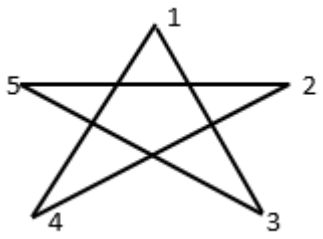
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung

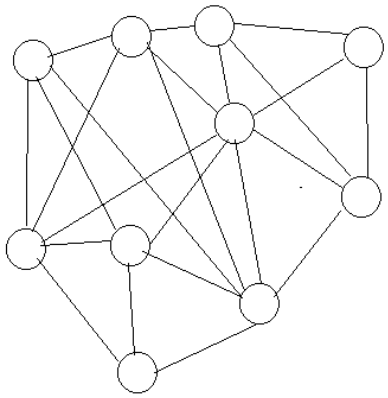
Nama :……
NIM :……
T.tangan:……

Solusi Kuis ke-4 IF2120 Matematika Diskrit (3 SKS) – Graf, Pohon, Kompleksitas Algoritma
Dosen: Rinaldi Munir, Harlili, Judhi Santoso
Kamis, 30 November 2017
Waktu: 90 menit

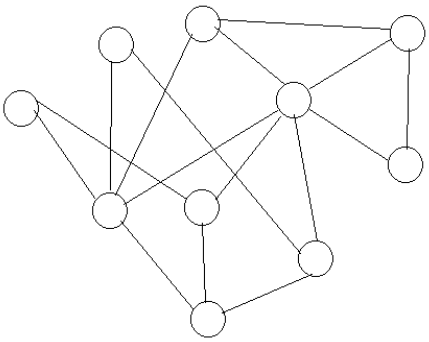
1. Tentukan apakah graf-graf di bawah ini planar atau tidak, jika tidak planar tunjukkan di dalam graf tersebut bagian graf yang mengandung K_5 atau $K_{3,3}$ (gambarkan dengan spidol/tinta merah atau garis tebal). Jika planar, maka gambar bentuk planarnya!



Gambar 1



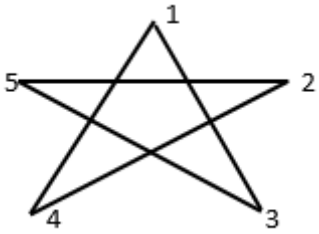
Gambar 2



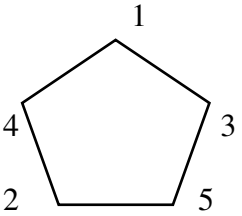
Gambar 3

Jawaban:

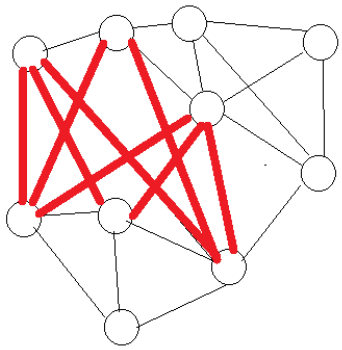
Gambar 1: Misalkan penomoran sisi-sisi pada gambar di atas:



Graf tersebut planar dengan graf bidang seperti di bawah ini:



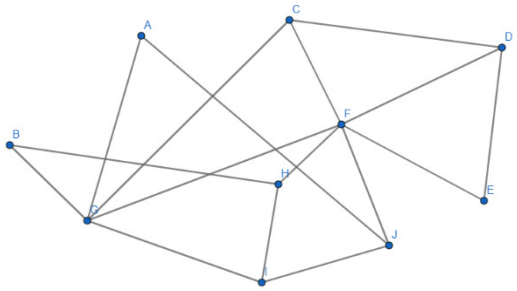
Gambar 2:



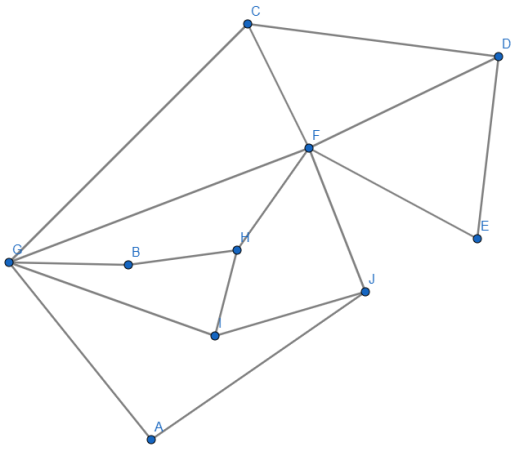
Bukan graf planar karena mengandung upagraf $K_{3,3}$

Gambar 3: Merupakan graf planar karena tidak mengandung upagraf K_5 ataupun $K_{3,3}$.

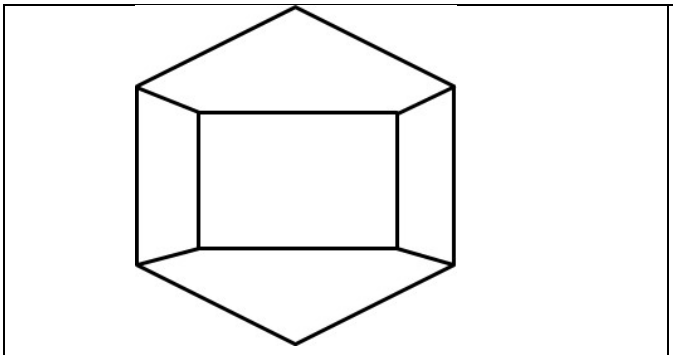
Awal:



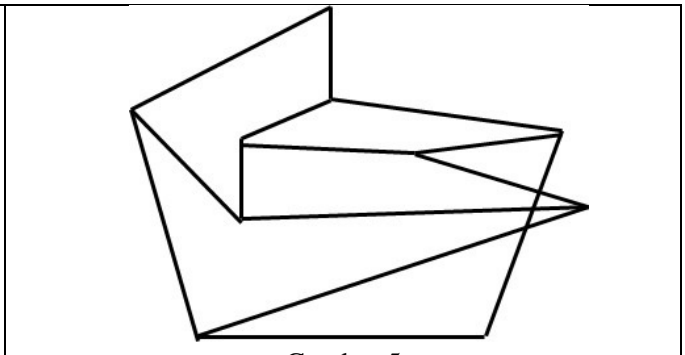
Akhir:



2. Apakah kedua graf dibawah ini isomorfik? Jika iya, tandai sisi-sisi yang berkoresponden antara A dan B. Jika tidak, buktikan.

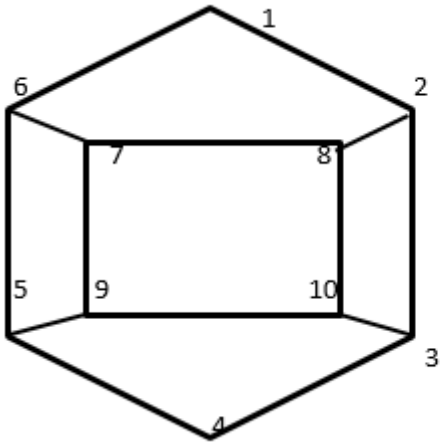


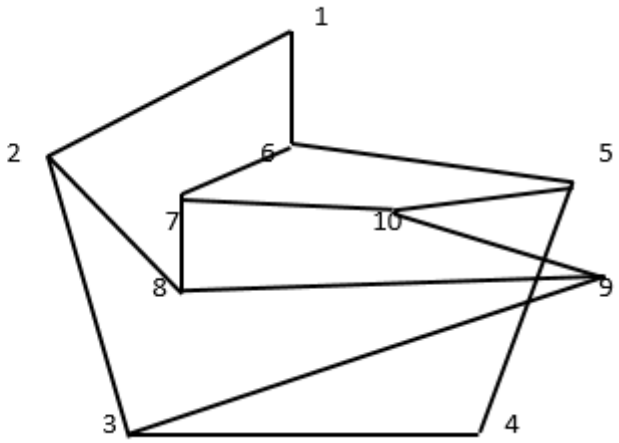
Gambar 4



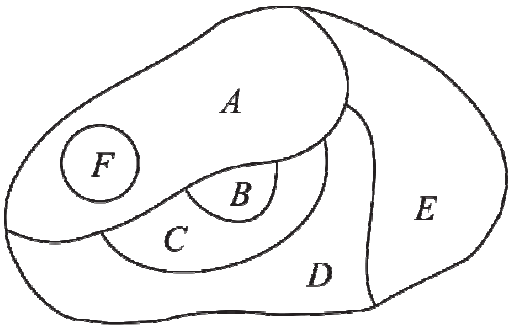
Gambar 5

Jawaban: Graf A dan B isomorfik, dengan sisi-sisi yang berkoresponden

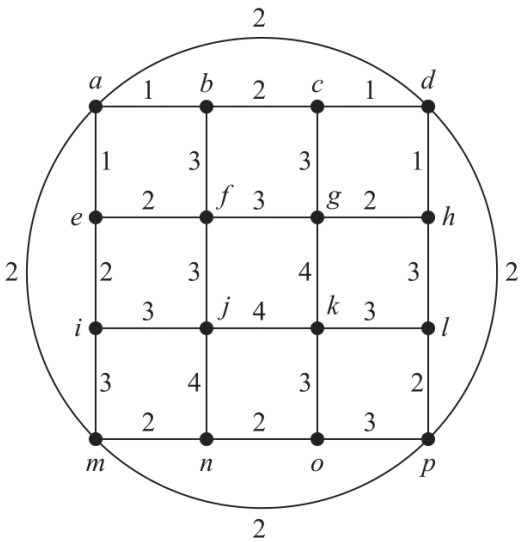




3. Lihat peta pada Gambar 6. Gambar perta tersebut pada lembar jawaban anda. Di atas gambar peta tersebut, gambarkan graf dual peta tersebut, kemudian tentukan jumlah warna minimal yang dibutuhkan sehingga tidak ada dua daerah bersebelahan yang berwarna sama.

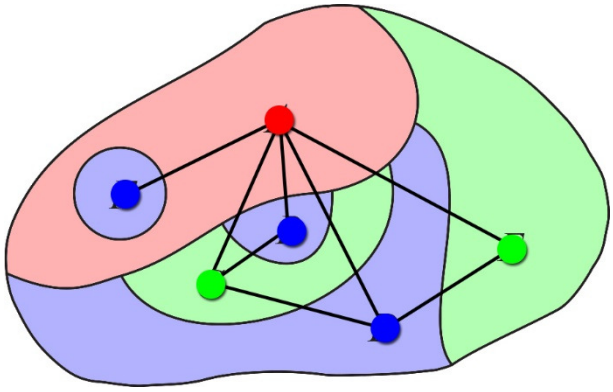


Gambar 6



Gambar 7

Jawaban:



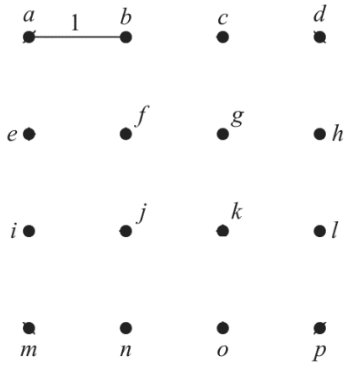
Dibutuhkan minimal 3 warna untuk mewarnai peta.

4. (a) Gunakan algoritma Prim untuk mencari *minimum spanning tree* dari graf Gambar 7. Jika ada beberapa sisi dengan bobot sama, prioritaskan sisi dengan label terkecil ((a, b) < (c, d); (e, i) < (g, h); dll.)
(b) Tentukan bobot total dari *minimum spanning tree* tersebut.

Jawaban:

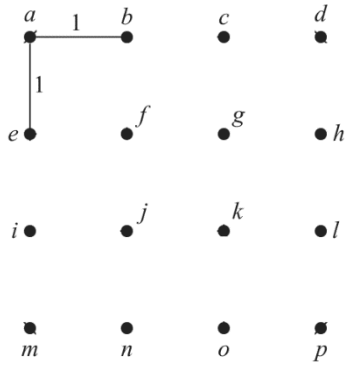
a. Tahapan pembentukan *minimum spanning tree:*

1



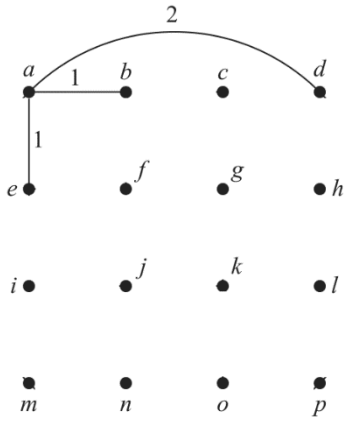
Tambah (a, b)
Bobot = 1

2



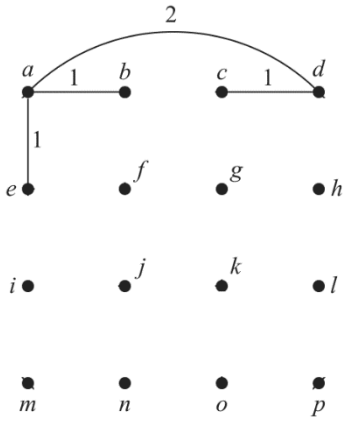
Tambah (a, e)
Bobot = 2

3



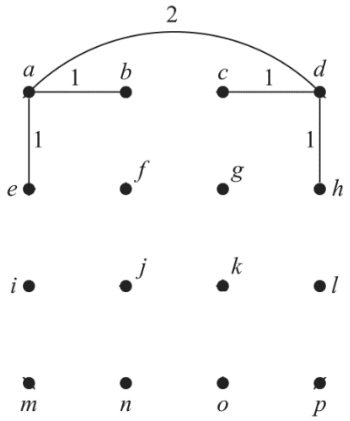
Tambah (a, d)
Bobot = 4

4



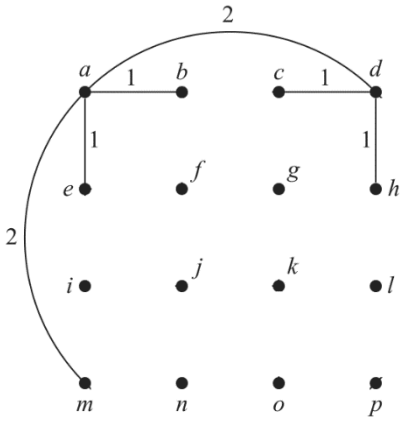
Tambah (c, d)
Bobot = 5

5



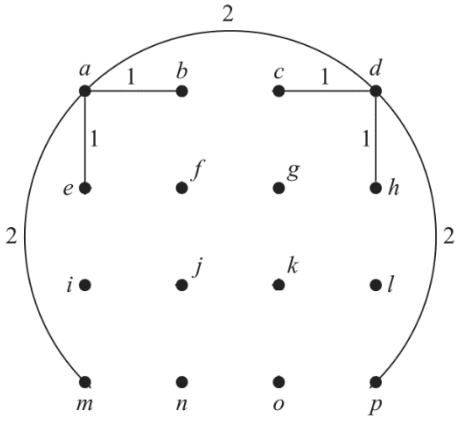
Tambah (a, d)
Bobot = 6

6



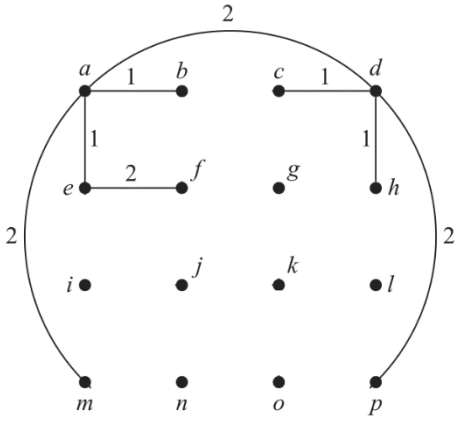
Tambah (a, m)
Bobot = 8

7



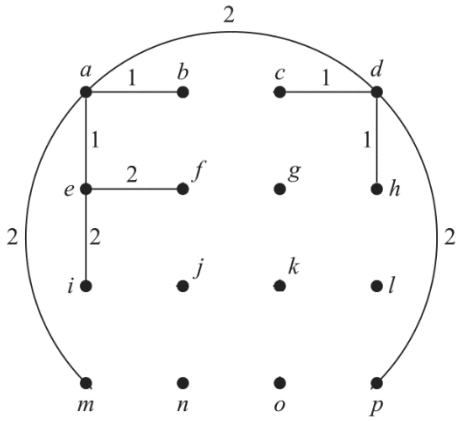
Tambah (d, p)
Bobot = 10

8



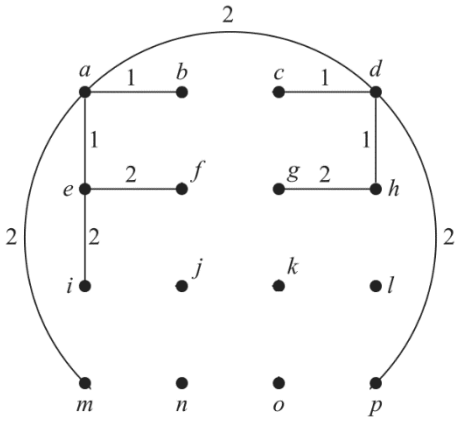
Tambah (e, f)
Bobot = 12

9



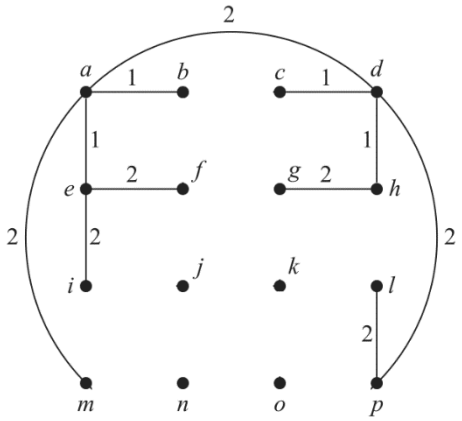
Tambah (e, i)
Bobot = 14

10



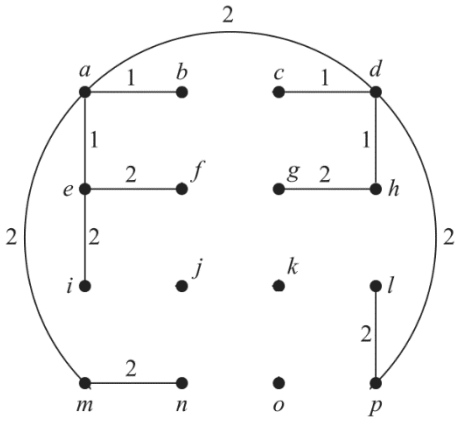
Tambah (g, h)
Bobot = 16

11



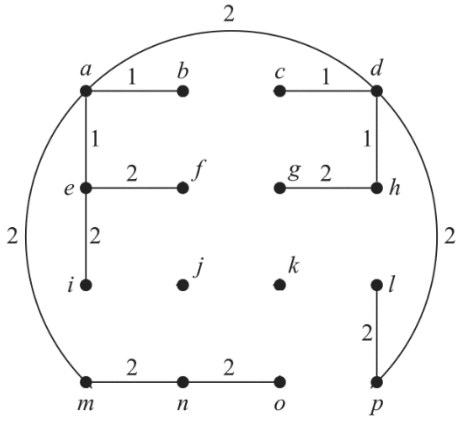
Tambah (l, p)
Bobot = 18

12



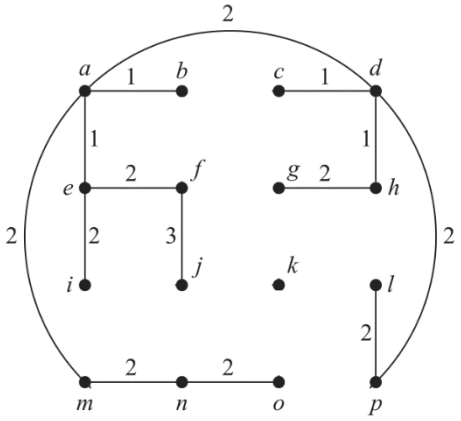
Tambah (m, n)
Bobot = 20

13



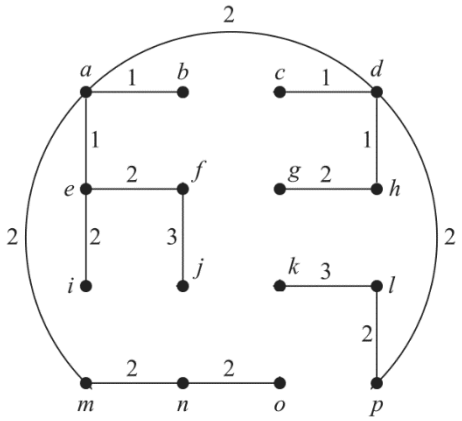
Tambah (n, o)
Bobot = 22

14



Tambah (f, j)
Bobot = 25

15

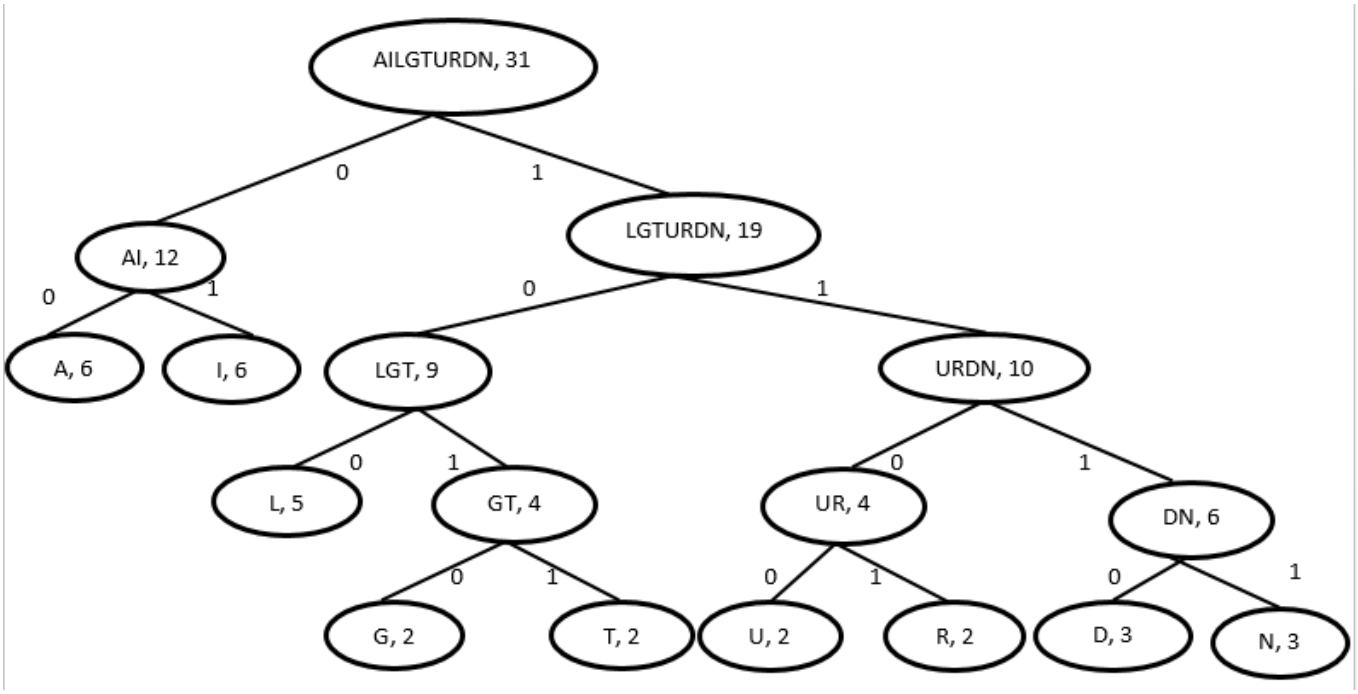


Tambah (k, l)
Bobot = 28

b. Total bobot = 28

5. Terdapat suatu pesan pesan sebaga berikut: ULARLARILAGIULATDIDINDINGLANTAI
Anda diminta melakukan kompresi data dnegan algoritma Huffman. Tentukan kode Huffman untuk setiap karakter di dalam pesan tersebut dan hitung ukuran bit yang dihasilkan jika pesan diubah menjadi kode Huffman (hitung panjang bit total saja). Gambarkan pohon Huffman-nya.

Jawaban: Pohon Huffman untuk teks 'ULARLARILAGIULATDIDINDINGLANTAI':



Tabel frekuensi untuk teks 'ULARLARILAGIULATDIDINDINGLANTAI':

Simbol	Frekuensi	Peluang	Kode Huffmann
U	2	2/31	1100
L	5	5/31	100
A	6	6/31	00

R	2	2/31	1101
I	6	6/31	01
G	2	2/31	1010
T	2	2/31	1011
D	3	3/31	1110
N	3	3/31	1111

6. Untuk soal a sampai d, sederhanakan bentuk dibawah ini sehingga menjadi O(f(n)). Buktikan untuk soal e.
   a. $O(N^2 + \log N + N \log N)$
   b. $O(N^3 + NM + M^3)$
   c. $O(N(\log N)^2 + N^2(\log N))$
   d. $O(N^3 + N^2(\log N))$
   e. Tunjukkan $T(n) = 2n^2 + 3n^2 \log n + 2 \log^2 n = O(n^2 \log n)$

Jawaban:

a. $O(N^2)$

b. $O(N^3 + NM + M^3)$

c. $O(N^2(\log N))$

d. $O(N^3)$

e. $T(n) = 2n^2 + 3n^2 \log n + 2 \log^2 n \leq 2n^2 \log n + 3n^2 \log n + 2n^2 \log n = 7n^2 \log n$ untuk $n \geq 1$
Dapat diambil $C = 7$ dan $n_0 = 1$ untuk memperlihatkan bahwa
$T(n) = 2n^2 + 3n^2 \log n + 2 \log^2 n = O(n^2 \log n)$

7. Hitunglah kompleksitas dari algoritma (dalam notasi *big-O)* berikut dihitung dari operasi *assignment* larik (*array*) dengan nilai Boolean.

```
var
     x    : array[1..10000] of boolean;
     i    : integer;
     N    : integer;
begin
     readln(N);
     x[0] := false;
     for i := 2 to N do
     begin
          x[i] := true;
     end;
     for i := 2 to N do
     begin
          if (x[i]) then
          begin
               for j := 2 to (N div i) do
               begin
                    x[i*j] := false;
               end;
          end;
     end;
end.
```

Jawaban: Kompleksitas algoritma ini adalah $O(N \log N)$. Algoritma ini melakukan pendataan bilangan yang habis dibagi 2, 3, 5, dan seterusnya (deret prima selanjutnya) hingga prima sebelum N. Algoritma ini sangat terkenal dengan nama *Sieve of Erathotenes.*