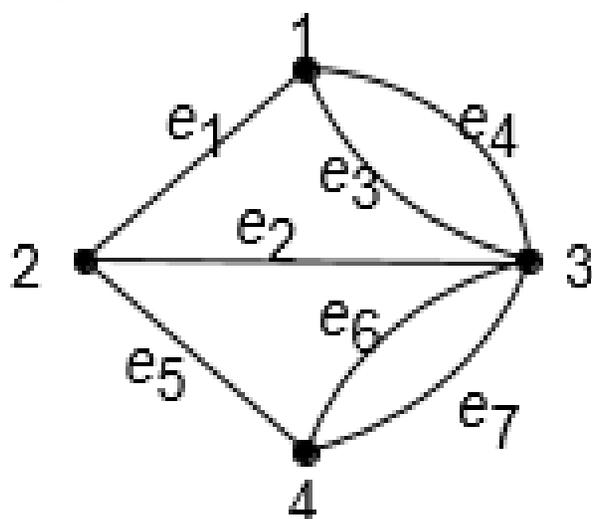
Graph sederhana (*simple graph*)

Graph yang tidak mengandung gelang maupun sisi-ganda dinamakan graph sederhana. G_1 adalah contoh graph sederhana



Graph tak-sederhana (*unsimple-graph*)

Graph yang mengandung sisi ganda atau gelang dinamakan graph tak-sederhana (*unsimple graph*). G_2 dan G_3 adalah contoh graph tak-sederhana

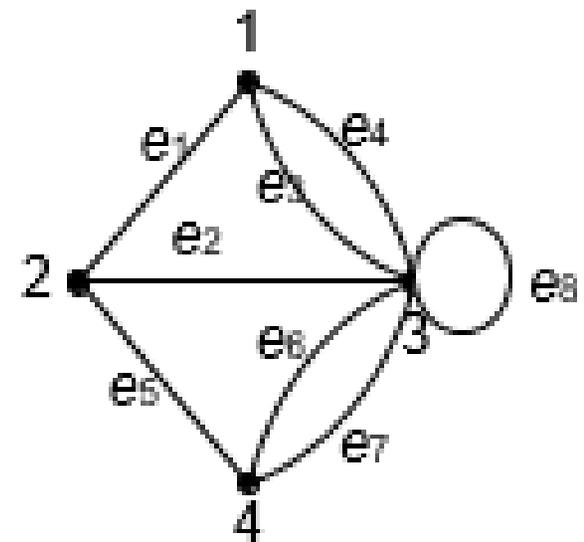
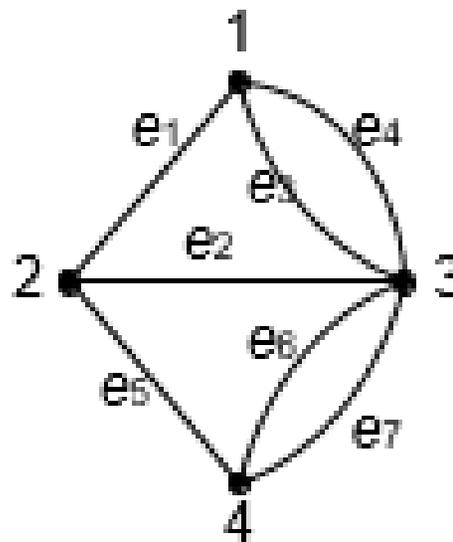
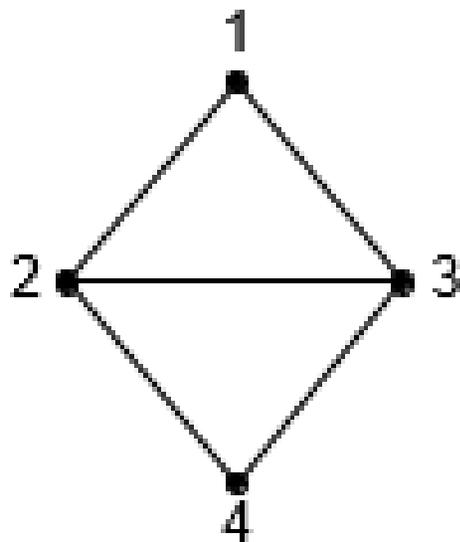


JENIS GRAPH

- Berdasarkan orientasi arah pada sisi, maka secara umum graph dibedakan atas 2 jenis:
- 1. **Graph tak-berarah** (*undirected graph*)
- Graph yang sisinya tidak mempunyai orientasi arah disebut graph tak-berarah. Tiga buah graph pada Gambar 2 adalah graph tak-berarah.
- 2. **Graph berarah** (*directed graph* atau *digraph*)
- Graph yang setiap sisinya diberikan orientasi arah disebut sebagai graph berarah. Dua buah graph pada Gambar 3 adalah graph berarah.

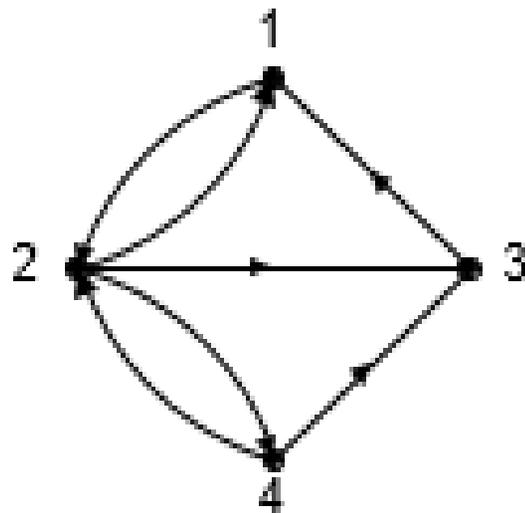
Graph tak-berarah (*undirected graph*)

Graph yang sisinya tidak mempunyai orientasi arah disebut graph tak-berarah. Graph G_1 , G_2 , dan G_3 adalah graph tak-berarah.



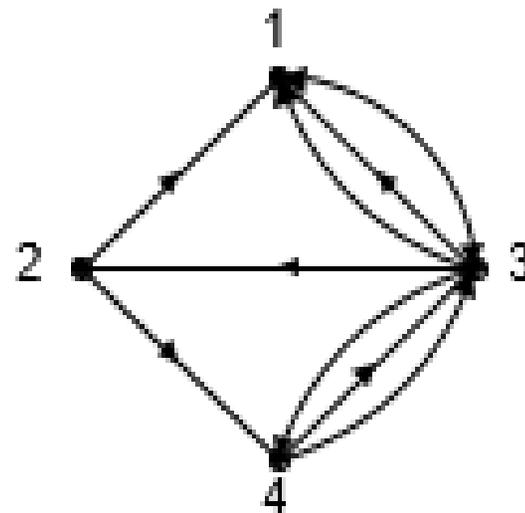
Graph berarah (*directed graph* atau *digraph*)

Graph yang setiap sisinya diberikan orientasi arah disebut sebagai graph berarah.



(a) G4

(a) graph berarah,

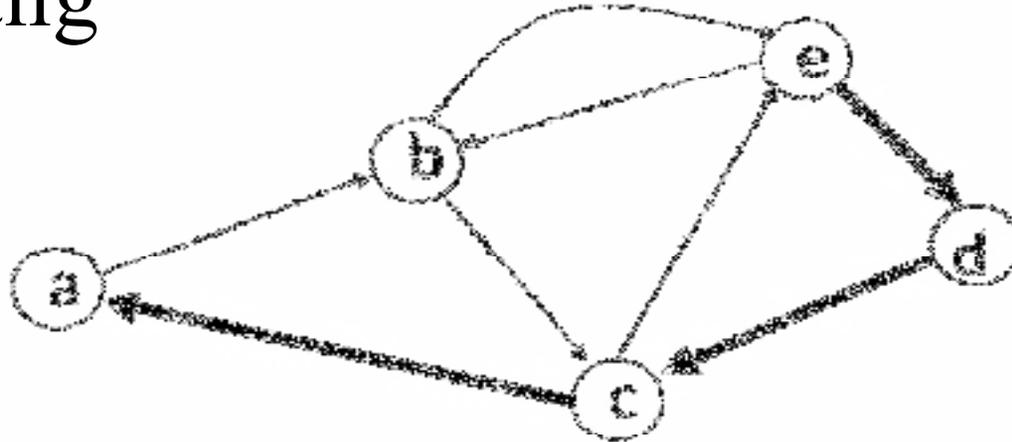


(b) G5

(b) graph-ganda berarah

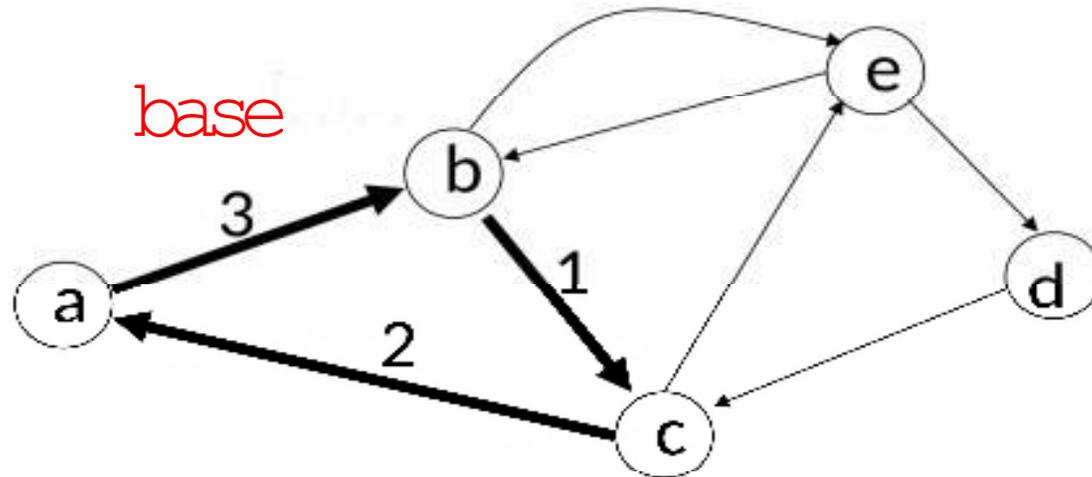
Path

Path adalah lintasan dimana tidak ada node yang diulang



Cycle

Sirkuit/cycle: lintasan yang memiliki node awal dan node akhir yang sama (lintasan yang kembali ke node awal).

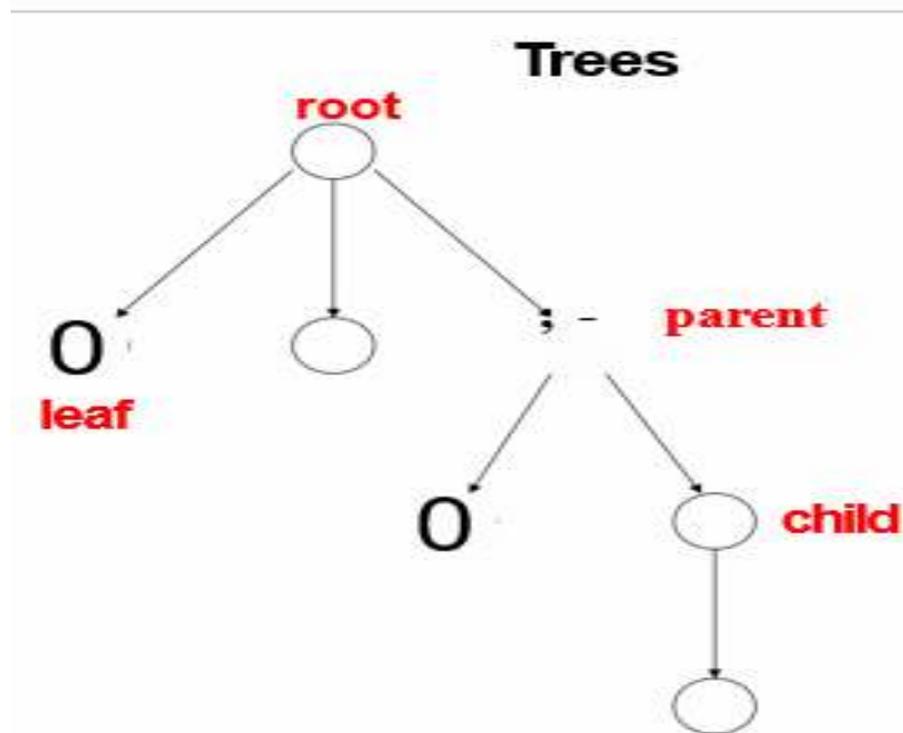


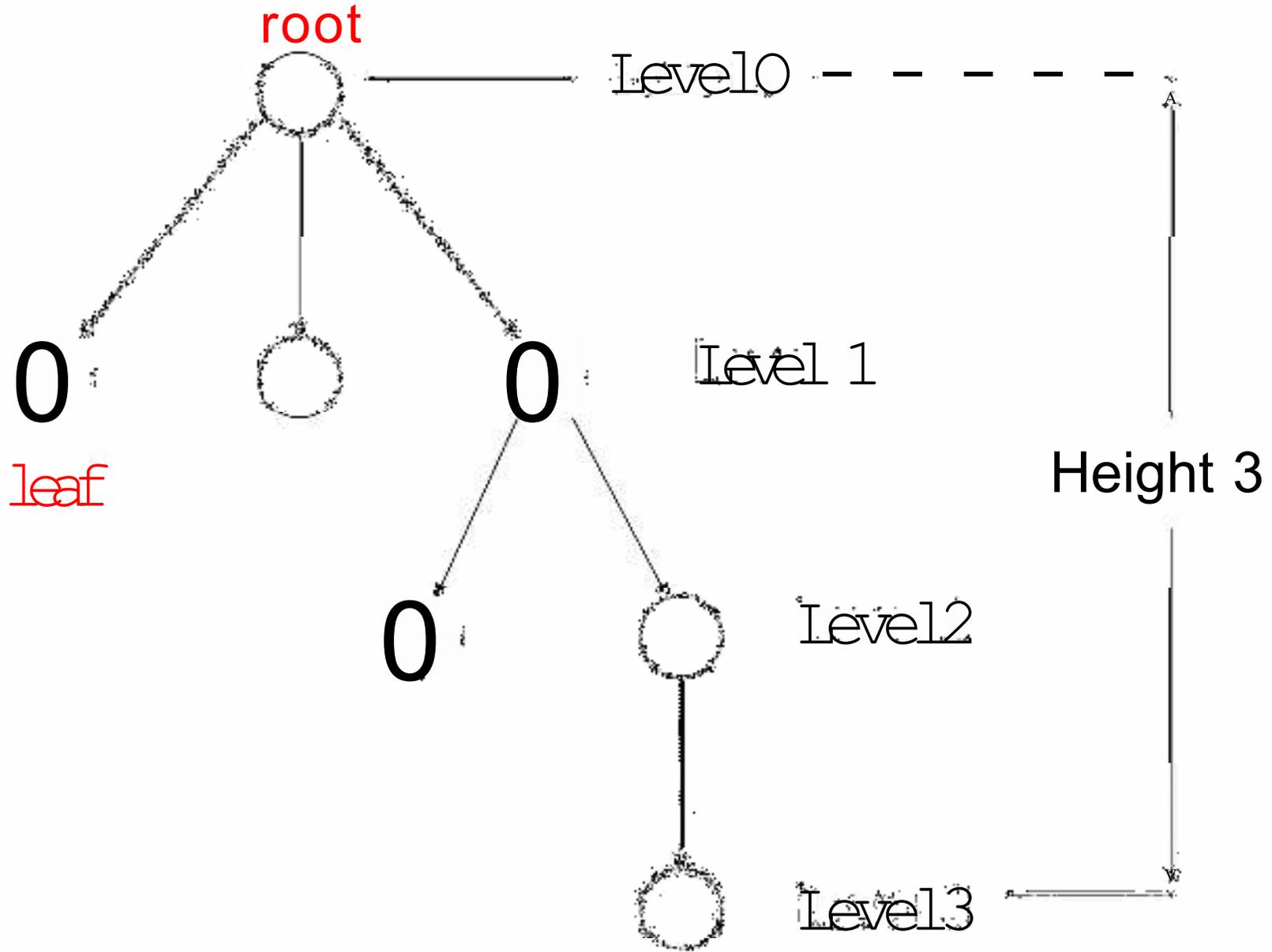
TREE (POHON)

Tree (Pohon) merupakan graph yang terhubung tanpa sirkuit. Sebuah tree, setiap pasang simpul terhubung hanya oleh satu lintasan saja .

Istilah yang digunakan dalam tree adalah :

- Root/ Akar : Suatu simpul tertinggi
- Daun/ leaf : simpul yang tidak memiliki anak
- Cabang / branch : simpul selain daun'





Teori Pembuktian

Prinsip Induksi Matematika merupakan cara yang di gunakan untuk melakukan pembuktian

Langkah Langkah :

1. Tujukan hal itu belaku untuk S_0 , di sebuah basis step
2. Tujukan hal itu belaku untuk S_{n+1} , di sebuah basis step
3. Dapat di tarik kesimpulan hal itu belaku untuk semua n

Contoh

$2^0 + 2^1 + \dots + 2^n = 2^{n+1} - 1$, untuk semua $n \in$ bilangan bulat positif

Maka dapat di tentukan

S_0 untuk $n=0$, maka

$$2^0 = 2^{0+1} - 1$$

$$1 = 2 - 1$$

$$1 = 1$$

S_{n+k} untuk $n=k$, maka juga benar

$$\begin{aligned} 2^0 + 2^1 + \dots + 2^k + 2^{k+1} &= 2^{k+1} - 1 + 2^{k+1} \\ &= 2 \cdot 2^{k+1} - 1 \end{aligned}$$

HIRARKI CHOMSKY

Bahasa	Mesin Otomata	Batasan Aturan Produksi.
Reguler Type 3	Finite State Automata, meliputi : <ul style="list-style-type: none">• DFA• NFA	α adalah sebuah simbol variabel. β maksimal memiliki sebuah simbol variabel yang bila ada terletak di posisi paling kanan, boleh tidak ada.
Bebas Konteks (Context Free) Tipe 2	Push Down Automata	α adalah sebuah simbol variabel.
Context Sensitive Tipe 1	Linier Bounded Automata	$ \alpha \leq \beta $
Unrestricted (Phase Structure) Natural Language Tipe 0	Mesin Turing	Tidak ada batasan.

Tata Bahasa Regular

Aturan :

Simbol pada Sebelah kiri harus berupa sebuah simbol variabel
Simbol pada sebelah kanan maksimal hanya memiliki sebuah symbol variabel dan bila ada terletak di posisi paling kanan.

Contoh :

$A \rightarrow b$ (Diterima)

$a \rightarrow B$ (Ditolak, karena simbol pada sebelah kiri harus berupa sebuah simbol variabel)

$A \rightarrow B$ (Diterima) $A \rightarrow bC$

(Diterima)

$A \rightarrow Bc$ (Ditolak, karena simbol variabel pada sebelah kanan harus berada pada posisi paling kanan)

$A \rightarrow bcD$ (Diterima)

$A \rightarrow bCD$ (Ditolak, karena simbol pada sebelah kanan maksimal hanya memiliki sebuah simbol variabel)

$Ab \rightarrow c$ (Ditolak, karena simbol pada sebelah kiri harus berupa sebuah simbol variabel)

Tata Bahasa Bebas Konteks (CFG)

Aturan :

Simbol pada Sebelah kiri harus berupa sebuah simbol variabel

Contoh :

$A \rightarrow b$ (Diterima)

$A \rightarrow B$ (Diterima)

$A \rightarrow bC$ (Diterima) $A \rightarrow Bc$ (Diterima) $A \rightarrow BcD$

(Diterima)

$A \rightarrow AAA$ (Diterima)

$a \rightarrow b$ (Ditolak, karena simbol pada sebelah kiri harus berupa sebuah simbol variabel) $Ab \rightarrow c$ (Ditolak, karena simbol pada sebelah kiri harus berupa sebuah simbol variabel)

$AB \rightarrow c$ (Ditolak, karena simbol pada sebelah kiri harus berupa sebuah simbol variabel)

Tata Bahasa Context Sensitive

Aturan :

Simbol pada Sebelah kiri harus minimal ada sebuah simbol variabel

Jumlah simbol pada ruas sebelah kiri harus lebih kecil atau sama dengan jumlah simbol pada ruas kanan

Contoh :

$A \rightarrow bc$ (Diterima)

$Ab \rightarrow cd$ (Diterima) $AB \rightarrow CD$ (Diterima)

$ABC \rightarrow DE$ (Ditolak, karena jumlah simbol pada ruas sebelah kiri lebih banyak dari jumlah simbol pada ruas kanan)

$Ab \rightarrow cDe$ (Diterima)

$bA \rightarrow cd$ (Diterima)

$a \rightarrow b$ (Ditolak, karena simbol pada sebelah kiri harus minimal ada sebuah simbol variabel)

Tata Bahasa Unrestricted

Aturan :

Simbol pada Sebelah kiri harus minimal ada sebuah simbol variabel

Contoh :

Abcdef \rightarrow g (Diterima)

aBCdE \rightarrow GHIJKL (Diterima)

abcdef \rightarrow GHIJKL (Ditolak, karena simbol pada sebelah kiri tidak ada sebuah simbol variabel)