LOGIKA INFORMATIKA
TIFS 1604

Seputar Pelaksanaan Perkuliahan

Mata Kuliah Logika Informatika

Outline

- Deskripsi Mata Kuliah
- Materi kuliah
- Silabus
- Referensi
- Evaluasi
- Lain-lain

Deskripsi Mata Kuliah

- Matakuliah ini memberikan suatu metode atau cara yang sistematis dalam berpikir (reasoning). Terdapat dua metode cara berpikir yang digunakan, yaitu Logika Proposisi dan Logika Predikat. Dengan menggunakan logika, diharapkan dapat mengurangi tindakan menebak dalam menghadapi dan menyelesaikan suatu masalah sehingga masalah tersebut dapat diselesaikan dengan suatu jawaban yang dikerjakan dengan sistematis. Cara berpikir dengan dasar logika ini dapat dijadikan program dan dilaksanakan oleh komputer sehingga komputer dapat melakukan kemampuan "berpikir" walaupun secara sederhana.

Materi Kuliah

- Cakupan Materi
  - Konsep logika, sejarah dan peranannya dalam Teknik Informatika
  - Representasi bilangan dan operasi aritmatika bilangan
  - Kalkulus proposisi dan kalkulus predikatif
  - Teori himpunan
  - Fungsi dan Relasi

Silabus

Topik	Deskripsi Materi
Pendahuluan	Konsep logika; sejarah; peranan logika dalam ranah ilmu Teknik Informatika
Representasi Bilangan	Sistem bilangan biner; Sistem bilangan desimal, Sistem bilangan hexadesimal; Konversi bilangan; Aritmatika bilangan
Logika Proposisional	Preposisi; Variabel dan konstanta proposisi; Tabel kebenaran; Proposisi majemuk; Tautologi; Ekuivalensi; Hukum-hukum logika;
Logika Predikatif	Komponen logika predikatif; interpretasi dan validity; derivasi
Himpunan	Himpunan; Operasi himpunan; Tuples, sequences dan Powersets;
Relasi	Relasi; komposisi relasi; Property relasi
Fungsi	Fungsi, operasi terhadap fungsi; invers fungsi

Referensi

- Buku
  - Jean-Paul Tremblay., 1996, "Logic and Discrete Mathematics", Prentice Hall, New Jersey
  - F. Soesianto & Djoni Dwijono, 2003, "Logika Proposisional", Andi Offset, Yogyakarta
  - F. Soesianto & Djoni Dwijono, 2003, "Logika Predikatif", Andi Offset, Yogyakarta
  - Rinaldi Munir, 2003, "Matematika Diskrit", Edisi Ke-2, Informatika, Bandung

Evaluasi

- Komponen
  - Kehadiran dan partisipasi : 10 %
  - Tugas 1 :
  - Tugas 2 : } 25% (+quiz)
  - Ujian Tengah Semester : 20%
  - Tugas 3 :
  - Tugas 4 : } 25% (+quiz)
  - Ujian Akhir Semester : 20%

Lain-lain

- Mahasiswa harus aktif dalam proses pembelajaran
- Mahasiswa harus tepat waktu, toleransi keterlambatan 30 menit
- Dilarang keras berbuat curang dalam pengerjaan tugas maupun ujian
- Keterlambatan pengumpulan tugas tidak ditolerir

LOGIKA INFORMATIKA

Suraya

Jurusan Teknik Informatika

Materi Perkuliahan

- Konsep Logika, Sejarah dan Peranannya
- Bentuk Formal Logika dan Kaidah-kaidah Dasarnya
- Logika Proposisi
  - Bentuk Argumen dan validitasnya
  - Variabel dan Konstanta proposional
- Logical Connectives

Sumber Literatur

- Text Book:
  - Jong Jek Siang., Drs, MSc., 2002, *"Matematika Diskrit dan Aplikasinya Pada Ilmu Komputer"*, Andi, Yogyakarta
  - Rinaldi Munir, 2003, *"Matematika Diskrit"*, Edisi Ke-2, Informatika, Bandung
  - F. Soesianto, Djoni Dwijono, "Logika Proposisional", Andi, Yogyakarta

- Link
  - http://www.cise.ufl.edu/cot3100/lects/Module-1-Logic.ppt
  - http://informatika.org/~rinaldi/Buku/Matematika%20Diskrit/Bab-01%20Logika_edisi%203.pdf
  - http://www.cise.ufl.edu/cot3100/lects/Module-1-Logic.ppt

Konsep Logika

- Logika
  - ✓ Ilmu tentang metode penalaran yang berhubungan dengan pembuktian validitas suatu argumen
  - ✓ Suatu argumen yang berisi pernyataan harus diubah menjadi bentuk logika agar dapat dibuktikan validitasnya

Logika mengkaji hubungan antara pernyataan-pernyataan (statement)

- Semua pengendara sepeda motor memakai helm.
- Setiap orang yang memakai helm adalah mahasiswa.

Jadi, semua pengendara sepeda motor adalah mahasiswa.

Konsep Logika

Logika matematika adalah sebuah alat untuk bekerja dengan pernyataan (*statement)* majemuk yang rumit. Terimasuk di dalamnya:

- Bahasa untuk merepresentasikan pernyataan
- Notasi yang tepat untuk menuliskan sebuah pernyataan
- Metodologi untuk bernalar secara objektif untuk menentukan nilai benar-salah dari pernyataan
- Dasar-dasar untuk menyatakan pembuktian formal dalam semua cabang matematika

Sejarah Logika

Sejarah Logika

- Aristoteles (322 B.C) → Logika Tradisional atau Logika Klasik
- George Boole dan Augustus De Morgan (abad XIX) → Logika Modern atau Logika Simbolik
- Gottlob Frege, Bertrand Russel, Alfred North Whitehead, John Stuart (abad XX) → pengembangan Logika Modern

Peranan Logika

- Bidang Matematika
  - Komputasi
  - Matematika Diskret
  - Aljabar Linier
- Elektronika
  - Rangkaian Digital
- Ilmu Komputer / Informatika
  - Membuat dan menguji program komputer
  - Artificial Intelligence
  - Expert Systems
  - Logic Programming
  - Soft Computing (kumpulan teknik – teknik perhitungan dalam ilmu komputer)

Dasar-dasar Logika

- Ada suatu argumen yang secara logis kuat, tetapi ada juga yang tidak
- Argumen terdiri dari proposisi atomik yang dirangkai dengan Logical Connectives membentuk proposisi majemuk
- Jenis Proposisi
  - **Proposisi Atomik**
  - **Proposisi Majemuk**
- Contoh1 : argumen logis
  1. Jika harga gula naik, maka pabrik gula akan senang
  2. Jika pabrik gula senang, maka petani tebu akan senang
  3. Dengan demikian, jika harga gula naik, maka petani tebu senang
- Pernyataan (1) dan (2) disebut premis-premis dari suatu argumen dan pernyataan (3) berisi kesimpulan atau conclusion.

Jika suatu argumen memiliki premis-premis yang benar, maka kesimpulan juga harus benar.

Dasar-dasar Logika

- Contoh2 : argumen logis
  1. Program komputer ini memiliki bug, atau masukannya salah
  2. Masukannya tidak salah
  3. Dengan demikian, program komputer ini memiliki bug
- Contoh3 : argumen logis
  1) Jika lampu lalu lintas menyala merah, maka semua kendaraan berhenti
  2) Lampu lalu lintas menyala merah
  3) Dengan demikian, semua kendaraan berhenti
- Contoh4 : argumen logis
  1) Jika saya makan, maka saya kenyang
  2) Saya tidak makan
  3) Dengan demikian, saya tidak kenyang

Dasar-dasar Logika

- Hypothetical Syllogism (contoh 1)
  1) Jika A maka B
  2) Jika B maka C
  3) Jika A maka C → kesimpulan

- Disjunctive Syllogism (contoh2)
  1) A atau B
  2) Bukan B
  3) A → kesimpulan

Dasar-dasar Logika

- Modus Ponens (contoh3)
  1) Jika A maka B
  2) A
  3) B
- Modus Tolens (contoh4)
  - Jika A maka B
  - Bukan A
  - Bukan B

Logika Proposisi

- *Logika proposisi* adalah logika pernyataan majemuk yang disusun dari pernyataan-pernyataan sederhana yang dihubungkan dengan *penghubung Boolean* (*Boolean connectives)*
- Beberapa aplikasinya dalam ilmu komputer:
  - Merancang sirkuit elektronik digital
  - Menyatakan kondisi/syarat pada program
  - *Query* untuk basisdata dan program pencari (*search engine)*

Chrysippus of Soli
(ca. 281 B.C. – 205 B.C.)

George Boole
(1815-1864)

Logika Proposisi

- Jenis Proposisi
  - ✓ Proposisi Atomik
  - ✓ Proposisi Majemuk

Atomic proposition adalah proposition yang tidak dapat dibagi lagi

Kombinasi dari **Atomic proposition** dengan berbagai penghubung membentuk **compound proposition** (proposition majemuk)

Definisi Proposisi

- Sebuah *proposisi* (*p*, *q*, *r*, …) adalah suatu kalimat (*sentence)* yang memiliki nilai kebenaran (*truth value)* **benar** (*true*), dengan notasi T, atau nilai kebenaran **salah** (*false*) dengan notasi F tetapi **tidak kedua-duanya**
- (Namun demikian, kadang kita *tidak tahu* nilai kebenarannya karena kasusnya tergantung situasi, dalam kasus ini kita harus mengggunakan asumsi)

Perhatikan

a) 6 adalah bilangan genap.

b) $x + 3 = 8$.

c) Ibukota Provinsi Jawa Barat adalah Semarang.

d) $12 \geq 19$.

e) Soekarno adalah Presiden Indonesia yang pertama.

f) Jam berapa kereta api Argo Bromo tiba di Gambir?

g) Kemarin hari hujan.

h) Kehidupan hanya ada di planet Bumi.

i) 1+2

j) Siapkan kertas ujian sekarang!

k) $x + y = y + x$ untuk setiap x dan y bilangan riil

Perhatikan

- “Hari ini hujan.” (Situasinya diberitahukan)
- “Beijing adalah ibu kota China.”
- “1 + 2 = 3”

<u>Berikut ini yang BUKAN proposisi:</u>

- “Siapa itu?” (pertanyaan)
- “La la la la la.” (kata-kata tak bermakna ☺)
- “Lakukan saja!” (perintah)
- “Ya, sepertinya begitu” (tidak jelas)
- “1 + 2” (expresi tanpa nilai benar/salah)

Logika Informatika

- Penting untuk bernalar matematis
- Logika: sistem yg didasarkan atas proposisi.
- Proposisi: pernyataan yang bernilai benar atau salah, tapi tidak kedua-duanya.
- Kita katakan bahwa nilai kebenaran dari suatu proposisi adalah benar (T) atau salah (F).
- Berkorespondensi dengan 1 dan 0 dalam dunia digital.

Contoh Proposisi

"Gajah lebih besar daripada kucing."

Ini suatu pernyataan ?	yes
Ini suatu proposisi ?	yes
Apa nilai kebenaran dari proposisi ini ?	true

Contoh Proposisi (2)

"1089 < 101"

Ini pernyataan ?	yes
Ini proposisi ?	yes
Apa nilai kebenaran dari proposisi ini ?	false

Contoh proposisi (3)

"y > 15"

Ini pernyataan ?	yes
Ini proposisi ?	no

Nilai kebenarannya bergantung pada nilai y, tapi nilai ini tidak spesifik.

Kita katakan tipe pernyataan ini adalah fungsi proposisi atau kalimat terbuka.

Contoh proposisi (4)

"Bulan ini Februari dan 24 < 5."

Ini pernyataan ?	yes
Ini proposisi ?	yes
Nilai kebenaran dari proposisi tersebut ?	false

Contoh proposisi (5)

"Jangan tidur di kelas!!!"

Ini pernyataan ? no

Ini permintaan.

Ini proposisi ? no

Hanya pernyataan yang dapat menjadi proposisi.

Contoh proposisi (6)

"Jika gajah berwarna hijau,
mereka dapat berlindung di bawah pohon cabe."

Ini pernyataan ?	yes
Ini proposisi ?	yes
Apa nilai kebenaran proposisi tersebut ?	True

Contoh proposisi (7)

"x < y jika dan hanya jika y > x."

Ini pernyataan ? yes

Ini proposisi ? yes

… sebab nilai kebenarannya tidak bergantung pada nilai x dan y.

Apa nilai kebenaran dari proposisi tsb ? true

Menggabungkan proposisi

Seperti dalam contoh sebelumnya, satu atau lebih proposisi dapat digabung membentuk sebuah proposisi majemuk (*compound proposition*).

Selanjutnya, notasi proposisi diformalkan dengan menggunakan alfabet seperti p, q, r, s, dan dengan memperkenalkan beberapa operator logika.

1. Gajah lebih besar daripada kucing
2. 1089 < 101”
3. y > 15
4. Bulan ini Februari dan 24 < 5.
5. Jangan tidur di kelas!.
6. Jika gajah berwarna merah, mereka dapat berlindung di bawah pohon cabe
7. x < y jika dan hanya jika y > x.

LOGIKA INFORMATIKA

Suraya

Jurusan Teknik Informatika

Konstanta dan Variabel Proposisi

- Variabel proposisi
  - ✓ Proposisi dapat dituliskan dengan simbol-simbol seperti A,B,C, …, yang hanya memiliki nilai benar (True) atau salah (False)
  - ✓ Contoh :

    A = harga gula naik

    B = pabrik gula senang

    C = petani tebu senang

    1) Jika A maka B
    2) Jika B maka C
    3) Jika A maka C
- Konstanta proposisi : T atau F
- Variabel dan konstanta proposisi adalah proposisi atomik.

Konstanta dan Variabel Proposisi

- Variabel dan konstanta proposisi adalah proposisi atomik.
- Proposisi Atomik
  - ✓ Proposisi yang berisi satu variabel proposisi atau satu konstanta proposisi
  - ✓ Contoh :

    Andi kaya raya (A)

    Antin hidup bahagia (B)
- Proposisi Majemuk
  - ✓ Semua proposisi bukan atomik yang memiliki minimal satu perangkai logika
  - ✓ Contoh :

    Andi kaya raya dan Antin hidup bahagia (A dan B)

Operator / Logical Connectives

- Sebuah *operator* atau *penghubung* menggabungkan satu atau lebih ekspresi *operand* ke dalam ekspresi yang lebih besar. (seperti tanda “+” di ekspresi numerik.)

- Operator *Uner* bekerja pada satu operand (contoh −3); Operator *biner* bekerja pada 2 operand (contoh 3×4).

- Operator *Proposisi* atau *Boolean* bekerja pada proposisi-proposisi atau nilai kebenaran, bukan pada suatu angka

Operator / Boolean Umum

Nama Resmi	Istilah	Arity	Simbol
Operator Negasi	NOT	Unary	$\neg$
Operator Konjungsi	AND	Binary	$\wedge$
Operator Disjungsi	OR	Binary	$\vee$
Operator Exclusive-OR	XOR	Binary	$\oplus$
Operator Implikasi	IMPLIES (jika-maka)	Binary	$\rightarrow$
Operator Biimplikasi (*Biconditional)*	IFF (jika dan hanya jika)	Binary	$\leftrightarrow$

Operator Negasi

- Operator negasi uner "¬" (*NOT*) mengubah suatu proposisi menjadi proposisi lain yang bertolak belakang nilai kebenarannya
- C*ontoh:* Jika *p* = Hari ini hujan
- maka ¬*p* = Tidak benar hari ini hujan
- Tabel kebenaran untuk NOT:

p	¬p
T	F
F	T

T = True; F = False

≡ Diartikan "didefinisikan sebagai"

Operator Konjungsi

- Operator konjungsi biner "$\wedge$" (*AND*) menggabungkan dua proposisi untuk membentuk logika konjungsinya
- *Cth:* p = Galih naik sepeda
  q = Ratna naik sepeda
- $p \wedge q$ = Galih dan Ratna naik sepeda

AND

Tabel Kebenaran Konjungsi

- Perhatikan bahwa

  Konjungsi $p1 \wedge p2 \wedge \ldots \wedge pn$ dari n proposisi akan

  memiliki 2^n baris pada tabelnya

p	q	$p \wedge q$
F	F	F
F	T	F
T	F	F
T	T	T

- Operasi ¬ dan $\wedge$ saja cukup untuk mengekspresikan *semua* tabel kebenaran Boolean!

Operator Disjungsi

Operator biner disjungsi "$\vee$" (*OR*) menggabungkan dua proposisi untuk membentuk logika disjungsinya

p="Mesin mobil saya rusak"

q="Karburator mobil saya rusak"

$p \vee q$="Mesin atau karburator mobil saya rusak."

Tabel Kebenaran Disjungsi

- Perhatikan bahwa $p \vee q$ berarti p benar, atau q benar, **atau keduanya** benar!
- Jadi, operasi ini juga disebut *inclusive or,* karena mencakup kemungkinan bahwa both p dan q keduanya benar.

p	q	$p \vee q$
F	F	F
F	T	**T**
T	F	**T**
T	T	T

Lihat bedanya dengan AND

- "¬" dan "$\vee$" keduanya membentuk opearator universal.

Proposi Bertingkat

- Gunakan tanda kurung untuk mengelompokkan *sub-ekspresi*:
  "<u>Saya baru saja bertemu teman lama</u>, <u>dan anaknya sudah dua atau tiga</u>." = $f \wedge (g \vee s)$
  - $(f \wedge g) \vee s$ artinya akan berbeda
  - $f \wedge g \vee s$ artinya akan ambigu
- Menurut perjanjian, "¬" presedensinya lebih tinggi dari "∧" dan "∨".
  - $\neg s \wedge f$ artinya $(\neg s) \wedge f$, **bukan** $\neg (s \wedge f)$

Latihan

Misalkan p="Tadi malam hujan",
q="Tukang siram tanaman datang tadi malam,"
r="Pagi ini kebunnya basah."

Terjemahkan proposisi berikut dalam bahasa Indonesia:

$\neg p$ = "Tadi malam tidak hujan."

$r \wedge \neg p$ = "Pagi ini kebunnya basah dan tadi malam tidak hujan."

$\neg r \vee p \vee q$ = "Pagi ini kebun tidak basah, atau tadi malam hujan, atau tukang siram tanaman datang tadi malam."

Operator Exclusive OR

Operator biner *exclusive-or* "$\oplus$" (*XOR*) menggabungkan dua proposisi untuk membentuk logika "exclusive or"-nya

p = "Saya akan mendapat nilai A di kuliah ini,"
q = "Saya akan *drop* kuliah ini,"
$p \oplus q$ = "Saya akan mendapat nilai A atau saya akan drop kuliah ini (tapi tidak dua-duanya!)"

Tabel Kebenaran Exclusive OR

- Perhatikan bahwa $p \oplus q$ berarti p benar, atau q benar tapi **tidak dua-duanya benar**!
- Disebut *exclusive or,* karena tidak memungkinkan p dan q keduanya benar
- "¬" dan "⊕" tidak membentuk operator universal

p	q	$p \oplus q$
F	F	F
F	T	T
T	F	T
T	T	**F**

Bahasa Alami sering Ambigu

- Perhatikan bahwa kata “atau” dapat bermakna *ambigu* berkenaan dengan kasus keduanya benar.
- “Tia adalah penulis atau Tia adalah aktris.” -
- “Tia perempuan atau Tia laki-laki” –
- Perlu diketahui konteks pembicaraannya!

p	q	p "or" q
F	F	F
F	T	T
T	F	T
T	T	?

Operator Implikasi

- Implikasi $p \rightarrow q$ menyatakan bahwa *p* mengimplikasikan *q.*
- *p* disebut *antecedent* dan *q* disebut *consequent*
- Jika *p* benar, maka *q* benar; tapi jika *p* tidak benar, maka *q* bisa benar - bisa tidak benar
- Contoh :

  p = Nilai ujian akhir anda 80 atau lebih

  q = Anda mendapat nilai A

  $p \rightarrow q$ = "Jika nilai ujian akhir anda 80 atau lebih, maka anda mendapat nilai A"

Implikasi $p \rightarrow q$

(a) Jika *p*, maka *q*	(*if p*, *then q*)
(b) Jika *p*, *q*	(*if p*, *q*)
(c) *p* mengakibatkan *q*	(*p implies q*)
(d) *q* jika *p*	(*q if p*)
(e) *p* hanya jika *q*	(*p only if q*)
(f) *p* syarat cukup agar *q*	(*p is sufficient for q*)
(g) *q* syarat perlu bagi *p*	(*q is necessary for p*)
(i) *q* bilamana *p*	(*q whenever p*)

Tabel Kebenaran Implikasi

- $p \rightarrow q$ salah <u>hanya jika</u> p benar tapi q tidak benar
- $p \rightarrow q$ tidak mengatakan bahwa hanya p yang menyebabkan q!
- $p \rightarrow q$ tidak mensyaratkan bahwa p atau q **<u>harus benar</u>**!
- *Cth.* "(1=0) $\rightarrow$ kucing bisa terbang" BENAR!

p	q	$p \rightarrow q$
F	F	T
F	T	T
T	F	**F**
T	T	T

Satu-satunya kasus SALAH!

Contoh Implikasi

- "Jika saya rajin kuliah hari ini, matahari akan bersinar esok hari" *True* / *False*?
- "Jika hari ini Selasa, maka saya adalah seekor pinguin." *True* / *False*?
- "Jika 1+1=6, Maka SBY adalah presiden."
  True / *False*?
- "Jika bulan dibuat dari keju, maka saya lebih kaya dari Bill Gates." *True* or *False*?

Converse, Inverse & Contrapositive

Beberapa terminologi dalam implikasi $p \rightarrow q$:

- *Converse*-nya adalah: $q \rightarrow p$.
- *Inverse*-nya adalah: $\neg p \rightarrow \neg q$.
- *Contrapositive*-nya adalah: $\neg q \rightarrow \neg p$.
- Salah satu dari ketiga terminologi di atas memiliki makna yang sama (memiliki tabel kebenaran yang sama) dengan $p \rightarrow q$. Bisa Anda sebutkan yang mana?

Bagaimana Menunjukkannya?

Membuktikan eqivalensi antara $p \rightarrow q$ dan contrapositive-nya dengan tabel kebenaran:

p	q	$\neg q$	$\neg p$	$p \rightarrow q$	$\neg q \rightarrow \neg p$
F	F	T	T	T	T
F	T	F	T	T	T
T	F	T	F	F	F
T	T	F	F	T	T

Operator Biimplikasi

- Operator biimplikasi $p \leftrightarrow q$ menyatakan bahwa *p* benar *jika dan hanya jika (ji**kk**a) q* benar
- *p* = “SBY menang pada pemilu 2004”
- *q* = “SBY akan menjadi presiden mulai tahun 2004.”
- $p \leftrightarrow q$ = “Jika dan hanya jika SBY menang pada pemilu 2004 maka dia akan menjadi presiden mulai tahun 2004.”

Biimplikasi p ↔ q

(a) *p* jika dan hanya jika *q*.

(*p if and only if q*)

(b) *p* adalah syarat perlu dan cukup untuk *q*.

(*p is necessary and sufficient for q*)

(c) Jika *p* maka *q*, dan sebaliknya.

(*if p then q, and conversely*)

(d) *p* jikka *q*

(p iff q)

Tabel Kebenaran Biimplikasi

- $p \leftrightarrow q$ benar jika p dan q memiliki nilai kebenaran yang sama.
- Perhatikan bahwa tabelnya adalah *kebalikan* dari tabel exclusive or $\oplus$!
  - $p \leftrightarrow q$ *artinya* $\neg(p \oplus q)$

p	q	$p \leftrightarrow q$
F	F	T
F	T	F
T	F	F
T	T	T

Perhatikan

Nyatakan pernyataan berikut dalam ekspresi logika :

"Anda tidak dapat terdaftar sebagai pemilih dalam Pemilu jika anda berusia di bawah 17 tahun kecuali kalau anda sudah menikah"

Misalkan :

p **: Anda berusia di bawah 17 tahun.**

q **: Anda sudah menikah.**

r **: Anda dapat terdaftar sebagai pemilih dalam Pemilu.**

maka pernyataan di atas dapat ditulis sebagai

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

Ringkasan

p	q	$\neg p$	$p \wedge q$	$p \vee q$	$p \oplus q$	$p \rightarrow q$	$p \leftrightarrow q$
F	F	T	F	F	F	T	T
F	T	T	F	T	T	T	F
T	F	F	F	T	T	F	F
T	T	F	T	T	F	T	T

TIFS 1604 – LOGIKA INFORMATIKA
Semester II

Suraya

Operator Logika

- Negasi (NOT)
- Konjungsi - Conjunction (AND)
- Disjungsi - Disjunction (OR)
- Eksklusif Or (XOR)
- Implikasi (JIKA – MAKA)
- Bikondisional (JIKA DAN HANYA JIKA)

Tabel kebenaran dapat digunakan untuk menunjukkan bagaimana operator-operator tsb menggabungkan proposisi-proposisi.

Negasi (NOT)

Operator Uner, Simbol: ¬

P	¬P
true	false
false	true

Conjunction (AND)

Operator Biner, Simbol: $\wedge$

p	q	$p \wedge q$
true	true	true
true	false	false
false	true	false
false	false	false

Disjunction (OR)

Operator Biner, Simbol: $\vee$

P	Q	$P \vee Q$
true	true	true
true	false	true
false	true	true
false	false	false

Exclusive Or (XOR)

Operator Biner, Simbol: $\oplus$

P	Q	$P \oplus Q$
true	true	false
true	false	true
false	true	true
false	false	false

Implikasi (JIKA - MAKA)

Implikasi $p \rightarrow q$ adalah proposisi yang bernilai salah jika p benar dan q salah, dan bernilai benar jika lainnya.

P	Q	$P \rightarrow Q$
true	true	true
true	false	false
false	true	true
false	false	true

Implikasi $p \rightarrow q$

- Jika p, maka q
- Jika p, q
- p mengakibatkan q
- p hanya jika q
- p cukup untuk q
- Syarat perlu untuk p adalah q

- q jika p
- q ketika p
- q diakibatkan p
- q setiap kali p
- q perlu untuk p
- Syarat cukup untuk q adalah p

Contoh Implikasi

Implikasi

"Jika hari ini hari Jumat maka 2+3 > 7."

bernilai benar untuk semua hari kecuali hari Jumat, walaupun 2+3 > 7 bernilai salah.

Kapan pernyataan berikut bernilai benar?

"Jika hari tidak hujan maka saya akan pergi ke Lembang."

Bikondisional
(JIKA DAN HANYA JIKA)

Operator Biner, Simbol: $\leftrightarrow$

P	Q	$P \leftrightarrow Q$
true	true	true
true	false	false
false	true	false
false	false	true

Pernyataan dan Operasi

Pernyataan-pernyataan dapat digabungkan dengan operasi untuk membentuk pernyataan baru.

P	**Q**	**$P \wedge Q$**	**$\neg (P \wedge Q)$**	**$(\neg P) \vee (\neg Q)$**
true	true	true		
true	false	false		
false	true	false		
false	false	false		

Pernyataan yang Ekivalen

P	Q	$\neg(P \wedge Q)$	$(\neg P) \vee (\neg Q)$	$\neg(P \wedge Q) \equiv (\neg P) \vee (\neg Q)$
true	true			
true	false			
false	true			
false	false			

Pernyataan $\neg(P \wedge Q)$ dan $(\neg P) \vee (\neg Q)$ ekivalen secara logika, karena $\neg(P \wedge Q)$**, dan** $(\neg P) \vee (\neg Q)$ punya nilai krbenaran yang sama.

Tautologi dan Kontradiksi

Tautologi adalah pernyataan yang selalu benar.

Contoh:

- $R \vee (\neg R)$
- $\neg(P \wedge Q) \leftrightarrow (\neg P) \vee (\neg Q)$

Jika $S \rightarrow T$ suatu tautologi, kita tulis $S \Rightarrow T$.

Jika $S \leftrightarrow T$ suatu tautologi, kita tulis $S \Leftrightarrow T$.

Tautologi dan Kontradiksi (2)

Kontradiksi adalah pernyataan yang selalu bernilai salah.

Contoh:

1. $\mathbf{R \wedge (\neg R)}$
2. $\mathbf{\neg(\neg(P \wedge Q) \leftrightarrow (\neg P) \vee (\neg Q))}$

Negasi dari suatu tautologi adalah suatu kontradiksi, negasi dari kontradiksi adalah suatu tautologi.

Konversi, Kontrapositif, & Invers

- $q \rightarrow p$ disebut konversi dari $p \rightarrow q$
- $\neg q \rightarrow \neg p$ disebut kontrapositif dari $p \rightarrow q$
- $\neg p \rightarrow \neg q$ disebut invers dari $p \rightarrow q$

Beberapa terminologi dalam implikasi $p \rightarrow q$:

- *Converse*-nya adalah: $q \rightarrow p$.
- *Inverse*-nya adalah: $\neg p \rightarrow \neg q$.
- *Contrapositive*-nya adalah: $\neg q \rightarrow \neg p$.

Ekspresi Logika

Contoh 4. Ubah ke dalam ekspresi logika:

"Anda mempunyai akses internet hanya jika anda mahasiswa PT IST-AKPRIND atau anda bukan mahasiswa UGM"

Solusi. Misal a : "Anda punya akses internet"

m: "Anda mhs PT IST-AKPRIND"

f : "Anda mhs UGM"

$$(m \vee \neg f) \rightarrow a$$

Ekspresi Logika (2)

Tugs I.

1. Ubah kedalam ekspresi logika kalimat di bawah ini dan gunakan tabel kebenaran untuk melihat validitasnya !!!.

   a. “Anda tidak boleh naik roller coaster jika tinggi anda kurang dari 100 cm, kecuali usia anda sudah melebihi 16 th.”

   b. “Saya akan ingat tentang kuliah besok hanya jika kamu mengirim sms.”

   c. “Pantai akan erosi ketika ada badai”

Puzzle Logika

2. Puzzle (Smullyan, ‘98)

   Suatu pulau mempunyai dua macam penghuni, yaitu penjujur (orang yg selalu berkata benar) dan pembohong (orang yg selalu berkata salah/bohong).

   Anda bertemu dua orang A dan B di pulau itu. Jika A berkata bhw “B penjujur” dan B berkata bhw “kami berdua mempunyai tipe yg berlainan”, maka apa yang dapat anda simpulkan tentang A dan B.

LOGIKA INFORMATIKA

Suraya

Jurusan Teknik Informatika

Materi Perkuliahan

- Arti Kalimat dan Interpretasi
- Logical Connectives
- Aturan Semantik
- Tabel Kebenaran

Arti Kalimat

- Arti kalimat = nilai kebenaran
- Setiap kalimat pada logika proposisi memiliki salah satu dari nilai {true, false}
- Arti kalimat kompleks yang terdiri atas n variabel merupakan fungsi dari nilai kebenaran n variabel tersebut
- Perlu tahu nilai kebenaran masing-masing variabel
- Perlu aturan untuk menghitung fungsi tersebut

Arti Kalimat

- Logika hanya berhubungan dengan bentuk (form) logis dari argumen-argumen, serta penarikan kesimpulan tentang validitas dari argumen tersebut
- Contoh 1:
  - Badu seorang manusia
  - Setiap manusia memiliki 2 mata
  - Maka Badu memiliki 2 mata
- Contoh 2:
  - Hewan meiliki 2 mata
  - Manusia memiliki 2 mata
  - Maka hewan sama dengan manusia

Interpretasi

- Interpretasi pada logika proposisi = pemberian nilai kebenaran pada semua variabel
- Contoh : $p \vee \neg q$
- 1 : p true dan q true
- 2 : p true dan q false
- 3 : p false dan q false
- 4 : p false dan q true

Aturan Semantik

- kalimat *true* bernilai true untuk semua interpretasi
- kalimat *false* bernilai false untuk semua interpretasi
- kalimat p,q,r,… bernilai sesuai interpretasinya
- *not F* bernilai true jika *F* false dan bernilai false jika *F* true
- $F \wedge G$ bernilai true jika *F* dan *G* keduanya true dan bernilai false jika tidak demikian
- $F \vee G$ bernilai false jika *F* dan *G* keduanya false dan bernilai true jika tidak demikian
- $F \rightarrow G$ bernilai false jika *F* true dan *G* false dan bernilai true jika tidak demikian

Tabel Kebenaran

- Dengan aturan semantik dapat ditentukan nilai kebenaran suatu kalimat kompleks untuk semua interpretasi yang mungkin
- Biasanya ditabelkan dan disebut tabel kebenaran
- Jika terdapat n variabel, maka terdapat 2^n baris tabel kebenaran

Operator / Logical Connectives

- Sebuah *operator* atau *penghubung* menggabungkan satu atau lebih ekspresi *operand* ke dalam ekspresi yang lebih besar. (seperti tanda “+” di ekspresi numerik.)

- Operator *Uner* bekerja pada satu operand (contoh -3); Operator *biner* bekerja pada 2 operand (contoh 3×4).

- Operator *Proposisi* atau *Boolean* bekerja pada proposisi-proposisi atau nilai kebenaran, bukan pada suatu angka

Operator / Boolean Umum

Nama Resmi	**Istilah**	**Arity**	**Simbol**
Operator Negasi	NOT	Unary	$\neg$
Operator Konjungsi	AND	Binary	$\wedge$
Operator Disjungsi	OR	Binary	$\vee$
Operator Exclusive-OR	XOR	Binary	$\oplus$
Operator Implikasi	IMPLIES (jika-maka)	Binary	$\rightarrow$
Operator Biimplikasi (*Biconditional)*	IFF (jika dan hanya jika)	Binary	$\leftrightarrow$

Operator Negasi

- Operator negasi uner "¬" (*NOT*) mengubah suatu proposisi menjadi proposisi lain yang bertolak belakang nilai kebenarannya
- C*ontoh:* Jika *p* = Hari ini hujan
- maka ¬*p* = Tidak benar hari ini hujan
- Tabel kebenaran untuk NOT:

p	¬p
T	F
F	T

T = True; F = False

≡ Diartikan "didefinisikan sebagai"

Operator Konjungsi

- Operator konjungsi biner "$\wedge$" (*AND*) menggabungkan dua proposisi untuk membentuk logika konjungsinya
- *Cth:* p = Badu menabrak pagar rumah
  q = Badu menginjak-injak pagar rumah
- $p \wedge q$ = Badu menabrak pagar rumah dan menginjak-injaknya

Tabel Kebenaran Konjungsi

- Perhatikan bahwa Konjungsi ***p*1** $\wedge$ ***p*2** $\wedge$ **…** $\wedge$ ***pn*** dari *n* proposisi akan memiliki 2^n baris pada tabelnya

p	q	$p \wedge q$
F	F	F
F	T	F
T	F	F
T	T	T

- Operasi ¬ dan $\wedge$ saja cukup untuk mengekspresikan *semua* tabel kebenaran Boolean!

Operator Disjungsi

Operator biner disjungsi "$\vee$" (*OR*) menggabungkan dua proposisi untuk membentuk logika disjungsinya

p="Saya memilih pizza untuk dinner"

q="Saya memilih fried chicken untuk dinner"

$p \vee q$="Saya memilih pizza atau fried chicken untuk dinner."

Tabel Kebenaran Disjungsi

- Perhatikan bahwa $p \vee q$ berarti p benar, atau q benar, **atau keduanya** benar!
- Jadi, operasi ini juga disebut *inclusive or,* karena mencakup kemungkinan bahwa both p dan q keduanya benar.

p	q	$p \vee q$
F	F	F
F	T	**T**
T	F	**T**
T	T	T

- "¬" dan "∨" keduanya membentuk opearator universal.

Proposi Bertingkat

- Gunakan tanda kurung untuk mengelompokkan *sub-ekspresi*:
  "Saya baru saja bertemu teman lama, dan anaknya sudah dua atau tiga." = $f \wedge (g \vee s)$
  - $(\boldsymbol{f} \wedge \boldsymbol{g}) \vee \boldsymbol{s}$ artinya akan berbeda
  - $\boldsymbol{f} \wedge \boldsymbol{g} \vee \boldsymbol{s}$ artinya akan ambigu
- Menurut perjanjian, "¬" presedensinya lebih tinggi dari "∧" dan "∨".
  - $\neg \boldsymbol{s} \wedge \boldsymbol{f}$ artinya $(\neg \boldsymbol{s}) \wedge \boldsymbol{f}$, **bukan** $\neg (\boldsymbol{s} \wedge \boldsymbol{f})$

Latihan

Misalkan p="Tadi malam hujan",
q="Tukang siram tanaman datang tadi malam,"
r="Pagi ini kebunnya basah."

Terjemahkan proposisi berikut dalam bahasa Indonesia:

$\neg p$ = "Tadi malam tidak hujan."

$r \wedge \neg p$ = "Pagi ini kebunnya basah dan tadi malam tidak hujan."

$\neg r \vee p \vee q$ = "Pagi ini kebun tidak basah, atau tadi malam hujan, atau tukang siram tanaman datang tadi malam."

Operator Exclusive OR

Operator biner *exclusive-or* "$\oplus$" (*XOR*) menggabungkan dua proposisi untuk membentuk logika "exclusive or"-nya

p = "Saya akan mendapat nilai A di kuliah ini,"
q = "Saya akan *drop* kuliah ini,"
$p \oplus q$ = "Saya akan mendapat nilai A atau saya akan drop kuliah ini (tapi tidak dua-duanya!)"

Tabel Kebenaran Exclusive OR

- Perhatikan bahwa $p \oplus q$ berarti p benar, atau q benar tapi **tidak dua-duanya benar**!
- Disebut *exclusive or,* karena tidak memungkinkan p dan q keduanya benar

p	q	$p \oplus q$
F	F	F
F	T	T
T	F	T
T	T	**F**

- "¬" dan "⊕" tidak membentuk operator universal

Bahasa Alami sering Ambigu

- Perhatikan bahwa kata “atau” dapat bermakna *ambigu* berkenaan dengan kasus keduanya benar.
- “Tia adalah penulis atau Tia adalah aktris.” -
- “Tia perempuan atau Tia laki-laki” –
- Perlu diketahui konteks pembicaraannya!

p	q	p "or" q
F	F	F
F	T	T
T	F	T
T	T	?

Operator Implikasi

- Implikasi $p \rightarrow q$ menyatakan bahwa *p* mengimplikasikan *q.*
- *p* disebut *antecedent* dan *q* disebut *consequent*
- Jika *p* benar, maka *q* benar; tapi jika *p* tidak benar, maka *q* bisa benar - bisa tidak benar
- Contoh :

  p = Nilai ujian akhir anda 80 atau lebih

  q = Anda mendapat nilai A

  $p \rightarrow q$ = "Jika nilai ujian akhir anda 80 atau lebih, maka anda mendapat nilai A"

Implikasi $p \rightarrow q$

(a) Jika *p*, maka *q*	(*if p*, *then q*)
(b) Jika *p*, *q*	(*if p*, *q*)
(c) *p* mengakibatkan *q*	(*p implies q*)
(d) *q* jika *p*	(*q if p*)
(e) *p* hanya jika *q*	(*p only if q*)
(f) *p* syarat cukup agar *q*	(*p is sufficient for q*)
(g) *q* syarat perlu bagi *p*	(*q is necessary for p*)
(i) *q* bilamana *p*	(*q whenever p*)

Tabel Kebenaran Implikasi

- $p \rightarrow q$ salah <u>hanya jika</u> p benar tapi q tidak benar
- $p \rightarrow q$ tidak mengatakan bahwa hanya p yang menyebabkan q!
- $p \rightarrow q$ tidak mensyaratkan bahwa p atau q **<u>harus benar</u>**!
- *Cth.* "(1=0) $\rightarrow$ kucing bisa terbang" BENAR!

p	q	$p \rightarrow q$
F	F	T
F	T	T
T	F	**F**
T	T	T

Satu-satunya kasus SALAH !

Contoh Implikasi

- "Jika saya rajin kuliah hari ini, matahari akan bersinar esok hari" *True* / *False*?
- "Jika hari ini Kamis, maka saya adalah seekor pinguin." *True* / *False*?
- "Jika 1+1=6, maka SBY adalah presiden."
  True / *False*?
- "Jika bulan dibuat dari keju, maka saya lebih kaya dari Bill Gates." *True* or *False*?

Converse, Inverse & Contrapositive

Beberapa terminologi dalam implikasi $p \rightarrow q$:

- *Converse*-nya adalah: $q \rightarrow p$.
- *Inverse*-nya adalah: $\neg p \rightarrow \neg q$.
- *Contrapositive*-nya adalah: $\neg q \rightarrow \neg p$.
- Salah satu dari ketiga terminologi di atas memiliki makna yang sama (memiliki tabel kebenaran yang sama) dengan $p \rightarrow q$. Bisa Anda sebutkan yang mana?

Bagaimana Menunjukkannya?

Membuktikan eqivalensi antara $p \rightarrow q$ dan contrapositive-nya dengan tabel kebenaran:

p	q	$\neg q$	$\neg p$	$p \rightarrow q$	$\neg q \rightarrow \neg p$
F	F	T	T	T	T
F	T	F	T	T	T
T	F	T	F	F	F
T	T	F	F	T	T

Operator Biimplikasi

- Operator biimplikasi $p \leftrightarrow q$ menyatakan bahwa *p* benar *jika dan hanya jika (ji**kk**a) q* benar
- *p* = “SBY menang pada pemilu 2004”
- *q* = “SBY akan menjadi presiden mulai tahun 2004.”
- $p \leftrightarrow q$ = “Jika dan hanya jika SBY menang pada pemilu 2004 maka dia akan menjadi presiden mulai tahun 2004.”

Biimplikasi p ↔ q

(a) *p* jika dan hanya jika *q*.

(*p if and only if q*)

(b) *p* adalah syarat perlu dan cukup untuk *q*.

(*p is necessary and sufficient for q*)

(c) Jika *p* maka *q*, dan sebaliknya.

(*if p then q, and conversely*)

(d) *p* jikka *q*

(p iff q)

Tabel Kebenaran Biimplikasi

- $p \leftrightarrow q$ benar jika p dan q memiliki nilai kebenaran yang sama.
- Perhatikan bahwa tabelnya adalah *kebalikan* dari tabel exclusive or $\oplus$!
  - $p \leftrightarrow q$ *artinya* $\neg(p \oplus q)$

p	q	$p \leftrightarrow q$
F	F	T
F	T	F
T	F	F
T	T	T

Perhatikan

Nyatakan pernyataan berikut dalam ekspresi logika :

"Anda tidak dapat terdaftar sebagai pemilih dalam Pemilu jika anda berusia di bawah 17 tahun kecuali kalau anda sudah menikah"

Misalkan :

p **: Anda berusia di bawah 17 tahun.**

q **: Anda sudah menikah.**

r **: Anda dapat terdaftar sebagai pemilih dalam Pemilu.**

maka pernyataan di atas dapat ditulis sebagai

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

Ringkasan

p	q	$\neg p$	$p \wedge q$	$p \vee q$	$p \oplus q$	$p \rightarrow q$	$p \leftrightarrow q$
F	F	T	F	F	F	T	T
F	T	T	F	T	T	T	F
T	F	F	F	T	T	F	F
T	T	F	T	T	F	T	T

Latihan - 1

- Gunakan konstanta proposisional A untuk “Bowo kaya raya” dan B untuk “Bowo hidup bahagia”.Lalu ubahlah pernyataan-pernyataan berikut menjadi bentuk logika :

1) Bowo tidak kaya raya
2) Bowo kaya raya dan hidup bahagia
3) Bowo kaya raya atau tidak hidup bahagia
4) Jika Bowo kaya raya, maka ia hidup bahagia
5) Bowo hidup bahagia jika dan hanya jika ia kaya raya

Latihan - 2

- Berilah konstanta proposisional, dan ubahlah pernyataan-pernyataan berikut menjadi bentuk logika :

1) Jika Bowo berada di Malioboro, maka Dewi juga berada di Malioboro
2) Pintu rumah Dewi berwarna merah atau coklat
3) Berita itu tidak menyenangkan
4) Bowo akan datang, jika ia mempunyai kesempatan
5) Jika Dewi rajin kuliah, maka ia pasti pandai

Latihan - 3

- Jawablah dengan tabel kebenaran :

1) Apakah nilai kebenaran dari $(A \wedge A)$?
2) Apakah nilai kebenaran dari $(A \vee A)$?
3) Apakah nilai kebenaran dari $(A \wedge \neg A)$?
4) Apakah $(A \rightarrow B)$ ekivalen dengan $(B \rightarrow A)$
5) Apakah $(A \rightarrow B) \rightarrow C$ ekivalen dengan $A \rightarrow (B \rightarrow C)$

Latihan - 4

- Buat tabel kebenaran untuk pernyataan berikut:

1) $\neg(\neg A \wedge \neg A)$
2) $A \wedge (A \vee B)$
3) $((\neg A \wedge (\neg B \wedge C)) \vee (B \wedge C)) \vee (A \wedge C)$
4) $(A \wedge B) \vee (((\neg A \wedge B) \rightarrow A) \wedge \neg B)$
5) $(A \rightarrow B) \leftrightarrow (\neg B \rightarrow \neg A)$

LOGIKA INFORMATIKA

Suraya

Jurusan Teknik Informatika

Materi Perkuliahan

- Ekivalensi Logis
- Pembuktian ekivalensi dengan Tabel Kebenaran
- Hukum-hukum Ekivalensi

Ekivalensi Proposisi

- Dua buah proposisi majemuk yang secara sintaksis (tertulis) berbeda dapat memiliki makna semantik yang sama. Kedua proposisi tersebut dikatakan “ekivalen”
- Kita akan pelajari:
  - Aturan dan hukum ekivalensi
  - Bagaimana membuktikan ekivalensi menggunakan *symbolic derivations*.

Ekivalensi Proposisi

- Contoh 1 :

1. Dewi sangat cantik dan peramah
2. Dewi peramah dan sangat cantik

   Ditulis $A \wedge B \equiv B \wedge A$

- Contoh 2 :

1. Badu tidak pandai atau dia tidak jujur
2. Adalah tidak benar jika Badu pandai dan jujur

   Ditulis $\neg A \vee \neg B \equiv \neg (A \wedge B)$

Ekivalensi Logika

- Proposisi majemuk *p* <u>ekivalen</u> dengan proposisi majemuk *q,* ditulis $p \Leftrightarrow q$, **IFF** proposisi majemuk $p \Leftrightarrow q$ apakah tautologi atau kontradiksi.
- Proposisi majemuk *p* dan *q* ekivalen satu sama lain **IFF** *p* dan *q* memiliki nilai kebenaran yang sama pada semua barisnya di tabel kebenaran

Membuktikan Ekivalensi dengan Tabel Kebenaran

Contoh. Buktikan $p \vee q \equiv \neg(\neg p \wedge \neg q)$.

p	q	$p \vee q$	$\neg p$	$\neg q$	$\neg p \wedge \neg q$	$\neg(\neg p \wedge \neg q)$
F	F	F	T	T	T	F
F	T	T	T	F	F	T
T	F	T	F	T	F	T
T	T	T	F	F	F	T

Hukum Ekivalensi

- *Identity*: $p \wedge \mathbf{T} \Leftrightarrow p$ $\quad p \vee \mathbf{F} \Leftrightarrow p$

  (*Identity of* $\wedge$ $\quad A \wedge 1 \Leftrightarrow A$ $\quad$ *Zero of* $\vee$ $\quad A \vee 0 \Leftrightarrow A$)
- *Domination*: $p \vee \mathbf{T} \Leftrightarrow \mathbf{T}$ $\quad p \wedge \mathbf{F} \Leftrightarrow \mathbf{F}$

  (*Identity of* $\vee$ $\quad A \vee 1 \Leftrightarrow 1$ $\quad$ *Zero of* $\wedge$ $\quad A \wedge 0 \Leftrightarrow 0$)
- *Idempotent*: $p \vee p \Leftrightarrow p$ $\quad p \wedge p \Leftrightarrow p$
- *Double negation:* $\neg\neg p \Leftrightarrow p$
- *Commutative:* $p \vee q \Leftrightarrow q \vee p$ $\quad p \wedge q \Leftrightarrow q \wedge p$
- *Associative:* $(p \vee q) \vee r \Leftrightarrow p \vee (q \vee r)$

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

Hukum Ekivalensi

- *Distributif*: $p \vee (q \wedge r) \Leftrightarrow (p \vee q) \wedge (p \vee r)$
  $p \wedge (q \vee r) \Leftrightarrow (p \wedge q) \vee (p \wedge r)$
- *De Morgan*:
  $\neg(p \wedge q) \Leftrightarrow \neg p \vee \neg q$
  $\neg(p \vee q) \Leftrightarrow \neg p \wedge \neg q$
- *Trivial tautology/contradiction*:
  $p \vee \neg p \Leftrightarrow \mathbf{T}$ $\quad$ $p \wedge \neg p \Leftrightarrow \mathbf{F}$
  $A \vee \neg A \Leftrightarrow 1$ $\quad$ $A \wedge \neg A \Leftrightarrow 0$

Hukum Ekivalensi

- *Absorption*: $p \vee (p \wedge q) \Leftrightarrow p$

  $p \wedge (p \vee q) \Leftrightarrow p$
- *Absorption*: $p \vee (\neg p \wedge q) \Leftrightarrow p \vee q$

  $p \wedge (\neg p \vee q) \Leftrightarrow p \wedge q$
- *Hukum lain*:

  $(p \wedge q) \vee (p \wedge \neg q) \Leftrightarrow p \qquad (p \vee q) \wedge (p \vee \neg q) \Leftrightarrow p$

  $(p \wedge q) \vee (\neg p \wedge q) \Leftrightarrow q \qquad (p \vee q) \wedge (\neg p \vee q) \Leftrightarrow q$

  $p \rightarrow q \Leftrightarrow \neg p \vee q$

  $p \rightarrow q \Leftrightarrow \neg (p \wedge \neg q)$

  $(p \leftrightarrow q) \Leftrightarrow (p \wedge q) \vee (\neg p \wedge \neg q)$

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

Definisi Operator dengan Ekivalensi

- Menggunakan ekivalensi, kita dapat mendefinisikan operator dengan operator lainnya
- Exclusive or: $p \oplus q \Leftrightarrow (p \vee q) \wedge \neg(p \wedge q)$
  $p \oplus q \Leftrightarrow (p \wedge \neg q) \vee (q \wedge \neg p)$
- Implikasi: $p \rightarrow q \Leftrightarrow \neg p \vee q$
- Biimplikasi: $p \leftrightarrow q \Leftrightarrow (p \rightarrow q) \wedge (q \rightarrow p)$
  $p \leftrightarrow q \Leftrightarrow \neg(p \oplus q)$

Contoh (1)

- Buktikan dengan *symbolic derivation* apakah
- $(p \wedge \neg q) \rightarrow (p \oplus r) \Leftrightarrow \neg p \vee q \vee \neg r$.

  $(p \wedge \neg q) \underline{\rightarrow} (p \oplus r) \Leftrightarrow$
- **[Expand definition of $\rightarrow$]** $\neg(p \wedge \neg q) \vee (p \underline{\oplus} r)$
- **[Defn. of $\oplus$]** $\Leftrightarrow \neg(p \wedge \neg q) \vee ((p \vee r) \wedge \neg(p \wedge r))$
- **[DeMorgan's Law]**
- $\Leftrightarrow (\neg p \vee q) \vee ((p \vee r) \wedge \neg(p \wedge r))$
- $\Leftrightarrow$ [associative law] *cont.*

Contoh (2)

- $(\neg p \vee q) \vee ((p \vee r) \wedge \neg(p \wedge r)) \Leftrightarrow$[$\vee$ commutes]
- $\Leftrightarrow \underline{(q \vee \neg p)} \vee ((p \vee r) \wedge \neg(p \wedge r))$[$\vee$ associative]
- $\Leftrightarrow q \vee \underline{(\neg p \vee}((p \vee r) \wedge \neg(p \wedge r)))$ [distrib.$\vee$over $\wedge$]
- $\Leftrightarrow q \vee (((\underline{\neg p \vee} (p \vee r)) \wedge (\underline{\neg p \vee} \neg(p \wedge r)))$
- [assoc.] $\Leftrightarrow q \vee (((\underline{(\neg p \vee p) \vee r}) \wedge (\neg p \vee \neg(p \wedge r)))$
- [trivial taut.] $\Leftrightarrow q \vee ((\underline{T} \vee r) \wedge (\neg p \vee \neg(p \wedge r)))$
- [domination] $\Leftrightarrow q \vee (\underline{T} \wedge (\neg p \vee \neg(p \wedge r)))$
- [identity] $\Leftrightarrow q \vee (\neg p \vee \neg(p \wedge r)) \Leftrightarrow$ *cont.*

Contoh (3)

- $q \vee (\neg p \vee \neg(p \wedge r))$
- [DeMorgan's] $\Leftrightarrow q \vee (\neg p \vee (\neg p \vee \neg r))$
- [Assoc.] $\Leftrightarrow q \vee ((\neg p \vee \neg p) \vee \neg r)$
- [Idempotent] $\Leftrightarrow q \vee (\neg p \vee \neg r)$
- [Assoc.] $\Leftrightarrow (q \vee \neg p) \vee \neg r$
- [Commut.] $\Leftrightarrow \neg p \vee q \vee \neg r$
- *Q.E.D. (quod erat demonstrandum)*

Contoh penyederhanaan ekspresi logika (tidak memungkinkan dimanipulasi lagi)

$(A \vee 0) \wedge (A \vee \neg A)$	
$\equiv A \wedge (A \vee \neg A)$	Zero of v (Identity Lows)
$\equiv A \wedge 1$	Tautologi
$\equiv A$	Identity of $\wedge$

Contoh penyederhanaan ekspresi logika (selasa)

$(A \wedge \neg B) v (A \wedge B \wedge C)$	
$\cong (A \wedge \neg B) v (A \wedge (B \wedge C))$	tambah kurung
$\cong A \wedge (\neg B v (B \wedge C))$	Distributif
$\cong A \wedge ((\neg B v B) \wedge (\neg B v C))$	Distributif
$\cong A \wedge (1 \wedge (\neg B v C))$	Tautologi
$\cong A \wedge (\neg B v C))$	Identity of $\wedge$

Sederhanakan Ekspresi Logika berikut (dengan Hukum Ekivalen):

1. $\neg A \rightarrow \neg(A \rightarrow \neg B)$
2. $A v (A \wedge B)$
3. $A \wedge (A v B)$
4. $((A \wedge (B \rightarrow C)) \wedge (A \rightarrow (B \rightarrow \neg C))) \rightarrow A$
5. $(A v B) \wedge \neg A \wedge \neg B$
6. $((A v B) \wedge \neg A) \rightarrow \neg B$
7. $(\neg A \rightarrow \neg B) \rightarrow ((A \rightarrow B) \rightarrow \neg A)$
8. Buktikan $(A \rightarrow B) \wedge (B \rightarrow A) \cong (A \wedge B) v (\neg A v B)$

LOGIKA INFORMATIKA

Suraya

Jurusan Teknik Informatika

Materi Perkuliahan

- Konsep Proposisi Majemuk
- Manfaat Skema
- Parsing
- Precedence Rules
- Tautologi, Kontradiksi dan Contingen

Ekspresi Logika (1)

- Ekspresi Logika adalah proposisi-proposisi yang dibangun oleh variabel-variabel logika yang berasal dari pernyataan atau argumen
- Contoh : A → B
- Setiap ekspresi logika dapat bersifat atomik atau majemuk tergantung dari variabel proposisional yang membentuknya bersama perangkai logika yang relevan

Ekspresi Logika (2)

- Contoh
  - Jika Dewi rajin belajar, maka ia akan lulus ujian dan ia dapat pergi nonton bioskop
- Diubah menjadi variabel proposisional :
  - A = Dewi rajin belajar
  - B = Dewi lulus ujian
  - C = Dewi pergi nonton bioskop
- Maka ekspresi logikanya :
  - **$A \rightarrow B \wedge C$**
  - **Urutan pengerjaan : $(A \rightarrow B) \wedge C$ atau $A \rightarrow (B \wedge C)$?**

    $\rightarrow$ ambigu

Skema (1)

- Skema merupakan cara untuk menyederhanakan suatu proposisi majemuk yang rumit, dengan memberi huruf tertentu untuk menggantikan satu sub ekspresi ataupun sub-sub ekspresi
- Suatu ekspresi logika tertentu, misal $(A \wedge B)$ dapat diganti dengan P, sedangkan $(A \vee B)$ dapat diganti dengan Q. Jadi P berisi variabel proposisional A dan B, demikian juga Q.
- Dalam hal ini, P maupun Q bukan variabel proposisional

Skema (2)

- Contoh :

$$P = (A \wedge B) \quad \text{dan} \quad Q = (A \vee B)$$
$$\therefore (P \rightarrow Q) = ((A \wedge B) \rightarrow (A \vee B))$$

- Perhatikan bahwa :
  - Ekspresi apa saja yang berbentuk ($\neg$P) disebut Negasi
  - Ekspresi apa saja yang berbentuk (P$\wedge$Q) disebut Konjungsi
  - Ekspresi apa saja yang berbentuk (P$\vee$Q) disebut Disjungsi
  - Ekspresi apa saja yang berbentuk (P$\rightarrow$Q) disebut Implikasi
  - Ekspresi apa saja yang berbentuk (P$\Leftrightarrow$Q) disebut Ekuivalensi

Skema (3)

- Well formed formulae (**Formula** adalah sekumpulan instruksi yang dimasukkan ke dalam sel untuk melakukan perhitungan (penambahan, pengurangan, perkalian, pembagian dan lain-lain)) (wff) :
  - Semua ekspresi atomik adalah fpe (*fully parenthisized expression*)
  - Jika P adalah fpe, demikian juga ($\neg$P)
  - Jika P dan Q adalah fpe, demikian juga (P$\wedge$Q), (P$\vee$Q), (P$\rightarrow$Q) dan (P$\leftarrow\rightarrow$Q)
  - Tak ada fpe lainnya

Menganalisis Proposisi Majemuk

- Contoh :

  [1] Jika Dewi lulus sarjana PTI, orang tuanya akan senang, dan dia dapat segera bekerja, tetapi jika dia tidak lulus, semua usahanya akan sia-sia

- Analisis

  [1.1] Jika Dewi lulus sarjana PTI, orang tuanya akan senang, dan dia dapat segera bekerja

  dengan

  [1.2] Jika dia tidak lulus, semua usahanya akan sia-sia

Menganalisis Proposisi Majemuk

- Sub proposisi skop kiri:

  [1.1.1] Jika Dewi lulus sarjana PTI

  dengan

  [1.1.2] Orang tuanya akan senang, dan Dewi dapat segera bekerja

- Sub sub proposisi skop kiri:

  [1.1.2.1] Orang tua Dewi akan senang

  dengan

  [1.1.2.2] Dewi dapat segera bekerja

Menganalisis Proposisi Majemuk

- Sub proposisi skop kanan:

  [1.2.1] Jika Dewi tidak lulus

  dengan

  [1.2.2] semua usaha Dewi akan sia-sia

- Teknik memilah-milah kalimat menjadi proposisi-proposisi yang atomik disebut *Parsing*.
- Hasilnya dapat diwujudkan dalam bentuk *Parse Tree*

Menganalisis Proposisi Majemuk

- Parse Tree diubah menjadi fpe sebagai berikut :
  - *A = Dewi lulus sarjana PTI*
  - *B = Orang tua Dewi senang*
  - *C = Dewi bekerja*
  - *D = Usaha Dewi sia-sia*
- Pernyataan tersebut ditulis :

$$\left(A \rightarrow \left(B \wedge C\right)\right) \wedge \left(\left(\neg A\right) \rightarrow D\right)$$

Menganalisis Proposisi Majemuk

- Contoh 1 :

1. Jika anda mengambil mata kuliah logika, dan anda tidak memahami tautology, maka anda tidak lulus mata kuliah tersebut

   – *A = anda mengambil mata kuliah logika*
   – *B = anda memahami tautology*
   – *C = anda lulus mata kuliah*

- Ekspresi logika :

  $(A \wedge \neg B) \rightarrow \neg C$

Menganalisis Proposisi Majemuk

- Contoh 2 :

1. Jika anda belajar rajin dan sehat, maka anda lulus ujian, atau jika anda tidak belajar rajin dan tidak sehat, maka anda tidak lulus ujian

- Variabel proposisinya :
  - A = anda belajar rajin
  - B = anda sehat
  - C = anda lulus ujian

- Ekspresi logika :

  $(((\mathbf{A} \wedge \mathbf{B}) \rightarrow \mathbf{C}) \vee ((\neg\mathbf{A} \wedge \neg\mathbf{B})$ **?**

Precedence Rules

untuk menjaga kebenaran sebuah pernyataan maka setiap operator/ penghubung diberikan aturan yang lebih tinggi

¬	∧	∨	⊕	→	↔

Contoh :

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

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

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

$p \leftrightarrow q \rightarrow r \equiv p \leftrightarrow (q \rightarrow r)$

Left Associate Rules

untuk operator/ penghubung yang setara digunakan left associate rule dimana operator sebelah kiri punya precedence lebih tinggi

Contoh :

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

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

Latihan

- Bagian 1
  - Ubahlah pernyataan-pernyataan berikut kedalam ekspresi logika :

  1. Jika tikus itu waspada dan bergerak cepat, maka kucing atau anjing itu tidak mampu menangkapnya
  2. Bowo membeli saham atau property untuk investasinya, atau dia dapat menanamkan uang di deposito bank dan mendapat bunga uang

- Bagian 2
  - Beri tanda kurung pada ekspresi berikut agar tidak ambigu

  1. $A \wedge B \wedge C \rightarrow D$
  2. $A \vee B \vee C \leftrightarrow \neg D$

Latihan

- Bagian 3
  - Jika nilai A dan B adalah T, sedangkan C dan D adalah F, carilah nilai kebenaran dari ekspresi logika berikut :

1. $A \wedge (B \vee C)$
2. $((A \vee B) \wedge C) \vee \neg((A \vee B) \wedge (B \vee D))$
3. $(\neg(A \wedge B) \vee \neg C) \vee (((\neg A \wedge B) \vee \neg D) \wedge C)$

Tautologi dan Kontradiksi

- *Tautology* adalah proposisi majemuk yang selalu bernilai **true** tidak peduli apa nilai kebenaran proposisi penyusunnya!
- Contoh: $p \vee \neg p$ [Apa tabel kebenarannya?]
- *Kontradiksi* adalah proposisi majemuk yang selalu bernilai **false** tidak peduli apapun!
- Contoh: $p \wedge \neg p$ [tabel kebenaran?]
- Proposisi majemuk selain itu disebut *contingencies*.

Tautologi

- Contoh 1:

  $A \vee \neg A$ *apakah tautology?*
- Buat tabel kebenarannya!
- Contoh 2 :

  $\neg(A \wedge B) \vee B$ *apakah Tautology*

Tautologi

- Contoh 3 :

  $(A \wedge B) \rightarrow (C \vee (\neg B \rightarrow \neg C))$
- Buat tabel kebenarannya!
- Contoh 4 :

  Jika $\neg(A \wedge B) \vee B$ adalah Tautology, buktikan $\neg(A \vee B) \wedge C) \vee C$ juga Tautology
  - Substitusi $\neg(A \wedge B) \vee B$ menjadi $\neg(P \wedge Q) \vee Q$
  - Misal P = $(A \vee B)$ dan Q = C
  - $\neg((A \vee B) \wedge C) \vee C$ akan menjadi $\neg(P \wedge Q) \vee Q$

Kontradiksi

- *Contoh 1 :*

  $A \wedge \neg A$ *apakah kontradiksi ?*
- *Contoh 2 :*

  $((A \vee B) \wedge \neg A) \wedge \neg B$
- Buat tabel kebenarannya!

Contingent

- *Contoh 1 :*

  $((A \wedge B) \rightarrow C) \rightarrow A$
- Buat tabel kebenarannya!
- Contoh 2 :

  $((A \rightarrow B) \wedge (\neg B \rightarrow C)) \rightarrow (\neg C \rightarrow A)$

Latihan

- Bagian 1
- Tentukan apakah ekspresi berikut ini termasuk tautology, kontradiksi atau contingrent

1. $A \rightarrow (B \rightarrow A)$
2. $\neg\neg A \rightarrow A$
3. $(\neg A \rightarrow \neg B) \rightarrow (B \rightarrow A)$

- Bagian 2
- Jika $A \vee \neg A$ adalah tautolgy, buktikan bahwa ekspresi berikut merupakan tautology

1. $(A \rightarrow B) \rightarrow \neg (A \rightarrow B)$
2. $\neg A \vee \neg\neg A$

- Contoh :

1. Jika anda mengambil mata kuliah logika, dan anda tidak memahami tautology, maka anda tidak lulus mata kuliah tersebut

2. Jika anda belajar rajin dan sehat, maka anda lulus ujian, atau jika anda tidak belajar rajin dan tidak sehat, maka anda tidak lulus ujian

Struktur Kendali

URUTAN KEPUTUSAN PENGULANGAN

Sequence (Urutan)

Pernyataan 1

Pernyataan 2

.

.

.

Pernyataan n

Sequence (Urutan)

Contoh:

```
If kondisi Then
Begin
    Pernyataan_11;
    Pernyataan_12;
    ….
    Pernyataan_1n;
End
Else
Begin
    Pernyataan_21;
    Pernyataan_22;
    ….
    Pernyataan_2n;
End
```

Decision (Keputusan)

Pada bentuk ini, pernyataan 1 hanya akan di jalankan kalau kondisi bernilai **True,** serta pernyataan 2 hanya akan di jalankan kalau kondisi bernilai **False**

Bagian kondisi berupa ekspresi yang telah kita bahas di depan (And, Or, dsb) → bisa coba misalkan ada n variabel menggunakan operator sembarang hasilnya seperti apa!

Decision (Keputusan)

```
If kondisi Then
    pernyataan _1
Else
    Pernyataan_2
```

Contoh:

```
Begin
        Write ('suhu tubuh :  ')
        ReadLn (suhu);
        If suhu > 37  Then
                WriteLn ('suhu tinggi !');
        Else
                WriteLn ('suhu tidak tinggi');
        WiteLn ('selesai');
End.
```

Repetition (Pengulangan)

Pernyataan **While** biasa digunakan untuk melakukan pengulangan yang jumlahnya tidak diketahui.

Pada bentuk ini, pengulangan terhadap pernyataan dilakukan terus selama kondisi bernilai **True,** bilai kondisi bernilai **False** maka pernyatan selesai untuk dieksekusi.

Repetition (Pengulangan)

Contoh:

```
Begin
   Pencacah := 1;
   While Pencacah <= 10 Do
   Begin
        WriteLn (pencacah);
        Pencacah := Pencacah + 1;
   End;
End.
```

While **kondisi** Do
    Pernyataan

LOGIKA INFORMATIKA

Suraya

Bahasan

- Penggunaan Logika dalam Pemrograman
- Representasi Algoritma
- Flowchart

Algoritma

- Algoritma adalah urutan langkah berhingga untuk memecahkan masalah logika atau matematika (Microsoft Book)
- Algoritma adalah urutan langkah-langkah logis penyelesaian masalah yang disusun secara sistematis

Contoh Algoritma

- Diberikan dua buah bejana A dan B. A berisi larutan berwarna Merah dan B berisi larutan berwarna Biru. Tukarkan isi kedua bejana sedemikian sehingga A berisi larutan berwarna Biru dan B berisi laruan berwarna Merah
- Langkah / algoritma :
  - Tuangkan larutan dari bejana A ke bejana C
  - Tuangkan larutan dari bejana B ke bejana A
  - Tuangkan larutan dari bejana C ke bejana B

Contoh Algoritma

- Memasak.
- Jika seseorang ingin mengirim surat kepada kenalannya di tempat lain, langkah yang harus dilakukan adalah:
  - Menulis surat
  - Surat dimasukkan ke dalam amplop tertutup
  - Amplop ditempeli perangko secukupnya.
  - Pergi ke Kantor Pos terdekat untuk mengirimkannya
- Dalam bidang komputer, algoritma sangat diperlukan dalam menyelesaikan berbagai masalah pemrograman, terutama dalam komputasi numeris.
- Tanpa algoritma yang dirancang baik, maka proses pemrograman akan menjadi salah, rusak, atau lambat dan tidak efisien

Struktur Kendali

Urutan | Keputusan | Pengulangan

Representasi Algoritma

- Dalam bahasa natural (Bahasa Indonesia, Bahasa Inggris, dan bahasa manusia lainnya)
  - Tapi sering membingungkan (*ambiguous*)
- Menggunakan *flow chart* (diagram alir)
  - Bagus secara visual akan tetapi repot kalau algoritmanya panjang
- Menggunakan pseudo-code
  - Sudah lebih dekat ke bahasa pemrograman, namun sulit dimengerti oleh orang yang tidak mengerti pemrograman

Algoritma Dalam Bahasa Natural

1. Ambil bilangan pertama dan set *maks* sama dengan bilangan pertama
2. Ambil bilangan kedua dan bandingkan dengan *maks*
3. Apa bila bilangan kedua lebih besar dari *maks*, set *maks* sama dengan bilangan kedua
4. Ambil bilangan ketiga dan bandingan dengan *maks*
5. Apabila bilangan ketiga lebih besar dari *maks*, set *maks* sama dengan bilangan ketiga
6. Variabel *maks* berisi bilangan terbesar. Tayangkan hasilnya

Algoritma dengan Flowchart

Algoritma dengan *pseudocode*

$maks \leftarrow$ bilangan pertama

if ($maks$ < bilangan kedua)

    $maks \leftarrow$ bilangan kedua

if ($maks$ < bilangan ketiga)

    $maks \leftarrow$ bilangan ketiga

Struktur Kendali

a. Sequence

Urutan

b. Decision

Keputusan

c. Repetition

Pengulangan

Figure 8-8

Pseudocode

```
action 1
action 2
   .
   .
   .
action n
```

a. Sequence

```
if (condition)
   then
      action
      action
      . . .
   else
      action
      action
      . . .
End if
```

b. Decision

```
while (condition)
      action
      action
      . . .
End while
```

c. Repetition

Urutan

Keputusan

Pengulangan

Keputusan

Contoh:

```
Begin
    Write ('suhu tubuh :  ')
    ReadLn (suhu);
    If suhu > 37  Then
        WriteLn ('suhu tinggi !');
    Else
        WriteLn ('suhu tidak tinggi');
    WiteLn ('selesai');
End.
```

```
if (condition)
  then
      action
      action
      . . .
  else
      action
      action
      . . .
End if
```

b. Decision

Pengulangan

While **kondisi**
Do
Pernyataan

Contoh:

```
Begin
   Pencacah := 1;
   While Pencacah <= 10 Do
   Begin
        WriteLn (pencacah);
        Pencacah := Pencacah + 1;
   End;
End.
```

```
while (condition)
     action
     action
     . . .
End
```

c. Repetition

Example 1

Example: Flowcharts

Example 2

Example: Flowcharts

LOGIKA INFORMATIKA

Suraya

Bahasan

- Operasi Penyederhanaan
- Falsifikasi
- Pohon Semantik

Penyederhanaan

- Penyederhanaan dilakukan menggunakan hukum-hukum logika
- Proses penyederhanaan akan berhenti pada bentuk ekspresi logika yang paling sederhana dan tidak mungkin disederhanakan lagi
- Perangkai → dan ←→ dapat diganti dengan perangkai dasar $\wedge$, $\vee$ dan $\neg$

Example #1

$$(\mathbf{A} \wedge \neg \mathbf{B}) \vee (\mathbf{A} \wedge \mathbf{B} \wedge \mathbf{C})$$

$$\equiv (A \wedge \neg B) \vee (A \wedge (B \wedge C)) \qquad \text{Asosiatif}$$

$$\equiv A \wedge (\neg B \vee (B \wedge C)) \qquad \text{Distributif}$$

$$\equiv A \wedge ((\neg B \vee B) \wedge (\neg B \vee C)) \qquad \text{Distributif}$$

$$\equiv A \wedge (1 \wedge (\neg B \vee C)) \qquad \text{Tautologi}$$

$$\equiv A \wedge (\neg B \vee C) \qquad \text{Identitas} \wedge$$

Example #2

$(\neg A \rightarrow \neg B) \rightarrow ((A \rightarrow B) \rightarrow \neg A)$	
$\equiv \neg(\neg\neg A \vee \neg B) \vee (\neg(\neg A \vee B) \vee \neg A)$	$\mathrm{A} \rightarrow \mathrm{B}$
$\equiv (\neg\neg\neg A \wedge \neg\neg B) \vee ((\neg\neg A \wedge \neg B) \vee \neg A)$	D'Morgan
$\equiv (\neg A \wedge B) \vee ((\mathrm{A} \wedge \neg \mathrm{B}) \vee \neg A)$	Komutatif
$\equiv (\neg A \wedge B) \vee (\neg A \vee \neg B)$	Absorpsi
$\equiv (\neg A \wedge B) \vee \neg A \vee \neg B$	*A*sosiatif
$\equiv \neg \mathrm{A} \vee (\neg A \wedge B) \vee \neg B$	Komutatif
$\equiv (\neg \mathrm{A} \vee (\neg A \wedge B)) \vee \neg B$	Asosiatif
$\equiv \neg \mathrm{A} \vee \neg \mathrm{B}$	Absorpsi

Soal

- Sederhanakan ekspresi logika berikut :

1. $A \wedge (\neg A \rightarrow A)$
2. $\neg(\neg A \wedge (B \vee \neg B))$
3. $\neg A \rightarrow \neg(A \rightarrow \neg B)$
4. $(A \rightarrow B) \rightarrow ((A \rightarrow \neg B) \rightarrow \neg A)$
5. $(A \rightarrow (B \vee \neg C)) \wedge \neg A \wedge B$

Falsifikasi (pengandaian bahwa kalimat salah)

if {(not p) or (not q)} then {not (p and q)}

dengan menggunakan aturan if-then maka antecedent /kejadian terdahulu (not p) or (not q) dan consequent {not(p and q)} masing-masing haruslah bernilai true dan false

Selanjutnya dari benarnya (not p) or (not q) kita tak dapat menyimpulkan tentang (not p) maupun (not q) sehingga kita beralih ke salahnya not(p and q) ; karena not (p and q)= false maka (p and q), dengan *aturan not*, bernilai true , seterusnya p and q berarti, dengan aturan and p dan q harus bernilai true, didapat :

Falsifikasi

(E : if {(not p) or (not q)} then {not(p and q)})
f f t <u>tf</u> f t f t t

Dari label terlihat bahwa p pada antecedent bernilai true, jadi (not p) bernilai false; begitu pula untuk (not q) akan bernilai false. Kesimpulan dari ini semua adalah antecedent, dengan <u>aturan or</u>, bernilai false. Tetapi didepan dikatakan bahwa antecedent bernilai true, sehingga terjadi kontradiksi ($\underline{t_f}$) yang berarti pengandaian bahwa kalimat salah adalah tidak benar, ini dapat disimpulkan bahwa kalimat E bernilai true yaitu kalimat <u>*valid*</u>.

Example

E : if {(not p) or (not q)} then {not(p and q)}
f

E : if {(not p) or (not q)} then {not(p and q)}
f t f

E : if {(not p) or (not q)} then {not(p and q)}
f t t t f t t

(E : if {(not p) or (not q)} then {not(p and q)})
f f t $\underline{t_f}$ f t f t t

Example

- **(E : if {(not p) or (not q)} then {not(p and q)})**
- **f f t $\underline{t_f}$ f t f t t**

- Jadi dari pengandaian ketidak-benarnya kalimat E, mengakibatkan terjadi $\underline{t_f}$, yaitu true sekaligus false yg berarti ada kontradiksi sehingga pengandaian diatas (bahwa kalimat E false) dicabut, yang berarti kalimat E true

Contoh-2

- **F : (if p then q) if and only if ((not p) or q)**
- **Andaikan F false maka akan dibuktikan terjadi kontra diksi dibawah suatu interpretasi.**

- **Menurut aturan if-and-only-if maka F dapat false untuk dua kemungkinan, yaitu : (a) ruas kiri true dan ruas kanan false, (b) ruas kiri false dan ruas kanan true.**
- **Kasus pertama yaitu if p then q adalah true dan ((not p) or q) adalah false, kita tulis sbb :**

  (if p then q) if and only if ((not p) or q)

  t f f

Contoh-2

- **F : (if p then q) if and only if ((not p) or q)**

- **3. Dari : (if p then q) if and only if ((not p) or q)**
  t f f

  Jika subkalimat (if p then q), ruas kiri, true maka kita tidak dapat menentukan nilai p dan q, sehingga kita lihat subkalimat ((not p) or q), ruas kanan, false; dengan demikian subkalimat dari subkalimat kanan, yaitu not p dan q harus dua-duanya false. Karena not p false mala p true.

- **Didapat : (if p then q) if and only if ((not p) or q)**
  ***tf* t f f f t f f**
- **Kesimpulan terjadi kontradiksi untuk kasus pertama maka haruslah kalimat F true.**

Contoh-2

Selanjutnya *kasus kedua* **yaitu : if p then q adalah false dan ((not p) or q) adalah true, kita tulis sbb :**

(if p then q) if and only if ((not p) or q)

f f t

Pada subkalimat , ruas kiri, (if p then q) false, maka jelaslah bahwa p bernilai true dan q bernilai false,

sehingga : (if p then q) if and only if ((not p) or q)

f t f f f t t_f **f**

Kesimpulan terjadi kontradiksi untuk kasus kedua maka kalimat F true.

Dari dua kasus tersebut maka disimpulkan F true

Contoh-3

1. **Apakah kalimat dibawah ini valid atau tak valid :**

   G : if {if p then q} then {if (not p) then (not q)}

Andaikan false maka : antecedentnya t dan konsekuen nya false jadi :

1. **if {if p then q} then {if (not p) then (not q)}**

   f t f

2. **if {if p then q} then {if (not p) then (not q)}**

   f t f t f t f f t

3. **Kesimpulan memang benar bahwa kalimat G false, pengandaian dibenarkan.**

Soal

1. Apakah kalimat dibawah ini valid atau tak valid :

   G : if {if(not p) then q} then {if (not q) then p } and (p or q)

2. **Apakah kalimat/formula dibawah ini tautologi :**

(a) $(p \wedge q) \rightarrow p$; **(b) $(p \wedge q) \rightarrow q$**

(c) $(p \wedge (p \rightarrow q)) \rightarrow q$; **(d) $\sim(\sim p) \leftrightarrow p$**

(e) $(p \leftrightarrow q) \leftrightarrow ((p \rightarrow q) \wedge (q \rightarrow p))$; **(f) $(p \vee (\sim p) \leftrightarrow (q \vee (\sim q))$**

3. Buktikan bahwa : $p \rightarrow (q \rightarrow r) \leftrightarrow (p \wedge q) \rightarrow r$; dengan tidak menggunakan tabel kebenaran

4. Seperti no. 3 untuk : $(\sim p \wedge (\sim q \wedge r)) \vee (q \wedge r) \vee (p \wedge r) \leftrightarrow r$

Soal

- **Tunjukan bahwa nilai kebenaran rumusan pernyataan berikut ini tak tergantung pada komponen-komponennya :**

  a. $(p \wedge (p \rightarrow q)$ **b. $(p \rightarrow q) \leftrightarrow (\sim p \vee q)$**

  c. $((p \rightarrow q) \wedge (q \rightarrow r)) \rightarrow (p \rightarrow r)$
- **Buktikan ekuivalensi berikut ini tanpa menggunakan tabel kebenaran .**

  a) $p \rightarrow (q \vee r) \cong (p \rightarrow q) \vee (p \rightarrow r)$;

  b) $(p \leftrightarrow q) \cong (p \wedge q) \vee (\sim p \vee \sim q)$

  c) $\sim(p \leftrightarrow q) \cong (p \wedge (\sim q)) \vee (\sim p \wedge q)$
- **Buktikan soal nomor 2 diatas dng tabel kebenaran.**
- **Tunjukan rumusan ini merupakan tautologi :**

  a) $(p \wedge q) \rightarrow (p \rightarrow q)$; **b) $p \rightarrow (q \rightarrow p)$;**

Pohon Semantik -1

1. Andaikan ingin membuktikan validitas kalimat :

G : if (If p then q) then (if (not p) then (not q))

p mempunyai dua kemungkinan nilai yaitu true dan false :

dari kalimat G : if (if p then q) then (if (not p) then (not q))

t **t**

Pohon Semantik -2

kalimat G : if (if p then q) then (if (not p) then (not q))
t f t

subkalimat G : (if (not p) then (not q))
f t

kalimat G : if (if p then q) then (if (not p) then (not q))
t t t f t

Pohon Semantik -3

Kalimat P: if (if p then q) then (if (not p) then(not q))
f f

Kalimat G : if (if p then q) then (if (not p) then (not q))
t f t f

Pohon Semantik -4

Perhatikan pada Node 4

Pohon Semantik -5

kalimat H	**:**	**if**	**q**	**then**	**(if**	**p**	**then**	**q).**
			t			**?**		**t**

kalimat H	**:**	**if**	**q**	**then**	**(if**	**p**	**then**	**q).**
		t	**t**		**t**	**?**		**t**

Pohon Semantik -6

kalimat H	**:**	**if**	**q**	**then**	**(if**	**p**	**then**	**q).**
		t	**f**		**?**	**?**		**f**

A. Falsifikasikan soal no. 1-2

1. **G : (if p then q) if and only if ((not p) or q)**
2. **if {(not p) or (not q)} then {not (p and q)}**

B. Gunakan pohon semantik untuk mengetahui validasi kalimat H di bawah ini:

kalimat H : if q then (if p then q)

TIFS 1604 Logika Informatika

Semester II

Suraya

Materi

- Logika Predikatif
- Fungsi proposisi
- Kuantor : Universal dan Eksistensial
- Kuantor bersusun

Logika Predikat

- *Logika Predikat* adalah perluasan dari logika proposisi dimana objek yang dibicarakan dapat berupa anggota kelompok.
- logika proposisi (ingat kembali) menganggap proposisi sederhana (kalimat) sebagai entitas tunggal
- Sebaliknya, *logika predikat* membedakan *subjek* dan *predikat* dalam sebuah kalimat.
  - Ingat tentang *subjek* dan *predikat* dalam kalimat?

Penerapan Logika Predikat

Merupakan notasi formal untuk menuliskan secara sempurna *definisi, aksioma, teorema* matematika dengan *jelas, tepat dan tidak ambigu* pada semua cabang matematika.

Logika predikat dengan simbol-simbol fungsi, operator “=”, dan beberapa aturan pembuktian **cukup** untuk mendefinisikan sistem matematika apapun, dan juga cukup untuk membuktikan apapun yang dapat dibuktikan pada sistem tersebut.

Subjek dan Predikat

- Pada kalimat “Kucing itu sedang tidur”:
  - frase “kucing itu” merupakan subjek kalimat
  - frase “sedang tidur” merupakan predikat kalimat- suatu properti yang bernilai TRUE untuk si subjek (objek pelaku)
  - dalam logika predikat, *predikat* dimodelkan sebagai sebuah fungsi $P(\cdot)$ dari objek ke proposisi.
  - $P(x)$ = “x sedang tidur” (x adalah sembarang objek).

Predikat

- Konvensi: variabel huruf kecil *x*, *y*, *z...* menyatakan objek/entitas;

  variabel huruf besar *P*, *Q*, *R*… menyatakan predikat.
- Perhatikan bahwa *hasil dari* menerapkan sebuah **predikat *P*** kepada **objek *x*** adalah sebuah **proposisi *P*(*x*)**. Tapi predikat *P* **sendiri** (*P*="sedang tidur") **bukan sebuah proposisi**
  - Contoh: jika *P*(*x*) = "*x* adalah bilangan prima", *P*(3) adalah *proposisi* : "3 adalah bilangan prima."

Fungsi Proposisi

- Logika predikat dapat digeneralisir untuk menyatakan fungsi proposisi dengan banyak argumen.
  - *Contoh:*

    $P(x,y,z)$ = "x memberikan pada y nilai z"

    jika x="Mike", y="Mary", z="A",

    maka $P(x,y,z)$ = "Mike memberi Mary nilai A."

Fungsi Proposisi

- Fungsi proposisi (kalimat terbuka) :

  Pernyataan yang mengandung satu buah variabel atau lebih.

- Contoh : $x - 3 > 5$.

  Misalkan kita sebut fungsi proposisi ini sebagai P(x), dimana P adalah predikat dan x adalah variabel.

Apakah nilai kebenaran dari P(2) ? Salah

Apakah nilai kebenaran dari P(8) ? Salah

Apakah nilai kebenaran dari P(9) ? Benar

Fungsi Proposisi

- Tinjau fungsi proposisi Q(x, y, z) yg didefinisikan:

$$\mathbf{x + y = z.}$$

- Disini, Q adalah **predikat** dan x, y, and z adalah **variabel**.

Apakah nilai kebenaran dari Q(2, 3, 5) ?	Benar
Apakah nilai kebenaran dari Q(0, 1, 2) ?	Salah
Apakah nilai kebenaran dari Q(9, -9, 0) ?	Benar

Semesta Pembicaraan

- Salah satu kelebihan predikat adalah bahwa predikat memungkinkan kita untuk menyatakan sesuatu tentang banyak objek pada satu kalimat saja.

- Contoh:

  $P(x)$="$x+1>x$". Kita dapat menyatakan bahwa "Untuk sembarang angka x, $P(x)$ bernilai TRUE" hanya dengan satu kalimat daripada harus menyatakan satu-persatu: $(\mathbf{0}+1>\mathbf{0}) \wedge (\mathbf{1}+1>\mathbf{1}) \wedge (\mathbf{2}+1>\mathbf{2}) \wedge \ldots$

- Kumpulan nilai yang bisa dimiliki variabel x disebut semesta pembicaraan untuk x *(x's universe of discourse)*

Ekspresi *Quantifier*

- *Quantifiers* merupakan notasi yang memungkinkan kita untuk mengkuantifikasi (menghitung) *seberapa* banyak objek di semesta pembicaraan yang memenuhi suatu predikat.

- "$\forall$" berarti FOR$\forall$LL (semua) atau *universal* quantifier. $\forall x\ P(x)$ berarti untuk semua x di semesta pembicaraan, P berlaku.

- "$\exists$" berarti $\exists$XISTS (terdapat) atau *existential* quantifier. $\exists x\ P(x)$ berarti terdapat x di semesta pembicaraan (bisa 1 atau lebih) dimana $P(x)$ berlaku.

Predikat & Kuantifier

Pernyataan “x > 3” punya 2 bagian, yakni “x” sebagai subjek dan “ adalah lebih dari 3” sebagai predikat P.

Kita dpt simbolkan pernyataan “x > 3” dengan P(x). Sehingga kita dapat mengevaluasi nilai kebenaran dari P(4) dan P(1).

Subyek dari suatu pernyataan dapat berjumlah lebih dari satu.

Misalkan Q(x,y): $x - 2y > x + y$

Kuantifikasi Universal $\forall$

- Mis. P(x) suatu fungsi proposisi.

  Kalimat yg dikuantifikasi secara universal :

  Untuk semua x dalam semesta pembicaraan, P(x) adalah benar.

- Dengan kuantifier universal $\forall$:
- $\forall x\ P(x)$ "untuk semua x P(x)"

  atau
- "untuk setiap x P(x)"

- (Catatan: $\forall x\ P(x)$ bisa benar atau salah, jadi merupakan *sebuah proposisi*, bukan *fungsi proposisi*.)

Kuantifikasi Universal $\forall$

- Contoh :

  S(x): x adalah seorang mahasiswa IST AKPRIND.

  G(x): x adalah seorang yang pandai.

- Apakah arti dari $\forall x\ (S(x) \rightarrow G(x))$?

- "Jika x adalah mahasiswa IST AKPRIND, maka x adalah seorang yang pandai"

  atau

- "Semua mahasiswa IST AKPRIND pandai."

Kuantifikasi Universal $\forall$

- Contoh:

  Misalkan semesta pembicaraan x adalah tempat parkir di FT IST AKPRIND.

  Misalkan $P(x)$ adalah predikat "x sudah ditempati."

  Maka *universal quantification* untuk $P(x)$, $\forall x\, P(x)$, adalah *proposisi:*

  - "Semua tempat parkir di FT IST AKPRIND sudah ditempati"
  - atau, "Setiap tempat parkir di FT IST AKPRIND sudah ditempati"

Kuantifikasi Universal $\forall$

"P(x) benar untuk semua nilai x dalam domain pembicaraan"

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

Soal 2. Tentukan nilai kebenaran $\forall x\ (x^2 \geq x)$ jika:

x bilangan real

x bilangan bulat

Untuk menunjukkan $\forall x\ P(x)$ salah, cukup dengan mencari satu nilai x dalam domain shg P(x) salah.

Nilai x tersebut dikatakan contoh penyangkal (*counter example*) dari pernyataan $\forall x\ P(x)$.

Kuantifikasi Eksistensial ∃

- **Kalimat yang di-kuantifikasi secara eksistensial**: Ada x di dalam semesta pembicaraan dimana P(x) benar.

- Dengan peng-kuantifikasi eksistensial ∃:
- ∃x P(x) "Ada sebuah x sedemikian hingga P(x)."
  "Ada sedikitnya sebuah x sedemikian hingga P(x)."

- (Catatan: ∃x P(x) bisa benar atau salah, jadi merupakan sebuah proposisi, tapi bukan **fungsi proposisi**.)

Kuantifikasi Eksistensial $\exists$

- Contoh :

  P(x): x adalah seorang dosen IT.

  G(x): x adalah seorang yang pandai.

- Apakah arti $\exists x\ (P(x) \wedge G(x))$?

- "Ada x sedemikian hingga x adalah seorang dosen IT dan x adalah seorang yang pandai."

  atau

- "Sedikitnya satu orang dosen IT adalah seorang yang pandai."

Kuantifikasi Eksistensial ∃

- Contoh lain :

  Misalkan semesta pembicaraan adalah bilangan riil.

- Apakah arti dari ∀**x**∃**y (x + y = 320)** ?

- "Untuk setiap **x** ada **y** sehingga **x + y = 320**."

Apakah pernyataan ini benar ? **Ya**

Apakah ini benar untuk bilangan cacah? **Tidak**

Kuantifikasi Eksistensial ∃

- Contoh:

  Misalkan semesta pembicaraan x adalah tempat parkir di FT IST AKPRIND.

  Misalkan $P(x)$ adalah predikat "x sudah ditempati."

  Maka *existential quantification* untuk $P(x)$, $\exists x\, P(x)$, adalah *proposisi*:

  - "Beberapa tempat parkir di FT IST AKPRIND sudah ditempati"
  - "Ada tempat parkir di FT IST AKPRIND yang sudah ditempati"
  - "Setidaknya satu tempat parkir di FT IST AKPRIND sudah ditempati"

Kuantifikasi Eksistensial ∃

"Ada nilai x dalam domain pembicaraan sehingga P(x) bernilai benar"

$$\exists x\ P(x).$$

Soal 3. Tentukan nilai kebenaran dari ∃x P(x) bila P(x) menyatakan "$x^2 > 12$" dan domain pembicaraan meliputi semua bilangan bulat positif tidak lebih dari 4.

Disproof dengan *counterexample*

- *Counterexample* dari $\forall x\ P(x)$ adalah sebuah objek c sehingga P(c) salah.

- Pernyataan seperti $\forall x\ (P(x) \rightarrow Q(x))$ dapat di-*disproof* secara sederhana dengan memberikan *counterexample*-nya.

Pernyataan: “Semua burung bisa terbang.”

Disproved dengan *counterexample*: Penguin.

Variabel bebas dan variabel terikat

- Sebuah ekspresi seperti $P(x)$ dikatakan memiliki variabel bebas x (berarti, x tidak ditentukan).

- Sebuah *quantifier* ($\forall$ atau $\exists$) *berlaku* pada sebuah ekspresi yang memiliki satu atau lebih variabel bebas, dan *mengikat* satu atau lebih variabel tersebut, untuk membentuk ekspresi yang memiliki satu atau lebih *variabel terikat*.

Contoh Pengikatan

- $P(x,y)$ memiliki 2 variabel bebas, x dan y.
- $\forall x\ P(x,y)$ memilki 1 variabel bebas, dan 1 variabel terikat. [yang mana?]
- "$P(x)$, dimana x=3" adalah cara lain mengikat x.
- Ekspresi dengan <u>nol</u> variabel bebas adalah sebuah proposisi aktual (nyata) : $\forall x\ P(x) \wedge \exists y\ R(y)$
- Ekspresi dengan <u>satu atau lebih</u> variabel bebas adalah sebuah predikat: $\forall x\ P(x,y)$

Negasi

Hubungan antara kuantor universal dengan kuantor eksistensial

E1 : $\neg(\forall x)\, p(x) \equiv (\exists x)\, \neg p(x)$

E2 : $\neg(\exists x)\, p(x) \equiv (\forall x)\, \neg p(x)$

E3 : $\neg(\forall x) p(x) \rightarrow q(x) \equiv (\exists x)\, p(x) \wedge \neg q(x)$

E4 : $\neg(\exists x) p(x) \wedge q(x) \equiv (\forall x)\, p(x) \rightarrow \neg q(x)$

Negasi

"Setiap mhs dalam kelas ini telah mengambil Kalkulus I"
$[\forall x\ P(x)]$

Apakah negasi dari pernyataan ini….?

"Ada seorang mhs dalam kelas ini yang belum mengambil Kalkulus I" $[\exists x \neg P(x)]$

Jadi, $\neg \forall x\ P(x) \equiv \exists x \neg P(x)$.

Negasi (2)

Soal 4. Carilah negasi dari pernyataan berikut:

"Ada politikus yang jujur"

"Semua orang Indonesia makan pecel lele"

Soal 5. Tentukan negasi dari:

$\forall x(x^2 > x)$

$\exists x\ (x^2 = 2)$

Kuantifier Bersusun (Nested Quantifier)

$\forall x\ \forall y\ (x+y = y+x)$

berarti $x+y = y+x$ berlaku untuk semua bilangan real x dan y.

$\forall x\ \exists y\ (x+y = 0)$

berarti untuk setiap x ada nilai y sehingga $x+y = 0$.

$\forall x\ \forall y\ \forall z\ (x+(y+z) = (x+y)+z)$

berarti untuk setiap x, y dan z berlaku hukum asosiatif $x+(y+z) = (x+y)+z$.

Kuantifier Bersusun (Nested Quantifier)

- **Rumusan penting**
  - $(\forall x)\ (\forall y)\ p(x,y) \leftrightarrow (\forall y)\ (\forall x)\ p(x,y)$
  - $(\forall x)\ (\forall y)\ p(x,y) \rightarrow (\exists y)\ (\forall x)\ p(x,y)$
  - $(\exists y)\ (\forall x)\ p(x,y) \rightarrow (\forall x)\ (\exists y)\ p(x,y)$
  - $(\forall x)\ (\exists y)\ p(x,y) \rightarrow (\exists y)\ (\exists x)\ p(x,y)$
  - $(\exists x)\ (\exists y)\ p(x,y) \leftrightarrow (\exists y)\ (\exists x)\ p(x,y)$

Soal-soal

Soal 6. Artikan kalimat ini dalam bhs Indonesia:

$$\forall x\ (C(x) \vee \exists y\ (\ C(y) \wedge F(x,y))),$$

bila C(x) : “x mempunyai komputer”,

F(x,y): “x dan y berteman”,

dan domainnya adalah semua mhs di kampus.

Soal 7. Bagaimana dengan berikut ini:

$$\exists x\ \forall y\ \forall z((F(x,y) \wedge F(x,z) \wedge (y \neq z) \rightarrow \neg F(y,z))$$

Soal 8. Nyatakan negasi dari pernyataan

$$\forall x\ \exists y\ (x*y = 1).$$

Latihan

- Jika $R(x,y)$="x percaya pada y," maka ekspresi dibawah ini berarti:
- $\forall x(\exists y\ R(x,y))$= Semua orang memiliki orang yang dipercayai.
- $\exists y(\forall x\ R(x,y))$= Ada seseorang yang dipercayai oleh semua orang (termasuk dirinya sendiri)
- $\exists x(\forall y\ R(x,y))$= Ada seseorang yang mempercayai semua orang).
- $\forall y(\exists x\ R(x,y))$= Semua orang memiliki seseorang yang mempercayainya
- $\forall x(\forall y\ R(x,y))$= Semua orang mempercayai semua orang, termasuk dirinya sendiri

Konvensi

- Terkadang semesta pembicaraan dibatasi dalam quantification, *contoh*,
  - $\forall x>0\ P(x)$ adalah kependekan dari "untuk semua x lebih besar dari nol, $P(x)$ berlaku."
    $= \forall x\ (x>0 \rightarrow P(x))$
  - $\exists x>0\ P(x)$ adalah kependekan dari "ada x lebih besar dari nol yang membuat $P(x)$ "
    $= \exists x\ (x>0 \wedge P(x))$

Aturan Ekivalensi Quantifier

- Definisi quantifiers:

  semesta pemb. =a,b,c,…

  $\forall x\, P(x) \Leftrightarrow P(\mathrm{a}) \wedge P(\mathrm{b}) \wedge P(\mathrm{c}) \wedge \ldots$

  $\exists x\, P(x) \Leftrightarrow P(\mathrm{a}) \vee P(\mathrm{b}) \vee P(\mathrm{c}) \vee \ldots$

- Kemudian kita bisa membuktikan aturan:

  $\forall x\, P(x) \Leftrightarrow \neg\exists x\, \neg P(x)$

  $\exists x\, P(x) \Leftrightarrow \neg\forall x\, \neg P(x)$

- Aturan ekivalensi *proposisi* mana yang digunakan untuk membuktikannya?

  E1 dan E2

Aturan Ekivalensi Quantifier

- $\forall x\ \forall y\ P(x,y) \Leftrightarrow \forall y\ \forall x\ P(x,y)$
  $\exists x\ \exists y\ P(x,y) \Leftrightarrow \exists y\ \exists x\ P(x,y)$
- $\forall x\ (P(x) \wedge Q(x)) \Leftrightarrow (\forall x\ P(x)) \wedge (\forall x\ Q(x))$
  $\exists x\ (P(x) \vee Q(x)) \Leftrightarrow (\exists x\ P(x)) \vee (\exists x\ Q(x))$
- Latihan:

  Bisakah Anda membuktikan sendiri?
  - Ekivalensi proposisi apa yang Anda gunakan?

Membuat Quantifier Baru

Sesuai namanya, quantifier dapat digunakan untuk menyatakan bahwa sebuah predikat berlaku untuk sembarang kuantitas (jumlah) objek.

Definisikan $\exists!x\ P(x)$ sebagai "$P(x)$ berlaku untuk *tepat satu x* di semesta pembicaraan."

$\exists!x\ P(x) \Leftrightarrow \exists x\ (P(x) \wedge \neg\exists y\ (P(y) \wedge y \neq x))$

"Ada satu x dimana $P(x)$ berlaku, dan tidak ada y dimana P(y) berlaku dan y berbeda dengan x."

Perhatikan

- Semesta pemb. = bilangan cacah 0, 1, 2, …
- "Sebuah bilangan *x* dikatakan *genap*, G(*x*), **iff** *x* sama nilainya dengan bilangan lain dikalikan 2."
  $\forall x\ \mathbf{(}G(x) \leftrightarrow (\exists y\ \ (x{=}2y))\mathbf{)}$
- "Sebuah bilangan *x* dikatakan *prima*, *P*(*x*), **iff** *x* lebih besar dari 1 dan *x* bukan merupakan hasil perkalian dari dua bilangan bukan-satu."
  $\forall x\ \mathbf{(}P(x) \leftrightarrow (x{>}1 \wedge \neg\exists yz\ \ x{=}yz \wedge y{\neq}1 \wedge z{\neq}1)\mathbf{)}$

TIFS 1604 Logika Informatika

Semester II 2009/2010

Suraya

Inferensi

- Definisi:

  Diberikan sejumlah premis A, B, C, D, …

  masing-masing dapat berupa pernyataan yang panjang. Dari premis-premis tersebut dapat disimpulkan K.

- Dapat dituliskan :

  A, B, C, D, …, H C K

Aturan Inferensi

- E.J Lemmon (1965) mendefinisikan 9 aturan inferensi dalam Logika Proposisional

- Asumsi

  Sembarang pernyataan dapat ditambahkan sebagai asumsi pada sembarang langkah penjabaran sebuah argumen

Modus Ponendo Ponens (MPP)

- Diberikan premis berupa sebuah pernyataan konditional A → B, dan premis A sebagai penegasan atas antesedennya, maka konklusinya adalah B
  - $A \rightarrow B, A \vdash B$
  - Ex. 1

    Jika Napoleon orang Perancis maka Napoleon orang Eropa

    Napoleaon orang Perancis

    ∴ Napoleon orang Eropa
  - Ex.2

    Jika ada api maka ada asap

    Benar bahwa ada api

    ∴ Ada asap

Modus Tollendo Tollens (MTT)

- Diberikan premis berupa sebuah pernyataan konditional A → B, dan premis ¬B sebagai sangkalan atas konsekuennya, maka konklusinya adalah ¬A
  - $A \rightarrow B,\ \neg B \vdash \neg A$
  - Ex. 1

    Jika Napoleon orang Perancis maka Napoleon orang Eropa

    Napoleaon bukan orang Eropa

    ∴ Napoleon bukan orang Perancis
  - Ex.2

    Jika ada bug pada program maka program tidak berjalan dengan baik

    Program berjalan dengan baik

    ∴ tidak ada bug

Double Negation

- Diberikan premis P, prinsip ini membawa kita kepada konklusi ¬¬P. Demikian juga sebaliknya, diberikan premis berupa sangkalan rangkap ¬¬P, prinsip ini mengijinkan kita untuk mengambil P sebagai konklusi.
  - $P \vdash \neg\neg P$ atau $\neg\neg P \vdash P$
  - Ex. 1

    Hari ini hujan

    ∴ Tdak benar hari ini tidak hujan

Conditional Proof

- Misalkan sebuah pernyataan B tergantung pada pernyataan A, maka prinsip ini mengijinkan kita untuk membuat konklusi bahwa A → B.
  - A, B ├ A → B
  - Ex. 1

    Ingin dibuktikan bahwa A → B ├ ¬B → ¬A

1. A → B	asumsi diketahui
2. ¬B	asumsi dipilih
3. ¬A	MTT (1,2)
4. ¬B → ¬A	CP (2,3)

Conditional Proof

- Ex. 2

  Ingin dibuktikan bahwa $P \rightarrow (Q \rightarrow R) \vdash Q \rightarrow (P \rightarrow R)$

1. $P \rightarrow (Q \rightarrow R)$	asumsi diketahui
2. Q	asumsi dipilih
3. P	asumsi dipilih
4. $Q \rightarrow R$	MPP (1,3)
5. R	MPP (2,4)
6. $P \rightarrow R$	CP (3,5)
7. $Q \rightarrow (P \rightarrow R)$	CP (2,6)

Introduksi -AND

- Diberikan dua pernyataan A dan B. Aturan inferensi ini mengijinkan untuk mengambil $A \wedge B$ sebagai konklusi.
  - $A, B \vdash A \wedge B$
  - Ex. 1

    Ingin dibuktikan bahwa $(P \wedge Q) \rightarrow R \vdash P \rightarrow (Q \rightarrow R)$

1. $(P \wedge Q) \rightarrow R$	asumsi diketahui
2. P	asumsi dipilih
3. Q	asumsi dipilih
4. $P \wedge Q$	Introduksi-And (2,3)
5. R	MPP(1,4)
6. $Q \rightarrow R$	CP (3,5)
7. $P \rightarrow (Q \rightarrow R)$	CP (2,6)

Eliminasi -AND

- Diberikan dua pernyataan A dan B. Aturan inferensi ini mengijinkan untuk mengambil A ataupun B sebagai konklusi.
  - $A \wedge B \vdash A$ atau $A \wedge B \vdash B$
  - Ex. 1

    Ingin dibuktikan bahwa $Q \rightarrow R \vdash (P \wedge Q) \rightarrow (P \rightarrow R)$

1. $Q \rightarrow R$	asumsi diketahui
2. $P \wedge Q$	asumsi dipilih
3. P	eliminasi-And (2)
4. Q	eliminasi-And (2)
5. R	MPP(1,4)
6. $P \wedge R$	Introduksi-And (3,5)
7. $(P \wedge Q) \rightarrow (P \rightarrow R)$	CP (2,6)

Introduksi -OR

- Diberikan pernyataan A sebagai premis. Aturan inferensi ini mengijinkan untuk mengambil $A \vee B$ sebagai konklusi, apapun pernyataan B.
  - $A \vdash A \vee B$
  - Ex. 1

    A := “Ratu Maria Antoinette dihukum guilotine”

    Introduksi-Or

    $A \vee B$:= “Ratu Maria Antoinette dihukum guilotine atau dihukum kursi listrik”

    dengan

    B := “Ratu Maria Antoinette dihukum kursi listrik”

Eliminasi -OR

- Diberikan $A \vee B$ serta sebuah bukti atas C dengan dasar A sebagai asumsi, serta sebuah bukti C dengan dasar B sebagai asumsi. Maka aturan dg inferensi ini diambil C sebagai konklusi
  - $A \vdash A \vee B$
  - Ex. 1

    Ingin dibuktikan bahwa $P \vee \neg Q, P \rightarrow R, S \rightarrow Q, \neg S \rightarrow R \vdash R$

1. $P \vee \neg Q$	asumsi diketahui
2. $P \rightarrow R$	asumsi diketahui
3. $S \rightarrow Q$	asumsi diketahui
4. $\neg S \rightarrow R$	asumsi diketahui
5. P	asumsi
6. R	MPP (2,5)
7. $\neg Q$	asumsi
8. $\neg S$	MTT (3,7)
9. R	MPP (4,8)
10.R	Eliminasi-Or (1,6,9)

Reductio ad Absordum (RAA)

- Sebuah pernyataan disebut kontradiksi jika dapat ditulis $P \rightarrow \neg P$. Misal dari asumsi A dan asumsi lain dapat dijabarkan sebuah kontradiksi. Maka aturan inferensi mengijinkan kita mengambil $\neg A$ sebagai konklusi.
  - $A \vdash A \vee B$
  - Ex. 1

    Ingin dibuktikan bahwa $P \rightarrow R, R \rightarrow S, S \rightarrow \neg Q \vdash \neg(P \wedge Q)$

1. $P \rightarrow R$	asumsi diketahui
2. $R \rightarrow S$	asumsi diketahui
3. $S \rightarrow \neg Q$	asumsi diketahui
4. $P \wedge Q$	asumsi
5. P	Eliminasi-And (4)
6. R	MPP (1,5)
7. S	MPP (2,6)
8. Q	Eliminasi-And (4)
9. $\neg\neg Q$	DN (8)
10. $\neg S$	MTT (3,9)
11. $\neg(P \wedge Q)$	RAA (4,7,10)

Latihan

- Buktikan dengan inferensi (beserta penjelasan) bahwa argumen berikut adalah valid

  1. Edi atau Andi yang membuat program
  2. Andi menggunakan bahasa Prolog
  3. Jika Andi tidak menguasai bahasa Pascal maka bukan Andi yang membuat program itu
  4. Jika Andi menguasai bahasa Pascal maka Andi tidak menggunakan bahasa Prolog
  5. Jadi Edi yang membuat program itu

TIFS 1604 Logika Informatika

Semester II

Suraya

Inferensi

- Definisi:

  Diberikan sejumlah premis A, B, C, D, …

  masing-masing dapat berupa pernyataan yang panjang. Dari premis-premis tersebut dapat disimpulkan K.

- Dapat dituliskan :

  A, B, C, D, …, H C K

Aturan Inferensi

- E.J Lemmon (1965) mendefinisikan 9 aturan inferensi dalam Logika Proposisional

- Asumsi

  Sembarang pernyataan dapat ditambahkan sebagai asumsi pada sembarang langkah penjabaran sebuah argumen

Modus Ponendo Ponens (MPP)

- Diberikan premis berupa sebuah pernyataan konditional A → B, dan premis A sebagai penegasan atas antesedennya, maka konklusinya adalah B
  - $A \rightarrow B, A \vdash B$
  - Ex. 1

    Jika Napoleon orang Perancis maka Napoleon orang Eropa

    Napoleaon orang Perancis

    ∴ Napoleon orang Eropa
  - Ex.2

    Jika ada api maka ada asap

    Benar bahwa ada api

    ∴ Ada asap

Modus Tollendo Tollens (MTT)

- Diberikan premis berupa sebuah pernyataan konditional A → B, dan premis ¬B sebagai sangkalan atas konsekuennya, maka konklusinya adalah ¬A
  - $A \rightarrow B,\ \neg B \vdash \neg A$
  - Ex. 1

    Jika Napoleon orang Perancis maka Napoleon orang Eropa

    Napoleaon bukan orang Eropa

    ∴ Napoleon bukan orang Perancis
  - Ex.2

    Jika ada bug pada program maka program tidak berjalan dengan baik

    Program berjalan dengan baik

    ∴ tidak ada bug

Double Negation

- Diberikan premis P, prinsip ini membawa kita kepada konklusi ¬¬P. Demikian juga sebaliknya, diberikan premis berupa sangkalan rangkap ¬¬P, prinsip ini mengijinkan kita untuk mengambil P sebagai konklusi.
  - $P \vdash \neg\neg P$ atau $\neg\neg P \vdash P$
  - Ex. 1

    Hari ini hujan

    ∴ Tdak benar hari ini tidak hujan

Conditional Proof

- Misalkan sebuah pernyataan B tergantung pada pernyataan A, maka prinsip ini mengijinkan kita untuk membuat konklusi bahwa A → B.
  - A, B ├ A → B
  - Ex. 1

    Ingin dibuktikan bahwa A → B ├ ¬B → ¬A

1. A → B	asumsi diketahui
2. ¬B	asumsi dipilih
3. ¬A	MTT (1,2)
4. ¬B → ¬A	CP (2,3)

Conditional Proof

- Ex. 2

  Ingin dibuktikan bahwa $P \rightarrow (Q \rightarrow R) \vdash Q \rightarrow (P \rightarrow R)$

1. $P \rightarrow (Q \rightarrow R)$	asumsi diketahui
2. Q	asumsi dipilih
3. P	asumsi dipilih
4. $Q \rightarrow R$	MPP (1,3)
5. R	MPP (2,4)
6. $P \rightarrow R$	CP (3,5)
7. $Q \rightarrow (P \rightarrow R)$	CP (2,6)

Introduksi -AND

- Diberikan dua pernyataan A dan B. Aturan inferensi ini mengijinkan untuk mengambil $A \wedge B$ sebagai konklusi.
  - $A, B \vdash A \wedge B$
  - Ex. 1

    Ingin dibuktikan bahwa $(P \wedge Q) \rightarrow R \vdash P \rightarrow (Q \rightarrow R)$

1. $(P \wedge Q) \rightarrow R$	asumsi diketahui
2. P	asumsi dipilih
3. Q	asumsi dipilih
4. $P \wedge Q$	Introduksi-And (2,3)
5. R	MPP(1,4)
6. $Q \rightarrow R$	CP (3,5)
7. $P \rightarrow (Q \rightarrow R)$	CP (2,6)

Eliminasi -AND

- Diberikan dua pernyataan A dan B. Aturan inferensi ini mengijinkan untuk mengambil A ataupun B sebagai konklusi.
  - $A \wedge B \vdash A$ atau $A \wedge B \vdash B$
  - Ex. 1

    Ingin dibuktikan bahwa $Q \rightarrow R \vdash (P \wedge Q) \rightarrow (P \rightarrow R)$

1. $Q \rightarrow R$	asumsi diketahui
2. $P \wedge Q$	asumsi dipilih
3. P	eliminasi-And (2)
4. Q	eliminasi-And (2)
5. R	MPP(1,4)
6. $P \wedge R$	Introduksi-And (3,5)
7. $(P \wedge Q) \rightarrow (P \rightarrow R)$	CP (2,6)

Introduksi -OR

- Diberikan pernyataan A sebagai premis. Aturan inferensi ini mengijinkan untuk mengambil $A \vee B$ sebagai konklusi, apapun pernyataan B.
  - $A \vdash A \vee B$
  - Ex. 1

    A := “Ratu Maria Antoinette dihukum guilotine”

    Introduksi-Or

    $A \vee B$:= “Ratu Maria Antoinette dihukum guilotine atau dihukum kursi listrik”

    dengan

    B := “Ratu Maria Antoinette dihukum kursi listrik”

Eliminasi -OR

- Diberikan $A \vee B$ serta sebuah bukti atas C dengan dasar A sebagai asumsi, serta sebuah bukti C dengan dasar B sebagai asumsi. Maka aturan dg inferensi ini diambil C sebagai konklusi
  - $A \vdash A \vee B$
  - Ex. 1

    Ingin dibuktikan bahwa $P \vee \neg Q, P \rightarrow R, S \rightarrow Q, \neg S \rightarrow R \vdash R$

1. $P \vee \neg Q$	asumsi diketahui
2. $P \rightarrow R$	asumsi diketahui
3. $S \rightarrow Q$	asumsi diketahui
4. $\neg S \rightarrow R$	asumsi diketahui
5. P	asumsi
6. R	MPP (2,5)
7. $\neg Q$	asumsi
8. $\neg S$	MTT (3,7)
9. R	MPP (4,8)
10. R	Eliminasi-Or (1,6,9)

Reductio ad Absordum (RAA)

- Sebuah pernyataan disebut kontradiksi jika dapat ditulis $P \rightarrow \neg P$. Misal dari asumsi A dan asumsi lain dapat dijabarkan sebuah kontradiksi. Maka aturan inferensi mengijinkan kita mengambil $\neg A$ sebagai konklusi.
  - $A \vdash A \vee B$
  - Ex. 1

    Ingin dibuktikan bahwa $P \rightarrow R, R \rightarrow S, S \rightarrow \neg Q \vdash \neg(P \wedge Q)$

1. $P \rightarrow R$	asumsi diketahui
2. $R \rightarrow S$	asumsi diketahui
3. $S \rightarrow \neg Q$	asumsi diketahui
4. $P \wedge Q$	asumsi
5. P	Eliminasi-And (4)
6. R	MPP (1,5)
7. S	MPP (2,6)
8. Q	Eliminasi-And (4)
9. $\neg\neg Q$	DN (8)
10. $\neg S$	MTT (3,9)
11. $\neg(P \wedge Q)$	RAA (4,7,10)

Latihan

- Buktikan dengan inferensi (beserta penjelasan) bahwa argumen berikut adalah valid
  1. Edi atau Andi yang membuat program
  2. Andi menggunakan bahasa Prolog
  3. Jika Andi tidak menguasai bahasa Pascal maka bukan Andi yang membuat program itu
  4. Jika Andi menguasai bahasa Pascal maka Andi tidak menggunakan bahasa Prolog
  5. Jadi Edi yang membuat program itu

TIFS 1604 Logika Informatika

Semester II

Suraya

Deduksi

- Definisi:

  s :≡ Socrates (filsuf Yunani kuno);

  $H(x)$:≡ "x is human";

  $M(x)$:≡ "x mortal".

- Premis:

  $H(\mathrm{s})$ *Socrates manusia.*

  $\forall x(\, H(x) \rightarrow M(x))$ *Semua manusia pasti mati.*

Deduksi

<u>Kesimpulan valid yang dapat diambil:</u>

- $H(s) \rightarrow M(s)$ **[Instantiate universal.]**
  If Socrates is human then he is mortal.
- $\neg H(s) \vee M(s)$ *Socrates is inhuman or mortal.*
- $H(s) \wedge (\neg H(s) \vee M(s))$ *Socrates is human, and also either inhuman or mortal.*
- $(H(s) \wedge \neg H(s)) \vee (H(s) \wedge M(s))$ **[Apply distributive law.]**
- $F \vee (H(s) \wedge M(s))$ **[Trivial contradiction.]**
- $H(s) \wedge M(s)$ **[Use identity law.]**
- $M(s)$ *Socrates is mortal.*

Contoh Lain

- Definisi:

  $H(x)$:≡ "x is human";
  $M(x)$:≡ "x is mortal";

  $G(x)$:≡ "x is a god"
- Premis:
  - $\forall x\ (H(x) \rightarrow M(x))$ ("Humans are mortal") and
  - $\forall x(\ G(x) \rightarrow \neg M(x))$ ("Gods are immortal").
- Buktikan $\neg\exists x\ (H(x) \wedge G(x))$
  ("No human is a god.")

Derivasi

- $\forall x\ (H(x) \rightarrow M(x))$ and $(\forall x\ G(x) \rightarrow \neg M(x).)$
- $\forall x\ (\neg M(x) \rightarrow \neg H(x))$ **[Contrapositive.]**
- $\forall x\ ([G(x) \rightarrow \neg M(x)] \wedge [\neg M(x) \rightarrow \neg H(x)])$
- $\forall x\ (G(x) \rightarrow \neg H(x))$ **[Transitivity of $\rightarrow$.]**
- $\forall x\ (\neg G(x) \vee \neg H(x))$ **[Definition of $\rightarrow$.]**
- $\forall x\ (\neg (G(x) \wedge H(x)))$ **[DeMorgan's law.]**
- $\neg \exists x\ (G(x) \wedge H(x))$ **[An equivalence law.]**

Derivasi

- Universal Instantiation (UI)
  - Aturan bagaimana $\forall$ dieliminasi dg operasi Instansiasi

$$\frac{\forall x(A)}{S_t^x(A)}$$

  - Ex. 1

$$\forall x\left(f(x) = x^3 - 2\right) \Rightarrow S_4^x\left(f(4) = 4^3 - 2\right) = 62$$

  - Ex.2

$$\frac{\forall x\ (\text{cat}(x) \Rightarrow \text{hastail}(x))}{\text{cat}(\text{Tom}) \Rightarrow \text{hastail}(\text{Tom})}$$

Derivasi

- Derivasi dg Universal **Instantiation (UI)**
  - *Ex.*

    $H(x) :\equiv$ "x is human";
    $M(x) :\equiv$ "x mortal".

    S :≡ Socrates (filsuf Yunani kuno);

  Prove : $\forall x\ (H(x) \Rightarrow M(x)),\ H(S) \vdash M(S)$
  - *Derivation*

1. $\forall x\ (H(x) \Rightarrow M(x))$	premise	all humans are mortal
2. $H(S)$	premise	*Socrates is human*
3. $(H(S) \Rightarrow M(S)$	S^x_S	If Socrates is human, he is mortal
4. $M(S)$	2,3 MP	Socrates is mortal

Derivasi

- Derivasi dg Universal **Instantiation (UI)**
  - *Ex.*

    f(x,y) :≡ "*x* is the father of y";
    s(*x,y*) :≡ "*x* is the son of y".
    d(*x,y*) :≡ "*x* is the daughter of y".
    D :≡ Daug; P :≡ Paul

  Prove : $\forall x\ (f(D,x) \Rightarrow s(x,D) \vee d(x,D)),\ f(D,P),\ \neg d(P,D) \vdash s(P,D)$
  - *Derivation*

1. $\forall x\ (f(D,x) \Rightarrow s(x,D) \vee d(x,D))$	premise
2. $f(D,P)$	premise
3. $\neg d(P,D)$	premise
4. $f(D,P) \Rightarrow s(P,D) \vee d(P,D)$	S^x_S
5. $s(P,D) \vee d(P,D)$	2,4 MP
6. $s(P,D)$	3,5 DS

Derivasi

- Universal Generalization (UG)
  - Aturan bagaimana ∀ digeneralisasi :Statement yg berlaku lokal menjadi statement yg berlaku global

$$\frac{A}{\forall x(A)}$$

  - Ex. 2

$$\frac{\begin{array}{l}\forall x\ (P(x)) \\ \forall x\ (P(x) \Rightarrow Q(Tom))\end{array}}{\forall x\ (Q(x))}$$

P(x) :≡ ‘x mhs TI’;

Q(x) :≡ ‘x menyukai programming’

Derivasi

- Derivasi dg Universal Generalization (UG)
  - Prove : $\forall x\ P(x), \forall x\ (P(x) \Rightarrow Q(x)) \vdash \forall x\ Q(x)$
  - *Derivation*

1. $\forall x\ P(x)$	premise
2. $\forall x\ (P(x) \Rightarrow Q(x))$	premise
3. $P(x)$	1, S^x_x UI
4. $P(x) \Rightarrow Q(x)$	2, S^x_x UI
5. $Q(x)$	3,4 MP
6. $\forall x\ Q(x)$	5 UG

  - Prove : $\forall x\ \forall y P(x,y) \vdash \forall y\ \forall x P(x,y)$
  - Prove : $\forall x\ P(x) \vdash \forall y\ P(y)$

Derivasi

- Existential Generalization (EG)
  - Aturan bagaimana ∃ digeneralisasi

$$\frac{S_t^x(A)}{\exists x(A)}$$

  - Ex. 1

    C :≡ ‘bibi Cordelia’;

    P(x) :≡ ‘x berumur lebih dari 100 tahun’;

$$\frac{P(C)}{\exists x P(x)}$$

  - Ex.2

    Setiap orang yang menang 1 milyar pasti kaya

    Mary menang 1 milyar

    Ada orang yang kaya

Derivasi

- Derivasi dg Existential Generalization (EG)
  - *Ex.*

    $W(x) :\equiv$ "*x* memenangkan 1 milyar";
    $R(x) :\equiv$ "*x* orang yang kaya".
    $M :\equiv$ "Mary";

  Prove : $\forall x\,(W(x) \Rightarrow R(x)),\ W(M) \vdash \exists x R(x)$
  - *Derivation*

1. $\forall x\,(W(x) \Rightarrow R(x))$	premise
2. $(W(M) \Rightarrow R(M)$	1, S^x_M
3. $W(M)$	premise
4. $R(M)$	2,3 MP
5. $\exists x R(x)$	4 EG

Derivasi

- Existential Instantiation (EI)
  - Aturan bagaimana ∃ dieliminasi

$$\frac{\exists x(A)}{S_t^x(A)}$$

  - Ex. 1

    P(x) :≡ ‘x does somersaults’;

    ∃xP(x) :≡ ‘somebody makes somersaults’;

$$S_t^x P(x) = P(t)$$

  - Ex.2

    Seseorang menang 1 milyar

    Setiap orang yg memiliki 1 milyar pasti kaya

    Ada seseorang yang kaya

Derivasi

- Derivasi dg Existential Instantiation (EI)
  - *Ex.*

    $W(x) :\equiv$ "*x* memenangkan 1 milyar";
    $R(x) :\equiv$ "*x* orang yang kaya".

    $b :\equiv$ "x"

  Prove : $\forall x\,(W(x) \Rightarrow R(x)),\ \exists x\, W(x) \vdash \exists x R(x)$
  - *Derivation*

1. $\exists x\, W(x)$	premise
2. $W(b)$	1, EI
3. $\forall x\,(W(x) \Rightarrow R(x))$	premise
4. $W(b) \Rightarrow R(b)$	3, S^x_b
5. $R(b)$	2,4 MP
6. $\exists x R(x)$	4, EG

LOGIKA INFORMATIKA

Suraya

Jurusan Teknik Informatika

Materi Perkuliahan

- Teori Himpunan
- Cara Penyajian Himpunan
- Kardinalitas
- Operasi himpunan
- Dualitas
- Pembuktian

Himpunan (Set)

- Himpunan (*set*) adalah kumpulan objek-objek yang *berbeda*.
- Objek di dalam himpunan disebut **elemen**, **unsur**, atau **anggota**.

Cara Penyajian Himpunan

- ***<u>Enumerasi</u>***

- **Contoh 1.**
  - Himpunan empat bilangan asli pertama: A = {1, 2, 3, 4}.
  - C = {kucing, a, Amir, 10, paku}
  - R = { a, b, {a, b, c}, {a, c} }
  - C = {a, {a}, {{a}} }
  - K = { {} }
  - Himpunan 100 buah bilangan asli pertama: {1, 2, ..., 100 }
  - Himpunan bilangan bulat ditulis sebagai {…, -2, -1, 0, 1, 2, …}.

Cara Penyajian Himpunan

- **Keanggotaan**
  - $x \in A$: x merupakan anggota himpunan A;
  - $x \notin A$: x bukan merupakan anggota himpunan A.

- **Contoh 2.**
- Misalkan:
  $A = \{1, 2, 3, 4\}$, $R = \{ a, b, \{a, b, c\}, \{a, c\} \}$ dan $K = \{\{\}\}$
- maka
  $3 \in A$
  $\{a, b, c\} \in R$
  $c \notin R$
  $\{\} \in K$
  $\{\} \notin R$

Cara Penyajian Himpunan

- **Contoh 3.**
- Bila

  $P1 = \{a, b\}$, $P2 = \{ \{a, b\} \}$, $P3 = \{\{\{a, b\}\}\}$,

  maka

  $a \in P1$

  $a \notin P2$

  $P1 \in P2$

  $P1 \notin P3$

  $P2 \in P3$

Cara Penyajian Himpunan

- ***<u>Simbol-simbol Baku</u>***

  P = himpunan bilangan bulat positif = { 1, 2, 3, ... }

  N = himpunan bilangan alami (natural) = { 1, 2, ... }

  Z = himpunan bilangan bulat = { ..., -2, -1, 0, 1, 2, ... }

  Q = himpunan bilangan rasional

  R = himpunan bilangan riil

  C = himpunan bilangan kompleks
- Himpunan yang universal: **semesta**, disimbolkan dengan U.
- Contoh: Misalkan U = {1, 2, 3, 4, 5} dan *A* adalah himpunan bagian dari U, dengan *A* = {1, 3, 5}.

Cara Penyajian Himpunan

- ***<u>Notasi Pembentuk Himpunan</u>***

  Notasi: { x | syarat yang harus dipenuhi oleh x }

- **Contoh 4.**

  (i) A adl himp. bilangan bulat positif yang lebih kecil dari 5

  A = { x | x adalah bilangan bulat positif lebih kecil dari 5}

  atau

  $A = \{ x \mid x \in P, x < 5 \}$ yang ekivalen dgn A = {1, 2, 3, 4}

  (ii) M = { x | x adalah mahasiswa yang mengambil kuliah TIFS 1604}

Cara Penyajian Himpunan

- ***<u>Diagram Venn</u>***
- **Contoh 5.**
- **Misalkan U = {1, 2, …, 7, 8},**
  ***A* = {1, 2, 3, 5} dan *B* = {2, 5, 6, 8}.**
- **Diagram Venn:**

Kardinalitas

- ***<u>Jumlah elemen di dalam A disebut kardinal dari himpunan A.</u>***

  Notasi: n(A) atau $|A|$
- ***<u>Contoh 6.</u>***

  (i) B = { x | x merupakan bilangan prima yang lebih kecil dari 20 },

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

  (ii) T = {kucing,a,Amir,10,paku}, maka $|T| = 5$

  (iii) A = {a, {a}, {{a}} }, maka $|A| = 3$

Himpunan Kosong

- *Himpunan dengan kardinal = 0 disebut himpunan kosong*

  Notasi : $\varnothing$ atau {}
- *Contoh 7.*

  (i) E = { x | x < x }, maka n(E) = 0

  (ii) P = { orang Indonesia yang pernah ke bulan },maka n(P) = 0

  (iii) A = {x |x adalah akar persamaan kuadrat $x^2 + 1 = 0$}, n(A) = 0
- *himpunan {{ }} dapat juga ditulis sebagai {$\varnothing$}*
- *himpunan {{ }, {{ }}} dapat juga ditulis sebagai {$\varnothing$, {$\varnothing$}}*
- *{$\varnothing$} bukan himpunan kosong karena ia memuat satu elemen yaitu himpunan kosong.*

Himpunan Bagian (Subset)

- ***Himpunan A dikatakan himpunan bagian dari himpunan B jika dan hanya jika setiap elemen A merupakan elemen dari B.***
- ***Dalam hal ini, B dikatakan superset dari A.***

  Notasi: A $\subseteq$ B
- ***Diagram Venn:***

Himpunan Bagian (Subset)

- ***Contoh 8.***

(i) $\{1, 2, 3\} \subseteq \{1, 2, 3, 4, 5\}$

(ii) $\{1, 2, 3\} \subseteq \{1, 2, 3\}$

(iii) $N \subseteq Z \subseteq R \subseteq C \rightarrow$

(iv) Jika $A = \{ (x, y) \mid x + y < 4, x \geq, y \geq 0 \}$ dan

$B = \{ (x, y) \mid 2x + y < 4, x \geq 0$ dan $y \geq 0 \}$,

maka $B \subseteq A$.

Himpunan Bagian (Subset)

- **TEOREMA 1.**

  Untuk sembarang himpunan A berlaku hal-hal sebagai berikut:

  (a) A adalah himpunan bagian dari A itu sendiri (yaitu, $A \subseteq A$).

  (b) Himpunan kosong merupakan himpunan bagian dari A ($\varnothing \subseteq A$).

  (c) Jika $A \subseteq B$ dan $B \subseteq C$, maka $A \subseteq C$

Himpunan Bagian (Subset)

- *$\varnothing \subseteq A$ dan $A \subseteq A$, maka $\varnothing$ dan A disebut himpunan bagian tak sebenarnya (improper subset) dari himpunan A.*
- *Contoh: A = {1, 2, 3}, maka {1, 2, 3} dan $\varnothing$ adalah improper subset dari A.*
- *$A \subseteq B$ berbeda dengan $A \subset B$*

  (i) $A \subset B$:A adalah himpunan bagian dari B tetapi $A \neq B$.

  A adalah himpunan bagian sebenarnya (proper subset) dari B.

  Contoh: {1} dan {2, 3} adalah proper subset dari {1, 2, 3}

  (ii) $A \subseteq B$: digunakan untuk menyatakan bahwa A adalah himpunan bagian (subset) dari B yang memungkinkan A = B.

Himpunan yang Sama

- $A = B$ jika dan hanya jika setiap elemen A merupakan elemen B dan sebaliknya setiap elemen B merupakan elemen A.
- $A = B$ jika A adalah himpunan bagian dari B dan B adalah himpunan bagian dari A. Jika tidak demikian, maka $A \neq B$.
- Notasi : $A = B \leftrightarrow A \subseteq B$ dan $B \subseteq A$

Himpunan yang Sama

- ***Contoh 9.***

(i) Jika A = { 0, 1 } dan B = { x | x (x – 1) = 0 }, maka A = B

(ii) Jika A = { 3, 5, 8, 5 } dan B = {5, 3, 8 }, maka A = B

(iii) Jika A = { 3, 5, 8, 5 } dan B = {3, 8}, maka A $\neq$ B

- *Untuk tiga buah himpunan, A, B, dan C berlaku aksioma berikut:*

(a) A = A, B = B, dan C = C

(b) jika A = B, maka B = A

(c) jika A = B dan B = C, maka A = C

Himpunan yang Ekivalen

- *Himpunan A dikatakan ekivalen dengan himpunan B jika dan hanya jika kardinal dari kedua himpunan tersebut sama.*

  Notasi : A ~ B $\leftrightarrow |A| = |B|$
- ***Contoh 10.***
- *Misalkan A = { 1, 3, 5, 7 } dan B = { a, b, c, d }, maka A ~ B sebab* $|A| = |B| = 4$

Himpunan yang Saling Lepas

- *Dua himpunan A dan B dikatakan saling lepas (disjoint) jika keduanya tidak memiliki elemen yang sama.*

  Notasi : A // B
- *Diagram Venn:*

- **Contoh 11.**

  Jika $A = \{ x \mid x \in P, x < 8 \}$ dan $B = \{ 10, 20, 30, ...\}$, maka $A // B$.

Himpunan Kuasa

- Himpunan kuasa (*power set*) dari himpunan A adalah suatu himpunan yang elemennya merupakan semua himpunan bagian dari A, termasuk himpunan kosong dan himpunan A sendiri.

  Notasi : $P(A)$ atau $2A$
- Jika $|A| = m$, maka $|P(A)| = 2^m$.
- **Contoh 12.**

  Jika $A = \{ 1, 2 \}$, maka $P(A) = \{ , \{ 1 \}, \{ 2 \}, \{ 1, 2 \}\}$

- **Contoh 13.**

  Himpunan kuasa dari himpunan kosong adalah $P(\varnothing) = \{\varnothing\}$,

  dan himpunan kuasa dari himpunan $\{\varnothing\}$ adalah $P(\{\varnothing\}) = \{\varnothing, \{\varnothing\}\}$.

Operasi terhadap Himpunan

<u>Intersection</u>

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

- **Contoh 14.**

  (i) Jika $A = \{2, 4, 6, 8, 10\}$ dan $B = \{4, 10, 14, 18\}$,

  maka $A \cap B = \{4, 10\}$

  (ii) Jika $A = \{ 3, 5, 9 \}$ dan $B = \{ -2, 6 \}$, maka $A \cap B = \varnothing$.

- Artinya: $A \,//\, B$

Operasi terhadap Himpunan

<u>Union</u>

- Notasi : $A \cup B = \{ x \mid x \in A \text{ atau } x \in B \}$

- **Contoh 15.**

  (i) Jika $A = \{ 2, 5, 8 \}$ dan $B = \{ 7, 5, 22 \}$, maka $A \cup B = \{ 2, 5, 7, 8, 22 \}$

  (ii) $A \cup \varnothing = A$

Operasi terhadap Himpunan

<u>Complement</u>

- Notasi : $\overline{A} = \{ x \mid x \in U, x \notin A \}$

- **Contoh 16.**

  Misalkan U = { 1, 2, 3, ..., 9 },

  jika A = {1, 3, 7, 9}, maka $\overline{A}$ = {2, 4, 6, 8}

  jika $A = \{ x \mid x/2\ P, x < 9 \}$, maka $\overline{A}$ = { 1, 3, 5, 7, 9 }

Operasi terhadap Himpunan

- **Contoh 17.** Misalkan:

  A = himpunan semua mobil buatan dalam negeri

  B = himpunan semua mobil impor

  C = himpunan semua mobil yang dibuat sebelum tahun 1990

  D = himpunan semua mobil yang nilai jualnya kurang dari Rp 100 juta

  E = himpunan semua mobil milik mahasiswa IST-AKPRIND

  "mobil mahasiswa di IST-AKPRIND produksi dalam negeri atau diimpor dari luar negeri"

  "semua mobil produksi dalam negeri yang dibuat sebelum tahun 1990 yang nilai jualnya kurang dari Rp 100 juta"

  "semua mobil impor buatan setelah tahun 1990 mempunyai nilai jual lebih dari Rp 100 juta"

Operasi terhadap Himpunan

* $(E \cap A) \cup (E \cap B)$ atau $E \cap (A \cup B)$
- $A \cap C \cap D$
- $\overline{C} \cap \overline{D} \cap B$

Operasi terhadap Himpunan

<u>Difference</u>

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

- **Contoh 18.**

  (i) Jika $A = \{ 1, 2, 3, ..., 10 \}$ dan $B = \{ 2, 4, 6, 8, 10 \}$, maka $A - B = \{ 1, 3, 5, 7, 9 \}$ dan $B - A = \varnothing$

  (ii) $\{1, 3, 5\} - \{1, 2, 3\} = \{5\}$, tetapi $\{1, 2, 3\} - \{1, 3, 5\} = \{2\}$

Operasi terhadap Himpunan

<u>Symmetric Difference</u>

- Notasi : $A \oplus B = (A \cup B) - (A \cap B) = (A - B) \cup (B - A)$

- **Contoh 19.**

  Jika A ={ 2, 4, 6 } dan B ={ 2, 3, 5 }, maka $A \oplus B$ = { 3, 4, 5, 6 }

- **Contoh 20.**

  U = himpunan mahasiswa

  P = himpunan mahasiswa yang nilai ujian UTS di atas 80

  Q = himpunan mahasiswa yang nilain ujian UAS di atas 80

  Seorang mahasiswa mendapat nilai A jika nilai UTS dan nilai UAS keduanya di atas 80, mendapat nilai B jika salah satu ujian di atas 80, dan mendapat nilai C jika kedua ujian di bawah 80.

  (i) "Semua mahasiswa yang mendapat nilai A" : $P \cap Q$

  (ii) "Semua mahasiswa yang mendapat nilai B" : $P \oplus Q$

  (iii) "Semua mahasiswa yang mendapat nilai C" : U – ($P \cup Q$)

Operasi terhadap Himpunan

<u>Symmetric Difference</u>

- **TEOREMA 2.** Beda setangkup memenuhi sifat-sifat berikut:

  (a) $A \oplus B = B \oplus A$ (hukum komutatif)

  (b) $(A \oplus B) \oplus C = A \oplus (B \oplus C)$ (hukum asosiatif)

Operasi terhadap Himpunan

<u>Cartesian Product</u>

- Notasi: $A \times B = \{(a, b) \mid a \in A \text{ dan } b \in B\}$

- **Contoh 20.**

  (i) Misalkan $C = \{1, 2, 3\}$, dan $D = \{a, b\}$, maka
  $C \times D = \{(1, a), (1, b), (2, a), (2, b), (3, a), (3, b)\}$

  (ii) Misalkan A=B=himp. semua bilangan riil, maka
  $A \times B$ = himpunan semua titik di bidang datar

Operasi terhadap Himpunan

<u>Cartesian Product</u>

- Catatan:
  1. Jika A dan B merupakan himpunan berhingga, maka:
     $|A \times B| = |A| \,.\, |B|$.
  2. Pasangan berurutan (a,b) berbeda dengan (b,a), dengan kata lain $(a, b) \neq (b, a)$.
  3. Perkalian kartesian tidak komutatif, yaitu $A \times B \neq B \times A$ dengan syarat A atau B tidak kosong. Pada Contoh 20(i) di atas, $D \times C = \{(a, 1), (a, 2), (a, 3), (b, 1), (b, 2), (b, 3)\} \neq C \times D$.
  4. Jika $A = \varnothing$ atau $B = \varnothing$, maka $A \times B = B \times A = \varnothing$

Operasi terhadap Himpunan

<u>Cartesian Product</u>

Contoh 21.

Misalkan

A = himpunan makanan = { *s* = soto, *g* = gado-gado, *n* = nasi goreng, *m* = mie rebus }

B = himpunan minuman = { *c* = coca-cola, *t* = teh, *d* = es dawet }

Berapa banyak kombinasi makanan dan minuman yang dapat disusun dari kedua himpunan di atas?

Cartesian Product

Jawab:

$|A \times B| = |A| \cdot |B| = 4 \cdot 3 = 12$ kombinasi makanan dan minuman, yaitu

$\{(s, c), (s, t), (s, d), (g, c), (g, t), (g, d), (n, c), (n, t), (n, d), (m, c), (m, t), (m, d)\}$.

Operasi terhadap Himpunan

Contoh 22.

- Daftarkan semua anggota himpunan berikut:

  (a) $P(\varnothing)$

  (b) $\varnothing \times P(\varnothing)$

  (c) $\{\varnothing\} \times P(\varnothing)$

  (d) $P(P(\{3\}))$

Operasi terhadap Himpunan

<u>Penyelesaian</u>:

- $P(\varnothing) = \{\varnothing\}$
- $\varnothing \times P(\varnothing) = \varnothing$ (ket: jika $A = \varnothing$ atau $B = \varnothing$ maka $A \times B = \varnothing$)
- $\{\varnothing\} \times P(\varnothing) = \{\varnothing\} \times \{\varnothing\} = \{(\varnothing,\varnothing)\}$
- $P(P(\{3\})) = P(\{\varnothing, \{3\}\}) = \{\varnothing, \{\varnothing\}, \{\{3\}\}, \{\varnothing, \{3\}\}\}$

Operasi terhadap Himpunan

Contoh 23.

(i) $A\ (B1B2\ \ ...\ Bn) = (A\ B1)\ (A\ B2)$

$...\ (A\ Bn)$

(ii) Misalkan $A = \{1, 2\}$, $B = \{a, b\}$,

dan $C = \{\alpha, \beta\}$,

cari kombinasi A x B x C

Operasi terhadap Himpunan

Jawab:

$$A \times B \times C = \{(1, a, \alpha), (1, a, \beta), (1, b, \alpha), (1, b, \beta), (2, a, \alpha), (2, a, \beta), (2, b, \alpha), (2, b, \beta)\}$$

Operasi terhadap Himpunan

- Hukum-hukum Himpunan

1. Hukum identitas: - $A \cup \varnothing = A$ - $A \cap \mathrm{U} = A$	2. Hukum *null*/dominasi: - $A \cap \varnothing = \varnothing$ - $A \cup \mathrm{U} = \mathrm{U}$
3. Hukum komplemen: - $A \cup \overline{A} = \mathrm{U}$ - $A \cap \overline{A} = \varnothing$	4. Hukum idempoten: - $A \cup A = A$ - $A \cap A = A$
5. Hukum involusi: - $\overline{(\overline{A})} = A$	6. Hukum penyerapan (absorpsi): - $A \cup (A \cap B) = A$ - $A \cap (A \cup B) = A$
7. Hukum komutatif: - $A \cup B = B \cup A$ - $A \cap B = B \cap A$	8. Hukum asosiatif: - $A \cup (B \cup C) = (A \cup B) \cup C$ - $A \cap (B \cap C) = (A \cap B) \cap C$

Operasi terhadap Himpunan

- Hukum-hukum Himpunan

9. Hukum distributif: - $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$ - $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$	10. Hukum De Morgan: - $\overline{A \cap B} = \overline{A} \cup \overline{B}$ - $\overline{A \cup B} = \overline{A} \cap \overline{B}$
11. Hukum 0/1 - $\overline{\varnothing} = U$ - $\overline{U} = \varnothing$	

Dualitas

Prinsip Dualitas

- Prinsip dualitas: dua konsep yang berbeda dapat dipertukarkan namun tetap memberikan jawaban yang benar.

  Contoh: AS → kemudi mobil di kiri depan
  Inggris (juga Indonesia) → kemudi mobil di kanan depan

  Peraturan:

  (a) di Amerika Serikat,
  - mobil harus berjalan di bagian *kanan* jalan,
  - pada jalan yang berlajur banyak, lajur *kiri* untuk mendahului,
  - bila lampu merah menyala, mobil belok *kanan* boleh langsung

  (b) di Inggris,
  - mobil harus berjalan di bagian *kiri* jalan,
  - pada jalur yang berlajur banyak, lajur *kanan* untuk mendahului,
  - bila lampu merah menyala, mobil belok *kiri* boleh langsung

  Prinsip **dualitas**:
  Konsep kiri dan kanan dapat dipertukarkan pada kedua negara tersebut sehingga peraturan yang berlaku di Amerika Serikat menjadi berlaku pula di Inggris.

Dualitas

- **(Prinsip Dualitas pada Himpunan).** Misalkan S adalah suatu kesamaan (*identity*) yang melibatkan himpunan dan operasi-operasi seperti $\cup$, $\cap$, dan komplemen. Jika S^* diperoleh dari S dengan mengganti $\cup \rightarrow \cap$, $\cap \rightarrow \cup$, $\varnothing \rightarrow \mathrm{U}$, $\mathrm{U} \rightarrow \varnothing$, sedangkan komplemen dibiarkan seperti semula, maka kesamaan S^* juga benar dan disebut dual dari kesamaan S.

1. Hukum identitas: $A \cup \varnothing = A$	Dualnya: $A \cap \mathrm{U} = A$
2. Hukum *null*/dominasi: $A \cap \varnothing = \varnothing$	Dualnya: $A \cup \mathrm{U} = \mathrm{U}$
3. Hukum komplemen: $A \cup \overline{A} = \mathrm{U}$	Dualnya: $A \cap \overline{A} = \varnothing$

Dualitas

4. Hukum idempoten: $A \cup A = A$	Dualnya: $A \cap A = A$
5. Hukum penyerapan: $A \cup (A \cap B) = A$	Dualnya: $A \cap (A \cup B) = A$
6. Hukum komutatif: $A \cup B = B \cup A$	Dualnya: $A \cap B = B \cap A$
7. Hukum asosiatif: $A \cup (B \cup C) = (A \cup B) \cup C$	Dualnya: $A \cap (B \cap C) = (A \cap B) \cap C$
8. Hukum distributif: $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$	Dualnya: $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$
9. Hukum De Morgan: $\overline{A \cup B} = \overline{A} \cap \overline{B}$	Dualnya: $\overline{A \cap B} = \overline{A} \cup \overline{B}$

10. Hukum 0/1 $\overline{\varnothing} = U$	Dualnya: $\overline{U} = \varnothing$

Dualitas

Contoh 23. Dual dari $(A \cap B) \cup (A \cap \overline{B}) = A$ adalah

$$(A \cup B) \cap (A \cup \overline{B}) = A.$$

Prinsip Inklusi-Eksklusi

Untuk dua himpunan A dan B:

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

$$|A \oplus B| = |A| + |B| - 2|A \cap B|$$

Contoh 24. Berapa banyaknya bilangan bulat antara 1 dan 100 yang habis dibagi 3 atau 5?

Dualitas

Penyelesaian:

A = himpunan bilangan bulat yang habis dibagi 3,

B = himpunan bilangan bulat yang habis dibagi 5,

$A \cap B$ = himpunan bilangan bulat yang habis dibagi 3 dan 5 (yaitu himpunan bilangan bulat yang habis dibagi oleh KPK – Kelipatan Persekutuan Terkecil – dari 3 dan 5, yaitu 15),

yang ditanyakan adalah $|A \cup B|$.

$$|A| = \lfloor 100/3 \rfloor = 33,$$
$$|B| = \lfloor 100/5 \rfloor = 20,$$
$$|A \cap B| = \lfloor 100/15 \rfloor = 6$$
$$|A \cup B| = |A| + |B| - |A \cap B| = 33 + 20 - 6 = 47$$

Jadi, ada 47 buah bilangan yang habis dibagi 3 atau 5.

Untuk tiga buah himpunan A, B, dan C, berlaku

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

Dualitas

Untuk himpunan $A_1, A_2, \ldots, A_r$, berlaku:

$$|A_1 \cup A_2 \cup \ldots \cup A_r| = \sum_i |A_i| - \sum_{1 \le i < j \le r} |A_i \cap A_j| + \sum_{1 \le i < j < k \le r} |A_i \cap A_j \cap A_k| + \ldots + (-1)^{r-1} |A_1 \cap A_2 \cap \ldots \cap A_r|$$

Partisi

- Partisi dari sebuah himpunan A adalah sekumpulan himpunan bagian tidak kosong A_1, A_2, … dari A sedemikian sehingga:

  (a) $A_1 \cup A_2 \cup \ldots = A$, dan

  (b) $A_i \cap A_j = \varnothing$ untuk $i \neq j$

Contoh 25. Misalkan $A = \{1, 2, 3, 4, 5, 6, 7, 8\}$, maka $\{ \{1\}, \{2, 3, 4\}, \{7, 8\}, \{5, 6\} \}$ adalah partisi A.

Himpunan Ganda

- Himpunan yang elemennya boleh berulang (tidak harus berbeda) disebut **himpunan ganda** (*multiset*).

  Contohnya, {1, 1, 1, 2, 2, 3}, {2, 2, 2}, {2, 3, 4}, {}.

- **Multiplisitas** dari suatu elemen pada himpunan ganda adalah jumlah kemunculan elemen tersebut pada himpunan ganda. Contoh: M = { 0, 1, 1, 1, 0, 0, 0, 1 }, multiplisitas 0 adalah 4.

- Himpunan (*set*) merupakan contoh khusus dari suatu *multiset*, yang dalam hal ini multiplisitas dari setiap elemennya adalah 0 atau 1.

- Kardinalitas dari suatu *multiset* didefinisikan sebagai kardinalitas himpunan padanannya (ekivalen), dengan mengasumsikan elemen-elemen di dalam *multiset* semua berbeda.

Operasi antara Dua Multiset

Misalkan P dan Q adalah *multiset*:

1. $P \cup Q$ adalah suatu *multiset* yang multiplisitas elemennya sama dengan multiplisitas maksimum elemen tersebut pada himpunan P dan Q.
   Contoh: $P = \{ a, a, a, c, d, d \}$ dan $Q = \{ a, a, b, c, c \}$,
   $P \cup Q = \{ a, a, a, b, c, c, d, d \}$

2. $P \cap Q$ adalah suatu *multiset* yang multiplisitas elemennya sama dengan multiplisitas minimum elemen tersebut pada himpunan P dan Q.
   Contoh: $P = \{ a, a, a, c, d, d \}$ dan $Q = \{ a, a, b, c, c \}$
   $P \cap Q = \{ a, a, c \}$

3. P – Q adalah suatu *multiset* yang multiplisitas elemennya sama dengan:
   - multiplisitas elemen tersebut pada P dikurangi multiplisitasnya pada Q, jika selisihnya positif
   - 0, jika selisihnya nol atau negatif.

   Contoh: $P = \{ a, a, a, b, b, c, d, d, e \}$ dan $Q = \{ a, a, b, b, b, c, c, d, d, f \}$ maka $P - Q = \{ a, e \}$

Operasi antara Dua Multiset

4. $P + Q$, yang didefinisikan sebagai jumlah (*sum*) dua buah himpunan ganda, adalah suatu *multiset* yang multiplisitas elemennya sama dengan penjumlahan dari multiplisitas elemen tersebut pada P dan Q.

   Contoh: $P = \{ a, a, b, c, c \}$ dan $Q = \{ a, b, b, d \}$,

   $P + Q = \{ a, a, a, b, b, b, c, c, d \}$

Pembuktian

- Pernyataan himpunan adalah argumen yang menggunakan notasi himpunan.
- Pernyataan dapat berupa:

1. Kesamaan (*identity*)

   Contoh: Buktikan "$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$"

2. Implikasi

   Contoh: Buktikan bahwa "Jika $A \cap B = \varnothing$ dan $A \subseteq (B \cup C)$ maka selalu berlaku bahwa $A \subseteq C$".

Pembuktian dg Diagram Venn

Contoh 26. Misalkan A, B, dan C adalah himpunan. Buktikan $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$ dengan diagram Venn.

Bukti:

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

Kedua digaram Venn memberikan area arsiran yang sama.
Terbukti bahwa $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$.

- Diagram Venn hanya dapat digunakan jika himpunan yang digambarkan tidak banyak jumlahnya.
- Metode ini *mengilustrasikan* ketimbang membuktikan fakta. Diagram Venn tidak dianggap sebagai metode yang valid untuk pembuktian secara formal.

Pembuktian dg Tabel Keanggotaan

Contoh 27. Misalkan A, B, dan C adalah himpunan. Buktikan bahwa $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$.

Bukti:

A	B	C	$B \cup C$	$A \cap (B \cup C)$	$A \cap B$	$A \cap C$	$(A \cap B) \cup (A \cap C)$
0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	0	0	0	0
1	0	0	0	0	0	0	0
1	0	1	1	1	0	1	1
1	1	0	1	1	1	0	1
1	1	1	1	1	1	1	1

Karena kolom $A \cap (B \cup C)$ dan kolom $(A \cap B) \cup (A \cap C)$ sama, maka $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$.

Pembuktian dg Aljabar Himpunan

Contoh 28. Misalkan A dan B himpunan. Buktikan bahwa $(A \cap B) \cup (A \cap \overline{B}) = A$

Bukti:

$$
\begin{aligned}
(A \cap B) \cup (A \cap \overline{B}) &= A \cap (B \cup \overline{B}) && \text{(Hukum distributif)} \\
&= A \cap \mathrm{U} && \text{(Hukum komplemen)} \\
&= A && \text{(Hukum identitas)}
\end{aligned}
$$

Contoh 29. Misalkan A dan B himpunan. Buktikan bahwa $A \cup (B - A) = A \cup B$

Bukti:

$$
\begin{aligned}
A \cup (B - A) &= A \cup (B \cap \overline{A}) && \text{(Definisi operasi selisih)} \\
&= (A \cup B) \cap (A \cup \overline{A}) && \text{(Hukum distributif)} \\
&= (A \cup B) \cap \mathrm{U} && \text{(Hukum komplemen)} \\
&= A \cup B && \text{(Hukum identitas)}
\end{aligned}
$$

Pembuktian dg Aljabar Himpunan

Contoh 30. Buktikan bahwa untuk sembarang himpunan A dan B, bahwa

(i) $A \cup (\overline{A} \cap B) = A \cup B$ dan

(ii) $A \cap (\overline{A} \cup B) = A \cap B$

Bukti:

(i) $A \cup (\overline{A} \cap B) = (A \cup \overline{A}) \cap (A \cap B)$ (H. distributif)

$= \mathrm{U} \cap (A \cap B)$ (H. komplemen)

$= A \cup B$ (H. identitas)

(ii) adalah dual dari (i)

$A \cap (\overline{A} \cup B) = (A \cap \overline{A}) \cup (A \cap B)$ (H. distributif)

$= \varnothing \cup (A \cap B)$ (H. komplemen)

$= A \cap B$ (H. identitas)

Pembuktian dg Definisi

- Metode ini digunakan untuk membuktikan pernyataan himpunan yang tidak berbentuk kesamaan, tetapi pernyataan yang berbentuk implikasi. Biasanya di dalam implikasi tersebut terdapat notasi himpunan bagian ($\subseteq$ atau $\subset$).

Contoh 31. Misalkan A dan B himpunan. Jika $A \cap B = \varnothing$ dan $A \subseteq (B \cup C)$ maka $A \subseteq C$. Buktikan!

Bukti:

(i) Dari definisi himpunan bagian, $P \subseteq Q$ jika dan hanya jika setiap $x \in P$ juga $\in Q$. Misalkan $x \in A$. Karena $A \subseteq (B \cup C)$, maka dari definisi himpunan bagian, x juga $\in (B \cup C)$. Dari definisi operasi gabungan ($\cup$), $x \in (B \cup C)$ berarti $x \in B$ atau $x \in C$.

(ii) Karena $x \in A$ dan $A \cap B = \varnothing$, maka $x \notin B$

Dari (i) dan (ii), $x \in C$ harus benar. Karena $\forall x \in A$ juga berlaku $x \in C$, maka dapat disimpulkan $A \subseteq C$.

Materi

- **Matriks**
- **Relasi**
  Representasi Relasi
  1. ***Representasi Relasi dengan Diagram Panah***
  2. ***Representasi Relasi dengan Tabel***
  3. ***Representasi Relasi dengan Matriks***
  4. ***Representasi Relasi dengan Graf Berarah***
- **Sifat-sifat Relasi Biner**
  1. **Refleksif** (*reflexive*)
  2. **Menghantar** (*transitive*)
  3. **Setangkup** (*symmetric*) dan **tak-setangkup** (*antisymmetric*)
  4. **Relasi Inversi**
  5. **Mengkombinasikan *Relasi***
  6. **Komposisi Relasi**
  7. **Relasi *n-ary***
     - ***7.1. Seleksi***
     - ***7.2. Proyeksi***
     - ***7.3. Join***

Matriks

- Matriks adalah susunan skalar elemen-elemen dalam bentuk baris dan kolom.

- Matriks A yang berukuran dari m baris dan n kolom ($m \times n$) adalah:

$$A = \begin{bmatrix} a_{11} & a_{12} & \Lambda & a_{1n} \\ a_{21} & a_{22} & \Lambda & a_{2n} \\ \text{M} & \text{M} & & \text{M} \\ a_{m1} & a_{m2} & \Lambda & a_{mn} \end{bmatrix}$$

- Matriks bujursangkar adalah matriks yang berukuran $n \times n$.

- Dalam praktek, kita lazim menuliskan matriks dengan notasi ringkas $A = [a_{ij}]$.

Contoh 1. Di bawah ini adalah matriks yang berukuran 3×4:

$$A = \begin{bmatrix} 2 & 5 & 0 & 6 \\ 8 & 7 & 5 & 4 \\ 3 & 1 & 1 & 8 \end{bmatrix}$$

- Matriks simetri adalah matriks yang $a_{ij} = a_{ji}$ untuk setiap i dan j.

Contoh 2. Di bawah ini adalah contoh matriks simetri.

$$\begin{bmatrix} 2 & 6 & 6 & -4 \\ 6 & 3 & 7 & 3 \\ 6 & 7 & 0 & 2 \\ -4 & 3 & 2 & 8 \end{bmatrix}$$

- Matriks *zero-one* (0/1) adalah matriks yang setiap elemennya hanya bernilai 0 atau 1.

Contoh 3. Di bawah ini adalah contoh matriks 0/1:

$$\begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

Relasi

- Relasi biner R antara himpunan A dan B adalah himpunan bagian dari $A \times B$.
- Notasi: $R \subseteq (A \times B)$.
- $a\ R\ b$ adalah notasi untuk $(a, b) \in R$, yang artinya a dihubungankan dengan b oleh R
- $a \not R\ b$ adalah notasi untuk $(a, b) \notin R$, yang artinya a tidak dihubungkan dengan b oleh relasi R.
- Himpunan A disebut daerah asal (*domain*) dari R, dan himpunan B disebut daerah hasil (*range*) dari R.

Contoh 3. Misalkan

A = {Amir, Budi, Cecep}, B = {IF221, IF251, IF342, IF323}

$A \times B$ = {(Amir, IF221), (Amir, IF251), (Amir, IF342),
(Amir, IF323), (Budi, IF221), (Budi, IF251),
(Budi, IF342), (Budi, IF323), (Cecep, IF221),
(Cecep, IF251), (Cecep, IF342), (Cecep, IF323) }

Misalkan R adalah relasi yang menyatakan mata kuliah yang diambil oleh mahasiswa pada Semester Ganjil, yaitu

R = {(Amir, IF251), (Amir, IF323), (Budi, IF221),
(Budi, IF251), (Cecep, IF323) }

- Dapat dilihat bahwa $R \subseteq (A \times B)$,
- A adalah daerah asal R, dan B adalah daerah hasil R.
- (Amir, IF251) $\in R$ atau Amir R IF251
- (Amir, IF342) $\notin R$ atau Amir $\not R$ IF342.

Contoh 4. Misalkan $P = \{2, 3, 4\}$ dan $Q = \{2, 4, 8, 9, 15\}$. Jika kita definisikan relasi R dari P ke Q dengan

$$(p, q) \in R \text{ jika } p \text{ habis membagi } q$$

maka kita peroleh

maka kita peroleh

$$R = \{(2, 2), (2, 4), (2, 8), (3, 9), (3, 15), (4, 4), (4, 8), \}$$

- **Relasi pada sebuah himpunan adalah relasi yang khusus**
- **Relasi pada himpunan A adalah relasi dari $A \times A$.**
- **Relasi pada himpunan A adalah himpunan bagian dari $A \times A$.**

Contoh 5. Misalkan R adalah relasi pada $A = \{2, 3, 4, 8, 9\}$ yang didefinisikan oleh $(x, y) \in R$ jika x adalah faktor prima dari y. Maka

$$R = \{(2, 2), (2, 4), (2, 8), (3, 3), (3, 9)\}$$

Representasi Relasi

1. Representasi Relasi dengan Diagram Panah

2. Representasi Relasi dengan Tabel

- Kolom pertama tabel menyatakan daerah asal, sedangkan kolom kedua menyatakan daerah hasil.

Tabel 1

A	*B*
Amir	IF251
Amir	IF323
Budi	IF221
Budi	IF251
Cecep	IF323

Tabel 2

P	*Q*
2	2
2	4
2	8
3	9
3	15
4	4
4	8

Tabel 3

A	*A*
2	2
2	4
2	8
3	3
3	9

3. Representasi Relasi dengan Matriks

- Misalkan R adalah relasi dari $A = \{a_1, a_2, \ldots, a_m\}$ dan $B = \{b_1, b_2, \ldots, b_n\}$.
- Relasi R dapat disajikan dengan matriks $M = [m_{ij}]$,

$$M = \begin{array}{c} \\ a_1 \\ a_2 \\ \vdots \\ a_m \end{array} \begin{array}{c} \begin{array}{cccc} b_1 & b_2 & \ldots & b_n \end{array} \\ \begin{bmatrix} m_{11} & m_{12} & \Lambda & m_{1n} \\ m_{21} & m_{22} & \Lambda & m_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ m_{m1} & m_{m2} & \Lambda & m_{mn} \end{bmatrix} \end{array}$$

yang dalam hal ini

$$m_{ij} = \begin{cases} 1, & (a_i, b_j) \in R \\ 0, & (a_i, b_j) \notin R \end{cases}$$

Contoh 6. Relasi R pada Contoh 3 dapat dinyatakan dengan matriks

$$\begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

dalam hal ini, a_1 = Amir, a_2 = Budi, a_3 = Cecep, dan b_1 = IF221, b_2 = IF251, b_3 = IF342, dan b_4 = IF323.

Relasi R pada Contoh 4 dapat dinyatakan dengan matriks

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

yang dalam hal ini, $a_1 = 2$, $a_2 = 3$, $a_3 = 4$, dan $b_1 = 2$, $b_2 = 4$, $b_3 = 8$, $b_4 = 9$, $b_5 = 15$.

4. *Representasi Relasi dengan Graf Berarah* ***(Senin)***

- Relasi pada sebuah himpunan dapat direpresentasikan secara grafis dengan **graf berarah** (*directed graph* atau *digraph*)
- Graf berarah tidak didefinisikan untuk merepresentasikan relasi dari suatu himpunan ke himpunan lain.
- Tiap elemen himpunan dinyatakan dengan sebuah titik (disebut juga simpul atau *vertex*), dan tiap pasangan terurut dinyatakan dengan busur (*arc*)
- Jika $(a, b) \in R$, maka sebuah busur dibuat dari simpul a ke simpul b. Simpul a disebut **simpul asal** (*initial vertex*) dan simpul b disebut **simpul tujuan** (*terminal vertex*).
- Pasangan terurut (a, a) dinyatakan dengan busur dari simpul a ke simpul a sendiri. Busur semacam itu disebut **gelang** atau **kalang** (*loop*).

Contoh 7. Misalkan $R = \{(a, a), (a, b), (b, a), (b, c), (b, d), (c, \text{a}), (c, d), (d, b)\}$ adalah relasi pada himpunan $\{a, b, c, d\}$.

R direpresentasikan dengan graf berarah sbb:

Sifat-sifat Relasi Biner

- Relasi biner yang didefinisikan pada sebuah himpunan mempunyai beberapa sifat.

1. **Refleksif** (*reflexive*)

- Relasi R pada himpunan A disebut **refleksif** jika $(a, a) \in R$ untuk setiap $a \in A$.

- Relasi R pada himpunan A tidak refleksif jika ada $a \in A$ sedemikian sehingga $(a, a) \notin R$.

Contoh 8. Misalkan $A = \{1, 2, 3, 4\}$, dan relasi R di bawah ini didefinisikan pada himpunan A, maka

(a) Relasi $R = \{(1, 1), (1, 3), (2, 1), (2, 2), (3, 3), (4, 2), (4, 3), (4, 4)\}$ bersifat refleksif karena terdapat elemen relasi yang berbentuk (a, a), yaitu (1, 1), (2, 2), (3, 3), dan (4, 4).

(b) Relasi $R = \{(1, 1), (2, 2), (2, 3), (4, 2), (4, 3), (4, 4)\}$ tidak bersifat refleksif karena $(3, 3) \notin R$.

Contoh 9. Relasi "habis membagi" pada himpunan bilangan bulat positif bersifat refleksif karena setiap bilangan bulat positif habis dibagi dengan dirinya sendiri, sehingga $(a, a) \in R$ untuk setiap $a \in A$.

Contoh 10. Tiga buah relasi di bawah ini menyatakan relasi pada himpunan bilangan bulat positif **N**.

R : x lebih besar dari y, $\quad S : x + y = 5, \quad T : 3x + y = 10$

Tidak satupun dari ketiga relasi di atas yang refleksif karena, misalkan (2, 2) bukan anggota R, S, maupun T.

■

- Relasi yang bersifat refleksif mempunyai matriks yang elemen diagonal utamanya semua bernilai 1, atau $m_{ii} = 1$, untuk $i = 1, 2, \ldots, n$,

$$\begin{bmatrix} 1 & & & & & \\ & 1 & & & & \\ & & 1 & & & \\ & & & 1 & & \\ & & & & 1 & \\ & & & & & 1 \end{bmatrix}$$

- Graf berarah dari relasi yang bersifat refleksif dicirikan adanya gelang pada setiap simpulnya.

2. **Menghantar** (*transitive*)

- Relasi R pada himpunan A disebut **menghantar** jika $(a, b) \in R$ dan $(b, c) \in R$, maka $(a, c) \in R$, untuk $a, b, c \in A$.
- Relasi R pada himpunan A disebut **menghantar** jika tidak ada $(a, b) \in R$ dan $(b, c) \in R$, sedemikian hingga $(a, c) \in R$, untuk $a, b, c \in A$.

Contoh 11. Misalkan $A = \{1, 2, 3, 4\}$, dan relasi R di bawah ini didefinisikan pada himpunan A, maka

(a) $R = \{(2, 1), (3, 1), (3, 2), (4, 1), (4, 2), (4, 3)\}$ bersifat menghantar. Lihat tabel berikut:

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

(b) $R = \{(1, 1), (2, 3), (2, 4), (4, 2)\}$ tidak manghantar karena $(2, 4)$ dan $(4, 2) \in R$, tetapi $(2, 2) \notin R$, begitu juga $(4, 2)$ dan $(2, 3) \in R$, tetapi $(4, 3) \notin R$.

(c) Relasi $R = \{(1, 1), (2, 2), (3, 3), (4, 4)\}$ jelas menghantar

(d) Relasi $R = \{(1, 2), (3, 4)\}$ menghantar karena tidak ada $(a, b) \in R$ dan $(b, c) \in R$ sedemikian sehingga $(a, c) \in R$.

Relasi yang hanya berisi satu elemen seperti $R = \{(4, 5)\}$ selalu menghantar.

Contoh 12. Relasi "habis membagi" pada himpunan bilangan bulat positif bersifat menghantar. Misalkan bahwa a habis membagi b dan b habis membagi c. Maka terdapat bilangan positif m dan n sedemikian sehingga $b = ma$ dan $c = nb$. Di sini $c = nma$, sehingga a habis membagi c. Jadi, relasi "habis membagi" bersifat menghantar.

Contoh 13. Tiga buah relasi di bawah ini menyatakan relasi pada himpunan bilangan bulat positif **N**.

R : x lebih besar dari y, $S : x + y = 6$, $T : 3x + y = 10$

- R adalah relasi menghantar karena jika $x > y$ dan $y > z$ maka $x > z$.
- S tidak menghantar karena, misalkan (4, 2) dan (2, 4) adalah anggota S tetapi $(4, 4) \notin S$.
- $T = \{(1, 7), (2, 4), (3, 1)\}$ menghantar.

- Relasi yang bersifat menghantar tidak mempunyai ciri khusus pada matriks representasinya

- Sifat menghantar pada graf berarah ditunjukkan oleh: jika ada busur dari *a* ke *b* dan dari *b* ke *c*, maka juga terdapat busur berarah dari *a* ke *c*.

3. Setangkup (*symmetric*) dan **tak-setangkup** (*antisymmetric*)

- Relasi R pada himpunan A disebut **setangkup** jika $(a, b) \in R$, maka $(b, a) \in R$ untuk $a, b \in A$.

- Relasi R pada himpunan A **tidak setangkup** jika $(a, b) \in R$ sedemikian sehingga $(b, a) \notin R$.

- Relasi R pada himpunan A sedemikian sehingga $(a, b) \in R$ dan $(b, a) \in R$ hanya jika $a = b$ untuk $a, b \in A$ disebut **tolak-setangkup**.

- Relasi R pada himpunan A **tidak tolak-setangkup** jika ada elemen berbeda a dan b sedemikian sehingga $(a, b) \in R$ dan $(b, a) \in R$, untuk a, b $\in$ A

Contoh 14. Misalkan $A = \{1, 2, 3, 4\}$, dan relasi R di bawah ini didefinisikan pada himpunan A, maka

(a) Relasi $R = \{(1, 1), (1, 2), (2, 1), (2, 2), (2, 4), (4, 2), (4, 4)\}$ bersifat **setangkup** karena jika $(a, b) \in R$ maka (b, a) juga $\in R$. Di sini $(1, 2)$ dan $(2, 1) \in R$, begitu juga $(2, 4)$ dan $(4, 2) \in R$.

(b) Relasi $R = \{(1, 1), (2, 3), (2, 4), (4, 2)\}$ **tidak setangkup** karena $(2, 3) \in R$, tetapi $(3, 2) \notin R$.

(c) Relasi $R = \{(1, 1), (2, 2), (3, 3)\}$ **tolak-setangkup** karena $1 = 1$ dan $(1, 1) \in R$, $2 = 2$ dan $(2, 2) \in R$, dan $3 = 3$ dan $(3, 3) \in R$. Perhatikan bahwa R juga setangkup.

(d) Relasi $R = \{(1, 1), (1, 2), (2, 2), (2, 3)\}$ **tolak-setangkup** karena $(1, 1) \in R$ dan $1 = 1$ dan, $(2, 2) \in R$ dan $2 = 2$ dan. Perhatikan bahwa R tidak setangkup.

(e) Relasi $R = \{(1, 1), (2, 4), (3, 3), (4, 2)\}$ **tidak tolak-setangkup** karena $2 \neq 4$ tetapi $(2, 4)$ dan $(4, 2)$ anggota R. Relasi R pada (a) dan (b) di atas juga setangkup.

(f) Relasi $R = \{(1, 2), (2, 3), (1, 3)\}$ tidak setangkup.
Relasi $R = \{(1, 1), (2, 2), (2, 3), (3, 2), (4, 2), (4, 4)\}$ tidak setangkup dan tidak tolak-setangkup. R tidak setangkup karena $(4, 2) \in R$ tetapi $(2, 4) \notin R$. R tidak tolak-setangkup karena $(2, 3) \in R$ dan $(3, 2) \in R$ tetap $2 \neq 3$.

Contoh 15. Relasi "habis membagi" pada himpunan bilangan bulat positif tidak setangkup karena jika a habis membagi b, b tidak habis membagi a, kecuali jika $a = b$. Sebagai contoh, 2 habis membagi 4, tetapi 4 tidak habis membagi 2. Karena itu, $(2, 4) \in R$ tetapi $(4, 2) \notin R$. Relasi "habis membagi" tolak-setangkup karena jika a habis membagi b dan b habis membagi a maka $a = b$. Sebagai contoh, 4 habis membagi 4. Karena itu, $(4, 4) \in R$ dan $4 = 4$.

Contoh 16. Tiga buah relasi di bawah ini menyatakan relasi pada himpunan bilangan bulat positif **N**.

R : x lebih besar dari y, $\quad S : x + y = 6$, $\quad T : 3x + y = 10$

- R bukan relasi setangkup karena, misalkan 5 lebih besar dari 3 tetapi 3 tidak lebih besar dari 5.
- S relasi setangkup karena (4, 2) dan (2, 4) adalah anggota S.
- T tidak setangkup karena, misalkan (3, 1) adalah anggota T tetapi (1, 3) bukan anggota T.
- S bukan relasi tolak-setangkup karena, misalkan $(4, 2) \in S$ dan $(4, 2) \in S$ tetapi $4 \neq 2$.
- Relasi R dan T keduanya tolak-setangkup (tunjukkan!).

- Relasi yang bersifat setangkup mempunyai matriks yang elemen-elemen di bawah diagonal utama merupakan pencerminan dari elemen-elemen di atas diagonal utama, atau $m_{ij} = m_{ji} = 1$, untuk $i = 1, 2, \ldots, n$:

$$\begin{bmatrix} & 1 & \\ & & 0 \\ 1 & & \\ & & \\ 0 & & \end{bmatrix}$$

- Sedangkan graf berarah dari relasi yang bersifat setangkup dicirikan oleh: jika ada busur dari *a* ke *b*, maka juga ada busur dari *b* ke *a*

- Matriks dari relasi tolak-setangkup mempunyai sifat yaitu jika $m_{ij} = 1$ dengan $i \neq j$, maka $m_{ji} = 0$. Dengan kata lain, matriks dari relasi tolak-setangkup adalah jika salah satu dari $m_{ij} = 0$ atau $m_{ji} = 0$ bila $i \neq j$:

$$\begin{bmatrix} & & 1 & \\ & & & 0 \\ 0 & & & 1 \\ & 1 & & \\ & & 0 & \end{bmatrix}$$

- Sedangkan graf berarah dari relasi yang bersifat tolak-setangkup dicirikan oleh: jika dan hanya jika tidak pernah ada dua busur dalam arah berlawanan antara dua simpul berbeda.

4. Relasi Inversi →

- Misalkan R adalah relasi dari himpunan A ke himpunan B. Invers dari relasi R, dilambangkan dengan R^{-1}, adalah relasi dari B ke A yang didefinisikan oleh

$$R^{-1} = \{(b, a) \mid (a, b) \in R\}$$

Contoh 17. Misalkan $P = \{2, 3, 4\}$ dan $Q = \{2, 4, 8, 9, 15\}$. Jika kita definisikan relasi R dari P ke Q dengan

$(p, q) \in R$ jika p habis membagi q

maka kita peroleh

$R = \{(2, 2), (2, 4), (2, 8), (3, 9), (3, 15), (4, 4), (4, 8)\}$

R^{-1} adalah *invers* dari relasi R, yaitu relasi dari Q ke P dengan

$(q, p) \in R^{-1}$ jika q adalah kelipatan dari p

maka kita peroleh

Jika M adalah matriks yang merepresentasikan relasi R,

$$M = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

maka matriks yang merepresentasikan relasi R^{-1}, misalkan N, diperoleh dengan melakukan *transpose* terhadap matriks M,

$$N = M^T = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

5. Mengkombinasikan Relasi

- Karena relasi biner merupakan himpunan pasangan terurut, maka operasi himpunan seperti irisan, gabungan, selisih, dan beda setangkup antara dua relasi atau lebih juga berlaku.

- Jika R_1 dan R_2 masing-masing adalah relasi dari himpuna A ke himpunan B, maka $R_1 \cap R_2$, $R_1 \cup R_2$, $R_1 - R_2$, dan $R_1 \oplus R_2$ [$A \oplus B = (A \cup B) - (A \cap B) = (A - B) \cup (B - A)$] juga adalah relasi dari A ke B.

Contoh 18.

Misalkan $A = \{a, b, c\}$ dan $B = \{a, b, c, d\}$.

Relasi $R_1 = \{(a, a), (b, b), (c, c)\}$
Relasi $R_2 = \{(a, a), (a, b), (a, c), (a, d)\}$
$R_1 \cap R_2, R_1 \cup R_2, R_1 - R_2, R_2 - R_1, R_1 \oplus R_2$

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

- Jika relasi R_1 dan R_2 masing-masing dinyatakan dengan matriks M_{R1} dan M_{R2}, maka matriks yang menyatakan gabungan dan irisan dari kedua relasi tersebut adalah

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

Contoh 19. Misalkan bahwa relasi R_1 dan R_2 pada himpunan A dinyatakan oleh matriks

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

maka

$$M_{R1 \cup R2} = M_{R1} \vee M_{R2} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

$$M_{R1 \cap R2} = M_{R1} \wedge M_{R2} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

6. Komposisi Relasi

- Misalkan R adalah relasi dari himpunan A ke himpunan B, dan S adalah relasi dari himpunan B ke himpunan C. Komposisi R dan S, dinotasikan dengan $S \circ R$, adalah relasi dari A ke C yang didefinisikan oleh

$$S \circ R = \{(a, c) \mid a \in A, c \in C, \text{ dan untuk beberapa } b \in B, (a, b) \in R \text{ dan } (b, c) \in S\}$$

Contoh 20. Misalkan

$$R = \{(1, 2), (1, 6), (2, 4), (3, 4), (3, 6), (3, 8)\}$$

adalah relasi dari himpunan $\{1, 2, 3\}$ ke himpunan $\{2, 4, 6, 8\}$ dan

$$S = \{(2, u), (4, s), (4, t), (6, t), (8, u)\}$$

adalah relasi dari himpunan $\{2, 4, 6, 8\}$ ke himpunan $\{s, t, u\}$.

Maka komposisi relasi R dan S adalah

Maka komposisi relasi R dan S adalah

$$S \circ R = \{(1, u), (1, t), (2, s), (2, t), (3, s), (3, t), (3, u)\}$$

Komposisi relasi R dan S lebih jelas jika diperagakan dengan diagram panah:

- Jika relasi R_1 dan R_2 masing-masing dinyatakan dengan matriks M_{R1} dan M_{R2}, maka matriks yang menyatakan komposisi dari kedua relasi tersebut adalah

$$M_{R2 \circ R1} = M_{R1} \cdot M_{R2}$$

yang dalam hal ini operator "." sama seperti pada perkalian matriks biasa, tetapi dengan mengganti tanda kali dengan "$\wedge$" dan tanda tambah dengan "$\vee$".

Contoh 21. Misalkan bahwa relasi R_1 dan R_2 pada himpunan A dinyatakan oleh matriks

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

maka matriks yang menyatakan $R_2 \circ R_1$ adalah

$$M_{R2 \circ R1} = M_{R1} \,.\, M_{R2}$$

Contoh 21. Misalkan bahwa relasi R_1 dan R_2 pada himpunan A dinyatakan oleh matriks

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

maka matriks yang menyatakan $R_2 \circ R_1$ adalah

$$M_{R2 \circ R1} = M_{R1} \,.\, M_{R2}$$

$$= \begin{bmatrix} (1 \wedge 0) \vee (0 \wedge 0) \vee (1 \wedge 1) & (1 \wedge 1) \vee (0 \wedge 0) \vee (1 \wedge 0) & (1 \wedge 0) \vee (0 \wedge 1) \vee (1 \wedge 1) \\ (1 \wedge 0) \vee (1 \wedge 0) \vee (0 \wedge 1) & (1 \wedge 1) \vee (1 \wedge 0) \vee (0 \wedge 0) & (1 \wedge 0) \vee (1 \wedge 1) \vee (0 \wedge 1) \\ (0 \wedge 0) \vee (0 \wedge 0) \vee (0 \wedge 1) & (0 \wedge 1) \vee (0 \wedge 0) \vee (0 \wedge 0) & (0 \wedge 0) \vee (0 \wedge 1) \vee (0 \wedge 1) \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

7. Relasi *n-ary*

- Relasi biner hanya menghubungkan antara dua buah himpunan.
- Relasi yang lebih umum menghubungkan lebih dari dua buah himpunan. Relasi tersebut dinamakan relasi *n-ary* (baca: ener).
- Jika $n = 2$, maka relasinya dinamakan relasi biner (bi = 2). Relasi *n-ary* mempunyai terapan penting di dalam basisdata.

- Misalkan $A_1, A_2, \ldots, A_n$ adalah himpunan. Relasi *n-ary* R pada himpunan-himpunan tersebut adalah himpunan bagian dari $A_1 \times A_2 \times \ldots \times A_n$, atau dengan notasi $R \subseteq A_1 \times A_2 \times \ldots \times A_n$. Himpunan $A_1, A_2, \ldots, A_n$ disebut daerah asal relasi dan n disebut **derajat**.

Contoh 22. Misalkan

NIM = {13598011, 13598014, 13598015, 13598019, 13598021, 13598025}
Nama = {Amir, Santi, Irwan, Ahmad, Cecep, Hamdan}
MatKul = {Logika Informatika, Algoritma, Struktur Data, Arsitektur Komputer}
Nilai = {A, B, C, D, E}

Relasi *MHS* terdiri dari 4-tupel (*NIM*, *Nama*, *MatKul*, *Nilai*):

$$MHS \subseteq NIM \times Nama \times MatKul \times Nilai$$

Satu contoh relasi yang bernama *MHS* adalah

MHS = {(13598011, Amir, Logika Informatika, A),
(13598011, Amir, Arsitektur Komputer, B),
(13598014, Santi, Arsitektur Komputer, D),
(13598015, Irwan, Algoritma, C),
(13598015, Irwan, Struktur Data C),
(13598015, Irwan, Arsitektur Komputer, B),
(13598019, Ahmad, Algoritma, E),
(13598021, Cecep, Algoritma, A),
(13598021, Cecep, Arsitektur Komputer, B),
(13598025, Hamdan, Logika Informatika, B),
(13598025, Hamdan, Algoritma, A),
(13598025, Hamdan, Struktur Data, C),
(13598025, Hamdan, Ars. Komputer, B) }

Relasi *MHS* di atas juga dapat ditulis dalam bentuk Tabel:

NIM	Nama	MatKul	Nilai
13598011	Amir	Logika Informatika	A
13598011	Amir	Arsitektur Komputer	B
13598014	Santi	Algoritma	D
13598015	Irwan	Algoritma	C
13598015	Irwan	Struktur Data	C
13598015	Irwan	Arsitektur Komputer	B
13598019	Ahmad	Algoritma	E
13598021	Cecep	Algoritma	B
13598021	Cecep	Arsitektur Komputer	B
13598025	Hamdan	Logika Informatika	B
13598025	Hamdan	Algoritma	A
13598025	Hamdan	Struktur Data	C
13598025	Hamdan	Arsitektur Komputer	B

- Basisdata (*database*) adalah kumpulan tabel.

- Salah satu model basisdata adalah **model basisdata relasional** (*relational database*). Model basisdata ini didasarkan pada konsep relasi *n-ary*.

- Pada basisdata relasional, satu tabel menyatakan **satu relasi**. Setiap kolom pada tabel disebut **atribut**. Daerah asal dari atribut adalah himpunan tempat semua anggota atribut tersebut berada.

- Setiap tabel pada basisdata diimplementasikan secara fisik sebagai sebuah ***file***.

- Satu baris data pada tabel menyatakan sebuah ***record***, dan setiap atribut menyatakan sebuah *field*.

- Secara fisik basisdata adalah kumpulan *file*, sedangkan *file* adalah kumpulan *record*, setiap *record* terdiri atas sejumlah *field*.

- Atribut khusus pada tabel yang mengidentifikasikan secara unik elemen relasi disebut **kunci** (*key*).

- Operasi yang dilakukan terhadap basisdata dilakukan dengan perintah pertanyaan yang disebut *query*.

- Contoh *query*:
  "tampilkan semua mahasiswa yang mengambil mata kuliah Matematika Diskrit"
  "tampilkan daftar nilai mahasiswa dengan NIM = 13598015"
  "tampilkan daftar mahasiswa yang terdiri atas NIM dan mata kuliah yang diambil"

- *Query* terhadap basisdata relasional dapat dinyatakan secara abstrak dengan operasi pada relasi *n-ary*.

- Ada beberapa operasi yang dapat digunakan, diantaranya adalah seleksi, proyeksi, dan join.

7.1. Seleksi

Operasi seleksi memilih baris tertentu dari suatu tabel yang memenuhi persyaratan tertentu.

Operator: σ

Contoh 23. Misalkan untuk relasi MHS kita ingin menampilkan daftar mahasiswa yang mengambil mata kuliah Logika Informatrika. Operasi seleksinya adalah

$$\sigma_{\text{Matkul="Logika Informatika"}} (\text{MHS})$$

Hasil: (13598011, Amir, Logika Informatika, A) dan
(13598025, Hamdan, Logika Informatika, B)

7.2. Proyeksi

Operasi proyeksi memilih kolom tertentu dari suatu tabel. Jika ada beberapa baris yang sama nilainya, maka hanya diambil satu kali.
Operator: π

Contoh 24. Operasi proyeksi

$$\pi_{\text{Nama, MatKul, Nilai}} (\text{MHS})$$

menghasilkan Tabel 3.5. Sedangkan operasi proyeksi

$$\pi_{\text{NIM, Nama}} (\text{MHS})$$

menghasilkan Tabel 3.6.

Tabel 3.5

Nama	MatKul	Nilai
Amir	Logika Informatika	A
Amir	Arsitektur Komputer	B
Santi	Algoritma	D
Irwan	Algoritma	C
Irwan	Struktur Data	C
Irwan	Arsitektur Komputer	B
Ahmad	Algoritma	E
Cecep	Algoritma	B
Cecep	Arsitektur Komputer	B
Hamdan	Logika Informatika	B
Hamdan	Algoritma	A
Hamdan	Struktur Data	C
Hamdan	Arsitektur Komputer	B

Tabel 3.6

NIM	Nama
13598011	Amir
13598014	Santi
13598015	Irwan
13598019	Ahmad
13598021	Cecep
13598025	Hamdan

7.3. Join

Operasi *join* menggabungkan dua buah tabel menjadi satu bila kedua tabel mempunyai atribut yang sama.
Operator: τ

Contoh 25. Misalkan relasi *MHS1* dinyatakan dengan Tabel 3.7 dan relasi *MHS2* dinyatakan dengan Tabel 3.8.

Operasi *join*

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

menghasilkan Tabel 3.9.

Tabel 3.7

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

Tabel 3.8

NIM	Nama	MatKul	Nilai
13598001	Hananto	Algoritma	A
13598001	Hananto	Basisdata	B
13598004	Heidi	Kalkulus I	B
13598006	Harman	Teori Bahasa	C
13598006	Harman	Agama	A
13598009	Junaidi	Statisitik	B
13598010	Farizka	Otomata	C

Tabel 3.9

NIM	Nama	JK	MatKul	Nilai
13598001	Hananto	L	Algoritma	A
13598001	Hananto	L	Basisdata	B
13598004	Heidi	W	Kalkulus I	B
13598006	Harman	L	Teori Bahasa	C
13598006	Harman	L	Agama	A

Tugas 4 (presentasi) →SELASA

A. Buatlah laporan di kumpul saat presentasi per masing-masing kelompok, isi laporan: Pendahuluan s.d. Kesimpulan + daftar pustaka.

B. Hasilnya dipresentasikan pada minggu ke XIII (topik 1,2, dan 3) dan XIV (topik 4,5, dan 6), dengan topik bahasan di bawah ini:

1. Algoritma (1 kelompok)→Kelompok Boby
2. Logika Predikatif (2 kelompok)→Kelompok Rosid, Riska
3. Inferensi (2 kelompok)→Kelompok Ikal
4. Deduksi dan derifasi (2 kelompok)→Kelompok Viky, Bagus
5. Strategi Pembalikan (1 kelompok)→Kelompok Fajar
6. Tablo Semantik (1 kelompok)→Kelompok Wiwa

Materi Ujian

1. Materi yang saya berikan setelah UTS
2. Strategi Pembalikan
3. Inferensi

Boby

Algoritma

1. Bobby FC(151051034) (-++)
2. Muh. S. Masnuh (151051046) (-++)
3. Ady A. (151051008) (--+)
4. Zubelina (151051017) (--+)
5. Martin (151051022) (--+)
6. Adithya P.P (151051010) (--+)

Riska

Logika Predikatif

1. Ryzka (151051020) ()
2. Ryzka (151051020)

Rosid

Logika Predikatif

1. Muh. Rosyif (151051001) ()

Bagus

Deduksi dan derifatif

1. Mh. Ardi S. (151051044)

Ikal

Inferensi

1. Ikal M. (151051039)

Wiwa

Tablo Semantik

1. Cholifah (151051054)

Viky

Deduksi dan Derivasi

1.

Fajar

Trategi Pembalikan

1. Fajar H (151051056) (--+)
2. Raditya DK (151051031) (-++)
3. Rahmat KN(151051038) (-++)
4. Mustofa WD (151051035) (nggak masuk)

Tugas 4 (presentasi) →KAMIS

A. Buatlah laporan di kumpul saat presentasi per masing-masing kelompok, isi laporan: Pendahuluan s.d. Kesimpulan + daftar pustaka.

B. Hasilnya dipresentasikan pada minggu ke XIII (topik 1,2, dan 3) dan XIV (topik 4,5, dan 6), dengan topik bahasan di bawah ini:

1. Algoritma (1 kelompok)→Kelompok Licha→XIII
2. Logika Predikatif (1 kelompok)→Kelompok Asih→XIII
3. Inferensi (1 kelompok)→Kelompok Luay→XIII
4. Deduksi dan derifasi (1 kelompok)→Kelompok Andika→XIV
5. Strategi Pembalikan (1 kelompok)→Kelompok Rasid→XIV
6. Tablo Semantik (1 kelompok)→Kelompok Yuda→XIV

Licha (XIII)

Asih (XIII)

Luay (XIII)

Logika Informatika (Senin)

- Fungsi (Arditya dan Wulan)
- Rekursi (Franko dan Abdul)
- Strategi Pembalikan (Diky dan Daniel)
- Tablo Semantik (Rizal dan Widi)

Fungsi

- Misalkan A dan B himpunan.
  Relasi biner f dari A ke B merupakan suatu fungsi jika *setiap* elemen di dalam A dihubungkan dengan tepat satu elemen di dalam B.

  Jika f adalah fungsi dari A ke B kita menuliskan

  $$f : A \rightarrow B$$

  yang artinya f **memetakan** A ke B.

- A disebut **daerah asal** (*domain*) dari f dan B disebut **daerah hasil** (*codomain*) dari f.

- Nama lain untuk fungsi adalah **pemetaan** atau **transformasi**.

- Kita menuliskan $f(a) = b$ jika elemen a di dalam A dihubungkan dengan elemen b di dalam B.

- Jika $f(a) = b$, maka b dinamakan **bayangan** (*image*) dari a dan a dinamakan **pra-bayangan** (*pre-image*) dari b.

- Himpunan yang berisi semua nilai pemetaan f disebut **jelajah** (*range*) dari f. Perhatikan bahwa jelajah dari f adalah himpunan bagian (mungkin *proper subset*) dari B.

- Fungsi adalah relasi yang khusus:
  1. Tiap elemen di dalam himpunan A harus digunakan oleh prosedur atau kaidah yang mendefinisikan f.

  2. Frasa "dihubungkan dengan tepat satu elemen di dalam B" berarti bahwa jika $(a, b) \in f$ dan $(a, c) \in f$, maka $b = c$.

- Fungsi dapat dispesifikasikan dalam berbagai bentuk, diantaranya:
  1. Himpunan pasangan terurut.
     Seperti pada relasi.
  2. Formula pengisian nilai (*assignment*).
     Contoh: $f(x) = 2x + 10$, $f(x) = x^2$, dan $f(x) = 1/x$.
  3. Kata-kata
     Contoh: "*f* adalah fungsi yang memetakan jumlah bit 1 di dalam suatu *string* biner".
  4. Kode program (*source code*)
     Contoh: Fungsi menghitung $|x|$

```
function abs(x:integer):integer;
begin
   if x < 0 then
      abs:=-x
   else
      abs:=x;
end;
```

Contoh 26. Relasi

$$f = \{(1, u), (2, v), (3, w)\}$$

dari $A = \{1, 2, 3\}$ ke $B = \{u, v, w\}$ adalah fungsi dari A ke B. Di sini $f(1) = u$, $f(2) = v$, dan $f(3) = w$. Daerah asal dari f adalah A dan daerah hasil adalah B. Jelajah dari f adalah $\{u, v, w\}$, yang dalam hal ini sama dengan himpunan B.

Contoh 27. Relasi

$$f = \{(1, u), (2, u), (3, v)\}$$

dari $A = \{1, 2, 3\}$ ke $B = \{u, v, w\}$ adalah fungsi dari A ke B, meskipun u merupakan bayangan dari dua elemen A. Daerah asal fungsi adalah A, daerah hasilnya adalah B, dan jelajah fungsi adalah $\{u, v\}$.

Contoh 28. Relasi

$$f = \{(1, u), (2, v), (3, w)\}$$

dari $A = \{1, 2, 3, 4\}$ ke $B = \{u, v, w\}$ bukan fungsi, karena tidak semua elemen A dipetakan ke B.

Contoh 29. Relasi

$$f = \{(1, u), (1, v), (2, v), (3, w)\}$$

dari $A = \{1, 2, 3\}$ ke $B = \{u, v, w\}$ bukan fungsi, karena 1 dipetakan ke dua buah elemen B, yaitu u dan v.

Contoh 30. Misalkan $f : \mathbf{Z} \rightarrow \mathbf{Z}$ didefinisikan oleh $f(x) = x^2$. Daerah asal dan daerah hasil dari f adalah himpunan bilangan bulat, dan jelajah dari f adalah himpunan bilangan bulat tidak-negatif.

- Fungsi f dikatakan **satu-ke-satu** (*one-to-one*) atau **injektif** (*injective*) jika tidak ada dua elemen himpunan A yang memiliki bayangan sama.

Contoh 31. Relasi

$$f = \{(1, w), (2, u), (3, v)\}$$

dari $A = \{1, 2, 3\}$ ke $B = \{u, v, w, x\}$ adalah fungsi satu-ke-satu,

Tetapi relasi

$$f = \{(1, u), (2, u), (3, v)\}$$

dari $A = \{1, 2, 3\}$ ke $B = \{u, v, w\}$ bukan fungsi satu-ke-satu, karena $f(1) = f(2) = u$.

Contoh 32. Misalkan $f : \mathbf{Z} \rightarrow \mathbf{Z}$. Tentukan apakah $f(x) = x^2 + 1$ dan $f(x) = x - 1$ merupakan fungsi satu-ke-satu?

<u>Penyelesaian</u>:

(i) $f(x) = x^2 + 1$ bukan fungsi satu-ke-satu, karena untuk dua x yang bernilai mutlak sama tetapi tandanya berbeda nilai fungsinya sama, misalnya $f(2) = f(-2) = 5$ padahal $-2 \neq 2$.

(ii) $f(x) = x - 1$ adalah fungsi satu-ke-satu karena untuk $a \neq b$,

$a - 1 \neq b - 1$.

Misalnya untuk $x = 2$, $f(2) = 1$ dan untuk $x = -2$, $f(-2) = -3$.

- Fungsi f dikatakan dipetakan **pada** (*onto*) atau **surjektif** (*surjective*) jika setiap elemen himpunan B merupakan bayangan dari satu atau lebih elemen himpunan A.

- Dengan kata lain seluruh elemen B merupakan jelajah dari f. Fungsi f disebut fungsi pada himpunan B.

Contoh 33. Relasi

$$f = \{(1, u), (2, u), (3, v)\}$$

dari $A = \{1, 2, 3\}$ ke $B = \{u, v, w\}$ bukan fungsi **pada** karena w tidak termasuk jelajah dari f.

Relasi

$$f = \{(1, w), (2, u), (3, v)\}$$

dari $A = \{1, 2, 3\}$ ke $B = \{u, v, w\}$ merupakan fungsi pada karena semua anggota B merupakan jelajah dari f.

Contoh 34. Misalkan $f : \mathbf{Z} \rightarrow \mathbf{Z}$. Tentukan apakah $f(x) = x^2 + 1$ dan $f(x) = x - 1$ merupakan fungsi pada?

<u>Penyelesaian</u>:

(i) $f(x) = x^2 + 1$ bukan fungsi pada, karena tidak semua nilai bilangan bulat merupakan jelajah dari f.

(ii) $f(x) = x - 1$ adalah fungsi pada karena untuk setiap bilangan bulat y, selalu ada nilai x yang memenuhi, yaitu $y = x - 1$ akan dipenuhi untuk $x = y + 1$.

- Fungsi f dikatakan **berkoresponden satu-ke-satu** atau **bijeksi** (*bijection*) jika ia fungsi satu-ke-satu dan juga fungsi pada.

Contoh 35. Relasi

$$f = \{(1, u), (2, w), (3, v)\}$$

dari $A = \{1, 2, 3\}$ ke $B = \{u, v, w\}$ adalah fungsi yang berkoresponden satu-ke-satu, karena f adalah fungsi satu-ke-satu maupun fungsi pada.

Contoh 36. Fungsi $f(x) = x - 1$ merupakan fungsi yang berkoresponden satu-ke-satu, karena f adalah fungsi satu-ke-satu maupun fungsi pada.

Fungsi satu-ke-satu,
bukan pada

Fungsi pada,
bukan satu-ke-satu

Buka fungsi satu-ke-satu
maupun pada

Bukan fungsi

- Jika f adalah fungsi berkoresponden satu-ke-satu dari A ke B, maka kita dapat menemukan **balikan** (*invers*) dari f.

- Balikan fungsi dilambangkan dengan f^{-1}. Misalkan a adalah anggota himpunan A dan b adalah anggota himpunan B, maka $f^{-1}(b) = a$ jika $f(a) = b$.

- Fungsi yang berkoresponden satu-ke-satu sering dinamakan juga fungsi yang *invertible* (dapat dibalikkan), karena kita dapat mendefinisikan fungsi balikannya. Sebuah fungsi dikatakan *not invertible* (tidak dapat dibalikkan) jika ia bukan fungsi yang berkoresponden satu-ke-satu, karena fungsi balikannya tidak ada.

Contoh 37. Relasi

$$f = \{(1, u), (2, w), (3, v)\}$$

dari $A = \{1, 2, 3\}$ ke $B = \{u, v, w\}$ adalah fungsi yang berkoresponden satu-ke-satu. Balikan fungsi f adalah

$$f^{-1} = \{(u, 1), (w, 2), (v, 3)\}$$

Jadi, f adalah fungsi *invertible*.

Contoh 38. Tentukan balikan fungsi $f(x) = x - 1$.

<u>Penyelesaian</u>:
Fungsi $f(x) = x - 1$ adalah fungsi yang berkoresponden satu-ke-satu, jadi balikan fungsi tersebut ada.
Misalkan $f(x) = y$, sehingga $y = x - 1$, maka $x = y + 1$. Jadi, balikan fungsi balikannya adalah $f^{1}(y) = y + 1$.

Contoh 39. Tentukan balikan fungsi $f(x) = x^2 + 1$.

<u>Penyelesaian</u>:

Dari Contoh 3.41 dan 3.44 kita sudah menyimpulkan bahwa $f(x) = x - 1$ bukan fungsi yang berkoresponden satu-ke-satu, sehingga fungsi balikannya tidak ada. Jadi, $f(x) = x^2 + 1$ adalah funsgi yang *not invertible*.

Komposisi dari dua buah fungsi.

Misalkan g adalah fungsi dari himpunan A ke himpunan B, dan f adalah fungsi dari himpunan B ke himpunan C. Komposisi f dan g, dinotasikan dengan $f \circ g$, adalah fungsi dari A ke C yang didefinisikan oleh

$$(f \circ g)(a) = f(g(a))$$

Contoh 40. Diberikan fungsi

$$g = \{(1, u), (2, u), (3, v)\}$$

yang memetakan $A = \{1, 2, 3\}$ ke $B = \{u, v, w\}$, dan fungsi

$$f = \{(u, y), (v, x), (w, z)\}$$

yang memetakan $B = \{u, v, w\}$ ke $C = \{x, y, z\}$. Fungsi komposisi dari A ke C adalah

$$f \circ g = \{(1, y), (2, y), (3, x)\}$$

Contoh 41. Diberikan fungsi $f(x) = x - 1$ dan $g(x) = x^2 + 1$. Tentukan $f \circ g$ dan $g \circ f$.

<u>Penyelesaian</u>:

(i) $(f \circ g)(x) = f(g(x)) = f(x^2 + 1) = x^2 + 1 - 1 = x^2$.

(ii) $(g \circ f)(x) = g(f(x)) = g(x - 1) = (x - 1)^2 + 1 = x^2 - 2x + 2$.

Beberapa Fungsi Khusus

1. Fungsi *Floor* dan *Ceiling*

Misalkan x adalah bilangan riil, berarti x berada di antara dua bilangan bulat.

Fungsi *floor* dari x:

$\lfloor x \rfloor$ menyatakan nilai bilangan bulat terbesar yang lebih kecil atau sama dengan x

Fungsi *ceiling* dari x:

$\lceil x \rceil$ menyatakan bilangan bulat terkecil yang lebih besar atau sama dengan x

Dengan kata lain, fungsi *floor* membulatkan x ke bawah, sedangkan fungsi *ceiling* membulatkan x ke atas.

Contoh 42. Beberapa contoh nilai fungsi *floor* dan *ceiling*:

$$\lfloor 3.5 \rfloor = 3 \qquad \lceil 3.5 \rceil = 4$$
$$\lfloor 0.5 \rfloor = 0 \qquad \lceil 0.5 \rceil = 1$$
$$\lfloor 4.8 \rfloor = 4 \qquad \lceil 4.8 \rceil = 5$$
$$\lfloor -0.5 \rfloor = -1 \qquad \lfloor -0.5 \rfloor = 0$$
$$\lfloor -3.5 \rfloor = -4 \qquad \lceil -3.5 \rceil = -3$$

Contoh 42. Di dalam komputer, data dikodekan dalam untaian *byte*, satu *byte* terdiri atas 8 bit. Jika panjang data 125 bit, maka jumlah *byte* yang diperlukan untuk merepresentasikan data adalah $\lceil 125/8 \rceil = 16$ *byte*. Perhatikanlah bahwa $16 \times 8 = 128$ bit, sehingga untuk *byte* yang terakhir perlu ditambahkan 3 bit ekstra agar satu *byte* tetap 8 bit (bit ekstra yang ditambahkan untuk menggenapi 8 bit disebut *padding bits*).

2. Fungsi modulo

Misalkan a adalah sembarang bilangan bulat dan m adalah bilangan bulat positif.

a mod m memberikan sisa pembagian bilangan bulat bila a dibagi dengan m

$a \bmod m = r$ sedemikian sehingga $a = mq + r$, dengan $0 \le r < m$.
(q=sisa pembagian bilangan bulat))

Contoh 43. Beberapa contoh fungsi modulo

$25 \bmod 7 = 4$
$15 \bmod 4 = 3$
$3612 \bmod 45 = 12$
$0 \bmod 5 = 5$
$-25 \bmod 7 = 3$ (sebab $-25 = 7 \cdot (-4) + 3$)

3. Fungsi Faktorial

$$n! = \begin{cases} 1 & , n = 0 \\ 1 \times 2 \times \Lambda \ . \times (n-1) \times n & , n > 0 \end{cases}$$

4. Fungsi Eksponensial

$$a^n = \begin{cases} 1 & , n = 0 \\ \underbrace{a \times a \times \Lambda \times a}_{n} & , n > 0 \end{cases}$$

Untuk kasus perpangkatan negatif,

$$a^{-n} = \frac{1}{a^n}$$

5. Fungsi Logaritmik

Fungsi logaritmik berbentuk

$$y = {}^a\log x \leftrightarrow x = a^y$$

Fungsi Rekursif

- Fungsi f dikatakan fungsi rekursif jika definisi fungsinya mengacu pada dirinya sendiri.

Contoh: $n! = 1 \times 2 \times \ldots \times (n-1) \times n = (n-1)! \times n.$

$$n! = \begin{cases} 1 & , n = 0 \\ n \times (n-1)! & , n > 0 \end{cases}$$

Fungsi rekursif disusun oleh dua bagian:

(a) *Basis*

Bagian yang berisi nilai awal yang tidak mengacu pada dirinya sendiri. Bagian ini juga sekaligus menghentikan definisi rekursif.

(b) *Rekurens*

Bagian ini mendefinisikan argumen fungsi dalam terminologi dirinya sendiri. Setiap kali fungsi mengacu pada dirinya sendiri, argumen dari fungsi harus lebih dekat ke nilai awal (basis).

- Contoh definisi rekursif dari faktorial:
  (a) basis:
  $n! = 1$, jika $n = 0$
  (b) rekurens:
  $n! = n \times (n - 1)!$, jika $n > 0$

5! dihitung dengan langkah berikut:

(1) $5! = 5 \times 4!$ (rekurens)
(2) $4! = 4 \times 3!$
(3) $3! = 3 \times 2!$
(4) $2! = 2 \times 1!$
(5) $1! = 1 \times 0!$
(6) $0! = 1$

(6') $0! = 1$
(5') $1! = 1 \times 0! = 1 \times 1 = 1$
(4') $2! = 2 \times 1! = 2 \times 1 = 2$
(3') $3! = 3 \times 2! = 3 \times 2 = 6$
(2') $4! = 4 \times 3! = 4 \times 6 = 24$
(1') $5! = 5 \times 4! = 5 \times 24 = 120$

Jadi, $5! = 120$.

Contoh 44. Di bawah ini adalah contoh-contoh fungsi rekursif lainnya:

1. $F(x) = \begin{cases} 0 & , x = 0 \\ 2F(x-1) + x^2 & , x \neq 0 \end{cases}$

2. Fungsi Chebysev

$$T(n,x) = \begin{cases} 1 & , n = 0 \\ x & , n = 1 \\ 2xT(n-1,x) - T(n-2,x) & , n > 1 \end{cases}$$

3. Fungsi fibonacci:

$$f(n) = \begin{cases} 0 & , n = 0 \\ 1 & , n = 1 \\ f(n-1) + f(n-2) & , n > 1 \end{cases}$$

Definisi Rekursif

Ada kalanya kita mengalami kesulitan untuk mendefinisikan suatu obyek secara eksplisit.

Mungkin lebih mudah untuk mendefinisikan obyek tersebut dengan menggunakan dirinya sendiri. Ini dinamakan sebagai proses rekursif.

Kita dapat mendefinikan barisan, fungsi dan himpunan secara rekursif.

Barisan yang didefinisikan secara rekursif

Contoh:

Barisan bilangan pangkat dari 2

$$a_n = 2^n \text{ untuk } n = 0, 1, 2, \ldots .$$

Barisan ini dapat didefinisikan secara rekursif:

$$a_0 = 1$$

$$a_{n+1} = 2a_n \quad \text{untuk } n = 0, 1, 2, \ldots$$

Langkah-langkah untuk mendefinisikan barisan secara rekursif:

1. Langkah basis: Spesifikasi anggota awal.
2. Langkah rekursif: Berikan aturan untuk membangun anggota baru dari anggota yang telah ada.

Contoh barisan yang didefinisikan secara rekursif

Berikan definisi rekursif dari $a_n=r^n$, dengan $r\in\mathbb{N}$, $r\neq 0$ dan n bilangan bulat positif.

Solusi:

Definisikan $a_0=r^0=1$

dan $a_{n+1}=r\cdot a_n$ untuk n = 0, 1, 2, …

Fungsi yang didefinisikan secara rekursif

Langkah-langkah untuk mendefinisikan fungsi dengan domain bilangan cacah:

1. Langkah basis: Definisikan nilai fungsi pada saat nol.
2. Langkah rekursif: Berikan aturan untuk mencari nilai fungsi untuk setiap bilangan bulat berdasarkan nilai fungsi pada bilangan bulat yang lebih kecil.

Definisi seperti itu disebut rekursif atau definisi induktif.

Contoh fungsi yang didefinisikan secara rekursif

$f(0) = 3$

$f(n + 1) = 2f(n) + 3$

Maka

$$f(0) = 3$$

$$f(1) = 2f(0) + 3 = 2 \cdot 3 + 3 = 9$$

$$f(2) = 2f(1) + 3 = 2 \cdot 9 + 3 = 21$$

$$f(3) = 2f(2) + 3 = 2 \cdot 21 + 3 = 45$$

$$f(4) = 2f(3) + 3 = 2 \cdot 45 + 3 = 93$$

Contoh fungsi yang didefinisikan secara rekursif (2)

Bagaimana kita dapat mendefinisikan fungsi faktorial f(n) = n! secara rekursif?

$$f(0) = 1$$

Karena (n+1)! = n! (n+1) maka

$$f(n + 1) = (n + 1)f(n)$$

$f(0) = 1$

$f(1) = 1\ f(0) = 1 \cdot 1 = 1$

$f(2) = 2\ f(1) = 2 \cdot 1 = 2$

$f(3) = 3\ f(2) = 3 \cdot 2 = 6$

$f(4) = 4\ f(3) = 4 \cdot 6 = 24$

Contoh fungsi yang didefinisikan secara rekursif (3)

Bagaimana kita dapat mendefinisikan fungsi

$$f(n) = \sum_{k=0}^{n} a_k$$

secara rekursif?

Contoh terkenal: Bilangan Fibonacci

$f_0 = 0$, $f_1 = 1$

$f_n = f_{n-1} + f_{n-2}$, $n=2,3,4,\ldots$

$f_0 = 0$

$f_1 = 1$

$f_2 = f_1 + f_0 = 1 + 0 = 1$

$f_3 = f_2 + f_1 = 1 + 1 = 2$

$f_4 = f_3 + f_2 = 2 + 1 = 3$

$f_5 = f_4 + f_3 = 3 + 2 = 5$

$f_6 = f_5 + f_4 = 5 + 3 = 8$

Tunjukkan bahwa untuk $n \geq 3$,

$f_n < \alpha^n$ dengan $\alpha = (1+\sqrt{5})/2$.

Himpunan yang didefinisikan secara rekursif

Langkah-langkah dalam mendefinisikan suatu himpunan secara rekursif:

1. Langkah basis:

   Spesifikasi koleksi awal dari anggota

2. Langkah rekursif:

   Mendefinisikan aturan konstruksi anggota baru dari anggota yang telah diketahui

Contoh himpunan yang didefinisikan secara rekursif

Misalkan S didefinisikan secara rekursif oleh:

$3 \in S$

$(x+y) \in S$ jika $x \in S$ dan $y \in S$

Maka S adalah himpunan bilangan bulat positif yang habis dibagi 3.

Bukti:

Misalkan A himpunan yang beranggotakan semua bilangan bulat positif yang habis dibagi 3.

Untuk membuktikan bahwa A = S, harus ditunjukkan

$$A \subseteq S \text{ and } S \subseteq A.$$

Bagian I: Akan dibuktikan $A \subseteq S$, yaitu menunjukkan bahwa setiap bilangan bulat positif yang habis dibagi 3 ada di S (dengan menggunakan induksi matematika).

Contoh himpunan yang didefinisikan secara rekursif (2)

Misalkan P(n): proposisi “3n anggota S”.

1. Langkah basis: P(1) benar, karena $3 \in S$.
2. Langkah induktif:

   Asumsikan P(k) benar, yaitu $3k \in S$.

   Akan ditunjukkan P(k+1) juga benar, yaitu

   $3(k+1) \in S$

   Karena $3k \in S$ dan $3 \in S$, berdasarkan definisi rekursif dari S, $3k+3 = 3(k+1)$ juga ada di S.
3. Konklusi:

   Jadi, setiap bilangan bulat positif yang habis dibagi 3 ada di S.

Kesimpulan dari bagian I adalah $A \subseteq S$.

Contoh himpunan yang didefinisikan secara rekursif (3)

Bagian II: Akan ditunjukkan $S \subseteq A$ dengan menggunakan definisi rekursif dari S.

Langkah basis:

Akan ditunjukkan setiap anggota awal S ada di A.

Karena 3 habis dibagi 3 maka $3 \in A$.

Langkah rekursif:

Akan ditunjukkan bahwa setiap bilangan bulat yang dibangun dengan mengunakan langkah rekursif juga merupakan anggota A, yaitu

$(x+y) \in A$ jika $x,y \in S$ (yang diasumsikan $\in A$).

Jika x dan y keduanya di A, maka $3 \mid x$ dan $3 \mid y$. Akibatnya, $3 \mid (x + y)$.

Kesimpulan dari bagian II adalah $S \subseteq A$.

Jadi, secara keseluruhan, berlaku $A = S$.

Induksi Struktural

Dalam membuktikan hasil-hasil yang berkaitan dengan himpunan yang didefinisikan secara rekursif, akan lebih mudah apabila digunakan suatu bentuk induksi matematika yang disebut induksi struktural.

Langkah-langkah dalam induksi struktural:

1. Langkah basis:

   Menunjukkan bahwa hasil yang akan dibuktikan berlaku untuk semua anggota awal.

2. Langkah rekursif:

   Menunjukkan bahwa jika hasil yang akan dibuktikan berlaku untuk anggota-anggota yang digunakan untuk membangun anggota baru, maka hasil tersebut juga berlaku untuk anggota yang baru dibangun.

Himpunan string atas alfabet

Himpunan string Σ^* atas alfabet Σ dapat didefinisikan secara rekursif oleh:

1. Langkah basis:

   $\lambda \in \Sigma^*$ (λ adalah string kosong yang tidak memuat simbol)

2. Langkah rekursif:

   Jika $w \in \Sigma^*$ dan $x \in \Sigma$, maka $wx \in \Sigma^*$

Contoh:

Jika $\Sigma = \{0,1\}$ maka string yang merupakan anggota Σ^* adalah:

- yang didefinisikan sebagai anggota Σ^* dalam langkah basis,
- 0 dan 1 yang dibentuk dalam langkah rekursif pertama,
- 00, 01, 10, dan 11 yang dibentuk dalam langkah rekursif kedua,
- dst

Himpunan string atas alfabet (2)

Konkatenasi

Sebagai operasi kombinasi dari dua string, konkatenasi didefinisikan secara rekursif sebagai:

1. Langkah basis:

   Jika $w \in \Sigma^*$, maka $w . \lambda = w$, dengan λ string kosong

2. Langkah rekursif:

   Jika $w_1 \in \Sigma^*$ dan $w_2 \in \Sigma^*$ dan $x \in \Sigma$, maka

$$w_1 . (w_2 x) = (w_1 . w_2) x$$

$w_1 . w_2$ seringkali ditulis sebagai $w_1 w_2$

Contoh:

Konkatenasi dari $w_1 =$ *meng* dan $w_2 =$ *apa* adalah

$$w_1 w_2 = \textit{mengapa}$$

Himpunan string atas alfabet (3)

Panjang string

Panjang dari string w, $\ell(w)$ dapat didefinisikan secara rekursif oleh:

$\ell(\lambda) = 0$,

$\ell(w\,x) = \ell(w) + 1$ jika $w \in \Sigma^*$ dan $x \in \Sigma$.

Gunakan induksi struktural untuk membuktikan bahwa

$$\ell(x\,y) = \ell(x) + \ell(y).$$

Perluasan induksi

Induksi matematika dapat diperluas untuk membuktikan hasil-hasil mengenai himpunan yang memiliki sifat terurut dengan baik.

Contoh: himpunan N x N

Contoh perluasan induksi

Misalkan $a_{m,n}$ didefinisikan secara rekursif untuk *(m,n)* $\in$ N x N oleh

$a_{0,0} = 0$ dan

$$a_{m,n} = \begin{cases} a_{m-1,n} + 1, \text{ jika } n = 0 \text{ dan } m > 0 \\ a_{m,n-1} + n, \qquad\qquad \text{jika } n > 0 \end{cases}$$

Tunjukkan bahwa

$$a_{m,n} = m + n(n+1)/2$$

untuk setiap *(m,n)* $\in$ N x N.

LOGIKA INFORMATIKA

Suraya

Strategi Pembalikan

- Menjelaskan konsistensi antara sekumpulan ekspresi ekspresi logika yang dibuat dari pernyataan pernyataan
- Menjelaskan teknik strategi pembalikan yang menyalahkan kesimpulan untuk membuktikan validitas suatu argumen
- Menjelaskan teknik model yang merupakan salah satu strategi pembalikan untuk memastikan nilai nilai premis benar yang harus diikuti oleh kesimpulan yang benar.

Strategi Pembalikan

- Bab sebelumnya kita membahas tabel kebenaran untuk membuktikan ekspresi ekspresi logika yang berupa tatutologi, kontradiksi dan contingent, selain itu juga membahas pemakaian hukum hukum logika untuk membuktikan tautologi ataupun penyederhanaan sesederhana mungkin suatu ekspresi logika yang rumit
- Bab ini akan membahas teknik strategi pembalikan (*refutation strategy*) untuk membuktikan validitas suatu ekspresi logika untuk argumen, disini kesimpulan argumen yang harus disalahkan dengan cara dinegasikan atau diberi nilai F

Konsistensi

Tabel kebenaran bermanfaat untuk membuktikan validitas ekspresi logika, tetapi memerlukan tabel yang sangat besar untuk menyelesaikan ekspresi logika yang banyak variabel proposionalnya (2^n).

Logika proposional tidak bisa menangani kerumitan bahasa yang dipergunakan sehari hari. Bahasa yang cukup rumit akan ditangani oleh logika predikat.

Contoh : sekumpulan pernyataan berikut ini

"Harga gula turun jika impor gula naik. Pabrik gula tidak senang jika harga gula turun. Impor gula naik. Pabrik gula senang".

Pernyataan pernyataan tersebut di atas disebut konsisten satu dengan lainnya jika semuanya bernilai benar. Diperhatikan pernyataan di atas bukan argumen karena tidak ada kesimpulan yang ditandai dengan kata "Dengan demikian"

Koleksi dari pernyataan pernyataan disebut konsisten jika pernyataan pernyataan tersebut secara simultan semuanya bernilai benar.

Konsistensi dapat dibuktikan dg membuat pernyataan menjadi ekspresi logika dan dibuktikan melalui tabel kebenaran

Langkah 1

Mengubah ke variabel proposional

A = Harga gula turun

B = Impor gula naik

C = Pabrik gula senang

Langkah 2

Mengubah pernyatan menjadi ekspresi logika

(1) $B \rightarrow A$

(2) $A \rightarrow \neg C$

(3) B

(4) C

Konsistensi dapat dibuktikan dg membuat pernyataan menjadi ekspresi logika dan dibuktikan melalui tabel kebenaran

Langkah (3)

Menyusun ekspresi logika menjadi satu kesatuan

(B→A)^(A→¬C)^B^C

Langkah (4)

Membuat tabel kebenaran:

A	B	C	B→A	¬C	A→¬C	
F	F	F	T	T	T	F
F	F	T	T	F	T	F
F	T	F	F	T	T	F
F	T	T	F	F	T	F
DST.						

Konsistensi

Tidak ada satu pun ekspresi logika $(A \rightarrow B) \wedge (\neg C \rightarrow A) \wedge B \wedge C$ yang mempunyai nilai T pada deretan pasangan yang sama sehingga hasilnya juga dipastikan F.

Jadi kumpulan pernyataan tersebut tidak konsisten.

Konsisten juga dapat diterapkan pada argumen, yang premis premis harus bernilai T dan kesimpulan bernilai T sehingga hasilnya juga harus T. Oleh karena itu argumen dapat disebut valid.

Contoh :

(1) Jika Peterpen mengadakan konser, maka penonton akan hadir jika harga tiket tidak terlalu tinggi

(2) Jika Peterpen mengadakan konser, maka harga tiket tidak terlalu tinggi

(3) *Dengan demikian*, jika Peterpen mengadakan konser, maka penonton akan hadir.

Konsistensi

Validitas di atas harus dibuktikan dengan tabel kebenaran

Langkah 1

Mengubah ke variabel proposional

A = Peterpen mengadakan konser

B = Penonton akan hadir

C = Harga tiket terlalu tinggi

Langkah 2

Mengubah pernyataan menjadi ekspresi logika

(1) $A \rightarrow (\neg C \rightarrow B)$

(2) $A \rightarrow \neg C$

(3) $A \rightarrow B$

Konsistensi

Langkah 3

Menyusun ekspresi logika menjadi satu kesatuan
Untuk argumen, cara menulis ekspresi logikanya ada beberapa pilihan”

(1) $((A \rightarrow (\neg C \rightarrow B)) \wedge (A \rightarrow \neg C)) \rightarrow (A \rightarrow B)$
(2) $\{A \rightarrow (\neg C \rightarrow B), A \rightarrow \neg C\} \models (A \rightarrow B)$

Untuk pembuatan tabel kebenaran sebaiknya kita gunakan penulisan ke (1) agar lebih mudah menyusunnya
Jika dengan strategi pembalikan, kesimpulan diberi negasi dan diberi operator ^

Operasi Strategi Pembalikan

Strategi pembalikan dilakukan dengan cara menyalahkan kesimpulan dari argumen yakni:

(1) Menegasi kesimpulan

(2) Memberi nilai F

Pada contoh argumen tentang konser Peterpen di atas kesimpulan akan dinegasikan dan akan ditulis

$A \rightarrow (\neg C \rightarrow B) \wedge (A \rightarrow \neg C) \wedge \neg (A \rightarrow B)$

Maka tabel kebenarannya sbb:

Operasi Strategi Pembalikan

A	B	C	¬C	¬C→B	A→(¬C→B)	A--¬C	A→B	¬(A→B)	
F	F	F	T	F	T	T	T	F	F
F	F	T	F	T	T	T	T	F	F
F	T	F	T	T	T	T	T	F	F
F	T	T	F	T	T	T	T	F	F
T	F	F	T	F	F	T	F	T	F
T	F	T	F	T	T	F	F	T	F
T	T	F	T	T	T	T	T	F	F
T	T	T	F	T	T	F	T	F	F

Ternyata hasil negasi dari kesimpulan dengan premis premis tidak konsisten, atau hasilnya F. Jadi disini kemungkinan negasi dari kesimpulan bernilai T bersama sama dengan premis premis. Karena strategi Pembalikan, hasil yang semula bernilai F justru menjadi bernilai T sehingga argumen di atas valid

Model dan Countermodel

Jika ada premis premis dan kesimpulan bernilai T, bisa dipastikan argumen tersebut valid, teknik ini disebut model, sedangkan kebalikannya disebut countermodel

Lihat contoh tentang konser Peterpen
$\{A \rightarrow (\neg C \rightarrow B), A \rightarrow \neg C\} \models (A \rightarrow B)$
Dan ditulis sebagai berikut:
$(A \rightarrow (\neg C \rightarrow B)) \wedge (A \rightarrow \neg C) \wedge (A \rightarrow B)$

Maka sekarang akan diberi nilai sbb:
(1) $(A \rightarrow (\neg C \rightarrow B)) \equiv T$ (premis 1)
(2) $(A \rightarrow \neg C) \equiv T$ (premis 2)
(3) $(A \rightarrow B) \equiv F$ (kesimpulan)

Model dan Countermodel

Setiap premis dan kesimpulan serta variabel proposional pasti mempunyai nilai dan ditulis sbb:

$v(A \rightarrow \neg C) \equiv T$, $v(\neg C) \equiv T$ dst
V berarti **value of** atau **nilai dari**
Teknik model akan dilakukan sesuai dengan langkah berikut ini:

Langkah 1 (cek dengan kesimpulan)
(1) Jika $(A \rightarrow B) \equiv F$, maka hanya ada satu kemungkinan yakni $v(A) = T$ dan $v(B) \equiv F$
(2) Jadi $v(A) \equiv T$
(3) Jadi $v(B) \equiv F$

Model dan Countermodel

Langkah 2 (cek dengan premis 1)

(1) Jika $v(A \rightarrow (\neg C \rightarrow B)) \equiv T$, sedang sudah diketahui $v(A) \equiv T$, maka $v(\neg C \rightarrow B) \equiv T$

(2) Jika $v(\neg C \rightarrow B) \equiv T$, sedangkan $v(B) \equiv F$, maka di sini hanya ada pilihan yakni $v(\neg C) \equiv F$

(3) (3) Jadi $v(\neg C) \equiv F$, maka $v(c) \equiv T$

Langkah 3 (cek dengan premis 2)

(1) Jika $v(A \rightarrow \neg C) \equiv T$, sedangkan $v(A) \equiv T$, dan $v(\neg C) \equiv F$

(2) Ini tidak mungkin terjadi. Jika $v(A) \equiv T$, dan $v(\neg C) \equiv F$, maka seharusnya $v(A \rightarrow \neg C) \equiv F$

Langkah 4 (kesimpulan)

(1) Jadi tidak mungkin pada saat yang sama $v(A \rightarrow (\neg C \rightarrow B)) \equiv T$, $v(A \rightarrow \neg C) \equiv T$ dan $v(A \rightarrow B) \equiv F$.

(2) Jika tidak mungkin, maka karena ada strategi pembalikan argumen di atas valid.

Model dan Countermodel

Hasil dalam bentuk tabel kebenarannya:

$(A \rightarrow (\neg C \rightarrow B)) \wedge (A \rightarrow \neg C) \wedge (A \rightarrow B)$

$(A \rightarrow (\neg C \rightarrow B)) \wedge (A \rightarrow \neg C)$

A	B	C	¬C	C→B	A→(¬C→B)	A→¬C		A→B	
T	F	T	F	T	T	F	F	F	F

Dalam kata lain, kesimpulan (A→B) adalah konsekuensi yang logis dari premis premis (A→(¬C→B)) dan (A→¬C), atau (A→B) adalah model dari (A→(¬C→B)) ^(A→¬C)

Perhatikan pada tabel kebenaran, jika tidak dilakukan strategi pembalikan. Penulisan ekspresi logika dari argumen tersebut adalah:

(((A→(¬C→B))^(A→¬C))→(A→B)

Model dan Countermodel

$(((A \rightarrow (\neg C \rightarrow B)) \wedge (A \rightarrow \neg C)) \rightarrow (A \rightarrow B)$

Tabel kebenarannya sebagai berikut:

A	B	C	¬C	¬C→B	A→(¬C→B)	A→¬C	A→B	
F	**F**	**F**	**T**	**F**	**T**	**T**	**T**	**T**
F	**F**	**T**	**F**	**T**	**T**	**T**	**T**	**T**
F	**T**	**F**	**T**	**T**	**T**	**T**	**T**	**T**
F	**T**	**T**	**F**	**T**	**T**	**T**	**T**	**T**
T	F	F	T	F	F	T	F	T
T	F	T	F	T	T	F	F	T
T	**T**	**F**	**T**	**T**	**T**	**T**	**T**	**T**
T	T	T	F	T	T	F	T	T

Hasilnya tautologi dan membuktikan argumennya valid.

Premis premis yang bernilai T dan kesimpulan T ada pada baris yang di tandai warna merah

LOGIKA INFORMATIKA

Suraya

Tablo Semantik

- Menjelaskan aturan dan pembuatan tablo semantik untuk membuktikan konsistensi dan validitas argumen dengan mengaplikasikan strategi pembalikan berdasarkan aturan pembuatan tablo semantik
- Memahami bahwa aturan tablo semantik sebenarnya identik dengan hukum hukum logika
- Memahami pentingnya strategi pembalikan dengan menegasi kesimpulan untuk membuktikan validitas argumen dengan tablo semantik.

Tablo Semantik

- Tablo Semantik berbasis pada strategi pembalikan, trategi pada tablo semantik dilakukan dengan memberi negasi pada kesimpulan dan memeriksa hasil yang diperoleh.
- Dibuktikan apakah kesimpulan yang bernilai F dapat diperoleh dari premis premis yang bernilai T. jika tidak bisa maka argumen disebut valid, tetapi jika bisa, argumen tidak valid
- Tablo semantik bentuk bentuk proposisi yang dibangun berdasarkan aturan aturan tertentu yang biasanya berbentuk pohon terbalik dengan cabag dan ranting yang relevan

Aturan Tablo Semantik

Ada 10 aturan dalam Tablo Semantik:

Aturan (1): A^B

Jika tablo berisi A^B, maka tablo dapat dikembangkan menjadi tablo baru dengan menambahkan A dan B pada tablo A^B.

Bentuknya seperti berikut:

A^B

A

B

Aturan (2): AvB

Jika tablo berisi AVB, maka tablo dapat dikembangkan membentuk tablo baru dengan menambahkan dua cabang baru, satu berisi A dan satunya B pada tablo AVB. Bentuknya seperti berikut:

Aturan Tablo Semantik

Aturan (3): A→B

Jika tablo berisi A→B, maka tablo dapat dikembangkan membentuk tablo baru dengan menambahkan dua cabang baru, satu berisi ¬ A dan satunya B pada tablo A→B.

Aturan (4): A↔B

Jika tablo berisi A←→B, maka tablo dapat dikembangkan membentuk tablo baru dengan menambahkan dua cabang baru, satu berisi A^B dan satunya ¬A^¬B pada tablo A←→B.

Aturan (5): ¬ ¬ A

¬ ¬ A

A

Jika tablo berisi ¬¬ A maka tablo dapat dikembangkan membentuk tablo baru berisi A pada tablo ¬¬A

Aturan Tablo Semantik

Aturan (6): ¬(A^ B)

¬(A^ B)

¬A ¬B

Jika tablo berisi ¬(A^ B) maka tablo dapat dikembangkan membentuk tablo baru dengan menambahkan dua cabang baru, satu berisi ¬A dan satunya ¬B pada tablo ¬(A^ B)

Aturan (7): ¬(AvB)

¬(AvB)

¬A

¬B

Jika tablo berisi ¬(AvB) maka tablo dapat dikembangkan membentuk tablo baru dengan menambahkan ¬A dan ¬B pada tablo ¬(AvB)

Aturan (8): ¬(A→B)

¬(A→B)

A

¬B

Jika tablo berisi ¬(A→B) maka tablo dapat dikembangkan membentuk tablo baru dengan menambahkan A dan ¬B pada tablo ¬(A→B)

Aturan Tablo Semantik

Aturan (9): ¬(A←→B)

Jika tablo berisi ¬(A←→ B) maka tablo dapat dikembangkan membentuk tablo baru dengan menambahkan dua cabang baru, satu berisi A^¬B dan satunya ¬A^B pada tablo ¬(A←→B)

Aturan (10): Jika ada bentuk logika A dan negasi (¬A) yang berada pada satu deretan cabang dan tablo, maka terjadi ketidakkonsistenan pada cabang tersebut, dan cabang dinyatakan “tertutup (closed)” dan cabang tersebut tidak bisa dikembangkan lagi.

Hal tersebut di atas disebabkan karena A dan ¬A tidak mungkin benar bersama sama pada satu saat tertentu.

Aturan Tablo Semantik

Definisi :

Jika semua cabang tablo tertutup, maka ekspresi logika disebut bersama sama tidak konsisten (mutually inconsistent) atau mereka tidak bisa bernilai benar bersama sama.

Tablo Semantik pada suatu Himpunan Ekspresi Logika

Contoh:

Apakah 2 buah ekspresi logika ini konsisten bersama sama

$\neg(A \rightarrow B)$ dan $\neg A \vee B$

Tablo Semantik pada suatu Himpunan Ekspresi Logika

Tablo Semantik yang dibuat seperti berikut:

Perhatikan bahwa dua cabang dari tablo di atas tertutup karena cabang sebelah kiri berisi A dan ¬A, sedang cabang kanan berisi B dan ¬B, maka kesimpulannya adalah tidak konsisten bersama sama

Pembenaran Aturan Tablo Semantik

Aturan Tablo semantik dapat dipandang sebagai aturan sistem deduktif atau sistem pembuktian yang tidak perli ditafsirkan pada kontens lain. Aturan Tablo semantik sangat sintaksis.

Aturan tablo semantik sangat beralasan dan realistis karena berbasis pada aturan hukum logika, lihat aturan aturannya:

Aturan (1) : A^B

A^B

A

B

Menunjukkan bahwa jika (A^B) benar, maka A dan B juga bernilai benar sehingga cabang tablo untuk ekspresi ini juga benar bersama sama.

Pembenaran Aturan Tablo Semantik

Aturan (2) : AvB

Aturan ini menunjukkan bahwa jika (AvB) benar, maka A bisa benar atau B juga benar. Untuk itu satu cabang tablo harus menunjukkan hal ini, atau ada konsistensi di sini.

Figure 8-8

Pembenaran Aturan Tablo Semantik

Aturan (3) : $A \rightarrow B$

$$A \rightarrow B$$

$$\neg A \qquad\qquad B$$

Pada hukum logika sudah diketahui $(A \rightarrow B) \equiv \neg A \vee B$ sehingga aplikasinya sama seperti hukum nomor (2)

Figure 8-8

Pembenaran Aturan Tablo Semantik

Aturan (4) : A←→B

A←→B

A^B ¬A^B

Pada hukum logika juga diketahui (A←→B) ≡ (A^B)v(¬A^¬B) sehingga aplikasinya sama seperti hukum nomor (2)

Pembenaran Aturan Tablo Semantik

Aturan (5) : ¬¬A

¬¬A

A

Ini merupakan aplikasi hukum negasi ganda, yakni ¬¬A ≡ A

Aturan (6) : ¬(A^B)

¬(A^B)

¬A ¬B

Pada hukum De Morgan sudah diketahui bahwa ¬(A^B) ≡ ¬A^¬B sehingga aturan nomro (2) dipakai sekali lagi.

Pembenaran Aturan Tablo Semantik

Aturan (7) : ¬(AvB)

¬(AvB)

¬A

¬B

Hukum De Morgan lainnya diketahui bahwa

¬(AvB) ≡¬A^¬B sehingga dipakai aturan nomor (1)

Pembenaran Aturan Tablo Semantik

Aturan (8) : $\neg(A \rightarrow B)$

$\neg(A \rightarrow B)$

A

$\neg B$

Penyederhanaan bisa dilakukan pada $\neg(A \rightarrow B)$ sehingga menjadi:

$\neg(A \rightarrow B)$

$\equiv \neg(\neg A \vee B)$	$A \rightarrow B$
$\equiv (\neg\neg A \vee \neg B)$	De Morgan'S Law
$\equiv (A \wedge \neg B)$	Law of Double Negation

Aturan (1) dapat dipakai pada ekspresi logika ini

Pembenaran Aturan Tablo Semantik

Aturan (9) : ¬(A←→B)

¬(A←→B)

A^¬B ¬A^B

Sedangkan untuk ¬(A←→B) dapat juga dilakukan penyederhanaan seperti berikut:

¬(A←→B)	
≡ ¬((A→B)^(B→A)	A→B
≡ ¬((¬AvB)^(¬BvA))	A→B
≡ (¬(¬AvB)v¬(¬BvA))	De Morgan'S Law
≡ (¬¬A^¬B)v(¬¬B^¬A)	De Morgan'S Law
≡ (A^¬B)v(B^¬A)	Law of Double Negation
≡ (A^¬B)v(¬A^B)	Komutatif

Aturan (2) dapat dipakai pada ekspresi logika ini

Pembenaran Aturan Tablo Semantik

Bagaimana jika terjadi tablo yang tidak tertutup dan memastikan adanya konsistensi.

Contoh:

(1) $A \vee \neg B$

(2) $B \wedge \neg C$

(3) $C \rightarrow A$

(4) B Aturan (1) pada baris (2)

(5) $\neg C$

(6) A $\neg B$ Aturan (2) pada baris (1)

Tutup

(7) $\neg C$ A Aturan (3) pada baris (3)

Tablo tidak dapat ditutup sehingga terjadi konsistensi bersama-sama (mutually consistency) pada himpunan ekspresi logika.

Pembenaran Aturan Tablo Semantik

Konsistensi bisa juga dibuktikan dengan teknik model, yaitu dengan mengambil satu variabel proposisi pada cabang yang tidak tertutup. misalnya A berilah nilai T pada variabel tersebut.

Pada contoh di atas, misalnya $V(B) \equiv T$, maka $V(\neg C) \equiv T$ jadi $V(C) \equiv F$. Periksa dengan baris (2). Jika $V(\neg C) \equiv T$, maka pasti $V(B) \equiv T$, maka $V(\neg B) \equiv F$. Periksa dengan baris (3). Jika $V(\neg B) \equiv F$. Periksa dengan baris (3). Jika $V(\neg B) \equiv F$, sedangkan $V(A) \equiv T$, maka $V(A \vee \neg B) \equiv T$. Jadi mudah ditebak bahwa $V(A \wedge \neg B) \equiv T$, $V(B \wedge \neg C) \equiv T$, dan $V(C \rightarrow A) \equiv T$

Tablo Semantik pada Argumen

Tablo semantik juga dapat diimplementasikan pada pembuktian validitas suatu argumen

Contoh:

Jika Badu mencontek saat ujian, maka dosen akan datang jika pengawas tidak lalai. Jika Badu mencontek saat ujian, maka pengawas tidak lalai. Dengan demikian, jika Badu mencontek, maka dosen akan datang.

Apakah argumen di atas valid, atau apakah kesimpulannya secara logis mengikuti premis-premisnya.

Tablo semnatik memakai teknik strategi pembalikan dengan menegasi kesimpulan.

Tablo Semantik pada Argumen

Perhatikan tahap tahap pembuktiannya

Langkah 1:

Membuat variabel proposional sbb:

A= Badu mencontek saat ujian

B= Dosen akan datang

C= Pengawas tidak lalai

Langkah 2:

Menyusunnya menjadi ekspresi logika.

(1)	$A \rightarrow (\neg C \rightarrow B)$	(premis)
(2)	$A \rightarrow \neg C$	(premis)
(3)	$A \rightarrow B$	(kesimpulan)

Jika ditulis akan menjadi sbb:

$\{A \rightarrow (\neg C \rightarrow B), A \rightarrow \neg C\} \models A \rightarrow B$

Tablo Semantik pada Argumen

Langkah 3:

Menyusunnya menjadi deretan untuk dibuat tablo dengan menegasi kesimpulan menjadi $\neg(A \rightarrow B)$ sehingga penulisannya menjadi sbb:

$(A \rightarrow (\neg C \rightarrow B)) \wedge (A \rightarrow \neg C) \wedge \neg(A \rightarrow B)$

Selanjutnya susun menjadi urutan sbb:

(1) $A \rightarrow (\neg C \rightarrow B)$

(2) $A \rightarrow \neg C$

(3) $\neg(A \rightarrow B)$

Tablo Semantik pada Argumen

Langkah 4:

Buatlah tablonya seperti berikut (Jangan lupa ikuti heuristik pembuatan tablo untuk mengefisienkan pencabangan tablo)

(1) $A \rightarrow (\neg C \rightarrow B)$

(2) $A \rightarrow \neg C$

(3) $\neg(A \rightarrow B)$

(4) $\neg\neg A$

$\neg B$

(5) A

(6) $\neg A$ — Tutup

$\neg C$

(7) $\neg A$ — tutup

$\neg C \rightarrow B$

(8) $\neg\neg C$

C — tutup

B — tutup

Tablo Semantik pada Argumen

Seluruh tablo ternyata tertutup, dan ini berarti terjadi ketidak konsistenan pada seluruh argumen.

Dapat disimpulkan :\

Dengan pemberian negasi dari kesimpulan jika premis premis benar, maka negasi dari kesimpulan tidak benar, dan sebenarnya kesimpulannya benar sehingga argumen dianggap valid.

THE END