Matematika Diskrit

1. Munir, Rinaldi. "(Buku Teks Ilmu Komputer) Matematika Diskrit". Informatika bandung. Bandung.2001
2. Limbong. Prijono."(Berhasil gemilang menguasai) Matematika Diskrit disertai contoh-Contoh Soal". CV. Utomo Bandung. Bandung.2006

Tujuan Pembelajaran

Agar mahasiswa dapat memahami dan mengerti tentang dasar-dasar Matematika Diskrit yang nantinya diterapkan dan sebagai prasyarat untuk matakuliah yang lebih tinggi disemester berikutnya Cont: Automata, Sistem Basis data, Rangkaian Listrik, Rangkaian Digital Dll

Dalam kehidupan sering kita membicarakan objek-objek diskrit (bulat) misalnya komputer, buku, mahasiswa dan lain-lain. Ilmu komputer atau informatika menjadikan objek diskrit ini sebagai titik pokok pembicaraan. Data yang diolah oleh komputer adalah data dalam bentuk diskrit, misalnya data angka, data karakter, data suara(digital), data gambar(digital).

Data diskrit akan kita pelajari dalam mata kuliah Matematika diskrit, dimana terminologi dasar tentang objek diskrit adalah **himpunan**.

1. HIMPUNAN

Himpunan adalah kumpulan benda atau objek nyata maupun abstrak yang mempunyai sifat-sifat tertentu yang sama.

Notasi:

Nama himpunan : A,B,C,…

Anggota himpunan : a,b,c,…

Contoh:

Himpunan software under windows:

A = { MsWord,MsExcel,MsPowerPoint,…}

atau

B = { x I x software under windows }

Cara menuliskan himpunan A disebut tabulasi(mendaftar semua anggotanya)

Cara menuliskan himpunan B disebut deskripsi(menyebutkan sifat-sifat anggotanya)

Masing-masing objek dalam himpunan A disebut anggota atau elemen himpunan, dituliskan:

$x \in A$ artinya x anggota himpunan A

$x \notin A$ artinya x bukan anggota himpunan A

n(A) artinya banyaknya anggota A

Kesamaan Dua Himpunan dan Subhimpunan

Dua himpunan A dan B dikatakan sama jika dan hanya jika keduanya bersama-sama memiliki anggota yang sama.

Contoh:

A = { a,b,c,d }

B = { c,d,a,b }

Maka

A = B

Himpunan A dikatakan sub himpunan B

jika dan hanya jika semua elemen-elemen A adalah juga menjadi elemen-elemen B

Contoh:

A = {Win3.1,Win3.11,Win95}

B = {Win3.1,Win3.11,Win95,Win97,
Win98,Win2000,WinXP}

Maka:

$$A \subset B$$

Macam-macam himpunan

- **Himpunan kosong(Empty set)**

  adalah himpunan yang tidak memiliki anggota.

  Notasi: ∅, { }

Contoh:

A = himpunan software aplikasi yang bisa dipakai dengan semua sistem operasi

A = ∅ = { }

- **Himpunan tunggal(singleton set)**

  adalah himpunan yang hanya memiliki satu anggota

Contoh:

A = himpunan device yang berfungsi sebagai input device sekaligus output device

A = { touch screen }

- **Himpunan Semesta(universal Set)**

  Dalam setiap membicarakan himpunan, maka semua himpunan yang ditinjau adalah subhimpunan dari sebuah himpunan tertentu yang disebut himpunan semesta.

Notasi: U

Contoh:

U = semesta pembicaraan, yaitu sistem operasi produksi Microsoft

U = { Win 3.1, …, Win XP }

- **Himpunan Kuasa(Power Set)**

  Adalah himpunan dari semua subhimpunan yang dapat dibuat dari sebuah himpunan

  Notasi : 2^A

  Banyaknya himpunan bagian dari sebuah himpunan A adalah:

  2^x ,x adalah banyak elemen A

Contoh:

A = {mouse, keyboard}

B = { monitor, printer, scanner}

Maka:

2^A = { A, {mouse},{keyboard}, ∅ }

2^B = {{monitor},{printer},{scanner},
{monitor,printer},{monitor,scanner},
{printer,scanner}, ∅ }

Banyaknya himpunan bagian dari A = 2^2 = 4

Banyaknya himpunan bagian dari A = 2^3 = 8

Operasi Himpunan

- **Union/Gabungan**

  Gabungan dua himpunan A dan B adalah himpunan yang anggotanya semua anggota A atau B atau keduanya. $A \cup B = \{x \; x \in A \text{ atau } x \in B\}$

  Notasi: $A \cup B$, $A + B$

Contoh:

A = { mouse, keyboard, scanner} ,

B = { monitor,printer}, C = { mouse, keyboard, CPU }

maka:

$A \cup B$ = {mouse, keyboard, scanner, monitor,printer}

$A \cup C$ = { mouse, keyboard, scanner , CPU }

- **Intersection/Irisan**

  Irisan dari dua himpunan A dan B adalah himpunan yang anggotanya dimiliki bersama oleh himpunan A dan B

  Notasi : $A \cap B = \{x \; x \notin A \text{ dan } x \in B\}$

Contoh:

A = { mouse,keyboard,touch sreen}

B = { monitor, touch screen, printer, scanner}

C = { monitor,printer, scanner}

Maka:

$A \cap B$ = { touch screen }

$A \cap C$ = { }

- **Relative Complement/Selisih**

  Selisih antara dua himpunan A dan B adalah himpunan yang anggotanya hanya menjadi anggota himpunan A tetapi tidak termasuk anggota himpunan B.

  Notasi: $A - B=\{x \mid x \in A \text{ dan } x \notin B\}$

  Contoh:

  A = { SQLserver,MySQL,MsAcces}

  B = { MySQL,MsAcces,Oracle}

  Maka:

  A — B = {SQL server }

- **Komplemen dari himpunan**

  Komplemen Himpunan A adalah himpunan yang anggotanya bukan anggota A

  Notasi : A’ , A^c

  Contoh:

  U = { Win3.1, Win3.11, Win95,Win97,Win98,
  Win98se, WinME,Win2000, WinXP,… }

  A = { Win3.1, Win3.11, Win95,Win97 }

  A’ = {Win98,Win98se, WinME,Win2000, WinXP,… }

- **Symmetric Difference/Beda Setangkup**

  Beda setangkup dua himpunan A dan B adalah himpunan yang merupakan anggota himpunan A atau anggota himpunan B tetapi bukan merupakan anggota kedua himpunan secara bersamaan.

  Notasi: $A \oplus B=\{x \; x \in A$ dan $x \in B$ tetapi $x \notin A \cap B\}$

  Contoh:

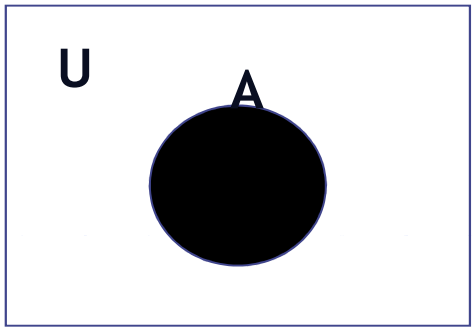
  A = { Win3.1, Win3.11, Win95,Win97 }

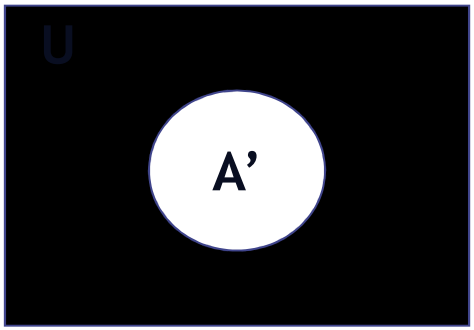
  B = { Win95,Win97,Win98,Win98SE, WinME,Win2000 }

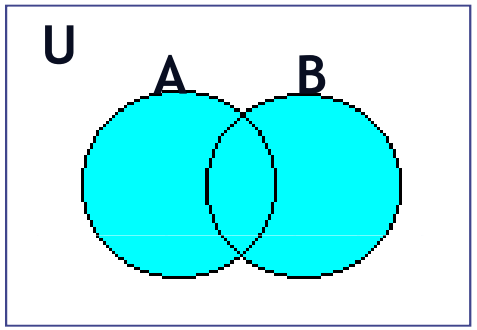
  $A \oplus B$ = { Win3.1, Win3.11, Win98, Win98SE ,WinME, Win2000 }

Diagram Venn

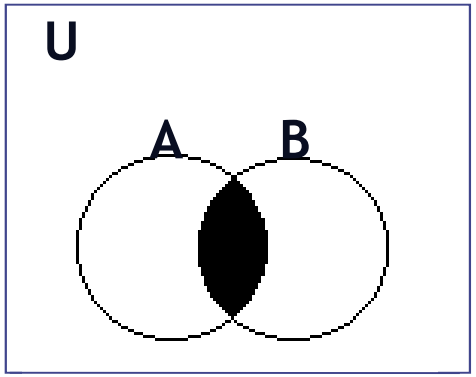
Adalah suatu cara untuk menggambarkan hubungan antara himpunan-himpunan.



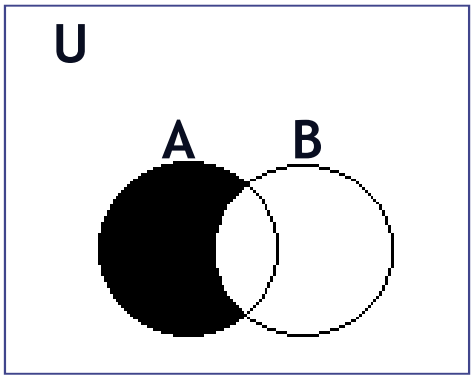




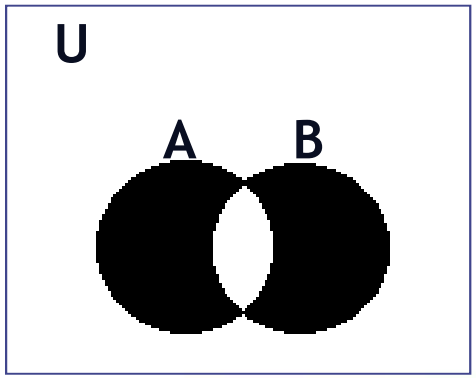
$A \cup B$



$A \cap B$



$A - B$



$A \oplus B$

Hukum-hukum Aljabar Himpunan

1. Hukum Idempoten

$A \cup B = A$

$A \cap A = A$

2. Hukum Komutatif

$A \cup B = B \cup A$

$A \cap B = B \cap A$

3. Hukum Asosiatif

$(A \cup B) \cup C = A \cup (B \cup C)$

$(A \cap B) \cap C = A \cap (B \cap C)$

4. Hukum identitas

$A \cup \varnothing = A$ $A \cap U = A$

$A \cup U = U$ $A \cap \varnothing = \varnothing$

5. Hukum Distributif

$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$

$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$

6. Hukum DeMorgan

$(A \cup B)^c = A^c \cap B^c$

$(A \cap B)^c = A^c \cup B^c$

Contoh:

Sederhanakan : $A \cup (A \cap B)$

$$A \cup (A \cap B) = (A \cap U) \cup (A \cap B)$$
$$= A \cap (U \cup B)$$
$$= A$$

Perhitungan Himpunan Gabungan

- Jumlah anggota dari gabungan himpunan A dan B:

$$N_{A \cup B} = N_A + N_B - N_{A \cap B}$$

- Jumlah anggota dari gabungan himpunan A , B dan C

$$N_{A \cup B \cup C} = N_A + N_B + N_C - N_{A \cap B} - N_{B \cap C} - N_{A \cap C} + N_{A \cap B \cap C}$$

Contoh:

Dalam suatu kelas x semua siswa belajar penggunaan software Maple dan Matlab. Kalau dihitung yang belajar Maple ada 20 siswa, 25% diantaranya juga belajar Matlab. Apabila diketahui perbandingan jumlah mahasiswa yang belajar Maple dan Matlab adalah 5 : 4, maka berapa jumlah mahasiswa dikelas x tersebut? Berapa jumlah mahasiswa yang hanya belajar Matlap?

Latihan

1. Tuliskan dalam bentuk deskripsi himpunan berikut uni:

   A = { Adobe Photoshop, Macromedia Fireworks, PrintShopPro, GIMP, … }

   B = { PHP, ASP, Cold Fusion, … }

   C = { Windows, Linux, Unix, MacOS, OS/2, … }

   D = { disket, CD-R, Hardisk, … }

2. Misalkan semesta pembicaraan adalah sistem produksi Microsoft dan himpunan-himpunan lainnya dinyatakan oleh:

   A = { win3.1, win3.11,win,95,win97}

   B = { win97,win98,win98SE,winME}

   C = { winME,win2000,winXP,… }

   carilah: a. $(A \cup B)- B$ b. $(A \cap B) \cup C'$ c. $(A \oplus B)-C$

   d. $(B-C) \oplus A$ e. $(A \cap B) \cup (A \cap C)'$

f. $(A-B) \cap C'$ g. 2^A h. 2^B

i. $N_{A \cap B}$ j. $N_{A \cup B}$

3. Dari 35 orang programmer yang mengikuti wawancara untuk sebuah pekerjaan diketahui

25 menguasai Pascal

28 menguasi C++

2 tidak menguasai keduanya

Berapa orang yang menguasai keduanya?

RELASI DAN FUNGSI

PERTEMUAN 2

Hubungan antara elemen himpunan dengan elemen himpunan lain disebut dengan relasi.

Misalkan variabel x dan y adalah bilangan real dalam interval tertutup $[x_1,x_2]$ dan $[y_1,y_2]$

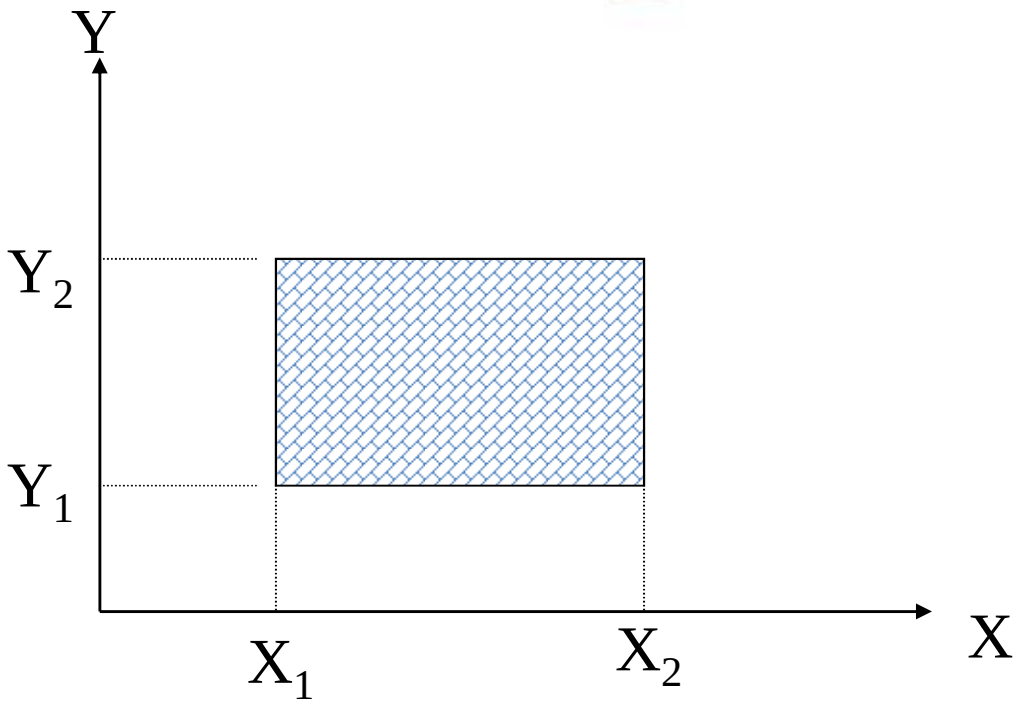
maka:

$XxY = \{ (x_1,y_1),(x_1,y_2),(x_2,y_1),(x_2,y_2) \}$

$YxX = \{ (y_1,x_1),(y_1,x_2),(y_2,x_1),(y_2,x_2) \}$

$XxX = \{(x_1,x_1),(x_1,x_2),(x_2,x_1),(x_2,x_2) \}$

$YxY = \{ (y_1,y_1),(y_1,y_2),(y_2,y_1),(y_2,y_2) \}$



Maka relasi R antara elemen-elemen dalam himpunan X dan himpunan Y adalah:

$$R \subseteq XxY$$

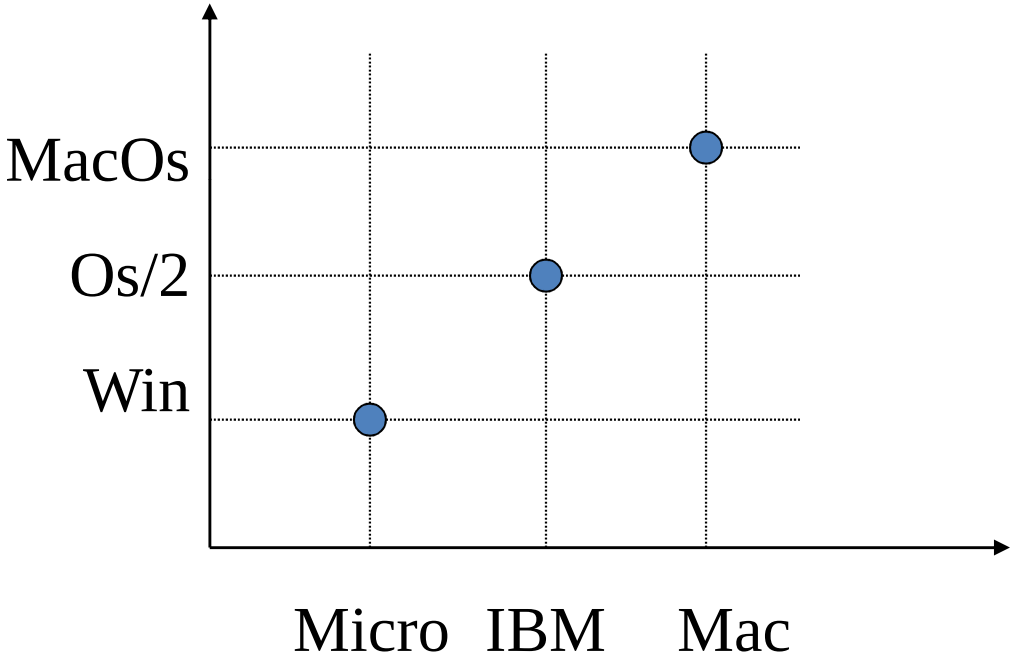
Relasi demikian disebut relasi binary, karena elemen dalam R terdiri dari pasangan 2 himpunan

PEMAPARAN RELASI

- PEMAPARAN KOORDINAT

  misalkan :

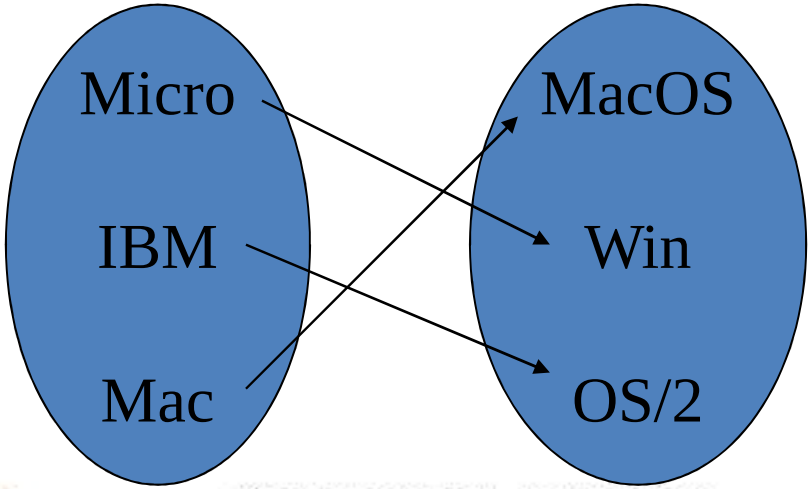
  R = {(Microsoft, Win), (IBM,OS/2), (Mac,MacOs)}



- PEMAPARAN MATRIKS

R	Micro	IBM	Mac
MacOS	0	0	1
OS/2	0	1	0
Win	1	0	0

- PEMAPARAN PEMETAAN



- **PEMAPARAN GRAPH BERARAH**

  Aturan-aturannya sbb:

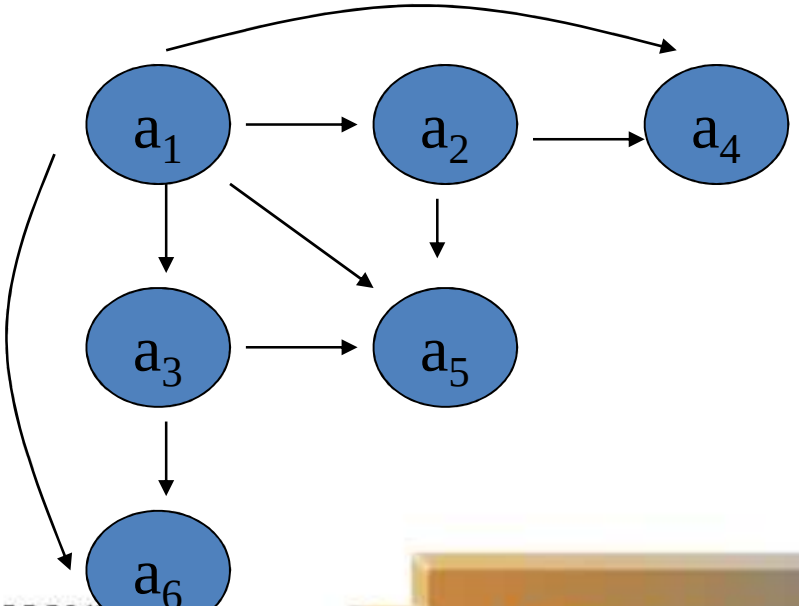
  a. Setiap anggota himpunan X digambarkan dengan lingkaran
  b. Garis berarah antar lingkaran menggambarkan adanya relasi antara anggota himpunan.

Contoh:

a_1 prasyarat tuk semua

a_3 prasyarat a_5 dan a_6

a_6 bukan prasyarat tuk semua



OPERASI DALAM RELASI BINARY

- INVERS RELASI (R^{-1})

  Didefinisikan dengan menukar susunan anggota disemua pasangan yang ada dalam relasi, jadi

  Jika R : X$\rightarrow$Y , maka R^{-1} : Y $\rightarrow$ X

- KOMPOSISI RELASI

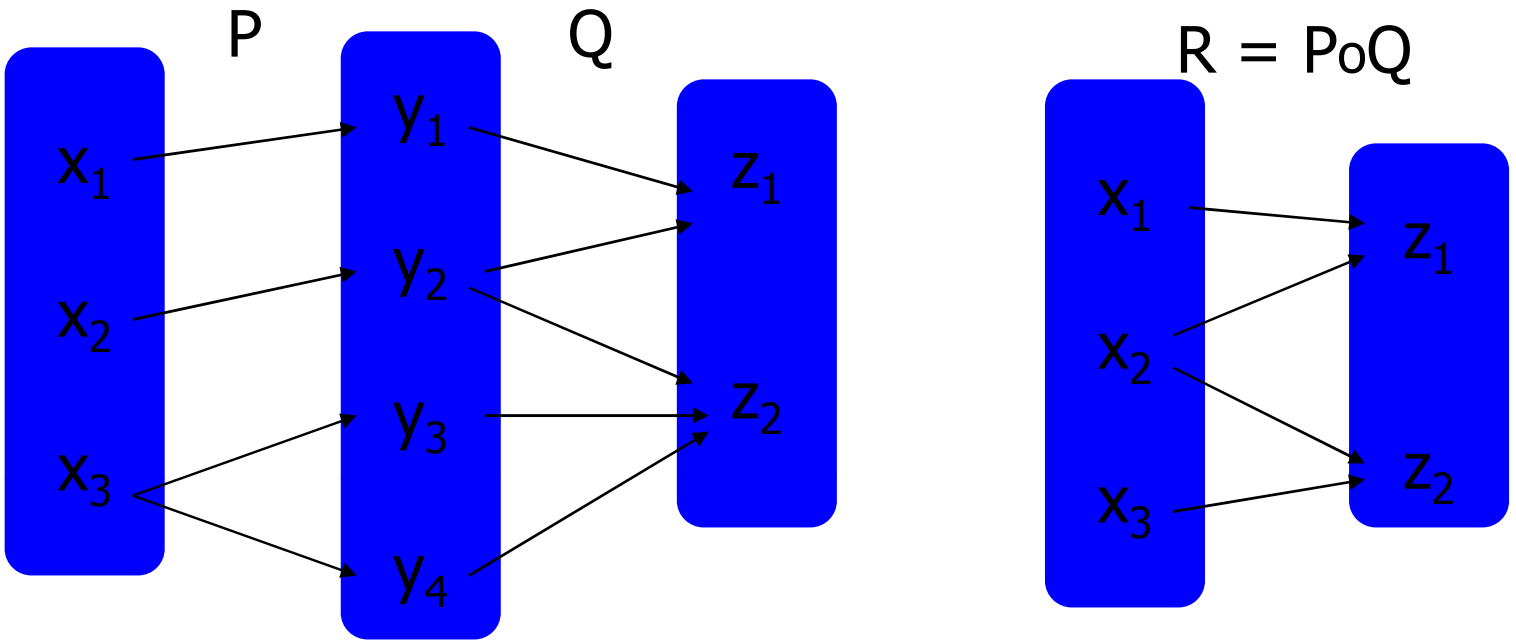
  Operasi mengkombinasikan 2 buah relasi binary yang cocok dan menghasilkan sebuah relasi binary yang baru.

  P : X $\rightarrow$Y dan Q: Y $\rightarrow$Z

  dimana Y di P harus sama dengan di Q

  relasi P ke Q atau PoQ, didefinisikan sebagai relasi: R : X$\rightarrow$Z

Contoh:



Sifat – sifat Relasi Biner

- Refleksif (*reflexive*)

  relasi R pada himp. A disebut reflesif jika (a,a) $\in$ R untuk setiap a $\in$ A

Contoh:

misalkan A={1,2,3} dan relasi R di bawah ini didefinisikan pada himpunan A, maka

a. R = {(1,1),(1,3),(2,1),(2,2),(3,3)} …. refleksif

b. R = {(1,1),(1,3),(2,1),(2,2)} …. Tidak refleksif

- Setangkup (*symmetric*)

  relasi R pada himp. A disebut setangkup jika untuk semua a, b $\in$ A, jika (a, b) $\in$ R, maka (b,a) $\in$ R

Contoh:

Misalkan A={1,2,3} dan relasi R di bawah ini didefinisikan pada himpunan A, maka

a. R = {(1,1),(1,2),(2,3),(2,1),(3,2)} … setangkup

b. R = {(1,1),(1,2),(2,3),(2,1),(3,3)} …tak setangkup

- Menghantar (*transitive*)

  Relasi R pada himpunan A disebut transitif jika (a,b) $\in$ R dan (b,c) $\in$ R maka (a,c) $\in$ R untuk a,b,c $\in$ R

Contoh:

Misalkan A={1,2,3,4} dan relasi R di bawah ini didefinisikan pada himpunan A, maka

a. R = {(2,1),(3,1), (3,2), (4,1),(4,2),(4,3)} …transitif

Pasangan berbentuk		
(a,b)	(b,c)	(a,c)
(3,2)	(2,1)	(3,1)
(4,2)	(2,1)	(4,1)
(4,3)	(3,1)	(4,1)
(4,3)	(3,2)	(4,2)

b. R ={(1,1),(2,3),(2,4),(4,2) …. tidak transitif

Mengkombinasikan Relasi

Jika R_1 dan R_2 masing-masing adalah relasi dari himp. A ke himp. B, maka $R_1 \cap R_2$, $R_1 \cup R_2$, R_1- R_2, $R_1 \oplus R_2$ juga relasi.

Contoh:

Misalkan A={a,b,c} dan {a,b,c,d}. Relasi R_1 = {(a,a),(b,b)(c,c)} dan relasi R_2 = {(a,a),(a,b),(a,c),(a,d)} adalah relasi dari A ke B. kombinasi relasi-relasi tersebut bisa berupa:

$R_1 \cap R_2 = \{(a,a)\}$

$R_1 \cup R_2 = \{(a,a),(b,b),(c,c),(a,b),(a,c),(a,d)\}$

$R_1 - R_2 = \{(b,b),(c,c)\}$

$R_1 \oplus R_2 = \{(b,b),(c,c),(a,b),(a,c),(a,d)\}$

Jika relasi R_1dan R_2 masing-masing dinyatakan dengan matriks M_{R1}dan M_{R2}, maka matriks yang menyatakan gabungan dan irisan dari kedua relasi tersebut adalah

$M_{R1 \cup R2} = M_{R1} \vee M_{R2}$ dan $M_{R1 \cap R2} = M_{R1} \wedge M_{R2}$

$$R_1 = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \quad \text{dan} \quad R_2 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

maka matriks yang menyatakan $R_1 \cup R_2$ dan $R_1 \cap R_2$ adalah: $M_{R1} \vee M_{R2} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}$ dan $M_{R1} \wedge M_{R2} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$

Relasi n-er (n-ary relation)

Tabel 1 PEMAIN

Nomor ID	Nama	Posisi	Umur
22012	Johnson	c	22
93831	Glover	Of	24
58199	Batty	p	18
84341	Cage	c	30
01180	Homer	1b	37
26710	Score	p	22
61049	Johnson	Of	30
39826	Singleton	2b	31

Tabel 1 bisa dinyatakan sebagai himpunan pasangan:

{(22012,johnson,c,22),(93831,glover,0f,24),…,
(39826,singleton,2b,31)}

dari 4-tupel.

- Basis data(*database*) merupakan kumpulan catatan yang dimanipulasi oleh komputer.
- Sistem manajemen basis *data(database management system*) merupakan program yang membantu pemakai mengakses informasi dalam basis data.
- Model basis data relasional yang ditemukan oleh E.F Codd pada tahun 1970, didasarkan pada konsep relasi n-er.

Istilah-istilah dalam basis data relasional

- Kolom-kolom dari *relasi n-er* disebut atribut(*attribute*)
- Daerah asal atribut adalah himpunan dimana semua anggota dalam atribut itu berada.
- Atribut tunggal atau kombinasi atribut bagi sebuah relasi merupakan kunci(*key*) jika nilai-nilai atribut secara unik mendefinisikan sebuah *n-tupel*
- Sistem manajemen basis data menjawab perintah-perintah(*queries*).

Operasi-operasi pada relasi dalam model basis data relasional

1. Seleksi

Operasi ini memilih *n*-tupel tertentu dari suatu relasi. Pilihan dibuat dengan persyaratan pada atribut.

Contoh1:

Relasi Pemain dari tabel 1.

PEMAIN [Posisi = c]

Akan memilih tupel : (22012,johnson,c,22) ,(84341,Cage,c,30)

2. Proyek

Operator proyek memilih kolom. Sebagai tambahan pengulangan akan dihilangkan.

Contoh 2.

PEMAIN[Nama,Posisi]

Akan memilih tupel : (Johnson,c), (Glover,of), (Batty,p),…,
(Singleton,2b)

3. Gabungan

Operasi seleksi dan proyek memanipulasi relasi tunggal; gabungan memanipulasi dua relasi. Operasi gabungan pada R_1 dan R_2 mengawali dengan menguji semua pasangan dari tupel, satu dari R_1 dan satu dari R_2. jika persyaratan gabungan dipenuhi, tupel-tupel akan dikombinasikan untuk membentuk tupel baru. Persyaratan gabungan menjelaskan hubungan antara atribut di R_1 dan atribut di R_2.

Contoh 3. (operasi gabungan tabel 1 dan 2)

Dengan persyaratan misal: Nomor ID = PID

Tabel 1 PEMAIN

Nomor ID	Nama	Posisi	Umur
22012	Johnson	c	22
93831	Glover	Of	24
58199	Batty	p	18
84341	Cage	c	30
01180	Homer	1b	37
26710	Score	p	22
61049	Johnson	Of	30
39826	Singleton	2b	31

Tabel 2. PENEMPATAN

PID	Tim
39826	Biru
26710	Merah
58199	Jingga
01180	Merah

Tabel 3. PeEMAIN [Nomor ID = PID] PENEMPATAN

Nomor ID	nama	Posisi	Umur	Tim
58199	Batty	p	18	Jingga
01180	Homer	1b	37	Merah
26710	Score	p	22	Merah
39826	singleton	2b	31	Biru

1. Nyatakan relasi yang diberikan oleh tabel berikut sebagai himpunan dari n-tupel

ID	Nama	Manajer
1089	Budi	Zamora
5624	Candra	Ivan
9843	Herman	Rudi
7610	Rian	Irwan

2. Nyatakan relasi yang diberikan oleh tabel berikut sebagai himpunan dari n-tupel

Dept.	Manajer
23	Zamora
10	Rudi
12	Irwan

3. Nyatakan relasi yang diberikan oleh tabel berikut sebagai himpunan dari n-tupel

Dept	No.Barang	banyaknya
23	23a	200
10	33c	45
23	500	56
25	11	150

4. Nyatakan relasi yang diberikan oleh tabel berikut sebagai himpunan dari n-tupel

Nama	No.Barang
United supplies	33c
ABC Limited	23a
ABC Limited	11
JCN Electronics	500

Untuk soal 5-8 tulislah serangkaian operasi relasi untuk menjawab permintaan. Juga berikanlah jawaban untuk permintaan tersebut.

5. Carilah nama-nama semua pekerja (jangan sertakan nama manajer)
6. Carilah semua nomor produk
7. Carilah semua produk yang dipasok oleh departemen 23
8. Carilah nomor produk dari produk-produk yang menangani paling sedikit 50 jenis barang.

Notasi Operasi Relasi

- **Notasi Operator Seleksi (σ)**

Tabel MHS

NIM	Nama	MK
135011	Adi	SIM
135011	Adi	OR
135015	Irma	SIM
135032	Rani	PTI

Contoh :

Operasi menampilkan pasangan terurut mahasiswa yang mengambil matkul SIM.

σMK= “SIM” (MHS)

- **Notasi Operator Proyek (Π)**

Tabel MHS

NIM	Nama	MK
135011	Adi	SIM
135011	Adi	OR
135015	Irma	SIM
135032	Rani	PTI

Contoh:

Notasi operasi proyeksi memilih kolom nama pada tabel MHS.

$$\Pi_{\text{Nama}} (\text{MHS})$$

- **Notasi operasi Join (τ)**

  Operasi join menggabungkan dua buah tabel menjadi satu bila kedua tabel mempunyai atribut yang sama.

  Contoh:

  Misalkan relasi MHS1 dan relasi MHS2 bila digabungkan maka notasi operasinya adalah:

  $\tau_{NIM,Nama}$ (MHS1,MHS2)

Tabel MHS1

NIM	Nama	JK
13598001	Hananto	L
13598002	Guntur	L
13598004	Heidi	W
13598006	Harman	L
13598007	Karim	L

Tabel MHS2

NIM	Nama	Matkul	Nilai
13598001	Hananto	SIM	A
13598001	Hananto	DBMS	B
13598004	Heidi	PTI	B
13598006	Harman	Statistik	C
13598006	Harman	OR	A
13598009	Yeni	Alin	B

Tabel Join

NIM	Nama	JK	Matkul	Nilai
13598001	Hananto	L	SIM	A
13598001	Hananto	L	DBMS	B
13598004	Heidi	W	PTI	B
13598006	Harman	L	Statistik	C
13598006	Harman	L	OR	A

FUNGSI

adalah bentuk khusus dari relasi. Definisi fungsi adalah sebagai berikut:

- Misalkan A dan B himpunan. Relasi biner f dari A ke B merupakan suatu fungsi jika untuk setiap elemen a di dalam A (disebut daerah asal/domain) terdapat satu elemen tunggal b di dalam B (disebut daerah hasil/range/codomain) sedemikian sehingga $(a,b) \in f$. Kita tulis f(a)=b. Jika f adalah fungsi dari A ke B, kita menuliskan $f : A \rightarrow B$ yang artinya f **memetakan** A ke B.

Contoh:

1. Diketahui A = {1, 2, 3} dan B = { u, v, w} maka

f = {(1,u), (2,v), (3,w)} adalah fungsi dari A ke B karena setiap anggota A memiliki satu kawan di B

Macam – macam fungsi

1. Fungsi **satu ke satu (one-to-one)**

   jika tidak ada dua elemen himpunan A yang memiliki bayangan yang sama, dengan kata lain jika a dan b adalah anggota himpunan A maka $f(a) \neq f(b)$ bilamana $a \neq b$.

Contoh:

F = {(1,w), (2,u), (3,v)} dari A = {1,2,3} ke B = {u,v,w,x} adalah fungsi satu ke satu

2. Fungsi **pada (onto)**

   jika setiap himpunan b merupakan bayangan dari satu atau lebih elemen himpunan A, dengan kata lain fungsi f adalah pada bila semua elemen B merupakan daerah hasil dari f.

Contoh:

F = {(1,w),(2,u),(3,v)} dari A={1,2,3} ke B={u,v,w} merupakan fungsi pada, karena semua elemen B termasuk ke dalam daerah hasil f.

Latihan:

Selidiki jenis fungsi atau bukan, fungsi satu-ke-satu atau bukan, fungsi pada atau bukan.

1. A={1,2,3,4} dan B={u,v,w} diberikan f={(1,u),(2,v),(3,w)}
2. A={1,2,3} dan B={u,v,w} diberikan f={(1,u),(1,v),(2,v),(3,w)}
3. A={1,2,3} dan B={u,v,w,x} diberikan f ={(1,w),(2,u),(3,v)}
4. A={1,2,3} dan B={u,v,w} diberikan f={(1,u),(2,u),(3,v)}
5. A={1,2,3} dan B={u,v,w} diberikan f={(1,u),(2,w),(3,v)}

QUANTIFIER (KUANTOR) dan Induksi matematika

PERTEMUAN KE-3

KUANTOR PERNYATAAN

Misalkan P(x) adalah pernyataan yang menyangkut variabel x dan D adalah sebuah himpunan, maka P adalah fungsi proposisi jika untuk setiap $x \in D$, berlaku P(x) adalah sebuah proposisi.

Contoh:

Misalkan P(x) merupakan pernyataan :

x adalah sebuah bilangan bulat genap.

Misalkan D = himpunan bilangan bulat positif

Maka fungsi proposisi P(x) dapat ditulis:

jika x = 1 maka proposisinya

1 adalah bilangan bulat genap. (F)

jika x = 2 maka proposisinya

2 adalah bilangan bulat genap. (T)

dst.

Untuk menyatakan kuantitas suatu objek proposisi digunakan notasi yang disebut kuantor

Macam-macam Kuantor

- Untuk setiap x, P(x)
  disebut kuantor universal
  Simbol: $\forall$
- Untuk beberapa x, P(x)
  disebut kuantor eksistensial
  Simbol: $\exists$

Contoh:

Misalkan x himpunan warga negara Indonesia,

P predikat membayar pajak, R predikat membeli Ms Word,

Maka:

1. $\forall x, P(x)$

   artinya: semua warga negara membayar pajak

2. $\exists x, R(x), P(x)$

   artinya: ada beberapa warga negara membeli Ms word membayar pajak

3. $\forall x, R(x) \rightarrow P(x)$

   artinya: semua warga negara jika membeli ms word maka membayar pajak

4. $\exists x, R(x) \wedge \overline{P(x)}$

   artinya: ada warga negara membeli ms word dan tidak membayar pajak

Negasi Kuantor

$\sim\forall x = \exists x$

$\sim\exists x = \forall x$

Sehingga:

$\sim(\forall x, P(x)) = \exists x, P(x)$

$\sim(\exists x, P(x)) = \forall x, P(x)$

$\sim(\forall x, P(x) \rightarrow Q(x)) = \exists x, (P(x) \rightarrow Q(x))$

$= \exists x, P(x) \wedge Q(x)$

1. Tentukan validitas pernyataan di bawah ini bila domain pembicaraannya himpunan bilangan real

(a) $\forall x, \forall y, P(x^2 < y+1)$

$\forall x, \exists y, P(x^2 < y+1)$

$\exists x, \forall y, P(x^2 < y+1)$

$\exists x, \exists y, P(x^2 < y+1)$

(b) $\forall x, \forall y, P[(x < y) \rightarrow (x^2 < y^2)]$

$\forall x, \exists y, P[(x < y) \rightarrow (x^2 < y^2)]$

$\exists x, \forall y, P[(x < y) \rightarrow (x^2 < y^2)]$

$\exists x, \exists y, P[(x < y) \rightarrow (x^2 < y^2)]$

2. Negasikan setiap pernyataan di bawah ini:

(a) $\forall x, P(x) \wedge \exists y, Q(y)$

(b) $\exists x, P(x) \vee \forall y, Q(y)$

(c) $\forall x, \exists y, [P(x) \vee Q(y)]$

Perhatikan argumen matematik berikut ini:

1. P(n) : Jumlah bilangan bulat positif dari sampai 1 sampai n adalah $n(n + 1)/2$

   misal untuk n = 5 adalah $5(5+1)/2=15$

   terlihat: $1+2+3+4+5=15$

2. P(n) : Jumlah dari n buah bilangan ganjil positif pertama adalah n^2

   misal untuk n = 3 adalah $3^2 = 9$

   terlihat : $1 + 3 + 5 = 9$

Induksi matematik merupakan teknik pembuktian yang baku dalam matematik, khususnya menyangkut bilangan bulat positif.

Prinsip Induksi Sederhana

Misalkan P(n) adalah pernyataan perihal bilangan bulat positif dan kita ingin membuktikan bahwa p(n) benar untuk semua bilangan bulat positif n. untuk membuktikan pernyataan ini, kita hanya perlu membuktikan pernyataan ini, kita hanya perlu menunjukkan bahwa:

1. p(1) benar, dan

2. Untuk semua bilangan bulat positif n ≥ 1,

jika p(n) benar maka p(n+1) juga benar.

- Langkah 1 dinamakan basis induksi
- Langkah 2 dinamakan langkah induksi
- Asumsi jika p(n) benar dinamakan hipotesis induksi.

Contoh:

Tunjukkan bahwa

untuk $n \geq 1$, $1+2+\ldots+n = n(n+1)/2$

Bukti:

Basis induksi. Untuk n=1 kita peroleh 1 = 1(1+1)/2, ini jelas benar sebab

1 = 1 (1+1)/2

= 1 (2)/2

= 2/2

= 1

Langkah induksi. Andaikan untuk $n \geq 1$ pernyataan

1+2+3+…+n = n(n+1)/2 adalah benar (hipotesis induksi)

Kita harus menunjukkan bahwa:

1+2+3+…+n + (n+1) = (n+1)[(n+1)]/2 juga benar

Untuk membuktikan ini tunjukkan bahwa:

$1+2+3+\ldots+ n + (n+1) = (1+2+3+\ldots+n)+(n+1)$

$= [n(n+1)/2]+(n+1)$

$= [(n2+n)/2]+(n+1)$

$= [(n2+n)/2]+[(2n+2)/2]$

$= (n2+3n+2)/2$

$= (n+1)[(n+1)+1]/2$

Karena langkah basis dan langkah induksi keduanya telah dibuktikan benar, maka untuk semua bilangan bulat positif n, terbukti bahwa:

$1+2+3+\ldots+n = n(n+1)/2$

Latihan

Buktikan dengan induksi matematik

1. Jumlah n buah bilangan ganjil positif pertama adalah n2
2. Untuk semua $n \geq 1$ maka n3 + 2n adalah kelipatan 3
3. $n! \geq 2n-1$ untuk n= 1,2,3,…
4. $1 \cdot 2 + 2 \cdot 3 + 3 \cdot 4 + \ldots + n(n+1) = n(n+1)(n+2)/3$

Pertemuan 4

Kombinatorial

Kaidah Dasar menghitung

Dalam kombinatorial ada dua kaidah dasar yang digunakan untuk menghitung, yaitu kaidah penjumlahan (rule of sum) dan kaidah perkalian (rule of product)

1. Kaidah Penjumlahan (rule of sum)

Bila percobaan 1 mempunyai m hasil percobaan yang mungkin terjadi(atau memiliki sebanyak m kemungkinan jawaban) dan percobaan 2 mempunyai n hasil percobaan yang mungkin (atau memiliki sebanyak n kemungkinan jawaban), maka bila hanya salah satu dari dua percobaan itu saja yang dilakukan (percobaan 1 “atau” percobaan 2), maka terdapat m+n hasil jawaban (atau memiliki m +n kemungkinan jawaban)

Contoh1:

Seorang mahasiswa akan memilih satu mata kuliah yang ditawarkan pagi dan sore. Untuk pagi ada 7 matakuliah dan sore ada 5 matakuliah yang ditawarkan. Maka mahasiswa tadi mempunyai 7+5 pilihan untuk memilih satu matakuliah tersebut.

2.Kaidah Perkalian (rule of product)

Bila percobaan 1 mempunyai m hasil percobaan yang mungkin terjadi(atau memiliki sebanyak m kemungkinan jawaban) dan percobaan 2 mempunyai n hasil percobaan yang mungkin (atau memiliki sebanyak n kemungkinan jawaban), maka bila kedua percobaan1 “dan” percobaan 2 dilakukan , maka terdapat mxn hasil jawaban (atau memiliki m xn kemungkinan jawaban

Perluasan kaidah

Kaidah penjumlahan dan kaidah perkalian di atas dapat diperluas hingga mengandung lebih dari 2 percobaan. Jika n buah percobaan masing-masing mempunyai p1, p2,…,pn hasil percobaan yang mungkin terjadi yang dalam hal ini setiap pi tidak bergantung pada pilihan sebelumnya, maka jumlah hasil percobaan yang mungkin terjadi adalah:

a. p1 x p2 x … x pn untuk kaidah perkalian

b. p1 + p2 + … + pn untuk kaidah penjumlahan

Contoh:

Jika harus menyusun jadwal tiga ujian ke dalam periode lima hari tanpa ada pembatasan mengenai berapa kali dibolehkan ujian dalam setiap harinya, berapakah kemungkinan jadwal yang dapat dibuat?

Jawab:

Karena penyusunan jadwal tiga ujian dan tanpa ada pembatasan dalam periode lima hari, maka jumlah kombinasi jadwal yang mungkin dibuat:

5.5.5 = 125

Prinsip Inklusi-Eksklusi

Adalah cara penghitungan dengan menggunakan prinsip perhitungan himpunan.

Contoh: Berapa banyak jumlah byte yang dimulai dengan ‘11’ atau diakhiri dengan ‘11’ ?

Petunjuk Penyelesaian:

Misalkan A = himpunan byte yang dimulai dengan ‘11’

B = himpunan byte yang diakhiri dengan ‘11’

$A \cap B$ = himpunan byte yang berawal dan berakhir dengan ‘11’

Maka

$A \cup B$ = himpunan byte yang berawal dengan ‘11’ atau berakhir dengan ‘11’

Dengan Rumus:

$$A \cup B = A + B - A \cap B$$

1. Pengertian Permutasi

   suatu susunan data dengan memperhatikan /membedakan urutan. Permutasi merupakan bentuk khusus aplikasi aturan perkalian.

Rumus:

1. Permutasi dari n objek seluruhnya:

   $_nP_n = n! = n.(n-1).(n-2)\ldots 2.1$

   $= n.(n-1)!$

2. Permutasi sebanyak r dari n objek:

$$_nP_r = \frac{n!}{(n-r)!}$$

3. Permutasi keliling (circular permutation)

   Sejumlah n objek yang berbeda dapat disusun secara teratur dalam sebuah lingkaran dalam (n-1)! cara

4. Permutasi dari n objek yang tidak seluruhnya dapat dibedakan:

$$\binom{n}{n_1,n_2,n_3,\ldots,n_k} = \frac{n!}{n_1!n_2!n_3!\ldots n_k!}$$

Contoh soal:

1. Ada berapa cara3 buku dapat diurutkan ?

   3! = 3.2.1 = 6 cara

2. Ada berapa cara 2 dari 4 buku dapat disusun ?

$${}_4P_2 = \frac{4!}{(4-2)!} = \frac{4!}{2!} = \frac{4.3.2.1}{2.1} = 4.3 = 12 \text{ cara}$$

3. 4 orang mahasiswa melakukan diskusi dengan membentuk sebuah lingkaran, ada berapa cara urutan dari 4 orang tadi?

   Jawab : 4! = 4.3.2.1 = 24 cara

4. Dalam berapa cara kata “diskrit” dapat diurutkan?

   jawab:

$$\frac{7!}{1!2!1!1!1!1!} = \frac{7.6.5.4.3.2!}{2!} = 2520 \text{ cara}$$

2. Kombinasi

Suatu susunan data tanpa memperhatikan urutannya.

$$ {}_nC_r = \frac{n!}{r!(n-r)!} $$

Contoh:

1. Ada berapa cara akan dipilih 2 orang dari 4 orang siswa?

Jawab:

$$ {}_4C_2 = \frac{4!}{2!(4-2)!} = \frac{4.3.2!}{2!\,2!} = \frac{12}{2} = 6 $$

Permutasi dan Kombinasi Bentuk Umum

Adalah menyusun obyek di mana tidak semua obyek bisa dibedakan (sama). Untuk rumus permutasi dan kombinasinya sama yaitu:

$$P(n;n1,n2,\ldots,nk) = C(n;n1,n2,\ldots,nk) = \frac{n!}{n1!.n2!\ldots nk!}$$

Contoh:

Berapa banyak string yang dapat dibentuk dengan menggunakan huruf-huruf dari kata MISSISSIPPI?

Jawab: S = {M,I,S,S,I,S,S,I,P,P,I}

huruf M = 1 buah n1 huruf S = 4 buah n3

huruf I = 4 buah n2 huruf P = 2 buah n4

n = 1 + 4 + 4 + 2 = 11 buah = jumlah elemen himpunan S

Cara 1

$$\text{Jumlah string} = P(11; 1,4,4,2) = \frac{11!}{1!4!4!2!} = 34650 \text{ buah}$$

Cara 2

Jumlah string = C(11,1).C(10,4).C(6,4).C(2,2)

$$= \frac{11!}{1!10!} \cdot \frac{10!}{4!6!} \cdot \frac{6!}{4!2!} \cdot \frac{2!}{2!0!}$$

$$= \frac{11!}{1!4!4!2!} = 34650 \text{ buah}$$

Kombinasi dengan Pengulangan

Jumlah kombinasi yang membolehkan adanya pengulangan elemen, yaitu dari n buah obyek kita akan mengambil r buah obyek, dengan pengulangan diperbolehkan.

$C(n+r-1, r) = C(n+r-1, n-1)$

Contoh:

Pada persamaan x1+x2+x3+x4 = 12, xi adalah bilangan bulat ≥ 0. Berapa jumlah kemungkinan solusinya?

Jawab:

misalkan 12 sebagai bola dan xi = 4 sebagai kotak (n=4 dan r=12), sehingga banyak kemungkinan yang bisa terjadi. Namun seluruhnya ada C(4+12-1,12) = C(15,12) = 455 buah kemungkinan solusi

Latihan:

1. Empat buah ujian dilakukan dalam periode enam hari. Berapa banyak pengaturan jadwal yang dapat dilakukan sehingga tidak ada dua ujian atau lebih yang dilakukan pada hari yang sama.
2. Berapa banyak string yang dapat dibentuk yang terdiri dari 4 huruf berbeda dan 3 angka yang berbeda pula?
3. Berapakah jumlah kemungkinan membentuk 3 angka dari 5 angka berikut: 1,2,3,4,5 jika:
   i. tidak boleh ada pengulangan angka
   ii. Boleh ada pengulangan angka.
4. String biner yang panjangnya 32 bit disusun oleh digit 1 atau 0. Berapa banyak string biner yang tepat berisi 7 buah bit 1?

5. Sebuah karakter dalam sistim ASCII berukuran 1 byte atau 8 bit (1 atau 0).
   a. Berapa banyak pola bit yang terbentuk? (atau berapa banyak karakter yang dapat dipresentasikan?)
   b. Berapa banyak pola bit yang mempunyai 3 bit 1?
   c. berapa banyak pola bit yang mempunyai bit 1 sejumlah genap?
6. Suatu panitia akan dibentuk dengan jumlah 5 orang. Berapa carakah pembentukan panitia tersebut dapat dilakukan jika calon anggota terdiri dari 4 orang pria dan 3 orang wanita dan panitia harus
   a. terbentuk tanpa persyaratan lain
   b. terdiri 3 pria dan 2 wanita
   c. terdiri 2 pria dan 3 wanita

Pertemuan 5

LOGIKA PROPOSISI

Pernyataan

- Logika proposisi berisi pernyataan-pernyataan (tunggal/majemuk)
- Pernyataan : kalimat deklarasi yang dinyatakan dengan huruf-huruf kecil.
- Pernyataan mempunyai sifat dasar yaitu benar atau salah tetapi tidak keduanya

Contoh:

1. Bilangan biner digunakan dalam sistem digital
2. Sistem analog lebih akurat daripada sistem digital
3. Pentium IV lebih bagus kinerjanya dan lebih mahal harganya daripada pentium III

Kalimat yang tidak termasuk pernyataan: *kalimat perintah, pertanyaan, keheranan, harapan, kalimat … walaupun …*

Pernyataan Majemuk

- **Negasi**

  Sebuah pernyataan yang meniadakan pernyataan yang ada, dapat dibentuk dengan menulis '*adalah salah bahwa*…' atau dengan menyisipkan kata '*tidak*'

  notasi: ~p , p'

  Contoh:

  p = keyboard merupakan output device

  ~ p = adalah salah bahwa keyboard merupakan output device

- Kebenaran sebuah negasi adalah lawan dari kebenaran pernyataannya.
- Tabel kebenaran negasi:

p	~p
+	–
–	+

❑ **Konjungsi**

Pernyataan gabungan dari dua pernyataan dengan kata hubung ‘dan’

Notasi: $p \wedge q$, pq , $p \times q$

Contoh:

p = sistem analog adalah suatu sistem dimana tanda fisik/kuantitas, dapat berbeda-beda secara terus menerus melebihi jarak tertentu.(*benar*)

q = sistem digital adalah suatu sistem dimana tanda fisik/kuantitas, hanya dapat mengasumsikan nilai yang berlainan. (*benar*)

r = sistem bilangan desimal adalah sistem bilangan yang digunakan dalam sistem digital. (*salah*)

Maka:

$p \wedge q$ adalah konjungsi yang benar

$q \wedge r$ adalah konjungsi yang salah

Disjungsi

Adalah pernyataan gabungan dari dua pernyataan dengan kata hubung 'atau'

Notasi: $p \vee q$, $p + q$

p	q	$p \wedge q$
+	+	+
+	−	−
−	+	−
−	−	−

p	q	$p \vee q$
+	+	+
+	−	+
−	+	+
−	−	−

Contoh:

p = keyboard adalah input device (*benar*)

q = harddisk adalah alat penentu kecepatan komputer (*salah*)

r = processor adalah otak dari komputer (*benar*)

Maka:

$p \vee q$ adalah disjungsi yang benar

$p \vee r$ adalah disjungsi yang benar

❑ Jointdenial(Not OR /NOR)

Adalah pernyataan gabungan yang dihasilkan dari menegasikan disjungsi.

Notasi: $p \downarrow q$, $\sim(p \vee q)$

p	q	$p \vee q$	$p \downarrow q$
+	+	+	–
+	–	+	–
–	+	+	–
–	–	–	+

Not And (NAND)

Adalah pernyataan gabungan yang dihasilkan dari menegasikan konjungsi.

Notasi: $\sim(p \wedge q)$, $p \uparrow q$

p	q	$(p \wedge q)$	$p \uparrow q$
+	+	+	–
+	–	–	+
–	+	–	+
–	–	–	+

❑ Exlusive OR(EXOR)

Adalah pernyataan gabungan di mana salah satu p atau q (tidak keduanya) adalah benar

Notasi : $p \oplus q$

p	q	$p \oplus q$
+	+	–
+	–	+
–	+	+
–	–	–

Exlusive NOR(EXNOR)

Adalah pernyataan gabungan dimana nilai kebenarannya benar bila kedua pernyataannya benar atau salah.

Notasi : $\sim(p \oplus q)$

p	q	$\sim(p \oplus q)$
+	+	+
+	–	–
–	+	–
–	–	+

KESETARAAN LOGIS

Dua buah pernyataan yang berbeda dikatakan setara/equivalen bila nilai kebenarannya sama

Contoh:

1. Tidak benar bahwa aljabar linier adalah alat matematika dasar untuk disain logika.(benar)
2. Aljabar boole adalah alat matematika dasar untuk disain logika.(benar)

Contoh:

Selidiki apakah kedua proposisi di bawah ini setara:

1. Tidak benar bahwa sistem bilangan biner dipergunakan dalam sistem digital atau sistem digital hanya dapat mengasumsikan nilai yang berlainan.
2. Sistem bilangan biner tidak dipergunakan dalam sistem digital dan tidak benar bahwa sistem digital hanya dapat mengasumsikan nilai yang berlaianan.

(hint: buktikan : $\sim(p \vee q) \equiv \sim p \wedge \sim q$)

Aljabar Proposisi

Aljabar proposisi adalah hukum-hukum aljabar yang dapat digunakan dalam proposisi.

Hukum-hukum tersebut adalah:

1. Idempoten

$p \vee p \equiv p$

$q \wedge q \equiv p$

2. Assosiatif

$(p \vee q) \vee r \equiv p \vee (q \vee r)$

$(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$

3. Distributif

$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$

$p \wedge (q \wedge r) \equiv (p \wedge q) \vee (p \wedge r)$

4. Komutatif

$p \vee q \equiv q \vee p$

$p \wedge q \equiv q \wedge p$

5. Identitas

$p \vee f \equiv p$

$p \vee t \equiv t$

$p \wedge f \equiv f$

$p \wedge t \equiv p$

6. Involution

$\sim\sim p \equiv p$

$\sim q$

7. Komplemen

$p \vee \sim p \equiv t$

$p \wedge \sim p \equiv f$

$\sim t \equiv f$

$\sim f \equiv t$

8. De Morgan's

$\sim(p \wedge q) \equiv \sim p \quad \vee$

$\sim(p \vee q) \equiv \sim p \wedge \sim q$

Contoh pemakaian hukum aljabar proposisi

Sederhanakan proposisi berikut ini:

1. $p \wedge (p \vee q)$

$p \wedge (p \vee q) \equiv (p \vee f) \wedge (p \vee q)$ …(*hk.identitas*)

$\equiv p \vee (f \wedge q)$ …(*hk.distribusi*)

$\equiv p \vee f$ …(*hk.identitas*)

$\equiv p$ …(*hk.identitas*)

2. Sederhanakan proposisi: $p \vee (p \wedge q)$

IMPLIKASI DAN BIIMPLIKASI

Implikasi

Jika memakai Ms Word **maka** *windows adalah sistem operasinya*

Artinya: Ms word tidak dapat digunakan tanpa windows tetapi windows dapat digunakan tanpa Ms word

Contoh pernyataan di atas disebut pernyataan beryarat (*conditional statement*)

Notasi: $p \rightarrow q$

Tabel kebenaran impilkasi

p	q	$p \rightarrow q$
+	+	+
+	–	–
–	+	+
–	–	+

Contoh: Misalkan pernyataan p adalah benar, q adalah salah dan r adalah benar, tentukan kebenaran proposisi berikut:

$$(p \vee q) \rightarrow \sim r$$

Variasi Implikasi

Jika implikasi: $p \rightarrow q$

Maka:	Konversnya	: $q \rightarrow p$
	Inversnya	: $\sim p \rightarrow \sim q$
	Kontrapositipnya	: $\sim q \rightarrow \sim p$

Contoh:

Tentukan konvers,invers, dan kontrapositif dari proposisi berikut:

Jika Ms Word aplikatifnya maka windows sistem operasinya

- Tabel kebenaran variasi implikasi:

p	q	~p	~q	$p \rightarrow q$	$\sim q \rightarrow \sim p$	$q \rightarrow p$	$\sim p \rightarrow \sim q$
+	+	–	–	+	+	+	+
+	–	–	+	–	-	+	+
–	+	+	–	+	+	–	-
–	–	+	+	+	+	+	+

setara setara

Kesimpulan:

Proposisi yang saling kontrapositif mempunyai nilai kebenaran yang sama(equivalen)

Contoh:

Buktikan bahwa:

Jika x^2 *bilangan genap*, maka *x juga bilangan genap*

Jawab:

Kontrapositif dari implikasi di atas adalah:

Jika *x* bukan *bilangan genap*, maka x^2 *juga bukan bilangan genap*

Setiap bilangan bulat bukan genap adalah ganjil, sehingga jika x ganjil ditulis sebagai

$x = 2k + 1$ (k bil. Bulat) akibatnya:

$$X^2 = (2k + 1)^2$$
$$= 4k^2 + 4k + 1$$
$$= 2(2k^2 + 2k) + 1$$

Karena kontrapositifnya benar akibatnya implikasinya juga benar.

Biimplikasi

Contoh pernyataan biimplikasi:

Ms word jika dan hanya jika *ingin membuat dokumen*

Notasi: $p \leftrightarrow q$

Kebenaran biimplikasi:

p	q	$p \leftrightarrow q$
+	+	+
+	–	–
–	+	–
–	–	+

Argumentasi

Argumentasi adalah kumpulan pernyataan – pernyataan atau premis-premis atau dasar pendapat serta kesimpulan(konklusi)

Notasi:

P(p,q,…)
Q(p,q,…)
⋮
∴ C(p,q,…)

P,Q,… masing-masing disebut premis
{P,Q,..} bersama-sama disebut hipotesa
C adalah kesimpulan/konklusi

Contoh:

Jika biner maka disain logika

Jika disain logika maka digital

∴ **Jika biner maka digital**

Kebenaran/validitas Argumen

Nilai kebenaran argument tergantung dari nilai kebenaran masing-masing premis dan kesimpulannya.

Suatu argumen dikatakan benar bila masing-masing premisnya benar dan kesimpulannya juga benar.

Contoh 1:

Jika biner maka disain logika

Jika disain logika maka digital

$\therefore$ Jika biner maka digital

Argumen tersebut dapat ditulis dengan notasi:

$p \rightarrow q$ disebut premis 1

$q \rightarrow r$ disebut premis 2

$\overline{\therefore p \rightarrow r}$ disebut konklusi

Perhatikan Tabel kebenaran

p	q	r	p→q	q → p	p →r
+	+	+	+	+	+
+	+	–	+	–	–
+	–	+	–	+	+
+	–	–	–	+	–
–	+	+	+	-	+
–	+	–	+	–	+
–	–	+	+	+	+
–	–	–	+	+	+

Semua premis dan konklusi benar sehingga argumentasi di atas **valid**.

Bentuk-bentuk dasar menarik kesimpulan

1. Conjunction

p

q

$\therefore p \wedge q$

2. Addition

p

$\therefore p \vee q$

3. Construction Dilemma

$(p \rightarrow q) \wedge (r \rightarrow s)$

$p \vee r$

$\therefore q \vee s$

4. Modus Ponens

$p \rightarrow q$

p

$\therefore q$

5. Modus Tollens

$p \rightarrow q$

$\sim q$

$\therefore \sim p$

6. Hypothetical syllogism

$p \rightarrow q$

$q \rightarrow r$

$\therefore p \rightarrow r$

7. Simplification

$p \wedge q$

$\therefore p$

8. Disjunctive syllogism

$p \vee q$

$\sim p$

$\therefore q$

9. Destructive Dilemma

$$(p \rightarrow q) \wedge (r \rightarrow s)$$

$$\sim q \vee \sim s$$

$$\therefore \sim p \vee \sim r$$

10. Absorption

$$p \rightarrow q$$

$$\therefore p \rightarrow (p \wedge q)$$

Contoh pemanfaatan:

Buatlah kesimpulan dari argumen di bawah ini sehingga argumen tersebut valid

1. Jika hasilnya akurat maka sistemnya digital
2. Jika sistem digital maka menggunakan bil. Biner
3. Hasilnya akurat

$\therefore$?

Jawab:

Premis 1 : $p \rightarrow q$
Premis 2 : $q \rightarrow r$
Premis 3 : p
$\therefore$?

Dengan hypothetical syllogism

$$\frac{\begin{array}{l} p \rightarrow q \\ q \rightarrow r \end{array}}{\therefore\ p \rightarrow r}$$

Sehingga argumentasi dapat ditulis kembali:

$$\frac{\begin{array}{l} p \rightarrow r \\ p \end{array}}{\therefore\ ?}$$

Dengan Modus Ponen, konklusinya adalah r

$$\frac{\begin{array}{l} p \rightarrow r \\ p \end{array}}{\therefore\ r}$$

Adalah valid

p	q	r	$p \rightarrow q$	$q \rightarrow r$
+	+	+	+	+
+	+	–	+	–
+	–	+	–	+
+	–	–	–	+
–	+	+	+	+
–	+	–	+	–
–	–	+	+	+
–	–	–	+	+
3		∴	1	2

Pertemuan 6

Pendahuluan Aljabar Boolean

1. Definisi Aljabar Boolean

Aljabar Boolean (B) merupakan aljabar yang terdiri atas suatu himpunan dengan operasi jumlah/disjungsi, kali/konjungsi dan komplemen/negasi serta elemen 0 dan 1 ditulis sebagai <B,+, . , ‘ ,0,1> yang memenuhi sifat-sifat:

1. Hukum identitas
   (i) a + 0 = a
   (ii) a . 1 = a
2. Hukum idempoten
   (i) a+a = a
   (ii) a.a = a
3. Hukum komplemen
   (i) a+a’ = 1
   (ii) a.a’ = 0
4. hukum dominasi
   (i) a.0 = 0
   (ii) a+1 = 1
5. Hukum involusi
   (i) (a’)’ = a
6. Hukum penyerapan
   (i) a+(a.b) = a
   (ii) a.(a+b) = a

7. Hukum komutatif
   (i) a+b = b+a
   (ii) a.b = b.a

8. Hukum asosiatif
   (i) a+(b+c) = (a+b)+c
   (ii) a.(b.c) = (a.b).c

9. Hukum distributif
   (i) a+(b.c) = (a+b).(a+c)
   (ii) a.(b+c) = (a.b)+(a.c)

10. Hukum De Morgan
   (i) (a+b)' = a'.b'
   (ii) (ab)' = a'+b'

11. Hukum 0/1
   (i) 0' = 1
   (ii) 1' = 0

Catatan:

Untuk penyederhanaan penulisan, penulisan a.b sebagai ab

Perbedaan antara aljabar Boolean dan aljabar biasa untuk aritmatika bilangan riil :

1. Hukum distributif + dan . Seperti a+(b.c) = (a+b) . (a+c) benar untuk aljabar Boolean tetapi tidak benar untuk aljabar biasa.
2. Aljabar Boolean tidak memiliki kebalikan perkalian (multiplicative inverse) dan penjumlahan, sehingga tidak ada operasi pembagian dan pengurangan.
3. Sifat no 2 mendefinisikan operator yang dinamakan komplemen yang tidak tersedia pada aljabar biasa.
4. Aljabar biasa memperlakukan bilangan riil dengan himpunan yang tidak berhingga. Aljabar Boolean memperlakukan himpunan elemen B yang sampai sekarang belum didefinisikan, tetapi pada aljabar Boolean dua nilai yaitu nilai 0 dan 1

Hal lain yang penting adalah membedakan elemen himpunan dan peubah (variabel) pada sistem aljabar.

	elemen himpunan	peubah
Aljabar biasa	bil riil	a, b, c
Aljabar Boolean	bil riil	x, y, z

Suatu aljabar Boolean harus memenuhi 3 syarat :

1. Elemen himpunan B
2. Kaidah/aturan operasi untuk dua operator biner
3. Himpunan B, bersama-sama dengan dua operator tersebut,memenuhi postulat Huntington.

2. Aljabar Boolean dua-nilai

Aljabar Boolean dua-nilai (two-valued Boolean algebra) didefinisikan pada sebuah himpunan dengan dua buah elemen, B = {0,1}, dengan kaidah untuk operator + dan .

Perhatikan:

a	b	a.b
0	0	0
0	1	0
1	0	0
1	1	1

a	b	a+b
0	0	0
0	1	1
1	0	1
1	1	1

a	A’
0	1
1	0

Priinsip Dualitas

Misalkan S adalah kesamaan tentang aljabar Boolean yang melibatkan operasi +, . , dan komplemen, maka jika pernyataan S* diperoleh dengan cara mengganti:

. Dengan +

+ dengan .

0 dengan 1

1 dengan 0

Maka kesamaan S* juga benar. S* disebut sebagai dual dari S.

Contoh :

Tentukan dual dari (i) a.(b+c) = (a.b)+(a.c)

(ii) a+0

Jawab:

(i) a+(b.c) = (a+b).(a+c)

(ii) a+0 = a mempunyai dual a.1 = a

Beberapa bukti dari sifat-sifat aljabar Boolean:

(2i) a+a = (a+a) (1) (identitas)

= (a+a) (a+a') (komplemen)

= a+ (a.a') (distributif)

= a+0 (komplemen)

= a (identitas)

Soal :

Buktikan bahwa untuk sembarang elemen a dan b dari aljabar Boolean:

(i) a+a'b =a+b

(ii) a(a'+b) = ab

(iii) a+1 = 1

(iv) (ab)' = a' + b'

Jawab:

(i) a+a'b = (a+ab) + a'b penyerapan

= a+(ab+a'b) Asosiatif

= a(a+a')b distributif

= a+1.b Komplemen

= a+b Identitas

Fungsi Boolean

Pada aljabar Boolean dua-nilai B{0,1}. Peubah (variable) x disebut peubah boolean atau peubah biner jika nilainya hanya dari B. Fungsi Boolean (disebut juaga fungsi biner) adalah ekspresi yang dibentuk dari peubah biner, dua buah operator + dan . , operator uner ($\bar{}$)/ ′, tanda kurung dan tanda sama dengan =. Setiap peubah boolen, termasuk komplemennya disebut **literal**.

Contoh-contoh fungsi Boolean:

1. f(x) = x
2. f(x,y) = x′y + xy′ + y′
3. f(x,y) = x′ y′
4. f(x,y) = (x+y)′
5. f(x,y,z) = xyz′

Pada contoh 5 terdiri dari 3 literal yaitu x, y dan z′. Fungsi tersebut akan bernilai 1 jika x = 1, y = 1, dan z = 0 sebab

F(1,1,0) = 1.1.0′ = (1.1).1 = 1.1 = 1

Dan bernilai 0 untuk yang lainnya.

Selain secara aljabar fungsi Boolean bisa dinyatakan dengan tabel kebenaran (truth table)

Contoh:

dari contoh 5 f(x,y,z) = xyz′ nyatakan f dalam tabel kebenaran.

x	y	z	z′	f(x,y,z)
0	0	0	1	0
0	0	1	0	0
0	1	0	1	0
0	1	1	0	0
1	0	0	1	0
1	0	1	0	0
1	1	0	1	1
1	1	1	0	0

Fungsi Komplemen

Fungsi komplemen dari suatu fungsi f yaitu f′ dapat dicari dengan menukarkan nilai 0 menjadi 1 dan nilai 1 menjadi 0 serta menukar + menjadi . Dan . menjadi +.

Ada dua cara yang dapat digunakan untuk membentuk fungsi komplemen:

1. Menggunakan hukum De Morgan

$(x_1+x_2+\ldots+x_n)' = x_1' x_2' \ldots x_n'$

dan dualnya:

$(x_1.x_2. \ldots . x_n)' = x_1'+ x_2'+ \ldots + x_n'$

Contoh: misal f (x,y,z) = x(y′z′+ yz) maka

$$f'(x,y,z) = x'+ (y'z'+yz)'$$
$$= x'+ (y'z')' (yz)'$$
$$= x'+(y +z) (y'+z')$$

2. Menggunakan prinsip dualitas.

Cari dual dari f, lalu komplemenkan setiap literalnya.

Contoh:

misal f (x,y,z) = x(y′z′+ yz) maka

Dual dari f: x + (y′+ z′) (y + z)

Komplemenkan tiap literalnya: x′+(y+z) (y′+z′) = f′

Jadi, f′(x,y,z) = x′+(y+z) (y′+z′)

Cari komplemen dari

1. f(x,y,z) = x′(yz′+y′z)
2. f(x) = x
3. f(x,y) = x′y + xy′ + y′
4. f(x,y) = x′ y′
5. f(x,y) = (x+y)′
6. f(x,y,z) = xyz′

Pertemuan 9

Fungsi Boolean, Bentuk Kanonik
Bentuk Baku dan aplikasinya

Bentuk fungsi Boolean mungkin mempunyai ekspresi aljabar yang berbeda akan tetapi sebenarnya nilai fungsinya sama.

Contoh:

$f(x,y) = x'y'$ dan $h(x,y) = (x + y)'$

$f(x,y,z) = x'y'z + xy'z' + xyz$ dan

$g(x,y,z) = (x+y+z)(x+y'+z)(x+y'+z')(x'+y+z')(x'+y'+z)$

Adalah dua buah fungsi yang sama. Fungsi yang pertama f muncul dalam bentuk penjumlahan dari perkalian, sedangkan fungsi yang kedua g muncul dalam bentuk perkalian dari hasil jumlah. Perhatikan juga bahwa setiap suku (term) mengandung literal yang lengkap x,y,dan z.

Bentuk Kanonik

Adalah fungsi Boolean yang dinyatakan sebagai jumlah dari hasil kali,hasil kali dari jumlah dengan setiap suku mengandung literal yang lengkap.

Ada dua macam bentuk kanonik:

1. Minterm atau sum-of-product (SOP)
2. Maxterm atau product-of-sum(POS)

		Minterm		Maxterm	
x	y	suku	lambang	suku	lambang
0	0	x′y′	m_0	x + y	M_0
0	1	x′y	m_1	x + y′	M_1
1	0	xy′	m_2	x′ + y	M_2
1	1	xy	m_3	x′ + y′	M_3

			Minterm		Maxterm	
x	y	z	suku	lambang	Suku	lambang
0	0	0	x′y′z′	m_0	x + y + z	M_0
0	0	1	x′y′z	m_1	x + y + z′	M_1
0	1	0	x’yz′	m_2	x + y′ + z	M_2
0	1	1	x′yz	m_3	x + y′ + z′	M_3
1	0	0	xy′z′	m_4	x′ + y + z	M_4
1	0	1	xy′z	m_5	x′+ y + z′	M_5
1	1	0	xyz′	m_6	x′ + y′ + z	M_6
1	1	1	xyz	m_7	x′ + y′+ z′	M_7

Perbedaan minterm dan maxterm adalah:

Untuk membentuk minterm perhatikan kombinasi peubah yang menghasilkan nilai 1. Kombinasi 001, 100 dan 111 dituliskan x′ y′ z, xy′ z′ dan xyz.

Untuk membentuk maxterm perhatikan kombinasi peubah yang menghasilkan nilai 0. kombinasi 000, 010, 011, 101 dan110 dituliskan (x+y+z), (x+y′ +z), (x+y′ +z′),(x′ +y+z′) dan (x′ +y′ +z)

Notasi Σ dan Π berguna untuk mempersingkat penulisan ekspresi dalam bentuk SOP dan POS.

x	y	z	f(x,y,z)
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Dari tabel diatas nyatakan fungsi tersebut dalam bentuk kanonik SOP dan POS!

Jawab:

1. SOP: perhatikan kombinasi peubah yang menghasilkan nilai 1

   $f(x,y,z) = x' y' z + xy' z' + xyz$

   dalam bentuk lain

   $f(x,y,z) = m_1 + m_4 + m_7 = \Sigma(1,4,7)$

2. POS: perhatikan kombinasi peubah yang menghasilkan nilai 0

   $f(x,y,z) = (x+y+z)(x+y' +z)(x+y' +z')(x' +y+z')(x' +y' +z)$

   dalam bentuk lain

   $f(x,y,z) = M_0M_2M_3M_5M_6 = \Pi(0,2,3,5,6)$

Latihan:

1. Nyatakan fungsi Boolean f(x,y,z) = x + y′z dalam SOP dan POS.
2. Nyatakan fungsi Boolean f(x,y,z) = xy + x′z dalam bentuk POS.
3. Carilah bentuk kanonik SOP dan POS dari f(x,y,z) = y′ + xy + x′yz′

Konversi antar bentuk kanonik

$m_j' = M_j$

Fungsi Boolean dalam bentuk SOP:

$f(x,y,z) = \Sigma(1,4,5,6,7)$

dikonversikan ke bentuk POS menjadi

$f(x,y,z) = \Pi(0,2,3)$

Bentuk Baku

Dua bentuk kanonik adalah bentuk dasar yang diperoleh dengan membaca fungsi dari tabel kebenaran. Bentuk ini umumnya sangat jarang muncul karena setiap suku (term) di dalam bentuk kanonik harus mengandung literal atau peubah yang lengkap baik dalam bentuk normal x atau dalam bentuk komplemennya x′.

Cara lain untuk mengekspresikan fungsi Boolean adalah bentuk baku (standard). Pada bentuk ini suku-suku yang dibentuk fungsi dapat mengandung satu, dua, atau sejumlah literal. Dua tipe bentuk baku adalah baku SOP dan baku POS.

Contoh:

f(x,y,z) = y′ + xy + x′yz (bentuk baku SOP)

F(x,y,z) = x(y′ + z)(x′ + y + z′) (bentuk baku POS)

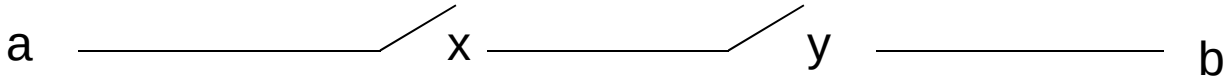
Aplikasi Aljabar Boolean

Aljabar Boolean mempunyai aplikasi yang luas, antara lain bidang jaringan pensaklaran dan rangkaian digital.

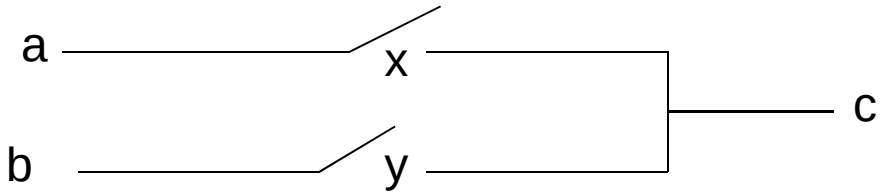
1. Aplikasi dalam jaringan pensaklaran (Switching Network)

Saklar adalah obyek yang mempunyai dua buah keadaan: buka dan tutup. Kita asosiasikan setiap peubah dalam fungsi Boolean sebagai “gerbang” (gate) didalam sebuah saluran yang dialiri listrik, air, gas, informasi atau benda lain yang mengalir secara fisik, gerbang ini dapat berupa kran di dalam pipa hirolik, transistor atau dioda dalam rangkaian listrik, dispatcher pada alat rumah tangga, atau sembarang alat lain yang dapat melewatkan atau menghambat aliran.

Kita dapat menyatakan fungsi logika untuk gerbang yang bersesuaian. Pada fungsi tersebut, peubah komplemen menyatakan closed gate, sedangkan peubah bukan komplemen menyatakan opened gate.

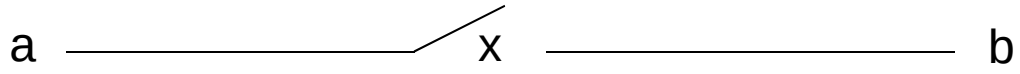


Saklar dalam hubungan SERI: logika AND

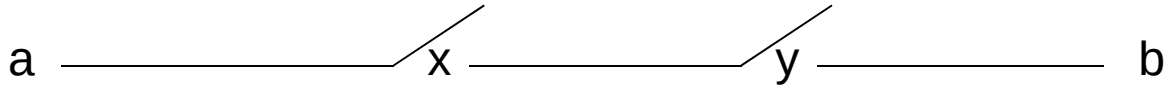


Saklar dalam hubungan PARALEL: logika OR

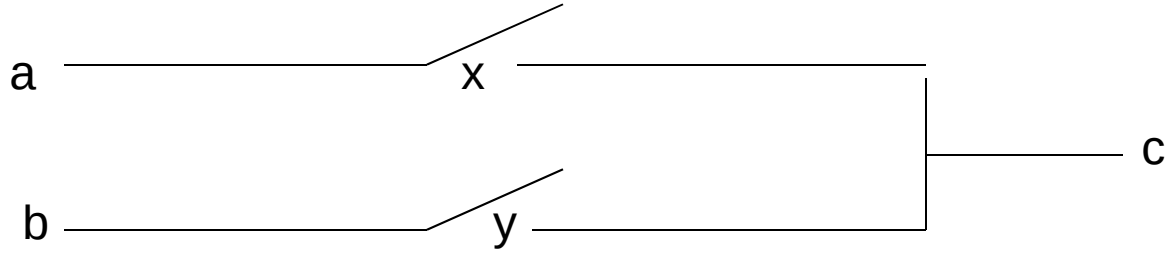
Contoh: tiga bentuk gate paling sederhana:

1. 

Output b hanya ada jika dan hanya jika x tertutup $\Rightarrow$ x

2. 

Output b hanya ada jika dan hanya jika x dan y tertutup $\Rightarrow$ xy

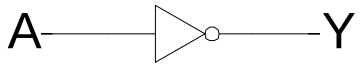
3. 

Output c hanya ada jika dan hanya jika x atau y tertutup $\Rightarrow$ x + y

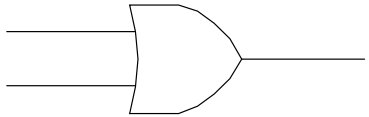
2. Aplikasi dalam rangkaian digital elektronik

Rangkaian digital elektronik biasanya dimodelkan dalam bentuk gerbang logika. Ada tiga macam gerbang logika dasar: AND, OR dan NOT. Secara fisik, rangkaian logika diimplementasikan dalam rangkaian listrik spesifik

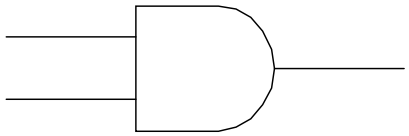
Gerbang NOT(inverter)



Gerbang OR



Gerbang AND



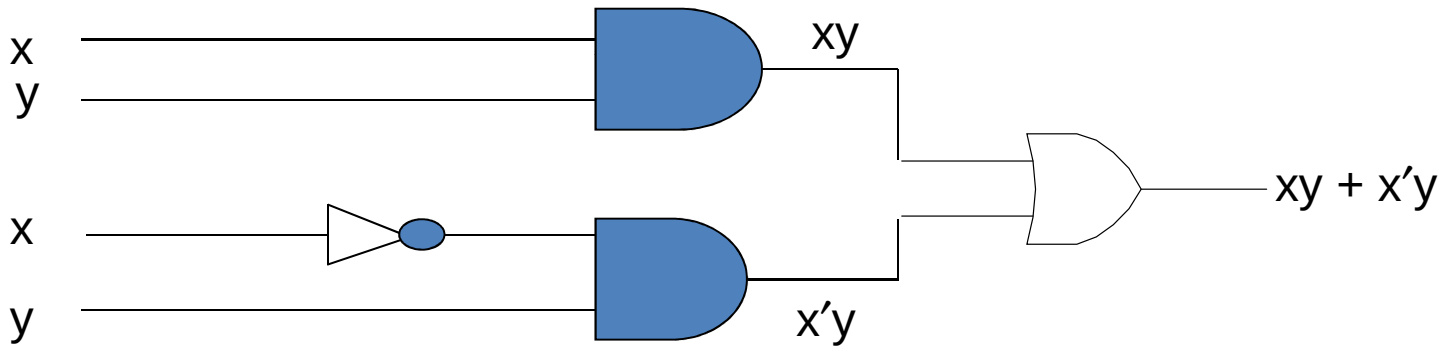
Latihan:

Nyatakan fungsi Boolean berikut ke dalam bentuk rangkaian pensaklaran dan rangkaian digital.

1. $f(x,y,z) = x'y + (x'+xy)z + x(y+y'z+z)$
2. $f(x,y) = xy' + x'y$
3. $f(x,y,z) = xy + xy'z + y(x' + z) + y'z'$

Contoh:

Nyatakan fungsi f(x,y,z) = xy + x′y ke dalam rangkaian logika.



Penyederhanaan Fungsi Boolean

Fungsi Boolean seringkali mengandung operasi-operasi biner yang tidak perlu, literal atau suku-suku yang berlebihan.

Contoh:

$f(x,y) = x'y + xy' + y'$ dapat disederhanakan menjadi $f(x,y) = x' + y'$

Penyederhanaan fungsi Boolean dapat dilakukan dengan 3 cara:

1. Secara aljabar, menggunakan rumus-rumus/aksioma-aksioma yang berlaku pada fungsi Boolean.
2. Menggunakan Peta Karnaugh
3. Menggunakan metode Quine Mc Cluskey (metode tabulasi)

1. Aljabar

   Jumlah literal di dalam sebuah fungsi Boolean dapat diminimumkan dengan manipulasi aljabar. Sayang tidak ada aturan khusus yang harus diikuti yang akan menjamin menuju ke jawaban akhir. Metode yang tersedia adalah prosedur ***cut-and-try*** yang memanfaatkan postulat, teorema dasar, dan metode manipulasi lain yang sudah dikenal.

Contoh:

1. $f(x,y) = x + x'y = (x + x')(x + y) = 1\ (x + y) = x + y$
2. $f(x,y) = x(x' + y) = xx' + xy = 0 + xy = xy$

2. Peta Karnaugh

Adalah sebuah diagram/peta yang terbentuk dari kotak-kotak yang bersisian. Tiap kotak merepresentasikan sebuah minterm. Peta Karnaugh dengan jumlah kotak lebih dari empat buah akan memiliki sisi yang berseberangan. Sisi yang berseberangan tersebut sebenarnya merupakan sisi yang bersisian juga. Artinya sebuah Peta Karnaugh dapat dibayangkan sebagai sebuah kubus atau balok atau silinder yang tersusun atas kotak-kotak itu.

Pembangunan Peta Karnaugh biasanya didasarkan pada tabel kebenaran fungsi Boolean yang akan disederhanakan.

a. Peta Karnaugh dengan dua peubah

m_0	m_1
m_2	m_3

X	Y = 0	Y = 1
0	x′y′	x′y
1	xy′	xy

Diberikan tabel kebenaran dan Peta Karnaugh-nya

x	y	f(x,y)
0	0	0
0	1	0
1	0	0
1	1	1

X	Y = 0	Y = 1
0	0	0
1	0	1

Fungsi Boolean yang merepresentasikan tabel diatas adalah f(x,y) = xy

b. Peta dengan tiga peubah

m0	m1	m3	m2
m4	m5	m7	m6

x	yz 00	01	11	10
0	x′y′z′	x′y′z	x′yz	x′yz′
1	xy′z′	xy′z	xyz	xyz′

Diberikan tabel kebenaran dan Peta Karnaugh-nya

x	y	Z	F(x,y,z)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

x	yz 00	01	11	10
0	0	0	0	1
1	0	0	1	1

Fungsi Boolean yang mereprentasikan tabel kebenaran adalah

f(x,y,z) = x′yz′ + xyz′ + xyz

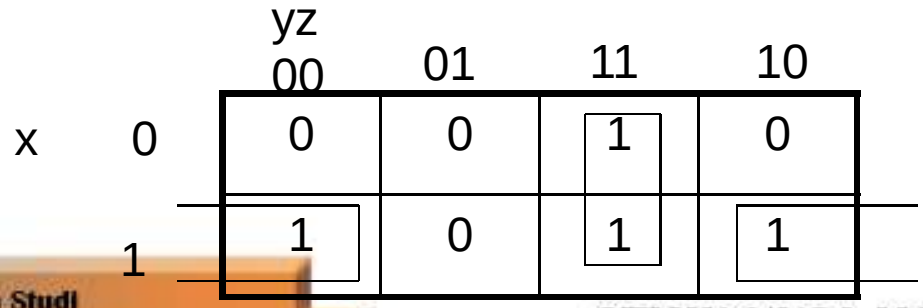
Teknik Minimisasi Fungsi Boolean dengan Peta Karnaugh

Penggunakan Peta Karnaugh dalam penyederhanaan fungsi Boolean dilakukan dengan menggabungkan kotak-kotak yang bersisian. Perhatikan bahwa kotak yang berseberangan juga dianggap sebagai kotak yang bersisian.

Contoh: Sederhanakan fungsi Boolean

$f(x,y,z) = x'yz + xy'z' + xyz + xyz'$

Jawab: Peta Karnaugh untuk fungsi tersebut adalah:



Hasil penyederhanaan: $f(x,y,z) = yz + xz'$

Latihan:

a. Sederhanakan dengan cara Aljabar

1. $f(x,y,z) = x'y'z + x'yz + xy'$
2. $f(x,y,z) = xy + x'z + yz$
3. $f(x,y,z) = (x + y)(x' + z)(y + z)$

b. Sederhanakan dengan metode Peta Karnaugh dan gambarkan rangkaian logika sebelum dan setelah disederhanakan

$f(x,y,z) = x'yz + x'yz' + xy'z' + xy'z$

Pertemuan 10
Algoritma dan Hubungan Rekurensi

Algoritma

adalah suatu langkah-langkah logis untuk menyelesaikan masalah. Bedanya kalau algoritma setiap langkah difokuskan pada sistem komputer dan data.

Beberapa hal yang harus dipahami dalam mencari algoritma antara lain:

- Masalah seperti apa yang hendak diselesaikan?
- Gagasan apa yang ada pada algoritma tersebut?
- Beberapa lama yang diperlukan untuk menyelesaikan masalah?
- Berapa jumlah data yang dapat ditangani oleh algoritma tersebut?

Untuk mengetahui seberapa besar kualitas suatu algoritma, dinyatakan dengan notasi-O besar (big O-notation) untuk menyatakan tingakat kekomplekan suatu algoritma.

Notasi O-besar

Menyatakan kelompok kompleksitas suatu algoritma atau jumlah operasi yang dilakukan oleh algoritma.

Tabel Kompleksitas dengan

Notasi O-besar

Kelompok Algoritma	Nama
O(1)	Konstan
O(log n)	Logaritmik
O(n)	Linier
O(nlogn)	N log n
O(n2)	Kuadratik
O(n3)	Kubik
O(2n)	Eksponensial
O(n!)	Faktorial

Contoh:

Jenis Algoritma	Kompleksitas
Linier search	O(n)
Binary search	O(log n)
Bubble sort	O(n2)
Selection sort	O(n2)
Insertion sort	O(n2)
Merge sort	O(n log n)
Quick sort	O(n log n)
Heap sort	O(n log n)

Definisi:

T(n) = O(f(n)) (T(n) adalah O(f(n)) yang artinya T(n) berorde paling besar f(n)) bila terdapat C dan n0 sedemikian hingga:

$T(n) \leq C(f(n))$

Untuk $n \geq n0$, artinya jika sebuah algoritma mempunyai waktu asimtotik O(f(n)), maka jika n dibuat semakin besar waktu yang dibutuhkan tidak akan pernah melebihi suatu tetapan C dikali f(n). Jadi f(n) adalah batas atas dari T(n) untuk n yang besar. T(n) dikatakan berorde paling besar f(n).

Penjelasan masing-masing kelompok algoritma sebagai berikut:

O(1) berarti waktu untuk menjalankan adalah konstan, artinya tidak bergantung pada ukuran data masukan n. Ini berarti waktu tetap sama meskipun n menjadi dua kali semula atau lebih.

O(log n) berarti perubahan waktu akan lebih kecil dari perubahan masukan n. Algoritma yang masuk dalam kelompok ini adalah algoritma yang memcahkan persoalan besar dengan membagi masalah besar menjadi masalah yang kecil yang berukuran sama. Cont: Binary search. Di sini bilangan pokok/basis log tidak terlalu penting karena semua fungsi algoritmik meningkat 2 kali semula jika n dinaikkan menjadi n2.

O(n) algoritma yang waktu bergantung kepada n secara linier, artinya jika n dinaikkan sebesar x kali maka waktu menjalankan algoritma itu juga naik sebesar x kali. Umunya algoritma pencarian beruntun

O(n log n)n Waktu pelaksanaan n log n terdapat pada algoritma yang memecahkan masalah menjadi beberapa bagian yang lebih kecil dimana setiap persoalan diselesaikan secara independen dan akhirnya solusi masing-masing digabung. Cont teknik bagi dan. Bila n dinaikkan menjadi 2 kali maka n log n meningkat tidak sampai dua kalinya, tetapi kalau n dinaikkan lebih dari 10 kali maka n log n naik lebih cepat dari kenaikkan n

O(n2) algoritma ini akan berjalan lebih lambat dari algoritma linier maupun logaritmik, sehingga hanya praktis untuk persoalan yang berukuran kecil. Bila n dinaikkan dua kali maka waktu pelaksanaan algoritma meningkat menjadi empat kali karena prose setiap masukkannya dalam dua buah kalang bersarang.

O(n3) memproses setiap masukkan dalam tiga buah kalang bersarang, misal perkalian matrik. Algoritma ini akan menjadi lebih lambat dari algoritma kuadratik meskipun n kecil atau besar. Bila n dinaikkan dua kali maka waktu pelaksanaan algoritma meningkat menjadi delapan kali.

O(2n) algoritma ini berjalan lebih lambat dari semua algoritma sebelumnya, khususnya untuk n besar. Bila n dinaikkan dua kali, waktu pelaksanaan menjadi kuadrat kali tetapi jika n= 100, maka waktu pelaksanaannya adalah 1000000 sama dengan algoritma n3.

O(n!) jenis ini memproses setiap masukkan dan menghubungkannya dengan n-1 masukkan lainnya, misal algoritma persoalan pedagang keliling. Bila n = 5 maka waktu pelaksanaan algoritma 120. Bila n dinaikkan dua kali, maka waktu pelaksanaan menjadi 2n!

Algoritma Pencarian

1. Pencarian beruntun (*sequential search*)
2. Pencarian Biner (*binary search*)

Pencarian beruntun pada array yang tidak terurut

Proses dimulai dari data pertama sampai data terakhir / data ke-n

Berikut ini algoritmanya:

```
Input ( cari)    { meminta data niali yang akan dicari}
   I=1           { indeks array dimulai dari 1}
   while (I<n) and (A[I] ≠ cari) do
        I = I+1
  end while
  if A[I] = cari then
            output   ('Data berada di index nomor', I)
  else      output ('Data tidak ditemukan')
   endif
```

Pencarian beruntun pada Array yang sudah Terurut

Kasus terburuk dari pencarian beruntun kalau data tidak ditemukan atau terletak paling akhir. Kalau data sudah terurut maka akan lebih baik kalau proses pencarian dihentikan apabila nilai data yang dibandingkan sudah lebih besar dari nilai data yang dicari.

Berikut ini algoritmanya:

```
Input ( cari)    { meminta data niali yang akan dicari}
   I=1           { indeks array dimulai dari 1}
   while (I<n) and (A[I] < cari) do
        I = I+1
  end while
  if A[I] = cari then
            output   ('Data berada di index nomor', I)
  else      output ('Data tidak ditemukan')
  endif
```

Pencarian Biner

Dilakukan untuk memperkecil jumlah operasi perbandingan yang harus dilakukan antara data yang dicari dengan data yang ada di dalam tabel, khususnya untuk data yang besar.

Prinsip dasarnya membagi 2 data sampai data ditemuakan.

Algoritmanya:

```
 Input (cari)   {meminta niali data yang akan dicari}
 Batasatas = 1    {indeks array dimulai dari 1}
Batasbawah = n
Ketemu = false
While (batasatas < batasbawah) and (not ketemu) do
Tengah = (batasatas+batasbawah)div2
If A[tengah] = cari then ketemu = true
  else if (A[tengah] <cari) then { cari dibagian kanan}
          batasatas = tengah + 1
  else batasbawah = tengah-1 {cari dibagian kiri}
  endif
  endif
endwhile
```

```
if (ketemu) then
        output (‘ data berada di index nomor’, tengah)
else output (‘data tidak ditemukan’)
endif
```

Algoritma Pengurutan

Yang akan dibahas meliputi bubble sort, selection sort, dan insertion sort

1. Bubble Sort

   menggunakan prinsip kerja gelembung udara

Algoritma Bubble sort

```
For I = 1 to (n-1)
    for J = N downto (I+1) do
        if data [J] < data [J-1] then
            Bubble = data [J]
            data [J] = data [J-1]
            data [J-1] = bubble
        endif
    endfor
endfor
```

2. Selection Sort

salah satu metode pengurutan berdasarkan prioritas antrian.

Algoritma Selection Sort

```
For I = 1 to (n-1)
        for J = (I+1) to N do
            if data [J] < data [kecil] then
                kecil = J
            endif
                endfor
                sementara = data[I]
  data[I] = data[kecil]
  data[kecil] = sementara
endfor
```

3. Insertion Sort

Metode ini dilakukan dengan penyisipan nilai data untuk suatu aaray A yang tidak terurut ke dalam suatu tempat kosong C dan memastikan nilai data C selalu dalam keadaan terurut.

Algoritma Insertion sort

```
For I = 2 to N do
        sementara = data[I]
         j = I – 1
       while (sementara < data[J]) and (J>1) do
             data [J+1] = data [J]
              J = J – 1
        endwhile
        if sementara ≥ data[J] then
            data[J+1] = sementara
        else data [J+1] = data [J]
                data [J] = sementara
       endif
  endfor
```

Hubungan Rekurensi

Sebuah hubungan rekurensi untuk urutan a0, a1,…adalah sebuah persamaan yang menghubungkan a0 dengan suku tertentu pendahulunya a0,a1,…,an-1. Kondisi awal untuk urutan a0,a1,… secara eksplisit memberikan nilai kepada sejumlah suku-suku tertentu dalam urutan itu.

Rekurensi adalah proses perulangan yaitu memanggil dirinya sendiri (fungsi/prosedur) dengan parameter yang berbeda, sampai pengulangan berhenti.

Cara lain menyelesaikan rekurensi adalah:

1. Memecahkan masalah yang besar menjadi dua submasalah.
2. Submasalah tersebut diselesaikan dengan cara yang sama untuk memecahkan masalah yang besar tersebut.

Permasalahan yang menggunakan metode rekuren diantaranya:

1. Faktorial
2. Fibonanci
3. Menara Hanoi

Pertemuan 11

TEORI, APLIKASI DAN TERMINOLOGI GRAF

Teori graf digunakan untuk mempresentasikan objek-objek diskrit dan hubungan antara objek-objek tersebut. Representasi visual dari graf adalah dengan menyatakan objek dinyatakan sebagai noktah, bulatan, atau titik, sedangkan hubungan antara objek dinyatakan dengan garis.

Definisi graf

Graf G didefinisikan sebagai pasangan himpunan (V,E) yang dalam hal ini:

V = himp berhingga dan tidak kosong dari simpul-simpul (vertices atau node)

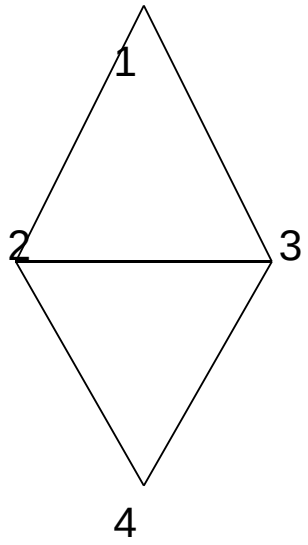
$= \{v_1, v_2, v_3, \ldots, v_n\}$

E = himp sisi (edges atau arcs) yang menghubungkan sepasang simpul.

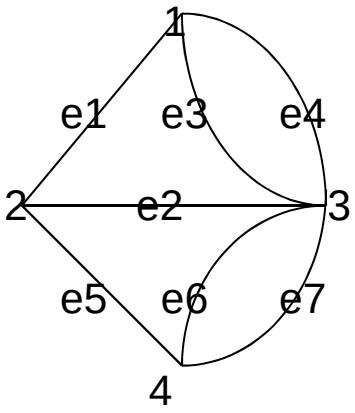
$= \{e_1, e_2, e_3, \ldots, e_n\}$

Simpul pada graf dapat dinomori dengan huruf , bilangan asli atau keduanya, sedangkan sisi yang menghubungkan simpul vi dengan simpul vj dinyatakan dengan pasangan (v_i,v_j) atau $e_1,e_2,e_3,\ldots$ atau $e = (v_i,v_j)$

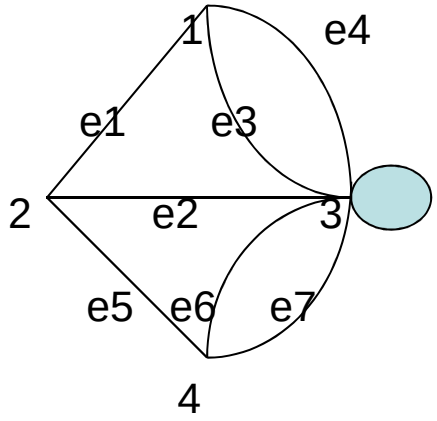
Contoh:



(a)G1



(b)G2



(c)G3

Tiga buah graf (a) graf sederhana, (b) graf ganda dan (c) graf semu

Jenis-jenis graf

Graf dapat dikelompokkan menjadi beberapa jenis bergantung pada sudut pandang pengelompokannya. Pengelompokan graf dapat dipandang berdasarkan ada tidaknya sisi ganda atau sisi kalang, berdasarkan jumlah simpul atau berdasarkan orientasi arah pada sisi.

Berdasarkan ada tidaknya gelang atau sisi ganda pada suatu graf dapat digolongkan menjadi dua jenis:

1. Graf sederhana (simple gragh)

   Graf yang tidak mengandung gelang maupun sisi ganda

2. Graf tak sederhana (unsimple graph)

   Graf yang mengandung sisi ganda atau gelang

Ada dua macam graf tidak sederhana yaitu:

a. graf ganda (multigraph)

graf yang mengandung sisi ganda

b. graf semu (pseudograph)

graf yang mengandung gelang

Berdasarkan jumlah simpul pada graf, digolongkan menjadi dua jenis:

1. Graf berhingga (limited graph)

graf yang jumlah n simpulnya berhingga (bisa dihitung)

2. Graf tak berhingga (unlimited graph)

graf yang jumlah n simpulnya tak berhingga (tak terbatas)

Berdasarkanorientasi arah pada sisi, dibedakan atas dua jenis:

1. Graf tak berarah (undirected graph)

   graf yang setiap sisinya tidak mempunyai orientasi arah (vi,vk) = (vk,vi)

2. Graf berarah (directed graph atai digraph)

   graf yang setiap sisinya diberikan orientasi arah (vi,vk) ≠ (vk,vi). Sisi berarah disebut busur/arc. vi dinamakan simpul asal vk dinamakan simpul terminal. Cont: aliran proses, peta lalu lintas suatu kota.

Contoh terapan graf

1. Rangkaian listrik
   menyatakan arus yang masuk dan ke luar setiap simpul
2. Isomer senyawa kimia karbon
   untuk menghitung isomer CH4 atom C dan atom H dinyatakan simpul dan ikatan antara C dan H sebagai sisi.
3. Transaksi konkuren pada basis data terpusat
   dalam bidang informatika, dalam basis data terpusat melayani beberapa transaksi yang dilakukan secara konkuren (bersamaan). Transaksi berupa pembacaan dan penulisan terhadap data yang sama. Persoalan kritis terjadi deadlock yaitu keadaan yang timbul akibat transaksi saling menunggu yang disebut hang. Digambarkan dengan graf.
4. Peng

4. Pengujiaan program

penerapan graf berarah di mana simpul menyatakan pernyataan atau kondisi yang dievaluasi dan busur menyatakan aliran kendali program ke pernyataan atau kondisi

```
cont: read x
        while x<> 9999 do
        begin
            if x < 0 then
            writeln ('masukkan tidak boleh negatif')
            else
                x := x+10;
            read x
        end
        writeln x
```

5. Terapan graf pada teori otomata
   digunakan untuk menggambarkan cara kerja dan arah kegiatan suatu mesin.
6. Turnamen round-robin
   Setiap tim bertanding dengan tim lain hanya satu kali, tim menyatakan simpul dan pertandingan menyatakan busur.

Terminologi Graf

Dalam Pembahasan mengenai graf , kita sering menggunakan terminologi (istilah) yang berkaitan dengan graf. Dibawah ini didaftarkan beberapa terminologi yang sering dipakai.

1. Ketetanggaan (Adjacent)

   dua buah simpul dikatakan bertetangga bila keduanya terhubung langsung.

2. Bersisian (Incidency)

   untuk sembarang sisi / edges e = (v_j,v_k) dikatakan

   e bersisian dengan simpul v_j

   e bersisian dengan simpul v_k

3. Simpul yang terpencil (Isolated graph)

   ialah simpul yang tidak mempunyai sisi yang bersisian dengannya, atau tidak ada satupun bertetangga dengan simpul-simpul lainnya.

4. Graf kosong

   Definisi graf menyatakan bahwa V tidak boleh kosong, sedangkan E boleh kosong. Jadi sebuah graf dimungkinkan tidak mempunyai sisi satupun tetapi simpulnya harus ada, minimal satu.

5. Derajat (degree)

   Derajat suatu simpul adalah jumlah sisi yang bersisian dengan simpul tersebut.

6. Lintasan (Path)

   lintasan dari simpul-simpul dalam G adalah rangkaian sisi-sisi yang menghubungkan dari simpul awal hingga simpul akhir.

Macam lintasan

- Lintasan sederhana (simple path) adalah lintasan dengan semua sisi yang dilalui hanya satu kali.
- Lintasan elementer (elementary path) adalah lintasan dengan semua simpul yang dilalui hanya muncul satu kali, kecuali mungkin simpul pertama dan simpul terakhir.
- Lintasan tertutup (closed walk) adalah lintasan yang berawal dan berakhir pada simpul yang sama.
- Lintasan terbuka (open walk) adalah lintasan yang berawal dan berakhir pada simpul yang tidak sama.

7. Siklus (cycle) atau sirkuit (circuit)

   lintasan elementer dengan simpul pertama sama dengan simpul yang terakhir.

Panjang sirkuit adalah jumlah sisi dalam sirkuit tersebut.

Sirkuit sederhana (simple path) adalah sirkuit dengan semua sisi yang dilalui hanya satu kali.

8. Terhubung (connected)

   dua buah simpul v_1 dan v_2 disebut terhubung jika terdapat lintasan dari v_1 ke v_2.

9. Pohon (tree) adalah graf terhubung yang tidak mempunyai sirkuit.

10. Upagraf (subgraf) dan komplemen upagraf

    upagraf adalah suatu graf yang merupakan bagian dari graf yang lain; Komplemen upagraf adalah kebalikan dari graf yang lain.

11. Upagraf rentang (spanning subgraf) adalah suatu graf bagian yang memuat semua simpul graf asal.

12. Cut-set dari graf terhubung G adalah himpunan sisi yang bila dibuang dari G memnyebabkan G tidak terhubung. Jadi cut-set selalu menghasilkan dua buah komponen.

13. Graf berbobot (weighted graph)

    adalah graf yang setiap sisinya diberi sebuah harga (bobot). Bobot pada setiap sisi dapat menyatakan jarak antara dua buah kota, biaya perjalanan, waktu tempuh pesan / message dari sebuah simpul komunikasi ke simpul komunikasi lain, ongkos produksi, dsb.

Beberapa graf sederhana khusus

a. Graf lengkap (*complete graph*) ialah graf sederhana yang setiap simpulnya mempunyai sisi ke semua simpul yang lainnya. Jumlah sisi pada graf lengkap n(n-1)/2

b. graf lingkaran adalah graf sederhana yang setiap simpulnya berderajat dua. Cont : hubungan LAN

c. Graf teratur (*regular graphs*) ialah graf yang setiap simpulnya mempunyai derajat yang sama. Jumlah sisinya nr/2, dimana n = simpul dan r = derajat.

d. Graf bipartite (*bipartite graph*) adalah graf G yang himpunan simpulnya dapat dipisah menjadi 2 himpunan bagian V_1 dan V_2, sehingga setiap sisi pada G menghubungkan simpul di V_1 ke sebuah simpul di V_2

Soal-soal Latihan

1. Himpunan simpul-simpul yang dihubungkan oleh sisi-sisi disebut…….

   a. Graf
   b. Pohon
   c. vertex
   d. edges
   e. node

2. Graf yang tidak mengandung gelang maupun sisi ganda disebut graf…….

   a. Berhingga
   b. Sederhana
   c. Berarah
   d. Tak sederhana
   e. Tak berhingga

2. Graf yang tidak mengandung gelang maupun sisi ganda disebut graf…….

a. Berhingga
b. Sederhana
c. Berarah
d. Tak sederhana
e. Tak berhingga

3. Dalam pengujian program kita menerapkan jenis graf…..

a. Sederhana
b. Tak berarah
c. Berarah
d. Tak sederhana
e. Tak berhingga

3. Dalam pengujian program kita menerapkan jenis graf…..
   a. Sederhana
   b. Tak berarah
   c. Berarah
   d. Tak sederhana
   e. Tak berhingga

4. Lintasan elementer dengan simpul awal sama dengan simpul akhir disebut…….
   a. Derajat
   b. Terhubung
   c. Simpul terpencil
   d. Siklus
   e. Pohon

4. Lintasan elementer dengan simpul awal sama dengan simpul akhir disebut…….

a. Derajat
b. Terhubung
c. Simpul terpencil
d. Siklus
e. Pohon

5. Jumlah sisi pada graf lengkap dirumuskan dengan…….

a. n-1
b. (n-1)/2
c. nr/2
d. 2n
e. n(n-1)/2

5. Jumlah sisi pada graf lengkap dirumuskan dengan…….

a. n-1
b. (n-1)/2
c. nr/2
d. 2n
e. n(n-1)/2

1. Himpunan simpul-simpul yang dihubungkan oleh sisi-sisi disebut…….

a. Graf
b. Pohon
c. vertex
d. edges
e. node

PERTEMUAN 12

TEORI GRAF LANJUTAN

Representasi graf

Untuk maksud pemprosesan graf dengan program komputer, graf harus dipresentasikan di dalam memori. Terdapat beberapa representasi yang mungkin untuk graf. Dalam bab ini akan dibahas 3 jenis representasi yang sering digunakan, yaitu matriks ketegangan, matriks bersisian dan senarai ketegangan.

1. Matriks ketetanggaan (adjacency matrix)

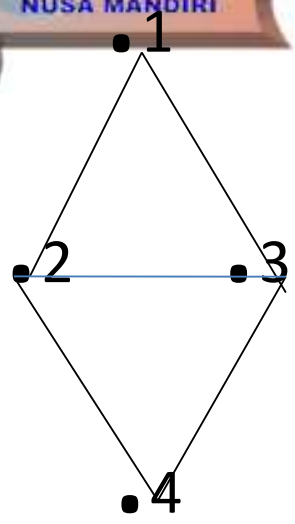
representasi jenis ini yang paling umum. Misalkan G=(V,E) adalah graf dengan n simpul n≥1. Matriks ketetanggaan G adalah matriks dwimatra yang berukuran n x n. Jika matriks dinamakan A = [aij] maka

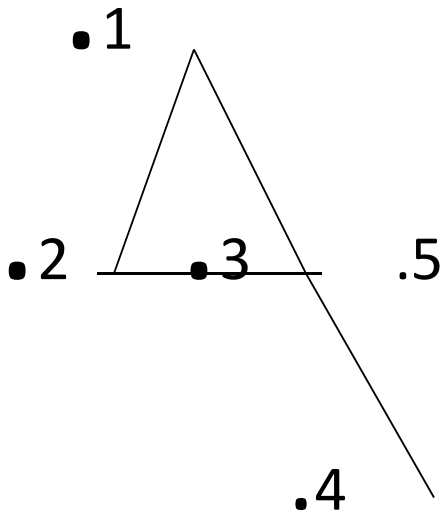
$$a_{ij} = \begin{cases} 1, \text{ jika simpul I dan j bertetangga} \\ 0, \text{ jika simpul I dan j tidak bertetangga} \end{cases}$$

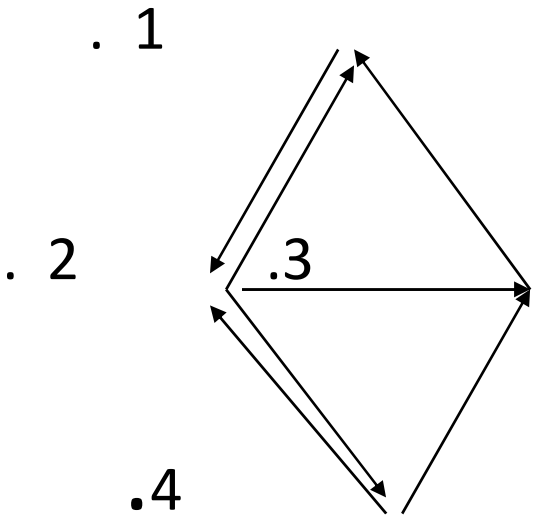
Matriks ketetanggan hanya bernilai 0 dan 1, maka matriks tersebut dinamakan matriks nol-satu (zero-one).selain itu matriks juga bisa dinyatakan dengan nilai false(menyatakan 0) dan true (menyatakan 1). Matriks ketetanggan didasarkan pada pengurutan nomor simpul, disini terdapat n! cara pengurutan nomor simpul berarti ada n! matriks ketetanggan berbeda untuk graf dengan n simpul.

Contoh:

Perhatikan graf sederhana dan matriks ketetanggaannya, dari graf terhubung, graf tak terhubung dan graf berarah berikut ini!







	1	2	3	4
1	0	1	1	0
2	1	0	1	1
3	1	1	0	1
4	0	1	1	0

	1	2	3	4	5
1	0	1	1	0	0
2	1	0	1	0	0
3	1	1	0	1	0
4	0	0	1	0	0
5	0	0	0	0	0

	1	2	3	4
1	0	1	0	0
2	1	0	1	1
3	1	0	0	0
4	0	1	1	0

Kelemahan dari matriks ketetanggaan adalah tidak dapat untuk mempresentasikan graf yang mempunyai sisi ganda. Untuk mengatasinya, maka elemen aij pada matriks ketetanggaan sama dengan jumlah sisi yang berasosiasi dengan (vi,vj). Matriks ketetanggaannya bukan lagi matriks nol-satu. Untuk graf semu, gelang pada simpul vi dinyatakan dengan nilai 1 pada posisi (i,i) dimatrks ketetanggaannya.

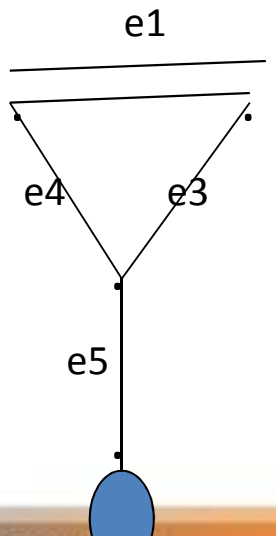
2. Matriks Bersisian (incidency matrix)

adalah matriks yang menyatakan kebersisian simpul dengan sisi. Misalkan G = (V,E) adalah graf dengan n simpul dan m buah sisi. Matriks bersisian G adalah matriks dwimatra yang berukuran n x m. Baris menunjukkan label simpul, sedangkan kolom menunjukkan label sisinya. Bila matriks disebut A = [aij], maka

$$a_{ij} = \begin{cases} 1, \text{ jika simpul i bersisian dengan sisi j} \\ 0, \text{ jika simpul i tidak bersisian dengan sisi j} \end{cases}$$

Matriks bersisian dapat digunakan untuk merepresentasikan graf yang mengandung sisi ganda atau sisi gelang.

Contoh:



	e1	e2	e3	e4	e5
1	1	1	0	1	0
2	1	1	1	0	0
3	0	0	1	1	1
4	0	0	0	0	1

3. Senarai Ketetanggaan (adjacency list)

Kelemahan matriks ketetanggaan adalah bila graf memiliki jumlah sisi relatif sedikit, matriksnya bersifat jarang (sparse) yaitu mengandung banyak elemen nol, sedangkan elemen yang bukan nol sedikit. Sehingga jika ditinjau dari teknis implementasi, kebutuhan ruang memorinya boros karena komputer menyimpan elemen 0 yang tidak perlu. Untuk mengatasinya kita gunakan representasi yang ketiga yaitu senarai ketetanggaan. Senarai ketetanggaan mengenumerasi simpul-simpul yang bertetangga dengan setiap simpul di dalam graf.

Contoh: Berdasarkan graf contoh representasi 1, kita buat senarai ketetanggaannya

SIMPUL	SIMPUL TETANGGA
1	2,3
2	1,3,4
3	1,2,4
4	2,3

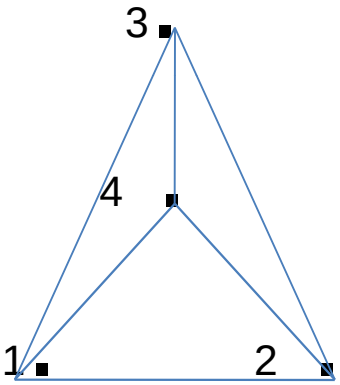
SIMPUL	SIMPUL TETANGGA
1	2,3
2	1,3
3	1,2,4
4	3
5	-

SIMPUL	SIMPUL TETANGGA
1	2
2	1,3,4
3	1
4	2,3

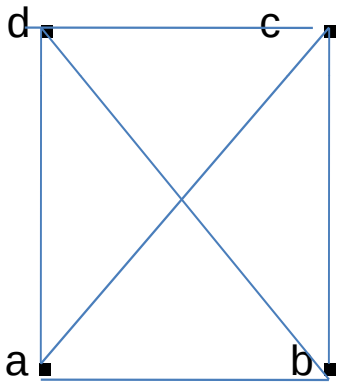
Graf Isomorfik (Isomorphic Graf)

Adalah dua buah graf yang sama tetapi secara geometri berbeda

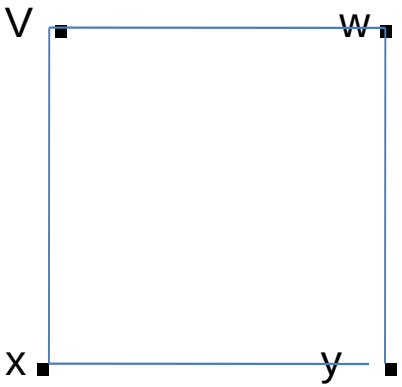
Cont:



G1



G2



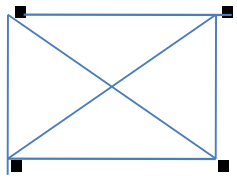
G3

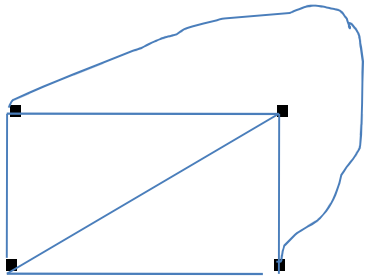
G1 isomorfik dengan G2, tetapi G1 tidak isomorfik dengan G3

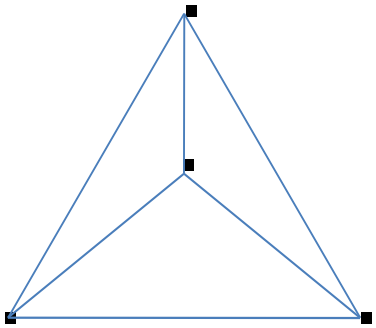
Graf Planar (Planar Graph) dan Graf Bidang (Plane Graph)

adalah graf yang dapat digambarkan pada bidang datar dengan sisi-sisi tidak saling memotong. Dan graf planar yang digambarkan dengan sisi-sisi yang tidak saling berpotongan disebut graf bidang (*plane graph*).

Cont:







Rumus Euler

Jumlah wilayah (f) pada graf planar sederhana juga dapat dihitung dengan rumus Euler sebagai berikut:

$n - e + f = 2$ atau $f = e - n + 2$

n = jumlah simpul

e = jumlah sisi

Cont:

Misal e = 11 dan n = 7, maka f = 11-7+2 = 6

Graf Dual

Adalah graf yang dibuat dengan cara setiap wilayah graf lama buatlah simpul untuk graf baru dan buat sisi baru yang memotong sisi graf lama untuk menghubungkan simpul graf yang baru.

Lintasan dan Sirkuit Euler

Lintasan Euler adalah lintasan yang melalui masing-masing sisi di dalam graf tepat satu kali.

Sirkuit Euler adalah suatu lintasan Euler yang kembali ke simpul awal, membentuk lintasan tertutup, dengan kata lain sirkuit euler adalah sirkuit yang melewati masing-masing sisi tepat satu kali.

Graf yang mempunyai sirkuit Euler disebut **graf Euler** sedangkan graf yang mempunyai lintasan Euler disebut graf **semi-Euler.**

Lintasan dan Sirkuit Hamilton

Lintasan hamilton adalah lintasan yang melalui tiap simpul di dalam graf tepat satu kali.

Sirkuit Hamilton adalah sirkuit yang melalui tiap simpul tepat satu kali, kecuali simpul asal (sekaligus simpul akhir) yang dilalui dua kali.

Graf yang mempunyai sirkuit Hamilton disebut **graf Hamilton**, sedangkan yang mempunyai lintasan Hamilton disebut **graf semi-Hamilton**

APLIKASI GRAF

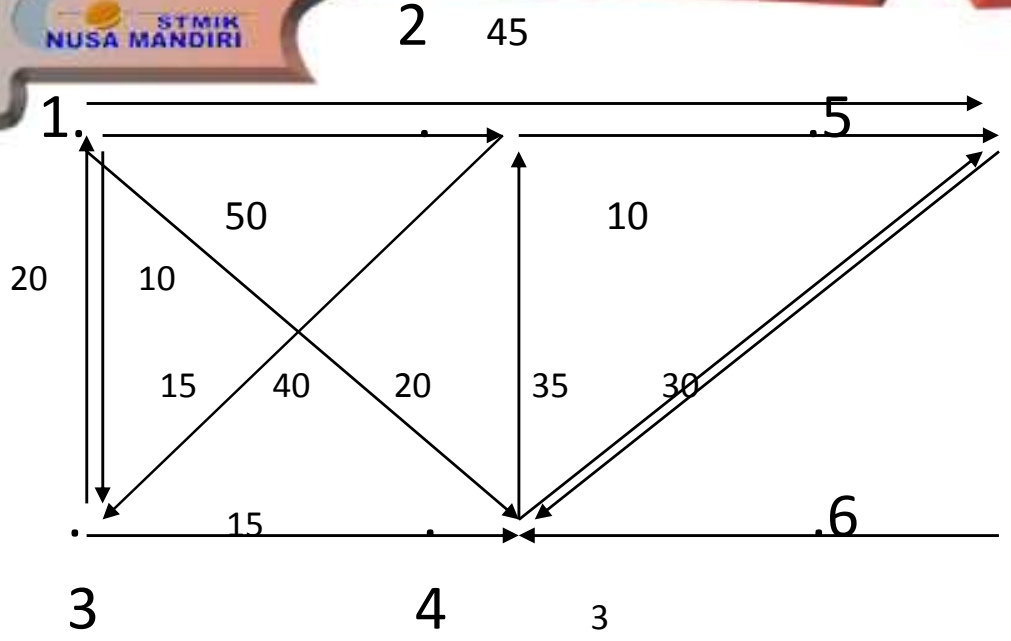
Terdapat banyak aplikasi graf, yang digunakan sebagai alat untuk merepresentasikan atau memodelkan persoalan. Berdasarkan graf yang dibentuk, barulah persoalan diselesaikan.Beberapa aplikasi yang berkaitan dengan lintasan/sirkuit didalam graf, yaitu menentukan lintasan terpendek (shortest path), persoalan pedagang keliling (travellingsalesperson problem), dan persoalan tukang pos Cina.

A. Lintasan Terpendek (Shortest Path)

Ciri:

1. lintasan memiliki arah
2. setiap lintasan ada bobotnya
3. setiap lintasan harus terhubung dengan simpul

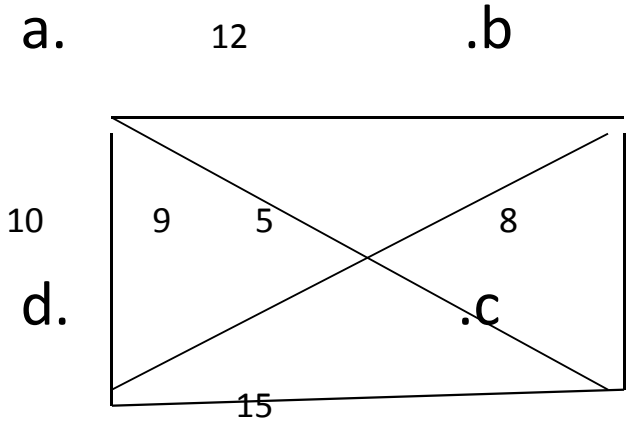
Misalkan simpul pada graf dapat merupakan terminal komputer atau simpul komunikasi dalam suatu jaringan, sedangkan sisi menyatakan saluran komunikasi yang menghubungkan dua terminal. Bobot pada graf dapat menyatakan biaya pemakaian saluran komunikasi antara dua terminal, jarak antara dua buah terminal, atau waktu pengiriman pesan (message) antara dua terminal. Persoalan lintasan terpendek disini adalah menentukan jalur komunikasi terpendek antara dua buah terminal komputer. Lintasan terpendek akan menghemat waktu pengiriman pesan dan biaya komunikasi.



Tentukan lintasan terpendeknya !

B. Peroalan perjalanan pedagang (travelling salesperson problem TSP)

Persoalan ini diilhami oleh seorang pedagang yang mengunjungi sejumlah kota. Uraiannya sebagai berikut: diberikan sejumlah kota dan jarak antar kota. Tentukan sirkuit terpendek yang harus dilalui oleh seorang pedagang bila pedagang itu berangkat dari sebuah kota asal dan menyinggahi setiap kota tepat satu kali dan kembali ke kota asal berangkat.



C. Persoalan tukang pos Cina (Chinese Postman Problem)

Ditemukan pertama kali oleh Mei Gan tahun 1962. Masalahnya adalah sebagai berikut:

"Seorang tukang pos akan mengantar surat ke alamat-alamat sepanjang jalan disuatu daerah. Bagaimana ia merencanakan rute perjalanannya supaya melewati setiap jalan tepat sekali dan kembali lagi ke tempat awal keberangkatan.

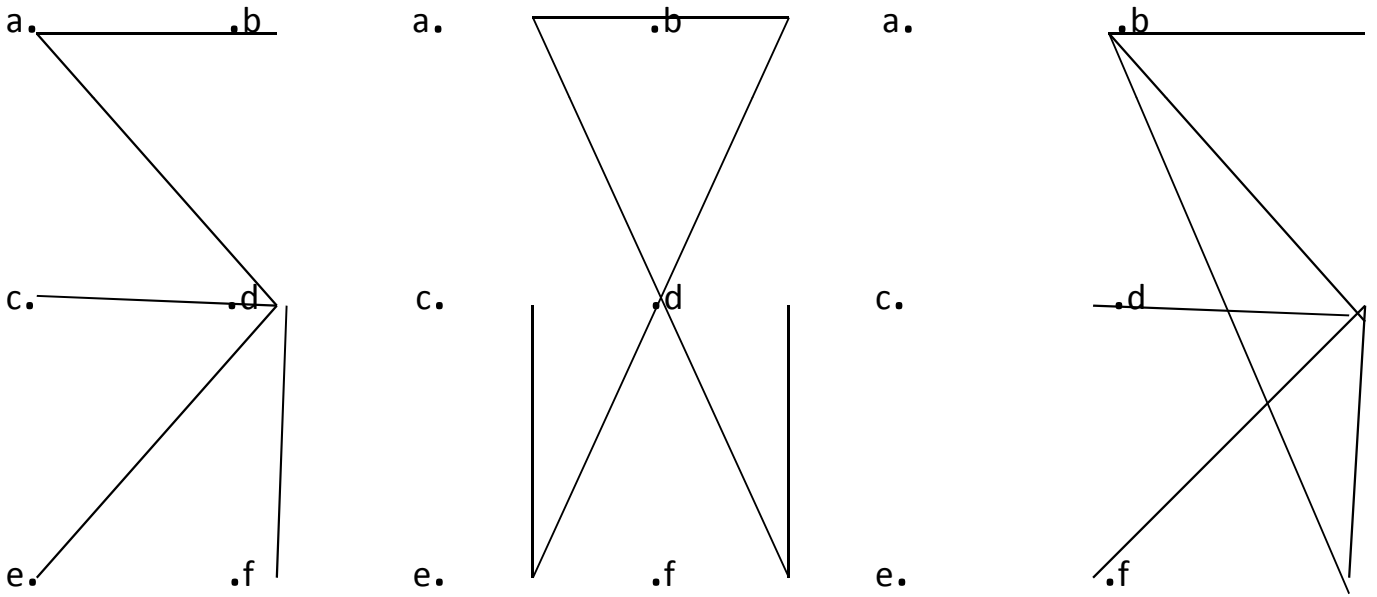
PERTEMUAN 13

POHON (TREE)

Definisi Pohon

Adalah bentuk khusus dari graf, atau graf tak berarah terhubung yang tidak mengandung sirkuit.

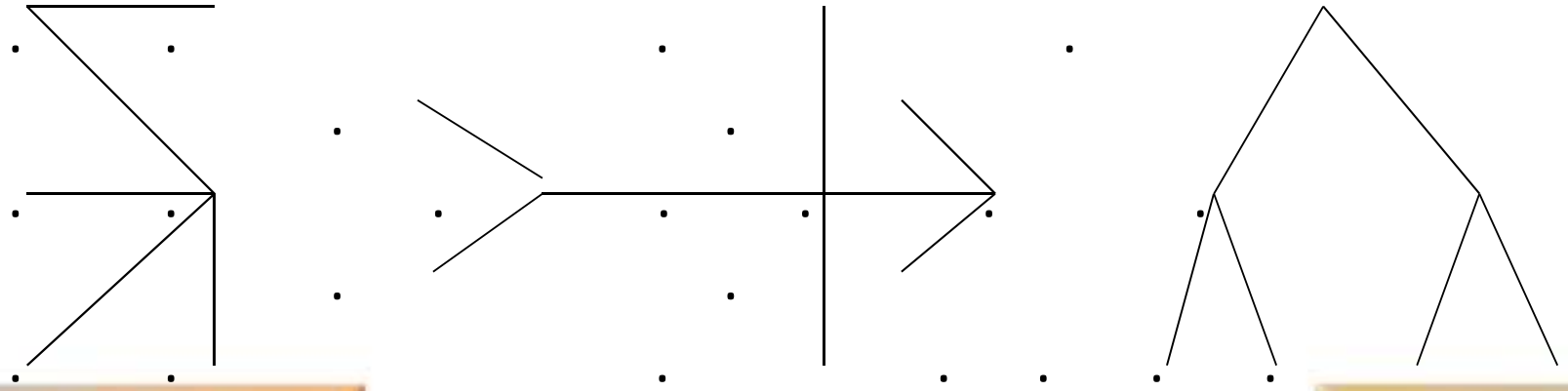
Contoh: bedakan mana yang pohon dan yang bukan pohon



G1 G2 G3

Contoh lain: misalkan himpunan V = {a,A,b,B,c,C,d,D} adalah empat pasangan suami-istri tukang gosip, dengan a,b,c dan d para suami dan A,B,C dan D para istri.Misalkan a mencaritakan gosip lewat telpon ke istri A, yang kemudian A menelpon para istri lainnnya untuk menyebarkan gosip itu, dan setiap istri itu menelpon dan menceritakan gosip kepada suami-suami masing2, bagaimana bentuk pohonnya?

Hutan (forest) adalah kumpulan pohon yang saling lepas. Atau graf tidak terhubung yang tidak mengandung sirkuit.



1.2 Sifat-sifat pohon

Teorema 1.

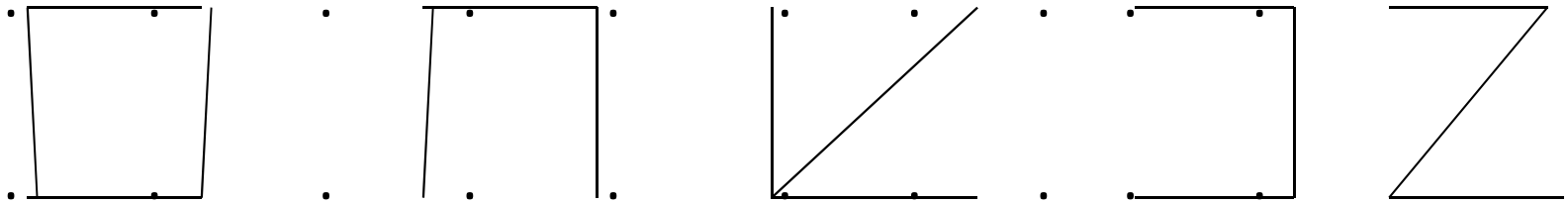
Misalkan T = (V,E) adalah graf tak-berarah sederhana dan jumlah simpulnya n. Maka, semua pernyataan di bawah ini adalah ekivalen:

1. T adalah pohon
2. Setiap pasang simpul di dalam T terhubung dengan lintasan tunggal.
3. T terhubung dan memiliki m = n-1 buah sisi.
4. T tidak mengandung sirkuit dan memiliki m = n-1 buah sisi.
5. T tidak tidak mengandung sirkuit dan penambahan satu sisi pada graf akan membuat hanya satu sirkuit.
6. G terhubung dan semua sisinya adalah jembatan (jembatan adalah sisi yang dihapus menyebabkan graf terpecah menjadi dua komponen)

1.3. Pohon Rentang

Misalkan G=(V,E) adalah graf tak-berarah terhubung yang bukan pohon, yang berarti di G terdapat beberapa sirkuit. G dapat diubah menjadi pohon T=(v1,E1) dengan cara memutuskan sirkuit-sirkuit yang ada. Secara berulang-ulang disebut pohon rentang (spanning tree).

Contoh: graf lengkap dengan empat buah pohon rentang.



penerapan pohon rentang misalnya pada pemeliharaan jalan raya. Karena keterbatasan dana pemeliharaan, pemerintah daerah mempertimbangkan hanya jalan-jalan sesedikit mungkin sehingga ke empat kota masih tetap terhubung satu sama lain.

Teorema 2: setiap graf terhubung mempunyai paling sedikit satu buah pohon rentang.

Dari teorema diatas graf yang tidak mengandung sirkuit adalah pohon rentang itu sendiri, sedangkan graf yang mengandung sirkuit, pohon rentangnya diperoleh dengan cara memutuskan sirkuit yang ada.

Sisi pada pohon rentang disebut **cabang** (branch) adalah sisi dari graf semula sedangkan **tali-hubung** (chord atau link) dari pohon adalah sisi yang tidak terdapat di dalam pohon rentang. Himpunan tali-hubung beserta simpul yang bersisian dengannya disebut **komplemen pohon**.

Graf G dengan n buah simpul dan m buah sisi, dapat dicari jumlah cabang dan tali-hubung dengan rumus:

jumlah cabang = n-1

jumlah tali-hubung = m-n+1

Graf tidak terhubung dengan k komponen, m buah sisi dan n buah simpul

jumlah cabang = n-k

jumlah tali-hubung = m-n+k

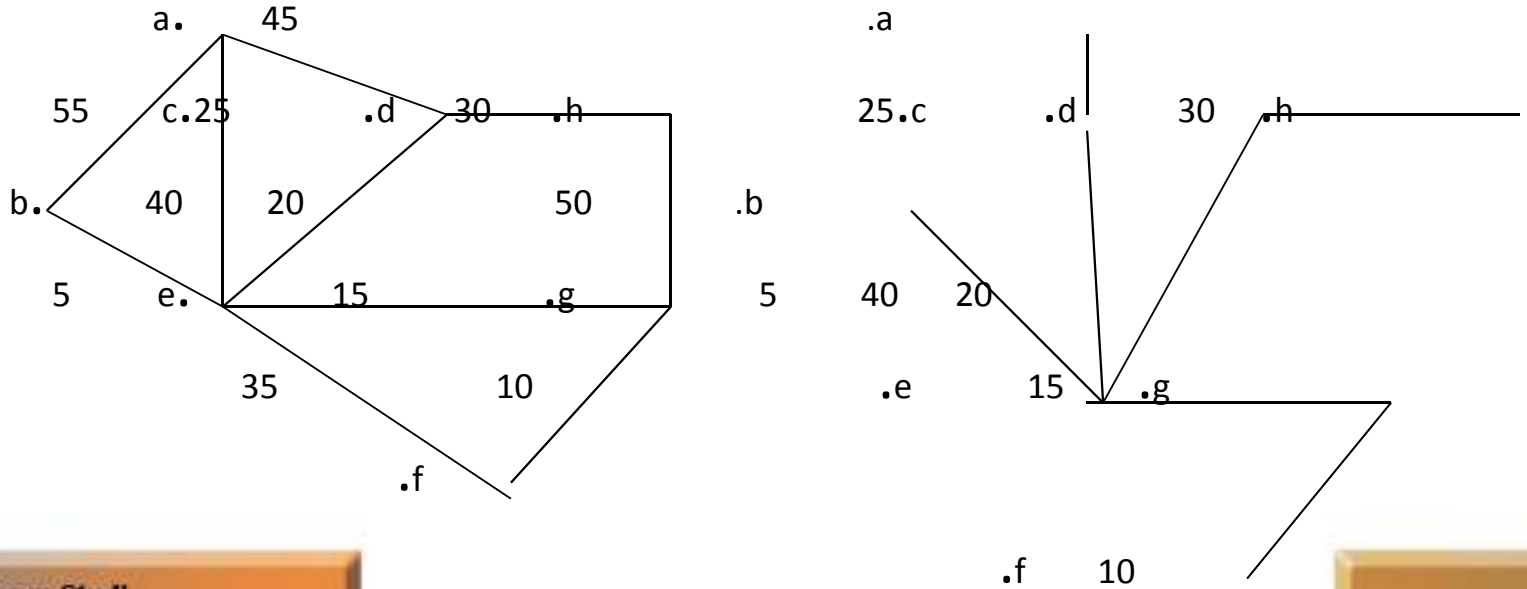
Jumlah cabang pada pohon rentang daribsebuah graf G disebut *rank* graf G dan jumlah tali-hubung G disebut *nullity* graf G. Sehingga :

rank + *nullity* = jumlah sisi graf G

1.3.1 Pohon Rentang Minimum (*minimum spanning tree*)

Jika G adalah graf berbobot, maka bobot pohon rentang T dari G didefinisikan sebagai jumlah bobot semua sisi di T. Semua pohon rentang di G, pohon rentang yang berbobot minimum dinamakan pohon rentang minimum (minimum spanning tree). Jenis ini mempunyai terapan yang luas.

Contohnya: Pemerintah akan membangun jalur rel kereta api yang menghubungkan sejumlah kota. Karena biayanya mahal, pembangunan jalur ini tidak perlu menghubungkan langsung dua buah kota, tetapi cukup membangun jalur kereta seperti pohon rentang. Karena dalam sebuah graf mungkin saja terdapat lebih dari satu pohon rentang, maka harus dicari pohon rentang yang mempunyai jumlah jarak terpendek, dengan kata lain harus dicari pohon rentang minimum.



Dua algoritma untuk menyelesaikan pohon rentang minimum yaitu :

1. Algoritma Prim
2. Algoritma Kruskal

1. Algoritma Prim

membentuk pohon rentang minimum langkah demi langkah. Setiap langkah kita ambil sisi dari graf G yang mempunyai bobot minimum namun terhubung dengan pohon rentang minimum yang telah terbentuk.

Algoritma Prim

langkah 1: ambil sisi dari graf G yang berbobot minimum,masukkan kke dalam T

langkah 2: pilih sisi (u,v) yang mempunyai bobot minimum dan bersisian dengan simpul di T, tetapi (u,v) tidak membentuk sirkuit di T. tambahkan (u,v) ke dalam T

Langkah 3: ulangi langkah 2 sebanyak n-2 kali

2. Algoritma Kruskal

Sisi-sisi graf diurut terlebih dahulu berdasarkan bobotnya yang meenaik (dari kecil ke besar). Sisi yang dimasukkan ke dalam himpunan T adalah sisi graf G sehingga T adalah pohon. Pada keadaan awal, sisi-sisi yang sudah diurut berdasarkan bobot membentuk hutan. Masing-masing pohon di hutan hanya berupa satu simpul. Hutan tersebut dinamakan hutan pohon rentang.

Algoritma Kruskal

(langkah 0: sisi-sisi dari graf sudah diurut menaik berdasarkan bobotnya dari bobot kecil ke bobot besar)

Langkah 1 : T masih kosong

Langkah 2 : Pilih sisi (u,v) dengan bobot minimum yang tidak membentuk sirkuit di T. Tambahkan (u,v) ke dalam T.

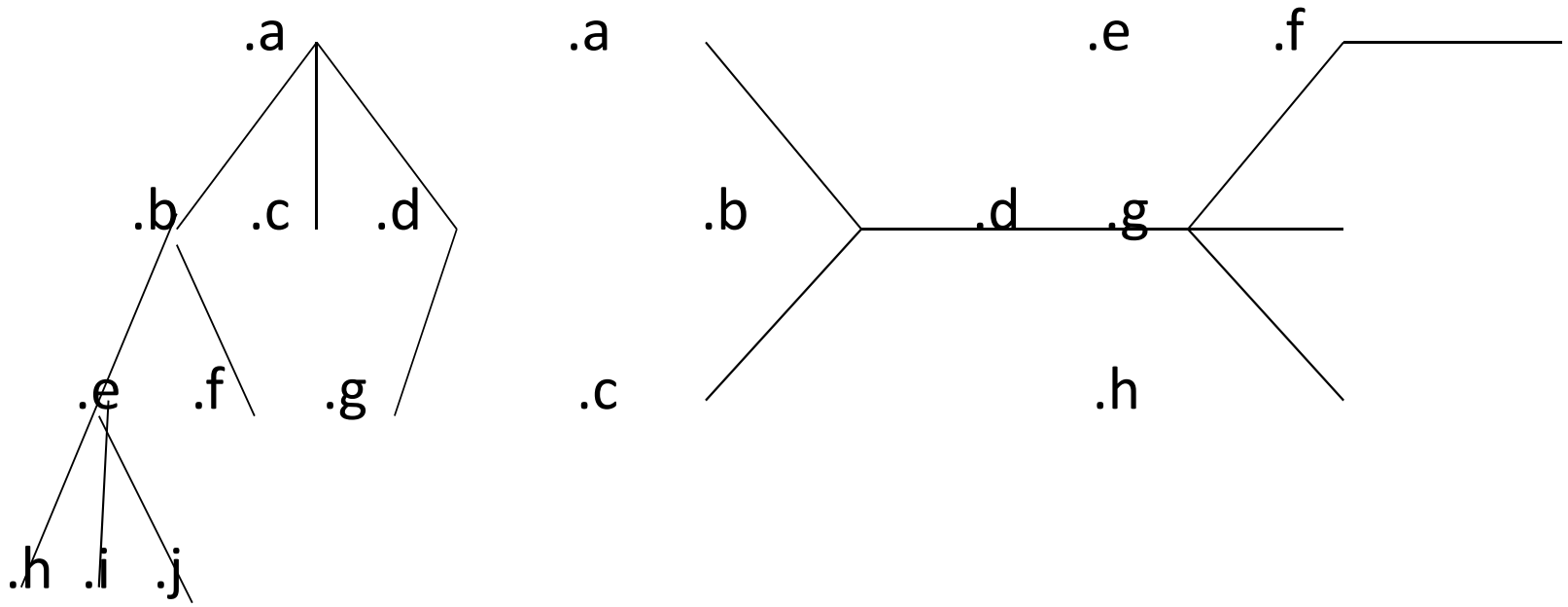
Langkah 3 : ulang langkah 2 sebanyak n-1 kali

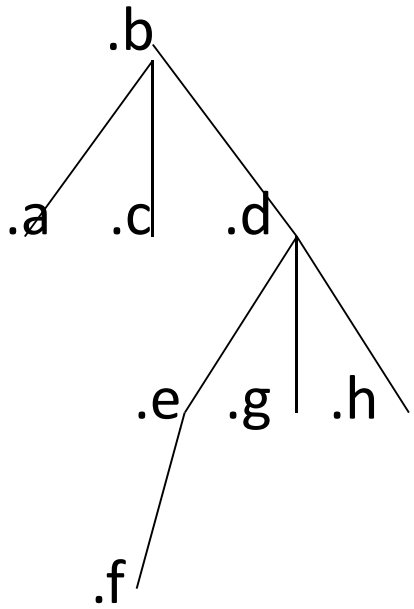
1.4 Pohon Berakar

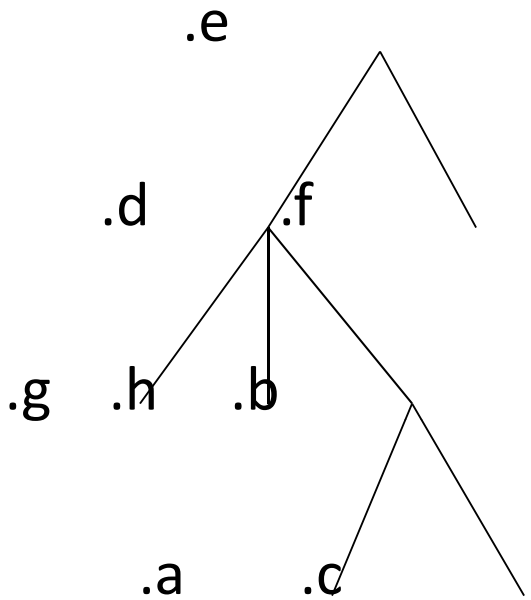
Dalam aplikasi pohon, sering simpul tertentu diperlukan sebagai akar (*root*). Sekali sebuah simpul ditetapkan sebagai akar, maka simpul-simpul lainnnya dapat dicapai dari akar dengan memberi arah pada sisi-sisi pohon yang mengikutinya. Pohon yang satu buah simpulnya diperlakukan sebagai akar dan sisi-sisinya diberi arah sehingga menjadi graf berarah dinamakan pohon berakar (*rooted tree*).

Akar mempunyai derajat masuk sama dengan nol dan simpul-simpul lainnya berderajat masuk sama dengan satu. Simpul yang mempunyai derajat keluar sama dengan nol disebut daun atau simpul terminal. Simpul yang mempunyai derajat keluar tidak sama dengan nol disebut simpul dalam atau simpul cabang. Setiap simpul di pohon dapat dicapai dari akar dengan sebuah lintasan tunggal (unik)

Sembarang pohon tak berakar dapat diubah menjadi pohon berakar dengan memilih sebuah simpul sebagai akar. Pemilihan simpul yang berbeda menjadi akar menghasilkan pohon berakar yang berbeda pula.







Beberapa terminologi pada pohon

1. Anak (child atau children) dan orangtua (parent)

   Misal X simpul pada pohon berakar, simpul Y dikatakan anak dari simpul X jika ada sisi yang menghubungkan X ke Y. dan X disebut orang tua Y

2. Lintasan (path)

   Lintasan dari simpul v1 ke vk, adalah runtunan simpul v1, v2,…,vk sedemikian sehingga vi orang tua dari vi+1 untuk $1 \leq i \leq k$ contoh lintasan dari a ke h adalah a, b, e, h dengan **panjang lintasan** adalah jumlah sisi yang dilalui dalam suatu lintasan k-1 ada 3

3. Keturunan (descendant) dan leluhur (ancestor)

   Jika terdapat lintasan dari simpul X ke Y di dalam pohon , maka X adalah leluhur Y dan Y adalah keturunan X

4. Saudara kandung (*sibling*)

   Simpul yang berorangtua yang sama adalah saudara sekandung.

5. Upapohon (*subtree*)

   Misalkan X adalah simpul di dalam pohon T. Yang dimaksud dengan upapohon dengan X sebagai akarnya ialah upagraf T' = (V',E') sedemikian sehingga V' mengandung X dan semua keturunannya dan E' mengandung sisi-sisi dalam semua lintasan yang berasal dari X.

6. Derajat (*degree*)

   Jumlah upapohon (atau jumlah anak) pada simpul tersebut.

7. Daun (*leaf*)

   Simpul yang berderajat nol (atau tidak mempunyai anak)

8. Simpul Dalam (*internal nodes*) adalah Simpul yang mempunyai anak.

9. Aras (*level*) atau tingkat

   Akar mempunyai aras = 0, sedangkan aras simpul lainnya = 1 + panjang lintasan dari akar ke simpul tersebut.

10. Tinggi (*height*) atau kedalaman (*depth*)

    Aras maksimum dari suatu pohon.

Jenis Pohon yang lain

1.5 Pohon Terurut adalah pohon berakar yang urutan anak-anaknya penting.

1.6 Pohon m-ary adalah pohon berakar yang setiap simpul cabangnya mempunyai paling banyak m buah anak. Pohon m-ary dikatakan teratur atau penuh jika setiap simpul cabangnya mempunyai tepat m anak. Jika m=2, disebut pohon biner (*binary tree*).

Jumlah daun pada pohon m-ary teratur m pangkat h, h tinggi pohon.

Hubungan antara jumlah daun dan simpul dalam pada pohon m-ary teratur

i = t – 1,

i = banyaknya simpul dalam

t = banyaknya simpul daun

1.7 Pohon biner

adalah pohon yang setiap simpul cabangnya mempunyai maksimum 2 buah anak. Anak pertama disebut anak kiri (*left child*) dan anak kedua disebut anak kanan (*right child*). Pohon yang akarnya adalah anak kiri disebut upapohon kiri (*left subtree*) dan sebaliknya disebut upapohon kanan (*right subtree*).

Pohon biner seimbang adalah pohon yang memiliki tinggi upapohon kiri dan kanan seimbang. Dibuat dengan menyebarkan setengah dari jumlah simpul upapohon kiri dan setengah dari jumlah simpul upapohon kanan.

Terapan Pohon Biner

1. Pohon ekspresi (*expression tree*)
2. Pohon keputusan (*decision tree*)
3. Kode Huffman (*Huffman code*)
4. Kode Prefiks (*Prefix code*)

Pohon pencarian biner (*binary search tree*)

NB: Dikembangkan sendiri

PERTEMUAN 14
BAHASA FORMAL DAN MESIN STATUS TERHINGGA

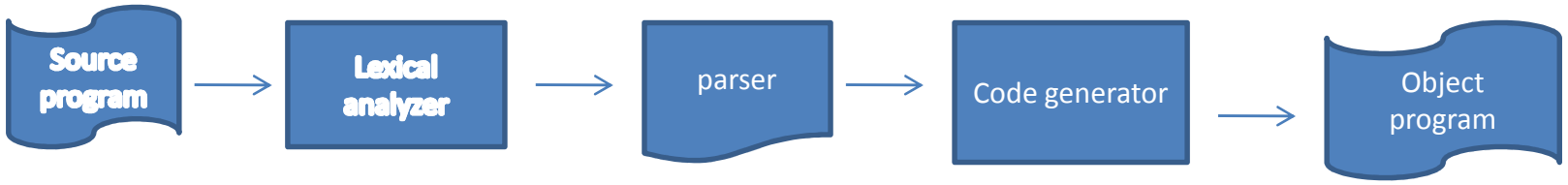
Bahasa Formal

Suatu dasar translasi bahasa berhubungan dengan pengolahan bahasa. Di mana dalam translasi bahasa terdapat istilah **bahasa formal** dan **bahasa natural**.

Bahasa natural adalah bagian dari komunikasi manusia. Cont : bahasa Inggris, Indonesia dll

Bahasa Formal adalah suatu bahasa yang harus mengikuti aturan bahasa pemrograman dan bahasa matematis seperti aljabar dan logika proposisi. Karena aturan tersebut akan mengkonstruksi programing translator untuk bahasa pemrograman.

Cont: language translator yaitu compiler untuk bahasa pemrograman. Yang pada dasarnya tugas translator adalah mengenali individual building block dari bahasa yang akan ditranslasi.



Proses translasi : Source Program menjadi object program

1. Proses mengenali runtunan multisimbol di dalam program ditangani oleh compiler yang disebut lexical analyser yaitu sub modul yang menerima source program sebagai string dari simbol dan menghasilkan string token. Token menyatakan objek tunggal yang dipresentasikan oleh lebih dari satu simbol dari input string.
2. Selanjutnya translator menganalisa pola dari token, misal if-then, if-then-else, atau perulangan loop. Analisis yang dilakukan disebut dengan parsing dan dilakukan oleh modul di dalam compiler yang disebut paser.

3. Ketika paser mengenali struktur di dalam program, maka modul selanjutnya adalah code generator untuk menghasilkan versi translasi dari bagian program. Setelah semua bagian translasi dibentuk, maka dibentuk versi translasi program yang disebut object program.

Tatabahasa Struktur Frasa

Dua hal penting untuk menspesifikasikan bahasa yaitu:

1. Jika diberikan spesifikasi suatu bahasa, secara otomatis akan membangkitkan satu atau lebih string di dalam bahasa itu.
2. Jika diberikan spesifikasi suatu bahasa, tentukan apakah suatu string tertentu akan termasuk di dalam bahasa itu atau tidak.

Suatu tatabahasa frasa dapat digunakan untuk menspesifikasikan suatu bahasa, yang terdiri dari empat unsur yaitu:

1. Himpunan terminal T, yaitu lambang atau simbol yang digunakan untuk membuat kalimat dalam bahasa, misal: objek/benda, dan kata sifat.
2. Himpunan nonterminal N, yaitu lambang antara yang digunakan untuk mendeskrepsikan struktur kalimat, misal: kalimat, frasa kata benda, kata-benda, kata-sandang dll
3. Himpunan produksi P, yaitu kaidah tatabahasa yang mengatur bagaimana kalimat di dalam bahasa itu dapat dibentuk, misal: $\alpha \rightarrow \beta$ dalam hal ini a dan β adalah rangkaian terminal dan non terminal.

4. Di antara semua nonterminal di dalam N, ada sebuah nonterminal khusus yang disebut sebagai simbol awal (*starting symbol*)

Jenis Tatabahasa dan jenis bahasa

Dalam bahasan ini akan digunakan A dan B untuk menyatakan nonterminal, a dan b untuk menyatakan terminal dan α dan β untuk menyatakan string terminal dan nonterminal.

Tatabahasa jenis -3 jika semua produksi di dalam tatabahasa berbentuk :

$A \longrightarrow a$ atau $A \longrightarrow Ba$

$A \longrightarrow aB$

Di dalam setiap produksi, string kirinya selalu berupa sebuah nonterminal tunggal, sedangkan string kanannya berupa sebuah terminal atau sebuah terminal yang diikuti dengan sebuah nonterminal.

Tatabahasa jenis-2 jika semua produksi di dalam tatabahasa berbentuk $A \longrightarrow \alpha$

di dalam setiap produksi, string kirinya selalu berupa sebuah nonterminal tunggal.

Tatabahasa jenis-1 jika semua produksi di dalam tatabahasa berbentuk $\alpha \longrightarrow \beta$

panjang β selalu lebih besar atau sama dengan α. Misalnya: $A \longrightarrow ab$

$A \longrightarrow aA$

$aAb \longrightarrow aBCb$

Tatabahasa jenis-0 yaitu tatabahasa struktur frasa tanpa pembatasan seperti yang telah didefinisikan oleh jenis 1, 2, 3. Semua bahasa pemrograman dapat dispesifikasi oleh tatabahasa struktur frasa dan kebanyakan merupakan bahasa jenis-2 misal: BASIC, Fortran, Pascal).

Automata Terhingga (*finite automata*)

Automata terhingga dan bahasa reguler adalah level terendah dari hirarki mesin dan bahasa. Salah satu aplikasinya adalah konstruksi pengkompilasi (compiler), yaitu pengenalan string dari simbol di kode sumber program yang harus direpresentasikan sebagai objek tunggal seperti nama variabel, konstanta dan *reserved word.*

Salah satu bentuk bentuk penggambaran untuk representasi strukturnya adalah diagram transisi (diagram status atau jaringan transisi)

Mesin Status Terhingga

Suatu mesin status terhingga mempunyai ciri-ciri sebagai berikut:

1. Himpunan terhingga status-status S={s0,s1, s2,…}
2. Unsur khusus di dalam himpunan S, yaitu s0 yang dinamakan status awal.
3. Himpunan terhingga huruf-huruf masukan I={i0,i1,i2,…}
4. Himpunan terhingga huruf-huruf keluaran O={o0,o1,o2,…}
5. Fungsi transisi yaitu fungsi f dari S x I ke S
6. Fungsi keluaran yaitu fungsi g dari S ke O

Suatu mesin status terhingga mempunyai sejumlah terhingga status. Setelah diterimanya sebuah huruf masukan, mesin akan memasuki status lain berdasarkan fungsi transisinya.

Mesin Status Terhingga sebagai Model Sistem Fisik

Mesin status terhingga dapat digunakan untuk memodelkan suatu sistem fisik, seperti cont: vending machine, mesin hitung modulo-3.

Mesin Status Terhingga sebagai Pengenal Bahasa

Mesin status terhingga dapat digunakan untuk memodelkan suatu alat untuk mengenali (menerima) kalimat-kalimat di dalam suatu bahasa.

Misalkan: O ={0,1} adalah simbol keluaran sebuah mesin status terhingga. Bernilai 1 artinya status menerima, dan bernilai 0 artinya status menolak.

Mesin Turing

Pada tahun 1936 seorang mtematikawan berkebangsaan Inggris bernama Alan Turing, mengusulkan suatu model sederhana yang mempunyai kemampuan sebuah komputer grneral-purpose. Mesin Turing mengenali kelas bahasa yang disebut sebagai himpunan terenumerasi rekursif dan dapat digunakan untuk menghitung kelas fungsi bilangan bulat yang dikenal sebagai fungsi rekursif parsial.

Komponen mesin turing :

1. Pengendali status terhingga
2. Pita masukan dengan sifat:
   - panjangnya tak terhingga (dari ujung kiri terbatas, ujung kanan tidak terbatas)
   - dapat dibaca dan ditulis.

Pada keadaan awal n sel pertama dari pita masukan berisi rangkaian simbol yang harus dikenali. Sel-sel lain disebelah kanan rangkaian simbol berisi simbol kosong.

Perilaku mesin Turing bergantung pada simbol masukan yang berada pada posisi kepala (*head*) baca/tulis dan status pengendalinya. Aksi-aksi yang dapat dilakukan yaitu:

1. Berubah status
2. Menuliskan simbol pada sel pita masukan. Aksi penulisan ini akan mengubah simbol yang sebelumnya berada pada sel tersebut.
3. Menggerakkan head ke kiri dan ke kanan.