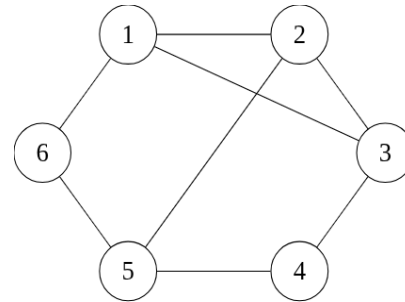


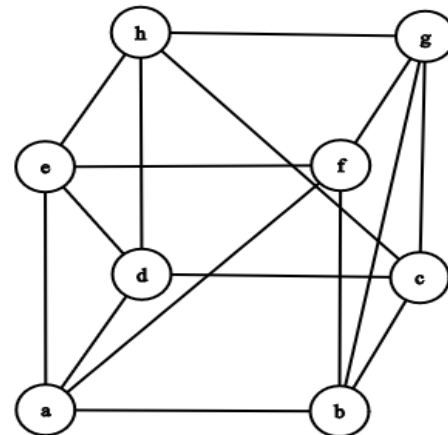
Kuis ke-4 IF1220 Matematika Diskrit (3 SKS) – Graf, Pohon, dan Kompleksitas Algoritma  
 Dosen: Rinaldi M  
 Selasa, 26 Mei 2026  
 Waktu: 90 menit

- (Nilai 10)** Sebuah negara memiliki 10 bandara. Pemerintah ingin membangun jalur penerbangan langsung sehingga setiap bandara terhubung langsung dengan tepat 5 bandara lainnya.
  - Berapa jumlah jalur penerbangan yang terbentuk?
  - Apakah jaringan penerbangan seperti ini mungkin dibuat? Jelaskan menggunakan *lemma* jabat tangan.
  - Tanpa menggambar grafnya, apakah jaringan ini planar? Jelaskan dengan teori graf.

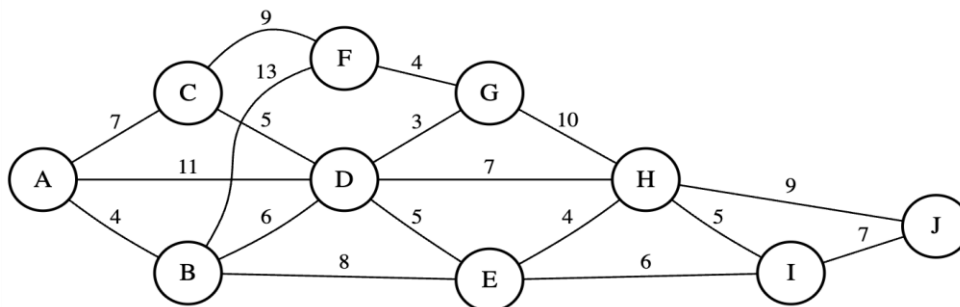
- (Nilai 10)** Diberikan sebuah graf  $G$  seperti gambar di samping
  - Apakah graf  $G$  tersebut merupakan graf bipartit? Jelaskan alasannya.
  - Tentukan bilangan kromatik ( $\chi(G)$ ) dari graf  $G$  dan berikan satu contoh kombinasi pembagian warnanya (gunakan warna mejikuhibiniu)!



- (Nilai 10)** Apakah graf  $G$  di kanan ini merupakan graf planar? Jika ya, gambarkanlah dalam bentuk planar. Jika tidak, buktikan dengan teorema Kuratowski.



- (Nilai 15)** Tentukanlah pohon merentang minimum serta bobot total akhir dari graf pada Gambar di bawah dengan menggunakan Algoritma Kruskal! Untuk setiap langkah, tuliskan sisi yang ditambahkan, bobot dari sisi tersebut, beserta gambar pohon rentang yang terbentuk sampai dengan langkah tersebut. Jika terdapat beberapa sisi dengan bobot yang sama, pilihlah sisi yang nama simpul-simpulnya lebih awal secara alfabetis (misalnya, A-B dipilih sebelum A-C, dan A-C dipilih sebelum B-C). Jawaban tanpa langkah yang lengkap tidak akan mendapatkan nilai penuh.



- (Nilai 15)** Budi ingin mengirimkan spesifikasi algoritma kepada rekannya secara efisien menggunakan Kode Huffman. Jika pesan yang dikirim adalah 'MINIMUM SPANNING TREE', tentukan kode Huffman untuk setiap karakter (termasuk spasi) dan hitung panjang total kode pesan dalam satuan bit!

6. **(Nilai 10)** Diberikan data: 47, 90, 67, 50, 95, 69, 39, 45, 22.
- Buatlah *Binary Search Tree* (BST) dari data tersebut dengan 47 sebagai akar.
  - Dari BST yang dibuat pada bagian (a), berapa banyak perbandingan yang dilakukan untuk menemukan 67? Tuliskan langkah-langkahnya.
  - Dari BST yang dibuat pada bagian (a), berikan sekuens penelusuran dengan cara *postorder*.
7. **(Nilai 15)** Diberikan dua buah kode program Python di bawah ini:

<pre>def algo_P(n):     total = 0     for i in range(1, n + 1):         for j in range(1, n + 1):             total += 1     return total</pre>	<pre>def algo_Q(n):     total = 0     i = 1     while i &lt;= n:         for j in range(1, n + 1):             total += 1         i *= 2     return total</pre>
---	---

- Tentukan  $T(n)$  secara eksak untuk kedua algoritma di atas!
  - Hitung  $T(8)$  masing-masing!
  - Mulai dari  $n$  berapa algoritma\_Q lebih sangkil (efisien) dari algoritma\_P?
  - Tentukan Big-O, Big- $\Omega$ , Big- $\Theta$  masing-masing algoritma.
8. **(Nilai 10)** Diberikan empat blok kode program Python berikut yang menerima input bilangan bulat positif  $n > 1$ . Dalam analisis ini, hitunglah kompleksitas berdasarkan berapa kali blok di dalam perulangan terdalam dieksekusi.
- Tentukan rumus  $T(n)$  untuk menentukan jumlah operasi secara eksak untuk setiap algoritma!
  - Tentukan notasi asimtotik Big-O, lalu urutkan dari paling efisien hingga tidak efisien!

<pre>def fungsi_A(n):     res = 0     i = 0     while i &lt; 2 * n:         res += (i % 2)         i += 2     return res</pre>	<pre>def fungsi_B(n):     res = 0     for i in range(n, 0, -1):         j = 1         while j &lt;= n:             res += (i * j)             j += 1     return res</pre>
<pre>def fungsi_C(n):     res = 0     k = n     while k &gt; 0:         res += (k % 2)         k = k // 2     return res</pre>	<pre>def fungsi_D(n):     res = 0     for i in range(1, n + 1):         p = 1         while p &lt; n:             res += i             p = p * 3     return res</pre>

9. **(Nilai 5)** Diberikan 5 algoritma dengan jumlah operasi  $T(n)$  sebagai berikut. Nyatakan Big-O masing-masing, lalu urutkan dari paling efisien ke paling tidak efisien!

$$A: T_A(n) = (n + 2)^2 - 4n$$

$$B: T_B(n) = n \log(n^4) + n^{1.5}$$

$$C: T_C(n) = \sum_{i=1}^n \left\lceil \frac{n}{i} \right\rceil$$

$$D: T_D(n) = 6(\log n)^3 + n$$

$$E: T_E(n) = 4^{\log_2 n} + n \log n$$