HIMPUNAN MAHASISWA TEKNIK INFORMATIKA
BINUS UNIVERSITY

HIMTI KIT

DISCRETE MATHEMATICS

2024

DAFTAR ISI

Introduction to Discrite Math

1. Apa itu Discrite Math?

Discrete Math adalah cabang matematika yang mempelajari struktur diskret, yaitu objek yang terpisah dan tidak berhubungan secara kontinu. Contoh-contoh dari objek diskret termasuk bilangan bulat, graf, dan pernyataan logika. Berbeda dengan matematika kontinu, seperti kalkulus, yang berurusan dengan perubahan dan aliran, Discrete Math lebih berfokus pada objek yang terhitung, seperti jumlah, urutan, dan pengelompokan.

Fun Fact: Algoritma enkripsi yang digunakan untuk mengamankan data dalam komunikasi digital, seperti email dan transaksi online, didasarkan pada konsep-konsep Discrete Math!

2. Manfaat Discrite Math

Menurut Kalian, Discrite Math itu manfaatnya apa aja yaa?? Kalian bisa isi jawaban kalian pada kolom ini

---

Manfaat Discrete Math Discrete Math sangat penting dalam bidang ilmu komputer dan teknologi informasi. Beberapa manfaat utama dari Discrete Math adalah:

- Pemecahan Masalah Logika dan Algoritma: Membantu dalam menyusun algoritma yang efisien dan efektif.
- Pengembangan Perangkat Lunak dan Hardware: Digunakan dalam merancang sistem perangkat lunak dan perangkat keras.
- Pemahaman Struktur Data: Membantu dalam memahami dan mengoptimalkan struktur data yang digunakan dalam pemrograman.

Sederhananya 3 poin saja ya kawan, **SEMUA BUTUH PRAKTEK**. Kenapa? Untuk kamu bisa lebih memahami lagi apa itu Discrite Math baik secara Teori dan Implementasinya.

Logic

Penjelasan dan Contoh

Logika adalah studi tentang prinsip-prinsip validitas dalam penalaran. Dalam Discrete Math, logika digunakan untuk memahami dan merumuskan pernyataan yang benar atau salah. Pada dasarnya, logika dalam Discrete Math melibatkan penggunaan proposisi, operator logika, dan hukum-hukum logika. Mari kita bahas ini lebih lanjut.

Proposisi

Proposisi adalah pernyataan yang dapat bernilai benar (true) atau salah (false), tetapi tidak keduanya. Contoh dari proposisi adalah:

- "2 adalah bilangan genap." (Benar)
- "5 lebih besar dari 10." (Salah)

Operator Logika

Operator logika digunakan untuk menghubungkan proposisi dan membentuk proposisi baru. Beberapa operator logika dasar adalah:

1. AND (∧)
2. OR (∨)
3. NOT (¬)
4. IMPLIES (→)
5. IF AND ONLY IF (↔)

.

Tabel Kebenaran :

A	B	¬A	¬B	A ∧ B	A ∨ B	A → B	A ↔ B
T	T	F	F	T	T	T	T
T	F	F	T	F	T	F	F
F	T	T	F	F	T	T	F
F	F	T	T	F	F	T	T

Penjelasan Tabel Kebenaran

- AND (∧): Pernyataan A ∧ B hanya benar jika kedua A dan B benar.
- OR (∨): Pernyataan A ∨ B benar jika salah satu atau kedua pernyataan A dan B benar.
- NOT (¬): Pernyataan ¬A benar jika A salah, dan salah jika A benar.
- IMPLIES (→): Pernyataan A → B benar kecuali jika A benar dan B salah.
- IF AND ONLY IF (↔): Pernyataan A ↔ B benar jika A dan B memiliki nilai kebenaran yang sama.

Hukum-Hukum Logika

Hukum-hukum logika adalah aturan yang digunakan untuk memanipulasi proposisi. Beberapa hukum logika yang penting meliputi:

1. Hukum Identitas
    - A∧A=AA \land A = AA∧A=A
    - A∨A=AA \lor A = AA∨A=A
2. Hukum Negasi
    - A∧¬A=FalseA \land \neg A = \text{False}A∧¬A=False
    - A∨¬A=TrueA \lor \neg A = \text{True}A∨¬A=True
3. Hukum Idempotensi
    - A∧True=AA \land \text{True} = AA∧True=A
    - A∨False=AA \lor \text{False} = AA∨False=A
4. Hukum Distributif
    - A∧(B∨C)=(A∧B)∨(A∧C)A \land (B \lor C) = (A \land B) \lor (A \land C)A∧(B∨C)=(A∧B)∨(A∧C)
    - A∨(B∧C)=(A∨B)∧(A∨C)A \lor (B \land C) = (A \lor B) \land (A \lor C)A∨(B∧C)=(A∨B)∧(A∨C)

Quantifier

Penjelasan dan Contoh

Predikat

Predikat adalah kalimat terbuka yang berisi sejumlah variabel dan akan menjadi
proposisi jika variabelnya diberi nilai tertentu.

Contoh :

$P(x, 2)$ = "$x > 2$" → Predikat, bukan proposisi

$P(3,4)$ = "3 < 4" → Proposisi, bukan predikat

- Mencari nilai kebenaran dari predikat

Contoh : Misal $P(x, y)$ menyatakan persamaan P = "$xy > x + y$" . Apa
nilai kebenaran dari proposisi $P(2,3)$ *dan* $P(1,4)$?

$P(2,3) = 2 \times 3 > 2 + 3$ → Benar

$P(1,4) = 1 \times 4 > 1 + 4$ → Sala

Kuantor

Kuantor adalah simbol yang digunakan untuk menyatakan kuantitas suatu objek dalam proposisi.

1. Kuantor Universal / Umum

Simbol : $\forall$

Definisi : Setiap / semua

Persamaan : $\forall x \epsilon B, M(x)$ atau $\forall x \epsilon B \to M(x)$

Contoh : “Setiap binatang pasti akan mati”

Dimana : B = himpunan semua binatang; M(x) = predikat “mati”

2. Proposisi Universal : "$\forall x \epsilon D, Q(x)$"
Definisi : Proposisi dinyatakan benar, jika dan hanya jika, Q(x) benar untuk
setiap x anggota domain (D).
Proposisi dinyatakan salah, jika dan hanya jika, Q(x) salah untuk setidaknya
satu x dalam D.
Dimana : D = domain dari x; Q(x) = predikat
Note : Nilai x yang Q(x)-nya salah disebut sebagai counterexample atau
contoh counter proposisi universal.
Contoh : D = {1, 2, 3, 4, 5} dan proposisi "$\forall x \epsilon D, x2 \geq x$". Tunjukkan bahwa
proposisi tersebut benar.
12 ≥ 1; 22 ≥ 2; 32 ≥ 3; 42 ≥ 4; 52 ≥ 5. Proposisi "$\forall x \epsilon D, x2 \geq x$" benar.

3. Kuantor Eksistensial / Khusus

Simbol : $\exists$

Definisi : Ada / terdapat

Persamaan : $\exists x \epsilon B, M(x)$

Contoh : “Ada seekor nyamuk yang mati”

Dimana : x = nyamuk; B = himpunan semua binatang; M(x) = predikat “mati”

4. Proposisi Eksistensial : "$\exists x \epsilon D$ sedemikian rupa sehingga $M(x)$"
Definisi : Proposisi dinyatakan benar, jika dan hanya jika M(x) benar untuk
setidaknya satu x dalam D.
Proposisi dinyatakan salah, jika dan hanya jika M(x) salah untuk semua x
dalam D.
Contoh : "$\exists t \epsilon A$ sedemikian rupa sehingga $t2 = t$". Tunjukkan bahwa
proposisi tersebut benar untuk A adalah bilangan bulat.
Jika t = 1, maka 12 = 1. Jadi "$t2 = t$" adalah benar setidaknya untuk satu
bilangan bulat t.
5. Proposisi Universal Bersyarat : "$\forall x, jika\ P(x)\ maka\ Q(x)$ atau $\forall x, P(x) \rightarrow$
$Q(x)$"
Contoh : $\forall x \epsilon R$, jika $x > 2$ maka $x2 > 4$
6. Negasi Proposisi Universal : "~$(x \epsilon D, Q(x)) \equiv \exists x \epsilon D$ sedemikian rupa
sehingga ~$Q(x)$"
Contoh : Proposisi “∀ bilangan prima p, p adalah

5. Proposisi Universal Bersyarat : "$\forall x$, *jika* $P(x)$ *maka* $Q(x)$
atau $\forall x, P(x) \rightarrow$
$Q(x)$"
Contoh : $\forall x \epsilon R$, jika $x > 2$ maka $x2 > 4$

6. Negasi Proposisi Universal : "~($x \epsilon D$, $Q(x)$) $\equiv$ $\exists x \epsilon D$
sedemikian rupa
sehingga ~$Q(x)$"
Contoh : Proposisi “∀ bilangan prima p, p adalah ganjil”
Maka negasinya adalah “∃ sebuah bilangan prima p sedemikian rupa sehingga
p tidak ganjil”

7. Negasi Proposisi Eksistensial : "~($\exists x \epsilon D$ sedemikian rupa sehingga $Q(x)$) ≡
$\forall x \epsilon D$, ~$Q(x)$"
Contoh : Proposisi “∃ sebuah segitiga T yang jumlah sudutnya sama dengan
200°”
Maka negasinya adalah “∀ segitiga T yang jumlah sudutnya tidak sama
dengan 200°”

8. Negasi Pernyataan Universal Bersyarat : "~($\forall x$, jika P(x) maka Q(x)) ≡ $\exists x$
sedemikian rupa sehingga P(x) dan ~Q(x)

Nested Quantifier

Kombinasi pasangan dua kuantor :

1. Kombinasi umum-umum : $\forall x \forall y\, P(x, y)$

P(x,y) benar untuk semua pasangan x,y.

2. Kombinasi khusus-khusus : $\exists x \exists y\, P(x, y)$

P(x,y) benar untuk sekurang-kurangnya satu pasangan x,y.

3. Kombinasi umum-khusus : $\forall x \exists y\, P(x, y)$

Untuk setiap nilai x dapat dicarikan sebuah nilai y sedemikian rupa sehingga
P(x,y) benar.

4. Kombinasi khusus-umum : $\exists x \forall y\, P(x, y)$

Terdapat setidaknya sebuah nilai x yang membuat P(x,y) selalu benar.

Method of Proof

Method of Proof adalah cara-cara yang digunakan untuk membuktikan kebenaran atau ketidakbenaran pernyataan matematika. Beberapa metode umum termasuk:

A. Metode Pembuktian Langsung (Direct Proof)
Metode pembuktian langsung dilakukan dengan menguraikan premis dengan dilandasi
oleh definisi, fakta, aksioma yang ada untuk sampai pada suatu kesimpulan
(konklusi).
Contoh : Jika A maka B. Jika B maka C.

● Definisi Bilangan Genap

Suatu bilangan bulat n disebut bilangan genap jika terdapat suatu bilangan
bulat k, sehingga n = 2k.

Contoh : Buktikan “Jika n adalah bilangan genap, maka $n2$ adalah bilangan
genap”.

Bukti : jika n genap, maka n = 2k, dimana k adalah bilangan bulat.

Dengan demikian, $n2 = (2k)2 = 4k2 = 2(2k2) = 2m$, dimana m = 2$k$2 dan m
= bilangan bulat.

Jadi n2 = 2m = 2 = bilangan genap → Terbukti.

● Definisi Bilangan Ganjil

Suatu bilangan bulat n disebut bilangan ganjil jika terdapat suatu bilangan
bulat k, sehingga n = 2k+1.

Contoh : Buktikan “Jika jumlah bilangan genap dan bilangan ganjil adalah
bilangan ganjil”.

Bukti : Misal m = 2p dan n = 2k+1

Maka, m+n = 2p+2k+1 = 2(p+k)+1

Misal p+k = i

Maka, 2(p+k)+1 = 2i+1 (bentuk bilangan ganjil) → Terbukti.

B. Metode Pembuktian Tidak Langsung (Indirect Proof)

Metode pembuktian tidak langsung memiliki 2 cara yaitu :

1. Kontraposisi

Pembuktian tidak langsung kontraposisi digunakan untuk membuktikan
pernyataan implikasi.

$p \rightarrow q \equiv {\sim}q \rightarrow {\sim}p$

Untuk membuktikan kebenaran $p \rightarrow q$, maka cukup membuktikan kebenaran
${\sim}q \rightarrow {\sim}p$.

2. Kontradiksi

Pembuktian tidak langsung kontradiksi dilakukan dengan mengandaikan
konklusi yang salah dan menemukan suatu hal yang bertentangan dengan
fakta, aksioma, atau teorema yang ada.

C. Induksi Matematika

Induksi matematika digunakan untuk membuktikan suatu pernyataan tertentu yang berlaku untuk bilangan asli.

Prinsip :

Misal P(n) adalah suatu pernyataan dimana n adalah bilangan asli.

Apabila P(1) benar, dan apabila P(k) benar maka P(k+1) juga benar. Maka, P(n) benar untuk semua n.

D. Contoh Penyangkal (Counter Example)
Counter example digunakan untuk membuktikan suatu pernyataan salah dengan memberikan sebuah contoh penyangkal.
Contoh : “Semua bilangan prima adalah bilangan ganjil”.
Contoh penyangkalnya : 2 adalah bilangan prima yang genap.
Dengan demikian, pernyataan “semua bilangan prima adalah bilangan ganjil” terbukti salah.

Fun Fact: Bukti matematika sering kali membantu menemukan pola baru dan pemahaman mendalam tentang matematika itu sendiri!

Set Theory and Function

A. Sets (Himpunan)
Himpunan adalah kumpulan objek yang berbeda tetapi dari satu segi dapat dianggap
sebagai satu kesatuan.
$x \in S$ menyatakan x adalah anggota himpunan S.
$x \notin S$ menyatakan x bukan anggota himpunan S.
Dua cara menyatakan himpunan :

1) Roster Method
Contoh : A = {Senin, Selasa, Rabu, Kamis, Jumat, Sabtu, Minggu}, merupakan
himpunan nama-nama hari dalam satu minggu.

2) Set Builder Notation
Bentuk umum : {x : P(x) adalah benar}, dimana P(x) merupakan suatu himpunan.
Contoh : Himpunan B dari semua bilangan bulat positif genap kurang dari 10:
B = {x | x adalah bilangan bulat positif genap kurang dari 10}, atau
B = {x ϵ Z+ | x adalah bilangan genap dan x < 10}

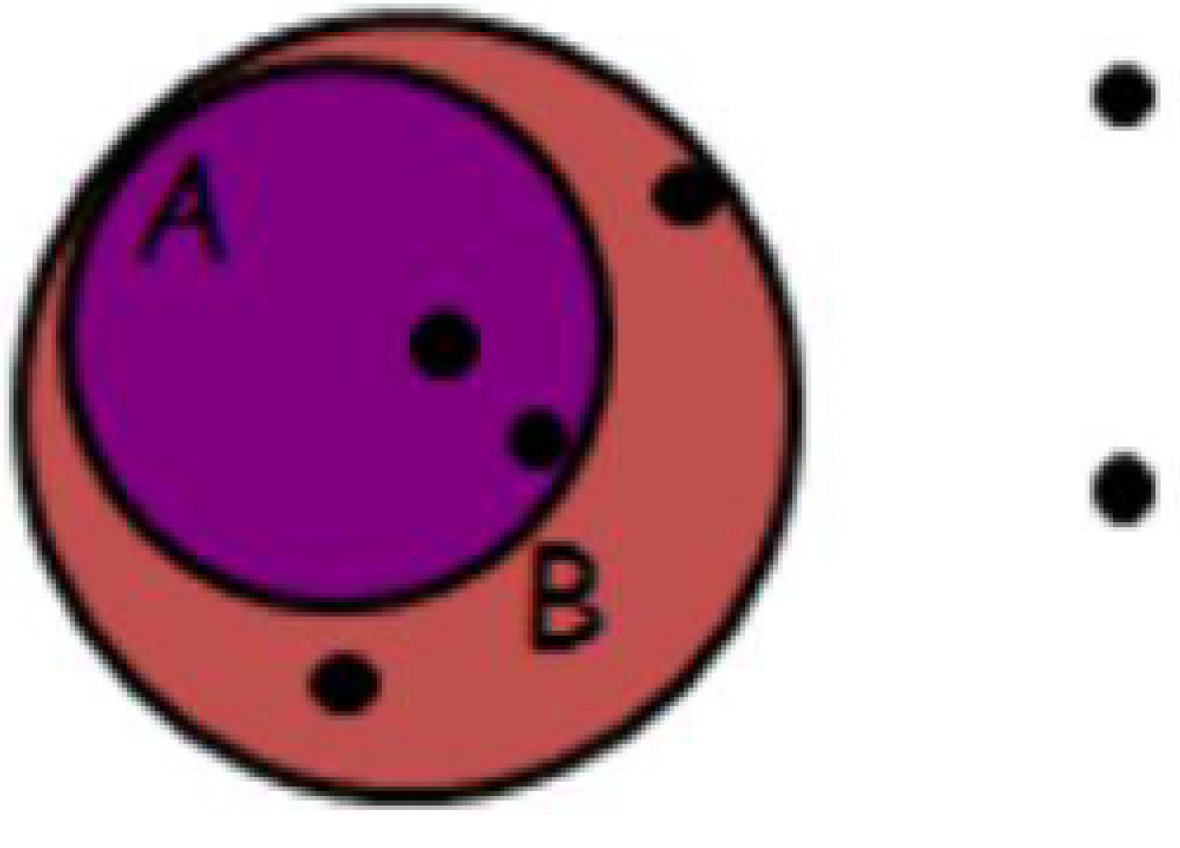
B. Diagram Venn

Superset dan Subset :

A ⊆ B menyatakan A adalah himpunan bagian dari B.

A ⊇ B menyatakan A memuat B.

A = B jika dan hanya jika A dan B memiliki elemen yang sama.



Kardinalitas : |S|
Menunjukkan banyaknya elemen berbeda pada suatu himpunan berhingga.
Contoh :
a. A = {1, 2, 3}, maka |A| = 3
b. B = {5, 5, 5, 5, 5}, maka |B| = 1
c. C = ∅, maka |C| = 0
d. D = {0, 1, 2, 3, ...}, maka |D| = tak hingga
Himpunan Kuasa : P(S)
Adalah himpunan dari seluruh himpunan bagian S.
Contoh :
a. S = {a}, maka P(S) = {∅, {a}}
b. S = {a, b}, maka P(S) = {∅, {a}, {b}, {a, b}}
c. S = ∅, maka P(S) = {∅}
Perkalian Kartesian :
Adalah perkalian antara himpunan A dan himpunan B, yaitu :
$A \times B = \{<a,b> : a \in A \wedge b \in B\}$
Contoh :
A = {1, 2, 3}, B = {a, b}
A × B = {(1, a), (1, b), (2, a), (2, b), (3, a), (3, b)}
B × A = {(a, 1), (a, 2), (a, 3), (b, 2), (b, 2), (b, 3)}

C. Operasi Himpunan

1. Gabungan (Union) : $A \cup B = \{x : x \in A \vee x \in B\}$

Contoh : $A = \{1, 3, 5\}$, $B = \{1, 2, 4, 6\}$

$A \cup B = \{1, 2, 3, 4, 5, 6\}$

2. Irisan (Intersection) : $A \cap B = \{x : x \in A \wedge x \in B\}$

Contoh : $A = \{2, 5, 8\}$, $B = \{1, 2, 5\}$

$A \cap B = \{2, 5\}$

3. Saling Asing (Disjoint) : Tidak mempunyai irisan

Contoh : $A = \{2, 4, 6, 8\}$, $B = \{1, 3, 5, 7, 9\}$

$A \cap B = \varnothing$. Sehingga A dan B adalah himpunan yang saling asing.

4. Komplemen : $A^C = \{x : x \notin A\}$

Contoh : S = warga Binus, A = {x : x adalah mahasiswa Binus}, B = {x : x adalah dosen Binus}.

A^C = {x : x adalah seluruh warga Binus yang bukan mahasiswa Binus}

5. Selisih Himpunan : $A - B = A \cap B^C$

Contoh : $A = \{1, 2, 4, 6\}$, $B = \{1, 3, 5, 6\}$

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

D. Fungsi
Fungsi f dari himpunan X ke himpunan Y adalah hubungan dimana setiap anggota X
(input) dipasangkan dengan satu anggota Y (output).
$f : X \to Y$
X = domain = daerah asal
Y = co-domain = daerah kawan

1. Fungsi One-to-One (Injective)
Sebuah fungsi dinyatakan one-to-one jika dan hanya jika f(a) = f(b) menyiratkan
bahwa a = b untuk semua a dan b dalam domain fungsi f.
Tips : Seluruh anggota domain sebuah fungsi memiliki satu pasangan yang berbeda.

2. Fungsi Onto
Sebuah fungsi f, dengan f(A) ke f(B) dinyatakan Onto jika dan
hanya jika untuk setiap element b ∈ B ada element a ∈ A
dengan f(A) = b.

3. Fungsi Berkorespondensi One-to-One (Bijective)
Sebuah fungsi dinyatakan berkorespondensi one-to-one jika sebuah fungsi memenuhi
kondisi one-two-one dan onto.

4. Fungsi Invers
Sebuah fungsi f yang berkorespondensi one-to-one dari A ke B, maka fungsi
inversnya adalah fungsi yang menetapkan elemen b milik b elemen unik a dalam A
sedemikian rupa sehingga f(a)=b.

5. Fungsi Komposisi
Fungsi komposisi adalah fungsi yang melibatkan lebih dari satu fungsi. Misalkan terdapat fungsi f(x) dan g(x), maka fungsi f(x) komposisi g(x) adalah fungsi yang dipetakan oleh fungsi g(x) kemudian dilanjutkan oleh fungsi f(x). Komposisi di notasikan dengan ($fog(x)$).
Tips : $(fog)(x) = f(g(x))$, fungsi g dikerjakan terlebih dahulu lalu hasilnya

Fuzzy Set

Himpunan fuzzy merupakan suatu grup yang mewakili suatu kondisi / keadaan
tertentu dalam suatu variabel fuzzy. Himpunan fuzzy memiliki nilai keanggotaan
dalam interval 0 sampai 1.

Contoh variabel fuzzy :
- Tinggi badan = {pendek, sedang, tinggi}
- Suhu = {panas, hangat, dingin}

Nilai keanggotaan himpunan fuzzy dihitung menggunakan fungsi keanggotaan.

Operasi Himpunan Fuzzy

1) Komplemen, $A(x) = 1 - A(x)$
Contoh : $A(x) = x - 6$, $maka\ A(x) = 1 - (x - 6) = -x + 7$

2) Gabungan (Union), $(A \cup B)(x) = max[A(x), B(x)]\ untuk\ semua\ x \in X$
Contoh : A(x) = 0/5,75 + 0/6 + 0,25/6,25 + 0,5/6,5 + 0,75/6,75 + 1/7 +
0,75/7,25 + 0,5/7,5 + 0,25/7,75 + 0/8 + 0/8,25
Komplemennya : $A(x)$ = 1/5,75 + 1/6 + 0,75/6,25 + 0,5/6,5 + 0,25/6,75 + 0/7
+ 0,25/7,25 + 0,5/7,5 + 0,75/7,75 + 1/8 + 1/8,25
Maka : $(A \cup A)(x)$ = 1/5,75 + 1/6 + 0,75/6,25 + 0,5/6,5 + 0,75/6,75 + 1/7 +
0,75/7,25 + 0,5/7,5 + 0,75/7,75 + 1/8 + 1/8,25

Operasi Himpunan Fuzzy

3) Irisan (Intersection), $(A \cap B)(x) = min[A(x), B(x)]$ *untuk semua* $x \in X$

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

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

Maka : $(A \cap A)(x)$ = 0/5,75 + 0/6 + 0,25/6,25 + 0,5/6,5 + 0,25/6,75 + 0/7 +
0,25/7,25 + 0,5/7,5 + 0,25/7,75 + 0/8 + 0/8,25

Himpunan Penyokong (Support Set)
Support set adalah himpunan dari elemen-elemen yang nilai
keanggotaannya lebih besar dari 0. $Supp(A) = \{x \in X \mid A(x) > 0\}$
Contoh : A(x) = 0/5,75 + 0/6 + 0,25/6,25 + 0,5/6,5 + 0,75/6,75 + 1/7 + 0,75/7,25 +
0,5/7,5
Supp(A) = {6.25, 6.5, 6.75, 7, 7.25, 7.5}

Himpunan Terpotong α (α Cut Set)
α Cut set adalah himpunan yang nilai keanggotaan tiap elemennya ≥ **α** .
Contoh : A(x) = 0/5,75 + 0/6 + 0,25/6,25 + 0,5/6,5 + 0,75/6,75 + 1/7 + 0,75/7,25 +
0,5/7,5
F0,3 = {6.5, 6.75, 7, 7.25, 7.5}

V. NUMBER THEORY

a) Keterbagian

1. Misalkan a dan b bilangan bulat, $a \neq 0$. Maka a habis membagi b jika terdapat bilangan bulat c sedemikian sehingga $b = ac$.
2. $a \mid b \rightarrow$ "a habis membagi b atau b habis dibagi a" jika $b = ac$, $c \in Z$ (Z=himpunan bilangan bulat) dan $a \neq 0$.
$a \nmid b \rightarrow$ "a tidak habis membagi b".
3. Jika a habis membagi b, a disebut faktor dari b, dan b disebut kelipatan dari a.

Contoh : $4 \mid 12$, karena $12 \div 4 = 3$ (bilangan bulat)
$4 \nmid 13$, karena $13 \div 4 = 3.25$ (bukan bilangan bulat)

Teorema I

Untuk bilangan bulat a, b, dan c berlaku :

(i) Jika $a \mid b$ dan $a \mid c$, maka $a \mid (b+c)$

(ii) Jika $a \mid b$, maka $a \mid bc$ untuk seluruh bilangan bulat c

(iii) Jika $a \mid b$ dan $b \mid c$, maka $a \mid c$

Teorema II

Jika a dan b adalah bilangan bulat dengan $b > 0$, maka ada

dengan tunggal pasangan bilangan bulat q dan r yang

memenuhi $a = bq + r$, dengan $0 \leq r < b$.

b sebagai pembagi, a sebagai dividen, q sebagai hasil bagi, dan r sebagai sisa bagi.

Notasi : q = a div b, r = a mod b

Contoh :

$22 = 3 \times 7 + 1$ (7 = 22 div 3, 1 = 22 mod 3) benar

$27 = 7 \times 4 - 1$ (4 = 27 div 7, -1 = 27 mod 7) salah

Teorema III

Jika a dan b bilangan bulat dan m bilangan bulat positif, maka

$a \equiv b \pmod{m}$ jika dan hanya jika m membagi habis $a - b$ atau

jika a dan b memberikan sisa yang sama jika dibagi oleh m.

Contoh : $17 \equiv 5 \bmod 6$, karena 6 membagi $17 - 5 = 12$

Teorema IV

Jika a dan b bilangan bulat dan m bilangan bulat positif, maka

$a \equiv b \pmod{m}$ jika dan hanya jika terdapat bilangan bulat k

sedemikian rupa sehingga $a = b + k \times m$.

Contoh : $17 \equiv 5 \bmod 6$, karena $17 = 5 + 2 \times 6$

Teorema V

Jika $a \equiv b \pmod{m}$ dan $c \equiv d \pmod{m}$, maka $a + c \equiv b + d \pmod{m}$ dan $ac \equiv bd \pmod{m}$.

Contoh : $7 \equiv 2 \pmod{5}$ dan $11 \equiv 1 \pmod{5}$, maka $7 + 11 \equiv 2 + 1 \pmod{5}$

b) Faktor Persekutuan Terbesar (FPB)

● Faktor persekutuan dari a dan b adalah bilangan bulat d yang memenuhi d | a dan d | b. Nilai terbesar dari d disebut faktor persekutuan terbesar (FPB) dari a dan b.

Notasi : (a, b) = d

Contoh : (10, 12) = 2

● Jika a dan b bilangan bulat positif dan (a, b) = 1, maka a relatif prima terhadap b.

Contoh : (17, 22) = 1

● Euclidean Algoritma

Tentukan FPB (4840, 1512) menggunakan Euclidean algoritma.

Jawab :

4840 = 3 × 1512 + 304

1512 = 4 × 304 + 296

304 = 1 × 296 + 8

296 = 37 × 8 + 0

Jadi, (4840, 1512) = 8

VI. COUNTING

a) Pencacahan (Counting)
Pencacahan adalah suatu cara untuk menghitung seluruh kemungkinan yang bisa terjadi dalam suatu eksperimen tertentu.

1. Aturan Perkalian
Digunakan jika :
- Ada satu kegiatan terdiri dari beberapa tahap
- Ada beberapa kegiatan berbeda yang semuanya harus dilakukan
Misalkan :
Ada $n1$ cara melakukan kegiatan 1.
Ada $n2$ cara melakukan kegiatan 2.
Ada nk cara melakukan kegiatan k.
Dimana semua kegiatan tersebut dilakukan bersamaan. Maka banyaknya cara melakukan seluruh kegiatan = $n1 \times n2 \times ... \times nk$
Contoh : Lili ingin membuat PIN ponselnya dengan 4 digit yang disusun dari angka 0
s/d 9, angka PIN bisa saja terdapat pengulangan. Berapa banyak peluang Lili
membuat PIN?
Jawab : $10 \times 10 \times 10 \times 10 = 10000$ PIN

2. Aturan Penjumlahan
Digunakan jika :
- Ada beberapa kegiatan berbeda namun hanya satu yang dilakukan.
- Saat sedang membagi kasus
Misalkan :
Ada $n1$ cara melakukan kegiatan 1.
Ada $n2$ cara melakukan kegiatan 2.
Ada nk cara melakukan kegiatan k.
Dimana semua kegiatan tersebut tidak dapat dilakukan bersamaan. Maka banyaknya
cara melakukan seluruh kegiatan = $n1 + n2 + \ldots + nk$
Contoh : Bibi memiliki 2 mobil, 4 sepeda motor dan 3 sepeda. Berapa banyak cara
Bibi dapat pergi ke kantor dengan kendaraannya?
Jawab : 2 + 4 + 3 = 9 cara

b) Peluang (Probability)
Peluang adalah ukuran numerik tentang suatu peristiwa yang dapat terjadi.
$0 \leq P(x) \leq 1$, $\Sigma P(x) = 1$
Contoh : melempar sebuah koin satu kali
Peluang muncul gambar : P(gambar) = 0,5
Peluang muncul angka : P(angka) = 0.5

Percobaan (Experiment) : memberi perlakuan terhadap sesuatu.
Contoh : melempar dadu

Ruang Sampel (S) : himpunan dari semua kemungkinan yang dapat terjadi dalam suatu eksperimen.
Contoh : Melempar sebuah koin dua kali berturut-turut → S = {AA, AG, GA, GG}

Kejadian / peristiwa (Event) : himpunan bagian dari ruang sampel yang mempunyai ciri tertentu.
Contoh : kejadian sekurang-kurangnya satu Gambar muncul jika sebuah koin
dilempar dua kali berturut-turut → A = {GG, GA, AG}

Hukum-hukum Dasar Peluang

1. Komplemen Suatu Kejadian

$P(A) + P(A^c) = 1 \rightarrow atau\ P(A) = 1 - P(A^c)$

Contoh : Sebuah koin dilempar 2 kali. Berapa peluang sedikitnya satu Angka muncul?

A = paling sedikit satu A muncul

A^c = tidak ada A yang muncul

S = {AA, AG, GA, GG}

P(A^c) = P(AA) = ¼

P(A) = 1 - P(A) = 1 - (¼) = ¾

2. Irisan Dua Kejadian

$A \cap B = \{x | x \in A \; dan \; x \in B\}$

3. Gabungan Dua Kejadian

$A \cup B = \{x | x \in A \; dan \; x \in B\}$

Rumus : $P(A \cup B) = P(A) + P(B) - P(A \cap B)$

4. Kejadian Terpisah

Jika kejadian A muncul, maka kejadian B tidak muncul, berlaku sebaliknya. Jadi, kejadian A dan B tidak dapat muncul bersamaan.

$P(A \cap B) = 0$

Rumus : $P(A \cup B) = P(A) + P(B)$

Peluang Bersyarat

P(A | B) = peluang terjadinya peristiwa A dengan syarat peristiwa B telah terjadi.

P(A | B) = $P(A \cap B)$

$P(B)$ → sehingga : $P(A \cap B)$ = P(B) P(A | B)

atau P(B | A) = $P(A \cap B)$

$P(Aa)$ → sehingga : $P(A \cap B)$ = P(A) P(B | A)

Untuk peristiwa saling bebas : $P(A \cap B)$ = P(A) P(B) = P(B) P(A)

Contoh : Sebuah Polres di Jakarta memiliki kekuatan 1000 personel polisi terdiri atas polisi wanita dan pria. Seminggu yang lalu sebagian anggota Polres tersebut mendapat kenaikan pangkat. Perhatikan tabel berikut ini.

a. Berapa peluang terjadinya anggota polisi adalah wanita atau yang naik pangkat?

b. Berapa peluang anggota polisi yang naik pangkat dengan syarat dia adalah
wanita?

Jawab :

P(Wanita) = 250 / 1000 = 0.25

P(NP) = 350 / 1000 = 0.35

P(NP ∩ Wanita) = 150 / 1000 = 0.15

a. P(Wanita ∪ NP) = 0.25 + 0.35 - 0.15 = 0.45

b. P(NP | Wanita) = 0.15 / 0.25 = 0.6

c) Permutasi

Permutasi tidak ada unsur yang sama. Banyaknya urutan yang dapat dibentuk dari n objek yang diambil dari N objek yang lebih besar.

Rumus : Pn

N = N!

(N−n)!

Contoh : Berapa banyak permutasi 3 huruf yang diambil dari huruf-huruf V, W, X, Y?

Jawab : n = 3 ; N = 4

P3

4 = 4!

(4 − 3)! = 4.3.2.1

1! = 24

Permutasi ada unsur yang sama

Misalkan pada kata “BUKU” terdapat dua huruf yang sama, yaitu U. Berapa banyak permutasi yang terbentuk?

Jawab : Banyaknya permutasi pada “BUKU” = 4!

2! = 4.3.2.1

2.1 = 12

Berikut diagram pohon permutasi kata “BUKU” :
Hasil permutasi kata “BUKU” dengan adanya susunan huruf yang sama :

1. BUKU 5. BUKU 9. UUBK 13. KUBU 17. KUUB 21. UUBK
2. BUUK 6. BUUK 10. UUKB 14. KUUB 18. KUBU 22. UUKB
3. BKUU 7. UKBU 11. UBUK 15. KBUU 19. UBUK 23. UKBU
4. BKUU 8. UKUB 12. UBKU 16. KBUU 20. UBKU 24. UKUB

Hilangkan susunan huruf yang sama, sehingga menjadi :

1. BUKU 3. BKUU 5. UKUB 7. UUKB 9. UBKU 11. KUUB
2. BUUK 4. UKBU 6. UUBK 8. UBUK 10. KUBU 12. KBUU

d) Kombinasi

Kombinasi adalah banyaknya susunan tanpa memperhatikan urutannya, yang mungkin dapat dibentuk dari n objek yang diambil dari N objek yang lebih besar.

Rumus : Cn

N = N!

n!(N−n)!

Contoh : Berapa banyak permutasi 3 huruf yang diambil dari huruf-huruf V, W, X, Y?

Jawab : n = 3 ; N = 4

C3

4 = 4!

3! (4 − 3)! = 4.3.2.1

(3.2.1)(1) = 4

Kombinasinya : ABC, ABD, ACD, BCD

e) Binomium Newton

Apabila a dan b adalah bilangan riil dan n adalah bilangan bulat positif, maka :

Rumus Binomium : $(a + b)n = \sum n$

$k=0$

$n!$

$k!(n-k!)\ an-kbk$

$(a + b)0\ 1$

$(a + b)1\ a + b$

$(a + b)2\ a2 + 2ab + b2$

$(a + b)3\ a3 + 3a2b + 3ab2 + b3$

Contoh : Hitunglah $(x + 5)4$

Tanpa Koefisien : $x450\ x351\ x252\ x153\ x054$

Dengan Koefisien : $1x450\ 4x351\ 6x252\ 4x153\ 1\ x054$

Jadi, $(x + 5)4 = x4 + 20x3 + 150x2 + 500x + 625$

VII. RELATION

a) Relasi

Relasi adalah hubungan antara anggota suatu himpunan dengan anggota himpunan lain.

- Relasi Biner : hubungan antara anggota dua buah himpunan.

$R : A \rightarrow B$ atau aRb

$R = \{(x, y) \in A \times B \mid (x, y) \in R\}$

Himpunan A disebut sebagai domain / daerah asal dari R, dan himpunan B disebut sebagai range / daerah hasil dari R.

Contoh :

A = {1, 2}

B = {Ami, Bibi, Cici}

$R : A \rightarrow B$, maka A×B = {(1, Ami), (1, Bibi), (1, Cici), (2, Ami), (2, Bibi), (2, Cici)}

- Relasi Pada Suatu Himpunan A : relasi himpunan dari himpunan A ke A. R : A → A atau aRa

Contoh : A = {1, 2, 3, 4}, pasangan terurut mana yang memiliki relasi R = {(a, b) | a habis membagi b}?

Jawab : karena (a, b) berada di R jika dan hanya jika a dan b bilangan bulat positif tidak melebihi 4 sehingga a habis membagi b.
Maka R = {(1, 1), (1, 2), (1, 3), (1, 4), (2, 2), (2, 4), (3, 3), (4, 4)

● Sifat - Sifat Relasi

1. Refleksif

Relasi R pada himpunan A disebut refleksif jika setiap anggota A berhubungan dengan dirinya. (a, a) ϵ R untuk setiap elemen a ϵ A.

Contoh : Perhatikan relasi pada A = {2, 4, 6, 8}

R1 = {(2, 2), (2, 4), (4, 2), (4, 4), (6, 8), (8, 2), (8, 8)} → tidak refleksif, karena tidak ada (6, 6)

R2 = {(2, 2), (2, 4), (4, 2)} → tidak refleksif, karena tidak ada (4, 4), (6, 6), (8, 8)

R3 = {(2, 2), (2, 4), (2, 8), (4, 2), (4, 4), (6, 6), (8, 2), (8, 8)} → refleksif

R4 = {(4, 2), (6, 2), (6, 4), (8, 2), (8, 4), (8, 6)} → tidak refleksif, karena tidak ada (2, 2), (4, 4), (6, 6), (8,8)

R5 = {(2, 2), (2, 4), (2, 6), (2, 8), (4, 4), (4, 6), (4, 8), (6, 6), (6, 8), (8,8)} → refleksif

R6 = {(6, 8)} → tidak refleksif, karena tidak ada (2, 2), (4, 4), (6, 6), (8,8)

2. Simetris

Relasi R pada himpunan A disebut simetris jika (a, b) $\in$ R, maka (b, a) $\in$ R untuk setiap a,b $\in$ A.

Contoh : Perhatikan relasi pada A = {2, 4, 6, 8}

R1 = {(2, 2), (2, 4), (4, 2), (4, 4), (6, 8), (8, 2), (8, 8)} → tidak simetris, karena ada (6, 8) dan (8, 2) tetapi tidak ada (8, 6) dan (2, 8)

R2 = {(2, 2), (2, 4), (4, 2)} → simetris, karena ada (2, 4) dan (4, 2)

R3 = {(2, 2), (2, 4), (2, 8), (4, 2), (4, 4), (6, 6), (8, 2), (8, 8)} → simetris

R4 = {(4, 2), (6, 2), (6, 4), (8, 2), (8, 4), (8, 6)} → tidak simetris, karena tidak ada (2,4), (2, 6), (4, 6), (2, 8), (4, 8), (6, 8)

R5 = {(2, 2), (2, 4), (2, 6), (2, 8), (4, 4), (4, 6), (4, 8), (6, 6), (6, 8), (8,8)} → tidak simetris, karena tidak ada (4, 2), (6, 2), (8, 2), (6, 4), (8, 4) (8, 6)

R6 = {(2, 6), (4, 6), (4, 8), (6, 2), (6, 4), (8, 4)} → simetris

3. Antisimetris

Relasi R pada himpunan A disebut antisimetris jika (a, b) $\in$ R dan (b, a) $\in$ R, berlaku hanya jika a = b, untuk setiap a,b $\in$ A.

Contoh : Perhatikan relasi pada A = {2, 4, 6, 8}

R1 = {(2, 2), (4, 4), (6, 6)} → antisimetris dan simetris

R2 = {(2, 2), (2, 4), (4, 4), (4, 6)} → antisimetris karena ada (2, 2) dan (4, 4); tidak

simetris karena tidak ada (4, 2) dan (6, 4)

4. Transitif

Relasi R pada himpunan A disebut transitif jika a berhubungan dengan b, dan b berhubungan dengan c, maka a berhubungan dengan c secara langsung.

$(a, b) \in R \land (b, c) \in R \rightarrow (a, c) \in R$

Contoh : Perhatikan relasi pada A = {2, 4, 6, 8}

R1 = {(2, 2), (2, 4), (4, 2), (4, 4), (6, 8), (8, 2), (8, 8)} → tidak transitif, karena tidak ada (6, 2)

R2 = {(2, 2), (2, 4), (4, 2)} → tidak transitif, karena tidak ada (4, 4)

R3 = {(2, 2), (2, 4), (2, 8), (4, 2), (4, 4), (6, 6), (8, 2), (8, 8)} → tidak transitif karena tidak ada (8, 4)

R4 = {(4, 2), (6, 2), (6, 4), (8, 2), (8, 4), (8, 6)} → transitif

R5 = {(2, 2), (2, 4), (2, 6), (2, 8), (4, 4), (4, 6), (4, 8), (6, 6), (6, 8), (8,8)} → transitif

R6 = {(6, 8)} → tidak transitif

b) Relasi Ekivalen

Relasi Ekivalen adalah relasi yang bersifat refleksif, simetris, dan transitif.

Contoh : Diketahui himpunan A = {1, 2, 3}

R1 = {(1, 1), (1, 2), (1, 3), (2, 2), (2, 1), (2, 3), (3, 1), (3, 2), (3, 3)}

R1 bersifat :

Reflektif → karena ada (1, 1), (2, 2), dan (3, 3)

Simetris → karena ada (1, 2), (2, 1), (1, 3), (3, 1), (2, 3), (3, 2)

Transitif → karena ada (1, 2), (2, 3), dan (1, 3)

Dengan demikian, relasi R1 adalah relasi ekivalen.

- Kelas Ekivalen

Misalkan R adalah relasi ekivalen pada himpunan A. Himpunan semua elemen yang berelasi dengan elemen a dari A disebut kelas ekivalen dari a. Kelas ekivalen dari a terhadap R dilambangkan dengan $[a]_R$. Jika hanya satu relasi, maka dilambangkan dengan [a]. $[a]_R = \{s \mid (a, s) \in R\}$

Contoh : A = {0, 1, 2, 6, 9}
R = {(a, b) | 2 habis membagi a - b, dan a,b $\in$ A}
Tentukan semua kelas ekivalen yang terbentuk.
Jawab : R = {(0, 0), (0, 2), (0, 6), (1, 1), (1, 9), (2, 0), (2, 2), (2, 6), (6, 0), (6, 2), (6, 6),
(9, 1), (9, 9)}
[0] = [2] = [6] = {0, 2, 6}
[1] = [9] = {1, 9}

● Partisi

Kelas ekivalen membentuk partisi dari himpunan A, yaitu sub-sub himpunan A yang mempunyai sifat : Jika A1, A2, ..., An ⊆ A, maka dipenuhi dua hal sekaligus :

- $A1 \cup A2 \cup ... \cup An = A$ {keseluruhan menjadi satu utuh A}
- $Ai \cap Aj$ = Ø, jika i≠j, dan i, j = 1, 2, ... , n {tidak ada irisan}

Contoh : Jika A = {-2, -1, 3, 4, 5, 8} dan relasi R = {(a, b) | 2
habis membagi (a-b), a,b ϵ A}. Tentukan partisi dari A
terhadap relasi R.

Jawab :

[-2] = {-2, 4, 8}

[-1] = {-1, 3, 5}

c) Poset (Partially Ordered Set / Himpunan Terurut Parsial)
Poset adalah sebuah himpunan dimana elemennya terurut berdasarkan sebuah relasi urut. Poset adalah suatu relasi yang bersifat refleksif, transitif, dan antisimetris.

Misalkan R adalah suatu relasi poset, maka harus :
1. Reflektif : $a \, R \, a, \ \forall a$

2. Transitif : $a \, R \, b \wedge b \, R \, c \rightarrow a \, R \, c, \ \forall a, b, c$

3. Antisimetris : $a \, R \, b \wedge b \, R \, a \rightarrow a = b, \ \forall a, b$

Contoh : Relasi ‘ $\leq$ ‘ pada bilangan riil
Reflektif : $a \leq a$ untuk setiap bilangan riil
Transitif : Jika $a \leq b$, $b \leq c$, maka $a \leq c$
Antisimetris : Jika $a \leq b$, $b \leq a$, maka $a = b$ $\therefore$ relasi ‘ $\leq$ ‘ pada bilangan riil adalah relasi poset.

● Comparable dan Non Comparable

Misalkan diketahui suatu relasi terurut parsial / poset pada himpunan A. Elemen a dan b dari A disebut comparable jika dan hanya jika a | b atau b | a. Diluar ketentuan
tersebut, a dan b disebut non comparable.

Contoh : Pada poset (Z+, |)

- 3 dan 6 adalah comparable
- 3 dan 9 adalah comparable
- 3 dan 5 adalah non comparable

● Total Order

Total order adalah urutan parsial dimana tiap pasangan elemennya comparable.

Contoh :

(Z+, ≤) adalah total order, karena setiap pasangan (a, b) dalam Z×Z, maka a ≤ b atau
b ≤ a.

(Z+, |) adalah bukan total order, karena memiliki elemen noncomparable, misalkan 3
dan 5

d) Diagram Hasse

Diagram Hasse adalah diagram khusus untuk menggambarkan poset.

Prosedur pembuatan diagram hasse :

1. Mulai dengan graf berarah relasi dimana semua panah menuju ke tempat yang
lebih atas
2. Hilangkan loop pada setiap titik
3. Hilangkan panah yang keberadaannya bisa diimplikasikan dengan sifat
transitif
4. Hilangkan petunjuk panah → graf tak berarah

● Elemen Maksimal dan Minimal

- Sebuah elemen poset disebut maksimal jika tidak kurang dari elemen poset manapun. Artinya, a maksimal dalam poset $(S, \preceq)$ jika tidak ada b ϵ S sehingga a $\preccurlyeq$ b.

- Sebuah elemen poset disebut minimal jika tidak lebih besar dari elemen poset manapun. Artinya, a minimal jika tidak ada elemen b ϵ S sehingga b $\preceq$ a.

● Elemen Terbesar dan Terkecil

- Sebuah elemen dalam poset yang lebih besar dari setiap elemen lainnya. Artinya, a
adalah elemen terbesar dari poset $(S, \preceq)$ jika b a untuk semua b ϵ S.
- Sebuah elemen dalam poset disebut elemen terkecil jika kurang dari elemen lainnya
dalam poset. Artinya, a adalah elemen terkecil dari $(S, \preceq)$ jika a b untuk
semua b ϵ S.

● Batas Atas dan Bawah

- Sebuah elemen yang lebih besar atau sama dengan semua elemen dalam himpunan A dari sebuah poset (S, $\preceq$). Jika u adalah elemen S sehingga a $\preceq$ u untuk semua elemen a ϵ A, maka u disebut batas atas (UB) dari A.

- Sebuah elemen yang kurang dari atau sama dengan semua elemen dalam A jika l adalah elemen S sehingga l $\preceq$ a untuk semua elemen a ϵ A, maka l disebut batas bawah (LB) dari A.

- Elemen x disebut batas terkecil (LUB) dari himpunan A (lub(A)) jika a $\preceq$ x bila a ϵ A, dan x $\preceq$ z bila z adalah batas atas dari A.

- Elemen y disebut batas terbesar (GLB) dari A (glb(A)) jika y adalah batas bawah dari A dan z $\preceq$ y bila z adalah batas bawah dari A.

e) Lattices

Lattices adalah sebuah poset dimana setiap pasangan elemen memiliki batas terbesar (lub) dan batas terkecil (glb).

Contoh :

Diagram hasse (a) dan (c) keduanya adalah lattices. Diagram hasse (b) bukan lattice, karena elemen b dan c tidak memiliki lub.

f) Merepresentasikan Relasi Menggunakan Matriks

Misalkan R adalah relasi dari A = {a1, a2, ..., am} ke B = {b1, b2, ..., bn}. Relasi R bisa direpresentasikan oleh matriks MR = [mij], dimana :

mij = { 0 $jika$ $(ai,bj) \notin R$.

1 $jika$ $(ai,bj) \in R$.

g) Merepresentasikan Relasi Menggunakan Grafik Berarah

Grafik berarah atau digraf terdiri dari himpunan V simpul (vertices / nodes) bersama dengan himpunan E poset dari elemen V yang disebut sisi (edges / arcs). Simpul a disebut simpul awal dari sisi (a, b), dan simpul b disebut simpul terminal.

VIII. GRAPH

a) Graf

Graf adalah himpunan pasangan simpul dan sisi yang ditulis dengan notasi G = (V, E).
Dimana :
V = vertex / simpul/ node = himpunan tidak kosong dari simpul-simpul = $\{V1, V2, \ldots, Vn\}$
E = edges / sisi / arc = himpunan sisi yang menghubungkan sepasang simpul = $\{e1, e2, \ldots, en\}$
Busur (arc) = sisi yang melengkung Himpunan sisi (E) dapat berupa himpunan kosong yakni simpul-simpul yang berdiri sendiri tanpa sisi, dan graf tersebut dinamakan graf kosong (empty graph / null graph).
Catatan :
- Sebuah simpul dapat berdiri sendiri tanpa sisi.
- Sebuah sisi harus memiliki minimal satu simpul, contohnya adalah graf loop.

● Terminologi Graf

1. Bertetangga (Adjacent)
Dua buah simpul dinyatakan adjacent / bertetangga jika kedua simpul tersebut terhubung langsung oleh suatu sisi.

2. Bersisian (Incidency)
Suatu sisi e dinyatakan bersisian dengan simpul v1 dan simpul v2 jika e menghubungkan kedua simpul tersebut. Dengan kata lain e = (v1, v2).

3. Simpul Terpencil (Isolated Vertex)
Suatu simpul dinyatakan simpul terpencil jika simpul tersebut tidak mempunyai
sisi yang bersisian dengannya

● Pengelompokan Graf
Graf dikelompokkan berdasarkan orientasi arah pada sisinya, yaitu graf tak berarah (undirected graph) dan graf berarah (directed graph).

1. Graf Tak Berarah (Undirected Graph)
Graf yang setiap sisinya tidak memiliki arah / panah. Urutan pasangan simpul pada graf tak berarah tidak diperhatikan. Sehingga (A, B) = (B, A) menunjukkan sisi yang sama.

a. Graf Sederhana (Simple Graph)
Graf sederhana merupakan graf tak berarah yang tidak memiliki loop maupun sisi ganda.

b. Graf Lengkap (Complete Graph) : Kn
Graf lengkap merupakan suatu graf sederhana yang setiap simpulnya terhubung ke semua simpul lainnya oleh satu sisi, maka setiap simpulnya bertetangga.

c. Graf Bipartit (Bipartite Graph)
Sebuah graf sederhana dinyatakan graf bipartit jika himpunan simpul pada graf tersebut dapat dipisah menjadi dua himpunan tak kosong yang terpisah, misal v1 dan v2, sedemikian sehingga setiap sisi pada graf menghubungkan sebuah simpul pada v1 dan sebuah simpul pada v2. Maka pada graf bipartit tidak ada sisi yang menghubungkan dua simpul pada v1 atau v2

d. Graf Bipartit Lengkap (Complete Bipartite Graph) :
Km,n
Setiap simpul daerah asal terhubung dengan seluruh simpul daerah tujuan

e. Graf Ganda (Multigraph)
Graf ganda merupakan graf tak berarah yang memiliki sisi ganda.

f. Graf Semu (Pseudograph)
Graf semu merupakan graf tak berarah yang memiliki loop.

2. Graf Berarah (Directed Graph)
Graf yang setiap sisinya mempunyai arah / panah. Urutan pasangan simpul pada graf berarah diperhatikan berdasarkan arah panah sisi tersebut.

● Derajat
Derajat suatu simpul merupakan jumlah sisi yang bersisian dengan simpul tersebut. Misalkan suatu simpul v mempunyai 3 buah sisi yang bersisian dengannya, maka dapat dikatakan simpul tersebut berderajat 3, atau dinotasikan dengan d(v) = 3.

● Walk, Trail, Path, dan Circuit
Walk : barisan simpul dan sisi $v0e1v1e2...vnen$ yang saling terhubung pada suatu
graf.
Trail : jalan yang semua sisinya berbeda.
Path : jalan yang semua simpulnya berbeda.
Circuit : jalan tertutup yang semua sisinya berbeda.

b) Keterhubungan Suatu Graf

Dua buah simpul $v1$ dan $v2$ pada suatu graf dikatakan terhubung jika terdapat lintasan dari $v1$ ke $v2$. Jika setiap pasang simpul $v1$ dan $v2$ dalam himpunan v pada suatu graf G terdapat lintasan dari vi dan vj maka graf tersebut dinamakan graf terhubung (connected graph). Jika tidak maka G dinamakan graf tak terhubung (disconnected graph).

- Graf terhubung lemah : jika terdapat path antara dua simpul di graf tidak berarah yang mendasarinya.
- Graf terhubung kuat : jika terdapat path dari a ke b dan dari b ke a setiap kali a dan b adalah simpul dalam graf.

c) Matriks Ketetanggaan

Matriks ketetanggaan untuk graf sederhana merupakan matriks bujur sangkar yang unsur-unsurnya hanya terdiri dari dua bilangan yaitu 0 (nol) dan 1 (satu). Baris dan kolom pada matriks ini, masing-masing merupakan representasi dari setiap simpul

pada graf tersebut. Misal aij merupakan unsur pada matriks tersebut, maka :

- Jika aij = 1 berarti simpul i dan simpul j bertetangga.
- Jika aij = 0 berarti simpul i dan simpul j tidak bertetangga.

d) Graf Isomorfik

Sifat - sifat graf isomorfik :

1. Memiliki jumlah simpul yang sama.
2. Memiliki jumlah sisi yang sama.
3. Memiliki simpul atau simpul tertentu berderajat sama.
4. Memiliki matriks ketetanggaan yang sama

e) Sirkuit Euler (Euler Circuit)
Sirkuit Euler merupakan graf terhubung yang tiap simpulnya berderajat genap

f) Lintasan Euler (Euler Path)
Lintasan Euler merupakan lintasan yang melalui masing-masing sisi dalam graf tepat
satu kali. Jika lintasan tersebut kembali ke simpul awal membentuk lintasan tertutup,
maka disebut sirkuit Euler.

g) Sirkuit Hamilton (Hamilton Circuit)
Sirkuit Hamilton merupakan lintasan tertutup yang melalui tiap simpul di dalam graf tepat satu kali, kecuali simpul awal / akhir dilalui dua kali. Graf yang memiliki sirkuit Hamilton disebut graf Hamilton.

h) Lintasan Hamilton (Hamilton Path)
Lintasan Hamilton merupakan lintasan yang melalui tiap simpul di dalam graf tepat satu kali. Graf yang memiliki lintasan Hamilton disebut graf semi Hamilton.

i) Pewarnaan Graf (Graph Coloring)

● Pewarnaan Simpul

Tujuan : agar simpul-simpul yang bertetangga memiliki warna yang berbeda. Jumlah warna minimum yang digunakan untuk mewarnai seluruh simpul suatu graf ditunjukkan oleh bilangan kromatik (χ). Untuk graf lengkap K_n memiliki χ = n karena semua simpul saling terhubung sehingga diperlukan n buah warna.

● Algoritma Welch-Powell

Algoritma Welch-Powell merupakan suatu cara yang efisien untuk mewarnai sebuah graf.

Langkah-langkah algoritma Welch-Powell :

1. Urutkan semua simpul berdasarkan derajatnya dari yang terbesar hingga terkecil.
2. Gunakan warna pertama (misal biru) untuk mewarnai simpul pertama (misal A). Kemudian warnai semua simpul yang tidak bertetangga dengan simpul A dengan warna biru.
3. Ulangi 2 langkah di atas untuk warna kedua dan seterusnya sampai semua simpul sudah memiliki warna.

● Pewarnaan Sisi

Pewarnaan sisi yakni memberikan warna berbeda pada sisi yang bersisian pada suatu graf tanpa loop sehingga tidak ada dua sisi yang bersisian mempunyai warna yang sama. Jumlah warna minimum untuk mewarnai seluruh sisi suatu graf disebut indeks kromatik (χ').

IX. TREE

a) Pohon (Tree)

Tree adalah graf tak berarah terhubung yang tidak mengandung sirkuit sederhana, tidak memiliki loop, tidak memiliki garis paralel, namun bersifat hierarki.

● Pohon Keluarga (Family Tree)

Pohon keluarga adalah graf yang mewakili bagian silsilah. Pohon keluarga menggunakan simpul untuk mewakili anggota keluarga dan sisi untuk mewakili hubungan.

● Istilah dalam Tree

- Hutan (Forest) : kumpulan tree yang tidak mengandung sirkuit dan tidak saling terhubung antar tree-nya.
- Pohon Semu (Trivial Tree) : tree yang hanya terdiri dari sebuah titik.
- Pohon Kosong : tree yang tidak mempunyai titik.
- Daun (Leaf / Terminal Vertex) : titik dalam tree yang berderajat 1.
- Titik Cabang (Branch / Internal Vertex) : titik dalam tree yang berderajat > 1

- Model-Model Tree

1. Model Pohon Keputusan (Decision Tree)

Pohon keputusan menggunakan struktur hirarki. Proses dari pohon keputusan dimulai dari simpul akar hingga simpul daun yang dilakukan secara rekursif. Dimana setiap percabangan menyatakan suatu kondisi yang harus dipenuhi pada setiap ujung pohon menyatakan kelas dari suatu data.

2. Model Pohon Parse (Parse Tree)

Pohon parse merupakan struktur pada proses syntactic parsing dimana parser akan mencoba mencari cara untuk membuat tree sesuai dengan grammar dari bahasa tersebut

b) Pohon Berakar (Rooted Tree)

Pohon berakar adalah pohon yang satu simpulnya telah ditetapkan sebagai akar dan sisi-sisinya diberi arah menjauhi akar, sehingga menjadi graf berarah

● M-ary Tree

Sebuah rooted tree disebut m-ary tree jika setiap internal vertex tidak memiliki lebih dari m anak. Binary Tree (Pohon Biner) adalah sebuah m-ary tree yang memiliki m = 2

● Full M-ary Tree

Sebuah tree disebut full m-ary tree jika setiap internal vertex memiliki tepat m anak.

● Balanced M-ary Tree

Pohon berakar m-ary dengan tinggi h disebut seimbang jika seluruh daunnya memiliki level h atau h-1.

● Binary Tree

Binary Tree merupakan pohon berakar dimana tiap parent memiliki dua child.

● Ordered Rooted Tree (Pohon Berakar Terurut)
Pohon berakar terurut adalah pohon berakar dengan kondisi child dari setiap internal vertexnya terurut

● Sifat-Sifat Tree

1. Pohon dengan n simpul memiliki n-1 sisi.
2. Pohon full m-ary dengan i simpul internal memiliki n = mi + 1 simpul.
3. Pohon full m-ary dengan :

a. n simpul, memiliki i = (n−1)
m simpul internal dan l = [(m−1)n+1]
m daun.

b. i simpul internal, memiliki n = mi + 1 simpul dan l = (m − 1) + 1 daun.

c. l daun, memiliki n = (ml−1)
(m−1) simpul dan i = (l−1)
(m−1) simpul internal.

c) Tree Traversal

Traversal : proses melakukan kunjungan pada setiap simpul pada suatu pohon biner
tepat satu kali.

- Universal Address System
- Binary Search Tree (BST)

1. Algoritma Pre-order Traversal : root, subtree kiri, subtree kanan
2. Algoritma In-order Traversal : subtree kiri, root, subtree kanan
3. Algoritma Post-order Traversal : subtree kiri, subtree kanan, root

- Notasi Infix, Prefix, dan Postfix

Ekspresi operasi aritmatika menggunakan pohon biner.

1. Notasi Infix : melintasi pohon berakar secara inorder
2. Notasi Prefix : melintasi pohon berakar secara preorder
3. Notasi Postfix : melintasi pohon berakar secara postorder.

d) Pohon Rentang (Spanning Tree)

Pohon rentang dari suatu graf sederhana G adalah subgraf dari G yang merupakan pohon yang berisi setiap simpul G.

- Setiap graf terhubung memiliki pohon rentang
- Dua pohon rentang dari suatu graf memiliki jumlah sisi yang sama

● Depth-First Search

Depth-First Search atau penelusuran kedalaman pertama juga disebut backtracking atau penelusuran mundur, karena algoritma mengembalikan ke simpul yang sebelumnya dikunjungi untuk menambahkan jalur.

Langkah-langkah DFS :

- Pilih simpul sembarang pada suatu graf.
- Lalu bentuk jalur hingga mengunjungi semua simpul.
- Jika jalur melewati semua simpul pada grafik, pohon yang terdiri dari jalur ini adalah pohon rentang.

- Breadth-First Search

Langkah-langkah BFS :

- Pilih simpul sembarang pada suatu graf.
- Lalu bentuk semua sisi yang bersisian dengan simpul tersebut.
- Lakukan hal di atas untuk simpul selanjutnya.

e) Pohon Rentang Minimum (Minimum Spanning Tree)
Tujuan : untuk mencari lintasan terpendek pada suatu pohon rentang dari graf berbobot.
Syarat : Sisi boleh dihilangkan dengan memilih bobot terkecil, tetapi simpul tidak boleh dihilangkan.
Semua simpul harus terhubung.
Jumlah garis = jumlah simpul - 1.
1. Algoritma Prim
Gunakan algoritma prim untuk merancang jaringan komunikasi dengan biaya minimum yang menghubungkan semua komputer yang diwakili oleh grafik

2. Algoritma Kruskal
Algoritma Kruskal memiliki metode dengan mengurutkan setiap sisi dari suatu graf G mulai dari yang terkecil. Kemudian setiap sisi yang tidak membentuk simple sirkuit dimasukkan ke dalam graf kosong T sebagai minimum spanning tree G.

X. FINITE STATE AUTOMATA

a) Finite State Automata

- Automata merupakan mesin abstrak yang dapat mengenali (recognize), menerima (accept), atau membangkitkan (generate) sebuah kalimat dalam bahasa tertentu.
- Finite State Automata (FSA)

FSA merupakan suatu model matematika yang dapat menerima input dan mengeluarkan output yang memiliki state yang berhingga banyaknya dan dapat berpindah dari satu state ke state lainnya berdasarkan input dan fungsi transisi.

M = (S, I, f, s0, F)

S = himpunan keadaan / state

I = himpunan simbol input

f = fungsi transisi, menggambarkan transisi FSA akibat pembacaan simbol input s0 = state awal (sebuah panah dengan lingkaran tunggal)

F = state penerima / akhir (lingkaran ganda)

- Pengenalan Bahasa oleh Finite State Machines

L(M) = bahasa yang dikenali oleh mesin M

M = seluruh himpunan string yang dikenal oleh mesin M

b) Designing Finite State Automata
Contoh : Rancanglah sebuah FSA yang mengenal tiap bahasa berikut :
(a) Himpunan bit string yang diawali dengan 00
(b) Himpunan bit string yang tidak berisi dua angka 0 berurutan
(c) Himpunan bit string yang diakhiri dengan dua angka 0

- FSA yang Ekivalen

Dua FSA disebut ekivalen jika keduanya dapat mengenali bahasa yang sama

c) Deterministic Finite State Automata
DFA (Deterministic Finite State Automata) aturan perpindahan dapat juga disebut fungsi transisi, yang dilambangkan dengan f.

c) Deterministic Finite State Automata
DFA (Deterministic Finite State Automata) aturan perpindahan dapat juga disebut fungsi transisi, yang dilambangkan dengan f.

Diagram keadaannya :
Diketahui diagram keadaan NFSA.
a. Tentukan tabel keadaannya.
b. Tentukan bahasa yang dikenali oleh NFSA tersebut.
Bahasa yang dikenal : $\{0n, 0n\ 01, 0n\ 11 \mid n \geq 0\}$.

e) Regular Expressions
Ekspresi reguler dalam himpunan I didefinisikan sebagai berikut :
- Simbol ∅ adalah ekspresi reguler
Menggambarkan empty set (himpunan kosong), yaitu himpunan tanpa strings.
- Simbol λ adalah ekspresi reguler
Menggambarkan himpunan $\{\lambda\}$ dimana himpunan mengandung string kosong.
- Simbol x adalah ekspresi reguler kapanpun x ∈ I
Menggambarkan himpunan {x} mengandung string dengan satu simbol x.
- Simbol (AB), (A∪B), dan A* adalah ekspresi reguler kapanpun A dan B merupakan ekspresi reguler.
(AB) menggambarkan rangkaian himpunan yang diwakili oleh A dan B. (A∪B) menggambarkan gabungan himpunan yang diwakili oleh A dan B. A* menggambarkan Kleene closure himpunan yang diwakili oleh A.