

Solusi UAS IF1220 Matematika Diskrit (3 SKS)

Dosen: Rinaldi M
 Rabu, 8 Juni 2026
 Waktu: 100 menit

Berdoalah terlebih dahulu sebelum mengerjakan ujian ini.

1. (Nilai 10) Pada suatu padang rumput berbentuk $grid\ 3 \times 3$, terdapat seekor serigala (S) dan tiga ekor domba, yaitu domba hitam (H), domba putih (P), dan domba kuning (K). Seekor domba terancam dimakan oleh serigala apabila posisi domba tersebut bersebelahan (*adjacent*) secara vertikal atau horizontal dengan posisi serigala. Untuk membantu memahami soal, perhatikan ilustrasi di kanan. Pada contoh tersebut, hanya terdapat satu domba yang terancam dimakan oleh serigala, yaitu domba hitam (H). Tentukanlah berapa banyak cara menyusun posisi serigala dan domba sedemikian sehingga tidak ada domba yang terancam dimakan oleh serigala.

S		K
H	P	

Jawaban:

Kita hitung banyak susunan posisi pada $grid\ 3 \times 3$ sehingga tidak ada domba yang berada tepat di atas, bawah, kiri, atau kanan serigala. Kita bagi berdasarkan posisi serigala:

1. Serigala di tengah
2. Serigala di sisi (bukan pojok)
3. Serigala di pojok

Pada tiap kasus, domba hanya boleh ditempatkan pada sel yang **tidak adjacent** terhadap serigala.

Kasus 1: Serigala di tengah

Pada $grid\ 3 \times 3$, sel tengah memiliki 4 tetangga: atas, bawah, kiri, kanan (berwarna merah)

	S	

Keempat sel itu tidak boleh ditempati domba. Maka sel aman yang tersisa hanya 4 pojok (berwarna putih).

Jumlah cara menempatkan 3 domba berbeda pada 4 sel adalah $P(4, 3) = 4 \times 3 \times 2 = 24$

Kasus 2: Serigala di sisi (bukan pojok)

Ada 4 posisi sisi. Setiap posisi sisi memiliki 3 tetangga yang bersebelahan langsung. Maka terdapat $9 - 1 - 3 = 5$ sel aman untuk domba.

	S	

S		

		S

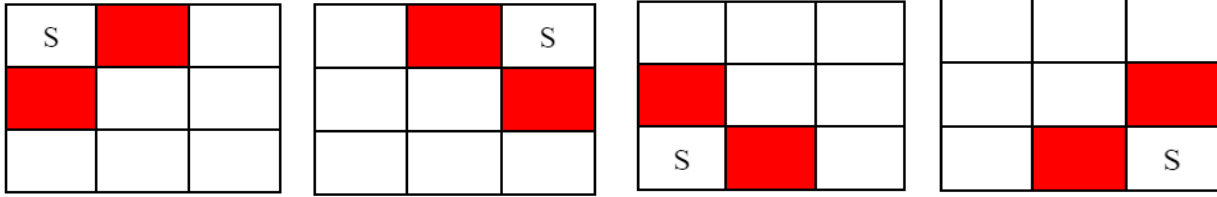
	S	

Jumlah cara menempatkan 3 domba berbeda adalah $P(5, 3) = 5! / (5 - 3)! = 5 \times 4 \times 3 = 60$

Karena ada 4 posisi sisi, maka terdapat $4 \times 60 = 240$ cara menempatkan 3 domba.

Kasus 3: Serigala di pojok

Ada 4 posisi pojok. Setiap pojok memiliki 2 tetangga bersebalahan langsung, maka sel aman sebanyak $9 - 1 - 2 = 6$



Jumlah cara menempatkan 3 domba berbeda adalah $P(6, 3) = 6 \times 5 \times 4 = 120$. Karena ada 4 pojok, maka jumlah cara penempatan domba adalah $4 \times 120 = 480$

Total seluruh susunan: $24 + 240 + 480 = 744$

Jadi, banyak cara menyusun posisi sehingga tidak ada domba yang terancam dimakan serigala adalah 744 cara.

2. (Nilai 10) Seorang administrator laboratorium komputer memiliki 33 unit lisensi *software* yang akan dibagikan kepada tujuh tim mahasiswa untuk proyek pemrograman. Setiap tim harus memperoleh minimal 1 lisensi. Namun, karena beberapa tim bekerja secara kolaboratif:

- total lisensi untuk Tim ke-3 dan Tim ke-4 harus berjumlah 11,
- total lisensi untuk Tim ke-5 dan Tim ke-6 harus berjumlah 9.

Berapa banyak cara administrator dapat membagikan lisensi *software* tersebut kepada ketujuh tim?

Jawaban:

Misalkan banyak lisensi yang diterima Tim ke- i adalah x_i , dengan:

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 = 33$$

dengan syarat:

- $x_3 + x_4 = 11$
- $x_5 + x_6 = 9$

Karena setiap tim minimal menerima 1 lisensi maka langsung distribusikan 1 lisensi untuk masing-masing tim:

- $x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 = 33 - 7 = 26$
- $x_3 + x_4 = 11 - 2 = 9$
- $x_5 + x_6 = 9 - 2 = 7$

Dengan menyulihkan $x_3 + x_4 = 9$ dan $x_5 + x_6 = 7$, maka kita dapat menyederhanakan *constraint* ini menjadi:

$$x_1 + x_2 + (x_3 + x_4) + (x_5 + x_6) + x_7 = 26 \rightarrow x_1 + x_2 + 9 + 7 + x_7 = 26 \rightarrow x_1 + x_2 + x_7 = 10$$

Sistem persamaan menjadi:

- $x_1 + x_2 + x_7 = 10$
- $x_3 + x_4 = 9$
- $x_5 + x_6 = 7$

Untuk $x_1 + x_2 + x_7 = 10$, jumlah cara membagikan: $C(n + r - 1, n - 1) = C(3 + 10 - 1, 3 - 1) = C(12, 2) = 66$

Untuk $x_3 + x_4 = 9$, jumlah cara membagikan: $C(2 + 9 - 1, 2 - 1) = C(10, 1) = 10$

Untuk $x_5 + x_6 = 7$, jumlah cara membagikan: $C(2 + 7 - 1, 2 - 1) = C(8, 1) = 8$

Jumlah cara membagikan seluruhnya: $C(12, 2) \times C(10, 1) \times C(8, 1) = 66 \times 10 \times 8 = 5280$

3. **(Nilai 10)** Tentukan semua nilai x yang memenuhi $x^{86} \equiv 6 \pmod{29}$. Jawaban harus dalam bentuk $x = a + bk$, $k = 0, 1, 2, \dots$

Jawaban: Berdasarkan Teorema Fermat: $x^{28} \equiv 1 \pmod{29}$, sehingga

$$x^{86} \equiv 6 \pmod{29}$$

$$(x^{28})^3(x^2) \equiv 6 \pmod{29}$$

$$(1)^3(x^2) \equiv 6 \pmod{29}$$

$$x^2 \equiv 6 \pmod{29}$$

Diketahui bilangan kuadrat yang memenuhi kekongruenan tersebut adalah 64 karena $64 \equiv (6 \pmod{29})$. Sehingga kekongruenan tersebut dapat ditulis ulang menjadi sebagai berikut:

$$x^2 \equiv 64 \pmod{29}$$

$$x^2 - 64 \equiv 0 \pmod{29}$$

$$(x + 8)(x - 8) \equiv 0 \pmod{29}$$

a. $x + 8 \equiv 0 \pmod{29}$

$$x \equiv -8 \pmod{29}$$

$$x \equiv 21 \pmod{29} \rightarrow x = 21 + 29k$$

b. $x - 8 \equiv 0 \pmod{29}$

$$x \equiv 8 \pmod{29} \rightarrow x = 8 + 29k$$

Maka seluruh nilai x yang memenuhi kekongruenan tersebut adalah $x = 29k + 21$ atau $x = 29k + 8$

4. **(Nilai 10)** Sebuah pesan plainteks disusun oleh alfabet $\{0, 1, 2, \dots, 9, A, B, C, \dots, Z, !\}$. Misalkan sebuah *password* dienkripsi dengan *Caesar Cipher*. Kunci yang digunakan adalah pergeseran $(k) = 28$.

(a) Tentukan rumus enkripsi dan dekripsinya

(b) Dengan menggunakan rumus dari jawaban (a), lakukan dekripsi cipherteks DPTUVC9KKC5R untuk mendapatkan kembali *password* semula.

Jawaban:

(a) Enkripsi: $C = (P + 28) \pmod{37}$

Dekripsi: $P = (C - 28) \pmod{37}$

(b) Dekripsi Cipherteks DPTUVC9KKC5R

Kodekan A = 10, B = 11, C = 12, ..., Z = 35, ! = 36

Gunakan persamaan dekripsi $P = (C - 28) \pmod{37}$

Cipherteks	Nilai C	$P = (C - 28) \pmod{37}$	Plainteks
D	13	22	M
P	25	34	Y
T	29	1	1
U	30	2	2
V	31	3	3
C	12	21	L
9	9	18	I
K	20	29	T
K	20	29	T
C	12	21	L
5	5	14	E
R	27	36	!

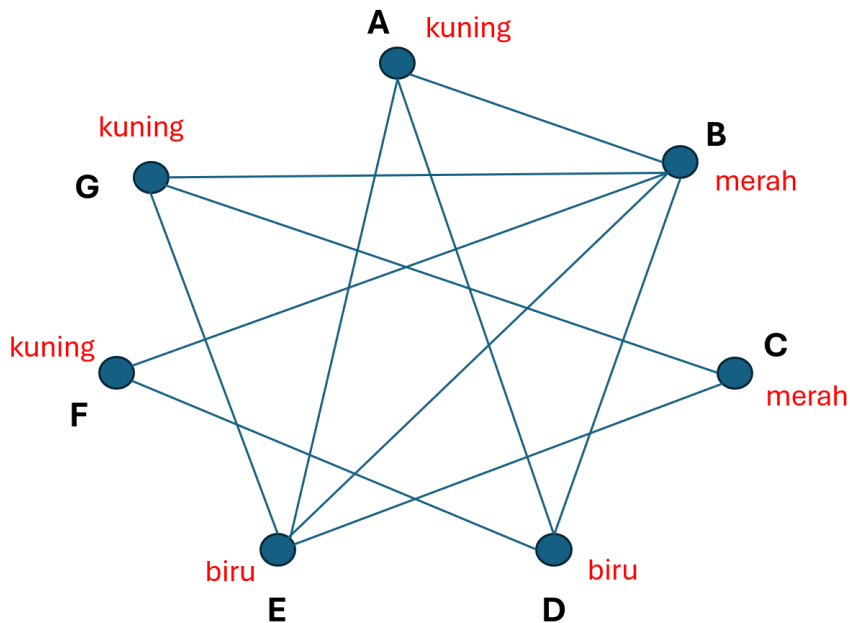
Hasil plainteks: MY123LITTLE!

5. (Nilai 10) Sebuah *compiler* sedang melakukan optimasi penggunaan register CPU untuk 7 variabel program: A, B, C, D, E, F , dan G . Dua variabel yang aktif pada waktu yang sama (*live simultaneously*) tidak dapat ditempatkan pada register yang sama, seperti pada tabel berikut:

Variabel	Tidak dapat ditempatkan pada register yang sama bersama variabel:
A	B, D
B	A, D, E, F, G
C	E, G
D	A, F, B
E	B, C, G
F	B, D
G	C, E, B

Compiler ingin menggunakan jumlah register seminimal mungkin agar kode yang dihasilkan lebih efisien. Tentukan berapa jumlah minimum register yang diperlukan

Jawaban: Soal ini adalah mencari bilangan kromatis graf dengan 7 buah simpul



Bilangan kromatis graf, $\chi(G) = 3$. Jadi dibutuhkan minimal 3 register, yaitu:

Register 1: variabel A, F, G

Register 2: variabel B, C

Register 3: variabel D, E

6. (Nilai 8+2) Diberikan graf G dengan himpunan simpul: $V = \{A, B, C, D, E, F, G, H, I, J, K, L\}$ dan himpunan sisi E :
- $E = \{ (A, B), (A, C), (A, D), (A, E), (B, C), (B, F), (B, G), (C, D), (C, G), (C, H), (D, E), (D, H), (D, I), (E, I), (E, J), (F, G), (F, K), (G, H), (G, K), (G, L), (H, I), (H, L), (I, J), (I, L), (J, L), (K, L) \}$
- Simpul-simpul graf akan diwarnai dengan Algoritma Welsh-Powell sehingga dua simpul bertetangga memiliki warna yang berbeda. Urutan warna pada setiap lelaran (iterasi) algoritma adalah *mejikuhibiniu*:
- Lelaran 1 warna 1 = Merah
 Lelaran 2 warna 2 = Jingga
 Lelaran 3 warna 3 = Kuning
 dst
- (a) Tuliskan simpul-simpul yang diwarnai dengan warna 1 pada iterasi 1, simpul-simpul yang diwarnai dengan warna 2 pada iterasi 2, dst.
 Contoh jawaban: Lelaran 1 warna merah: A, H, I, L
 Lelaran 2 warna jingga: B, D, G, J, K
 dst
- (b) Berapa bilangan kromatis graf tersebut?

Jawaban:

- (a) Urutkan simpul-simpul berdasarkan derajatnya, dari besar ke kecil:

Simpul	Derajat
G	6
H	5
C	5
I	5
L	5
D	5
A	4
B	4
E	4
F	3
J	3
K	3

Lelaran 1: Warna 1 (merah)

Mulai dari simpul dengan derajat terbesar pertama: $G \rightarrow$ merah

Simpul pada urutan berikutnya yang tidak bertetangga dengan G dan dapat diwarnai merah adalah I, A , maka $\{G, I, A\}$ berwarna merah.

Lelaran 2: Warna 2 (biru)

Simpul belum diwarnai di dalam urutan di dalam tabel: $H \rightarrow$ Biru

Simpul pada urutan berikutnya yang tidak bertetangga dengan H dan dapat diwarnai biru adalah L, B , dan E , maka $\{H, L, B, E\}$ berwarna biru.

Lelaran 3: Warna 3 (hijau)

Simpul belum diwarnai di dalam urutan di dalam tabel: $C \rightarrow$ Hijau

Simpul pada urutan berikutnya yang tidak bertetangga dengan C dan dapat diwarnai hijau adalah F dan J , maka $\{C, F, J\}$ berwarna biru.

Lelaran 4: Warna 4 (Kuning): simpul D dan J

Hasil akhir

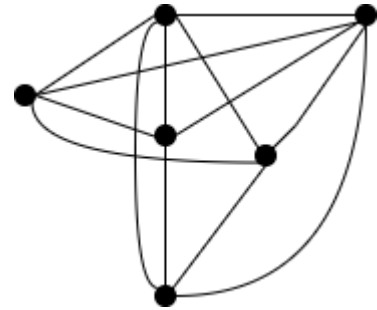
Warna	Simpul
Merah	G, I, A

Warna	Simpul
Biru	H, L, B, E
Hijau	C, F, J
Kuning	D, J

(b) **Bilangan Kromatik** $\chi(G) = 4$

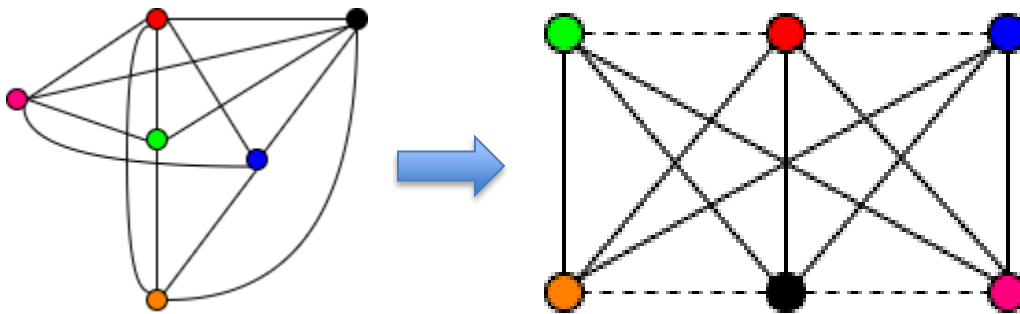
Cataan: Jawaban di atas bisa berbeda tergantung pengurutan simpul-simpulnya, namun bilangan kromatis tyetap sama yaitu 4.

7. **(Nilai 10)** Seorang mahasiswa Elektro akan membuat rangkaian *integrated circuit* (IC) pada sebuah PCB (*Printed Circuit Board*). Desainnya tampak seperti pada graf di sebelah kanan. Salah satu syarat yang harus dipenuhi desain rangkaian IC harus dapat diubah sedemikian rupa sehingga semua kawat tidak saling bersilangan karena dapat menimbulkan korsleting dan mengganggu sinyal. Tentukan apakah desain sudah memenuhi syarat tersebut. Jika ya, gambarkan salah satu susunan rangkaian IC yang memenuhi syarat. Jika tidak, jelaskan alasannya dengan tegas menggunakan Teorema Kuratowski.



Jawaban:

Syarat agar rangkaian IC tidak menimbulkan korsleting adalah representasi grafnya planar. Berdasarkan teorema Kuratowski, graf tersebut mengandung upagraf yang isomorfik dengan $K_{3,3}$ sehingga graf tidak planar dan tidak mungkin tidak bersilangan.



8. **(Nilai 10)** Sebuah pesan telah dikompresi dengan algoritma Huffman. Kode Huffman untuk setiap simbol di dalam pesan adalah sebagai berikut: A = 11, B = 101, E = 0110, I = 1000, K = 000, L = 0111, M = 1001, N = 001, S = 0100, Y = 0101

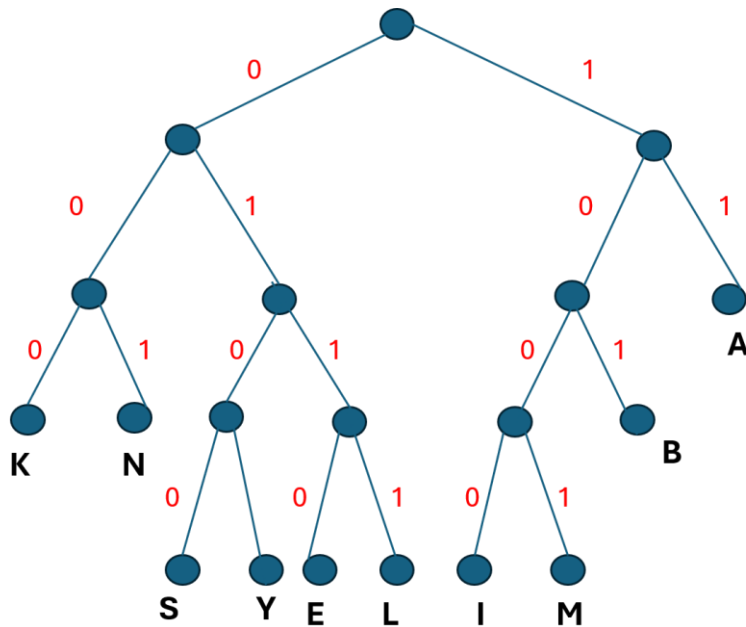
Hasil *encoding* pesan tersebut dengan kode Huffman adalah:

1011110111010001011110101100111100010011110001101111001

- (a) Gambarkan pohon *prefix* (pohon Huffman) yang bersesuaian dengan kode Huffman tersebut
- (b) Lakukan dekomposisi (*decoding*) string biner tersebut menjadi pesan semula.

Jawaban:

- (a) Pohon Huffman



(b) Hasil decoding:

101 11 101 11 0100 0101 11 101 0110 11 001 11 1000 1001 11 1000 11 0111 1001

B A B A S Y A B E A N A I M A I A L M

BABASYABEANAIMAIALM

9. (Nilai 15) Diberikan tiga buah potongan algoritma berikut:

<p>Algoritma A(n):</p> <ol style="list-style-type: none"> 1. count ← 0 2. for i ← 1 to n do 3. for j ← 1 to i do 4. count ← count + 1 5. endfor 6. endfor 7. return count 	<p>Algoritma B(n):</p> <ol style="list-style-type: none"> 1. count ← 0 2. for i ← 1 to n do 3. j ← 1 4. while j ≤ n do 5. count ← count + 1 6. j ← 2*j 7. endwhile 8. endfor 9. return count 	<p>Algoritma C(n):</p> <ol style="list-style-type: none"> 1. count ← 0 2. for i ← 1 to n do 3. for j ← 1 to i*i do 4. for k ← 1 to j do 5. count ← count + 1 6. endfor 7. endfor 8. endfor 9. return count
--	---	---

Tentukan:

(a) Fungsi $T(n)$ berdasarkan jumlah operasi tambah (+) pada instruksi $\text{count} \leftarrow \text{count} + 1$

(b) Notasi Big-O, Big-Omega, dan Big-Tetha

(c) Algoritma apa yang paling lambat waktunya?

Jawaban:

(a) Algoritma A(n):

$$T(n) = \sum_{i=1}^n \sum_{j=1}^i 1 = \sum_{i=1}^n i = \frac{n(n+1)}{2} = \frac{n^2+n}{2}$$

Algoritma B(n):

Kalang terdalam: $j \leftarrow 2*j$

Nilai j : 1, 2, 4, 8, ...

Loop berhenti saat: $2^k > n$, sehingga jumlah iterasi: $k = \lfloor \log_2 n \rfloor + 1$

$$\begin{aligned}
 T(n) &= \sum_{i=1}^n (\log_2 n + 1) \\
 &= n(\log_2 n + 1) \\
 &= n \log_2 n + n
 \end{aligned}$$

Algoritma C(n):

$$T(n) = \sum_{i=1}^n \sum_{j=1}^{i^2} j =$$

Gunakan rumus:

$$\sum_{j=1}^m j = \frac{m(m+1)}{2}$$

dengan $m = i^2$:

$$\begin{aligned}
 T(n) &= \sum_{i=1}^n \frac{i^2(i^2 + 1)}{2} \\
 &= \frac{1}{2} \sum_{i=1}^n (i^4 + i^2)
 \end{aligned}$$

$$= \frac{1}{2} \left(\sum_{i=1}^n (i^4) + \sum_{i=1}^n (i^2) \right)$$

(sampai jawaban dalam bentuk ini sudah cukup. Jika dilanjutkan maka:

Jumlah pangkat empat:

$$\sum_{i=1}^n i^4 = \frac{n(n+1)(2n+1)(3n^2+3n-1)}{30}$$

Jumlah kuadrat:

$$\sum_{i=1}^n i^2 = \frac{n(n+1)(2n+1)}{6}$$

Maka:

$$T = \frac{1}{2} \left\{ \frac{n(n+1)(2n+1)(3n^2+3n-1)}{30} + \frac{n(n+1)(2n+1)}{6} \right\}$$

$$T = \frac{1}{2} \left\{ \frac{n(n+1)(2n+1)[(3n^2+3n-1)+5]}{30} \right\}$$

$$T = \frac{1}{2} \cdot \frac{n(n+1)(2n+1)(3n^2+3n+4)}{30}$$

$$T = \frac{n(n+1)(2n+1)(3n^2+3n+4)}{60}$$

(b) Algoritma A(n): $T(n) = \frac{n^2+n}{2}$ adalah $O(n^2)$, $\Omega(n^2)$, $\Theta(n^2)$

Algoritma B(n): $T(n) = n \log_2 n + n$ adalah $O(n \log n)$, $\Omega(n \log n)$, $\Theta(n \log n)$

Algoritma C(n): $T(n) = \frac{1}{2} (\sum_{i=1}^n (i^4) + \sum_{i=1}^n (i^2))$ adalah $O(n^5)$, $\Omega(n^5)$, $\Theta(n^5)$

(c) Algoritma yang paling lambat adalah algoritma C(n)

$$n \log n < n^2 < n^5$$

10. (Nilai 5) Urutkan fungsi berikut berdasarkan kompleksitas waktu asimptotiknya (Big-O) dari paling cepat hingga paling lambat:

$$T_1(n) = n^2 \log n + n^3$$

$$T_2(n) = 2^{\sqrt{n}} + n \log n$$

$$T_3(n) = n(\log n)^2$$

$$T_4(n) = \log(\log n) + n^2$$

$$T_5(n) = 2 \log(n!)$$

$$T_6(n) = n^3 + 3^n$$

Jawaban:

$$T_1(n) = n^2 \log n + n^3 \text{ adalah } O(n^3)$$

$$T_2(n) = 2^{\sqrt{n}} + n \log n \text{ adalah } O(2^{\sqrt{n}})$$

$$T_3(n) = n(\log n)^2 \text{ adalah } O(n \log^2 n)$$

$$T_4(n) = \log(\log n) + n^2 \text{ adalah } O(n^2)$$

$$T_5(n) = 2 \log(n!) \text{ adalah } O(n \log n) \text{ sebab } \log(n!) = O(n \log n)$$

$$T_6(n) = n^3 + 3^n \text{ adalah } O(3^n)$$

Urutan:

$$n \log n < n(\log n)^2 < n^2 < n^3 < 2^{\sqrt{n}} < 3^n$$

atau:

$$T_5 < T_3 < T_4 < T_1 < T_2 < T_6$$

11. (Nilai 2) Apa prediksi (atau harapan) nilai anda untuk mata kuliah ini? (A/AB/B/BC/C/D/E)

TOTAL NILAI = 102