

1. Misalkan sebuah barisan bilangan riil didefinisikan dengan relasi berikut:  $x_1 = \sqrt{2}$  dan  $x_{n+1} = \sqrt{2 + x_n}$  untuk  $n \geq 1$ .  
Gunakan prinsip induksi matematika untuk membuktikan bahwa  $x_n < 2$  untuk semua  $n \geq 1$ . (Nilai = 10)

**Penyelesaian:**

Misalkan  $P(n)$  adalah proposisi bahwa  $x_n < 2$

**Basis Induksi:**

Untuk  $n = 1$ , diketahui  $x_1 = \sqrt{2}$ . Karena  $\sqrt{2} < \sqrt{4} = 2$ , maka  $x_1 < 2$ .  $P(1)$  terbukti benar.

**Langkah Induksi:**

Asumsikan  $P(n)$  benar untuk suatu bilangan bulat  $n \geq 1$ , yaitu:

$$x_n < 2 \text{ (Hipotesis Induksi)}$$

Akan dibuktikan bahwa  $P(n+1)$  juga benar, yaitu  $x_{n+1} < 2$

Berdasarkan definisi barisan pada soal:

$$x_{n+1} = \sqrt{2 + x_n}$$

Dengan mensubstitusi batas atas dari hipotesis induksi ( $x_n < 2$ ), kita peroleh:

$$x_{n+1} < \sqrt{2 + 2}$$

$$x_{n+1} < \sqrt{4}$$

$$x_{n+1} < 2$$

Karena  $x_{n+1} < 2$ , maka  $P(n+1)$  terbukti benar. Karena langkah basis dan langkah induksi telah terbukti, berdasarkan prinsip induksi matematika,  $x_n < 2$  untuk semua  $n \geq 1$ .

2. Buat sebuah poligon dengan  $n$  sisi. Pilih satu titik sudut, misalkan titik  $X$ . Hubungkan titik  $X$  dengan tiap titik sudut lain di dalam poligon tersebut. Buktikan dengan induksi matematika bahwa banyaknya segitiga yang tercipta adalah  $n - 2$ .

(Nilai = 10)

**Penyelesaian:**

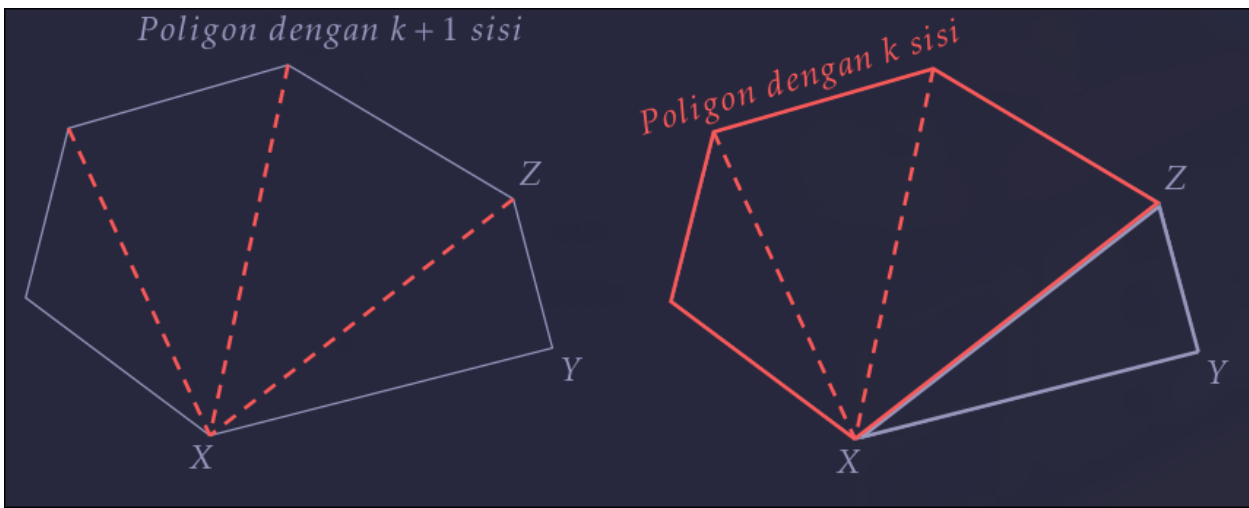
Poligon dengan sisi paling sedikit adalah segitiga, sehingga nilai  $n$  terkecil di mana pernyataan tersebut berlaku adalah  $n = 3$ . Kita akan gunakan ini sebagai basis induksi ketimbang menggunakan  $n = 1$  atau  $n = 2$  (tidak ada poligon bersisi 1 ataupun 2).

*Basis induksi ( $n = 3$ ):*

Cukup jelas bahwa kalau kita pilih suatu titik sudut pada segitiga lalu menghubungkannya dengan titik-titik sudut lainnya, maka akan terbentuk  $3 - 2 = 1$  segitiga, yakni segitiga itu sendiri.

*Langkah induksi:*

Kita asumsikan bahwa jika kita hubungkan tiap titik sudut suatu poligon bersisi  $n$  dengan titik  $X$ , maka akan terbentuk  $n-2$  segitiga. Maka akan kita buktikan bahwa dengan melakukan hal yang sama pada poligon dengan  $n + 1$  sisi, akan terbentuk  $n - 1$  segitiga.



Misalkan bangun di atas adalah poligon dengan  $n+1$  sisi. Kita cuil sebuah segitiga  $XYZ$  dari poligon tersebut seperti pada gambar sebelah kanan. Maka kita dapatkan suatu poligon dengan  $n$  sisi. Berdasarkan hipotesis induksi, poligon dengan  $n$  sisi tersebut punya  $n - 2$  segitiga. Jika ditambah segitiga  $XYZ$ , maka ada  $(n - 2) + 1 = n - 1$  segitiga.

Karena poligon bersisi  $n + 1$  kita tadi pada dasarnya terdiri atas poligon bersisi  $n$  dan segitiga  $XYZ$ , maka banyaknya segitiga yang terbentuk dalam poligon tersebut adalah  $n - 1$ .

3. Diketahui sebuah fungsi Boolean  $F$  dengan empat variabel  $A, B, C, D$  yang telah disederhanakan menjadi bentuk berikut: (Nilai = 20)

$$F(A, B, C, D) = AB' + B'C + AC'D'$$

- Gambarkan rangkaian logika fungsi  $F$
- Nyatakan fungsi  $F$  tersebut ke dalam bentuk kanonik SOP menggunakan notasi  $\sum m$ .
- Nyatakan fungsi  $F$  tersebut ke dalam bentuk kanonik POS menggunakan notasi  $\prod M$ .
- Gambarkan Peta Karnaugh (K-Map) 4 variabel untuk fungsi tersebut dan buktikan apakah penyederhanaan di atas sudah paling minimal atau masih bisa dipangkas lagi!

**Penyelesaian:**

**Suku 1:**  $AB'$  (Hilang  $C$  dan  $D$ )

$$AB'(C + C')(D + D') = (AB'C + AB'C')(D + D') = AB'CD + AB'CD' + AB'C'D + AB'C'D'$$

Biner: 1011,1010,1001,1000  $\rightarrow$  m11,m10,m9,m8

**Suku 2:**  $B'C$  (Hilang  $A$  dan  $D$ )

$$(A + A')B'C(D + D') = (AB'C + A'B'C)(D + D') = AB'CD + AB'CD' + A'B'CD + A'B'CD'$$

Biner: 1011,1010,0011,0010  $\rightarrow$  m11,m10,m3,m2

**Suku 3:**  $AC'D'$  (Hilang  $B$ )

$$A(B + B')C'D' = ABC'D' + AB'C'D'$$

Biner: 1100,1000  $\rightarrow$  m12,m8

**Bentuk SOP Kanonik:**  $F(A, B, C, D) = \sum(2, 3, 8, 9, 10, 11, 12)$

POS didapatkan dari daftar yang tidak masuk SOP kanonik

$$F(A, B, C, D) = \prod M(0, 1, 4, 5, 6, 7, 13, 14, 15)$$

AB\CD	00	01	11	10
00	0	0	1	1
01	0	0	0	0
11	1	0	0	0
10	1	1	1	1

$$f(A,B,C,D)=AB'+B'C+AC'D' \text{ (sama)}$$

Berarti fungsi tersebut sudah dalam bentuk paling sederhana.

4. Diberikan dua buah bilangan biner masing-masing terdiri dari 2 bit, yaitu  $A = a_1a_2$  dan  $B = b_1b_2$ . Rancang sebuah rangkaian logika yang menghasilkan keluaran  $F = 1$  jika nilai  $A > B$ , dan  $F = 0$  jika tidak. Tentukan fungsi Boolean yang merepresentasikan kondisi tersebut menggunakan tabel kebenaran dan penyederhanaan dengan Peta Karnaugh. Selanjutnya, ubah fungsi yang diperoleh menjadi bentuk yang hanya menggunakan gerbang NAND dengan memanfaatkan hukum De Morgan. **(Nilai = 20)**

**Penyelesaian:**

Tabel Kebenaran

a1	a2	b1	b2	F
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

Peta Karnaugh:

a1a2/b1b2	00	01	11	10
00	0	0	0	0
01	1	0	0	0
11	1	1	0	1
10	1	1	0	0

Bentuk minimal:

$$F = a1b1' + a1a2b2' + a2b1'b2'$$

Bentuk

$$F = F''$$

$$F = ((a1b1') + (a1a2b2') + (a2b1'b2'))''$$

$$F = ((a1b1')' \cdot (a1a2b2')' \cdot (a2b1'b2')')'$$

NAND:

5. Diberikan ekspresi sumasi sebagai berikut  $S = \sum_{i=1}^{100} \sum_{j=1}^5 (2i - j + 3)$

(Nilai = 10)

- (a) Sederhanakan sumasi terhadap indeks  $j$  sehingga  $S$  hanya terdapat satu sumasi berindeks  $i$   
 (b) Tentukan nilai total dari  $S$

Penyelesaian:

(a)

$$\begin{aligned} \sum_{i=1}^{100} \sum_{j=1}^5 (2i - j + 3) &= \sum_{i=1}^{100} \sum_{j=1}^5 2i - \sum_{j=1}^5 j + \sum_{j=1}^5 3 \\ &= \sum_{i=1}^{100} (5 \times 2i) - (15 + (5 \times 3)) \\ &= \sum_{i=1}^{100} 10i - 15 + 15 \\ &= \sum_{i=1}^{100} 10i \end{aligned}$$

(b)  $\sum_{i=1}^{100} 10i = 10 \sum_{i=1}^{100} i$

$$\begin{aligned} &= 10 \times 100(101) / 2 \\ &= 10 \times 50(101) \\ &= 10 \times 5050 \\ &= \mathbf{50500} \end{aligned}$$

6. Putra adalah mahasiswa semester 2 yang sedang belajar pemrograman rekursif. Ia membuat fungsi untuk menghitung sesuatu sebagai berikut:

(Nilai = 15)

```
def mystery(n):
    if n == 0:
        return 2
    if n == 1:
        return 5
    return 3 * mystery(n - 1) - 2 * mystery(n - 2)
```

- (a) Nyatakan fungsi  $mystery(n)$  sebagai relasi rekurens dan kondisi awalnya!  
 (b) Hitung  $mystery(6)$  secara iteratif!  
 (c) Apakah relasi rekurens ini homogen linier? Jika iya, tentukan solusinya!

Penyelesaian:

a.  $a_n = 3a_{n-1} - 2a_{n-2}$  dengan  $a_0 = 2$  dan  $a_1 = 5$

b. Perhitungan iteratif:

-  $a_2 = 3(5) - 2(2) = 15 - 4 = 11$

- $a_3 = 3(11) - 2(5) = 33 - 10 = 23$
- $a_4 = 3(23) - 2(11) = 69 - 22 = 47$
- $a_5 = 3(47) - 2(23) = 141 - 46 = 95$
- $a_6 = 3(95) - 2(47) = 285 - 94 = 191$

Jadi  $\text{mystery}(6) = 191$

c. Iya, homogen lanjar.

- Persamaan karakteristik:  $r^2 - 3r + 2 = 0$
- Faktor:  $(r - 2)(r - 1) = 0 \rightarrow r_1 = 2, r_2 = 1$
- Solusi umum:  $a_n = \alpha_1 \cdot 2^n + \alpha_2 \cdot 1^n$
- Substitusi kondisi awal:
- $a_0 = 2 \rightarrow \alpha_1 + \alpha_2 = 2 \dots(1)$
- $a_1 = 5 \rightarrow 2\alpha_1 + \alpha_2 = 5 \dots(2)$
- Dari  $(2) - (1)$ :  $\alpha_1 = 3$ , maka  $\alpha_2 = -1$
- Solusi eksplisit:  $a_n = 3 \cdot 2^n - 1^n$

7. Sebuah server pengelola file *pdf* memiliki sistem *task queue* untuk melayani penggunaannya. Pada setiap awal iterasi (detik ke- $n$ ), jumlah tugas bertambah sebesar 10% dari jumlah tugas pada detik sebelumnya (detik ke- $(n-1)$ ) karena ada *overhead*. Secara bersamaan, setiap detik server juga menyelesaikan 67.000 tugas. Diketahui jumlah tugas pada detik ke-0 adalah  $a_0 = 2.000.000$  tugas. (Nilai = 15)

- Tentukan relasi rekursif untuk  $a_n$ , yakni jumlah tugas pada detik ke- $n$ .
- Server akan mengalami *overload* jika jumlah tugas dalam *task queue* melebihi 3.000.000 tugas. Tentukan pada detik ke berapa server pertama kali mencapai *overload*.

Catatan: Tuliskan langkah perhitungan iterasi dengan detail. Jika terdapat hasil desimal pada setiap iterasi, bulatkan menggunakan fungsi *floor* (pembulatan ke bilangan bulat ke bawahnya) sebelum melanjutkan ke iterasi berikutnya.

- Untuk memperbaiki masalah *overload* yang cepat terjadi seperti pada poin (b), *system engineer* harus meningkatkan pemrosesan server. Tentukan berapa jumlah tugas minimum yang harus diselesaikan server setiap detiknya agar jumlah proses di *task queue* tidak pernah bertambah dari jumlah awal  $a_0$ .

**Penyelesaian:**

(a)  $a_n = 1,1a_{n-1} - 67000$

(b)

$$a_0 = 2.000.000$$

$$a_1 = 1,1(2.000.000) - 67.000 = 2.200.000 - 67.000 = 2.133.000$$

$$a_2 = 1,1(2.133.000) - 67.000 = 2.346.300 - 67.000 = 2.279.300$$

$$a_3 = 1,1(2.279.300) - 67.000 = 2.507.230 - 67.000 = 2.440.230$$

$$a_4 = 1,1(2.440.230) - 67.000 = 2.684.253 - 67.000 = 2.617.253$$

$$a_5 = 1,1(2.617.253) - 67.000 = 2.878.978 - 67.000 = 2.811.978$$

$$a_6 = 1,1(2.811.978) - 67.000 = 3.093.175 - 67.000 = \mathbf{3.026.175}$$

Server akan mengalami *overload* pertama kali pada **detik ke-6**

(c)

$$1,1a_n - X \leq a_0$$

$$1,1(2.000.000) - X \leq 2.000.000$$

$$2.200.000 - X \leq 2.000.000$$

$$\mathbf{X \geq 200.000}$$

Bukti: Dengan  $X = 200.000$  (equilibrium)

$$a_1 = 1,1(2.000.000) - 200.000$$

$$a_1 = 2.200.000 - 200.000 = \mathbf{2.000.000}$$

Dengan pemrosesan 200.000 per detik (equilibrium), setiap iterasi berikutnya juga maksimal 2.000.000 tugas (sama dengan  $a_0$ ) sehingga server tidak akan pernah melebihi  $a_0$

8. (Bonus nilai = 5). Buktikan  $((a' + b)(a' + b'))' = a$  menggunakan hukum-hukum aljabar boolean, tuliskan hukumnya.

**Penyelesaian: Ada banyak cara membuktikan, salah satunya di bawah ini**

$$\begin{aligned} ((a' + b)(a' + b'))' &= (a'(a' + b') + b(a' + b'))' && \text{(Hukum Distributif)} \\ &= (a'a' + a'b' + ba' + bb')' && \text{(Hukum Distributif)} \\ &= (a' + a'b' + ba' + bb')' && \text{(Hukum Idempoten)} \\ &= (a' + a'b' + ba' + 0)' && \text{(Hukum Komplementer)} \\ &= (a' + ba' + 0)' = a && \text{(Hukum Penyerapan)} \\ &= (a' + 0)' = a && \text{(Hukum Penyerapan)} \\ &= (a')' && \text{(Hukum Identitas)} \\ &= a && \text{(Hukum Involusi)} \end{aligned}$$

**Atau**

$$\begin{aligned} ((a' + b)(a' + b'))' &= (a' + b)' + (a' + b')' && \text{(Hukum De Morgan)} \\ &= (a')' \cdot (b)' + (a')' \cdot (b')' && \text{(Hukum De Morgan)} \\ &= a \cdot b' + a \cdot b && \text{(Hukum Involusi)} \\ &= a(b + b') = a && \text{(Hukum Distributif)} \\ &= a \cdot 1 && \text{(Hukum Komplementer)} \\ &= a && \text{(Hukum Identitas)} \end{aligned}$$

**Total Nilai = 100 + bonus 5**