PROGRAM STUDI TEKNIK INFORMATIKA
Sekolah Teknik Elrektro dan Informatika

INSTITUT TEKNOLOGI BANDUNG

Pengantar Matematika Diskrit

Bahan Kuliah IF2120 Matematika Diksrit

RINALDI MUNIR

Lab Ilmu dan Rekayasa Komputasi
Kelompok Keahlian Informatika

Institut Teknologi Bandung

Kampus ITB yang indah…

Foto oleh Eko Purwono (AR ITB)

Inilah STEI-ITB…

LabTek V, di sini Informatika ITB berada

Salah satu mata kuliahnya....

IF2120 Matematika Diskrit

Sumber gambar: http://www.zazzle.com/i_can_be_functionally_discrete_or_continuous_tshirt-235341012435015470

Rasa ingin tahu adalah ibu dari semua ilmu pengetahuan

Tak kenal maka tak sayang, tak sayang maka tak cinta

Perjalanan satu mil dimulai dari satu langkah

Apakah Matematika Diskrit itu?

- Apa yang dimaksud dengan kata **diskrit** (*discrete*)?
- Objek disebut diskrit jika:

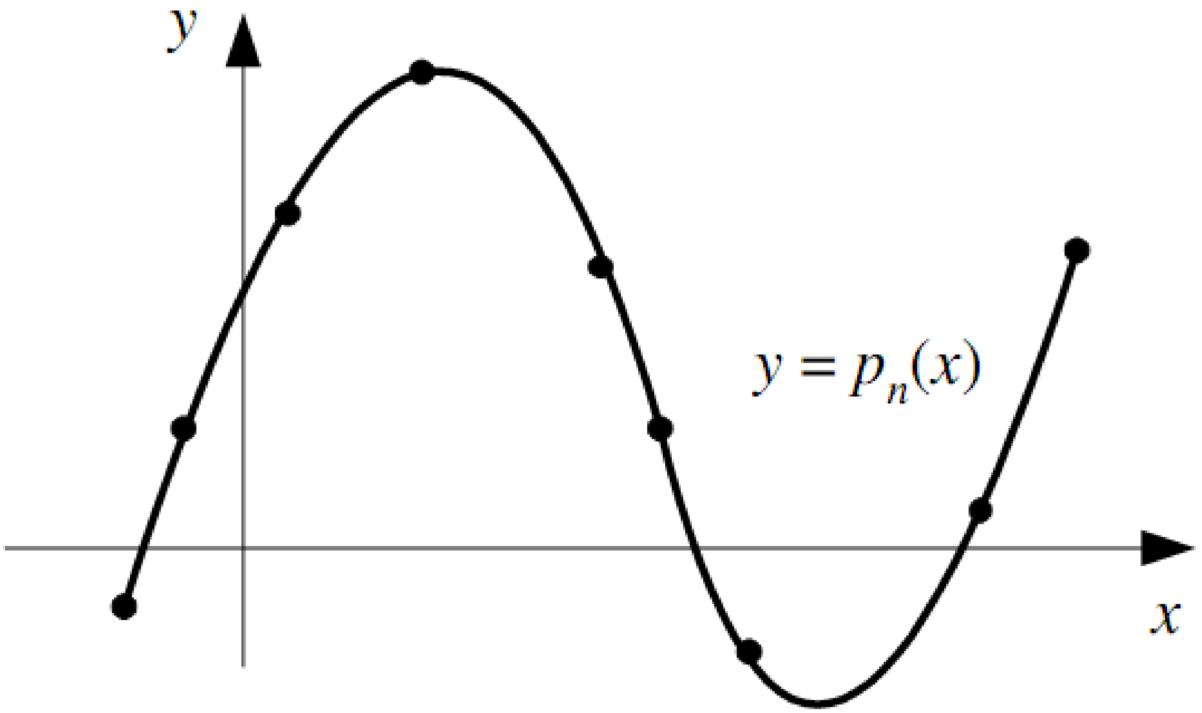
  \- terdiri dari elemen yang berbeda (*distinct*) dan terpisah secara individual, atau

  \- elemen-elemennya tidak bersambungan (*unconnected*).

  Contoh: himpunan bilangan bulat (*integer*)
- Lawan kata diskrit: **kontinyu** atau **menerus** (*continuous*).

  Contoh: himpunan bilangan riil (*real*)

Diskrit versus kontinu



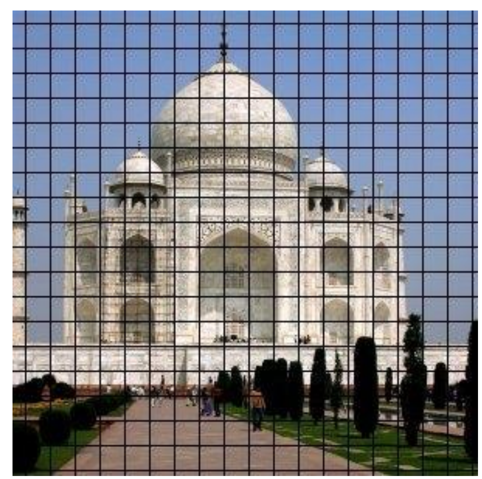
Kurva mulus: himpunan menerus
Titik-titik tebal di kurva: himpunan diskrit

- Matematika Diskrit: cabang matematika yang mengkaji objek-objek yang nilainya berbeda (*distinct*) dan terpisah (*separate*) satu sama lain.

- Lawannya: Matematika Menerus (*continuous mathematics*), yaitu cabang matematika dengan objek yang sangat mulus (*smoothy*), termasuk di dalamnya *calculus*.

- Komputer digital bekerja secara diskrit. Informasi yang disimpan dan dimanipulasi oleh komputer adalah dalam bentuk diskrit.

- Kamera digital menangkap gambar (analog) lalu direpresentasikan dalam bentuk diskrit berupa kumpulan *pixel* atau *grid.* Setiap *pixel* adalah elemen diskrit dari sebuah gambar



Topik bahasan di dalam Matematika Diskrit:

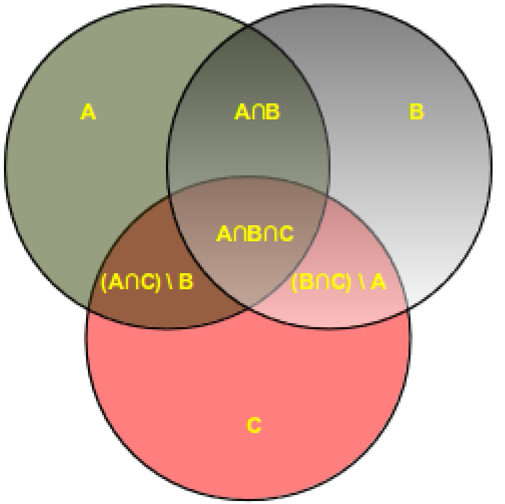
- Logika (*logic*) dan penalaran → Pindah ke kuliah Logika Komputasional
- Teori Himpunan (*set*) √
- Relasi dan Fungsi (*relation and function*) √
- Induksi Matematik (*mathematical induction*) √
- Algoritma (*algorithms*) √ → sebagian
- Teori Bilangan Bulat (*integers*) √
- Barisan dan Deret (*sequences and series*) → kuliah Kalkulus
- Teori Grup dan *Ring* (*group and ring*) → advance
- Aljabar Boolean (*Boolean algebra*) √
- Kombinatorial (*combinatorics*) √
- Teori Peluang Diskrit (*discrete probability*) → ke kuliah Probstat
- Fungsi Pembangkit dan Analisis Rekurens →ke kuliah Modsim
- Teori Graf √
- Pohon √
- Kompleksitas Algoritma (*algorithm complexity*) √
- Otomata → ke kuliah TBO
- Relasi Rekurens √

1. Logika

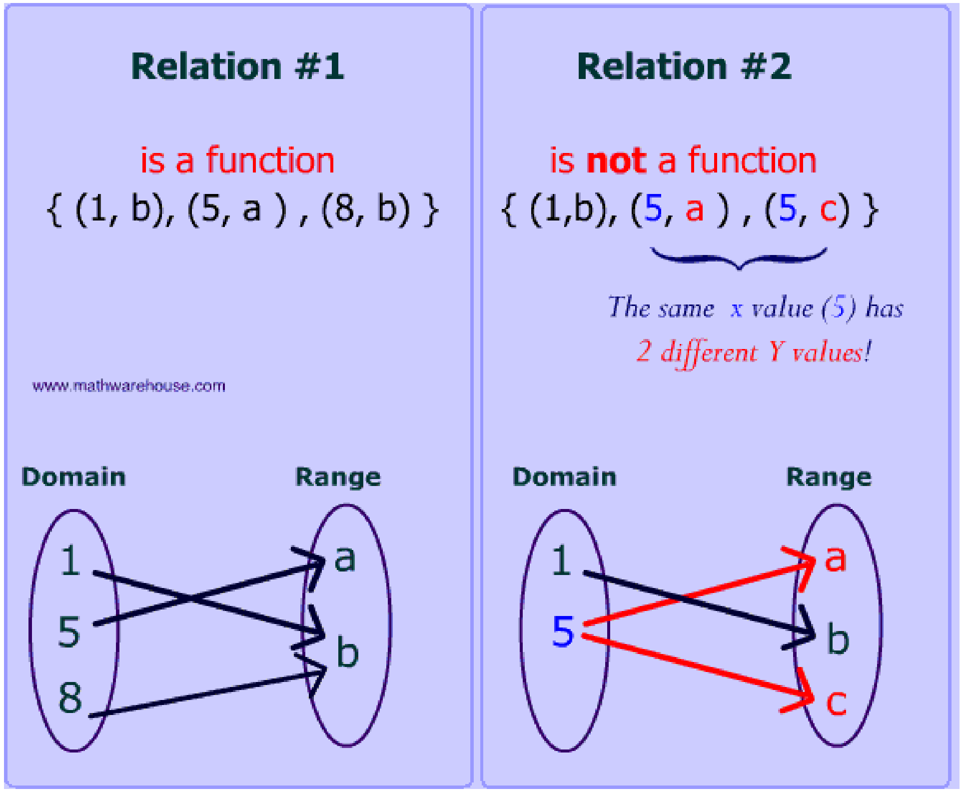
Basic statement	Equivalent
$p \vee q$	$q \vee p$
$p \wedge q$	$q \wedge p$
$\neg(p \wedge q)$	$\neg p \vee \neg q$
$\neg(p \vee q)$	$\neg p \wedge \neg q$
$p \rightarrow q$	$\neg p \vee q$
	$\neg q \rightarrow \neg p$
$p \leftrightarrow q$	$(p \rightarrow q) \wedge (q \rightarrow p)$
	$(\neg p \vee q) \wedge (\neg q \vee p)$
$p \wedge (q \wedge r)$	$(p \wedge q) \wedge r$
$p \vee (q \vee r)$	$(p \vee q) \vee r$
$p \wedge (q \vee r)$	$(p \wedge q) \vee (p \wedge r)$
$p \vee (q \wedge r)$	$(p \vee q) \wedge (p \vee r)$
$p \rightarrow (q \vee r)$	$(p \wedge \neg q) \rightarrow r$

2. Teori Himpunan

1). if $A \subset B$ and $B \subset C$, then $A \subset C$ (transitivity),
2). if $A \subset B$ and $B \subset A$, then $A = B$,
3). $A \cup A = A$,
4). $A \cup \varnothing = A$,
5). $A \cap A = A$,
6). $A \cap \varnothing = \varnothing$,
7). $A - A = \varnothing$,
8). $A \cup B = B \cup A$ (commutability of addition),
9). $A \cap B = B \cap A$ (commutability of multiplication),
10). $(A \cup B) \cup C = A \cup (B \cup C)$ (associativity of addition),
11). $(A \cap B) \cap C = A \cap (B \cap C)$ (associativity of multiplication),
12). $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$ (distributivity of multiplication over addition),
13). $A \cap (B - C) = (A \cap B) - (A \cap C)$ (distributivity of multiplication over subtraction),
14). $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$ (distributivity of addition over multiplication).

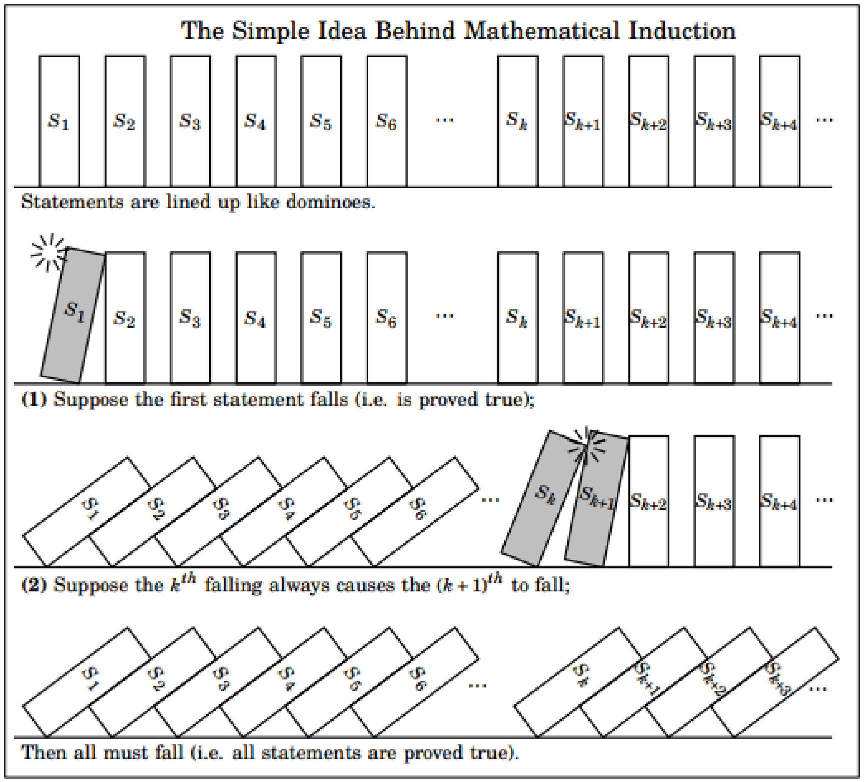


3. Relasi dan Fungsi



Sumber: www.mathwarehouse.com

4. Induksi Matematik



prove by mathematical induction

$$1.2.3 + 2.3.4 + \ldots + n(n+1)(n+2) = \frac{n(n+1)\,(n+2)\,(n+3)}{4}$$

let P(n): $1.2.3 + 2.3.4 + \ldots + n(n+1)(n+2) = \frac{n(n+1)\,(n+2)\,(n+3)}{4}$

for n=1, L.H.S = 1.2.3 = 6, R.H.S $= \frac{1.(1+1).(1+2).(1+3)}{4} = 6$

$\therefore$ *P(1) is true.*

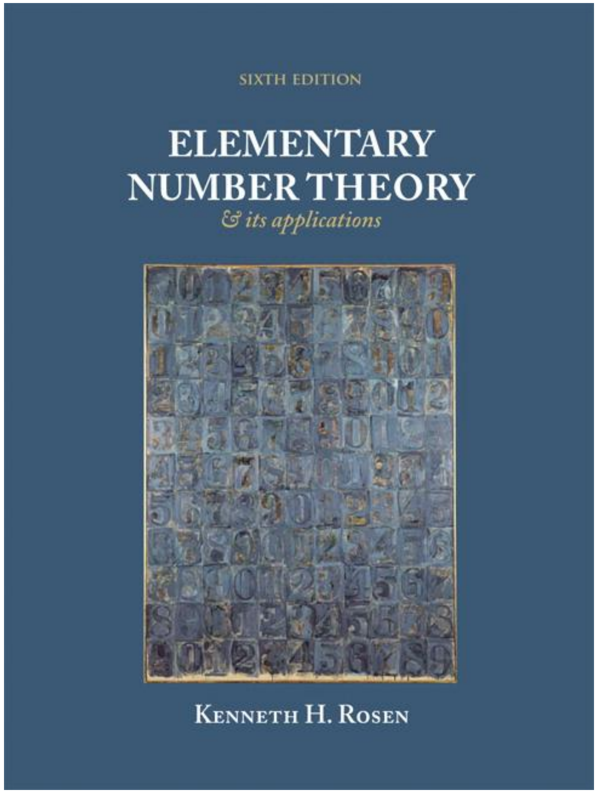
Assume P(k) is true

[Sumber gambar: math.stackexchange.com]()

5. Teori Bilangan

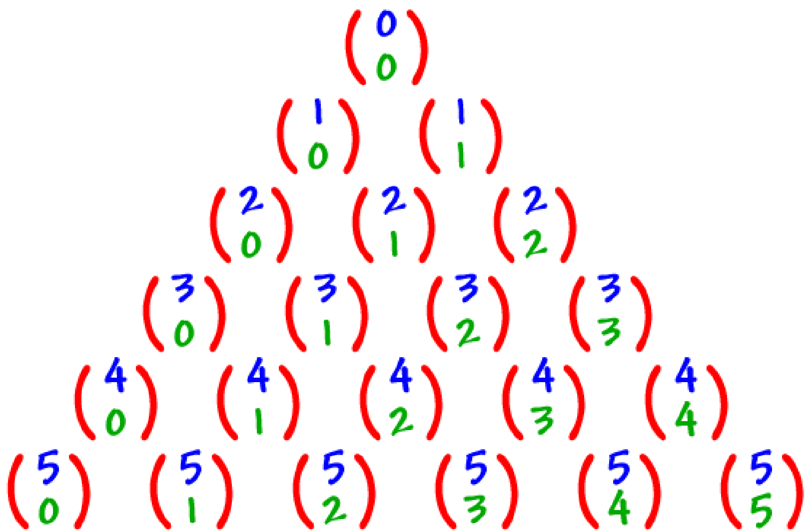
$$
\begin{aligned}
&\begin{cases} N \equiv 4 \pmod{7} \\ N \equiv 6 \pmod{11} \end{cases} \\
&N = 7k+4 \\
&7k+4 \equiv 6 \pmod{11} \\
&7k \equiv 2 \pmod{11} \\
&-21k \equiv -6 \pmod{11} \\
&k \equiv -6 \pmod{11} \\
&k = 11m-6 \\
&N = 77m-38 \\
&1000 < 77m-38 < 2000 \Rightarrow 13 < m < 27 \\
&77m-38 \equiv -m+1 \equiv 27-m \pmod{13}
\end{aligned}
$$

Sumber: mymathforum.com



Sumber: www.pearsonhighered.com

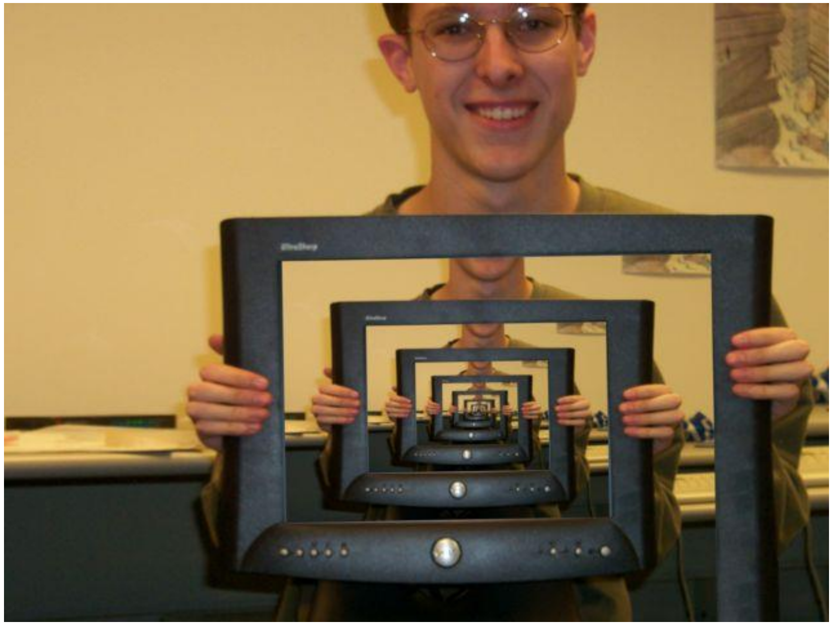
6. Kombinatorial



Sumber: www.coolmath.com

Sumber: ronsden.com

7. Rekursif dan relasi rekurens

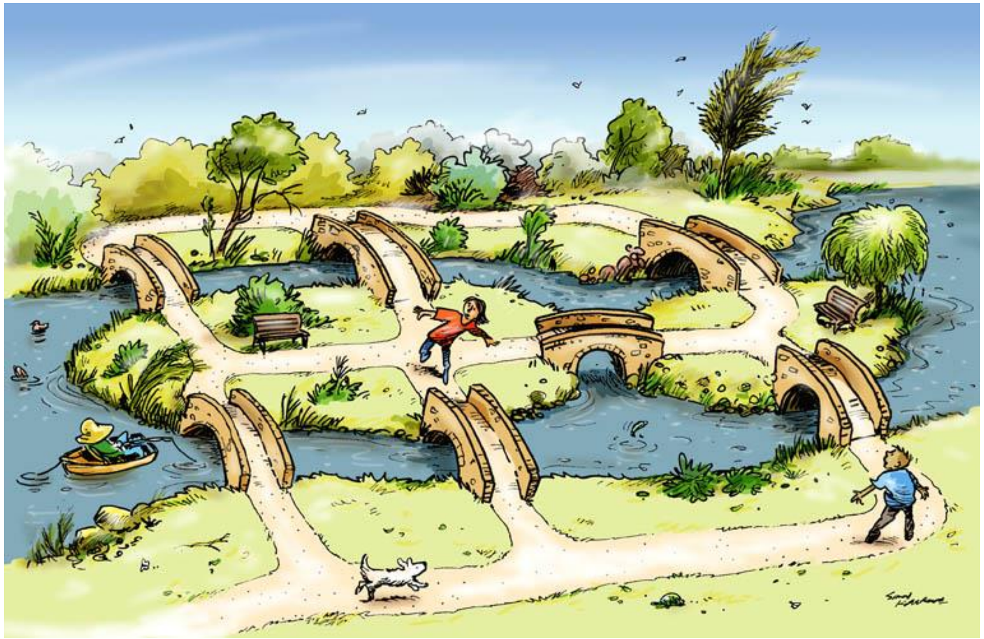


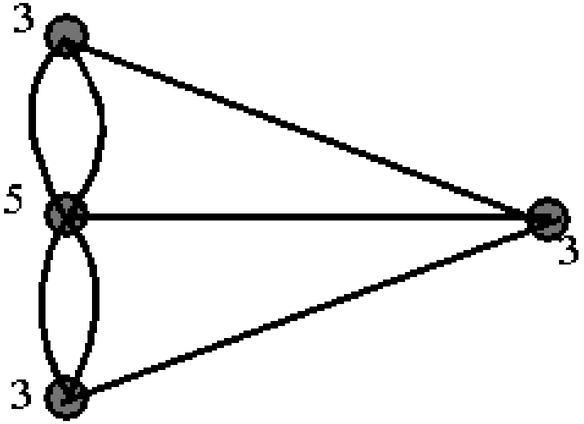
Sumber: www.ilxor.com

$$
\begin{aligned}
g(n) &= 4g(n-1)+4 \\
&= 4(4g(n-2)+4)+4 \\
&= 4^2 g(n-2)+4^2+4 \\
&= 4^2(4g(n-3)+4)+4^2+4 \\
&= 4^3 g(n-3)+4^3+4^2+4 \\
&= \vdots \\
&= 4^n g(0)+4^n+4^{n-1}+\cdots+4^3+4^2+4 \\
&= 4^n+4^{n-1}+\cdots+4^3+4^2+4 \\
&= 4\left(\frac{4^n-1}{3}\right) \\
&= \frac{4^{n+1}-4}{3}
\end{aligned}
$$

Sumber: cas.bethel.edu

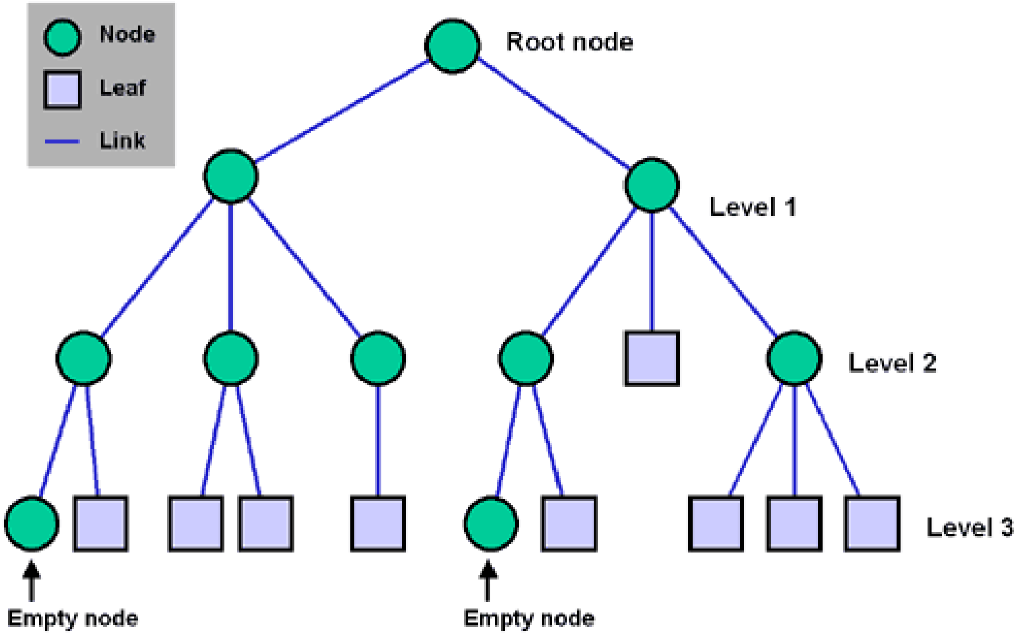
8. Teori Graf

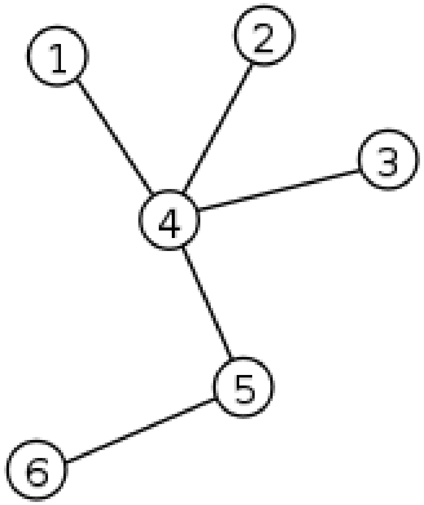




Sumber: simonkneebone.com

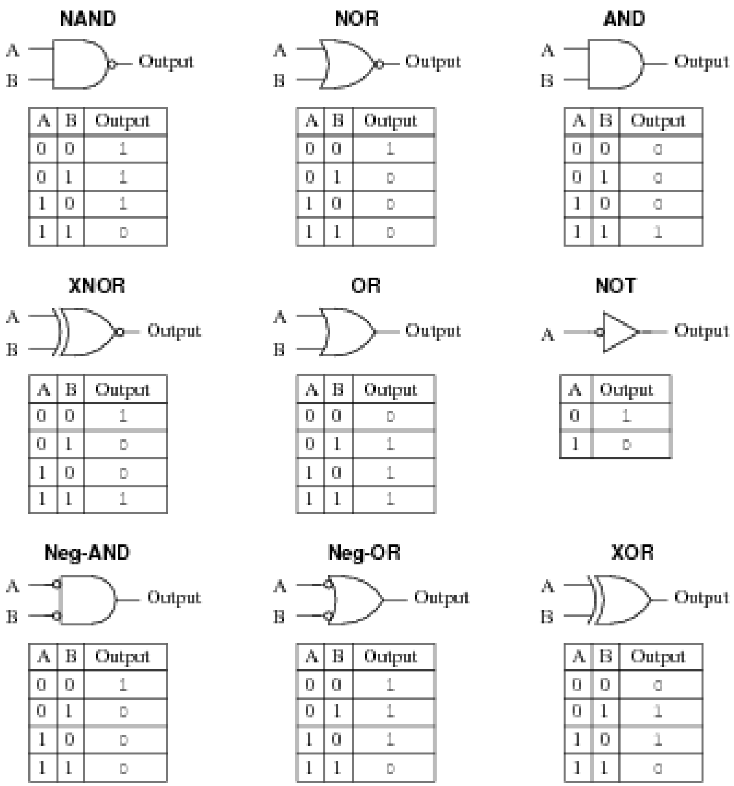
9. Pohon





Sumber: ubuntuforums.org

10. Aljabar Boolean



NAND

A	B	Output
0	0	1
0	1	1
1	0	1
1	1	0

NOR

A	B	Output
0	0	1
0	1	0
1	0	0
1	1	0

AND

A	B	Output
0	0	0
0	1	0
1	0	0
1	1	1

XNOR

A	B	Output
0	0	1
0	1	0
1	0	0
1	1	1

OR

A	B	Output
0	0	0
0	1	1
1	0	1
1	1	1

NOT

A	Output
0	1
1	0

Neg-AND

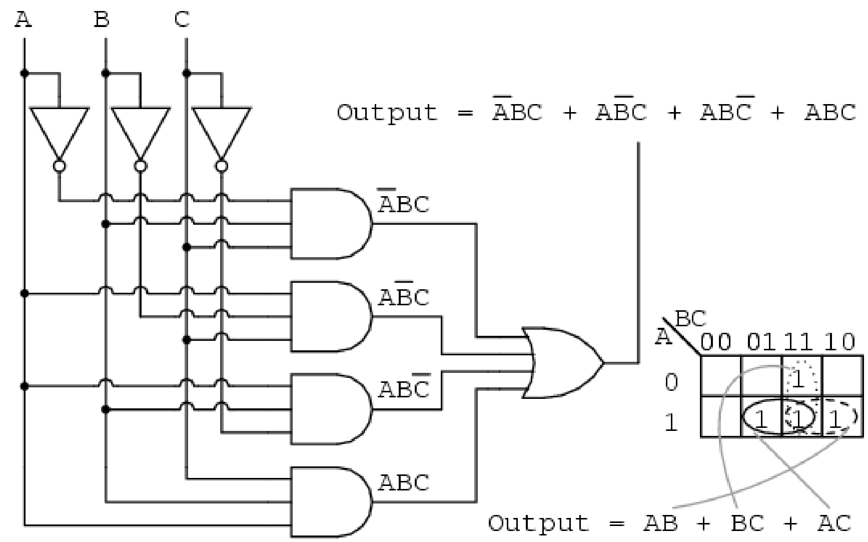
A	B	Output
0	0	1
0	1	0
1	0	0
1	1	0

Neg-OR

A	B	Output
0	0	1
0	1	1
1	0	1
1	1	0

XOR

A	B	Output
0	0	0
0	1	1
1	0	1
1	1	0

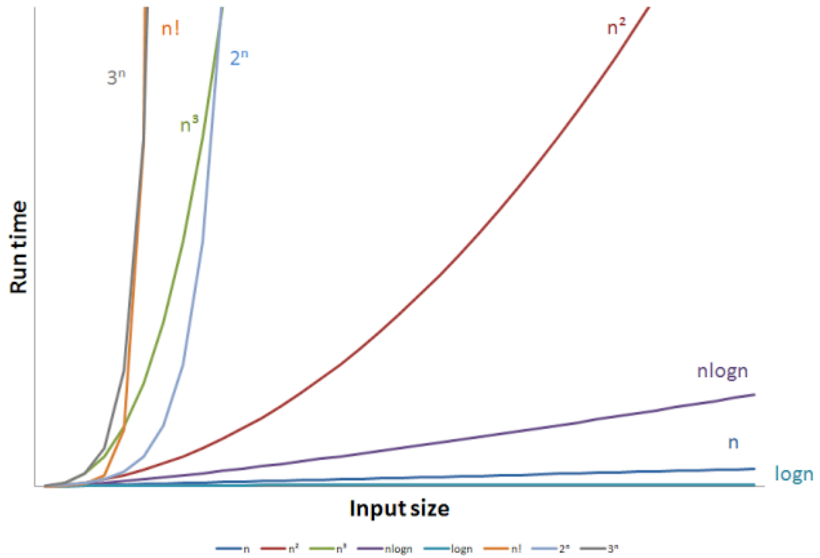


Sumber: www.ibibilio.org

Sumber: www.allaboutcircuits.com

11. Kompleksitas Algoritma

$T(n)$	Name	Problems
$O(1)$	constant	easy-solved
$O(logn)$	logarithmic	
$O(n)$	linear	
$O(nlogn)$	linear-logarithmic	
$O(n^2)$	quadratic	
$O(n^3)$	cubic	
$O(2^n)$	exponential	hard-solved
$O(n!)$	factorial	



Sumber: agafonovslava.com

Sumber: blog.philenotfound.com

Contoh-contoh persoalan di dalam Matematika Diskrit:

- Berapa banyak kemungkinan jumlah *password* yang dapat dibuat dari 8 karakter?

- ISBN sebuah buku adalah 978-602-6232-42-7. Verifkasilah apakah nomor ISBN tersebut valid?

- Berapa banyak *string* biner yang panjangnya 8 bit yang mempunyai bit 1 sejumlah ganjil?

- Bagaimana menentukan lintasan terpendek dari satu kota *a* ke kota *b*?

- Buktikan bahwa perangko senilai n ($n \geq 8$) rupiah dapat menggunakan hanya perangko 3 rupiah dan 5 rupiah saja

- Diberikan dua buah algoritma untuk menyelesaian sebuah persoalan, algoritma mana yang terbaik?

- Bagaimana rangkaian logika untuk membuat peraga digital yang disusun oleh 7 buah batang (*bar*)?

- Dapatkah kita melalui semua jalan di sebuah kompleks perumahan tepat hanya sekali dan kembali lagi ke tempat semula?

- "Makanan murah tidak enak", "makanan enak tidak murah". Apakah kedua pernyataan tersebut menyatakan hal yang sama?

Mengapa Mempelajari Matematika Diskrit?

Ada beberapa alasan:

1. Mengajarkan mahasiswa untuk berpikir secara matematis

   → mengerti argumen matematika

   → mampu membuat argumen matematika.

Contoh: Jumlah derajat semua simpul pada suatu graf adalah genap, yaitu dua kali jumlah sisi pada graf tersebut. Akibatnya, untuk sembarang graf G, banyaknya simpul berderajat ganjil selau genap.

2. Mempelajari fakta-fakta matematika dan cara menerapkannya.

Contoh: *(Chinese Remainder Problem)* Pada abad pertama, seorang matematikawan China yang bernama Sun Tse mengajukan pertanyaan sebagai berikut:

Tentukan sebuah bilangan bulat yang bila dibagi dengan 5 menyisakan 3, bila dibagi 7 menyisakan 5, dan bila dibagi 11 menyisakan 7.

3. Matematika diskrit memberikan landasan matematis untuk kuliah-kuliah lain di informatika.

→ algoritma, struktur data, basis data, otomata dan teori bahasa formal, jaringan komputer, keamanan komputer, sistem operasi, teknik kompilasi, dsb.

- Matematika diskrit adalah matematika yang khas informatika

→ **Matematika-nya orang Informatika**!

Lima pokok kuliah di dalam Matematika Diskrit

1. Penalaran matematika (*Mathematical reasoning*)
   Mampu membaca dan membentuk argumen matematika
   (Materi: logika)

2. Analisis kombinatorial (*Combinatorial analysis*)
   Mampu menghitung atau mengenumerasi objek-objek
   (materi: kombinatorial → permutasi, kombinasi, dll)

3. Sruktur diskrit

   Mampu bekerja dengan struktur diskrit. Yang termasuk struktur diskrit: Himpunan, Relasi, Permutasi dan kombinasi, Graf, Pohon, *Finite-state machine*

4. Berpikir algoritmik

   Mampu memecahkan persoalan dengan menspesifikasikan algoritmanya

   (Materi: pada sebagian besar kuliah ini dan kuliah Algoritma dan Struktur Data)

5. Aplikasi dan pemodelan

   Mampu mengaplikasikan matematika diskrit pada hampir setiap area bdiang studi, dan mampu memodelkan persoalan dalam rangka *problem-solving skill*.

   (Materi: pada sebagian besar kuliah ini)

Kemana lanjutan kuliah ini?

- Matematika Diskrit memberikan dasar untuk banyak mata kuliah mata kuliah:
    1. IF2122 Probabilitas dan Statistik

        Materi: Kombinatorial
    2. IF2130 Organisasi dan Arsitektur Komputer

        Materi: Aljabar Boolean
    3. IF2220 Teori Bahasa dan Otomata

        Materi: Himpunan, Graf
    4. IF2240 Basisdata

        Materi: Relasi dan Fungsi

5. IF2211 Strategi Algoritma

Materi: Graf, pohon, kombinatorial, kompleksitas algoritma

6. IF3130 Jaringan Komputer

Materi: Graf, pohon

7. IF2110 Algoritma dan Struktur Data

Materi: Graf, pohon, kompleksitas algoritma

8. IF4020 Kriptografi

Materi: Teori Bilangan, Kombinatorial

9. Dan masih banyak kuliah lainnya

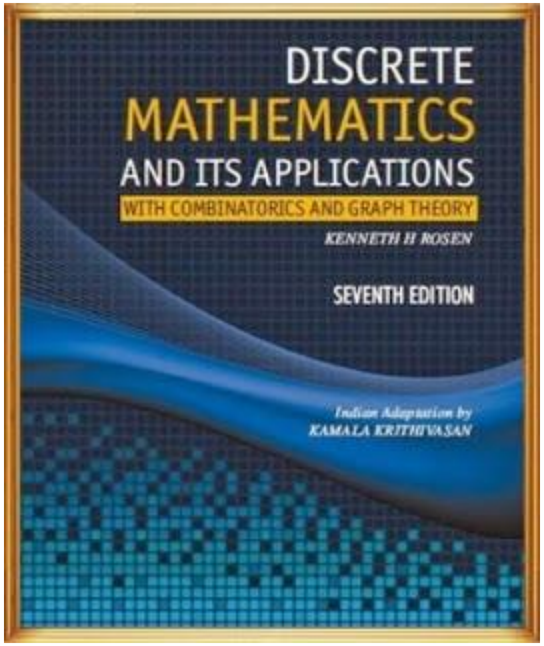
Moral of this story…

- Mahasiswa informatika harus memiliki pemahaman yang kuat dalam Matematika Diskrit, agar tidak mendapat kesulitan dalam memahami kuliah-kuliah lainnya di informatika.

Referensi Kuliah

Utama:

1. Kenneth H. Rosen, *Discrete Mathematics and Application to Computer Science 7th Edition*, Mc Graw-Hill.



2. Rinaldi Munir, *Diktat kuliah Matematika Diskrit (Edisi Keempat*), Teknik Informatika ITB, 2003. (juga diterbitkan dalam bentuk buku oleh Penerbit Informatika)

Pendukung:

1. Susanna S. Epp, *Discrete Mathematics with Application*, 4th Edition, Brooks/Cle, 2010
2. Peter Grossman, *Discrete Mathematics for Computing*, 2nd edition, Palgrave MacMillan, 2002
3. C.L. Liu, *Element of Discrete Mathematics*, McGraw-Hill, Inc, 1985.
4. Richard Johsonbaugh, *Discrete Mathematics*, Prentice-Hall, 1997.

URL

- Informasi perkuliahan (bahan kuliah, bahan ujian, soal kuis tahun2 sebelumnya, pengumuman, dll), bisa diakses di:

  http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/matdis.htm

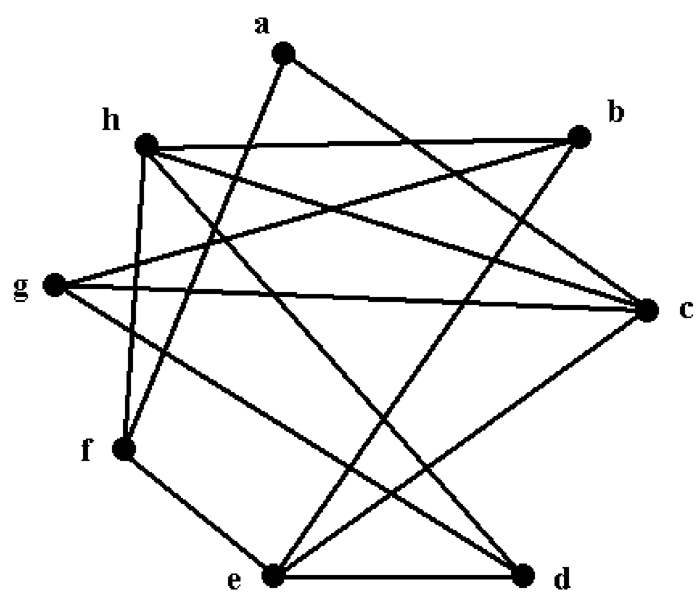
atau masuk dari:

  http://informatika.stei.itb.ac.id/~rinaldi.munir/

Contoh-contoh soal Kuis/UTS/UAS

1. Misalkan ada sejumlah n ganjil orang ($n > 1$) yang berkumpul di sebuah lapangan, di sini mereka masing-masing memegang sebuah kue *pie* yang siap dilemparkan ke orang lain yang paling dekat dengannya. Jarak antar orang berbeda (tidak ada jarak antar pasangan yang sama). Jika semua orang harus melempar kue dengan simultan(bersamaan), buktikan dengan induksi matematik bahwa minimal ada satu orang yang tidak terkena lemparan kue.

2. Tunjukkan bahwa graf G berikut ini tidak planar. Buktikan dengan menggunakan :
   a) Teorema Kuratowski
   b) Ketidaksamaan Euler



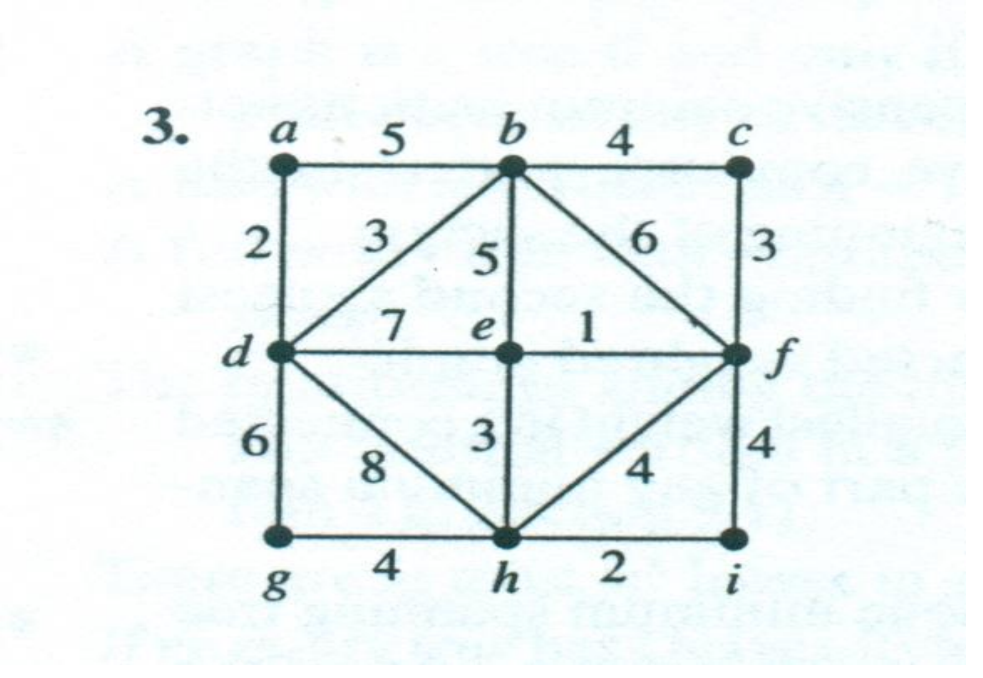
3. Diketahui *R* suatu relasi pada himpunan bilangan bulat sehingga *a R b* jika dan hanya jika *a* dan *b* keduanya negatif atau keduanya positif. Buktikan apakah *R* adalah relasi kesetaraan (ekuivalen).

4. Berapa banyak solusi bilangan bulat dari

$$x_1 + x_2 + x_3 + x_4 + x_5 = 21$$

jika $0 \leq x_1 \leq 10$?

5. Carilah pohon merentang minimum dari graf dibawah ini dengan menggunakan Algoritma Kruskal, serta tuliskan setiap langkahnya.



6. Misalkan terdapat string: “RAJA PDJAJARAN”

a. Gambarkan pohon Huffman dengan terlebih dulu menghitung frekuensi kemunculan tiap karakternya (termasuk spasi)

b. Tentukan kode Huffman untuk masing-masing karakter dalam bentuk tabel lalu hitung panjang rangkaian bit yang dihasilkan jika string di atas diubah menjadi kode huffman yang telah dibuat

c. Tentukan kata yang terbentuk dari rangkaian bit 10010001 dengan proses decoding menggunakan kode Huffman diatas (jika tidak ada cukup tulis “tidak ada”)