



LAPORAN PRAKTIKUM

IF310303

PRAKTIKUM SISTEM DIGITAL

MODUL: 3

MINIMISASI FUNGSI BOOLEAN

NAMA : Muhammad Alwiza Ansyar

NIM : M0520051

HARI : Jumat

TANGGAL : 23 Oktober 2020

WAKTU : 10.15 – 11.05 WIB

ASISTEN : Akhtar Bariq Rahman

PROGRAM STUDI INFORMATIKA

UNIVERSITAS SEBELAS MARET

2020

Modul 3

MINIMISASI FUNGSI BOOLEAN

Muhammad Alwiza Ansyar (M0520051) / 23 Oktober 2020

Email : alwiza21@student.uns.ac.id

Asisten : Akhtar Bariq Rahman

Abstraksi— Berikut merupakan laporan praktikum untuk modul 3 yang memiliki fokus bahasan tentang penyederhanaan fungsi Boolean. Praktikum ini juga membahas tentang *Summary-of-Product* (SOP) dan *Product-of-Summary* (POS). Laporan ini membahas tentang pengekspresian fungsi Boolean yang disediakan ke dalam bentuk SOP maupun POS dan menyederhanakannya. Pembahasan tersebut juga diikuti dengan rangkaian logika dan tabel kebenarannya.

Kata kunci— SOP, POS, bentuk kanonik, bentuk baku

I. PENDAHULUAN

Fungsi Boolean adalah fungsi yang terdiri dari Aljabar Boolean. Variabel *input* yang ada pada fungsi disebut dengan peubah atau literal. Fungsi Boolean dapat dinyatakan dengan tabel kebenarannya.

Pada tabel kebenaran, fungsi dapat ditulis dengan hanya menuliskan *input* dan *output*nya saja sehingga kita tidak diperlihatkan ekspresi Boolean seperti apa yang digunakan untuk mendapatkan *output* fungsi tersebut. Namun, kita tetap dapat mencari tahu ekspresi Boolean dari fungsi tersebut dengan menuliskan fungsi dalam bentuk kanonik menggunakan SOP atau POS. Akan tetapi, bentuk kanonik mengandung setiap literal yang ada sehingga ekspresi Boolean menjadi panjang. Maka, kita sederhanakan bentuk fungsi menjadi bentuk baku supaya ekspresi Booleannya lebih pendek. Penyederhanaan tersebut dapat dilakukan dengan cara Karnaugh's *Map Method*.

II. DASAR TEORI

2.1 Fungsi Boolean

Fungsi Boolean adalah fungsi yang dibentuk oleh n variabel Aljabar Boolean. Ekspresi Boolean adalah Fungsi Boolean pula. Variabel pada fungsi Boolean disebut peubah atau literal

2.2 Bentuk Kanonik dan Bentuk Baku

Fungsi Boolean dapat dinotasikan dalam bentuk kanonik maupun bentuk baku. Bentuk kanonik adalah bentuk yang memperlihatkan setiap literal yang ada pada fungsi, baik dalam bentuk biasa maupun komplemen. Bentuk baku adalah bentuk yang tidak diharuskan untuk memperlihatkan setiap literal yang ada pada fungsi. Bentuk baku merupakan bentuk yang lebih sederhana daripada bentuk kanonik.

2.3 Sum-of-Product (SOP)

Sum-of-Product (SOP) adalah kombinasi literal yang menghasilkan *output* fungsi berupa 1. Setiap suku pada SOP kanonik disebut *minterm*. SOP mengekspresikan fungsi dengan menambahkan (menggunakan OR) setiap *minterm*, yang mana *minterm* mengekspresikan suku SOP dengan mengkalikan (menggunakan AND) setiap literal.

Minterm dapat dinotasikan dengan m_p dengan p adalah urutan *output* fungsi yang dimulai dari 0 sampai dengan $2^n - 1$ dengan n adalah jumlah literal. Pada *minterm*, 0 diekspresikan menjadi literal komplemen sedangkan 1 diekspresikan menjadi literal.

2.4 Product-of-Sum (POS)

Product-of-Sum (POS) adalah kombinasi literal yang menghasilkan *output* fungsi berupa 0. Setiap suku pada SOP kanonik disebut *maxterm*. POS mengekspresikan fungsi dengan mengkalikan (menggunakan AND) setiap *maxterm*, yang mana *maxterm* mengekspresikan suku POS dengan menambahkan (menggunakan OR) setiap literal.

Maxterm dapat dinotasikan dengan M_p dengan p adalah urutan *output* fungsi yang dimulai dari 0 sampai dengan $2^n - 1$ dengan n adalah jumlah literal. Pada *maxterm*, 0 diekspresikan menjadi literal sedangkan 1 diekspresikan menjadi literal komplemen.

2.5 Karnaugh's Map Method

Karnaugh's *Map Method* atau metode peta Karnaugh adalah salah satu metode untuk menyederhanakan persamaan Aljabar Boolean. Metode ini diperkenalkan pertama kali oleh Maurice Karnaugh pada tahun 1953. Metode ini menggunakan peta Karnaugh yang mana peta tersebut memetakan setiap kombinasi literal pada fungsi dalam bentuk tabel. Kombinasi literal yang akan disederhanakan diekspresikan dengan 1 sedangkan kombinasi literal yang tidak akan disederhanakan diekspresikan dengan 0.

Penyederhanaan dilakukan dengan cara mengelompokkan 1 yang berdekatan. Pengelompokkan tersebut terdiri dari kelipatan 2 yang ditandai dengan membuat kotak, membuat lingkaran, atau menyamakan warna sel tersebut pada tabel. Setiap kelompok lalu disederhanakan dengan menghilangkan literal yang mengandung 0 dan 1.

III. ALAT DAN LANGKAH PERCOBAAN

3.1 Alat

1. PC/Laptop
2. Aplikasi Digital Works

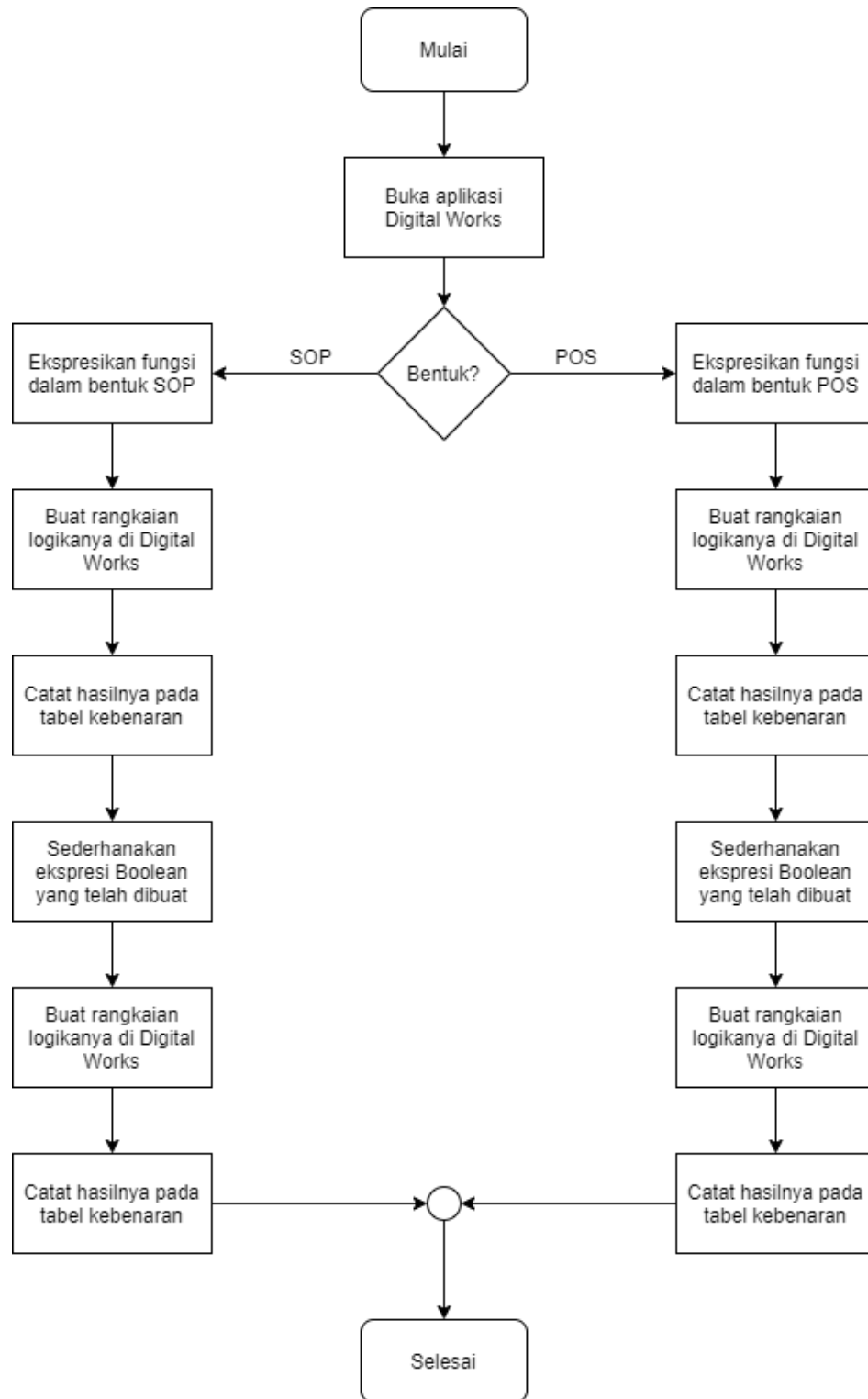
3.2 Langkah Percobaan

1. Buka Aplikasi Digital Works

A	B	C	F(A,B,C)
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

2. Buat fungsi Boolean dari tabel di atas dalam bentuk SOP dan POS
3. Buat rangkaian logikanya di Digital Works
4. Amati dan catat hasilnya dalam tabel kebenaran
5. Sederhanakan fungsi Boolean yang telah dibuat
6. Buatlah rangkaian logikanya di Digital Works
7. Amati dan catat hasilnya dalam tabel kebenaran

Diagram alur:



IV. HASIL DAN ANALISIS PERCOBAAN

4.1 Bentuk SOP

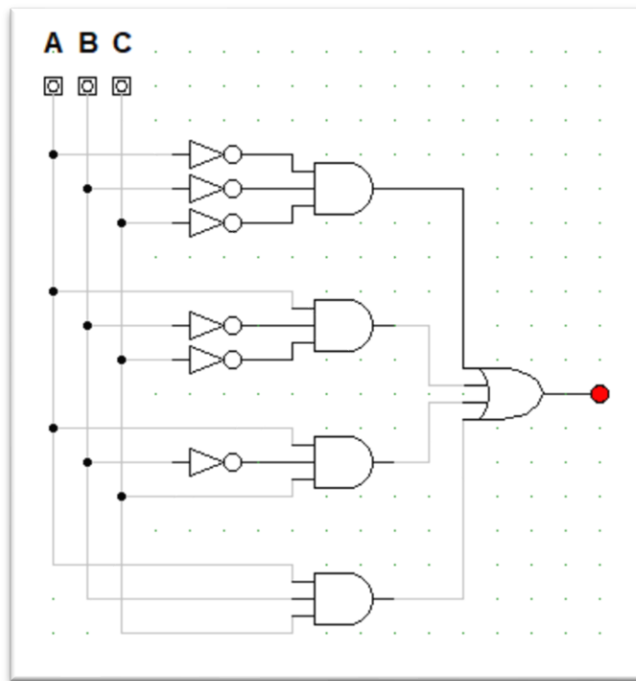
A	B	C	F(A,B,C)
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

Kombinasi literal yang memiliki *output* 1 adalah 000, 100, 101, dan 111. Maka:

$$F(A,B,C) = m_0 + m_4 + m_5 + m_7 = \sum (0, 4, 5, 7)$$

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

Rangkaian logika:



Tabel kebenaran:

A	B	C	A'B'C'	AB'C'	AB'C	ABC	F(A,B,C)
0	0	0	1	0	0	0	1
0	0	1	0	0	0	0	0
0	1	0	0	0	0	0	0
0	1	1	0	0	0	0	0
1	0	0	0	1	0	0	1
1	0	1	0	0	1	0	1
1	1	0	0	0	0	0	0
1	1	1	0	0	0	1	1

Untuk menyederhanakan ekspresi di atas, digunakan Karnaugh's *Map Method*

$$F(A,B,C) = A'B'C' + AB'C' + AB'C + ABC \text{ [bentuk kanonik]}$$

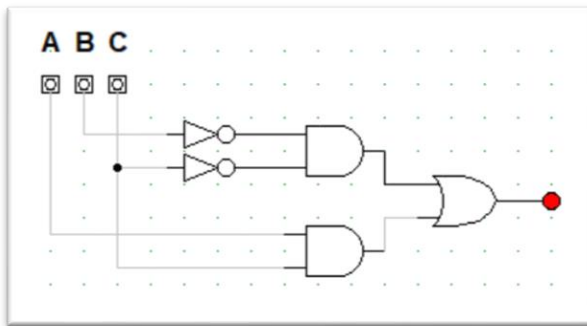
Peta Karnaugh:

A \ BC	BC			
	00	01	11	10
0	1	0	0	0
1	1	1	1	0

Penyederhanaan:

$$F(A,B,C) = B'C' + AC \text{ [bentuk baku]}$$

Rangkaian logika:



Tabel kebenaran:

A	B	C	B'C'	AC	F(A,B,C)
0	0	0	1	0	1
0	0	1	0	0	0
0	1	0	0	0	0
0	1	1	0	0	0
1	0	0	1	0	1
1	0	1	0	1	1
1	1	0	0	0	0
1	1	1	0	1	1

4.2 Bentuk POS

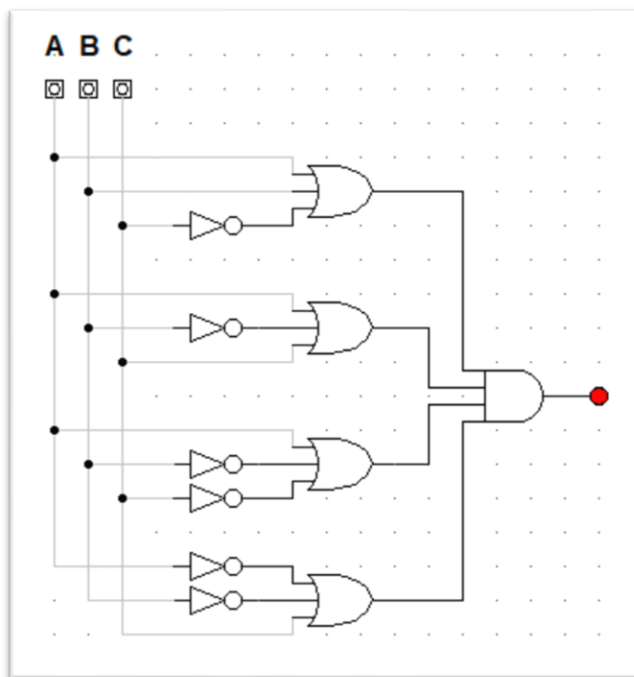
A	B	C	F(A,B,C)
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

Kombinasi yang memiliki *output* 0 adalah 001, 010, 011, dan 110. Maka:

$$F(A,B,C) = M_1 \cdot M_2 \cdot M_3 \cdot M_6 = \prod (1, 2, 3, 6)$$

$$F(A,B,C) = (A+B+C') \cdot (A+B'+C) \cdot (A+B'+C') \cdot (A'+B'+C)$$

Rangkaian logika:



Tabel kebenaran:

A	B	C	A+B+C'	A+B'+C	A+B'+C'	A'+B'+C	F(A,B,C)
0	0	0	1	1	1	1	1
0	0	1	0	1	1	1	0
0	1	0	1	0	1	1	0
0	1	1	1	1	0	1	0
1	0	0	1	1	1	1	1
1	0	1	1	1	1	1	1
1	1	0	1	1	1	0	0
1	1	1	1	1	1	1	1

Untuk menyederhanakan ekspresi di atas, digunakan *Karnaugh's Map Method*

$$F(A,B,C) = (A+B+C') \cdot (A+B'+C) \cdot (A+B'+C') \cdot (A'+B'+C) \text{ [bentuk kanonik]}$$

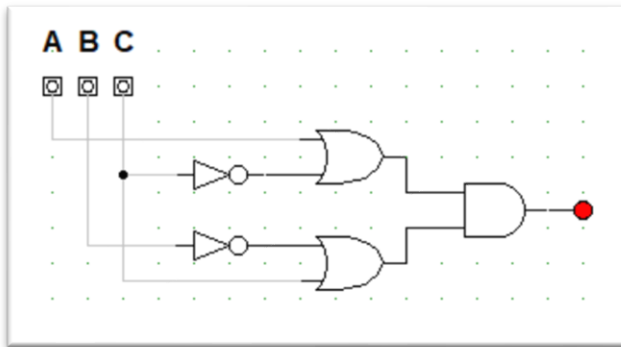
Peta Karnaugh:

A \ BC	00	01	11	10
0	0	1	1	1
1	0	0	0	1

Penyederhanaan:

$$F(A,B,C) = (A+C') \cdot (B'+C)$$

Rangkaian logika:



Tabel kebenaran:

A	B	C	A+C'	B'+C	F(A,B,C)
0	0	0	1	1	1
0	0	1	0	1	0
0	1	0	1	0	0
0	1	1	0	1	0
1	0	0	1	1	1
1	0	1	1	1	1
1	1	0	1	0	0
1	1	1	1	1	1

IV.KESIMPULAN

Fungsi Boolean adalah fungsi yg terdiri dari Aljabar Boolean. Fungsi Boolean dapat diekspresikan dalam bentuk *Summary-of-Product* (SOP) dan *Product-of-Summary* (POS). Bentuk SOP terdiri dari kombinasi literal (variabel pada fungsi) yang memiliki *output* berupa 1 dan setiap suku dari SOP disebut *minterm*. Bentuk POS terdiri dari kombinasi literal yang memiliki *output* berupa 0 dan setiap suku dari POS disebut *maxterm*. Awalnya bentuk SOP dan POS adalah kanonik. Bentuk kanonik mengekspresikan setiap literal yang ada sehingga ekspresi Boolean menjadi panjang. Maka, bentuk kanonik disederhanakan dengan mengubahnya menjadi bentuk baku. Penyederhanaan pada praktikum ini menggunakan metode peta Karnaugh.

Setelah dilakukan percobaan, setiap tabel kebenaran dari SOP, POS, serta bentuk sederhananya memiliki *output* yang sama sehingga bentuk-bentuk tersebut adalah sama dengan tabel kebenaran yang diberikan.

Bentuk sederhana juga sangat terlihat lebih simpel dan mudah dimengerti daripada bentuk kanonik. Hal ini saya rasa akan diperlukan pada pengaplikasiannya di dunia nyata karena bentuk yang sederhana akan menghemat gerbang logika dan waktu pengerjaan yang diperlukan

Jika diperhatikan lebih lanjut, setiap *minterm* hanya akan memiliki hasil 1 sebanyak sekali pada setiap kombinasi literal. Hal ini berhubungan dengan bentuk SOP yang merupakan kombinasi literal dengan *output* fungsi berupa 1. Kemudian, dikarenakan bentuk SOP menggunakan OR untuk setiap *minterm*nya, *output* fungsi 1 hanya didapat dari kombinasi *minterm* dengan 1 yang muncul sebanyak sekali (0001, 0010, 0100, atau 1000). Hal tersebut saya rasa mencerminkan definisi dari *minterm* sendiri yaitu *minimal term* yang mana *term* hanya memerlukan nilai 1 secara *minimal* (paling sedikit) yaitu dengan munculnya 1 sebanyak sekali supaya fungsi Boolean tersebut memiliki *output* 1

Begitu pula dengan *maxterm*. Setiap *maxterm* hanya menghasilkan 0 sebanyak sekali pada setiap kombinasi literal yang mana hal tersebut berhubungan dengan bentuk POS sendiri yang merupakan kombinasi literal dengan *output* fungsi berupa 0. Kemudian, dikarenakan bentuk POS menggunakan AND untuk setiap *maxterm*nya, *output* fungsi 0 hanya didapat dari kombinasi *maxterm* dengan 0 yang muncul sebanyak sekali (1110, 1101, 1011, atau 0111). Hal tersebut mencerminkan definisi dari *maxterm* sendiri yaitu *maximal term* yang mana *term* memerlukan nilai 1 secara maksimal (paling banyak) yaitu dengan munculnya 1 di semua *maxterm* supaya fungsi Boolean tersebut memiliki *output* 1.

V. DAFTAR PUSTAKA

Anonim. 2019. *Metode Peta Karnaugh (K-MAP)*, diakses dari <https://djukarna.wordpress.com/2019/11/28/metode-peta-karnaugh-k-map>, pada 30 Oktober 2020

Damayanti, Anggita. 2014. *ALJABAR BOOLEAN (SOP & POS)*, diakses dari <http://damayanti22.blogspot.com/2014/01/aljabar-boolean-sop-pos.html>, pada 30 Oktober 2020

Darmali, Harjanti. 2018. *Fungsi Boolean, Bentuk Kanonik dan Bentuk Baku*, diakses dari <https://slideplayer.info/slide/12900714/>, pada 30 Oktober 2020

Muhammad Alwiza Ansyar. Saya adalah seorang mahasiswa yang berasal dari Bogor. Saat ini, saya sedang menempuh pendidikan di Universitas Sebelas Maret jurusan Informatika.....