MATEMATIKA DISKRIT

GRAHA ILMU

Edisi
2

MATEMATIKA DISKRIT

Samuel Wibisono

MATEMATIKA DISKRIT

Oleh : Samuel Wibisono

Editor : Asrining Rizky Rachmawati

Edisi Kedua
Cetakan Pertama, 2008

GRAHA ILMU
Candi Gebang Permai Blok R/6
Yogyakarta 55511
Telp. : 0274-882262; 0274-4462135
Fax. : 0274-4462136
E-mail : info@grahailmu.co.id

Wibisono, Samuel

MATEMATIKA DISKRIT/Samuel Wibisono

- Edisi Kedua – Yogyakarta; Graha Ilmu, 2008
xii + 196 hlm, 1 Jil. : 23 cm.

ISBN: 978-979-756-413-1

1. Matematika I. Judul

KATA PENGANTAR

Memasuki era globalisasi, mempersiapkan sumber daya manusia yang profesional dalam bidangnya merupakan prasyarat utama untuk dapat survive dalam pasar global yang penuh tantangan dan persaingan.

Dengan latar belakang tersebut di atas dan banyaknya keluhan pembaca tentang: “Apa manfaat belajar matematika buat mereka? atau Apa hubungan matematika yang mereka pelajari dengan jurusan yang mereka ambil?”, penulis menyadari bahwa sasaran dalam proses pembelajaran mata kuliah ini harus dipertajam, sehingga mampu mendukung terciptanya sarjana-sarjana baru dalam bidang teknik informatika, sistem informatika, manajemen informatika, maupun teknik komputer, yang handal dan mempunyai daya saing yang tinggi karena telah dibekali dengan logika dan konsep dasar matematika diskrit, sehingga mampu menyelesaikan segala persoalan yang dihadapi, melalui rancangan usulan penyelesaian problem atau kasus.

Hasil proses pembelajaran yang penulis harapkan setelah pembaca membaca buku ini, adalah:

- Pembaca mengenal konsep dasar logika dan matematika diskrit dengan baik.
- Pembaca memahami konsep dasar logika dan matematika diskrit sehingga mampu menggunakannya untuk menyelesaikan permasalahan yang sesuai.
- Pembaca dapat merancang, menganalisa dan mensintesa beberapa kasus aplikasi dalam berbagai bidang, khususnya TI dan komputer.

Pada kesempatan ini penulis menyampaikan terimakasih kepada Pimpinan dan Staf Universitas Bina Nusantara dan Universitas Indonesia Esa Unggul, di mana penulis diberi kesempatan mengampu mata kuliah Matematika Diskrit ini, rasa terima kasih juga penulis sampaikan kepada Penerbit Graha Ilmu yang telah memberikan kepercayaan, sehingga buku edisi 2 ini dapat diterbitkan.

Terakhir, kami sampaikan rasa terima kasih kepada rekan-rekan dosen pengampu mata kuliah Matematika Diskrit utamanya Dr. Frans Susilo SJ. yang berkenan memberikan kritik dan saran yang membangun guna penyempurnaan buku ini, kritik dan saran yang membangun dari rekan-rekan masih kami tunggu untuk edisi mendatang.

Demikian semoga bermanfaat.

Jakarta, Agustus 2008

Samuel Wibisono

DAFTAR ISI

-oo0oo-

1

LOGIKA PROPOSISI

1.1. PERNYATAAN

Logika proposisi sering juga disebut logika matematika ataupun logika deduktif.

Logika proposisi berisi pernyataan-pernyataan (dapat tunggal maupun gabungan).

Pernyataan adalah kalimat deklarasi yang dinyatakan dengan huruf-huruf kecil, misalnya:

$$p, q, r, s$$

Pernyataan mempunyai sifat dasar yaitu dapat bernilai benar (pernyataan benar) atau bernilai salah (pernyataan salah), tetapi tidak mungkin memiliki sifat kedua-duanya.

Kebenaran atau kesalahan sebuah pernyataan dinamakan nilai kebenaran dari pernyataan tersebut.

Contoh:

1. Bilangan biner digunakan dalam sistem digital adalah pernyataan yang benar.

2. Sistem analog lebih akurat daripada sistem digital adalah pernyataan yang salah.
3. Astaga, mahal sekali harga notebook itu adalah kalimat keheranan, bukan pernyataan.
4. Siang tadi notebook Ira jatuh dari meja adalah bukan pernyataan karena dapat bernilai benar maupun bernilai salah.
5. Corezdeo lebih bagus kinerjanya dan lebih mahal dari pentium IV generasi sebelumnya adalah pernyataan yang benar.

Kalimat-kalimat yang tidak termasuk pernyataan, adalah:

- Kalimat perintah
- Kalimat pertanyaan
- Kalimat keheranan
- Kalimat harapan
- Kalimat ……walaupun…..

1.2 Pernyataan Gabungan

Beberapa pernyataan dapat digabung dengan kata penghubung dan, atau, tidak/bukan, serta variatifnya, yang selanjutnya disebut pernyataan gabungan atau pernyataan majemuk atau *compound statement*.

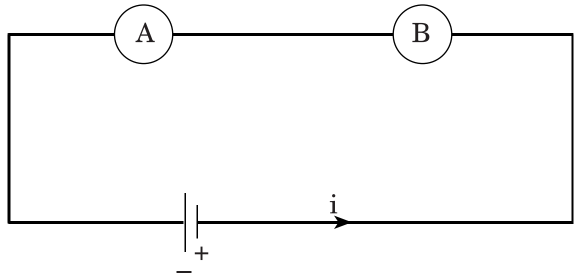
Macam-macam pernyataan gabungan.

1.2.1 Konjungsi

Konjungsi adalah pernyataan gabungan dari dua pernyataan dengan kata penghubung **dan** Notasi-notasi konjungsi:

$$p \wedge q\text{ , } p \text{ x } q\text{, } p.q\text{, } pq$$

Bagaimana menentukan benar atau salah sebuah konjungsi? Konjungsi dianalogikan dengan sebuah rangkaian listrik seri:



Bila lampu B dan lampu A hidup maka arus listrik dapat mengalir dari kutup positip menuju kutup negatip sebuah baterai, akibatnya kedua lampu A dan B menyala/hidup.Bila lampu B mati dan lampu A hidup atau sebaliknya, maka arus listrik tidak dapat mengalir menuju kutub negatip baterai, akibatnya kedua lampu A dan B tidak menyala/mati. Demikian juga bila lampu A dan B mati. Dengan demikian dapat di simpulkan bahwa konjungsi benar bila keduanya hidup, selain itu salah.

Tabel Kebenaran Konjungsi

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

atau

p	$\wedge$	q
+	+	+
+	−	−
−	−	+
−	−	−

dimana + berarti benar dan - berarti salah

Contoh:

p = sistem analog adalah suatu sistem dimana tanda fisik/ kuantitas, dapat berbeda secara terus-menerus melebihi jarak tertentu adalah pernyataan benar

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

r = sistem bilangan desimal adalah sistem bilangan yang digunakan dalam sistem digital adalah pernyataan yang salah

s = aljabar linear adalah alat matematika dasar untuk disain logika adalah pernyataan salah.

Maka:

$p \wedge q$ adalah konjungsi yang benar karena p benar, q benar.

$q \times r$ adalah konjungsi yang salah karena q benar, r salah.

$r \cdot s$ adalah konjungsi yang salah karena r salah, s salah.

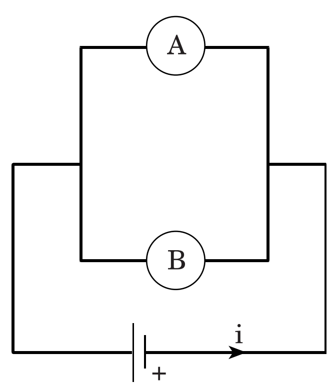
1.2.2 Disjungsi

Disjungsi adalah pernyataan gabungan dari dua pernyataan dengan kata penghubung **atau**.

Notasi-notasi disjungsi:

$$p \vee q, \; p + q$$

Bagaimana menentukan benar atau salah sebuah disjungsi? Disjungsi dapat dianalogikan dengan sebuah rangkaian listrik yang pararel:



Bila lampu A dan lampu B hidup maka arus listrik i dapat bergerak/mengalir dari kutup positip ke kutup negatip sebuah baterai, akibatnya lampu A dan B menyala.

Bila lampu A hidup dan lampu B mati (atau sebaliknya), maka arus listrik i masih dapat mengalir dari kutup positip ke kutup negatip sebuah baterai. Akibatnya lampu yang hidup akan menyala dan yang mati tidak menyala.

Bila lampu A dan B mati, maka arus listrik i tidak dapat mengalir ke kutup negatip.

Dengan demikian dapat disimpulkan bahwa disjungsi salah bila kedua lampu mati, selain itu benar.

Tabel Kebenaran Disjungsi

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

atau

p	$\vee$	q
+	+	+
+	+	–
–	+	+
–	–	–

Catatan:

Simbol tabel kebenaran yang biasa digunakan :

Benar = T, B, +, 1
Salah = F, S, –, 0

Contoh:

p = keyboard adalah alat yang dapat digunakan untuk input data kedalam komputer adalah pernyataan benar.

q = Harddisk adalah alat yang menentukan kecepatan kerja komputer adalah pernyataan salah.

r = Procesor alat yang berfungsi sebagai otak dari sebuah komputer adalah pernyataan benar.

s = Windows XP adalah sistematika menulis buku adalah pernyataan salah.

Maka:

$p \vee q$ adalah disjungsi yang benar karena p benar, q salah.

$p \vee r$ adalah disjungsi yang benar karena p benar, r benar.

$q \vee s$ adalah disjungsi yang salah karena q salah, s salah.

1.2.3 Negasi

Negasi adalah sebuah pernyataan yang meniadakan pernyataan yang ada, dapat di bentuk dengan menulis "adalah salah bahwa…" atau dengan menyisipkan kata " tidak" dalam sebuah pernyataan.

Notasi-notasi negasi:

$$\sim p, p', \overline{p}$$

Contoh:

p = Harddisk adalah alat yang menentukan kecepatan kerja komputer adalah pernyataan salah

Maka

~ p = Adalah salah bahwa harddisk adalah alat yang menentukan kecepatan kerja komputer adalah pernyataan benar.

Jadi kebenaran sebuah negasi adalah lawan dari kebenaran pernyataannya.

Tabel kebenaran negasi:

p	~p
+	–
–	+

1.2.4 Jointdenial (Not OR/ NOR)

Jointdenial adalah pernyataan gabungan yang dihasilkan dari menegasikan disjungsi.

Notasi NOR:

$$p \downarrow q, \text{p nor q}, \sim (p \vee q)$$

Karena jointdenial adalah negasi dari or, maka tabel kebenaran NOR adalah sebagai berikut:

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

atau

$\sim$	(p	$\vee$	q)
−	+	+	+
−	+	+	−
−	−	+	+
+	−	−	−

1.2.5 Not And (NAND)

NAND adalah pernyataan gabungan yang dihasilkan dari menegasikan konjungsi.

Notasi NAND:

$$\sim (p \wedge q), (p \wedge q)'$$

Karena NAND negasi dari konjungsi, maka tabel kebenaran NAND adalah sebagai berikut:

p	q	$p \wedge q$	$\sim (p \wedge q)$
+	+	+	−
+	−	−	+
−	+	−	+
−	−	−	+

atau

$\sim$	(p	$\wedge$	q)
−	+	+	+
+	+	−	−
+	−	−	+
+	−	−	−

1.2.6 Exclusive or (exor)

Exor adalah pernyataan gabungan dimana salah satu p atau q (tidak kedua-duanya) adalah benar

Notasi exor:

$$p \veebar q$$

Contoh:

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

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

r = sistem bilangan desimal adalah sistem bilangan yang digunakan dalam system digital adalah pernyataan yang salah.

s = aljabar linear adalah alat matematika dasar untuk disain logika adalah pernyataan salah.

Maka:

$p \veebar q$ adalah exor yang salah karena p benar, q benar.
$p \veebar r$ adalah exor yang benar karena p benar, r salah.
$s \veebar q$ adalah exor yang benar karena q benar, s salah.
$r \veebar s$ adalah exor yang salah karena r salah, s salah.

dengan demikian tabel kebenaran exor dapat ditulis sebagai berikut:

p	q	$p \veebar q$
+	+	–
+	–	+
–	+	+
–	–	–

atau

p	$\veebar$	q
+	–	+
+	+	–
–	+	+
–	–	–

1.2.7 Exclusive NOR (ExNOR)

EXNOR adalah pernyataan gabungan **ingkaran** dari EXOR di mana nilai kebenarannya benar bila kedua pernyataannya benar atau salah.

Notasi EXNOR:

$$\sim (p \veebar q)$$

Contoh:

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

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

r = sistem bilangan desimal adalah sistem bilangan yang digunakan dalam sistem digital adalah pernyataan yang salah

s = aljabar linear adalah alat matematika dasar untuk disain logika adalah pernyataan salah.

Maka:

p EXNOR q, adalah pernyataan yang benar
p EXNOR r, adalah pernyataan yang salah
s EXNOR q, adalah pernyataan yang salah
r EXNOR s, adalah pernyataan yang benar

Dengan demikian tabel kebenaran EXNOR:

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

1.3 TAUTOLOGI DAN KONTRADIKSI

Proposisi dipandang dari nilai kebenarannya dapat digolongkan menjadi 2 yaitu

1.3.1 Tautologi

Tautologi adalah proposisi yang selalu benar apapun pernyataannya.

Notasi tautologi:

$$p \vee \sim p$$

Contoh:

p = Harddisk adalah alat yang menentukan kecepatan kerja komputer adalah pernyataan salah

~p = adalah salah bahwa harddisk adalah alat yang menentukan kecepatan kerja komputer adalah pernyataan benar.

Maka

$p \vee \sim p$ adalah proposisi yang benar

Tabel kebenaran tautologi:

p	~ q	p∨ ~ p
+	–	+
–	+	+

atau

p	∨	~ p
+	+	–
–	+	+

1.3.2 Kontradiksi

Kontradiksi adalah proposisi yang selalu salah apapun pernyataannya

Notasi kontradiksi:

$$p \wedge \sim p$$

Contoh:

p = Harddisk adalah alat yang menentukan kecepatan kerja komputer adalah pernyataan salah

~p = adalah salah bahwa harddisk adalah alat yang menentukan kecepatan kerja komputer adalah pernyataan benar.

Maka

$p \wedge \sim p$ adalah proposisi yang salah

Tabel kebenaran kontradiksi:

p	~ p	$p \wedge \sim p$
+	–	–
–	+	–

1.4 KESETARAAN LOGIS

Dua buah pernyataan yang berbeda dikatakan setara bila nilai kebenarannya sama

Contoh:

1. Tidak benar, bahwa aljabar linear adalah alat matematika dasar untuk disain logika adalah pernyataan benar.
2. Aljabar Boole adalah alat matematika dasar untuk disain logika adalah pernyataan benar.

Kedua pernyataan di atas mempunyai nilai kebenaran yang sama. Jadi kedua pernyataan di atas setara/ekivalen.

Akibatnya dua proposisi P(p, q, r, ...) dan Q(p, q, r, ...) dapat dikatakan setara jika memiliki tabel kebenaran yang sama. Dua buah proposisi yang setara dapat dinyatakan dengan P(p, q, r, ...) ≡ Q(p, q, r, ...).

Contoh:

Selidiki apakah kedua proposisi di bawah setara:

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

Kedua proposisi di atas dapat dituliskan dengan notasi sbb:

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

sehingga tabel kebenarannya sebagai berikut:

p	q	~p	~q	$(p \vee q)$	$\sim (p \vee q)$	$\sim p \vee \sim q$
+	+	–	–	+	–	–
+	–	–	+	+	–	–
–	+	+	–	+	–	–
–	–	+	+	–	+	+

Jadi, kedua proposisi tersebut setara atau $\sim (p \vee q) \equiv \sim p \wedge \sim q$

1.5 ALJABAR PROPOSISI

Aljabar proposisi merupakan penerapan hukum-hukum aljabar dalam logika proposisi.

Hukum-hukum tersebut adalah:

1. Idempoten

   $p \vee p \equiv p$

   $p \wedge p \equiv p$

2. Asosiatif

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

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

3. Komutatif

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

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

4. Distribusi

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

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

5. Identitas

$p \vee f \equiv p$ $\quad$ $p \wedge f \equiv f$

$p \vee t \equiv t$ $\quad$ $p \wedge t \equiv p$

6. Komplemen

$p \vee \sim p = t$ $\quad$ $\sim t = f$

$p \wedge \sim p = f$ $\quad$ $\sim f = t$

7. Involution

$\sim p(\sim p) \equiv p$

8. De Morgan's

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

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

9. Absorbsi

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

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

10. Implikasi

$p \rightarrow q = \sim p \vee q$

11. Biimplikasi

$p \leftrightarrow q = (p \rightarrow q) \wedge (q \rightarrow p)$

12. Kontraposisi
    $p \to q = \sim q \to \sim p$

Salah satu manfaat hukum-hukum aljabar proposisi adalah untuk menyederhanakan pernyataan gabungan.

Contoh:

Sederhanakan proposisi di bawah (buktikan hukum. Absorbsi):

$$\begin{aligned} p \wedge (p \vee q) &\equiv (p \vee f) \wedge (p \vee q) \\ &\equiv p \vee (f \wedge q) \\ &\equiv p \vee f \\ &\equiv p \end{aligned}$$

1.6 IMPLIKASI DAN BIIMPLIKASI

1.6.1 Implikasi

Perhatikan pernyataan berikut: jika memakai Microsoft Word maka Windows adalah sistem operasinya.

Microsoft Word merupakan syarat cukup bagi Windows, sedangkan Windows merupakan syarat perlu bagi Microsoft Word, artinya Microsoft Word tidak dapat digunakan tanpa windows tetapi Windows dapat digunakan tanpa Microsoft Word.

Contoh pernyataan di atas disebut pernyataan bersyarat atau conditional statement.

Notasi implikasi:

$p \to q$

dibaca: jika p maka q

1.6.1.1 Kebenaran implikasi

1. Jika Microsoft Word maka Windows sistem operasinya adalah implikasi benar, karena keduanya buatan Microsoft.

Mengacu pada implikasi di atas maka:

2. Jika Microsoft Word maka bukan Windows sistem operasinya adalah pernyataan salah, karena sistem operasi Microsoft Word adalah Windows
3. Jika bukan Microsoft Word maka Windows sistem operasinya adalah pernyataan benar karena aplikasi under Windows tidak hanya Microsoft Word
4. Jika bukan Microsoft word maka bukan windows sistem operasi-nya adalah pernyataan benar, karena aplikasi selain Microsoft Word, sistem operasinya bisa jadi bukan Windows.

Tabel kebenaran implikasi sebagai berikut:

p	q	$p \to q$
+	+	+
+	–	–
–	+	+
–	–	+

Contoh:

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

$$(p \vee q) \to \overline{r}$$

Jawab:

Proposisi di atas dapat diubah menjadi

$$(t \vee f) \to f$$
$$t \to f$$
$$f$$

Jadi proposisi di atas salah

Bukti dengan tabel :

p	$\vee$	q	$\rightarrow$	$\bar{r}$	r
+		+		−	+
+		+		+	−
+	+	−	Δ	−	+
+		−		+	−

1.6.1.2 Konvers, Invers, dan Kontraposisi

Jika implikasi: $p \rightarrow q$

Maka: Konversnya : $q \rightarrow p$

Inversnya : $\sim p \rightarrow \sim q$

Kontrapositipnya : $\sim q \rightarrow \sim p$

Contoh:

Jika Microsoft Word maka Windows sistem operasinya adalah implikasi yang benar, berdasarkan implikasi di atas maka:

Konversennya : Jika Windows sistem operasinya maka Microsoft Word aplikatifnya.

Inversenya : Jika bukan Microsoft Word maka bukan Windows sistem operasinya

Kontrapositipnya : Jika bukan windows sistem operasinya maka bukan Microsoft Word aplikatifnya

Tabel kebenaran

p	q	$\sim p$	$\sim q$	$p \rightarrow q$	$\sim q \rightarrow \sim p$	$q \rightarrow p$	$\sim p \rightarrow \sim q$
+	+	−	−	+	+	+	+
+	−	−	+	−	−	+	+
−	+	+	−	+	+	−	−
−	−	+	+	+	+	+	+

setara setara

Jadi dapat disimpulkan bahwa proposisi yang saling kontra-positif mempunyai nilai kebenaran yang sama (*ekuivalen*).

Berdasarkan sifat tersebut maka kita dapat membuktikan suatu dalil dalam bentuk implikasi melalui nilai kebenaran kontra-positipnya.

Contoh:

Buktikan bahwa:

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

dapat ditulis : x^2 = genap $\rightarrow$ x = genap

Jawab:

Kontrapositif dari implikasi di atas adalah:

Jika x bukan bilangan genap maka x^2 juga bukan bilangan genap.

dapat ditulis :

Jika x = ganjil maka x^2 = ganjil

Setiap bilangan bulat bukan genap adalah ganjil, sehingga x ganjil ditulis x = 2k + 1, k bilangan bulat, akibatnya:

$$\begin{aligned} x^2 &= (2k+1)^2 \\ &= 4k^2 + 4k + 1 \\ &= 2(2k^2 + 2k) + 1 \end{aligned}$$

Karena	k	bilangan bulat maka:
	k^2	juga bilangan bulat
	2k	juga bilangan genap
	$2k^2 + 2k$	juga bilangan genap

sehingga x^2 = bilangan ganjil, karena bilangan genap ditambah 1 sama dengan bilangan ganjil.

Jadi kontrapositipnya benar akibatnya implikasinya juga benar.

1.6.2 *Biimplikasi*

Perhatikan pernyataan berikut:

Microsoft Word jika dan hanya jika ingin membuat dokumen dengan sistem operasi Windows

Pernyataan tersebut disebut biimplikasi atau biconditional statement.

Notasi biimplikasi : $p \leftrightarrow q$
dibaca: p jika dan hanya jika q

1.6.2.1. *Kebenaran Biimplikasi*

1. Microsoft Word jika dan hanya jika ingin membuat dokumen dengan sistem operasi Windows adalah pernyataan benar

Berdasarkan biimplikasi diatas, maka:

2. Microsoft Word jika dan hanya jika tidak membuat dokumen dengan sistem operasi Windows adalah pernyataan salah
3. Bukan Microsoft Word jika dan hanya jika membuat dokumen dengan sistem operasi Windows adalah pernyataan salah
4. Bukan Microsoft Word jika dan hanya jika tidak membuat dokumen dengan sistem operasi Windows adalah pernyataan benar

Tabel kebenaran biimplikasi:

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

1.7 ARGUMENTASI

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

Notasi:

$$\begin{array}{l} P(p,q,\cdots) \\ Q(p,q,\cdots) \\ \vdots \\ \therefore C(p,q,\cdots) \end{array}$$

P, Q, …	masing-masing disebut dasar pendapat atau premis
{P, Q, … }	bersama-sama disebut hipotesa
C	adalah conclusion/kesimpulan

Contoh:

Jika rajin belajar maka lulus ujian
tidak lulus ujian

$\therefore$ tidak rajin belajar

1.7.1. Kebenaran/Validitas Argumen

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

Suatu argument dikatakan valid bila masing-masing premisnya benar dan kesimpulannya juga benar.

Contoh 1:

Jika merancang gerbang logika maka memakai sistem bilangan biner

Jika memakai sistem bilangan biner maka sistem yang dibangun digital

---

$\therefore$ Jika merancang gerbang logika maka sistem yang dibangun digital

Argumen tersebut dapat dituliskan dengan notasi sebagai berikut:

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

Sekarang perhatikan tabel kebenaran:

p	q	r	$p \rightarrow q$	$q \rightarrow r$	$p \rightarrow r$
+	+	+	+	+	+
+	+	−	+	−	−
+	−	+	−	+	+
+	−	−	−	+	−
−	+	+	+	+	+
−	+	−	+	-	+
−	−	+	+	+	+
−	−	−	+	+	+

Keterangan :

Lingkari tabel premis 1 dan tabel premis 2 yang keduanya sama dengan benar. Kemudian tandai tabel kesimpulan dengan Δ. (Kesimpulan yang sejajar dengan premis 1 dan 2 yang telah dilingkari). Perhatikan tanda yang ada di dalam Δ, ternyata semua bernilai benar.

Kesimpulan:

Argumen tersebut di atas valid, karena dengan premis yang benar semua kesimpulannya juga benar semua.

Contoh 2:

Jika merancang gerbang logika maka memakai sistem bilangan biner
Memakai sistem bilangan biner

$\therefore$ Merancang gerbang logika

Argumen di atas dapat dituliskan dengan notasi

$p \rightarrow q$	disebut premis 1
q	disebut premis 2
$\therefore p$	disebut kesimpulan

Dengan cara yang sama kita dapat menentukan nilai kebenaran argumen di atas.

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

Kesimpulan:

Argumen di atas tidak valid karena dengan premis-premis benar, kesimpulannya bisa benar, bisa salah.

1.7.2 Bentuk-bentuk Dasar Menarik Kesimpulan

1. Conjunction

$$\begin{array}{l} p \\ \underline{q \quad\quad} \\ \therefore p \wedge q \end{array}$$

2. Addition

$$\frac{p}{\therefore p \vee q}$$

3. Modus Ponens

$$\begin{array}{l} p \rightarrow q \\ \underline{p \qquad} \\ \therefore q \end{array}$$

4. Constructive Dilemma

$$\begin{array}{l} (p \rightarrow q) \wedge (r \rightarrow s) \\ \underline{p \vee r \qquad\qquad\quad} \\ \therefore q \vee s \end{array}$$

5. Hypothetical syllogism

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

6. Simplification

$$\frac{p \wedge q}{\therefore p}$$

7. Disjunctive syllogism

$$\begin{array}{l} p \vee q \\ \underline{\sim p \qquad} \\ \therefore q \end{array}$$

8. Modus Tollens

$$\begin{array}{l} p \rightarrow q \\ \underline{\qquad \sim q} \\ \therefore \quad \sim p \end{array}$$

9. Destructive Dilemma

$$\begin{array}{l} (p \to q) \wedge (r \to s) \\ \sim q \vee \sim s \\ \hline \therefore \quad \sim p \vee \sim r \end{array}$$

10. Absorption

$$\begin{array}{l} \quad p \to q \\ \hline \therefore p \to (p \wedge q) \end{array}$$

Contoh pemanfaatan:

Buatlah kesimpulan dari argumen di bawah sehingga argumen tersebut valid

1. Jika hasilnya akurat maka sistemnya digital
2. Jika sistem digital maka rancangan jaringannya kombinasi
3. Jika sistem digital maka menggunakan dua nilai tanda bilangan biner
4. Hasil akurat

∴ ?

Jawab:

$$\begin{array}{ll} \text{Premis 1 :} & p \to q \\ \text{Premis 2 :} & q \to r \\ \text{Premis 3 :} & q \to s \\ \text{Premis 4 :} & p \\ \hline & \therefore ? \end{array}$$

Dengan Hypothetical Syllogism

$$\begin{array}{l} \quad p \to q \\ \quad q \to r \\ \hline \therefore p \to r \end{array} \qquad\qquad \begin{array}{l} \quad p \to q \\ \quad q \to s \\ \hline \therefore p \to s \end{array}$$

Sehingga argumentasi dapat ditulis ulang

$$\begin{array}{l} p \to r \\ p \to s \\ p \\ \hline \therefore ? \end{array}$$

Dengan Modus Ponen

$$\begin{array}{l} p \to r \\ p \\ \hline \therefore r \end{array} \qquad \begin{array}{l} p \to s \\ p \\ \hline \therefore s \end{array}$$

Sehingga argumentasi dapat ditulis ulang

$$\begin{array}{l} r \\ s \\ \hline \therefore ? \end{array}$$

Dengan conjuntion kesimpulannya dapat ditulis $r \wedge s$, sehingga argumentasi menjadi

$$\begin{array}{l} r \\ s \\ \hline \therefore r \wedge s \end{array}$$

adalah valid

Bukti dengan tabel kebenaran

p	q	r	s	$p \rightarrow q$	$q \rightarrow r$	$q \rightarrow s$	$r \wedge s$
+	+	+	+	+	+	+	+
+	+	+	–	+	+	–	–
+	+	–	+	+	–	+	–
+	+	–	–	+	–	–	–
+	–	+	+	–	+	+	+
+	–	+	–	–	+	+	–
+	–	–	+	–	+	+	–
+	–	–	–	–	+	+	–
–	+	+	+	+	+	+	+
–	+	+	–	+	+	–	–
–	+	–	+	+	–	+	–
–	+	–	–	+	–	–	–
–	–	+	+	+	+	+	+
–	–	+	–	+	+	+	–
–	–	–	+	+	+	+	–
–	–	–	–	+	+	+	–
4				1	2	3	$\therefore$

$\therefore$ argumen di atas valid

1.8 KUANTOR PERNYATAAN

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

Contoh:

Misalkan P(x) adalah pernyataan dengan x adalah sebuah bilangan genap bulat.

Misalkan D = himpunan bilangan bulat positip

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)
dan seterusnya.

Jadi dapat kita lihat ada sejumlah (kuantitas) proposisis yang benar. Untuk menyatakan kuantitas suatu objek dalam proposisi tersebut digunakan notasi-notasi yang disebut kuantor.

1.8.1 Macam-macam Kuantor

Macam-macam kuantor yang sering digunakan dalam proposisi:

1. Untuk setiap x, P(x)
   disebut kuantor universal
   simbol yang digunakan $\forall$
2. Untuk beberapa (paling sedikit satu) $x, P(x)$
   disebut kuantor existensial
   simbol yang digunakan $\exists$

Contoh

Misalkan x himpunan warga negara Indonesia, P predikat membayar pajak, R predikat membeli printer,

Maka

1. $\forall x P(x)$, artinya: Semua warga negara membayar pajak
2. $\exists x R(x) P(x)$, artinya: Ada beberapa warga negara pembeli printer membayar pajak
3. $\forall x R(x) \rightarrow P(x)$, artinya: Setiap warga negara jika membeli printer maka membayar pajak

4. $\exists x R(x) \wedge \overline{P}(x)$, artinya: Ada warga negara membeli printer dan tidak membayar pajak

1.8.2 Negasi Kuantor

$$\sim \forall x = \exists x$$
$$\sim \exists x = \forall x$$

maka :

$$\sim (\forall x P(x)) = \exists x \overline{P(x)}$$
$$\sim (\exists x P(x)) = \forall x \overline{P(x)}$$
$$\sim (\forall x P(x) \rightarrow Q(x)) = \exists x \overline{(P(x) \rightarrow Q(x))}$$
$$= \exists x P(x) \wedge \overline{Q(x)}$$
$$\sim (\exists x P(x) \rightarrow Q(x)) = \forall x \overline{(P(x) \rightarrow Q(x))}$$
$$= \forall x P(x) \wedge \overline{Q(x)}$$

Soal - soal :

1. Tuliskan tabel kebenaran dari proposisi di bawah:
   (a) $\overline{p} \wedge (p \vee q)$
   (b) $\sim (p \wedge q) \vee (r \wedge \sim q)$
   (c) $(p \wedge q) \rightarrow (p \vee q)$
   (d) $\sim (p \wedge q) \vee \sim (p \leftrightarrow q)$
   (e) $\sim (p \veebar q) \leftrightarrow (p \leftrightarrow q)$
   (f) $(p \wedge \sim (\sim p \vee q)) \vee (p \wedge q)$
   (g) $\sim ((\sim p \wedge q) \vee (\sim p \wedge \sim q)) \vee (p \wedge q)$
2. Sederhanakanlah proposisi di bawah:
   (a) $(p \vee q) \wedge (\overline{p} \vee q) \wedge (p \vee \overline{q}) \wedge (\overline{p} \vee \overline{q})$
   (b) $(p \wedge \sim (p \vee \sim q)) \vee q \wedge (q \vee p)$

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

(d) $(p \vee (\sim q \rightarrow p)) \vee ((p \leftrightarrow \sim q) \rightarrow (q \wedge \sim p))$

(e) $(p \wedge (q \rightarrow \sim r)) \vee ((\sim p \vee r) \leftrightarrow \sim q)$

3. Buktikanlah bahwa proposisi $P \equiv Q$

   (a) $P \equiv p \vee \sim p$

   $Q \equiv (p \wedge q) \rightarrow (p \leftrightarrow q)$

   (b) $P \equiv p \rightarrow (p \vee q)$

   $Q \equiv (p \wedge q) \rightarrow (p \leftrightarrow q)$

   (c) $P \equiv (p \wedge q)$

   $Q \equiv (p \downarrow p) \downarrow (q \downarrow q)$

   (d) $P \equiv (p \vee q)$

   $Q \equiv (p \downarrow q) \downarrow (p \downarrow q)$

4. Dengan kontrapositif buktikanlah kebenaran implikasi di bawah:

   (a) Jika hasil kali 2 bilangan adalah ganjil, maka kedua bilangan tersebut adalah ganjil

   (b) Jika x bukan bilangan bulat kalipatan 3, maka x^2 juga bukan bilangan bulat kelipatan 3

5. Selidiki validitas argumentasi di bawah :

   (a) 1. Jika microsoft word maka windows sistem operasinya

   2. Jika bukan product microsoft maka bukan windows sistem operasinya

   3. Linux

   $\therefore$ bukan microsoft word

   (b) Buat kesimpulan yang valid dari argumentasi di bawah:

1. Jika memakai sistem digital maka hasilnya akurat dan jika merancang gerbang logika harus menguasai Aljabar Boole
2. Sistim digital atau gerbang logika
3. Tidak akurat atau bukan Aljabar Boole
4. Tidak akurat

$\therefore$?

(c) 1. MsOffice mudah dipakai maka banyak pembeli dan mudah dicari
2. Karena mudah dicari dan banyak pembeli maka dibajak
3. Karena dibajak maka negara dirugikan
4. Negara tidak dirugikan

$\therefore$ bukan microsoft Office

(d)
$$p \to r$$
$$p \to q$$
$$\therefore p \to (r \wedge q)$$

(e)
$$p \to (r \vee q)$$
$$r \to \overline{q}$$
$$\therefore p \to r$$

6. Tentukan nilai kebenaran pernyataan di bawah, 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) $\forall x \forall y\, P\Big((x < y) \rightarrow \big(x^2 < y^2\big)\Big)$

$\forall x \exists y\, P\Big((x < y) \rightarrow \big(x^2 < y^2\big)\Big)$

$\exists x \forall y\, P\Big((x < y) \rightarrow \big(x^2 < y^2\big)\Big)$

$\exists x \exists y\, P\Big((x < y) \rightarrow \big(x^2 < y^2\big)\Big)$

-oo0oo-

2

TEORI HIMPUNAN

2.1 HIMPUNAN

Salah satu kemampuan yang kita kuasai setelah kita mempelajari logika proposisi adalah kemampuan untuk membedakan. Membedakan apakah tautologi, kontradiksi atau bentuk proposisi yang lain, membedakan apakah proposisi bernilai benar atau salah, membedakan apakah kuantor universal atau existential.

Untuk dapat menguasai teori himpunan, kemampuan untuk membedakan sangat diperlukan, karena himpunan merupakan kumpulan benda atau objek yang didefinisikan secara jelas. Himpunan dapat dipandang sebagai kumpulan benda-benda yang berbeda tetapi dalam satu segi dapat ditanggapi sebagai suatu kesatuan. Objek-objek ini disebut anggota atau elemen himpunan.

Notasi:

Himpunan : A, B, C, ...

Anggota himpunan : a, b, c, ...

Contoh:

Kita definisikan himpunan software under windows, maka kita menulis

A = {MsWord, MsExcel, Ms PowerPoint, ...}

atau

B = {x | x software under windows}

Cara menuliskan himpunan A disebut menulis secara tabulasi Cara menuliskan himpunan B disebut menulis secara deskripsi.

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

2.1.1 Kardinalitas

Jumlah elemen di dalam A disebut kardinal dari himpunan A.

Notasi: $n(A)$ atau $|A|$

Contoh.

$B = \{ x \mid x$ merupakan bilangan prima yang lebih kecil dari 20 },

atau $B = \{2, 3, 5, 7, 11, 13, 17, 19\}$ maka $|B| = 8$

T = {perkutut,kutilang,kenari,dara,beo}, maka $|T| = 5$

$A = \{a, \{a\}, \{\{a\}\} \}$, maka $|A| = 3$

2.1.2 Himpunan Berhingga dan Tak Berhingga

Himpunan berhingga adalah himpunan dimana jumlah anggota-nya berhingga artinya bila kita menghitung elemen-elemen yang berbeda dari himpunan ini, maka proses berhitungnya dapat selesai.

Bila tidak demikian maka himpunan tak berhingga.

A = himpunan software anti virus
A = {x|x software anti virus}
A = (Norton, McAfee, Panda, KaperSky, Norman)

Contoh:

B = himpunan bilangan asli
B = (x|x bilangan asli}
B = {1, 2, 3,......}

maka A berhingga

2.1.3 Kesamaan Dua Himpunan dan Subhimpunan

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

Contoh:

A = {WordPad, MsWord, WordPerfect, WS}
B = {WordPerfect, WS, MsWord, WordPad}

Maka

A = B

Dua himpunan A dan B dengan elemen-elemen yang berbeda dikatakan setara jika dan hanya jika jumlah anggota himpunan A sama dengan jumlah anggota himpunan B.

Contoh:

A = {MsExcel, Lotus 123}
B = {Mouse, Keyboard}

Maka

A~B

Himpunan A dikatakan sub himpunan B jika dan hanya jika semua elemen-elemen A adalah anggota himpunan B.

Contoh:

A = {Win3.1, Win3.11, Win95, Win97}
B = {Win3.1, Win3.11, Win95, Win97, Win98, Win98SE, WinME, Win2000, WinXP}

Maka

$A \subset B$

Bila tidak demikian dikatakan bukan sub himpunan.

Contoh:

A = {WinXP, Linux, Unix}
B = {Win3.1, Win3.11, Win95, Win97, Win98, Win98SE, WinME, Win2000, WinXP}
C = {monitor, printer, scanner}

Maka

$A \not\subset B$, A bukan sub himpunan B
$C \not\subset B$, C bukan sub himpunan B

2.1.4 Macam-macam Himpunan

2.1.4.1 Himpunan Kosong/Entry Set

Himpunan dengan kardinal = 0 disebut dengan himpunan kosong.

Notasi: $\varnothing, \{\ \}$

Contoh:

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

$A = \varnothing = \{\ \}$

2.1.4.2 Singleton Set

Singleton set adalah himpunan yang hanya memiliki 1 anggota

Contoh:

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

A = {touch screen}

2.1.4.3 Himpunan Semesta/Universal Set

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

Dengan kata lain himpunan semesta adalah himpunan dari semua objek yang berbeda.

Notasi: U

Contoh:

U = Semesta pembicaraan, yaitu sistem operasi produksi Microsoft

U = {Win 3.1, ..., WinXP, ...}

2.1.4.4 Himpunan Kuasa

Dari sebuah himpunan, kita dapat membuat subhimpunan subhimpunannya.

Himpunan dari semua subhimpunan yang dapat dibuat dari sebuah himpunan disebut himpunan kuasa.

Banyaknya himpunan bagian dari sebuah himpunan A adalah

2^x, x adalah banyak elemen A

Notasi: 2^A

Contoh:

A = {mouse, keyboard}

B = {monitor, printer, scanner}

Maka

$$2^A = \{A, \{mouse\}, \{keyboard\}, \varnothing\}$$
$$2^B = \{B, \{monitor\}, \{printer\}, \{scanner\}, \{monitor, printer\}, \{monitor, scanner\}, \{printer, scanner\}, \varnothing\}$$

2.2 OPERASI HIMPUNAN

2.2.1 Union/Gabungan dari 2 himpunan

Gabungan 2 himpunan A dan B adalah himpunan yang anggotanya semua anggota A atau B atau keduanya.

Notasi:

$A \cup B$

$A + B$

Contoh:

A = {mouse, keyboard}

B = {monitor, printer, scanner}

C = {mouse, keyboard, CPU, monitor}

Maka

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

2.2.2 Intersection/Irisan dari 2 Himpunan

Irisan dari 2 himpunan A dan B adalah himpunan yang anggotanya dimiliki bersama oleh himpunan A dan B.

Notasi: $A \cap B$

Contoh:

A = {mouse, keyboared, touch screen}
B = {monitor, touch screen, printer, scanner}

Maka

$A \cap B = \{\text{ tauch screen}\}$

2.2.3. Relative Complement/Selisih Antara 2 Himpunan

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

Notasi:

A – B

Contoh:

A = {SQL server, MySQL, MsAcces}

B = {MySQL, MsAcces, Oracle}

Maka:

$A - B = \{\text{SQL server}\}$

2.2.4 Komplemen dari Himpunan

Komplemen dari sebuah himpunan A adalah himpunan yang anggotanya bukan anggota A.

Dengan kata lain komplemen A adalah himpunan yang anggotanya merupakan hasil dari U – 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, ...}

2.2.5 Symmetic Difference/Beda Setangkup

Beda setangkup 2 himpunan A dan B adalah himpunan yang anggotanya merupakan anggota himpunan A atau anggota himpunan B tetapi bukan merupakan anggota kedua himpunan secara bersamaan.

Notasi:

$A \oplus B$

Contoh:

A = {Win3.1, Win3.11, Win95, Win97}
B = {Win95, Win97, Win98, Win98SE, WinME, Win2000}

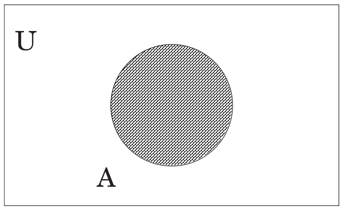
Maka

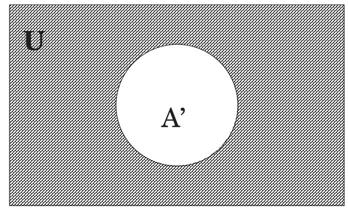
$A \oplus B$ = {Win3.1, Win3.11, Win98, Win98SE, WinME, Win2000}

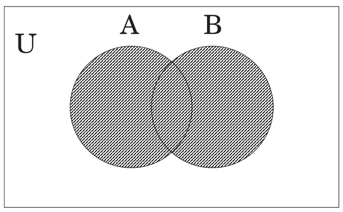
2.3 DIAGRAM VENN

Diagram venn adalah suatu cara untuk menggambarkan hubungan antara himpunan-himpunan. Dalam diagram venn himpunan biasanya dinyatakan dengan suatu daerah bidang yang dibatasi oleh sebuah lingkaran.

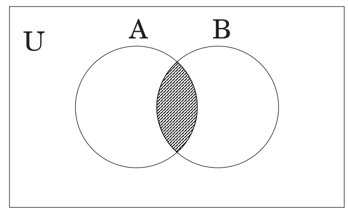
Contoh:



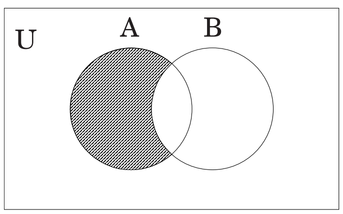




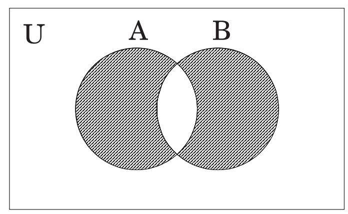
$A \cup B$



$A \cap B$



$A - B$



$A \oplus B$

2.4 HUKUM-HUKUM ALJABAR HIMPUNAN

Hukum-hukum aljabar yang berlaku pada proposisi, berlaku juga bagi himpunan, yaitu:

1. Hukum Idempoten

   $A \cup A = A$

   $A \cap A = A$

2. Hukum Asosiatif

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

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

3. Hukum komutatif

$A \cup B = B \cup A$

$A \cap B = B \cap A$

4. Hukum Distribusi

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

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

5. Hukum Identitas

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

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

6. Hukum Involution

$\left(A^C\right)^C = A$

7. Hukum Komplemen

$A \cup A^C = U$ $\quad$ $U^C = \varnothing$

$A \cap A^C = \varnothing$ $\quad$ $\varnothing^C = U$

8. Hukum DeMorgan

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

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

9. Hukum penyerapan (absorpsi):

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

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

Contoh

Sederhanakan

$A \cup (A \cap B)$

Jawab

$$
\begin{aligned}
A \cup (A \cap B) &= (A \cap U) \cup (A \cap B) \\
&= A \cap (U \cup B) \\
&= A \cap U \\
&= A
\end{aligned}
$$

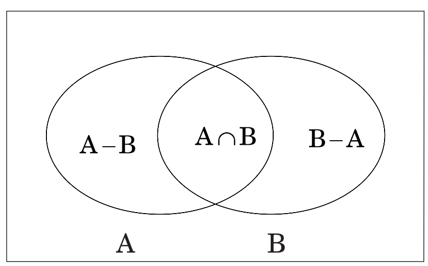
2.5 PERHITUNGAN HIMPUNAN GABUNGAN

Satu hal yang penting dalam matematika diskrit adalah proses menghitung, seperti bagaimana kita menghitung jumlah anggota dari sebuah himpunan.

Berikut adalah proses penghitungan jumlah anggota dari himpunan gabungan.

2.5.1. Gabungan dari 2 Himpunan

Jumlah angota dari 2 himpunan yang digabungkan dapat dicari sebagai berikut:



$$N_A = N_{A-B} + N_{A\cap B}$$
$$N_B = N_{B-A} + N_{A\cap B}$$
$$N_A + N_B = N_{A-B} + N_{B-A} + 2N_{A\cap B} \qquad (1)$$
$$N_{A\cup B} = N_{A-B} + N_{B-A} + N_{A\cap B} \qquad (2)$$

Substitusi (2) ke (1)

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

Sehingga

$$N_{A \cup B} = N_A + N_B - N_{A \cap B} \qquad (3)$$

2.5.2 Gabungan dari 3 Himpunan

Jumlah anggota dari 3 himpunan yang digabungkan dapat dicari sebagai berikut:

$$(A \cup B \cup C) = A \cup (B \cup C), \text{ asosiatif}$$

Substitusikan rumus (3), maka

$$N_{A \cup B \cup C} = N_A + N_{B \cup C} - N_{A \cap (B \cup C)}$$

Substitusikan rumus (3), ke $N_{B \cup C}$

$$N_{A \cup B \cup C} = N_A + N_B + N_C - N_{B \cap C} - N_{A \cap (B \cup C)} \qquad (4)$$

Hukum distribusi :

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

Hukum distribusi dan rumus (3) dapat dipakai pada suku $N_{A \cap (B \cup C)}$, karena

$$\begin{aligned} N_{A \cap (B \cup C)} &= N_{(A \cap B) \cup (A \cap C)} \\ &= N_{A \cap B} + N_{A \cap C} - N_{(A \cap B) \cap (A \cap C)} \\ &= N_{A \cap B} + N_{A \cap C} - N_{A \cap B \cap C} \end{aligned}$$

Substitusikan ke persamaan (4) diperoleh:

$$N_{A \cup B \cup C} = N_A + N_B + N_C - N_{B \cap C} - N_{A \cap B} - N_{A \cap C} + N_{A \cap B \cap C} \qquad (5)$$

SOAL-SOAL

1. Tuliskan dalam bentuk deskripsi

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

   B = {SQL Server, MySQL, Ms Access, Oracle, SAP DB, PostGre SQL, ...}

   C = {PHP, ASP, Cold Fusion, ...}

   D = {Windows, Linux, Unix, MacOS, OS/2, ...}

   E = {disket, CD-R, Hardisk, ...}

   F = {mouse, keyboard, touch screen, ...}

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

   A = {Win3.1, Win3.11, Win95, 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. $N_{A \cup B}$
   j. $N_{A \cap B}$

3. Dari 1200 mahasiswa TI diketahui

   582 menguasai Linux
   627 menguasai Windows
   543 menguasai Unix
   227 menguasai Linux dan Windows
   307 menguasai Linux dan Unix
   250 menguasai Windows dan Unix
   222 orang menguasai ketiganya.

   Berapa orang yang tidak menguasai ketiga jenis sistem operasi di atas?

   Berapa orang yang hanya menguasai Linux tetapi tidak menguasai Windows dan Unix?

4. Dari 37 orang programmer yang mengikuti wawancara untuk sebuah pekerjaan diketahui

   25 menguasai Pascal
   28 menguasai C++
   2 tidak menguasai keduannya

   Berapa orang yang menguasai keduannya?

5. Hasil survey mengenai input data dari kelas Akuntansi Komputasi diketahui

   32 orang suka memakai mouse
   20 orang suka memakai touch screen
   45 orang suka memakai keyboard
   15 orang suka mouse dan keyboard
   7 orang suka mouse dan touch screen
   10 orang suka keyboard dan touch screen
   5 orang suka memakai ketiganya

   Berapa jumlah mahasiswa yang disurvei?

   Berapa jumlah mahasiswa yang hanya suka memakai satu jenis input devices?

Berapa jumlah mahasiswa yang suka memakai keyboard dan mouse tetapi tidak suka memakai touch screen?

6. Dalam suatu kelas x semua ikut belajar pengunaan software Maple dan Matlab.
   Kalau dihitung yang belajar Maple ada 20 mahasiswa, 25% di antaranya juga belajar Matlab. Apabila diketahui perbandingan jumlah mahasiswa yang belajar Maple dan Matlab adalah 5 : 4, maka berapa jumlah mahasiswa di kelas x tersebut? Berapa jumlah mahasiswa yang hanya belajar Maple?
7. Dalam kelas x perbandingan jumlah mahasiswa yang ikut belajar penggunaan software Java, C, dan Pascal adalah 5:4:3.
   Kalau dihitung yang belajar:
   - Java ada 50 mahasiswa; 10% di antaranya juga belajar C dan Pascal sekaligus; 20% di antaranya belajar C dan 20% lagi belajar Pascal.
   - Pascal dan C tetapi tidak belajar Java 10 orang.
     Berapa jumlah mahasiswa kelas x?
     Berapa jumlah mahasiswa yang hanya belajar Pascal tetapi tidak belajar Java maupun C?

     Gambarkan dengan diagram venn!
8. Misalkan A himpunan mahasiswa tahun pertama, B himpunan mahasiswa tahun ke dua, C himpunan mahasiswa jurusan Matematika, D himpunan mahasiswa jurusan Teknik Informatika, E himpunan mahasiswa yang mengambil kuliah Matematika Diskrit, F himpunan mahasiswa yang nonton pertandingan tinju pada hari Senin malam, G himpunan mahasiswa yang belajar sampai lewat tengah malam pada hari Senin malam.

Nyatakan pernyataan bereikut dalam notasi teori Himpunan :

a. Semua mahasiswa tahun ke dua jurusan Teknik Informatika mengambil kuliah matematika Diskrit.
b. Hanya mereka yang mengambil kuliah Matematika Diskrit atau yang nonton pertandingan tinju atau yang belajar sampai lewat tengah malam pada hari Senin malam.
c. Mahasiswa yang mengambil kuliah Matematika Diskrit tidak ada yang nonton pertandingan tinju pada hari senin malam.
d. Semua mahasiswa tahun ke dua yang bukan dari jurusan Matematika ataupun jurusan Teknik Informatika pergi nonton pertandingan tinju.

9. Diantara 100 mahasiswa, 32 orang mempelajari Matematika, 20 orang mempelajari Fisika, 45 orang mempelajari Biologi, 15 orang mempelajari Matematika dan Biologi, 7 orang mempelajari Matematika dan Fisika, 10. Orang mempelajari Fisika dan Biologi, 30 orang tidak mempelajari satupun diantara ketiga bidang tersebut.
   a. Hitung banyaknya mahasiswa yang mempelejari ke 3 bidang tersebut
   b. Hitung banyaknya mahasiswa yang hanya mempelajari satu dari ke tiga bidang tersebut.
10. Survey 25 mobil baru yang dijual memiliki (A) AC, (R) Radio, (W) Power Window dengan penyebaran sebagai berikut : 15 (A), 12 (R), 11 (W), 5 (A & W), 9 (A & R), 4 (R & W), 3 (A&R&W).
    Jumlah mobil yang :
    a. Hanya ber Power Window
    b. Hanya ber AC

c. Hanya ber Radio
d. Hanya ber R dan W tetapi tidak ber A.
e. Hanya ber A dan R tetapi tidak ber W.
f. Tidak memakai ketiga-tiganya.

-oo0oo-

3

TEORI HIMPUNAN FUZZY

Teori himpunan fuzzy merupakan pengembangan teori himpunan (*crisp set*). Dalam perjalanannya perkembangan teori himpunan fuzzy dapat dibagi menjadi 3 phase, yaitu:

- Phase akademik, periode 1965-1977
- Phase tranformasi, periode 1978-1988
- Phase fuzzy boom, periode setelah, 1989

Teori himpunan fuzzy diperkenalkan oleh Prof Lotfi A. Zadeh pada tahun 1965 dan sekarang telah banyak digunakan di bidang industri dan niaga.

3.1 Fungsi Keanggotaan

Berbeda dengan teori himpunan di mana nilai keanggotaan hanya bernilai 1 atau 0, fungsi keanggotaan himpunan fuzzy ada didalam interval 0 sampai 1.

Contoh:

A = Himpunan sistem operasi yang banyak digunakan masyarakat pengguna.

Dalam teori himpunan (crisp set) himpunan A ditulis

A = {Linux, Unix, Windows, MacOs, OS2}

Artinya, Linux, Unix, Windows, MacOs, Os2 adalah anggota himpunan dengan nilai keanggotaan 1, selain kelima elemen diatas bukan anggota himpunan maka nilai keanggotaannya 0.

Dari kelima anggota himpunan A tersebut kita tidak dapat memperoleh informasi mana yang sangat banyak, banyak, cukup, kurang atau sedikit diminati oleh masyarakat pengguna, karena derajat keanggotaan kelima anggota himpunan tersebut sama.

Dalam teori himpunan fuzzy himpunan A dapat ditulis:

A = {<Linux,0.7>, <Unix,0.5>,< Windows,0.9>, <MacOs,0.2>, < OS2,0.4>}

atau

A = 0,7/Linux + 0,5/Unix + 0,9/Windows + 0,2/MacOs + 0,4/OS2

Artinya, windows paling banyak diminati oleh masyarakat pengguna karena memiliki nilai keanggotaan 0,9 disusul Linux 0,7 dan seterusnya sampai sistem operasi yang paling sedikit peminatnya yaitu MacOs dengan keanggotaan 0,2.

Notasi keanggotaan himpunan fuzzy:

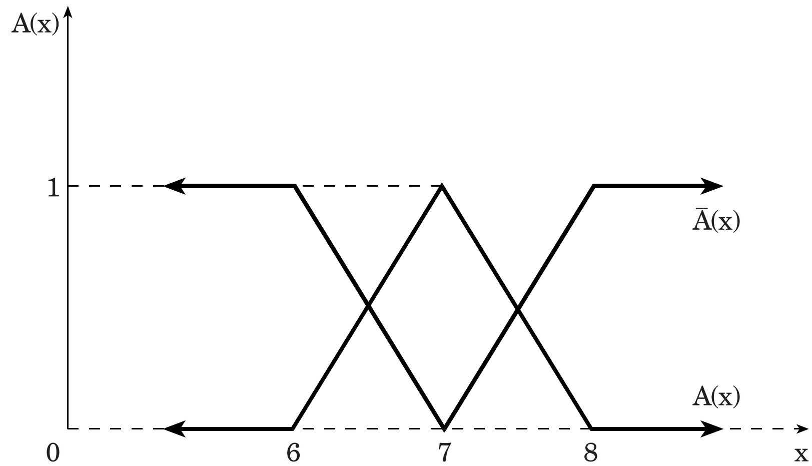
$$A : x \rightarrow [0,1]$$

Karena derajat keanggota himpunan fuzzy ada dalam interval 0 sampai 1, maka ada kalanya keanggotaan himpunan fuzzy dinyatakan dalam bentuk fungsi.

Contoh:

$$A(x) = \left\{ \begin{array}{l} x - 6 \text{ untuk } 6 \leq x \leq 7 \\ 8 - x \text{ untuk } 7 \leq x \leq 8 \\ 0 \text{ untuk } x < 6 \text{ dan } x > 8 \end{array} \right\}$$

Gambar dari fungsi keanggotaan A(x) tersebut adalah:



Gambar 3.1 Fungsi keanggotaan A (x) dan $\bar{A}$ (x)

3.2 OPERASI HIMPUNAN FUZZY

3.2.1 Komplemen

Komplemen himpunan fuzzy A adalah $\bar{A}$ dengan fungsi keanggotaan:

$$\bar{A}(x) = 1 - A(x)$$

Lihat gambar 3.1

Contoh:

$$A(x) = x - 6 \text{ untuk } 6 \le x \le 7$$

$$\begin{aligned} \bar{A}(x) &= 1 - A(x) \\ &= 1 - (x - 6) \\ &= -x + 7, \text{ untuk } 6 \le x \le 7 \end{aligned}$$

Dengan cara yang sama kita dapat mencari fungsi keanggotaan $\overline{A}(x)$ untuk A(x) = 8-x.

3.2.2 Gabungan/Union Himpunan Fuzzy

Gabungan himpunan fuzzy A dan B adalah himpunan fuzzy $A \cup B$, dengan fungsi keanggotaan

$$(A \cup B)(x) = \max\left[A(x), B(x)\right]$$

untuk semua $x \in X$.

Contoh:

Misalkan A(x) fungsi keangotaan himpunan fuzzy terbatas (finite):

A(x) = 0/5,75 + 0/6 + 0,25/6,25 + 0,5/6,5 + 0,75/6,75 + 1/7 + 0,75/7,25 + 0,5/7,5 + 0,25/7,75 + 0/8 + 0/8,25

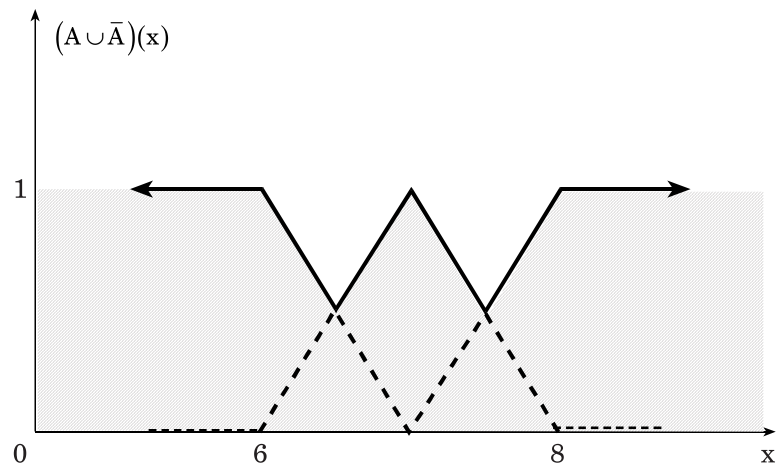
dan komplemennya adalah:

$\overline{A}(x)$ = 1/5,75 + 1/6 + 0,75/6,25 + 0,5/6,5 + 0,25/6,75 + 0,7 + 0,25/7,75 + 0,5/7,5 + 0,75/7,75 + 1/8,25

Maka:

$$(A \cup \overline{A})(x) = 1/5{,}75 + 1/6 + 0{,}75/6{,}25 + 0{,}5/6{,}75 + 0{,}75/6{,}75 + 1/7 + 0{,}75/7{,}75 + 0{,}5/7{,}5 + 0{,}75/7{,}75 + 1/8 + 1/8{,}35$$

Gambar $(A \cup \bar{A})(x)$ adalah:



Gambar fungsi keanggotaan $(A \cup \bar{A})(x)$

3.2.3 Irisan/Intersection Himpunan Fuzzy

Irisan dari himpunan fuzzy A dan B adalah himpunan fuzzy dengan fungsi keanggotaan $A \cap B$

$$(A \cap B)(x) = \min[A(x), B(x)]$$

untuk semua $x \in X$

Contoh:

Misalkan A(x) fungsi keangotaan himpunan fuzzy terbatas (finite):

A(x) = 5/5,75+0/6+0,25/6,25+0,5/6,5+0,75/6,75+1/7+0,75/7,25+0,5/7,5+0,25/7,75+0,8+0,8,25

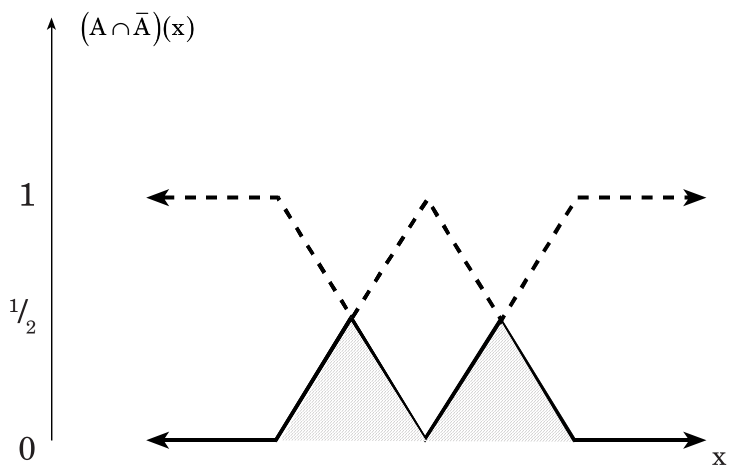
dan kpmplemennya adalah:

$\overline{A}(x)$ = 1/5.75+1/6+0,75/6,25+0,5/6,5+0,25/6,75+0/7+0,25/7,75+0,5/7,5+0,75/7,75+1/8+1/8,25

Maka

$$(A \cap \bar{A})(x) = 0/5{,}75+0/6+0{,}25/6{,}25+0{,}5/6{,}5+0{,}25/6{,}75+ 0{,}7+0{,}25/7{,}25+0{,}5/7{,}25+0{,}25/7{,}75+0{,}8+0/8{,}25$$

Gambar $(A \cap \bar{A})(x)$ adalah:



Gambar fungsi keanggotaan $(A \cap \bar{A})(x)$

3.2.4 Pemotongan/Cut Himpunan Fuzzy

Pemotongan pada sebuah himpunan fuzzy dapat dilakukan dimana saja pada selang nilai derajat keanggotaan himpunan fuzzy tersebut. Hasil pemotongan sebuah himpunan fuzzy adalah himpunan fuzzy yang memiliki derajat keanggotaan lebih besar atau sama dengan nilai potongnya

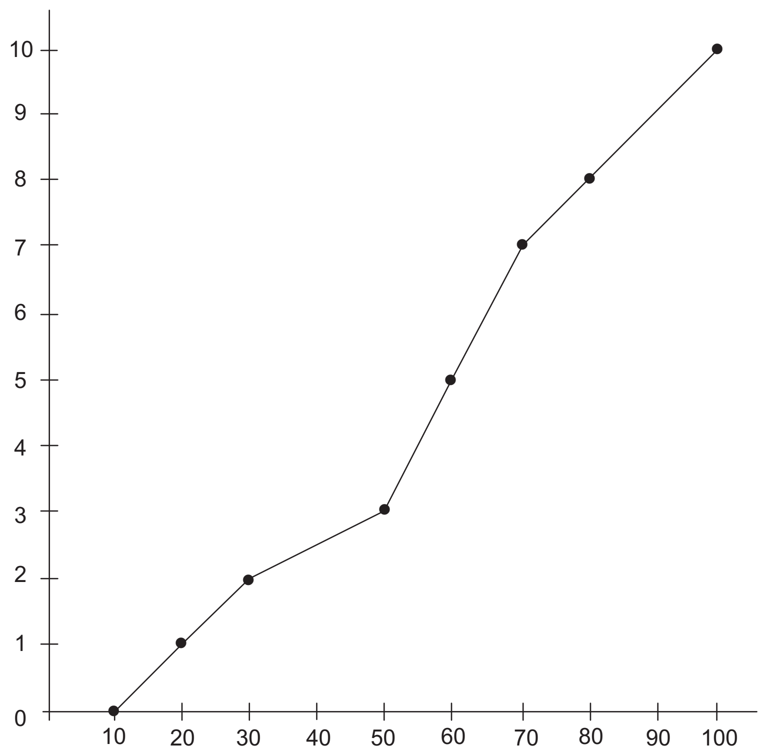
Notasi:

$$A_{\infty} = \{x \in X \mid A(x) \geq \infty\}$$

Contoh:

Himpunan fuzzy terbatas dimana sumbu Y atau A (X) dengan selang nilai 0 sampai 1 mewakili derajat keanggotaan processor : 286(10),386(20), 486(30), Pentium 1(50), Pentium 2(60), Pentium 3(70), Pentium 4(80) dan core2duo(100), serta sumbu X mewakili semesta pembicaraan yaitu harga terhadap produk yang berhubungan sebagai berikut :

A (x) = 0/10 + 0,1/20 + 0,2/30 + 0,3/50 + 0,5/60 + 0,7/70 + 0,8/80 + 1/100



Maka:

$$\begin{aligned}
A_0 &= A(x)\\
A_{0,1} &= 0{,}1/20 + 0{,}2/30 + 0{,}3/50 + 0{,}5/60 + 0{,}7/70 + 0{,}8/80 + 1/100\\
A_{0,2} &= 0{,}2/30 + 0{,}3/50 + 0{,}5/60 + 0{,}7/70 + 0{,}8/80 + 1/100\\
A_{0,3} &= 0{,}3/50 + 0{,}5/50 + 0{,}7/70 + 0{,}8/80 + 1/100\\
A_{0,5} &= 0{,}5/50 + 0{,}7/70 + 0{,}8/80 + 1/100\\
A_{0,7} &= 0{,}7/70 + 0{,}8/80 + 1/100\\
A_{0,8} &= 0{,}8/80 + 1/100\\
A_1 &= 1/100
\end{aligned}$$

Perhatikan bahwa:

$$\begin{aligned}
A_1 &= 1/100\\
A_{0,8} &= 0{,}8/8 + 1/100
\end{aligned}$$

Maka

$$A_1 \cup A_{0,8} = A_{0,8}$$

demikian juga

$$A_{0,8} \cup A_{0,5} = A_{0,5}$$

dan seterusnya sehinga dapat disimpulkan

$$A_1 \cup A_{0,8} \cup_{0,7} \cup A_{0,5} \cup A_{0,3} \cup A_{0,2} \cup A_{0,1} \cup A_0 = A(x)$$

dinotasikan:

$$\begin{aligned}
&A = \cup A_{\mu}\\
&\propto \in [0,1]
\end{aligned}$$

Bagaimana dengan irisan? Kita perhatikan:

$$
\begin{aligned}
A_0 &= A(x) \\
A_{0,1} &= 0,1/20 + 0,2/30 + 0,3/50 + 0,5/60 + 0,7/70 + 0,8/80 + 1/100 \\
A_{0,2} &= 0,2/30 + 0,3/50 + 0,5/60 + 0,7/70 + 0,8/80 + 1/100 \\
A_{0,3} &= 0,3/50 + 0,5/50 + 0,7/70 + 0,8/80 + 1/100 \\
A_{0,5} &= 0,5/50 + 0,7/70 + 0,8/80 + 1/100 \\
A_{0,7} &= 0,7/70 + 0,8/80 + 1/100 \\
A_{0,8} &= 0,8/80 + 1/100 \\
A_1 &= 1/100
\end{aligned}
$$

Maka

$$
\begin{aligned}
A_0 \cap A_{0,1} &= A_{0,1} \\
A_{0,1} \cap A_{0,2} &= A_{0,2}
\end{aligned}
$$

sehinga dapat disimpulkan

$$A_0 \cap A_{0,1} \cap A_{0,2} \cap A_{0,3} \cap A_{0,5} \cap A_{0,7} \cap A_{0,8} \cap A_1 = A_1$$

3.2.5 Pendukung (Support) Himpunan Fuzzy

Pendukung himpunan fuzzy terbatas A pada semesta pembicaraan X adalah himpunan yang terdiri dari elemen X yang derajat keanggotaannya lebih besar dari 0.

Notasi:

$$\text{Supp}(A) = \{x \in X \mid A(x) > 0\}$$

Contoh:

$A(x) = 0/5{,}75+0/6+0{,}25/6{,}25+0{,}5/6{,}5+0{,}75/6{,}75+1/7$
$+0{,}75/7{,}25+0{,}5/7{,}5+0{,}25/7{,}75+0/8$

Supp (A) = {6.25, 6.5, 6.75, 7, 7.25, 7.5, 7.75}

3.2.5.1 Inti (Core) Himpunan Fuzzy

Inti himpunan fuzzy terbatas A pada semesta pembicaraan X adalah himpunan yang terdiri dari elemen X yang derajat keanggotaanya sama dengan 1

Notasi:

$$\text{Core}(A) = \{x \in X \mid A(x) = 1\}$$

Contoh:

A(x) = 0/5,75 + 0/6 +0,25/6,25+0,5/6,5+0,75/6,75+1/7+ 0,75/7,25+0,5/7,5+0,25/7,75+0/8+0/8,25

Core (A) = {7}

3.2.5.2 Tinggi (Height) Himpunan Fuzzy

Tinggi dari himpunan fuzzy dapat dilihat dari nilai tertinggi derajat keanggotaan himpunan fuzzy tersebut.

Notasi:

h(A)

Bila h (A) = 1, maka himpunan fuzzy dikatakan normal

Bila h (A) < 1, maka himpunan fuzzy dikatakan subnormal.

Contoh:

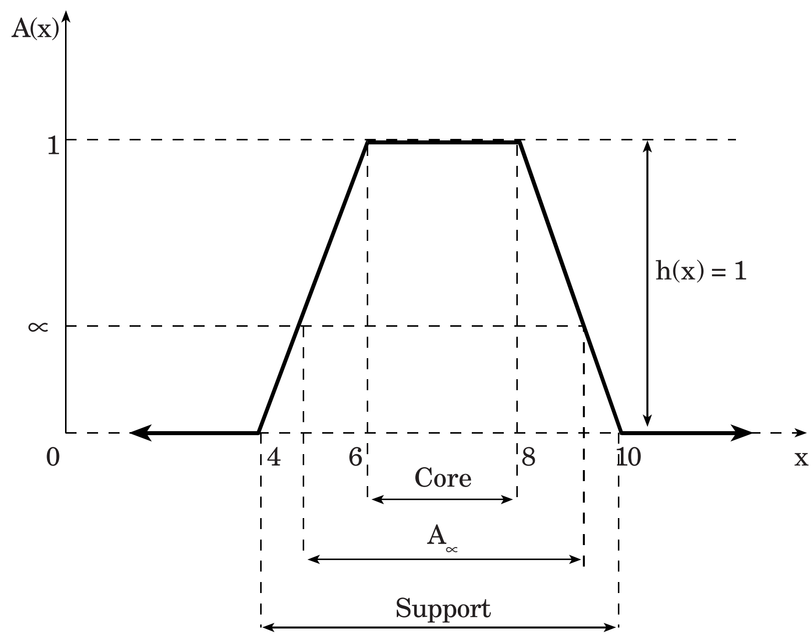
Himpunan fuzzy A pada semesta pembicaraan X dinyatakan dengan fungsi keanggotaan sebagai berikut:

$$(x) = \left\{ \begin{array}{cc} 0, & \text{untuk } x < 4 \\ 1/2\,x - 2, & \text{untuk } 4 \le x \le 6 \\ 1, & \text{untuk } 6 \le x \le 8 \\ -1/2\,x + 5 & \text{untuk } 8 \le x \le 10 \\ 0, & \text{untuk } x > 10 \end{array} \right\}$$

Gambarkan fungsi keanggotaan tersebut, kemudian tentukan Supp(A) dan h(A)

Jawab:

3.2.6 Scalar Cardinality



$$\mathrm{Supp}(A) = \{x \in X \mid 4 < A(x) < 10\}$$

$$\mathrm{Core}(A) = \{x \in X \mid 6 \le A(x) \le 8\}$$

$$h(A) = 1$$

Scalar cardinality dari sebuah himpunan fuzzy A pada semesta pembicaraan X adalah jumlah semua derajat keanggota-an elemen X dalam himpunan fuzzy A

Notasi:

$$|A| = \sum_{x \in X} A(x)$$

Contoh:

$$A_0 = A(x)$$
$$A_{0,1} = 0{,}1/20 + 0{,}2/30 + 0{,}3/50 + 0{,}5/60 + 0{,}7/70 + 0{,}8/80 + 1/100$$
$$A_{0,2} = 0{,}2/30 + 0{,}3/50 + 0{,}5/60 + 0{,}7/70 + 0{,}8/80 + 1/100$$
$$A_{0,3} = 0{,}3/50 + 0{,}5/50 + 0{,}7/70 + 0{,}8/80 + 1/100$$
$$A_{0,5} = 0{,}5/50 + 0{,}7/70 + 0{,}8/80 + 1/100$$
$$A_{0,7} = 0{,}7/70 + 0{,}8/80 + 1/100$$
$$A_{0,8} = 0{,}8/80 + 1/100$$
$$A_1 = 1/100$$

Maka:

$$|A(x)| = 0{,}1 + 0{,}2 + 0{,}3 + 0{,}5 + 0{,}7 + 0{,}8 + 1 = 3{,}6$$

3.3 Kesamaan dan Himpunan Bagian

Himpunan fuzzy A dikatakan sama dengan himpunan fuzzy B (A = B) jika dan hanya jika

$$A(x) = B(x)$$

untuk setiap $x \in X$

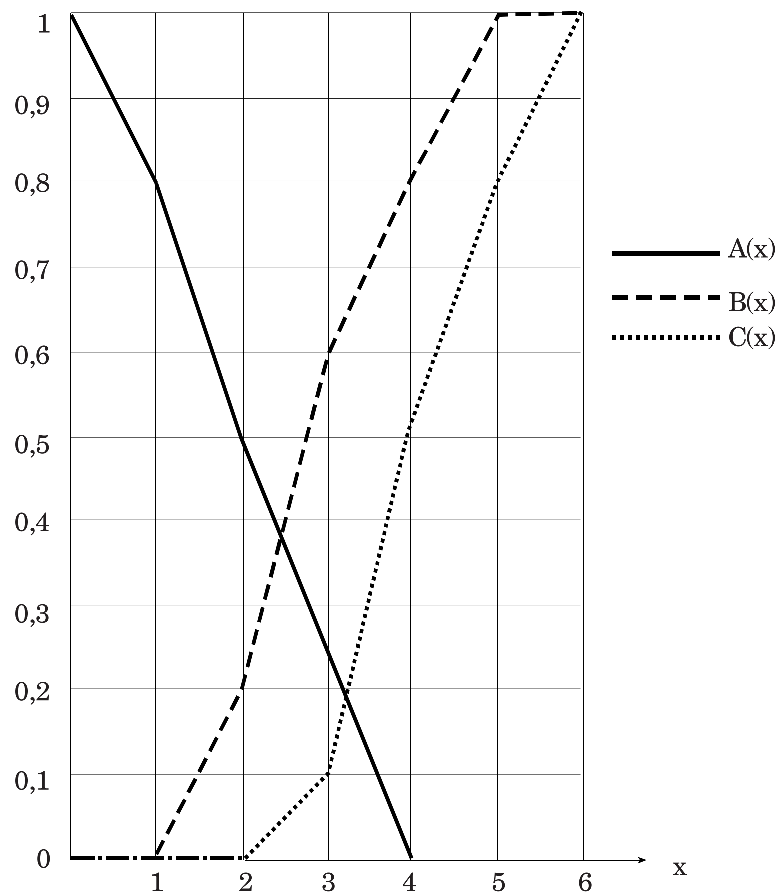
Himpunan fuzzy A dikatakan himpunan bagian dari himpunan fuzzy B, jika dan hanya jika

$$A(x) \leq B(x)$$

untuk setiap $x \in X$

Contoh:

Himpunan fuzzy A (masyarakat berpendidikan rendah), himpunan fuzzy B (masyarakat berpendidikan sedang), himpunan fuzzy C (masyarakat berpendidikan tingi) digambarkan dengan grafik tingkat pendidikan bawah.



Misalkan semesta pembicaraan X adalah

$$x = \{0, 1, 2, 3, 4, 5, 6\} \quad \text{atau}$$
$$x = \{SD, SMP, SMA, D_3, S_1, S_2, S_3\}$$

Maka dapat dibuat tabel dari ketiga himpunan fuzzy A(x), B(x), dan C(x), sebagai berikut:

Pendidikan	A(x)	B(x)	C(x)
0 = SD	1	0	0
1 = SMP	0,8	0	0
2 = SMA	0,5	0,2	0
3 = D3	0,25	0,6	0,1
4 = S1	0	0,8	0,5
5 = S3	0	1	0,8
6 = S3	0	1	1

Jadi:

$$A \neq B \neq C$$

karena derajat keangotaannya tidak sama untuk setiap elemen x

$$C(x) \subset B(x)$$

karena derajat keanggotaan $C(x) \leq B(x)$ untuk semua elemen x.

SOAL-SOAL

1. a. Gambarkanlah himpunan fuzzy yang fungsi keanggotaannya digambarkan oleh:

$$A(x) = \begin{cases} a\left(1 - \dfrac{|x-b|}{c}\right), & \text{untuk } b-c \leq x \leq b+c \\ 0, & \text{untuk } x < b-c \text{ dan } x > b+c \end{cases}$$

Bila a = 1;

b. Tuliskanlah himpunan fuzzy A

c. Gambar dan tuliskanlah himpunan fuzzy $\bar{A}$

d. Gambarkan dan tuliskanlah himpunan fuzzy $\bar{A} \cup A$ dan $\bar{A} \cap A$

2. a. Kerjakan separti nomer 1a bila fungsi keanggotaanya

$$A(x) = \begin{cases} \dfrac{(a-x)e}{a-b}, & \text{untuk } a \le x \le b \\ e, & \text{untuk } b \le x \le c \\ \dfrac{(d-x)e}{d-c}, & \text{untuk } c \le x \le d \\ 0, & \text{untuk } x < a \text{ dan } x > d \end{cases}$$

Bila e = 0,5;

b. Kerjakan seperti 1b
c. Kerjakan seperti 1c
d. Kerjakan seperti 1d
e. Tuliskan $\mathrm{Supp}(A \cup \bar{A})$, $\mathrm{Core}(A \cup \bar{A})$, dan $h(A \cup \bar{A})$
f. Tunjukan bahwa $h(A \cap \bar{A})$ tidak pernah lebih besar dari 0,5 dan $h(A \cup \bar{A})$ selalu $\ge 0{,}5$

3. Misalkan himpunan fuzzy A dan B didefinisikan oleh fungsi keanggotaan

$$A(x) = \begin{cases} 1 - \dfrac{|x-6|}{4}, & \text{untuk } 2 \le x \le 10 \\ 0, & \text{untuk } x < 2 \text{ dan } x > 10 \end{cases}$$

$$B(x) = \begin{cases} 1 - \dfrac{|x-8|}{4}, & \text{untuk } 4 \le x \le 12 \\ 0, & \text{untuk } x < 4 \text{ dan } x > 12 \end{cases}$$

a. Gambarkan himpunan fuzzy A dan B
b. Tuliskan himpunan fuzzy A dan B
c. Gambar dan tiliskan himpunan fuzzy $\bar{A}$ dan $\bar{B}$
d. Gambar dan tuliskan himpunan fuzzy $A \cup B$, $A \cap B$, $\bar{A} \cup \bar{B}$, dan $\bar{A} \cap \bar{B}$
e. Tuliskan supp dari $A \cup B, A \cap B, \bar{A} \cup \bar{B}, \bar{A} \cap \bar{B}$

f. Tuliskan core dari $A \cup B, A \cap B, \bar{A} \cup \bar{B}, \bar{A} \cap \bar{B}$

g. Tuliskan height dari $A \cup B, A \cap B, \bar{A} \cup \bar{B}, \bar{A} \cap \bar{B}$

4. Kerjakan seperti nomer 3 bila:

$$A(x) = \begin{cases} 1 - \dfrac{|x-4|}{2}, & \text{untuk } 2 \le x \le 6 \\ 0, & \text{untuk } x < 2 \text{ dan } x > 6 \end{cases}$$

$$B(x) = \begin{cases} x/4, & \text{untuk } 0 \le x \le 2 \\ 1/2, & \text{untuk } 2 \le x \le 6 \\ -1/4\,x + 2, & \text{untuk } 6 \le x \le 8 \\ 0, & \text{untuk } x < 0 \text{ dan } x > 8 \end{cases}$$

5. Hitung scalar cardinality untuk masing - masing himpunan fuzzy di bawah

(a) A = 0,4/v + 0,2/w + 0,5/x + 0,4/y + 1/z

(b) $B = \dfrac{x}{x+1}$, untuk $x \in X$, X = { 0,1,...,10 }

(c) $C = 1 - \dfrac{1}{x}$, untuk $x \in X$, , X = {1,...,10 }

6. Fuzzy set A, B dan C dinyatakan oleh fungsi keanggotaan

$$A(x) = \frac{x}{x+2}\,;\ B(x) = 2^{-x}\,;\ C(x) = \frac{1}{1+10(x-2)}$$

Bila X himpunan bilangan Real dengan interval X = [0, 10]

a) gambarkan graph fungsi keanggotaan A (x), B (x) dan C (x)

b) gambarkan graph fungsi keanggotaan A (x), B (x) dan C (x) kemudian tuliskan formulasi matematikanya.

c) Seperti nomor (b) untuk $A \cup B, A \cup C, B \cup C$

d) Seperti nomor (b) untuk $A \cap B, A \cap C, B \cap C$

e) Seperti nomor (b) untuk $A \cup B \cup C, A \cap B \cap c$

7. Himpunan Fuzzy A, B dan C dimana X = [0,80] dinyatakan oleh fungsi keanggotaan

$$A(x) = \begin{Bmatrix} 1 & \text{untuk } x < 20 \\ (35 - x)/15 & \text{untuk } 20 \le x \le 35 \\ 0 & \text{untuk } x > 35 \end{Bmatrix}$$

$$B(x) = \begin{Bmatrix} 0 & \text{untuk } x < 20 \text{ atau } x > 60 \\ (x - 20)/15 & \text{untuk } 20 \le z \le 35 \\ (60 - x)/15 & \text{untuk } 45 \le x \le 60 \\ 1 & \text{untuk } 35 < x < 45 \end{Bmatrix}$$

$$A(x) = \begin{Bmatrix} 0 & \text{untuk } x < 45 \\ (x - 45)/15 & \text{untuk } 45 \le x \le 60 \\ 1 & \text{untuk } x > 60 \end{Bmatrix}$$

Kerjakan seperti no 6

-oo0oo-

4

LOGIKA FUZZY

4.1 Pengantar

Istilah "Logika fuzzy" dalam berbagai literatur telah digunakan dalam dua pengertian yang berbeda. Dalam pengertian yang luas, logika fuzzy dipandang sebagai suatu sistim konsep, prinsip, dan metoda dalam hubungan dengan cara memberi alasan yang mendekati nilai sebenarnya. Dalam pengertian yang sederhana, dipandang sebagai logika dengan nilai kebenaran beragam dan ada dalam interval antara 0 dan 1.

Dalam pengertian luas, logika fuzzy adalah suatu wilayah aplikasi dalam teori himpunan fuzzy, dimana penggunaan konsep, prinsip dan metode yang dikembangkan dalam teori himpunan fuzzy digunakan untuk merumuskan berbagai format yang mendekati, dalam mengambil keputusan. Dalam susunan yang memanfaatkan seperangkat teori himpunan fuzzy untuk mendapatkan keputusan, sangat penting untuk menetapkan suatu hubungan antara derajat keanggotaan himpunan fuzzy dan derajat kebenaran fuzzy proposisi. Hubungan ini juga penting untuk mengembangkan konsep tambahan yang diperlukan logika fuzzy, seperti qualifier, quantifiers dan probabilitas fuzzy.

4.2. Logika dengan Nilai Kebenaran Beragam

Sebagaimana telah kita ketahui, semua proposisi pada logika klasik hanya mempunyai 2 nilai kebenaran yaitu true/benar/1 dan false/salah/0.

Logika dengan nilai kebenaran beragam adalah pengembangan dari logika klasik dengan menyisipkan nilai tengah diantara 0 dan 1, kemudian antara 0 dan ½ serta ½ dan 1, demikian seterusnya sesuai dengan situasi persoalan yang dihadapi. Maksud dari penyisipan nilai tengah ini adalah untuk memberikan nilai kebenaran sebuah proposisi yang nilai kebenarannya bisa setengah benar atau setengah salah, misalnya: "Tadi pagi terjadi tabrakan pesawat terbang di bandara Sukarno-Hatta." Adalah pernyataan yang nilai kebenarannya ½ benar atau ½ salah.

Pada buku ini penulis membatasi pembahasan hanya sampai proposisi dengan 3 nilai kebenaran yaitu:

Benar/true/1
Salah/False/0

Setengah benar atau setengah salah = ½

		Lukasiewicz				Bochvar				Kleene				Heyting				Reichenbach			
a	b	$\wedge$	$\vee$	$\rightarrow$	$\leftrightarrow$	$\wedge$	$\vee$	$\rightarrow$	$\leftrightarrow$	$\wedge$	$\vee$	$\rightarrow$	$\leftrightarrow$	$\wedge$	$\vee$	$\rightarrow$	$\leftrightarrow$	$\wedge$	$\vee$	$\rightarrow$	$\leftrightarrow$
0	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
0	½	0	½	1	½	½	½	½	½	0	½	1	½	0	½	1	0	0	½	1	½
0	1	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0
½	0	0	½	½	½	½	½	½	½	0	½	½	½	0	½	0	0	0	½	½	½
½	½	½	½	1	1	½	½	½	½	½	½	½	½	½	½	1	1	½	½	1	1
½	1	½	1	1	½	½	½	½	½	½	1	1	½	½	1	1	½	½	1	1	½
1	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0
1	½	½	1	½	½	½	½	½	½	½	1	½	½	½	1	½	½	½	1	½	½
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Dampak dari adanya nilai kebenaran ½, memunculkan 5 tabel kebenaran yang ditulis oleh para ahli yaitu Lukasiewicz, Bochvar, Kleene, Heyting, dan Reichenbach, seperti tabel diatas.

Operasi logika fuzzy Lukasiewicz:

$$\bar{a} = 1 - a$$

$$a \vee b = \max(a, b)$$

$$a \wedge b = \min(1, 1 + b - a)$$

$$a \rightarrow b = \min(1, 1 + b - a)$$

$$a \leftrightarrow b = 1 - \left| b - a \right|$$

$$a \vee b = (a \rightarrow b) \rightarrow b$$

$$a \wedge b = \overline{\bar{a} \vee \bar{b}}$$

$$a \leftrightarrow b = (a \rightarrow b) \wedge (b \rightarrow a)$$

Untuk operasi peniadaan/negasi kelima tabel kebenaran berlaku

$$\bar{a} = 1 - a$$

Logika fuzzy telah banyak diaplikasikan dalam berbagai bidang terutama computer service dan computer engineering, selain itu juga dipakai untuk membantu manusia dalam melakukan pengambilan keputusan yang tidak hanya bisa dijawab denga *Ya* atau *Tidak,* seperti pada Fuzzy Inference System, Fuzzy Clustering, Fuzzy Database, Fuzzy Linier Programming, Fuzzy Integer Transportation Problem dan sebagainya.

Berikut contoh pada Fuzzy Inference System:

Sebuah perusahaan perakit CPU mempunyai data-data permintaan, persediaan dan produksi dalam 1 bulan terakhir sebagai berikut:

* Permintaan terbesar mencapai 1000 unit per hari, terkecil mencapai 200 unit per hari.

** Persediaan di gudang terbanyak mencapai 120 unit per hari, terkecil mencapai 20 unit per hari.

*** Produksi maksimum 1400 unit per hari, minimum 400 unit per hari

Berapa CPU harus dirakit bila jumlah permintaan 800 unit dan persediaan digudang masih 60 unit, bila proses produksi mengikuti aturan fuzzy:

Jika permintaan turun dan persediaan banyak maka produksi barang berkurang.

Jawab:

Fungsi permintaan:

Permintaan maksimum = 1000 unit per hari, memiliki derajat keanggotaan 1 dan titik koordiant (1000,1).

Permintaan minimum 200 unit per hari, memiliki derajat keanggotaan 0 dan titik koordiant (200,0).

Fungsi keanggotaan fuzzy untuk permintaan naik adalah:

$$\frac{Y - Y_1}{Y_2 - Y_1} = \frac{X - X_1}{X_2 - X_1}$$

$$\frac{Y - 0}{1 - 0} = \frac{X - 200}{1000 - 200}$$

$$Y = \frac{X - 200}{800}$$

Komplemen Permintaan naik adalah permintaan turun, fungsi keanggotannya adalah

$$Y = 1 - Y = 1 - \frac{x - 200}{800} = \frac{-X + 1000}{800}$$

Dengan cara yang sama dapat dicari fungsi keanggotaan fuzzy untuk persediaan naik.

maksimum (120,1) dan minimum (20,0)

$$\frac{Y-0}{1-0} = \frac{X-20}{120-20}$$
$$Y = \frac{X-20}{100}$$

Komplemen persediaan naik adalah persediaan turun, fungsi keanggotaannya

$$Y = 1 - Y = 1 - \frac{x-20}{100} = \frac{-X+120}{100}$$

Dengan cara yang sama fungsi keanggotaan fuzzy untuk produksi naik dapat dicari:

maksimum (1400,1) dan minimum (400,0)

$$\frac{Y-0}{1-0} = \frac{X-400}{1400-400}$$
$$Y = \frac{X-400}{1000}$$

Komplemen produksi naik adalah produksi turun, dengan fungsi keanggotaan:

$$Y = 1 - Y = 1 - \frac{x-400}{1000} = \frac{-X+1400}{1000}$$

jadi bila ada permintan 800 unit dan prsediaan di gudang ada 60 unit, mengikuti aturan produksi: **Jika permintaan turun dan persediaan banyak maka produksi barang berkurang.**

Jika permintaan turun:

$$Y = \frac{-X+1000}{800}, \text{ untuk } x = 800,\ Y = 0{,}25$$

dan persediaan banyak:

$$Y = \frac{X - 20}{100}, \text{ untuk } X = 60, Y = 0{,}40$$

Karena penggabungan pernyataannya **dan**, maka operasi Fuzzy yang digunakan Intersection yaitu min (0.25, 0.40) = 0.25

Maka produksi berkurang:

$$Y = \frac{-X + 1400}{1000}$$
$$0.25 = (-X + 1400) : 1000$$
$$X = 1150 \text{ Unit}$$

Jadi, CPU yang harus diproduksi 1150 unit.

4.3 Soal-soal

1. Buat tabel kebenaran $\sim(p \wedge q) \equiv \sim p \vee \sim q$ dengan tabel kebenaran Lukasiewicz, Bochvar, Kleene, Heyting, dan Reichenbach.
2. Nyatakan modus ponen: $((p \to q) \wedge p) \wedge q$ dengan tabel kebenaran Lukasiewizc, Bochvar, Kleene, dan Heyting.
3. Tentukan tabel kebenaran proposisi di bawah dengan kelima tabel kebenaran yang telah diberikan.
   a. $p \vee \bar{p}$
   b. $p \wedge \bar{p}$
   c. $p \wedge (p \vee q)$
   d. $p \vee (p \wedge q)$
4. Tunjukkan bahwa
   a. $p \to q = 1$ jika dan hanya jika $p \leq q$
   b. $p \leftrightarrow q = 1$ jika dan hanya jika $p = q$

Berlaku untuk kelima tabel kebenaran yang telah diberikan

5. Selidiki validitas modus sillogisme dengan kelima tabel kebenaran yang telah diberikan.
6. Kerjakan seperti contoh fuzzy inference di atas (halaman 70).

   Bila produksi perusahaan tersebut menggunakan aturan fuzzy sebagai berikut:

   (a). Jika permintaan turun dan persediaan sedikit maka produksi barang berkurang

   (b). Jika permintaan naik dan persediaan banyak maka produksi barang bertambah

   (c). Jika permintaan naik dan persediaan sedikit maka produksi barang bertambah

7. Sebuah perusahaan perakitan CPU memiliki data penjualan 1 bulan terakhir sebagai berikut:

   Permintaan terbesar mencapai 500 unit/hari dan terkecil 100 unit/hari.

   Persediaan terbanyak mencapai 60 unit/hari dan terkecil 10 unit/hari.

   Produksi terbesar mencapai 700 unit/hari dan terkecil 200 unit/hari.

   Tentukan berapa unit CPU yang harus diproduksi jika ada permintaan sebanyak 400 unit dan persediaan digudang hanya ada 30 unit CPU, bila proses produksi menggunakan logika fuzzy seperti dibawah:

   a. Jika permintaan turun dan persediaan banyak maka produksi arang berkurang.
   b. Jika permintaan turun dan persediaan sedikit maka produksi barang berkurang.
   c. Jika permintaan naik dan persediaan banyak maka produksi barang bertambah.

d. Jika permintaan naik dan persediaan sedikit maka produksi barang bertambah.
e. Hitung produksi rata - rata terbobot dari ke 4 aturan produksi diatas.

Catatan rumus rata - rata terbobot:

$$Z = (\alpha_1 Z_1 + \alpha_2 Z_2 + \alpha_3 Z_3 + \alpha_4 Z_4) / (\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4)$$

-oo0oo-

5

RELASI KLASIK

5.1 Pendahuluan

Relasi Klasik (*crisp relation*) menggambarkan ada tidaknya interaksi atau koneksi antara elemen-elemen dari 2 atau lebih himpunan dalam urutan tertentu.

Contoh:

Dua orang yaitu Rosa dan Marina memiliki hubungan sebagai berikut; "Rosa adalah kakak kandung Marina" jadi relasinya adalah hubungan famili. Banyak sekali jenis relasi, tetapi ada 2 yang sering digunakan yaitu relasi; "lebih besar dari dan kurang dari".

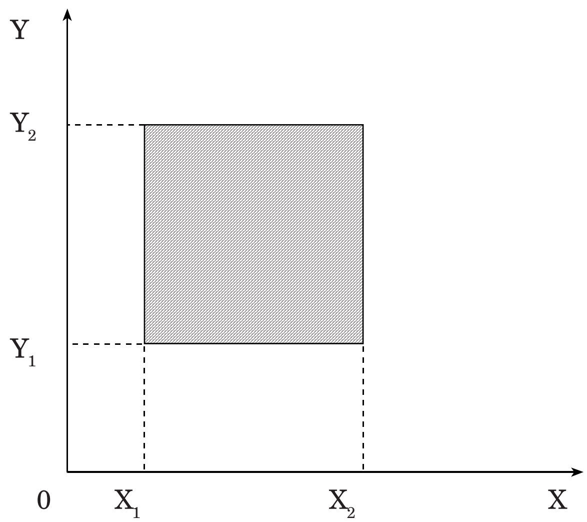
Kita dapat mendefisinikan sebuah relasi melalui perkalian skalar pada koordinat cartesian dimana sumbu x mewakili variabel x dan sumbu y mewakili variabel y. Misalnya variable x dan y adalah bilangan real dalam interval tertutup $[x_1, x_2]$ dan $[y_1, y_2]$ atau misalkan himpunan:

X = {x1,x2}

Y = {y1,y2}

Maka:

$X \times Y = \{(x_1, y_1), (x_1, y_2), (x_2, y_1), (x_2, y_2)\}$
$Y \times X = \{(y_1, x_1), (y_2, x_1), (y_1, x_2), (y_2, x_2)\}$
$X \times X = \{(x_1, x_1), (x_1, x_2), (x_2, x_1), (x_2, x_2)\}$
$Y \times Y = \{(y_1, y_1), (y_1, y_2), (y_2, y_1), (y_2, y_2)\}$



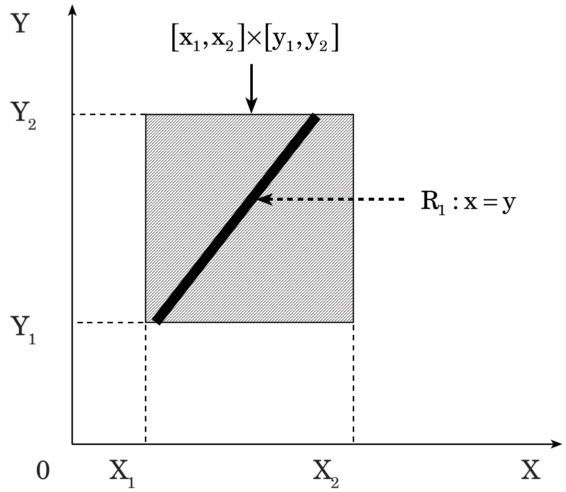
Maka relasi R antara elemen-elemen dalam himpunan X dan himpunan Y adalah:

$R \subseteq X \times Y$

Pasangan-pasangan elemen dalam R menggambarkan relasi, karena ada 2 himpunan yang terlibat dalam relasi R, maka relasi demikian disebut relasi binary.

Contoh:

Misalkan kita definisikan sebuah relasi X = Y dengan notasi R_1, maka $R_1 \subset X \times Y$ dapat kita gambarkan dalam koordinat cartesian sebagai berikut:



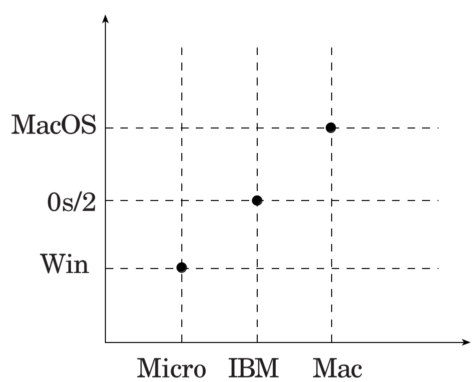
Relasi dapat melibatkan n himpunan yang disebut relasi berdimensi n. Dalam buku ini hanya dibahas relasi binary.

5.2 Pemaparan Relasi

5.2.1 Pemaparan Koordinat

Relasi dapat dipaparkan melalui sistem koordinat, sebagai contoh relasi.

R = {(Microsoft, Windows), (IBM, 0s/2), (Macintosh, MacOS)}



Tanda titik pada gambar di atas menjelaskan bahwa pasangan tersebut termasuk dalam relasi.

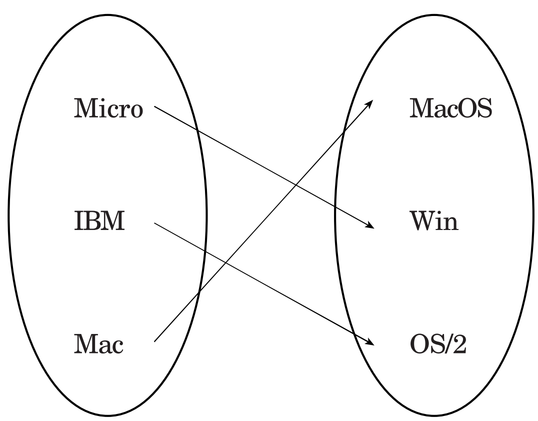
5.2.2 Pemaparan Matrik

Relasi dapat dipaparkan melalui sebuah matrik, yaitu dengan nilai 1 apabila ada relasi antara 2 elemen pasangan terurut, atau O apabila tidak ada relasi antara 2 elemen pasangan terurut.

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

5.2.3 Pemetaan

Pemetaan adalah paparan visual relasi dengan menghubungkan anggota suatu himpunan dengan anggota himpunan yang lain, sebagai contoh:

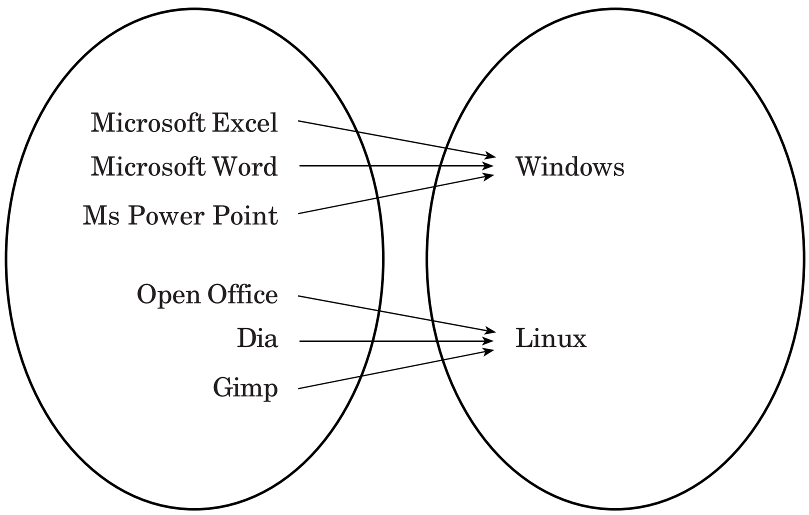


Dalam sebuah relasi, satu anggota pada himpunan pertama dapat dipetakan ke lebih dari satu anggota himpunan kedua. Relasi binary adalah relasi dimana tidak ada anggota pada himpunan pertama yang dihubungkan dengan lebih dari satu anggota pada himpunan kedua. Relasi binary disebut fungsi dengan notasi:

$R \subseteq A \times B$ atau

$R : A \to B$

Contoh:



5.2.4 Graph Berarah

Graph berarah merupakan gambaran yang paling tepat untuk relasi $R \subseteq X^2$ dengan aturan-aturan sebagai berikut:

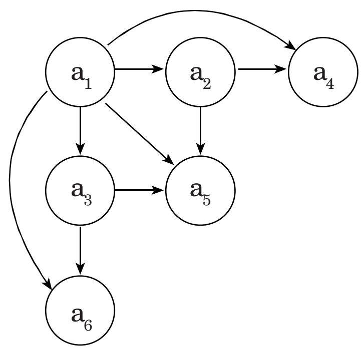
a. Setiap anggota himpunan X digambarkan dengan lingkaran
b. Garis berarah antar lingkaran menggambarkan adanya relasi antara anggota himpunan, jadi pasangan-pasangan anggota himpunan tersebut termasuk dalam relasi.

Contoh:

a_1 Prasyarat untuk semua bagian yang lain

a_3 Prasyarat untuk a_5 dan a_6

a_6 Bukan prasyarat untuk semua bagian yang lain.



5.3 OPERASI DALAM RELASI BINARY

Semua operasi dalam himpunan juga dapat diaplikasikan ke dalam relasi, namun demikian ada beberapa operasi yang tidak ada hubungannya dengan operasi dalam himpunan, seperti inverse relasi dan komposisi relasi

5.3.1 Inverse Relasi (R^{-1})

Inverse relasi R^{-1} adalah kebalikan dari relasi R, inverse relasi R, didefinisikan dengan menukar susunan anggota di semua pasangan yang ada dalam relasi, jadi

Jika: $R: X \to Y$, maka

$R^{-1}: Y \to X$

dan kebalikan dari R^{-1} adalah relasi R yang asli, yaitu

$$\left(R^{-1}\right)^{-1} = R$$

untuk semua relasi binary R.

5.3.2 Komposisi Relasi

Komposisi relasi adalah operasi mengkombinasikan 2 buah relasi binary yang cocok/sesuai dan menghasilkan sebuah relasi binary yang baru. Agar dua buah relasi dapat dikomposisikan maka relasi P dan Q didefinisikan sebagai:

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

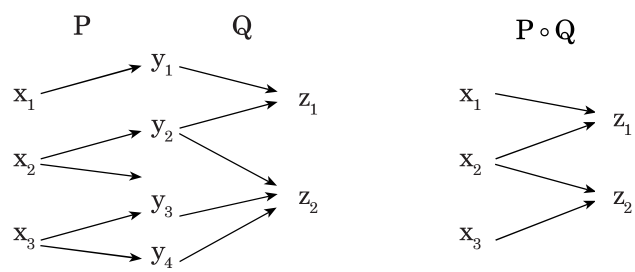
di mana Y di P harus sama dengan Y di Q.

Relasi P ke Q atau $P \circ Q$, didefinisikan sebagai relasi:

$$R : X \rightarrow Z$$

Dengan $(x, z) \in R$ jika dan hanya jika anggota y dalam himpunan Y mempunyai pasangan minimal 1 dalam himpunan P dan Q.

Contoh:



Sifat-sifat Komposisi Relasi

a. Asosiatif
   $(P \circ Q) \circ R = P \circ (Q \circ R)$
b. Tidak Komutatif
   $P \circ Q \neq Q \circ P$
c. $(P \circ Q)^{-1} = Q^{-1} \circ P^{-1}$

5.4 Ekivalen, Kompatibel dan Ordering Relasi

Ekivalen relasi, kompatibel relasi dan ordering relasi adalah tiga jenis relasi yang penting.

5.4.1 Relasi Ekivalen

Sebuah relasi binary dikatakan ekivalen bila memenuhi sifat refleksi, simetri,dan transitif.

Sebuah relasi bersifat refleksi jika dan hanya jika $(x, x) \in R$ untuk setiap $x \in X$.

Sebuah relasi bersifat simetri jika dan hanya jika untuk setiap pasangan anggota himpunan X katakanlah (x, y) adalah anggota relasi, maka (y, x) juga anggota relasi. Atau jika $(x, y) \in R$ maka $(y, x) \in R$.

Sebuah relasi bersifat transitif jika dan hanya jika untuk 3 anggota x,y,z dalam himpunan X; $(x, y) \in R$, $(y, z) \in R$ maka $(x, z) \in R$.

Contoh:

Misalkan nama mahasiswa, nilai , mata kuliah, umur ditabelkan seperti di bawah.

Tabel 5.1 Contoh Relasi Ekivalen

Nama	Nilai	Mata kuliah	Umur
Ali	B	MatDis	19
Beni	C	Met Num	19
Cica	C	Kalkulus	20
Dani	A	Kalkulus	19
Eva	A	Kalkulus	19
Fani	A	Fisika	21
Galih	B	Alin	21
Hani	C	MatDis	19
Ina	B	MatDis	19
Jono	B	Fisika	21

Karena huruf pertama nama-nama mahasiswa berlainan, maka himpunan mahasiswa dapat kita definisikan sebagai

X = {A, B, C, D, E, F, G, H, I, J}

Sekarang kita buat sebuah relasi R dari X ke X berdasarkan nilai mahasiswa

$R : X \to X$
R = {(A, A), (A, G), (A, I), (A, J), (B, B), (B, C), (B, H), (C, B), (C, C), (C, H), (D, D),(D, E), (D, F), (E, D), (E, E), (E, F), (F, D), (F, E), (F, F), (G, A), (G, G), (G, J), (G , I), (H, B), (H, C), (H, H), (I, A), (I, G), (I, I), (I, J), (J, A), (J, G), (J, I), (J, J)}

Bila relasi $R : X \to X$ kita paparkan dalam bentuk matrik:

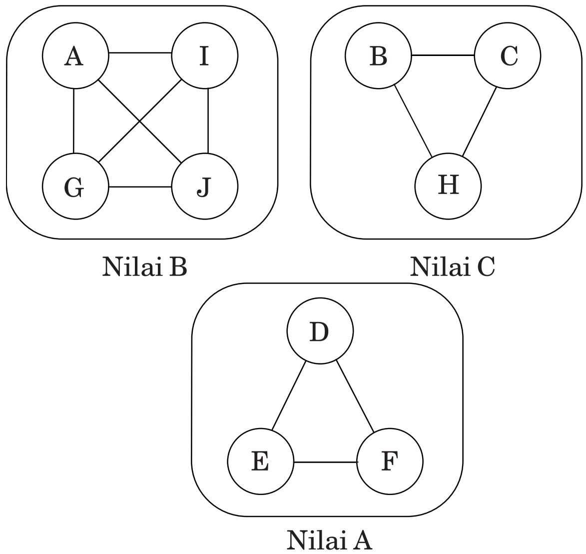
R	A	B	C	D	E	F	G	H	I	J
A	1	0	0	0	0	0	1	0	1	1
B	0	1	1	0	0	0	0	1	0	0
C	0	1	1	0	0	0	0	1	0	0
D	0	0	0	1	1	1	0	0	0	0
E	0	0	0	1	1	1	0	0	0	0
F	0	0	0	1	1	1	0	0	0	0
G	1	0	0	0	0	0	1	0	1	1
H	0	1	1	0	0	0	0	1	0	0
I	1	0	0	0	0	0	1	0	1	1
J	1	0	0	0	0	0	1	0	1	1

Perhatikan;

R refleksi karena(A,A),(B,B),..., (J,J) anggota relasi
R simetri karena (A, G), (G, A), ... semua pasangan bolak-baliknya anggota R
R transitif karena (A,G), (G,J) dan (A,J) anggota R

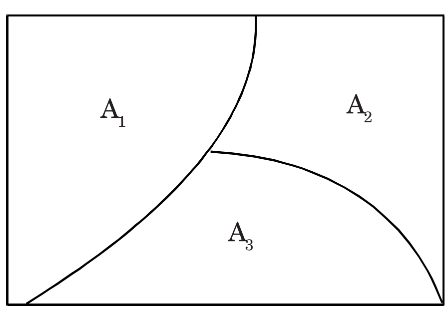
Jadi $R : X \to X$ berdasarkan nilai mahasiswa adalah relasi ekivalen.

Paparan relasi ekivalen dengan graph berarah.



Pada graph di atas setiap lingkaran mempunyai relasi dengan dirinya sendiri (refleksi) dan garis penghubung boleh tidak diberi arah, yang berarti setiap garis penghubung mempunyai arah bolak-balik. Relasi ekivalen yang kita kelompok berdasarkan nilai diatas disebut equivalen classes.

Partisi adalah himpunan bagian dari suatu himpunan dengan aturan: tidak overlap, lengkap dan bukan subhimpunan kosong. Partisi dari equivalen classes di atas adalah:



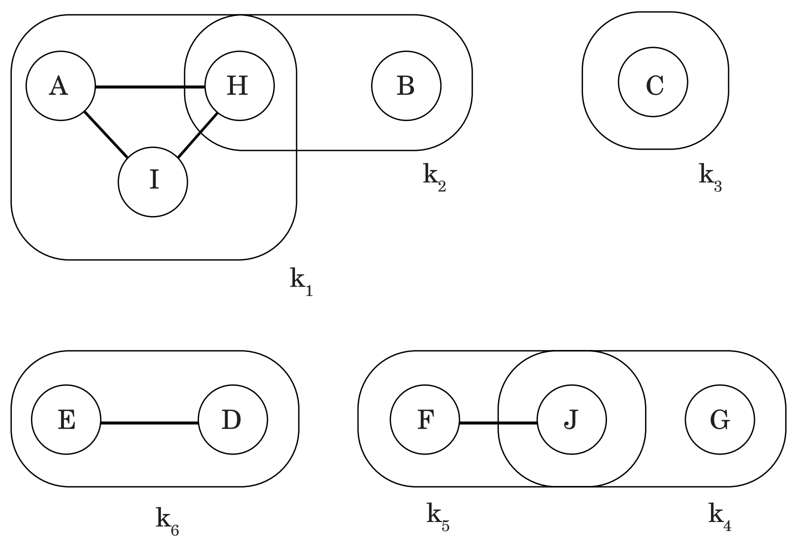
Sel A_1 adalah himpunan bagian dari himpunan relasi $R: X \rightarrow X$ dengan nilai B

Sel A_2 adalah himpunan bagian dari himpunan relasi $R: X \rightarrow X$ dengan nilai C

Sel A_3 adalah himpunan bagian dari himpunan relasi $R: X \rightarrow X$ dengan nilai A

5.4.2 Relasi Kompatibel

Sebuah relasi binary dikatakan kompatibel bila memenuhi sifat refleksi dan simetri, tetapi tidak harus transitif. Dari Tabel 5.1 kita dapat membuat relasi kompatibel sebagai berikut:



Dari contoh di atas ada enam kelompok mahasiswa dengan relasi kompatibel, yaitu:

- Ali, Hani dan Ina
- Hani dan Beni
- Cica dengan dirinya sendiri
- Galih dan Jono

- Fani dan Jono
- Dani dan Eva

5.4.3 Poset (Partially Orderet Set)

Sebuah relasi binary R pada himpunan semesta S dikatakan poset, jika relasi R tersebut bersifat: refleksi, antisimetri dan transitif.

Sebuah relasi binari bersifat anti simetri jika dan hanya jika untuk x dan y anggota himpunan X, bila $(x, y) \in R$ dan $(y, x) \in R$ maka x = y.

Partially ordered set sering dinyatakan dengan "mendahului" atau "didahului" seperti

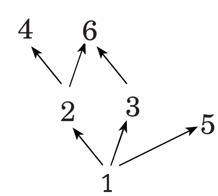
$a < b$, a mendahului b
$a \leq b$, a langsung mendahului b
$b > a$, b didahului a
$b \geq a$, b langsung didahului a
$a // b$, a tidak dapat di bandingkan dengan b

Partially ordered set sering kali dipaparkan dengan diagram Hess seperti contoh dibawah.

Misalkan relasi R adalah hubungan dalam himpunan A = {1, 2, 3, 4, 5, 6} yang didefinisikan oleh

"x membagi y"

maka R adalah sebuah orde partial dalam A yang dapat digambarkan dengan diagram Hess sebagai berikut:



Dari diagram Hess di atas dapat kita lihat bahwa:

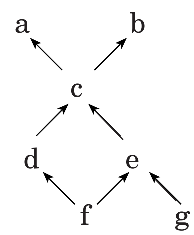
$1 < 4$, 1 mendahului 4
$1 \leq 2$, 1 langsung mendahului 2
$2 // 3$, 2 tidak dapat dibandingkan dengan 3
$4 > 1$, 4 didahului 1
$2 \geq 1$, 2 langsung didahului 1

Dalam Poset terdapat istilah-istilah yang penting seperti:

- *Upper bound* (ub) = batas atas
  adalah semua elemen himpunan diatas himpunan bagian yang akan kita cari batas atas nya, dimana setiap elemen dalam himpunan bagian itu dapat dibandingkan dengan semua elemen batas atasnya
- *Least upper bound* (lub) = supremum = batas atas terkecil
  adalah elemen dari upper bound yang paling dekat atau langsung didahului himpunan bagian yang kita cari batas atas terkecilnya
- *Lower bound* (lb) = batas bawah
  adalah semua elemen himpunan di bawah himpunan bagian yang akan kita cari batas bawah nya, dimana setiap elemen dalam himpunan bagian itu dapat dibandingkan dengan semua elemen batas bawah nya.
- *Greatest lower bound* (glb) = Infimum = batas bawah terbesar.
  adalah elemen dari lower bound yang paling dekat atau langsung mendahului himpunan bagian yang kita cari batas bawah terbesarnya.

Contoh:

Misalkan himpunan A = {a, b, c, d, e, f, g} diorder menurut diagram Hess di bawah.



Pandang sub himpunan A yaitu himpunan B = {c, d, e}

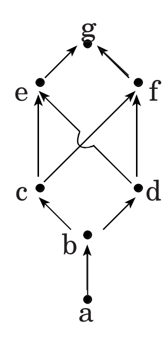
Maka

- batas atas dari B = ub (B) = a, b, c. c termasuk batas atas karena c mendominsi d dan e. c termasuk batas atas dari B karena c langsung didahului oleh d dan e
- batas bawah dari B = lb (B) = f, g bukan batas bawah dari B karena g tidak mendahului d, g dan d tidak dapat dibandingkan
- batas atas terkecil dari B adalah c karena c langsung mendahului a dan b (c mendominasi a dan b)
- batas bawah terbesar dari B = glb (B) = f

Poset dapat memiliki glb dan lub lebih dari 1 (tidak tunggal)

Contoh:

Misalkan himpunan A = {a, b, c, d, e, f, g} di order dengan diagram Hess



Pandang himpunan B = {c, d}, maka

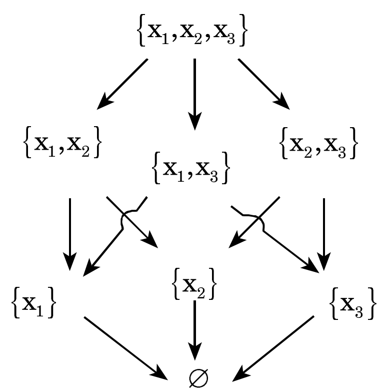
gl b (B) = b
lub (B) = e dan f

Namun demikian ada Poset khusus yang hanya boleh memiliki 1 buah (tunggal) glb dan lub,poset demikian disebut latis (Lattice). Dengan kata lain Lattice adalah poset yang setiap 2 elemennya mempunyai lub dan glb masing-masing satu buah (tunggal).

Contoh:

$A = \{x_1, x_2, x_3\}$ dan
$2^A = \{\varnothing, \{x_1\}, \{x_2\}, \{x_3\}, \{x_1, x_2\}, \{x_1, x_3\}, \{x_2, x_3\}, \{x_1, x_2, x_3\}\}$

diorder dengan diagram Hess



Kita perhatikan disini bahwa setiap 2 elemen dalam poset diatas hanya memiliki 1 lub dan 1 glb.

- lub dari $\{\{x_1\}, \{x_2\}\}$ adalah ϕ
  glb dari $\{\{x_1\}, \{x_2\}\}$ adalah $\{x_1, x_2\}$
- lub dari $\{\{x_1, x_2\}, \{x_1\}\}$ adalah $\{x_1\}$
  glb dari $\{\{x_1, x_2\}, \{x_1\}\}$ adalah $\{x_1, x_2\}$

SOAL

1. Misalkan himpunan A = {1, 2, 3} dan relasi yang ada adalah $R : A \rightarrow A$. Tentukan apakah relasi-relasi dibawah mempunyai sifat refleksi, simetri, transitif atau antisimetri.

   a) $R_1 = \{(a, b) | a < b\}$

   b) $R_2 = \{(a, b) | a \leq b\}$

   c) $R_3 = \{(a, b) | a = b\}$

   d) $R_4 = \{(a, b) | 2a = b\}$

   e) $R_5 = \{(a, b) | a = b - 1\}$

   f) $R_6 = \{(0, 0), (0, 1), (1, 0), (1, 1), (1, 2), (2, 2), (0, 2), (3, 3)\}$

   g) $R_7 = \{(a, b) | a < b \text{ atau } a > b\}$

   h) $R_8 = \{(0,0), (0,3), (1,1), (2,2), (1,0), (0,1), (3,1), (3,3),$
   $3,0), (1,3)\}$

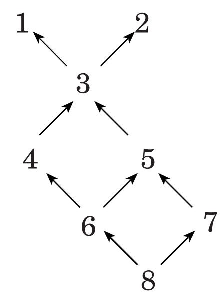
   i) $R_9 = \{(a, b) | a = b \text{ atau } a = b - 1 \text{ atau } a - 1 = b\}$

   j) $R_{10} = \{(0, 1), (1, 3), (2, 1), (3, 2)\}$

2. Tentukan R^{-1} dari masing-masing relasi pada nomer 1.
3. Carilah komposisi relasi di bawah dimana masing-masing relasinya diambil dari soal nomor 1 dan 2

   a) $R_1 \circ R_1$
   b) $R_1 \circ R_1^{-1}$
   c) $R_1 \circ R_2$
   d) $R_3 \circ R_2^{-1}$
   e) $R_3 \circ R_3$
   f) $R_4 \circ R_5$
   g) $R_6 \circ R_6$
   h) $R_6 \circ R_6^{-1}$
   i) $R_1 \circ R_7$
   j) $R_2 \circ R_7$
   k) $R_3 \circ R_7$
   l) $R_8 \circ R_8$
   m) $R_8 \circ R_8^{-1}$
   n) $R_9 \circ R_{10}$

4. Misalkan himpunan A = {1, 2, …, 8} diorder dengan diagram Hess



a) Tuliskan simbol-simbol >, ≥, //

1 … 2

1 … 5

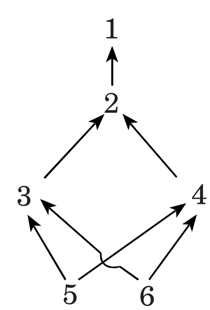
8 … 5

6 … 4

6 … 5

b) Pandang himpunan-himpunan bagian A yaitu himpunan B = {4, 5, 6}, C = {2, 3, 6} dan D = {4, 5, 7}. Tentukan masing-masing ub, lub, lb, dan glb dari himpunan B, C dan D.

5. Misalkan himpunan A = {1, 2, 3, …, 6} diorder dengan diagram Hess



pandang himpunan bagian A yaitu B = { 2,3,4 }

Tentukan:

ub, lub, lb dan glb dari B

6. Dari Tabel 5.1 selidikilah apakah $R : X \to X$ adalah relasi yang ekuivalen dipandang dari mata kuliah yang diambil, kalau ya buatlah partisinya berdasarkan kelas ekuivalennya.
7. Sama dengan Soal nomor 6, dipandang dari umur mahasiswa.
8 Relasi R adalah relasi dalam himpunaan A = { 1, 2, 3, 4, 6, 8, 9, 12, 16, 18, 24, 27, 36, 48, 54, 72, 81, 108, 144, 162, 216, 324, 432, 648, 1296 }

   yang didefinisikan oleh : " x membagi y "

   (a). Gambarkan poset diatas dengan diagram Hess.

   (b). Cari : ub, lub, lb dan glb dari himpunan-himpunan:

   B = { 8, 12, 18, 27 }

   C = { 12, 18, 36, 72, 108, 216 }

   D = { 6, 12, 18, 24, 36, 54 }

   E = { 6, 12, 36, 72 }

-oo0oo-

6

FUNGSI

6.1 DEFINISI FUNGSI

Fungsi adalah sebuah relasi binary dimana masing-masing anggota dalam himpunan A (domain) hanya mempunyai satu bayangan pada himpunan B (kodomain).

Notasi Fungsi:

$f : A \rightarrow B$

dibaca f adalah fungsi dari A ke dalam B atau f memetakan A ke dalam B.

Jika himpunan A = B, maka $f : A \rightarrow A$ disebut operator atau transformasi pada A.

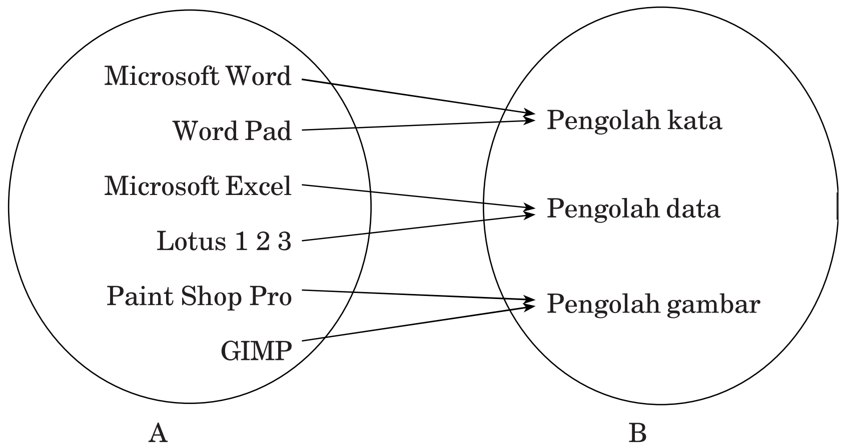
Contoh:

Misalkan A = {Microsoft Word, Word Pad, Microsoft Excel, Lotus 123, Paint Shop Pro, Gimp}

B = {Pengolah kata, Pengolah data, Pengolah gambar}

Misalkan $f : A \rightarrow B$

Maka:



Himpunan A disebut ranah (domain) dari fungsi f.
Himpunan B disebut ko-ranah (kodomain) dari fungsi f.

Pengolah kata adalah bayangan dari Microsoft Word dan Word Pad, dinyatakan oleh:

f (Microsoft Word) dan
f (Word Pad)

Jangkauan (range) dari f adalah (Pengolah kata, Pengolah data dan Pengolah gambar).

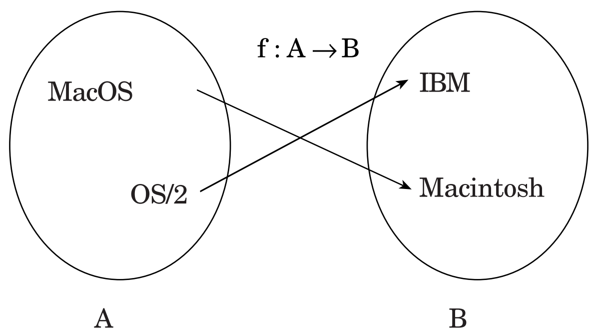
6.2 MACAM-MACAM FUNGSI

6.2.1 Fungsi satu-satu

Sebuah fungsi $f : A \rightarrow B$ dikatakan fungsi satu-satu jika dan hanya jika setiap elemen pada himpunan A mempunyai bayangan yang tidak sama pada elemen himpunan B.

Contoh:

A = himpunan sistem operasi
A = {MacOS, OS/2}
B = himpunan Komputer
B = {IBM, Macintosh}

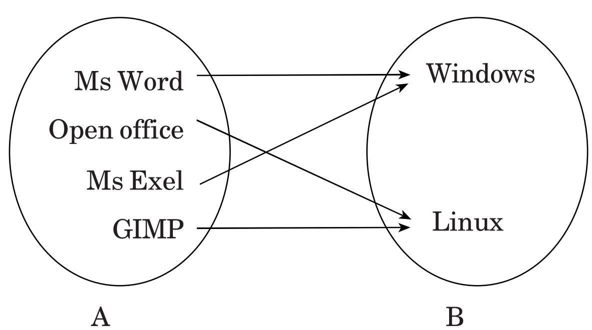


6.2.2 Fungsi pada

Sebuah fungsi $f : A \rightarrow B$ dikatakan fungsi pada jika dan hanya jika setiap elemen himpunan B muncul sebagai bayangan dari sekurang-kurangnya satu elemen himpunan A.

Contoh:

A = himpunan sofware aplikasi
B = himpunan sistem operasi

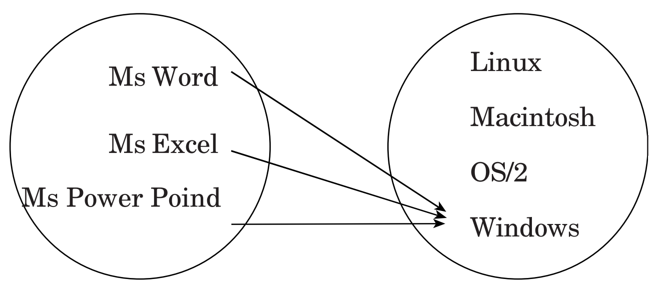


6.2.3 Fungsi konstan

Suatu fungsi $f : A \to B$ dikatakan fungsi konstan jika dan hanya jika hanya ada satu elemen himpunan B yang menjadi bayangan dari seluruh elemen himpunan A.

Contoh:

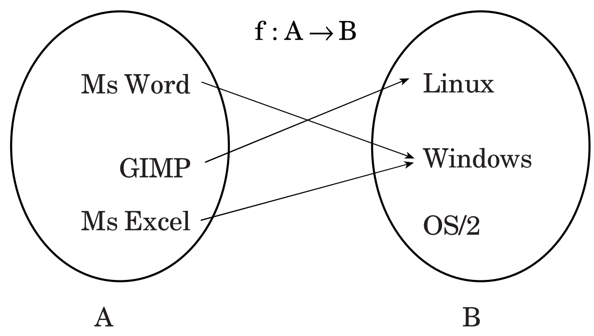
A = himpunan sofware aplikasi
B = himpunan sistem operasi



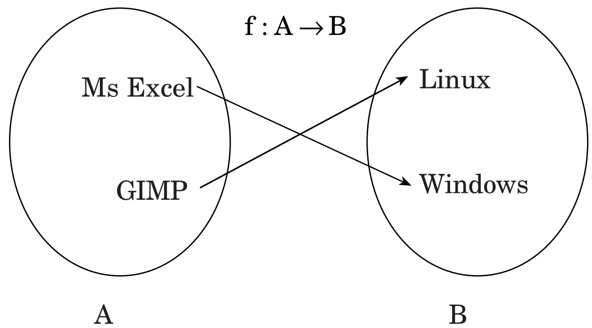
6.2.4 Fungsi Invers

Fungsi invers $f^{-1} : B \to A$ adalah sebuah fungsi dimana untuk setiap $b \in B$ mempunyai bayangan tunggal dalam himpunan A. Dengan demikian hanya fungsi satu-satu yang memiliki fungsi invers.

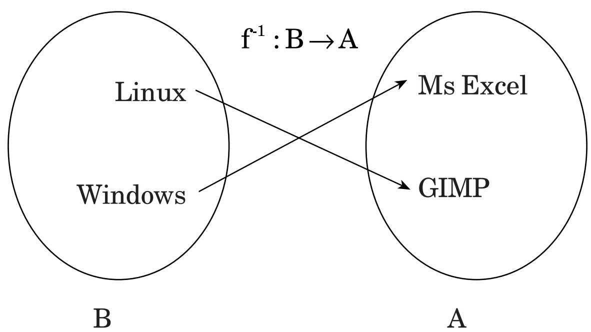
Contoh:



$f : A \rightarrow B$ bukan fungsi satu-satu , sehingga tidak memiliki fungsi invers f^{-1}



$f : A \rightarrow B$ adalah fungsi satu-satu sehingga fungsi invers $f^{-1} : B \rightarrow A$ ada yaitu



Contoh lain :

Misalkan $f(x) = {}^{3}\log(x-2)$, maka $f^{-1}(x)$ adalah

$$y = {}^{2}\log(x-2)$$
$$3^{y} = x - 2$$
$$x = 3^{y} + 2$$
$$y = 3^{x} + 2$$
$$\therefore f^{-1} = 3^{x} + 2$$

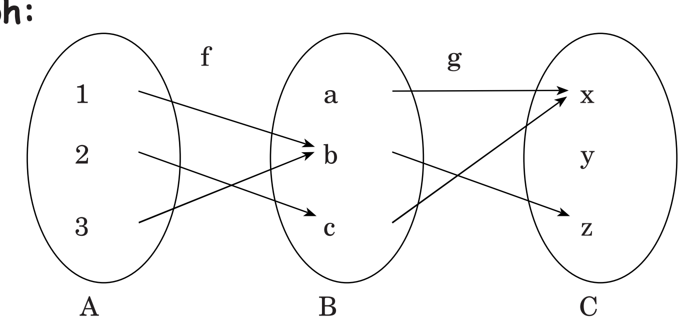
6.3 KOMPOSISI FUNGSI

Komposisi fungsi dari fungsi f dan g dinyatakan oleh $(g \circ f)$ atau (gf).

Jika $f : A \to B$ dan $g : B \to C$, maka

$$(g \circ f) : A \to C$$
$$(g \circ f)(a) \equiv g(f(a))$$

Contoh:



Maka $(g \circ f) : A \to C$ adalah

$$(g \circ f)(1) = g(f(1)) = g(b) = z$$
$$(g \circ f)(2) = g(f(2)) = g(c) = x$$
$$(g \circ f)(3) = g(f(3)) = g(b) = z$$

Misalkan $f(x) = x^2 - 1$ dan $g(x) = x + 3$.

Maka

$$(f \circ g)(2) = f(g(2)) = f(5) = 24$$
$$(g \circ f)(2) = g(f(2)) = g(3) = 6$$

6.4 Fungsi Karakteristik

Fungsi karakteristik adalah sebuah fungsi yang memetakan semesta pembicaraan ke dalam himpunan {1,0}, dinotasikan

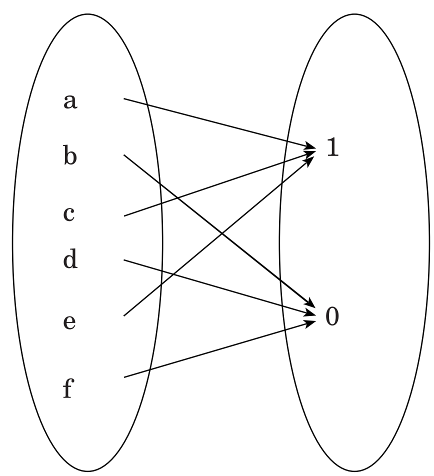
$K_A : U \to (0,1)$

dimana $K_A(x) = \begin{cases} 1 \text{ jika } x \in A \\ 0 \text{ jika } x \notin A \end{cases}$

Contoh:

1) Misalkan U = {a, b, c, d, e, f}
A = {a, c, e}

Maka $K_A : U \to (0,1)$ dapat didefinisikan melalui diagram di bawah:



2) Misal U = {a, e, i, o, u}
A = {a, e, i}
B = {e, i, o}

Buktikan $K_{A \cap B} = K_A K_B$

Jawab:

$e \in (A \cap B)$ maka $e \in A$ dan $e \in B$

$\therefore K_{A \cap B}(e) = 1,\ K_A(e) = 1,$ dan $K_B(e) = 1$

Jadi:

$K_{A \cap B}(e) = (K_A K_B)(e) = 1 \cdot 1 = 1$

$\therefore K_{A \cap B} = K_A \cdot K_B$

SOAL

1. Buatlah 5 buah contoh fungsi satu-satu yang ada kaitan-nya dengan software maupun hardware.
2. Buatlah 5 buah contoh fungsi pada yang ada kaitanya dengan software maupun hardware.
3. Misalkan fungsi-fungsi f dan g pada bilangan-bilangan riel $R^{\#}$ didefinisikan oleh

   $f(x) = 2x^2 + x - 3$

   $g(x) = 5x - 2$

   a) Carilah rumus-rumus dari $g \circ f$ dan $f \circ g$.
   b) Hitung $(g \circ f)(3)$ dan $(f \circ g)(3)$.
4. Misalkan $f : R^{\#} \to R^{\#}$ didefinisikan oleh $f(x) = 3x - 2$.
   a) Buktikanlah bahwa $f(x)$ adalah fungsi satu-satu dan fungsi pada
   b) Carilah rumus fungsi invers f^{-1}.
5. Misalkan U = {1, 2, 3, . . . , 10}
   A = {1, 2, . . . , 6}
   B = {5, 6, . . . , 10}

Buktikan:

a. $K_{A \cap B} = K_A \cdot K_B$

b. $K_{A \cup B} = K_A + K_B - K_{A \cap B}$

c. $K_{A - B} = K_A - K_{A \cap B}$

-oo0oo-

2

ALJABAR BOOLE

Aljabar Boole adalah cabang ilmu matematika yang di perlukan untuk mempelajari disain logika dari sebuah sistem digital. Aljabar Boole dikembagkan oleh George Boole tahun 1847 untuk memecahkan persoalan logika matematika.

7.1 APLIKASI ALJABAR BOOLE DALAM JARINGAN SWITCHING

Jaringan komunikasi biasa digambarkan dalam node dan link; Node merepresentasikan end-terminal perangkat jaringan, digambarkan dengan lingkaran/kotak; Link merepresentasikan hubungan/koneksi antar nodes, digambarkan dengan garis. Sebagai perangkat jaringan node dapat berfungsi sebagai: Routing, Switching, maupun Multiplexing.

Transmisi data/informasi jarak jauh biasa dilakukan melalui jaringan switching node, transmisi dimulai dan diakhiri di perangkat yang disebut station, yang dapat berupa komputer, terminal, telepon, dll.

Switching node pada umumnya mengenal dua keadaan (bilangan biner) 0 dan 1, 0 digunakan untuk jangkauan tegangan antara 0 sampai 0,8 volt, 1 digunakan untuk jangkauan tegangan antara 2 volt sampai 5 volt. Jadi 0 dan 1 menyatakan variabel tegangan

Untuk dapat menyelesaikan persoalan jaringan switching kita harus mengenal terlebih dahulu hukum-hukum aljabar Boole sebagai berikut: Bila $a, b, c \in B$ (B =himpunan Boole) maka memenuhi hukum-hukum:

1. Komutatif

   $a + b = b + a$

   $a \times b = b \times a$

2. Distributif

   $a + (b \times c) = (a + b) \times (a + c)$

   $a \times (b + c) = (a \times b) + (a \times c)$

3. Identitas

   $a + 0 = a$,0 ≡ elemen zero

   $a \times 1 = a$,1 ≡ elemen unit

4. Komplemen

   $a + a' = 1$

   $a \times a' = 0$

5. Idempoten

   $a + a = a$

   $a \times a = a$

6. Boundednes

   $a + 1 = 1$

   $a \times 0 = 0$

7. Absorbsi

   $a + (a \times b) = a$

   $a \times (a + b) = a$

8. Involusi

$$(a')' = a$$
$$0' = 1$$
$$1' = 0$$

9. Asosiatif

$$(a + b) + c = a + (b + c)$$
$$(a \times b) \times c = a \times (b \times c)$$

10. De Morgan

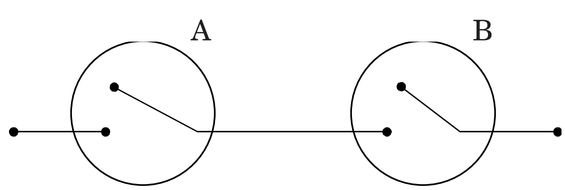
$$(a + b)' = a' \times b'$$
$$(a \times b)' = a' + b'$$

Kita perhatikan bahwa setiap hukum di atas mengganti operasi logik + dengan x , x dengan +, 0 dengan 1, 1 dengan 0. Ini menunjukkan bahwa hukum-hukum aljabar Boole memenuhi prinsip duality, yaitu dual suatu hukum merupakan hukum juga.

Jaringan switching pada umumnya dibentuk dari rangkaian dasar seri (AND) dan pararel (OR).

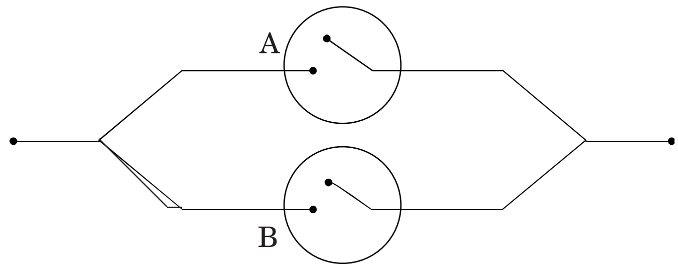
Rangkaian seri (AND)

Notasi: $A \times B$ atau $A \wedge B$ atau AB



Rangkaian pararel (OR)

Notasi: $A + B$ atau $A \vee B$



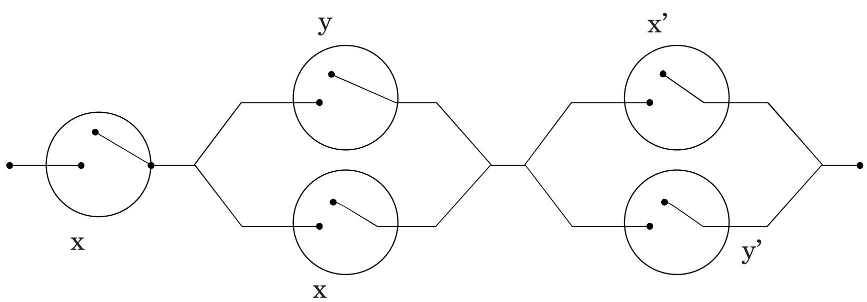
Contoh:

Gambarkan jaringan swiching yang diberikan oleh polinomial Boole $(x \wedge (y \vee x) \wedge (x \wedge y)')$, kemudian sederhanakanlah jaringan tersebut; Bilamana jaringan tersebut on atau off?

Jawab:

$x \wedge (y \vee x) \wedge (x \wedge y)'$

$\equiv x \wedge (y \vee x) \wedge (x' \vee y')$, DeMorgan



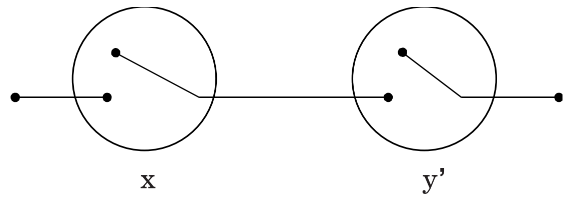
$\equiv x \wedge (y \vee x) \wedge (x' \vee y')$, absorbsi

$\equiv x \wedge (x' \vee y')$, distribusi

$\equiv (x \wedge x') \vee (x \wedge y')$, komplemen

$\equiv 0 \vee (x \wedge y')$, identitas

$\equiv x \wedge y'$



x	y	y’	$x \wedge y'$
1	0	1	1
1	1	0	0
0	0	1	0
0	1	0	0

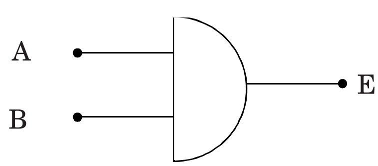
Jadi jaringan switching tersebut on bila x on dan y off atau jaringan on bila x on dan y' on.

7.2 APLIKASI ALJABAR BOOLE PADA RANGKAIAN LOGIK (GATE)

Sebagian besar rangkaian dalam hardware sistem pengolah data adalah rangkaian logik, yang dapat bekerja sebagai penguat, pembanding, perata, osilator, penjumlahan, pengendali, penyandi, dll.

Ada beberapa simbol yang sering digunakan dalam rangkaian logik yaitu:

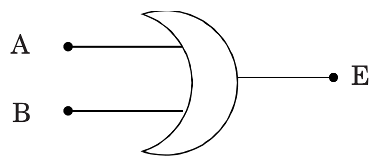
1. AND



A	B	$A \times B$
0	0	0
0	1	0
1	0	0
1	1	1

$E \equiv A \times B$

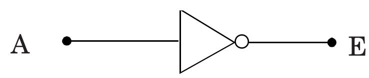
2. OR



A	B	A + B
0	0	0
0	1	1
1	0	1
1	1	1

$E \equiv A + B$

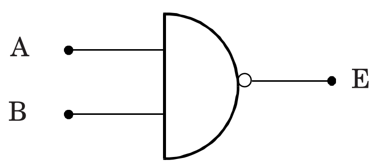
3. NOT



A	A’
0	1
1	0

$E \equiv A'$

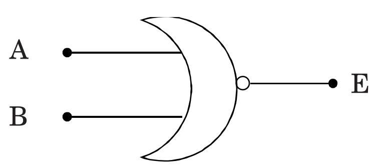
4. NOT AND (NAND)



A	B	E
0	0	1
0	1	1
1	0	1
1	1	0

$E \equiv (A \times B)'$

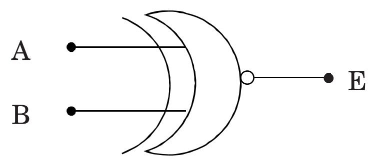
5. NOT OR (NOR)



A	B	E
0	0	1
0	1	0
1	0	0
1	1	0

$E \equiv (A + B)'$

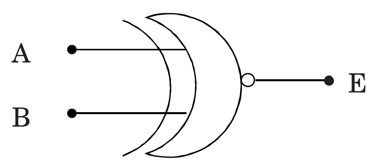
6. Exlusive OR (EXOR)



A	B	E
0	0	0
0	1	1
1	0	1
1	1	0

$E \equiv AB' + A'B \equiv A \oplus B$

7. Exclusive NOR (EXNOR)



A	B	E
0	0	1
0	1	0
1	0	0
1	1	1

$E \equiv AB + A'B'$

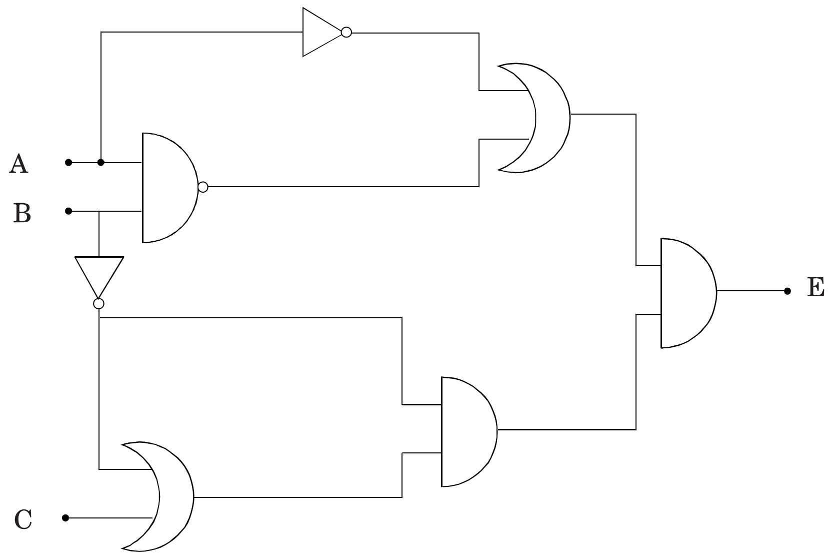
Contoh:

Gambarkan rangkaian logika yang dinyatakan oleh:

$$E = \left(A' \vee (A \wedge B)'\right) \wedge \left(B' \wedge (B' \vee C)\right)$$

Kemudian sederhanakan.

Jawab



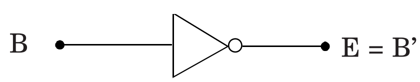
$$E \equiv \left(A' \vee (A \wedge B)'\right) \wedge \left(B' \wedge (B' \vee C)\right)$$

$$E \equiv \left(A' \vee (A \wedge B)'\right) \wedge B' \qquad \text{, Absorbsi}$$

$$E \equiv \left(A' \vee A' \vee B'\right) \wedge B' \qquad \text{, De Morgan}$$

$$E \equiv \left(A' \vee B'\right) \wedge B' \qquad \text{, Idempoten}$$

$$E \equiv B' \qquad \text{, Absorbsi}$$



7.3 Aplikasi Aljabar Boole dalam Operasi Kelipatan Persekutuan Kecil (KPK) dan Faktor Persekutuan Besar (FPB)

Dalam aljabar Boole operasi + sama dengan operasi KPK dan operasi x sama dengan operasi FPB, untuk mengingat kembali operasi KPK dan FPB perhatikan contoh berikut:

Contoh:

Carilah KPK dan FPB dari 45, 48 dan 72.

Jawab:

Faktor prima dari 45 adalah $3^2 \times 5$
Faktor prima dari 48 adalah $2^4 \times 3$
Faktor prima dari 72 adalah $2^3 \times 3^2$

Jadi KPK dari 45, 48 dan 72 adalah $2^4 \times 3^2 \times 5 = 720$
Jadi FPB dari 45, 48 dan 72 adalah 3.

Perhatikan untuk KPK, semua faktor prima yang ada dikalikan, faktor yang sama diambil pangkat tertinggi; untuk FPB hanya faktor prima yang sama dalam 45, 48 dan 72 dikalikan, diambil faktor prima dengan pangkat terendah.

Contoh:

Misalkan diketahui himpunan Boole

$B = D_{60} = \{1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30, 60\}$

Cari:

1. 5 + 12
   KPK dari 5 dan 12 adalah 60
2. 5 x 12
   FPB dari 5 dan 12 adalah 1
3. Elemen zero=?
   a + 0 = a
   identitas

+	1	2	3	4	5	6	10	12	15	20	30	60
1	1	2	3	4	5	6	10	12	15	20	30	60
2	2	2	6	4	10	6	10	12	30	20	30	60
3	3	6	3	12	15	6	30	12	15	60	30	60
4	4	4	12	4	20	12	20	12	60	20	60	60
5	5	10	15	20	5	30	10	60	15	20	30	60
6	6	6	6	12	30	6	30	12	30	60	30	60
10	10	10	30	20	10	30	10	60	30	20	30	60
12	12	12	12	12	60	12	60	12	60	60	60	60
15	15	30	15	60	15	30	30	60	15	60	30	60
20	20	20	60	20	20	60	20	60	60	20	60	60
30	30	30	30	60	30	30	30	60	30	60	30	60
60	60	60	60	60	60	60	60	60	60	60	60	60

Perhatikan, yang memenuhi rumus identitas a + 0 = a adalah 1, jadi elemen zero dari B = D_{60} adalah 1.

4. Elemen Unit = ?

   $a \times 1 = a$

   identitas

$\times$	1	2	3	4	5	6	10	12	15	20	30	60
1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	2	1	2	1	2	2	2	1	2	2	2
3	1	1	3	1	1	3	1	3	3	1	3	3
4	1	2	1	4	1	2	2	4	1	4	2	4
5	1	1	1	1	5	1	5	1	5	5	5	5
6	1	2	3	2	1	6	2	6	3	2	6	6
10	1	2	1	2	5	2	10	2	5	10	10	10
12	1	2	3	4	1	6	2	12	3	4	6	12
15	1	1	3	1	5	3	5	3	15	15	15	15
20	1	2	1	4	5	2	10	4	5	20	10	20
30	1	2	3	2	5	6	10	6	15	10	30	30
60	1	2	3	4	5	6	10	12	15	20	30	60

Perhatikan, yang memenuhi rumus identitas a x 1 = a adalah 60, jadi elemen unit dari B = D_{60} adalah 60.

5. $10' = ?$

$a' + a = \text{unit}$, komplemen

$10' + 10 = 60$

$\ldots + 10 = 60$, lihat tabel soal nomer 3

$\left.\begin{array}{l} 12 + 10 = 60 \\ 60 + 10 = 60 \end{array}\right\}$, jadi 10' = 12, karena 12 faktor dari 60

7.4 Minimal DNF (Disjunctive Normal Form)

Minimal dnf adalah ekspresi Boole yang ada dalam bentuk minimal, dimana suku-sukunya tidak ada satu didalam yang lain. Minimal dnf dapat dicari dengan 2 cara, yaitu:

1) Dengan teori include dan teori konsensus.
2) Dengan peta karnaugh.

7.4.1 Dengan Teori Include dan Konsensus

Sebelum mempelajari teori include dan konsensus ada baiknya kita kenal dulu beberapa istilah, yaitu:

a) Fundamental product (perkalian dasar) adalah perkalian dua atau lebih variabel-variabel Boole yang tidak memuat variabel yang saling komplemen atau sama.

Contoh:

$ab, abc', a'bc', \ldots$ disebut fundamental product

$aba, a'ba, abb', \ldots$ bukan fundamental product

b) Ekspresi Boole (E)
Adalah penjumlahan satu atau lebih fundamental product.

Contoh:

$E = ab + ab'c' + abc'$

Ekspresi Boole dalam bentuk dnf, jika suku-suku E tidak ada satu di dalam yang lain.

Contoh:

$E_1 = ab + a'bc' + ab'c$, bentuk dnf

$E_2 = ab + abc$, bukan dnf karena ab di dalam abc

Teori Include

Jika fundamental product p_1, termasuk didalam fundamental product p_2, maka $p_1 + p_2 = p_1$

Contoh:

ab + abc = ab

Konsensus

Jika fundamental product p_1 dan p_2 memiliki satu elemen saja yang saling komplemen, maka konsensus (Q) dari p_1 dan p_3 adalah perkalian elemen-elemen p_1 dan p_2 dimana elemen-elemen yang saling komplemen dihilangkan dan tidak ada pengulangan.

Contoh:

$\left.\begin{array}{l} P_1 = abc' \\ P_2 = abcd \end{array}\right\}$ maka konsensus dari P_1 dan P_2 adalah Q = abd

$\left.\begin{array}{l} P_1 = abc' \\ P_2 = a'bcd \end{array}\right\}$ tidak ada konsensus sebab ada 2 elemen yang saling komplemen

Teori Konsensus

Jika Q konsensus dari P_1 dan P_2 maka

$P_1 + P_2 + Q = P_1 + P_2$

Contoh:

$$\left.\begin{aligned} P_1 &= abc' \\ P_2 &= abcd \end{aligned}\right\} Q = abd$$

$$\begin{aligned} P_1 + P_2 + Q &= abc' + abcd + abd \\ &= abc' + abcd + (c + c')abd \\ &= abc' + abcd + abc'd + abcd \\ &= \underbrace{abc' + abc'd}_{\text{include}} + \underbrace{abcd + abcd}_{\text{include}} \\ &= abc' + abcd \\ &= P_1 + P_2 \end{aligned}$$

Ekspresi Boole minimal adalah ekspresi Boole (E) yang semua fundamental productnya sudah minimal dnf (tidak ada satu di dalam yang lain).

Contoh:

$E_1 = a'b'$, minimal dnf

$E_2 = ab + ab'c + a'bc'$, belum minimal

E_2 dapat diminimalkan dengan teori konsensus sebagai berikut:

konsensus

$$E_2 = ab + ab'c + a'bc'$$

konsensus

include

$$E_2 = ab + ac + ab'c + a'bc' + bc'$$

$$E_2 = ab + ac + bc'$$

include

E_2 adalah ekspresi Boole minimal

Suku-suku dari ekspresi Boole minimal disebut prime implicant (p_i) dari ekspresi Boole (B) yang memenuhi sifat

$$E + P_i = E$$

dan tidak ada fundamental product lain yang termasuk P_i mempunyai sifat-sifat tersebut.

Contoh:

bc' adalah prime implicant dari

$$\begin{aligned} E_2 &= ab + ab'c + a'bc' \\ E_2 + P_1 &= ab + ab'c + a'bc' + bc' \\ &= ab + ab'c + a'bc' \\ &= E_2 \end{aligned}$$

,karena bc' konsensus dari ab dan a'bc'maka ab + a'bc' + bc' = ab + a'bc'

Jadi ekspresi Boole minimal sama dengan jumlah dari prime implicantnya.

7.4.2 Peta Karnaugh

Adalah peta dari ekspresi Boole yang dapat digunakan untuk mencari prime implicant dan ekspresi Boole minimal.

Peta untuk 2 variable

	x	x'
y		
y'		

y \ x	0	1
0		
1		

Peta untuk 3 variable

	xy	x'y	x'y'	xy'
z				
z'				

yz \ x	0	1
00		
01		
11		
10		

Peta untuk 4 variable

	xy	x'y	x'y'	xy'
za				
z'a				
z'a'				
za'				

za \ xy	00	01	11	10
00				
01				
11				
10				

Contoh:

Cari ekspresi Boole minimal dari:

$E_1 = abcd + a'bcd + a'b'cd + ab'cd + abcd' + a'b'cd' + ab'cd'$

Jawab: $E_1 = c$ (lihat peta)

$E_2 = abcd + ab'cd + abcd' + ab'cd'$

Jawab: $E_2 = ac$ (lihat peta)

$E_3 = xyz'a + xyz'a' + x'y'z'a' + xy'z'a + xy'z'a'$

Jawab: $E_3 = xz' + y'z'$ (lihat peta)

Jawab:

	ab	a'b	a'b'	ab'
cd	1	1	1	1
c'd				
c'd'				
cd'	1	1	1	1

$E_1 = c$

cd \ ab	00	01	11	10
00				
01				
11	1	1	1	1
10	1	1	1	1

$E_1 = c$

	ab	a'b	a'b'	ab'
cd	√			√
c'd				
c'd'				
cd'	√			√

$E_2 = ac$

cd \ ab	00	01	11	10
00				
01				
11			√	√
10			√	√

$E_2 = ac$

	xy	x'y	x'y'	xy'
zd				
z'a	√		√	√
z'a'	√		√	√
za'				

$E_3 = xz' + y'z'$

za \ xy	00	01	11	10
00	√		√	√
01	√		√	√
11				
10				

$E_3 = xz' + y'z'$

SOAL-SOAL

1. Gambarkanlah jaringan switching yang dinyatakan dengan polinominal Boole di bawah, kemudian sederhanakan dan gambarkan bentuk sederhananya, kapan jaringan tersebut on atau off.
   a) $(A \wedge B) \vee (A \wedge B') \vee (A'B')$
   b) $(A \wedge (C \vee B') \vee (B \wedge C)')$
   c) $\big(((A \vee B) \wedge C) \vee A'\big) \wedge B$
   d) $(A \wedge B) \vee \big(A' \wedge (B' \vee A \vee B)\big)$
   e) $(A \vee B) \wedge C \wedge (A \vee B \vee C')$
   f) $(A \wedge B) \vee C \vee (A' \vee C')$
   g) $(B \wedge (A \vee C)) \vee (A \vee C)$
   h) $(A \wedge (B \vee A) \wedge (A \wedge B)') \wedge (A' \vee B)'$

2. Gambarkanlah gebang logika yang dinyatakan dengan ekspresi Boole di bawah, kemudian sederhanakan dan gambarkan bentuk sederhananya.
   a) $\big((A' + A) \times (B \times B')\big)' = E$
   b) $\Big(\big((A \times B') + \big((A \times B) + B\big) \times B\Big) + A = E$
   c) $\big((A + B) \times C\big)' + \Big(\big((A + B) \times C\big) \times D\Big) = E$
   d) $(A' \times A) + B + (B + B) = E$
   e) $(A' \times A') + (B' \times A) = E$
   f) $\Big(\big((A \times B) + C\big)' \times D\Big) \times \Big(\big((A \times B) + C\big) + D\Big) = E$
   g) $ABC + AB'(A'C')' = E$
   h) $A'B'C' + A'BC' + AB'C' + ABC' = E$
   i) $(A+B+C)(A+B+C')(A'+B+C')(A'+B+C')(A'+B'+C') = E$
   j) $A'BC' + (A+B')C = E$
   k) $[\ BC\ \{(A' + AB)' + (ABC)'\}']'$

3. Rancanglah sebuah jaringan logika penjumlah yang mempunyai 3 inputan dan 2 outputan, yaitu x, y, z sebagai input dan c, s sebagai output, dimana c = 1 bila 2 atau 3 inputnya sama dengan 1 serta s = 1 bila hanya 1 atau ketiga inputnya sama dengan 1 .
4. Bila diketahui $B = D_{70}$, carilah:
   a) elemen zero
   b) elemen unit
   c) $10 + (35 \times 70)$
   d) $10'$
5. Bila diketahui $B = D_{90}$, carilah:
   a) elemen zero
   b) elemen unit
   c) $15 \times (30 + 18)$
   d) $30'$
6. Cari prime implicant dari expresi Boole dibawah:

$$\begin{aligned} E_1 = {} & a'b'c'd' + ab'c'd' + a'bc'd + a'b'cd + \\ & a'bcd + abcd + a'b'cd' + abcd' + ab'cd' \end{aligned}$$

$$\begin{aligned} E_2 = {} & a'bc'd' + abc'd' + a'b'c'd + abc'd + \\ & ab'c'd + ab'cd + a'bcd' \end{aligned}$$

$$\begin{aligned} E_3 = {} & a'b'c'd' + a'bc'd' + abc'd' + ab'c'd' + \\ & a'b'c'd + abc'd + abcd + ab'cd' \end{aligned}$$

$$\begin{aligned} E_4 = {} & a'bc'd + abc'd + a'b'cd + a'bcd + abcd + \\ & ab'cd + a'b'cd' + abcd' + ab'cd' \end{aligned}$$

$$\begin{aligned} E_5 = {} & a'bc'd' + abc'd' + a'b'c'd + a'bc'd + \\ & abc'd + a'b'cd + abcd + ab'cd + abcd' + ab'cd' \end{aligned}$$

7. Rancang sebuah jaringan logika pengurang yang mempunyai 3 inputan x, y, dan z dan 2 keluaran b dan d. b=1 bila ketiga inputnya sama dengan 1 atau z=1 atau y=1 atau z=y=1. d=1 bila ketiga inputnya atau satu inputnya sama dengan 1.

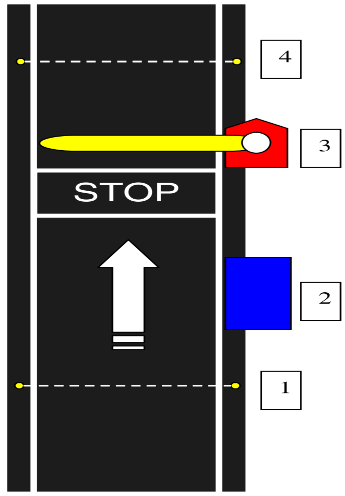
8. Rancang sebuah jaringan logika pembanding 1 bit yang mempunyai 2 inputan A dan B dan 3 outputan yaitu x bila $A < B$, y bila $A = B$ dan z bila $A > B$.

9. Kembangkan rancangan jaringan logika pada soal no. 8 untuk pembanding 2 bit.

10. Seorang mahasiswa ingin merancang sebuah jaringan logika yang mampu merubah bilangan biner tak berbobot menjadi bilangan biner berbobot (dekoder) seperti :

Desimal	Biner tak berbobot	Biner berbobot
25	11001	$\underline{0010}$ $\underline{0101}$ 2 5
43	101011	$\underline{0100}$ $\underline{0011}$ 4 3

bila mahasiswa tersebut membatasi hanya untuk bilangan biner 4 bit atau maksimum 1111 selesaikan pekerjan mahasiswa tersebut.

11. 

Gambar diatas adalah gerbang TOL otomatis dengan 4 alat yaitu

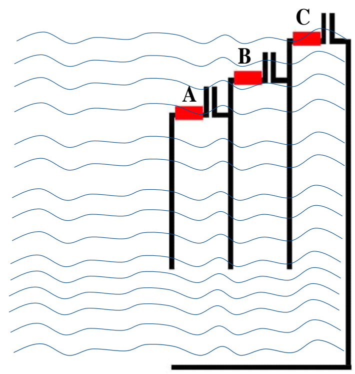
1. Sensor masuk mobil,
2. Box kartu yang memiliki sensor,
3. Mesin portal,
4. Sensor keluar mobil.

Berikut mekanismenya;

1. Jika alat sensor masuk = 1 , Sensor Box = 0 dan sensor keluar = 0, maka mesin box mengeluarkan kartu tanda masuk TOL
2. Jika pengemudi telah mengambil kartu pada box kemudian sensor box aktif yaitu = 1 , sensor masuk = 0, sensor keluar = 0, maka mesin portal terbuka.
3. Jika mobil telah melewati sensor keluar sehingga sensor masuk = 0 , sensor box = 0 dan sensor keluar = 1 , maka mesin portal tertutup kembali.

Rancanglah rangkaian logika dengan 3 inputan X untuk sensor masuk, Y untuk sensor box, Z untuk sensor keluar, disertai dengan tabel kebenaran.

12.



Gambar di atas a adalah alat peringatan dini bahaya banjir, Pada alat tesebut terdapat 3 buah sensor yang berfungsi mencek ketinggian air. berikut mekanisme keja alat tersebut;

1. Saat keadaan normal ketiga sensor = 0.
2. Siaga 3 saat air menyentuh sensor A, maka A = 1 , B=0 dan C =0.
3. Siaga 2 saat air menyentuh sensor B, maka A=1, B=1, dan C =0.
4. Siaga 1 saat air menyentuh sensor C, maka ketiga sensor = 1.

buatlah rangkaian logika untuk alat tersebut yang dapat memberikan peringatan dini pada saat siaga1, siaga2 dan siaga3.

13. Integrasikanlah soal nomor 3 dan soal no 7 sehingga menjadi sebuah jaringan logika dengan 3 inputan dan 4 outputan.

-oo0oo-

8

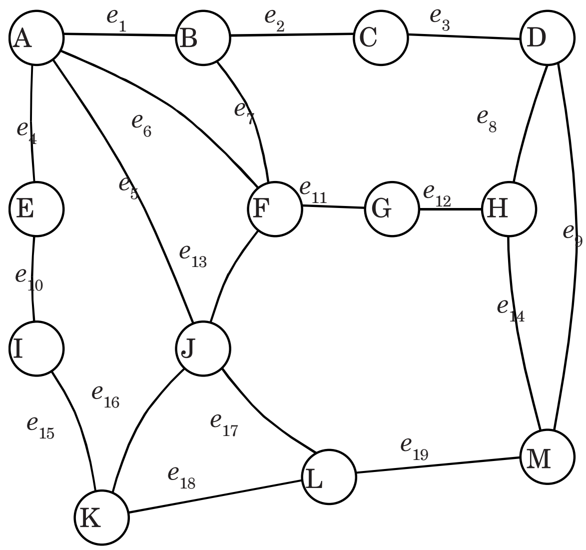
TEORI GRAPH

8.1. Pendahuluan

Dalam kehidupan sehari-hari, banyak persoalan yang dapat disimpulkan sebagai persoalan yang berhubungan dengan himpunan dan relasi binary, di mana logika dari persoalan tersebut sering kali dapat kita gambarkan dengan sebuah graph.

Contoh:

Seorang programer ingin membuat software sistem jaringan trasportasi sedemikian rupa sehingga apabila sebuah kendaraan bergerak dari titik A kesemua titik lain kemudian kembali ke titik A dapat dilakukan dengan efisien.



Kita lihat di sini titik A, B, ... M menggambarkan himpunan titik-titik lampu merah dimana kendaraan tertahan selama 1 menit, garis atau rusuk mengambarkan relasi "terhubung" antara titik-titik yang ada. Jadi dapat kita simpulkan, graph adalah gambaran logika dari suatu kejadian, proses, peristiwa atau hal-hal lain yang saling berkaitan. Graph adalah himpunan pasangan terurut (V, E), dimana V adalah himpunan vertex/titik dan E adalah himpunan edge/rusuk. Untuk terhubung ke b oleh suatu garis/rusuk jika {a, b} $\in E$. Bila kita perhatikan graph diatas, ternyata unsur-unsur graph adalah vertek/titik-titik simpul/noktah yang diwakili oleh A,B ....., M dan rusuk/edge yang diwakili oleh e_1, e_2,..., e_{19}.

A dikatakan berdekatan/berdampingan/adjacent dengan B, E, F dan J. e_1 dikatakan bertemu/incident dengan e_2 dan e_7 di titik B.

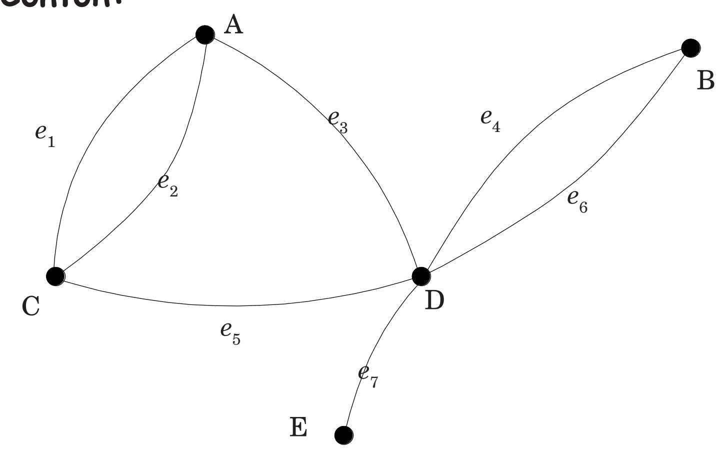
8.2 Macam-macam Graph

Macam-macam graph dilihat dari stukturnya ada 6 macam graph, yaitu:

a) Multigraph

Multigraph adalah graph yang mempunyai satu atau lebih pasangan rusuk ganda yang menghubungkan 2 buah titiknya.

Contoh:

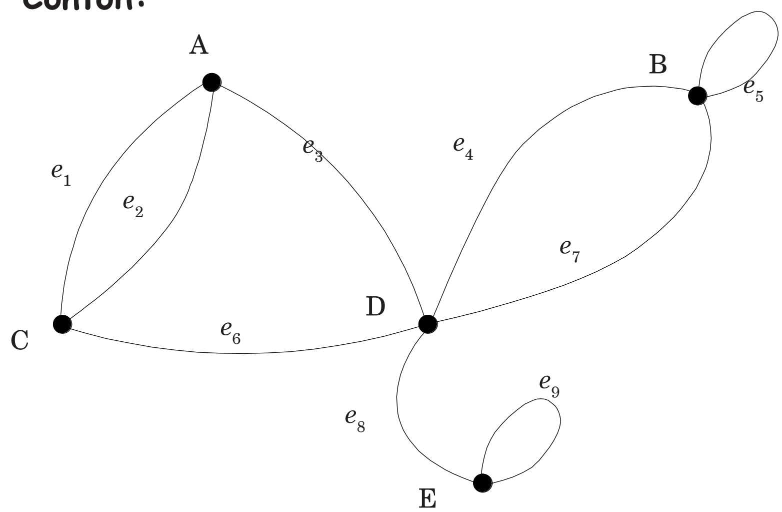


Titik A dan C dihubungkan oleh 2 buah rusuk, e_1 dan e_2, demikian juga titik B dan D dihubungkan oleh rusuk e_4 dan e_6.

b) Pseudograph

Pseudograph adalah graph yang memiliki satu atau lebih pasang rusuk ganda yang menghubungkan 2 buah titiknya (multigraph) dan memiliki satu atau lebih loap pada titiknya.

Contoh:



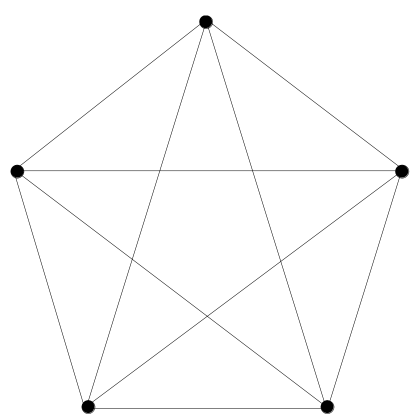
Graph di atas selain memiliki rusuk ganda juga memiliki dua buah loap dititik B dan E. Loap adalah rusuk yang ujungnya hanya memiliki sebuah titik.

c) **Trivialgraph**
Trivialgraph adalah graph yang hanya terdiri dari satu titik.

d) **Graph lengkap**
Graph lengkap adalah graph yang setiap titiknya terhubung dengan semua titik yang lain dengan hanya satu rusuk.

Contoh:

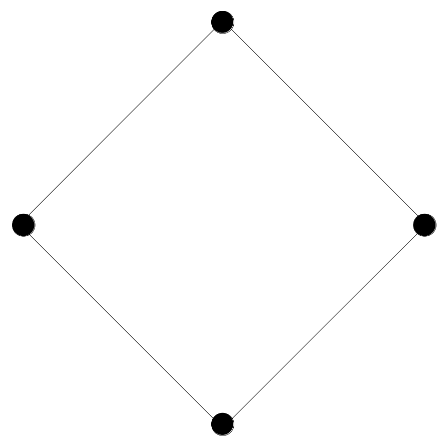
K_5



e) **Graph teratur**

Graph teratur adalah graph yang setiap titiknya mempunyai sejumlah incident rusuk yang sama.

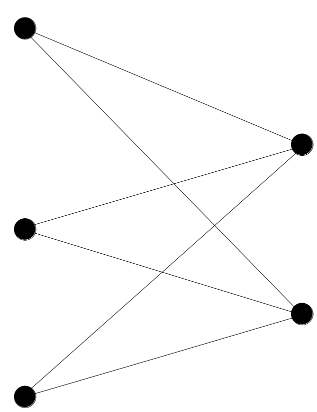
Contoh:



f) **Bipartitegraph**

Bipartitegraph adalah graph yang titik-titiknya dapat dikelompokkan menjadi dua, titik-titik dalam satu kelompok tak terhubung dan titik-titik antar kelompok terhubung lengkap.

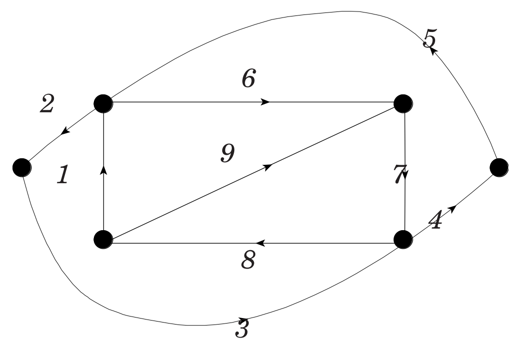
Contoh:



Dilihat dari lintasannya ada 3 macam graph, yaitu:

a) **Traversable graph**

Traversable graph adalah graph yang semua rusuk-rusuknya dapat dilalui masing-masing sekali atau graph yang dapat digambar tanpa mengangkat pensil.



Contoh:

Teori Euler:

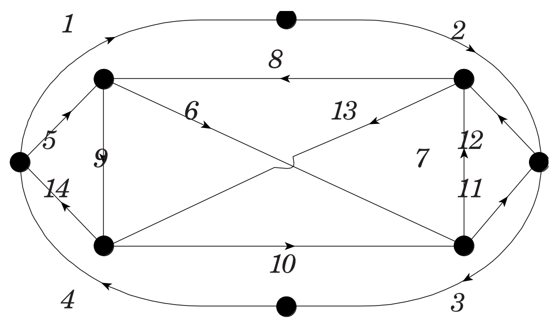
- Semua graph terhubung yang mempunyai titik ganjil maksimum dua adalah traversable.
- Traversable lintasannya selalu dimulai dari titik ganjil pertama dan diakhiri pada titik ganjil kedua.

Titik ganjil adalah titik dimana rusuk yang incident/bertemu dengan titik tersebut berjumlah ganjil.

c) **Eulerian graph**

Eulerian graph adalah graph yang semua rusuknya dapat dilalui masing-masing sekali dan memiliki lintasan tertutup, artinya titik awal sama dengan titik akhir.

Contoh:



Teori Euler:

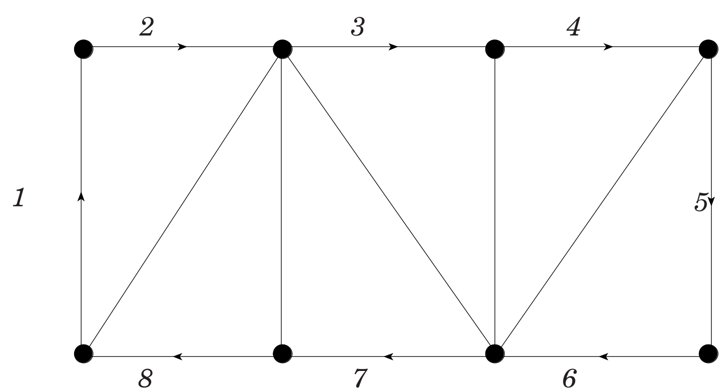
Bila sebuah graph semua titiknya genap maka graph tersebut mempunyai lintasan euler.

Karena graph euler dapat digambar tanpa angkat pensil maka euler graph juga merupakan traversable graph.

c) **Hameltonian graph**

Hameltonian graph adalah graph yang semua titik-titiknya dapat dilalui masing-masing sekali dan mempunyai lintasan tertutup, artinya titik awal sama dengan titik akhir.

Contoh:



8.3 KONEKSITAS

Hubungan atau lintasan antar titik dalam sebuah graph dapat dibedakan manjadi beberapa jenis, yaitu:

a) **Walk**
Walk adalah lintasan dari suatu titik ke titik yang lain.

Contoh:

Misalkan titik mewakili kota dan rusuk mewakili jalan, maka dari Jakarta ke Bogor kita dapat membuat banyak walk, yaitu:

Jakarta – Jagorawi – Bogor
Jakarta – Tangerang – Bogor
Jakarta – Cikampek – Bandung – Bogor
dan lain-lain.

b) **Closed Walk**
Closed Walk adalah walk yang titik awal sama dengan titik akhir

Contoh:

Jakarta – Cikampek – Jakarta
Jakarta – Jagorawi – Bogor – Tangerang – Bogor – Jagorawi –Jakarta
dan lain-lain.

c) **Trail**
Trail adalah walk yang semua rusuknya berlainan, artinya yang kita perhatikan adalah lintasannya.

Contoh:

Jl. Borobudur – Jl. Prambanan – Jl. Mendut

Jl. Merdeka Barat – Jl. M.H. Thamrin – Jl. Sudirman

dan lain-lain.

d) Path

Path adalah walk yang semua titiknya berlainan, artinya yang kita perhatikan kotanya.

Contoh:

Jakarta – Cikampek – Purwakarta
Jakarta – Bogor – Cianjur – Bandung
dan lain-lain.

e) Cycle

Cycle adalah path yang tertutup, artinya titik awal sama dengan titik akhir.

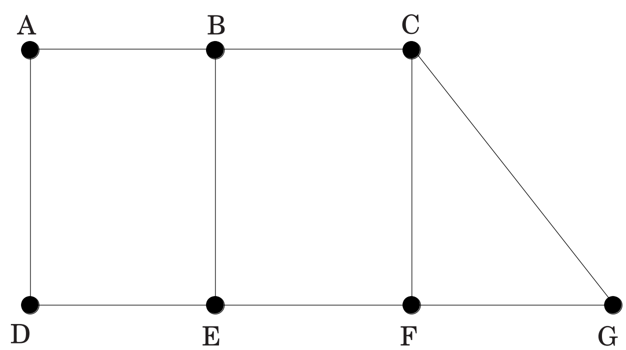
Contoh:

Jakarta – Tangerang – Bogor – Jakarta
Jakarta – Cikampek – Padalarang – Cianjur – Bogor – Jakarta. dan lain-lain.

f) Girth

Girth adalah cycle terpendek dari cycle-cycle yang dimiliki oleh sebuah graph.

Contoh:



Graph di atas mempunyai banyak cycle, tetapi ada satu yang terpendek yang disebut girth, yaitu CGFC, panjangnya 3 (banyak rusuk yang membentuk cycle)

g) Circumference

Circumference adalah cycle terpanjang dari cycle-cycle yang dimiliki oleh sebuah graph.

Contoh:

Dari contoh graph (f) diatas, A B C G F E D A adalah circum ference dengan panjang = 7. (banyaknya rusuk yang membentuk cycle)

8.4 BERKAITAN DENGAN JARAK

Dalam sebuah graph, mengetahui hal-hal yang berkaitan dengan jarak penting, antara lain untuk menentukan jari-jari, diameter, sentral, dan pusat graph.

Jarak antara dua titik adalah walk yang semua titiknya berlainan dan mempunyai lintasan terpendek.

Contoh:

Kita dapat membuat banyak walk yang semua titiknya berlainan antara Jakarta – Bogor, yaitu

Jakarta – Jagorawi – Bogor
Jakarta – Tangerang – Bogor
Jakarta – Cikampek – Padalarang – Puncak – Bogor.

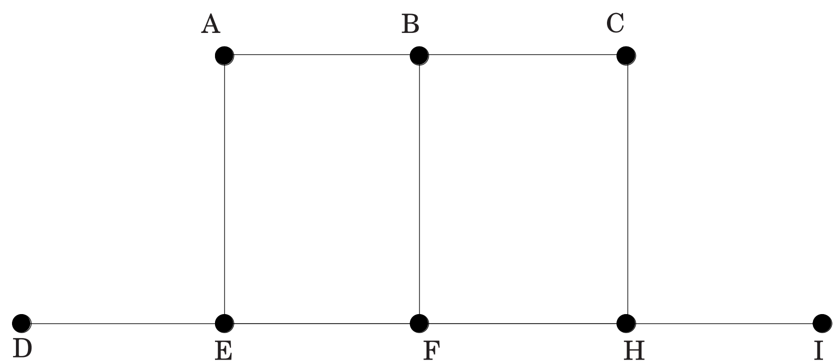
Dari contoh lintasan-lintasan di atas yang disebut jarak adalah lintasan Jakarta – Jagorawi – Bogor karena terpendek.

Ada beberapa hal yang berkaitan dengan jarak, yaitu.

a) Eksentrisitas suatu titik (e(u))

Eksentrisitas suatu titik adalah jarak terpanjang suatu titik terhadap semua titik dalam sebuah graph.

Contoh:



Jarak A – B = 1

A – C = 2

A – D = 2

A – E = 1

A – F = 2

A – H = 3

A – I = 4

Jadi eksentrisitas titik A = *e* (A) = 4

b) Jari - jari graph (r(G))

Jari-jari adalah eksentrisitas titik yang terkecil dalam sebuah graph.

Contoh:

Dari contoh graph diatas, eksentrisitas titik-titiknya, sebagai berikut:

e (A) = 4
e (B) = 3
e (C) = 4
e (D) = 4
e (E) = 3

e (F) = 2
e (H) = 3
e (I) = 4
Jadi jari-jari graph = r (G) = 2

c) **Diameter graph (d(G))**
Diameter graph adalah eksentrisitas titik yang terbesar dalam sebuah graph.

Contoh:

Dari graph di atas dapat disimpulkan bahwa diameter graph d (G) = 4.

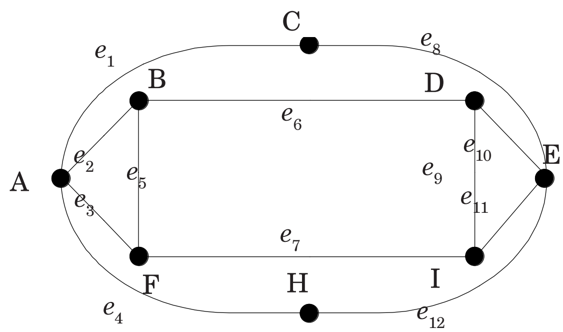
d) **Titik sentral graph**
Titik sentral graph adalah titik-titik simpul yang nilai eksentrisitasnya sama dengan nilai jari-jarinya. Dari cotoh di atas titik sentral graph adalah titik F.

e) **Pusat graph**
Pusat graph adalah himpunan titik-titik yang nilai eksentrisitasnya sama dengan nilai jari-jarinya. Dari contoh diatas, pusat graph adalah {F}

8.5 DERAJAT/DEGREE SUATU TITIK

Seperti kita ketahui sebuah titik dalam graph dapat mempunyai 1 atau lebih rusuk yang incident padanya atau tidak ada satupun rusuk yang incident padanya. Derajat sebuah titik adalah banyaknya rusuk yang incident pada titik tersebut. Titik ganjil adalah titik yang derajatnya ganjil.Titik genap adalah titik yang berderajat genap.

Contoh:



Maka derajat titik-titiknya adalah:

$$
\begin{aligned}
\deg(A) &= 4\\
\deg(B) &= 3\\
\deg(C) &= 2\\
\deg(D) &= 3\\
\deg(E) &= 4\\
\deg(F) &= 3\\
\deg(H) &= 2\\
\deg(I) &= 3
\end{aligned}
$$

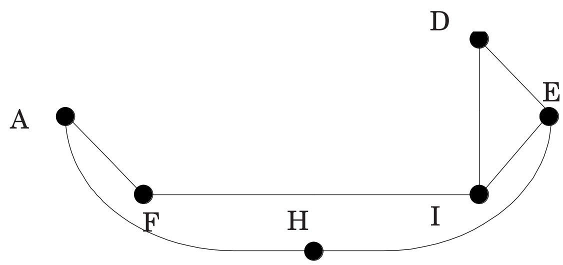
Jumlah degree = 24
Jumlah rusuk = 12
Jumlah degree = 2 kali jumlah rusuk.

8.6 Titik Potong Graph (Cut Point)

Sebuah graph dapat dipotong pada sebuah atau lebih titiknya, jika suatu titik dalam sebuah graph dinyatakan sebagai titik potong, maka titik tersebut dan semua rusuk yang incident pada titik itu dihilangkan.

Contoh:

Bila titik-titik B dan C pada contoh graph pada bagian 8.5 dinyatakan sebagai cut point, maka terjadi graph baru seperti di bawah:



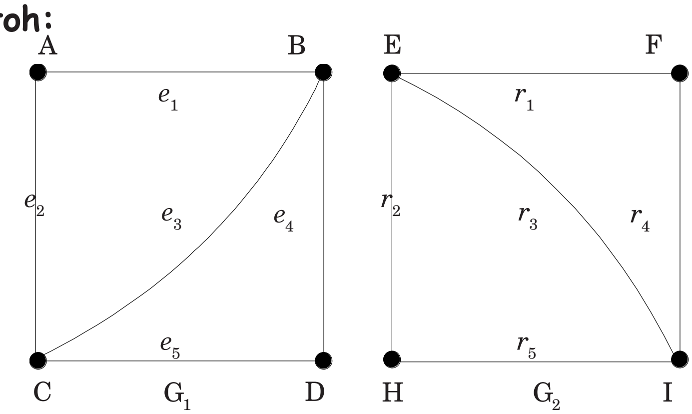
8.7 UKURAN SECARA GRAFIKAL

Sebuah graph dapat kita pelajari melalui ukuran grafisnya, yang meliputi:

- Jumlah rusuk
- Jumlah titik
- Derajat titik
- Titik potong

Dua buah graph yang mempunyai ukuran-ukuran grafis sama disebut Isomorphic graph.

Contoh:



G_1 dan G_2 isomorphis, ukuran grafisnya sama dan berkorespondensi 1 – 1 antara titik-titik dan rusuk-rusuk yaitu:

titik-titik	rusuk-rusuk
	$e_1 - r_4$
A – F	$e_2 - r_1$
B – I	$e_3 - r_3$
C – E	$e_4 - r_5$
D – H	$e_5 - r_2$

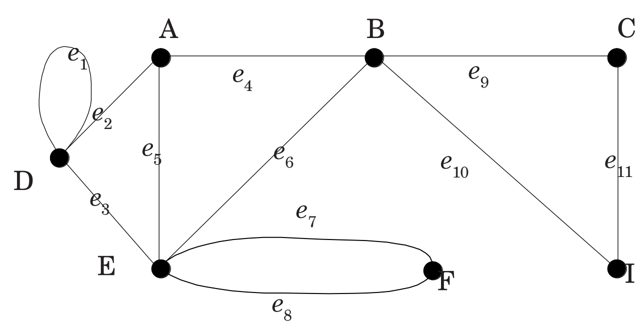
8.8 MATRIK GRAPH

Sebuah graph dapat kita sajikan dalam bentuk matrik, yaitu:

a) Matrik titik (Adjacent Matrix)
b) Matrik rusuk (Edge Matrix)
c) Matrik titik – rusuk (Incidence Matrix)

Contoh:

Nyatakanlah graph di bawah dalam bentuk matrik titik, rusuk dan titik rusuk.



Matrik titik dari graph di atas adalah matrik 7 X 7, karena graph di atas mempunyai 7 buah titik

$$
\begin{array}{c}
\\
\mathrm{A}\\ \mathrm{B}\\ \mathrm{C}\\ \mathrm{D}\\ \mathrm{E}\\ \mathrm{F}\\ \mathrm{I}
\end{array}
\begin{array}{c}
\begin{array}{ccccccc} \mathrm{A} & \mathrm{B} & \mathrm{C} & \mathrm{D} & \mathrm{E} & \mathrm{F} & \mathrm{G} \end{array}\\
\begin{pmatrix}
0 & 1 & 0 & 1 & 1 & 0 & 0\\
1 & 0 & 1 & 0 & 1 & 0 & 1\\
0 & 1 & 0 & 0 & 0 & 0 & 1\\
1 & 0 & 0 & 2 & 1 & 0 & 0\\
1 & 1 & 0 & 1 & 0 & 2 & 0\\
0 & 0 & 0 & 0 & 2 & 0 & 0\\
0 & 1 & 1 & 0 & 0 & 0 & 0
\end{pmatrix}
\end{array}
$$

Cara mengisi elemen-elemen matrik:

- Baris 1 kolom 1, dari A ke A = 0
- Baris 1 kolom 2, dari A ke B = 1,titik A dan B terhubung oleh sebuah rusuk
- Baris 4 kolom 4, dari D ke D = 2, titik D mempunyai loop
- Baris 5 kolom 6, dari E ke F = 2, titik E dan F terhubung oleh 2 buah rusuk e_7 dan e_8
- Baris 7 kolom 1, dari I ke A = 0, titik I dan A tidak terhubung oleh sebuah rusuk.

Matrik rusuk dari graph di atas adalah matrik 11 X 11, karena graph di atas mempunyai 11 rusuk.

	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8	e_9	e_{10}	e_{11}
e_1	0	1	1	0	0	0	0	0	0	0	0
e_2	1	0	1	1	1	0	0	0	0	0	0
e_3	1	1	0	0	1	1	1	1	0	0	0
e_4	0	1	0	0	1	1	0	0	1	1	0
e_5	0	1	1	1	0	1	1	1	0	0	0
e_6	0	0	1	1	1	0	1	1	1	1	0
e_7	0	0	1	0	1	1	0	1	0	0	0
e_8	0	0	1	0	1	1	1	0	0	0	0
e_9	0	0	0	1	0	1	0	0	0	1	1
e_{10}	0	0	0	1	0	1	0	0	1	0	1
e_{11}	0	0	0	0	0	0	0	0	1	1	0

Cara mengisi elemen-elemen matrik:

Bila sebuah rusuk bertemu dengan rusuk yang lain disebuah titik maka elemen matriknya = 1, bila tidak bertemu di satu titik maka elemen matriknya = 0.

Matrik titik-rusuk dari graph di atas adalah matrik 7 X 11, karena graph tersebut memiliki 7 titik dan 11 rusuk.

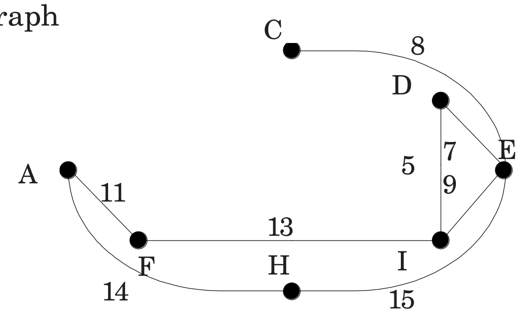
	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8	e_9	e_{10}	e_{11}
A	0	1	0	1	1	0	0	0	0	0	0
B	0	0	0	1	0	1	0	0	1	1	0
C	0	0	0	0	0	0	0	0	1	1	0
D	1	1	1	0	0	0	0	0	0	0	0
E	0	0	1	0	1	1	1	1	0	0	0
F	0	0	0	0	0	0	1	1	0	0	0
I	0	0	0	0	0	0	0	0	0	1	1

Cara mengisi elemen-elmen metrik

Bila sebuah rusuk bertemu dengan sebuah titik maka nilai elemen matrik = 1, bila tidak bertemu maka nilai elemen matrik = 0.

8.9 LABELED DIGRAPH

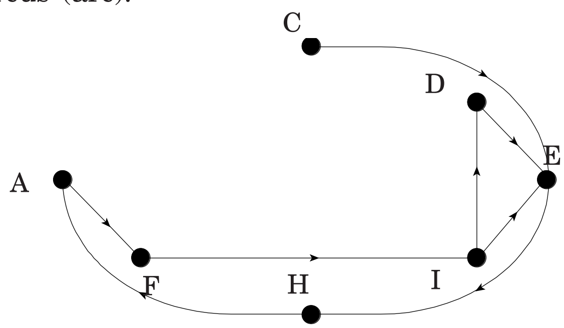
Dalam menggambarkan logika suatu kejadian sebuah graph sering kali diberi label/bobot, graph demikian disebut Labeled graph



Contoh:

Rusuk AF mempunyai bobot 11, rusuk AH mempunyai bobot 14 dan seterusnya. Bobot di sini bisa menyatakan jarak, selisih bunga deposito, kecepatan atau apa saja maksud pembuat graph.

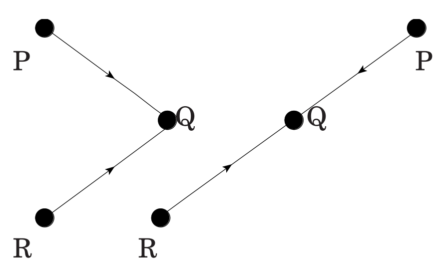
Rusuk sebuah graph dapat pula diberi arah untuk menggambarkan logika sebuah sistem yang berarah, graph demikian disebut digraph, rusuk yang berarah sering kali disebut arcus (arc).



Contoh:

Berkaitan dengan digraph maka hubungan antar titik dapat dikategorikan menjadi 3 macam, yaitu:

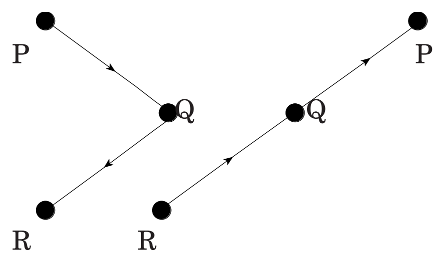
a) **Lemah (weak)**
Hubungan antar titik dalam digraph dikatakan lemah apabila arcusnya berlawanan



Contoh:

b) Unilateral

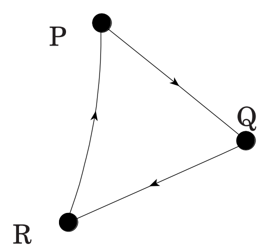
Hubungan antar titik dalam digraph dikatakan unilateral bila arcusnya searah



Contoh:

c) Kuat (strong)

Hubungan antar titik dalam digraph dikatakan kuat bila arcusnya searah dan tertutup

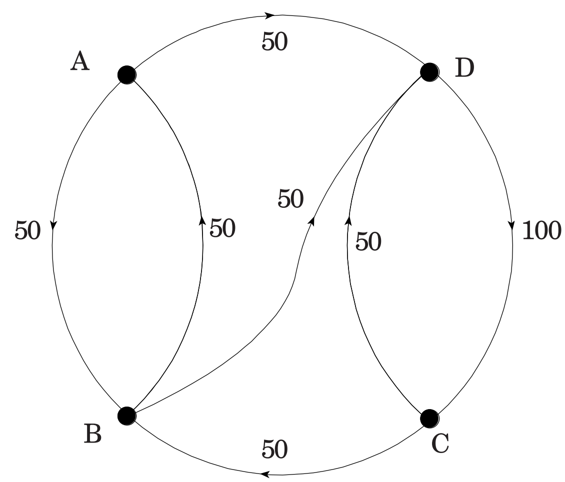


Contoh:

Rusuk sebuah graph dapat diberi bobot sekaligus diberi arah, graph demikian disebut labeled digraph.

Contoh:

A, B, C dan D bermain tembakan



A dapat manembak ke B dan D, jadi bobot AB = AD = 50%
B dapat manembak ke A dan D, jadi bobot BA = BD = 50%
C dapat manembak ke B dan D, jadi bobot CB = CD = 50%
D hanya dapat menembak ke C, jadi bobot DC = 100%

Arcus menunjukkan arah tembakan, bobot menyatakan peluang masing-masing tembakan.

8.10 DERAJAT TITIK PADA DIAGRAPH

Derajat sebuah titik pada digraph dapat dibagi menjadi dua, yaitu

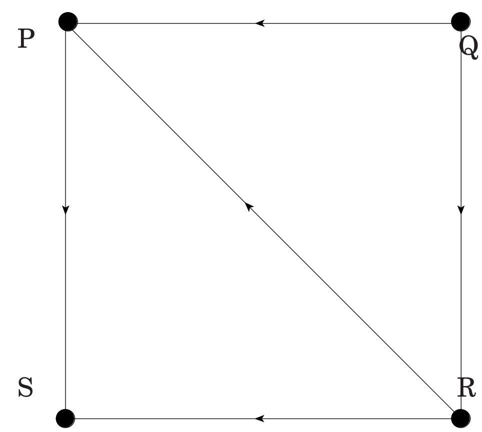
a) In degree
b) out degree

Indegree sebuah titik adalah jumlah rusuk yang masuk kesebuah titik.

Outdegree sebuah titik adalah jumlah rusuk yang keluar dari sebuah titik. Titik yang indegreenya = 0 disebut sumber/asal/source.

Titik yang outdegreenya = 0 disebut tujuan/sink.

Contoh:

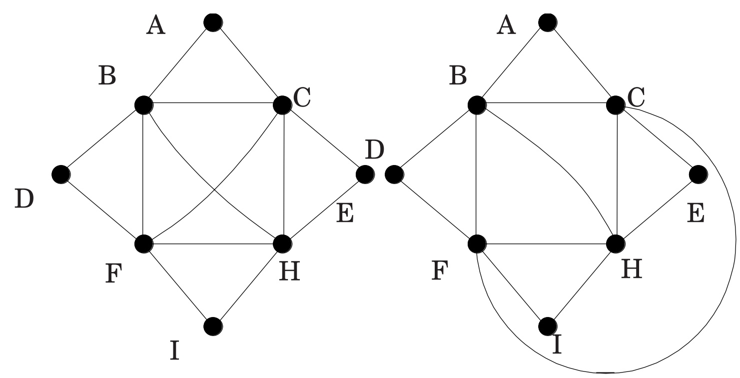


Titik Q = sumber, karena indegree Q = 0
Indegree P = 2, karena ada 2 rusuk yang masuk ke P
Outdegree P = 1, karena ada 1 rusuk yang keluar dari P
Indegree R = 1
Outdegree R = 2

Titik S = tujuan, karena outdegree S = 0

8.11 Graph Bidang (Planar Graph)

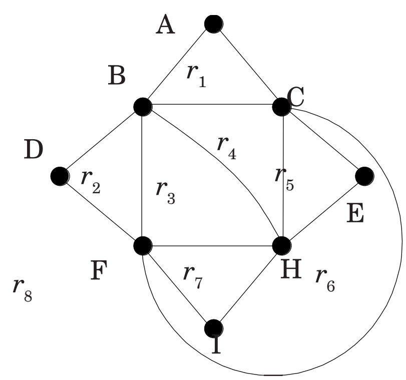
Sebuah graph dikatakan graph bidang bila rusuk-rusuknya terletak pada bidang datar serta tidak saling berpotongan selain di titiknya. Graph bidang dapat dibuat dari sebuah graph sebidang (a planar graph), seperti di bawah.



Aplanar ➔ Planar graph

Graph bidang disebut peta (map), rusuk-rusuk graph bidang memisahkan graph bidang atas wilayah-wilayah/daerah-daerah/region, karena wilayah dibatasi oleh rusuk-rusuk, maka wilayah dalam graph bidang dapat dibedakan menurut jumlah rusuk yang membatasi wilayah tersebut (derajat wilayah)

Contoh:



Rusuk-rusuk pada graph bidang di atas membagi graph bidang tersebut atas 8 wilayah, yaitu r_1 sampai r_8, dimana

r_1, r_2, r_3, r_4, r_5, dan r_7 berderajat 3
r_6 berderajat 5
r_8 berderajat 5

Rumus-rumus Euler.

Jika sebuah peta mempunyai titik sebanyak V, mempunyai wilayah sebanyak R dan mempunyai rusuk sebanyak E, maka peta tersebut memenuhi rumus-rumus Euler sebagai berikut:

1. $V + R - E = 2$
2. $\sum \deg(r) = 2\sum E$
3. $E \leq 3V - 6$

Contoh:

Dari peta di atas diketahui E = 14, banyak titik V = 8 dan bayak wilayah R = 8, sehingga

1. $V + R - E = 8 + 8 - 14 = 2$
2. $\sum \deg(r) = (3 \times 6) + (5 \times 2) = 28$
   $\sum E = 14$
   $\therefore \quad 28 = 2.14$
3. $14 \leq 3.8 - 6$
   $14 \leq 18$

8.12 PEWARNA PETA

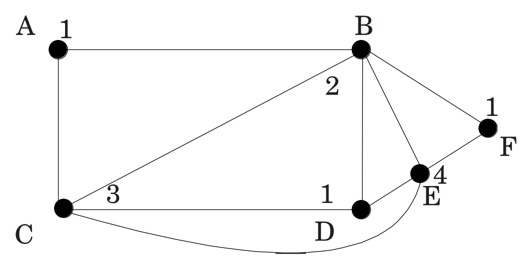
Pewarnaan sebuah peta dapat dilakukan dalam 3 cara, yaitu:

a) mewarnai titik.
b) mewarnai rusuk.
c) mewarnai wilayah.

Ada beberapa prinsip dalam mewarnai peta, yaitu:

- Banyak warna yang harus digunakan harus seminimum mungkin, banyak warna minimum disebut bilangan kromatik (X (G)).
- Dua buah titik yang terhubung oleh satu atau lebih rusuk tidak boleh diberi warna yang sama (pewarnaan titik).
- Dua buah rusuk atau lebih yang bertemu pada sebuah titik tidak boleh diberi warna sama (pewanaan rusuk).
- Dalam mewarnai peta pakailah sebuah warna secara optimum, artinya warna kedua digunakan setelah warna pertama tidak dapat digunakan lagi, demikian seterusnya sampai semua titik / rusuk / region terwarnai semua.

Contoh:

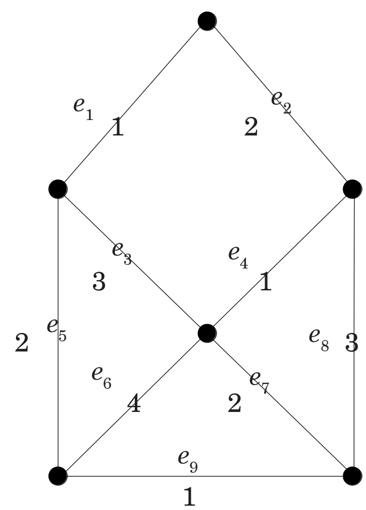


Mewarnai titik

1) Titik A kita beri warna 1,
2) Titik D dan F kita beri warna 1 karena baik titik D maupun F tidak saling terhubung langsung dengan titik A
3) Titik B, C dan E saling terhubung langsung sehingga harus diberi warna yang berbeda, yaitu warna 2, 3, dan 4.

   Jadi bilangan kromatik X (G) = 4

Mewarnai rusuk



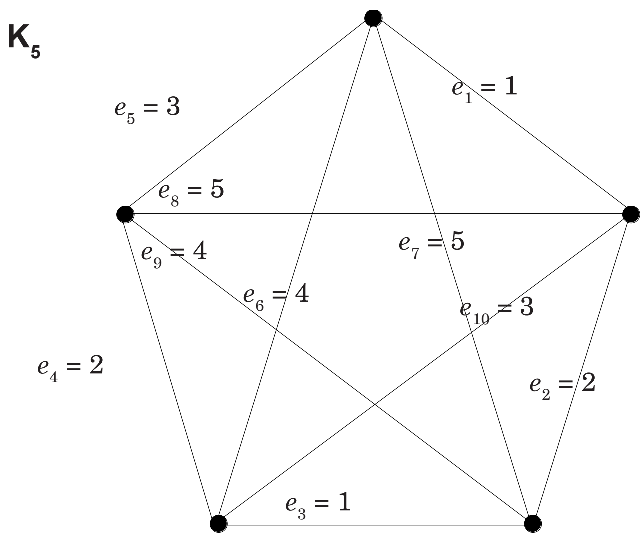
1) e_1 kita beri warna 1,
2) e_4 dan e_9 kita beri warna 1, karena e_1, e_4, dan e_9 tidak saling terhubung langsung oleh sebuah titik.
3) e_2 kita beri warna 2,
4) e_5 dan e_7 dapat diberi warna 2, karena e_2, e_5, dan e_7 tidak saling terhubung melalui sebuah titik.
5) e_3 dan e_8 diberi warna 3
6) e_6 diberi warna 4

Jadi bilangan kromatik graph di atas X (G) = 4

Dalam hal mewarnai rusuk untuk graph lengkap (K_n), bilangan kromatik dari K_n memenuhi rumus:

$$X(K_n) = \begin{cases} n, & \text{bila n ganjil} \\ n-1, & \text{bila n genap} \end{cases}$$

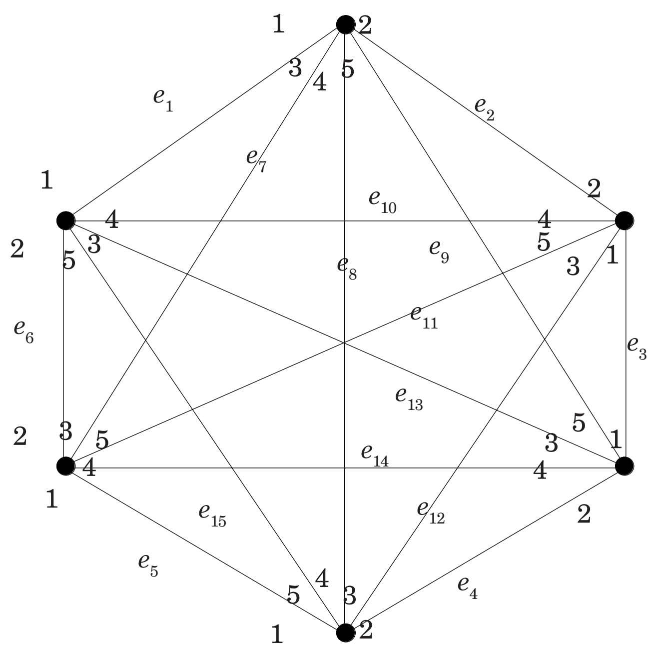
Contoh:



1. e_1 dan e_3 diberi warna 1
2. e_2 dan e_4 diberi warna 2
3. e_5 dan e_{10} diberi warna 3
4. e_9 dan e_6 diberi warna 4
5. e_8 dan e_7 diberi warna 5

Jadi $X(K_5) = 5$

K_6

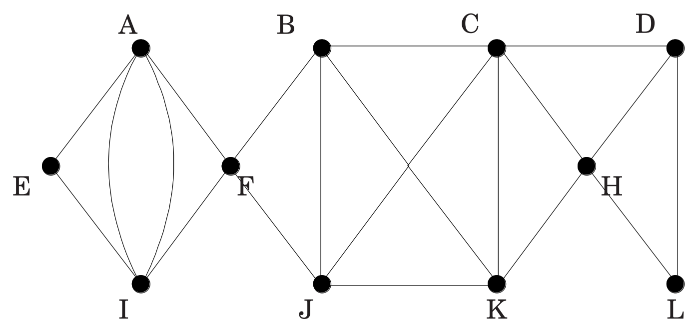


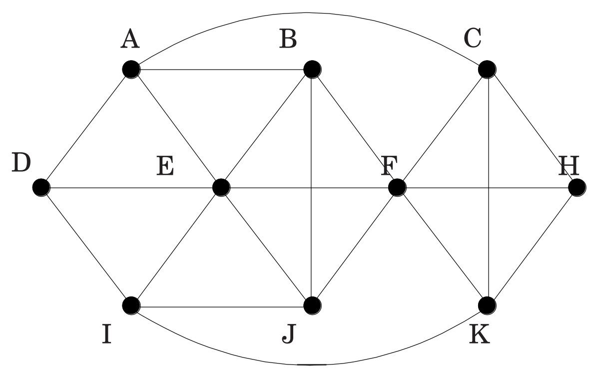
1. e_1, e_3 dan e_5 diberi warna 1
2. e_2, e_4 dan e_6 diberi warna 2
3. e_7, e_{12} dan e_{13} diberi warna 3
4. e_8, e_{10} dan e_{14} diberi warna 4
5. e_9, e_{15} dan e_{11} diberi warna 5

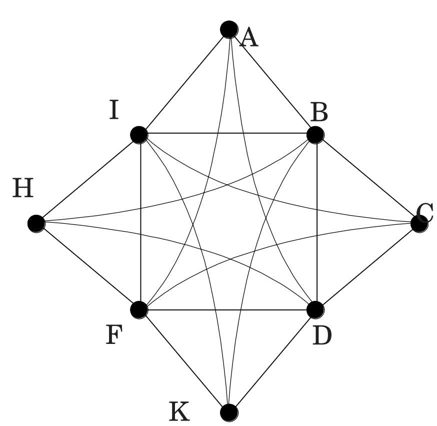
Jadi bilangan kromatik X $(K_6) = 5$.

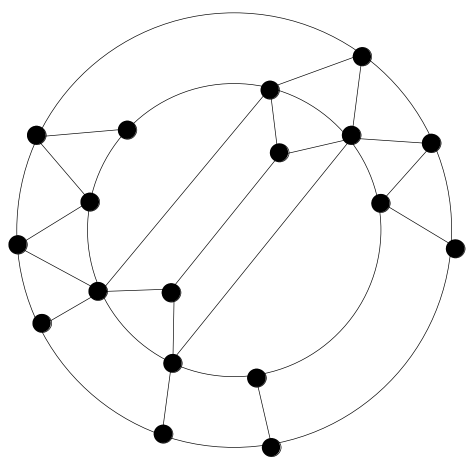
SOAL-SOAL

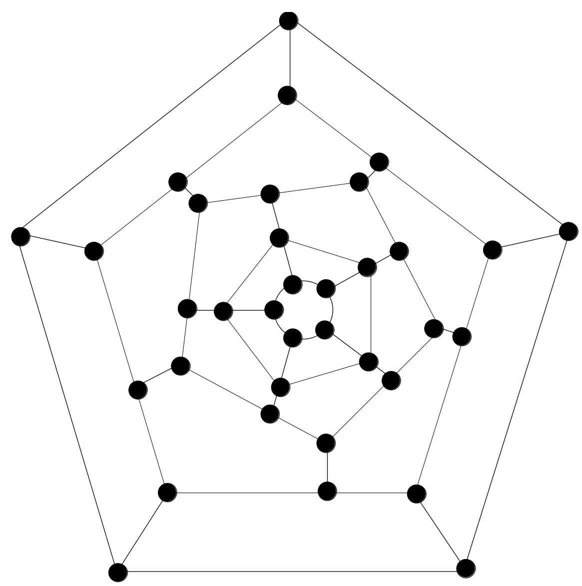
1. Buatlah lintasan yang mungkin (tranversable, euler dan hamelton) pada graph di bawah



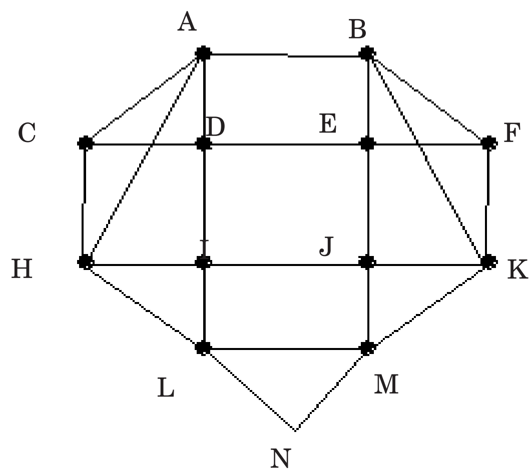


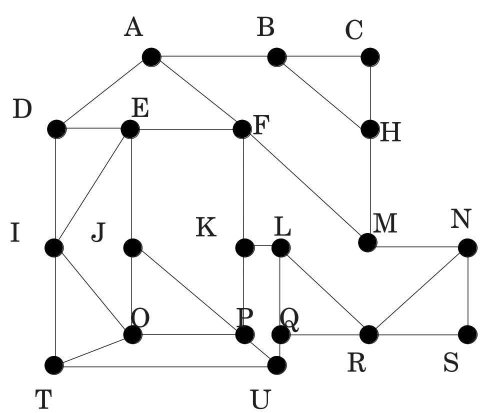






2. Seorang programer mendapat tugas membuat software untuk dipasang pada otak sebuah rudal jelajah yang akan ditembakan ke suatu wilayah yang digambarkan oleh graph di bawah, bila titik-titik mewakili obyek vital yang akan dihancurkan dan bila daya hancur ledakan rudal sampai radius 2 dari pusat ledakan, dititik manakah programer tesebut harus menjatuhkan rudal jelajah supaya daya hancurnya maksimum.





3. Gambarkanlah graph yang dipaparkan melalui matrik titik dibawah, kemudian tuliskan kembali dalam bentuk matrik titik rusuk

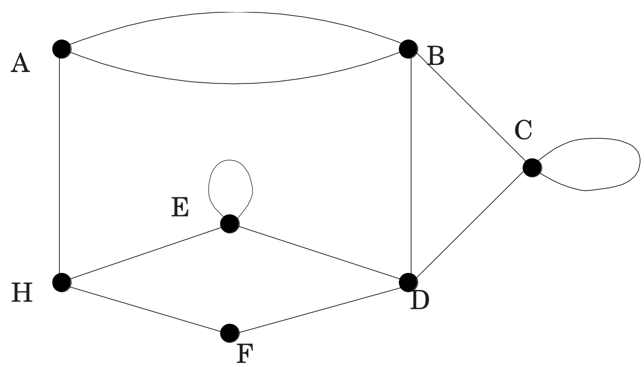
	A	B	C	D	E
A	0	1	0	0	1
B	1	0	1	0	1
C	0	1	2	0	1
D	0	0	0	0	2
E	1	1	1	2	0

	A	B	C	D	E	F
A	0	0	1	0	0	1
B	0	2	0	1	2	0
C	1	0	0	0	0	1
D	0	1	0	0	1	0
E	0	2	0	1	0	0
F	1	0	1	0	0	0

4. Gambarkanlah graph yang dipaparkan oleh matrik titik _ rusuk dibawah, kemudian tuliskan kembali dalam bentuk matrik rusuk

	A	B	C	D	E	F
e_1	1	0	0	1	0	0
e_2	1	0	0	0	1	0
e_3	1	0	0	0	0	1
e_4	0	1	0	1	0	0
e_5	0	1	0	0	1	0
e_6	0	1	0	0	0	1
e_7	0	0	1	1	0	0
e_8	0	0	1	0	1	0
e_9	0	0	1	0	0	1

5. Tuliskanlah graph di bawah dalam matrik titik, rusuk, titik – rusuk.



6. Warnailah K_7, K_8, dan K_9.
7. Sebuah graph direpresentasikan oleh matrik titik di bawah :

	A	B	C	D	E	F	G	H	I	J	K
A					1	1		2			
B			1			1			1	1	
C		1		1			1		1	1	
D			1				1				1
E	1							1			
F	1	1						1	1		
G			1	1						1	1
H	2				1	1					
I		1	1			1				1	
J		1	1				1		1		
K				1			1				

a) Warnailah graph di atas dengan pewarnaan titik, rusuk, dan region serta tuliskan masing - masing bilangan kromatisnya.
b) Tentukan pusat dan sentral graph

8. Seorang ahli jaringan mendapat order memasang RW - NET dengan sistem tanpa kabel, apabila titik - titik pelanggan digambarkan dengan matrik titik seperti di bawah dan radius sinyalnya 2 step dari titik hotspot, dimana ahli jaringan harus menempatkan hotspotnya, agar jangkauannya maksimum?

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
A		1		1		1														
B	1		1				1													
C		1					1													
D	1				1			1												
E				1		1		1	1											
F	1				1					1		1								
G		1	1									1								
H				1	1									1					1	
I					1									1	1					
J																				
K										1						1	1			
L						1	1						1							
M																				
N												1				1	1			
O								1	1						1				1	
P											1						1			1
Q									1	1				1						1
R												1					1			
S								1						1						1
T																1	1		1	

9. Sebuah graph dinyatakan dengan matrik titik di bawah :

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
A		1			1											
B	1		1			1	1									
C		1		1				1								
D			1			1	1									
E	1	1				1										
F		1			1		1			1			1			
G		1		1		1		1			1	1				
H			1	1			1					1				
I				1									1	1		
J						1	1				1			1	1	
K							1			1		1			1	1
L								1								1
M						1			1					1		
N									1	1			1		1	
O										1	1			1		
P											1	1				

a) Warnai graph di atas dengan pewarnaan titik, rusuk, dan region, kemudian tuliskan bilangan kromatik masing - masing pewarnaan tersebut.

b) Andaikan titik mempresentasikan terminal yang akan berlangganan internet dan rusuk menyatakan jarak antar terminal dalam hal ini dinyatakan sebagai 50 meter, dimana hotspot harus dipasang agar jangkauan internet maksimum bila radius signal hotspot hanya 100 meter.

10. Diketahui Adjacency Matrik dari graph G sebagai berikut :

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

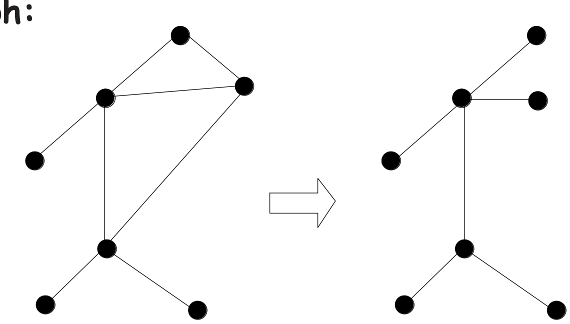
a) Gambarkan graph G
b) Warnailah titik dan rusuk graph G
c) Tentukan bilangan kromatik X(G) dan X'(G)

8.13 Pohon/Tree

Dalam dunia informatika pohon memegang peranan penting bagi seorang programer untuk menggambarkan hasil karyanya; bagi seorang user, setiap kali berhadapan dengan monitor untuk menjalankan program aplikasi selalu akan menelusuri bagian-bagian dari pohon sebelum sampai pada program aplikasi yang dimaksud.

Pohon adalah sebuah graph yang mempunyai n buah titik, n – 1 rusuk dan tidak mempunyai lingkaran (cycle free) serta merupakan graph terhubung.

Contoh:



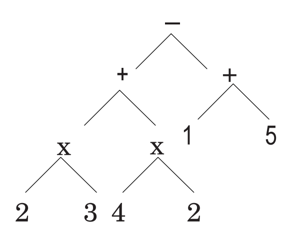
bukan pohon karena ada cycle

pohon; tidak ada cycle, $n = 7$; $E = 6$

Diagram pohon dapat digunakan sebagai alat untuk memaparkan logika sebuah persoalan dengan menggambarkan semua alternatif pemecahannya.

Contoh:

Pernyataan aritmatik 2 x 3 + 4 x 2 – 1 + 5 dapat dijelaskan dengan diagram pohon berikut :



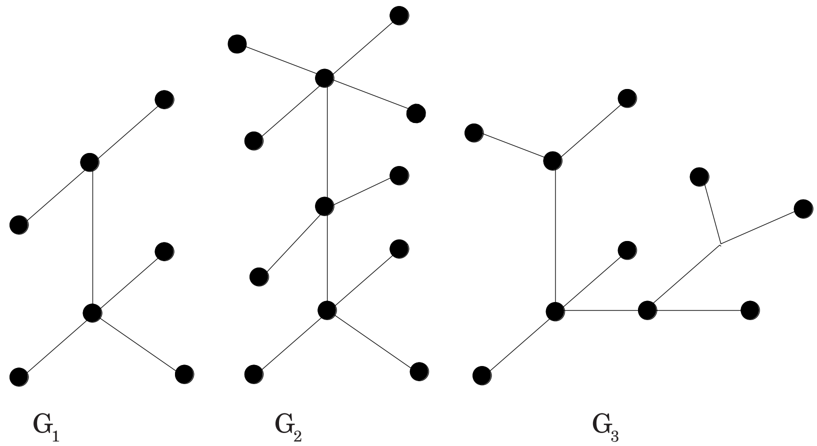
((2 x 3) + (4 x 2)) – (1 + 5) = 8

Perhatikan operator – paling atas disebut akar pohon, operator-operator di bawahnya disebut titik cabang pohon, bilangan-bilangan disebut daun pohon.

Hubungan antara pohon, titik, dan rusuk dapat dinyatakan sebagai:

Banyak rusuk = Banyak titik – Banyak pohon.

Contoh



Dari diagram pohon G_1, G_2, dan G_3, di atas dapat diketahui bahwa:

Jumlah pohon = 3 yaitu

G_1 , jumlah titik = 7 , jumlah rusuk 6
G_2 , jumlah titik = 12 , jumlah rusuk 11
G_3 , jumlah titik = 11 , jumlah rusuk 10
Jadi banyak titik seluruhnya = 30
banyak rusuk seluruhnya = 27
banyak pohon = 3

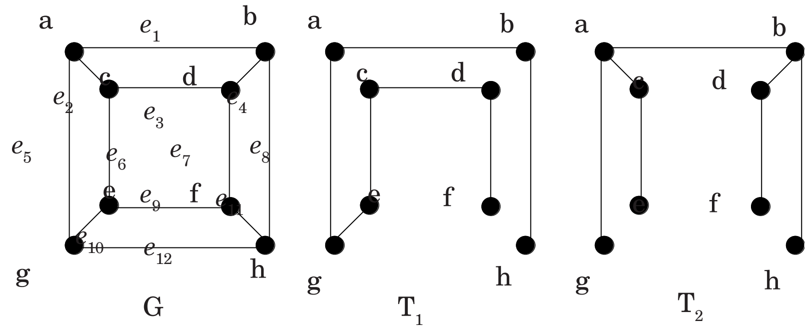
sehingga

$$27 = 30 - 3$$

8.13.1 Spanning Tree

Sebuah pohon katakanlah T disebut spanning tree dari sebuah graph G, jika T adalah subgraph dari G yang mencakup semua titik graph G.

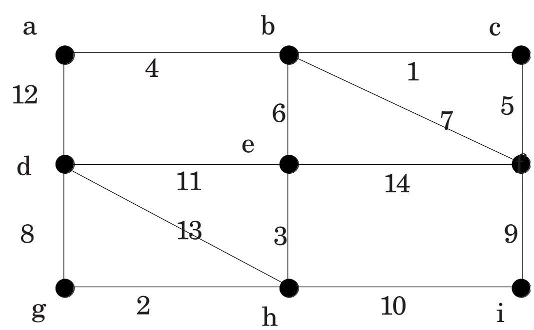
Contoh:



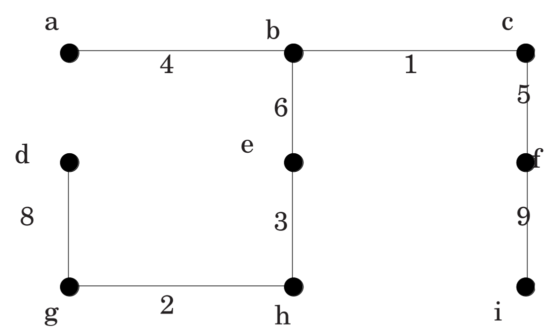
T_1, T_2, ... adalah spanning tree dari graph G, karena T_1, T_2, merupakan subgraph dari G yang mencakup semua titik graph G.

Minimal Spanning Tree

Misalkan pada graph di bawah titik-titik merepresentasikan kota dan rusuk merepresentasikan jaringan jalan raya yang akan dibangun dengan bobot/label merepresentasikan rencana biaya antar kota, maka untuk mencari biaya minimal rencana pembuatan jalan yang menghubungkan semua kota, kita memerlukan minimum spanning tree.



Graph rencana biaya pembuatan jaringan jalan raya yang menghubungkan 9 kota.



Minimal spanning tree, menggambarkan jaringan jalan raya yang menghubungkan 9 kota dengna biaya minimum = 38

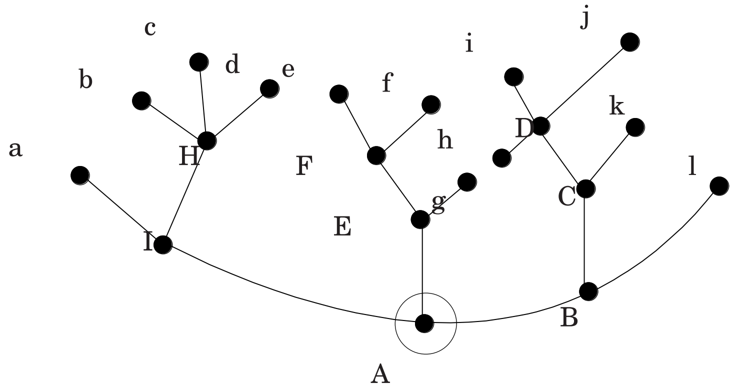
8.13.2 Pohon Berakar (Rooted Tree)

Seperti pohon alami pohon dalam graph juga mempunyai akar, cabang, dan daun.

Akar pohon adalah titik yang indegreenya nol (titik sumber). Setiap titik dapat dianggap atau dijadikan akar, titik yang dianggap sebagai akar ditandai dengan lingkaran yang mengelilingi titik tersebut.

Daun pohon adalah setiap titik (bukan akar) yang indegreenya 1 dan outdegreenya 0 (rink).

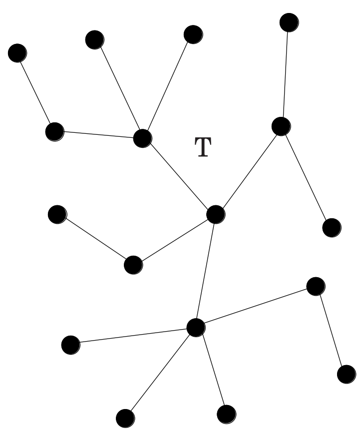
Tinggi pohon adalah panjang rusuk maksimum dari akar sampai daun.



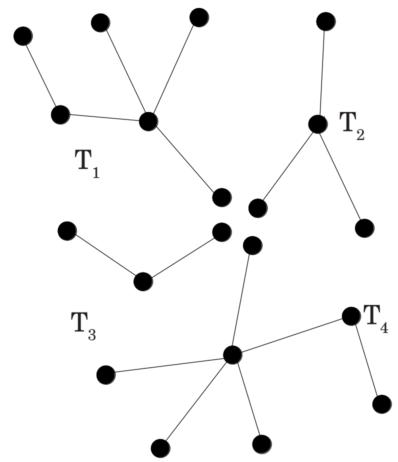
A = akar pohon
B, C, D, E, F, G, H, I = titik cabang
a, b, ..., k, l = daun
Tinggi pohon = 4, yaitu dari A ke j

a) Warnailah graph di atas dengan pewarnaan titik, rusuk, dan region serta tuliskan masing - masing bilangan kromatisnya.
b) Tentukan pusat dan sentral graph

Sebuah pohon dapat dipotong pada sembarang titik cabangnya menjadi dua atau lebih sub pohon sesuai dengan banyaknya rusuk pada titik cabang tersebut.



Misalkan pohon di samping dipotong pada titik cabang T, maka akan terjadi 4 sub pohon baru karena titik T mempunyai 4 rusuk, yaitu:

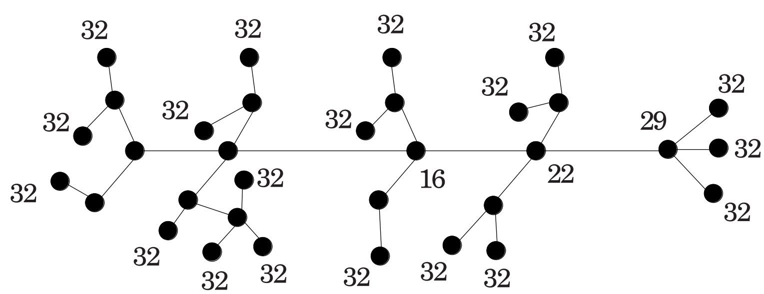


T_1 dengan 5 rusuk
T_2 dengan 3 rusuk
T_3 dengan 2 rusuk
T_4 dengan 6 rusuk

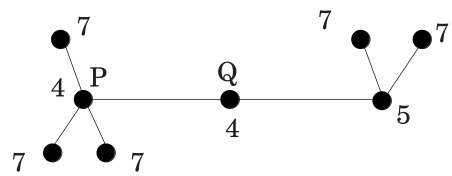
Sehingga berat pohon dititik T dapat diketahui adalah 6, karena berat pohon di suatu titik adalah jumlah rusuk maksimum dari semua cabang di titik tersebut.

Titik berat pohon adalah titik di mana berat pohon di titik tersebut minimum, jika titik berat pohon jumlahnya lebih dari satu titik maka himpunan titik berat tersebut disebut pusat berat.

Contoh:



Titik berat pohon di atas adalah 16, karena 16 adalah berat minimum dari semua titik yang ada.

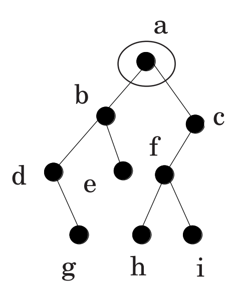


Titik berat pohon di atas adalah 4, yaitu di titik P dan Q, dan pusat berat pohon = {P, Q}.

Pohon binary adalah jenis pohon berakar yang penting, bagi setiap orang yang mempelajari teknologi informasi, karena pada umumnya karya mereka direpresentasikan dengan pohon binary, dimana pada pohon binary setiap titik cabang pohon dapat mempunyai:

- dua anak cabang, satu kekiri dan satu kekanan.
- satu anak cabang, atau
- tidak mempunyai anak cabang.

Contoh:



Titik b dan f mempunyai 2 anak cabang, titik d dan c mempunyai satu anak cabang titik g, h, dan i tidak mempunyai anak cabang.

Pohon binary dikatakan full bila setiap titiknya mempunyai dua anak cabang atau tidak punya anak cabang.

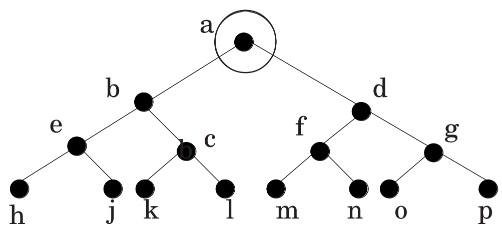
Dalam pohon binary dikenal istilah-istilah:

- titik internal adalah titik yang mempunyai 2 anak cabang
- titik terminal adalah titik yang tidak mempunyai anak cabang (daun).

Theorema:

Bila sebuah pohon binary full dan mempunyai i titik internal, maka titik terminalnya ada sebanyak i + 1 dan jumlah semua titiknya ada sebanyak 2i + 1.

Contoh:

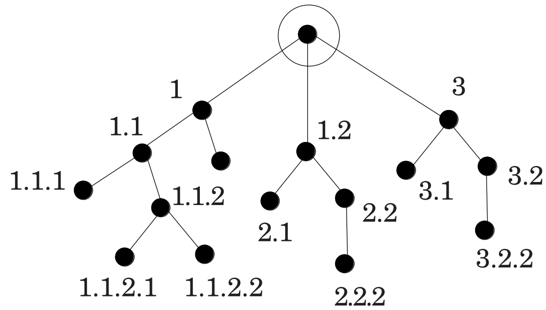


Titik internal adalah a, b, c, d, e, f, dan g, jadi i = 7
Titik terminal adalah h, j, k, l, m, n, o, dan p, jadi ada 8 atau i +1.
Jumlah seluruh titik ada 15 titik atau 2i + 1

8.13.3 Pohom Berurut Berakar (*Orderd Rootes Tree*)

Pohon berurut berakar adalah pohon berakar yang diberi label secara berurut dan sistematis, dimulai dari akar sebagai source/sumber/titik awal, semua cabang dari akar diberi nomor urut 1, 2, 3, ... sesuai dengan banyaknya cabang. Kemudian pada cabang 1 kita telusuri sampai ketemu anak cabang dan kita beri nomor 1.1, 1.2, 1.3, ... sesuai banyaknya anak cabang. Demikian seterusnya sampai seluruh titik bernomor, sistem demikian disebut universal adress system.

Contoh:



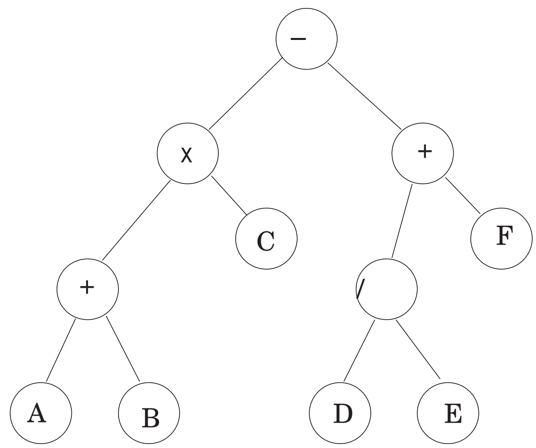
Gambar pohon berurut berakar di atas disebut Lexicographic order. Lexicographic order dapat digunakan untuk menggambarkan pernyataan aritmatika sebagai berikut:

Misalkan (A + B) x C – D/E +F akan kita gambar dalam bentuk Lexicographic maka pertama-tama harus kita tentukan dulu letak akar dengan cara mengikuti aturan urut-urutan hitung matematika yaitu x, /, +, –, sehingga pernyataan aritmatika di atas dapat ditulis kembali sebagai berikut:

((A +B) x C) – ((D/E) +F)

Sekarang dapat kita ambil titik akarnya yaitu operator – , ruas kiri dari operator – adalah cabang kiri dan ruas kanan dari operator – adalah cabang kanan.

Langkah berikutnya kita cari titik anak cabang di ruas kiri, ternyata operator x, operators x memisahkan (A + B) di ruas kiri dan C di ruas kanan. Di ruas kiri operator x ada anak cabang terakhir, yaitu operator + yang memisahkan A di ruas kiri dan B di ruas kanan, cara yang sama berlaku juga untuk cabang sebelah kanan.



Setelah kita mampu menggambarkan pernyataan aritmatika dalam bentuk Lexicographic, selanjutnya kita belajar menuliskan pernyataan aritmatika tersebut dalam susunan Lukasiwicz, yaitu bentuk prefix dan postfix.

Bentuk prefix adalah cara menuliskan pernyataan aritmatik dengan meletakkan simbol operator sebelum argumen, dimulai dari akar, ke cabang kiri, ke cabang kiri dan seterusnya sampai selesai baru pindah ke cabang kanan. Jadi bentuk prefix dari Lexicographic di atas adalah:

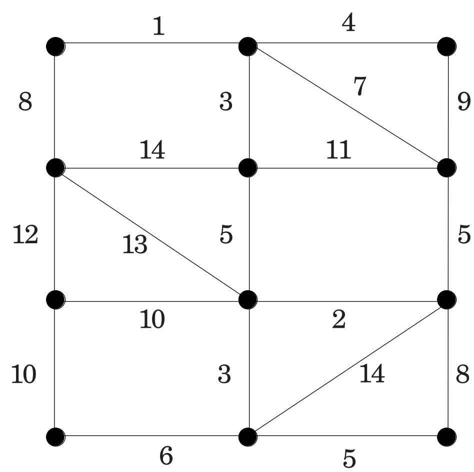
– x + ABC +/DEF

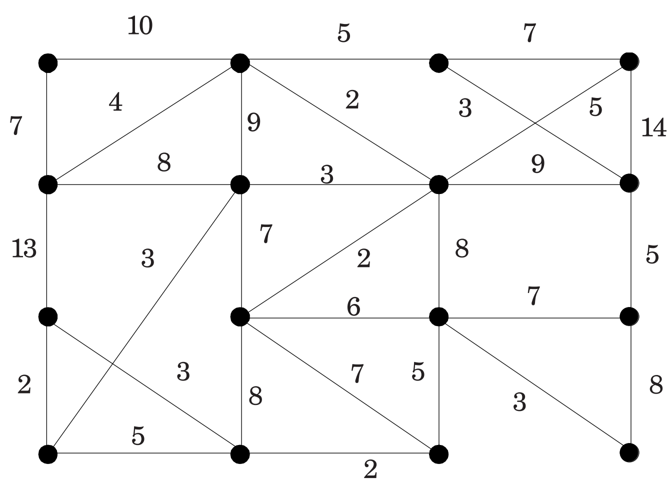
Bentuk postfix adalah cara menulis pernyataan aritmatika dengan meletakkan simbol operator sesudah argumen atau dengan cara nulis dari sebelah kanan ke kiri (seperti tulisan arab), dimulai dari akar, ke cabang kanan, ke cabang kanan dan seterusnya sampai selesai baru pindah ke cabang kiri. Jadi bentuk post fix dari Lexicographic di atas adalah:

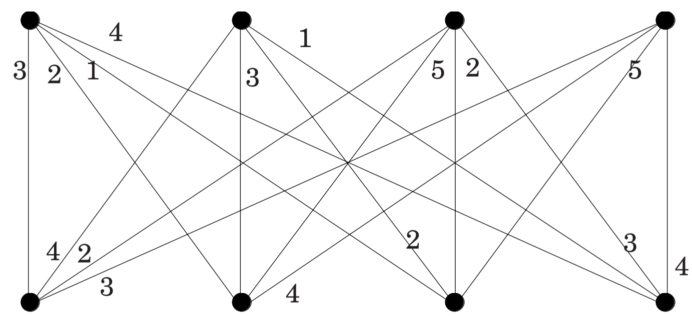
AB + C x DE/F+-

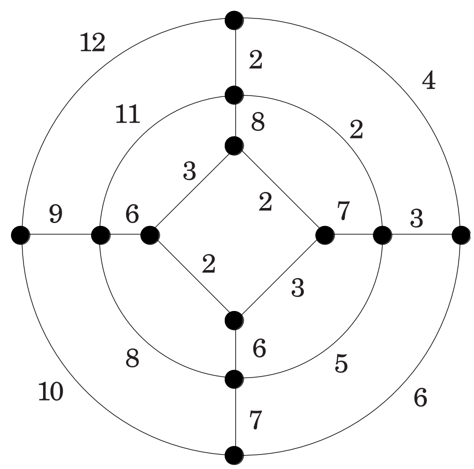
SOAL-SOAL

1. Cari dan gambarkan spanning tree minimum dari graph di bawah:









2. Gambarkanlah Lexicographic dari pernyataan aritmatika di bawah, kemudian tuliskan bentuk prefix dan postfixnya.
   a) (–7 + 6) x 3 – (4/2) + 5 – 2
   b) (5 x 4 x 3 x 2) – (7 + 3 x 5 – 6)
   c) (–3 x –5) / (5 – 3 + (–3 x –8) + (6 – 2 x –1 + 3))
   d) (A x B – C / D + E) + (A – B – C – D x D) / (A + B + C + D)
3. Gambarkan Lexicographic dari postfix form dibawah, kemudian tuliskan bentuk prefixnya.

a) AB + CD x EF / – – D x
b) ABC x x CDA + / –
c) AB – C + DBCA – x – +
d) ABCDE + x – /

4. Gambarkan Lexicographic dari prefix form di bawah, kemudian tuliskan bentuk postfixnya.
   a) + x – – x + x ABCBDC – E – C
   b) + x – AB – x CB + x DC – E – C
   c) – + x AB x CDA
   d) – + x AB x CA + DE
5. Di suatu wilayah baru akan dibangun 9 (sembilan) pemukiman yang masing-masing dinamai P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8, dan P_9. Selanjutnya akan dibangun jaringan jalan raya sebagai prasarana perhubungan antar daerah pemukiman itu. Hasil survei panjang jalan raya (km) yang harus dibangun tercantum dalam tabel berikut:

Daerah pemukiman	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9
P_1	0	80	60	90	*	120	*	*	*
P_2		0	90	*	110	*	160	*	*
P_3			0	50	70	*	*	*	*
P_4				0	80	70	*	100	*
P_5					0	*	60	140	*
P_6						0	*	80	*
P_7							0	100	50
P_8								0	70

* berarti tidak mungkin dibuat jalan raya karena alasan geografis

a) Jika besarnya biaya pembangunan jaringan ini dianggap sebanding dengan panjang jalan raya yang akan dibangun, maka tentukn bentuk jaringan dengan biaya minimum agar tidak ada daerah pemukiman terpencil.
b) Tentukan panjang jalan raya total untuk kasus diatas.

6. Seorang ahli jaringan mendapat order membangun jaringan internet di gedung perkantoran 25 lantai dengan sistim kabel. Bila rancangan jaringan setiap lantai sama dan koneksi antar lantai memerlukan kabel sepanjang 5 m, berapa panjang kabel minimum yang harus disiapkan bila rancangan jaringan tiap lantai digambarkan dengan matrik titik di bawah, dimana elemen matrik menyatakan jarak antar terminal dalam meter.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
A				7	10	15	12							
B				8	9	13	10							
C				8	11	14	9							
D	7	8	8					14	12	16				
E	10	9	11					13	14	15				
F	15	13	14					20	15	12				
G	12	10	6					25	18	14				
H				14	13	20	25				16	18	20	24
I				12	14	15	48				20	16	13	15
J				13	15	12	14				22	18	21	19
K								16	20	22				
L								18	16	18				
M								20	13	21				
N								24	15	19				

7. Kerjakan seperti no.6, bila rancangan jaringan setiap lantai seperti matrik di bawah:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
A		13			15											
B	13		8		8	10	6									
C		8		10				12								
D			10				8	16								
E	15	8				14			18							
F		10			14		8			15			18			
G		6		8		8		13		5	11					
H			12	16			13					9				
I					18								6	10		
J						15	5				13			14	11	
K							11			13		15			10	12
L								9			15					18
M						18			6					10		
N									10	14			10		5	
O										11	10			5		
P											12	18				

-oo0oo-

9

MESIN MATEMATIKA

9.1 PENDAHULUAN

Dalam bab VII telah kita bahas tentang rangkaian logika dimana outputnya hanya tergantung pada inputnya, jadi dapat disimpulkan bahwa dalam rangkaian logika tidak ada memory.

Dalam bab ini akan dibahas tentang rangkaian kombinasi yang outputnya tidak hanya tergantung dari input tetapi juga tergantung dari keadaan/state mesin pada waktu input data. Keadaan/state mesin sebelum kita input data ditentukan oleh proses data sebelumnya di dalam mesin tersebut. Dalam hal ini rangkaian kombinasi tersebut memiliki memory dan disebut mesin matematik.

Naom Chomsky (1950) menciptakan model matematika sebagai sarana untuk mendeskripsikan bahasa serta untuk mencoba menjawab pertanyaan - pertanyaan :

– apakah bahasa secara umum?
– bagaimana manusia mengembangkan bahasa?
– bagaimana manusia memahami bahasa?
– apa gagasan - gagasan yang dapat dinyatakan dan bagaimana caranya?

– bagaimana manusia membangun kalimat - kalimat dari gagasan gagasan yang berbeda dipikirannya?

Naom Chomsky mengemukakan perangkat format yang disebut grammar untuk memodelkan properti - properti bahasa, grammar berisi sejumlah aturan yang menspesifikasikan bahasa tertentu, bahasa berisi semua string yang dapat dihasilkan dengan menggunakan aturan - aturan grammar.

Grammar sangat bermanfaat bagi ilmu informatika / komputer, karena dengan grammar kita dapat mendeskripsikan dan mendefinisikan sintaks bahasa pemrograman dan bahasa - bahasa formal lainnya, merancang kompilator dan lain - lain, dengan bantuan mesin abstrak sederhana atau mesin matematika.

Definisi mesin matematika:

- Mesin matematika adalah model-model komputasi secara matematis.
- Mesin matematika adalah dasar-dasar dari komputer modern yang berfugsi untuk menjalankan program.
- Mesin matematika merupakan model komputasi yang dapat berfungsi sebagai acceptor dan tranducer sebagai acceptor artinya bila kita memberi input maka respon mesin tersebut menolak atau menerima.
  Contoh: pemakaian dari mesin ini yaitu pada mesin ATM, ponsel ataupun PC ketika kita diminta memasukan nomer PIN/Password.

  Sebagai trasducer artinya bila kita memberi input maka mesin tersebut akan memberi outputan.

Contoh:

Mesin penjumlah, inputannya dua bilangan dan outputnya jumlah kedua bilangan tersebut; vending machine, inputannya koin dan barang pilihan, outputannya berupa barang pilihan dan mungkin uang kembaliannya.

9.2 FINITE STATE AUTOMATA (FSA)

Ditinjau dari tingkat kesulitannya mesin matematika dibagi menjadi 3 kategori yaitu:

- Finite Automata (FA)
- Push Down Automata (PDA)
- Turing Machine (TM)

Dalam buku ini kita hanya akan membahas jenis yang paling mudah yaitu FA. Finite State Automata sering disebut Finite Automata adalah mesin abstrak berupa model metematika dengan masukan dan keluaran diskrit serta dapat mengenal bahasa paling sederhana (bahasa regular). Finite Automata memiliki state yang banyaknya berhingga dan memiliki fungsi transisi yang mendeskripsikan perpindahan dari sebuah state ke state yang lain. Finite Automata tidak memiliki tempat penyimpanan sehingga kemampuan mengingatnya terbatas.

Mekanisme kerja Finite Automata banyak diaplikasikan seperti pada:

- Sistim elevator
- Mesin jaja (Vending Mechine)
- Pengatur lampu lalu lintas (traffic light regulator)
- Sirkuit penyaklaran(switching) di komputer dan telekomunikasi
- Protokol komunikasi
- Analisis Leksikal
- Neuron Nets
- Pengecek parity (perity checking)

FA dapat digambarkan dalam 3 cara, yaitu:

- Diagraph
- Notasi Formal 5 – tuple
- Tabel State

9.2.1 Menggambarkan FA dengan Digraph

Seperti telah dibahas dalam bab 8 bahwa graph adalah himpunan titik dan rusuk, maka menggambar FA dengan digraph berarti menggambarkan FA dengan titik dan rusuk berarah

Titik menggambarkan state, yaitu state menerima ditandai dengan lingkaran ganda dan state menolak ditandai dengan lingkaran tunggal.

Rusuk berarah menggambarkan transisi/perpindahan, bila pada state tertentu mendapat input tertentu maka pindah ke state yang ditunjuk oleh arah dari rusuk berarah. Label menggambarkan inputan yang diberikan pada state tertentu, rusuk berarah tanpa label menunjukkan di state mana pertama-tama kita bekerja (initial state).

Contoh:

FA untuk parity cheking.

Dalam media penyimpan data magnetik tape data sering digambarkan dalam barisan bilangan binary 0 dan 1 dalam sebuah frame. Sebuah frame terdiri dari 9 bits, dimana 8 bits menggambarkan data/karakter dan 1 bits di depan sebagai check bit.

0	0	1	0	0	1	0	0	1

check bit data bits

Sebuah frame dari tape dengan labeled bits

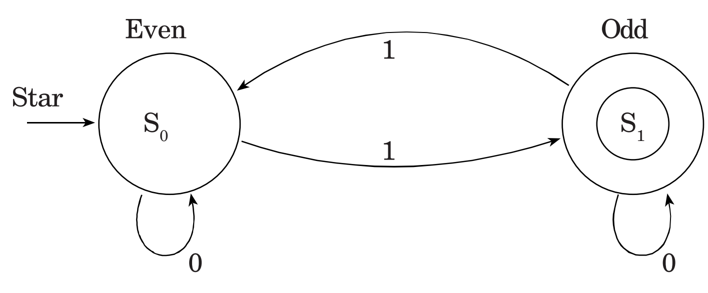
Data bit dari frame diatas adalah 0 1 0 0 1 0 0 1 yang identik dengan $0 \times 2^7 + 1 \times 2^6 + 0 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 64 + 8 + 1 = 73$.
Check bit bernilai 0 bila jumlah bit '1' dalam data bits berjumlah ganjil.
Check bit bernilai 1 bila jumlah bit '1' dalam data bits berjumlah genap.

Bagai mana kita membuat FA dengan digraph sebagai model matematis untuk persoalan check bit ini?

Pertama kita perhatikan state dalam persoalan check bit ini, ada 2 state yaitu jumlah bit '1' ganjil dan genap. Jadi ada 2 buah titik yaitu titik ganjil dan genap. Langkah *kedua* kita tentukan state menerima dan state menolak, menerima ditandai dengan lingkaran ganda dan state menolak ditandai dengan lingkaran tunggal. Langkah *ketiga* kita identifikasi inputannya, inputnya 0 dan 1 artinya setiap state akan mendapat input 0 dan 1, bila mendapat input 0 akan bertransisi kemana dan bila dapat input 1 bertransisi kemana. Karena statenya hanya dua, maka transisi yang mungkin adalah transisi ke state yang lain atau kedirinya sendiri.

Langkah *keempat* kita tentukan mulai dari mana kita masuk atau proses dimulai dari mana (initial state) ditandai dengan panah tanpa label.

Jadi FA untuk odd parity adalah



Untuk frame 0 0 1 0 0 1 0 0 1

Data	pertama	0, So	transisi ke So
Data	kedua	0, So	transisi ke So
Data	ketiga	1, So	transisi ke S1
Data	keempat	0, S1	transisi ke S1
Data	kelima	0, S1	transisi ke S1
Data	keenam	1, S1	transisi ke So
Data	ketujuh	0, So	transisi ke So
Data	kedelapan	0, So	transisi ke So
Data	kesembilan	1, So	transisi ke S1

Jadi data string tersebut berhenti pada lingkaran ganda artinya string tersebut diterima, dan paritynya benar berisi 0 (jenis odd parity).

9.2.2 Menggambarkan FA dengan definisi Formal 5-Tuple

Dalam definisi formal FA dinotasikan dengan 5 tuple yaitu S, S, d, So dan A, dimana

S adalah himpunan state- state

S adalah himpunan simbol-simbol input

d adalah fungsi transisi, contoh d $(S_{0,}1) = S_1$, artinya S_0 mendapat input 1 bertransisi ke S_1

S_0 adalah state awal (initial state)

A adalah himpunan state penerima (accepting state)

Jadi masalah odd parity di atas dapat ditulis dengan definisi formal sebagai berikut:

$$\begin{aligned} &S = \{S_0, S_1\} && d(S_0,0) = S_0 \\ &\sum = \{0, 1\} && d(S_0,1) = S_1 \\ &S_0 = S_0 && d(S_1,0) = S_1 \\ &A = \{S_1\} && d(S_1,1) = S_0 \end{aligned}$$

9.2.3 Menggambarkan FA dengan Tabel State

Menggambarkan FA dengan tabel state adalah membuat matrik dari FA tersebut, dimana baris menggambarkan transisi state dengan bermacam input dan kolom menunjukan transisi state dengan sebuah input. State menerima dibedakan dengan state menolak yaitu state menerima diberi tanda*. Jadi masalah odd parity diatas dapat digambarkan dengan tabel state sebagai berikut:

	0	1
S_0	S_0	S_1
S_1^*	S_1	S_0

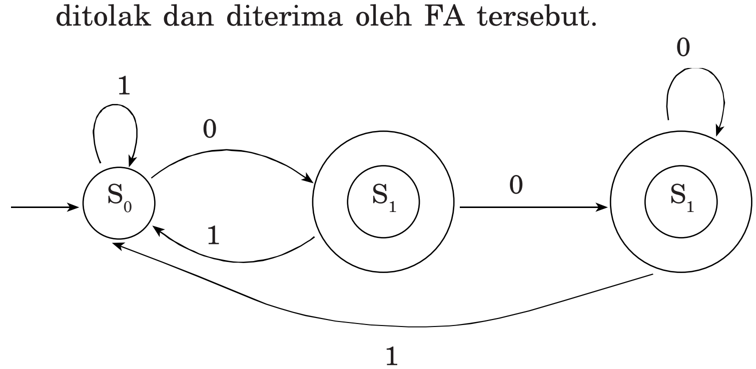
9.2.5 Non-Deterministik Finite Automata (NFA)

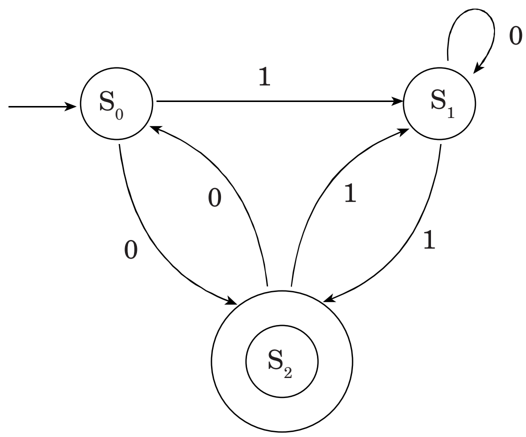
Semua aturan yang berlaku dalam deterministric fininte automata atau sering disebut finite automata, berlaku juga dalam non-deterministic finite automata, bedanya NFA memberikan kemungkinan lebih dari satu transisi untuk setiap input yang kita berikan pada sebuah state.

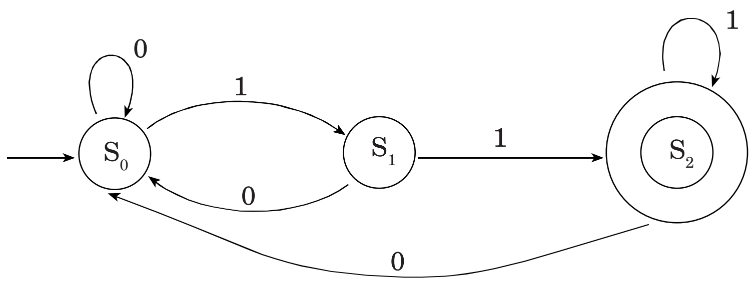
Latihan

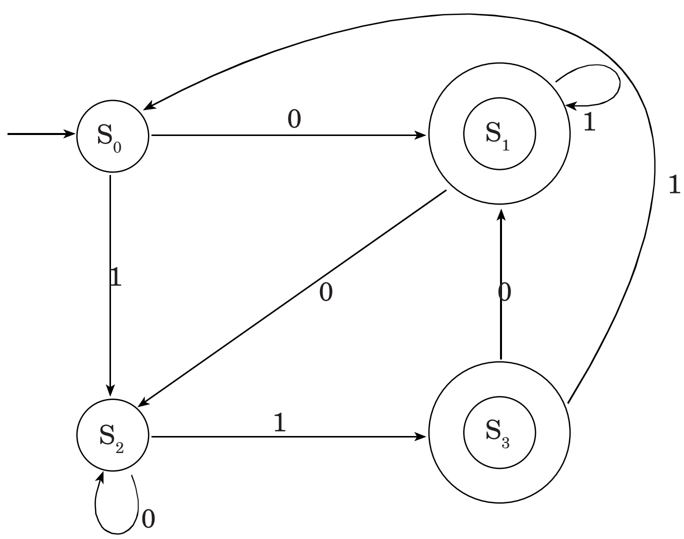
1a. Gambarkanlah FA di bawah dengan tabel state dan definisi formal.

1b. Berikan masing-masing sebuah contoh string yang ditolak dan diterima oleh FA tersebut.




S0
S1
S2
1
0
0
0
1
1


S0
S1
S2
0
1
1
1
0
0


S0
S1
S2
S3
0
1
1
1
0
0
1
0

2a. Gambarkan kembali FA dalam bentuk tabel state di bawah dengan digraph dan definisi formal

2b. Berikan masing-masing sebuah contoh string yang ditolak dan diterima oleh FA tersebut.

	0	1
$S_0{}^*$	S_1	S_0
S_1	S_2	S_0
S_2	S_0	S_2

	0	1
$S_0{}^*$	S_0	S_1
$S_1{}^*$	S_0	S_2
S_2	S_0	S_1

	a	b	c
$S_0{}^*$	S_1	S_0	S_2
$S_1{}^*$	S_0	S_3	S_0
S_2	S_3	S_2	S_0
S_3	S_1	S_0	S_1

3a. Gambarkanlah kembali FA dalam bentuk definisi formal di bawah dengan digraph dan tabel state

3b. Berikanlah masing-masing sebuah contoh string yang ditolak dan diterima oleh FA tersebut

$$
\begin{array}{lll}
S = \{S_0, S_1, S_2, S_3, S_4\} & d(S_0,0) = S_0 & d(S_2,0) = S_3 \\
\sum = \{0,1\} & d(S_0,1) = S_1 & d(S_2,1) = S_2 \\
S_0 = S_0 & d(S_1,0) = S_0 & d(S_3,0) = S_0 \\
A = \{S_4\} & d(S_1,1) = S_2 & d(S_3,1) = S_4 \\
 & & d(S_4,0) = S_4 \\
 & & d(S_4,1) = S_4
\end{array}
$$

$$
\begin{array}{lll}
S = \{S_0, S_1, S_2\} & d(S_0,0) = S_0 & d(S_2,0) = S_2 \\
\sum = \{0,1\} & d(S_0,1) = S_1 & d(S_2,1) = S_2 \\
S_0 = S_0 & d(S_1,0) = S_0 & \\
A = \{S_2\} & d(S_1,1) = S_2 &
\end{array}
$$

$S = \{S_0, S_1, S_2, S_3, S_4\}$	$d(S_0,0) = S_3$	$d(S_2,1) = S_0$
$\sum = \{0,1\}$	$d(S_0,1) = S_1$	$d(S_3,0) = S_1$
$S_0 = S_0$	$d(S_1,0) = S_1$	$d(S_3,1) = S_4$
$A = \{S_0, S_4\}$	$d(S_1,1) = S_1$	$d(S_4,0) = S_3$
	$d(S_2,0) = S_1$	$d(S_3,1) = S_1$

4. Gambarkan kasus di bawah dengan diagraph finite automata:
   Seorang pemburu ingin membawa seekor rusa hasil buruannya menyebrangi sebuah sungai, bersama pemburu itu juga dibawa seekor anjing pemburu dan seikat rumput untuk persediaan makanan sirusa, anjing dan rusa tidak bisa tinggal berdua demikian juga dengan kambing dan seikat rumput, sedangkan perahu yang tersedia hanya mampu membawa si pemburu dan salah satu bawaannya

5. Sebuah mesin jaja hanya dapat menerima inputan 500 rupiah dan 1000 rupiah, dan hanya dapat memberikan outputan sejenis barang senilai 2000 rupiah. Gambarkan finite state dengan segala kemungkinan input pada mesin jaja tersebut dengan :
   diagraph, tabel state, dan definisi formal.

6. Sebuah mesin telepon koin hanya dapat menerima inputan koin 500 rupiah dan 1000 rupiah dan dapat memberi outputan durasi bicara :
   2 menit bila jumlah input 1000 rupiah
   5 menit bila jumlah input 1500 rupiah
   10 menit bila jumlah input 2000 rupiah

   Gambarkan dengan diagraph, tabel state, dan definisi formal finite state automata mesin telepon koin tersebut dengan segala kemungkinan input yang dapat terjadi, bila aturan mesin tersebut : “pilih durasi bicara kemudian masukkan koin serta tidak ada uang kembali ” (dengan diagraph, tabel state, dan definisi formal)

7. Elevator Controller mempunyai tabel state sebagai berikut :

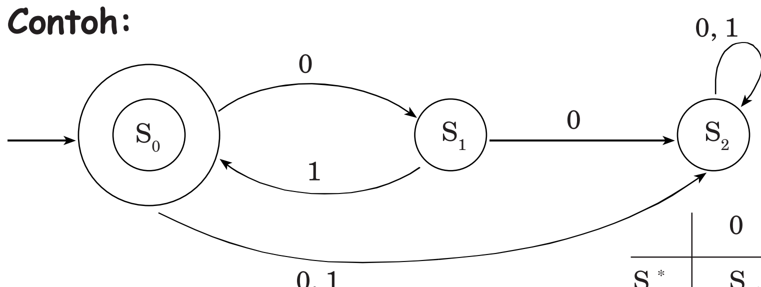
State	input 0	1	2
W1 (wait on 1)	W1	W1	UP
U1 (start up)	UP	U1	UP
UP (going up)	W2	D2	W2
DN (going down)	W1	W1	U1
W2 (wait on 2)	W2	DN	W2
D2 (start down)	DN	DN	D2

gambarkan diagraph dan definisi formal dari elevator controller tersebut.

9.2.5 Non-Deterministik Finite Automata (NFA)

Semua aturan yang berlaku dalam deterministric fininte automata atau sering disebut finite automata, berlaku juga dalam non-deterministic finite automata, bedanya NFA memberikan kemungkinan lebih dari satu transisi untuk setiap input yang kita berikan pada sebuah state.

Contoh:



	0	1
S_0^*	S_1, S_2	S_2
S_1	S_2	S_0
S_2	S_2	S_2

Perhatikan NFA diatas, pada state So bila mendapat input 0, maka fungsi transisinya bisa ke S1 bisa ke S2

Bahasa/languages yang diterima oleh NFA tersebut adalah:

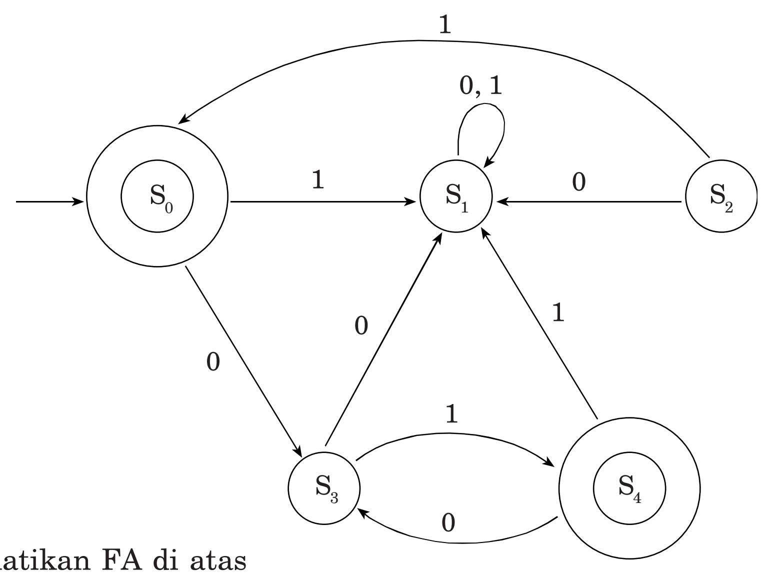
$$L\ (M) = \{e,\ 0\ 1,\ 0\ 1\ 0\ 1,\ 0\ 1\ 0\ 1\ 0\ 1,\ ....\}$$

dimana e adalah empty string

Difinisi

Bila M adalah sebuah NFA, maka ada sebuah deterministik finite automata yang menerima L(M) dari NFA tersebut (mesin ekivalen).

Deterministic finite automata yang menerima L (M) dari NFA di atas adalah:



Perhatikan FA di atas

Empty string (e) diterima

String 0 1 diterima

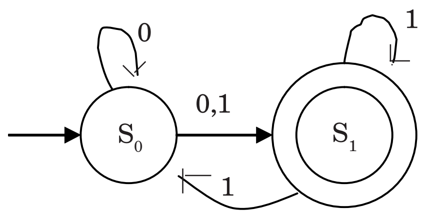
String 0 1 0 1 diterima

String 0 1 0 1 0 1 diterima dan seterusnya

Jadi L (M) = {e, 0 1, 0 1 0 1, 0 1 0 1 0 1, ...}

Latihan

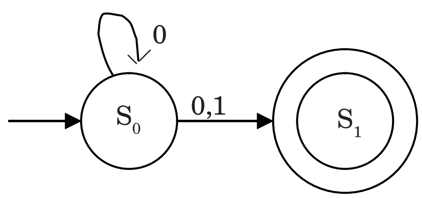
1. Sebuah NFA digambarkan dengan diagraph dan tabel state sebagai berikut



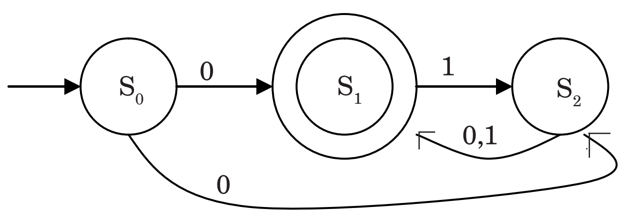
	0	1
S_0	S_0,S_1	S_1
S_1*	ϕ	S_0,S_1

Gambarkanlah sebuah FA yang ekivalen (dapat menerima bahasa yang sama) dengan NFA diatas.

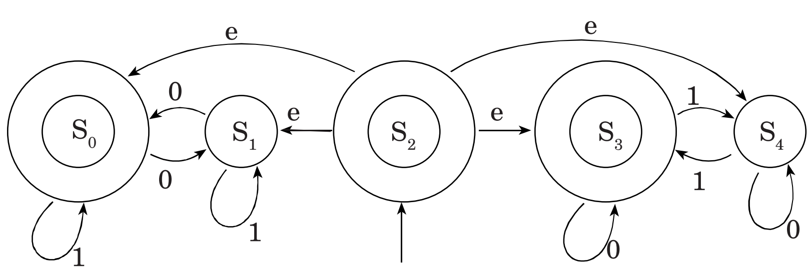
2. Kerjakan seperti no.1 unuk NFA



3. Kerjakan seperti no.1 unuk NFA



4. Tentukan L (M) dari NFA di bawah kemudian buatlah sebuah FA yang menerima L (M) tersebut.



5. Gambarkan dengan digraph NFA yang dinyatakan dengan tabel di bawah, kemudian tuliskan kembali dengan definisi formal

	a	b
S_0^*	ϕ	S_1, S_2
S_1	S_2	S_0, S_1
S_2	S_0	ϕ

	a	b
S_0	ϕ	S_3
S_1^*	S_1, S_2	S_3
S_2	ϕ	S_0, S_1, S_3
S_3	ϕ	ϕ

	a	b	c
S_0^*	S_1	ϕ	ϕ
S_1^*	S_0	S_2	S_0, S_2
S_2	S_0, S_1, S_3	S_0	S_0

	a	b	c
S_0^*	S_1	S_0, S_1, S_3	ϕ
S_1	S_1, S_3	ϕ	ϕ
S_2	ϕ	S_0, S_3	S_1, S_2
S_3	ϕ	ϕ	S_0

6. Tentukan L(M) dari masing - masing NFA pada soal No.5, kemudian buatlah sebuah FA yang menerima L(M) dari NFA tersebut.
7. Tuliskan NFA pada soal nomer 4 kedalam bentuk definisi formal dan tabel state.
8. Buatlah FA untuk mengenal deklarasi 'variable' pada pascal.
9. Buatlah FA untuk mengenal identifier dalam Pascal/C seperti; Total, Sum1, Sum2, dan sebagainya.

9.2.6 Finite State Transducers

Seperti telah kita pelajari di atas bahwa output dari sebuah FA sangat terbatas, yaitu menerima atau menolak artinya sebuah string yang kita input kesebuah FA bisa diterima/accepted atau ditolak/rejected. Jadi walaupun FA sangat bermanfaat sebagai pengertian dasar dalam menganalisa bahasa, namun tidak dapat digunakan untuk menjelaskan suatu proses translasi dari satu bahasa ke bahasa yang lain. Sebuah FA dapat di modifikasi sehingga dapat berfungsi sebagai translator, FA yang demikian disebut Finite tranducer atau FA dengan output.

Finite state transduser dapat digambarkan dengan:

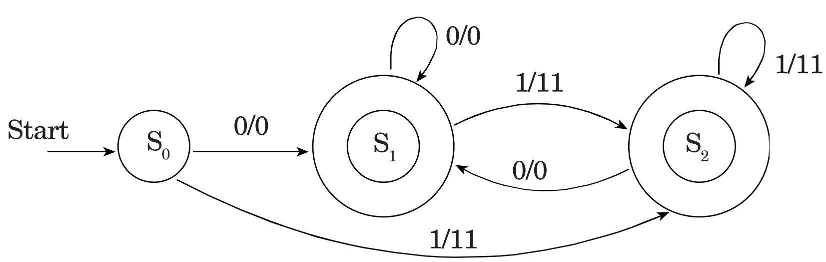
- digraph
- tabel state

9.2.6.1 Menggambarkan finite state transducer dengan digraph

Misalkan sebuah finite transducer akan mentranslasikan sebuah bahasa ke bahasa yang lain, katakanlah

Bahasa asal		Bahasa baru
1 1 0 1	$\longrightarrow$	1 1 1 1 0 1 1

artinya finite transducer tersebut akan menyisipi '1' setiap dapat input '1', maka kita dapat menggambarkan proses traslasi di atas dengan digraph, berikut:



Perhatikan proses translasinya:

Start, So dapat input 1, outputnya 1 1, transisi ke S_2
S2 dapat input 1, outputnya 1 1, transisi ke S_2
S2 dapat input 0, outputnya 0, transisi ke S_1
S1 dapat input 1, outputnya 1 1, transisi ke S_2

Jadi string 1 1 0 1 diterima dan FA tersebut memberikan output 1 1 1 1 0 1 1.

9.2.6.2 Menggambarkan finite state transducer dengan tabel

Sama halnya dengan menggambarkan FA dengan tabel, menggambarkan funite state transducer dengan tabel hanya melakukan sedikit perubahan pada fungsi transisi d , dalam finite state transducer fungsi transisi ditambah dengan outputtan, seperti berikut.

	0	1
S_0	S_1, 0	S_2, 11
S_1^*	S_1, 0	S_2, 11
S_2^*	S_1, 0	S_2, 11

Latih:

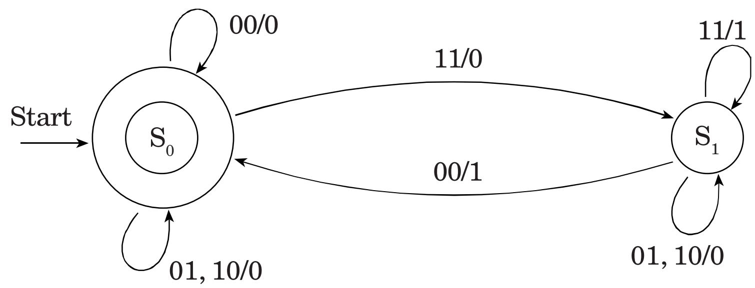
1. Buatlah sebuah finite state transducer yang mentranslasikan string 1 0 1 0 0 1 1 menjadi string 1 0 0 1 0 0 0 0 1 1 dalam bentuk digraph maupun table state
2. Buatlah sebuah finite trasducer yang mampu memberikan output hasil penjumlahan dua buah bilangan integer 5 (1 0 1) dan 7 (1 1 1).

Perhatikan

$$\begin{array}{r} 0\,1\,0\,1 \\ 0\,1\,1\,1 \\ \hline 1\,1\,0\,0 \end{array} +$$

dalam hal ini pada initial state input yang diberikan adalah 1 1 dan outputnya 0, input berikutnya adalah 0 1 atau 1 0 dan outputnya 0, input berikutnya adalah 1 1 dan outputnya 1, input terakhir adalah 0 0 dan outputnya 1.

Jadi finite transducernya dapat digambarkan dengan digraph sebagai berikut.



Start, S_1 dapat input 11, output 0, transisi ke S1
S_1 dapat input 01, output 0, transisi ke S1
S_1 dapat input 11, output1, transisi ke S1
S_1 dapat input 00, output1, transisi ke S0

Jadi hasil akhir dari proses penjumlahan tersebut adalah 1100

2b. Kerjakan dengan cara yang sama untuk 8 + 9, 6 + 5, dan 7 +8

3. Buatlah sebuah finite transducer yang mampu memberikan output hasil pengurangan dua buah bilangan integer:

   14 – 5, 16 – 9 dan 15 - 7

4. Buatlah sebuah finite state transducer yang mampu mentranslasi bilangan berbasis 2 (binary) menjadi bilangan berbasis 4.

-oo0oo-

DAFTAR PUSTAKA

Firrar Utdirartatmo, Teori Bahasa Dan Otomata, Graha Ilmu, Yogyakarta, Edisi 2,2005.

Jonhsonbaugh, Ricard, *Discrete Mathematics*. Prentice Hall Int, New Jersey, 2001.

Klin, George J dan Tina A. Folger, *Fuzzy Sets, Uncertainty and Information,* Prentice Hall Int, New Jersey, 1998.

Klin, George J, Ute H. St Clair dan Bo Yuan, *Fuzzy Sets Theory*, Prentice Hall Int, 1997.

Sri Kusumadewi, Hari Purnomo, Aplikasi Logika Fuzzy, Graha Ilmu,Yogyakarta, 2004.

Sumarna, Elektronika Digital, Graha Ilmu, Yogyakarta, 2006.

Witala, Stephen A. *Discrete Mathematics A Unified Approach*. Mc Graw Hill Int, Singapore, 1987.

-oo0oo-

2

TENTANG PENULIS

SAMUEL WIBISONO menyelesaikan pendidikan dasar dan menengahnya di Surakarta. Beliau menyelesaikan pendidikan di Akademi Meteorologi dan Geofisika Jakarta, serta lulus Jurusan Fisika Fakultas MIPA Universitas Indonesia, dan menyelesaikan Program Pasca-Sarjana bidang Optoelektronika dan Aplikasi Laser tahun 1997, di Universitas Indonesia.

Sejak tahun 1983 beliau mengabdi di Badan Meteorologi dan Geofisika dan terakhir sebagai dosen di Pusdiklat Meteorologi dan Geofisika. Beliau juga mengampu mata kuliah Matematika Diskrit sejak tahun 1997 di berbagai universitas antara lain Universitas Bina Nusantara Jakarta, Universitas Indonesia Esa Unggul Jakarta, Universitas Kristen Krida Wacana, AMIK BK3 Tangerang.