

EL2208

MINIMISASI LOGIC

KELOMPOK 1 | MAY 2022



KAYYISA ZAH RATUL FIRDAUS

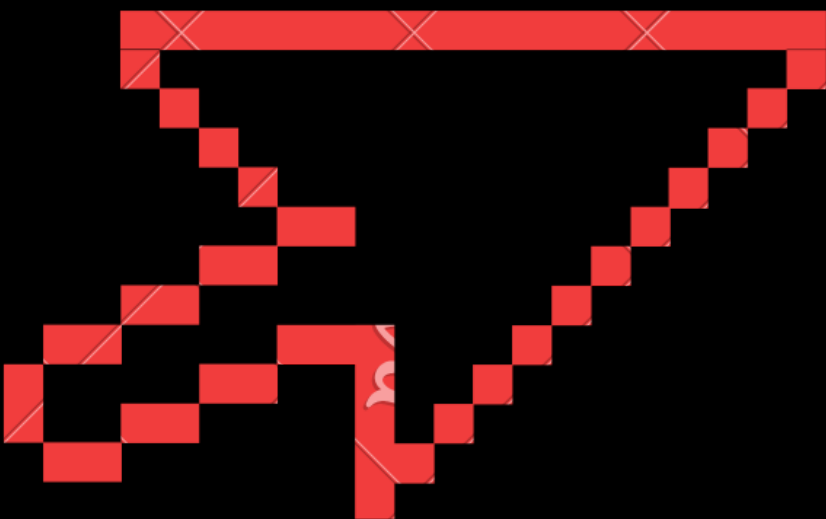
18320011



PUTRI ALFIYYAH DIANTI

18320041

EKSPLORASI ALGORITMA MINIMISASI LOGIC FUNCTION



BOOLEAN ALGEBRA

Sebuah set aturan yang digunakan dalam melakukan penyederhanaan fungsi rangkaian logika.

Terdapat 3 bagian :

1. Postulates
2. Properties
3. Theorem

BOOLEAN ALGEBRA PROPERTIES.

1a: $A \cdot 0 = 0$

1b: $A + 0 = A$

2a: $A \cdot 1 = A$

2b: $A + 1 = 1$

3a: $A \cdot A = A$

3b: $A + A = A$

4a: $A \cdot \overline{A} = 0$

4b: $A + \overline{A} = 1$

5a: $\overline{\overline{A}} = A$

5b: $A = \overline{\overline{A}}$

6a: $\overline{A \cdot B} = \overline{A} + \overline{B}$

6b: $\overline{A + B} = \overline{A} \cdot \overline{B}$

Postulates

Commutative	$A \cdot B = B \cdot A$	$A + B = B + A$
Associative	$A \cdot (B \cdot C) = (A \cdot B) \cdot C$	$A + (B + C) = (A + B) + C$
Distributive	$A \cdot (B + C) = A \cdot B + A \cdot C$	$A + (B \cdot C) = (A + B) \cdot (A + C)$

Properties

1a: $A = 1$ (if $A \neq 0$)

1b: $A = 0$ (if $A \neq 1$)

2a: $0 \cdot 0 = 0$

2b: $0 + 0 = 0$

3a: $1 \cdot 1 = 1$

3b: $1 + 1 = 1$

4a: $1 \cdot 0 = 0$

4b: $1 + 0 = 1$

5a: $\overline{1} = 0$

5b: $\overline{0} = 1$

7a: $A \cdot (A + B) = A$

7b: $A + A \cdot B = A$

8a: $(A + B) \cdot (A + \overline{B}) = A$

8b: $A \cdot B + A \cdot \overline{B} = A$

9a: $(A + \overline{B}) \cdot B = A \cdot B$

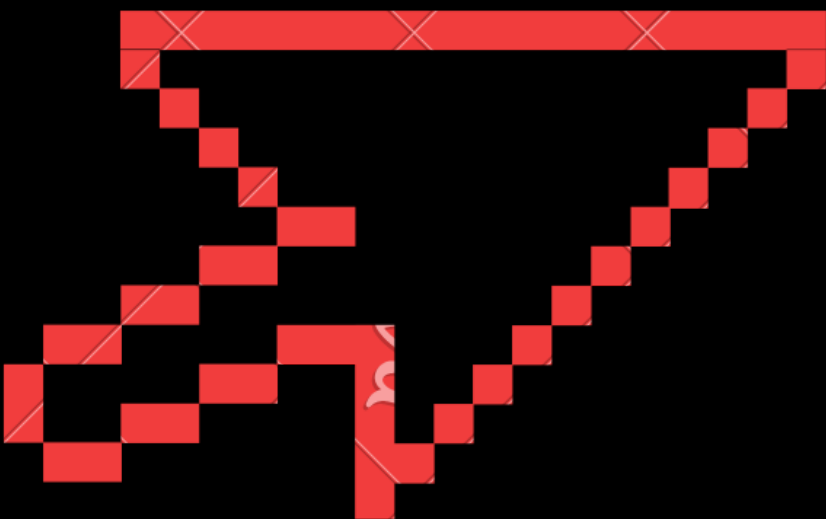
9b: $A \cdot \overline{B} + B = A + B$

10: $A \oplus B = \overline{A} \cdot B + A \cdot \overline{B}$

11: $A \odot B = \overline{A} \cdot \overline{B} + A \cdot B$

$\oplus = \text{XOR}$, $\odot = \text{XNOR}$

Theorems



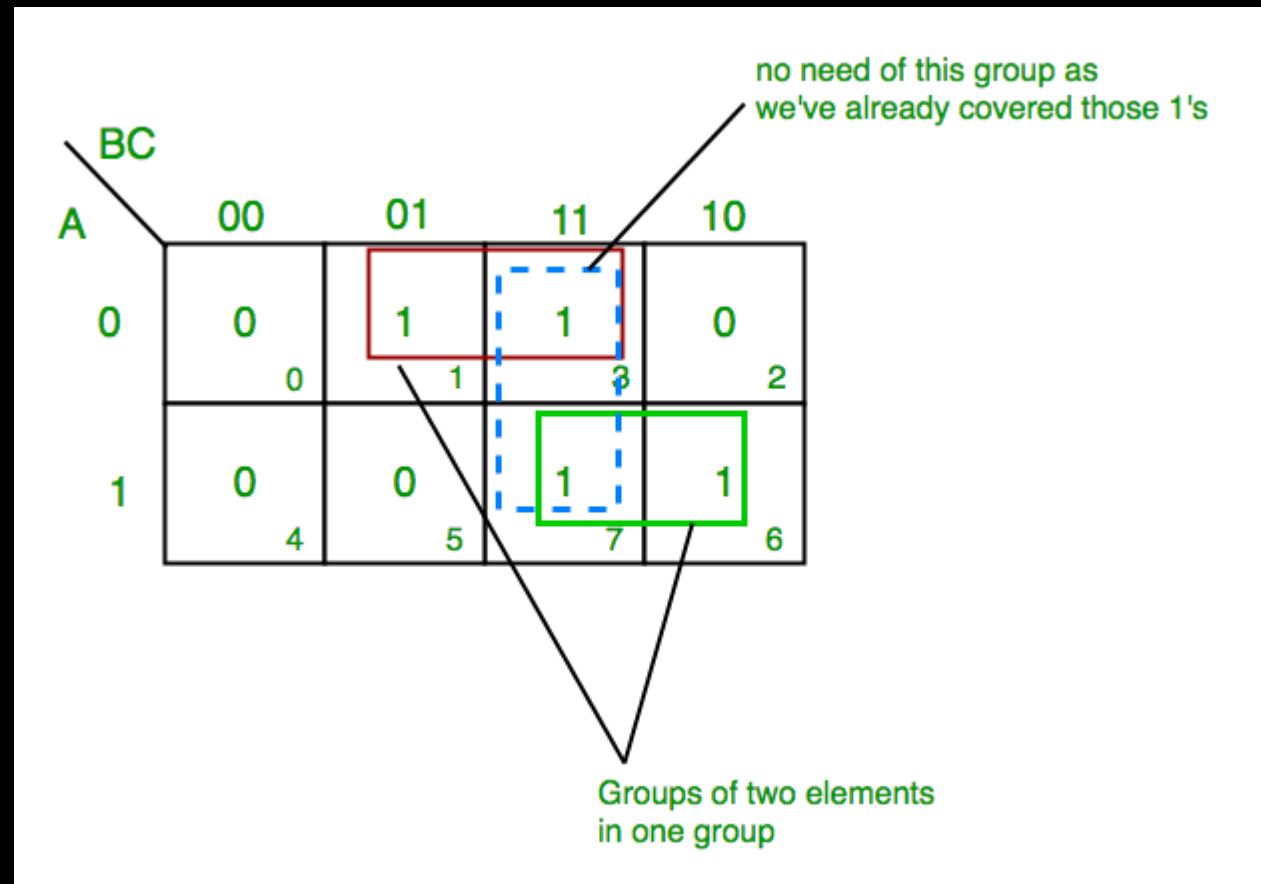
KARNAUGH MAPS

Pengaplikasiannya menggunakan tabel dengan bentuk tabel dan jumlah kolom dibentuk berdasarkan jumlah variabel .

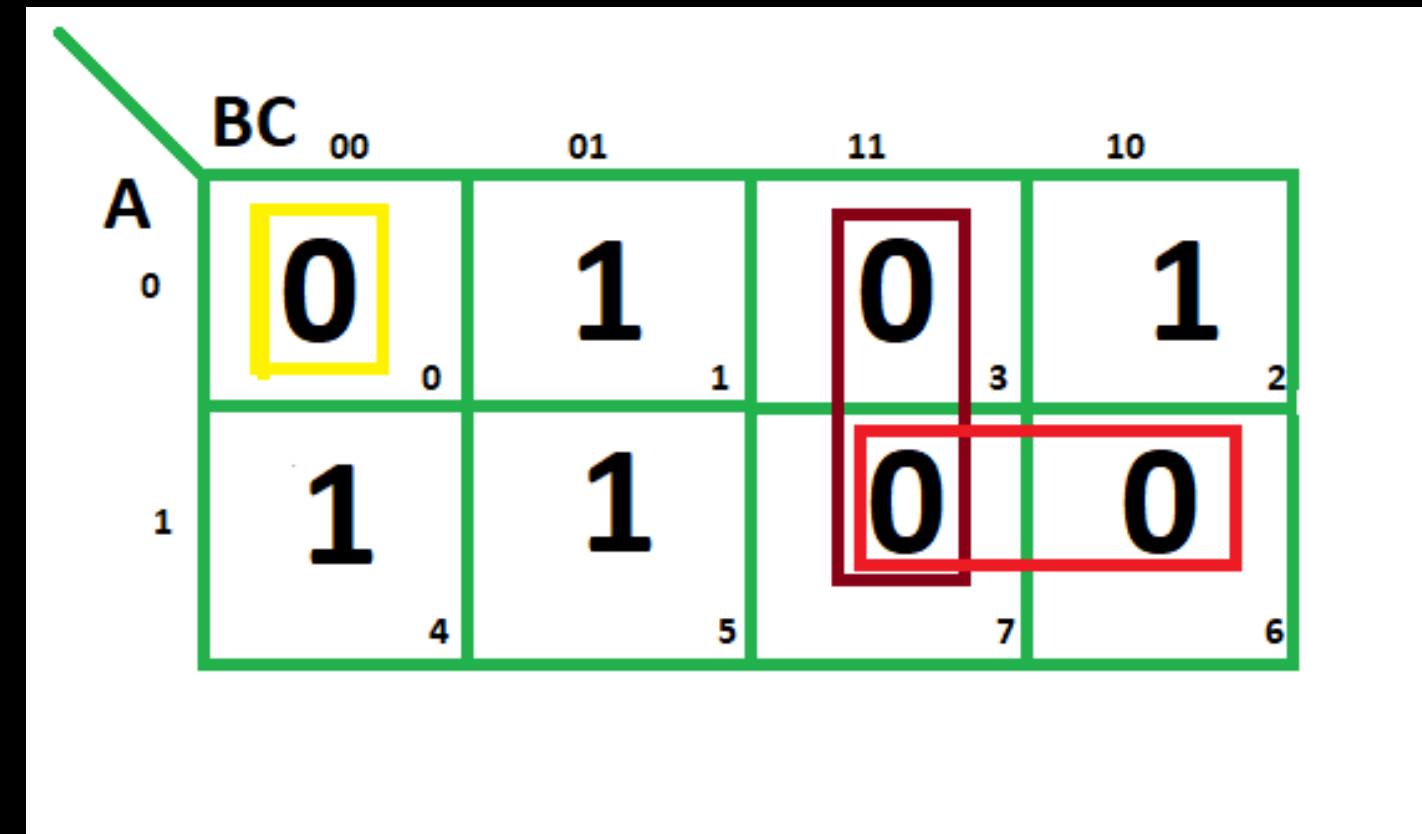
Ada 2 jenis :

1. SOP (Sum of Product)
2. POS (Product of Sum)

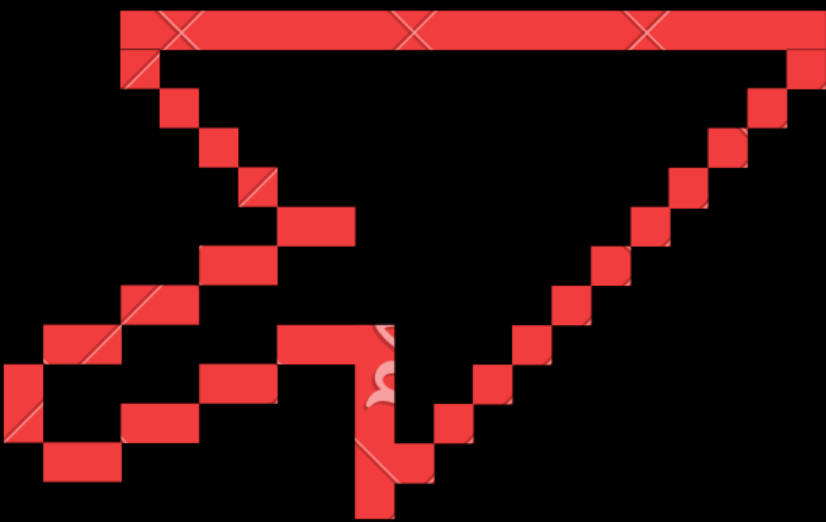
Karnaugh Maps



Bentuk tabel K-maps 3 variabel dengan penyelesaian SOP (sum of Product)



Bentuk tabel K-maps 3 variabel dengan penyelesaian POS (Product of Sum)



QUINE MCCLUSKEY

Minimisasi logic dengan prime implicant.
Metode ini biasanya digunakan untuk jumlah variabel yang besar (di atas 6).

Proses penyelesaian cukup panjang namun terstruktur .

Quine McCluskey

Contoh Penyelesaian Metode Quine McCluskey

$$F(A,B,C,D) = M(0,1,3,7,8,9,11,15)$$

Tabel 2.3.1. Pengubahan Ke Biner

0	0000
1	0001
3	0011
7	0111
8	1000
9	1001
11	1011
15	1111

Tabel 2.3.2. Pengelompokkan Berdasarkan Jumlah "1"

Group	Minterm	Bin Rep (ABCD)
0	M0	0000
1	M1	0001
	M8	1000
2	M3	0011
	M9	1001
3	M7	0111
	M11	1011
4	M15	1111

Tabel 2.3.4. Prime Implicant 1

Group	Matched Pairs	Bin . Rep
0	M0-M1	000-
	M0-M8	-000
1	M1-M3	00-1
	M1-M9	-001
	M8-M9	100-
2	M3-M7	0-11
	M3-M11	-011
	M9-M11	10-1
3	M7-M15	-111
	M11-M15	1-11

Quine McCluskey

Tabel 2.3.5. Prime Implicant II

Gro up	Matched Pairs	Bin . Rep (ABCD)	Prime Implicant
0	M0-M1- M8-M9	-00-	\overline{BC}
	M0-M8- M1-M9	-00-	
1	M1-M3- M9-M11	-0-1	$\overline{B}D$
	M1-M9- M3-M11	-0-1	
2	M3-M7- M11-M15	--11	CD
	M3-M11- M7-M15	--11	



Prime Implicant	Minterm Involved	0	1	3	7	8	9	11	15
\overline{BC}	0,1,8,9	X*	x			X*	x		
$\overline{B}D$	1,3,9,11		x	X			x	x	
CD	3,7,11,15			x	X*			x	X*

Kemudian pilih prime Implicant yang memiliki 1 buah "X" pada kolomnya . Dan hasil akhir persamaan adalah $(-B-C + -CD)$



PENGIMPLEMENTASIAN PROGRAM BAHASA C

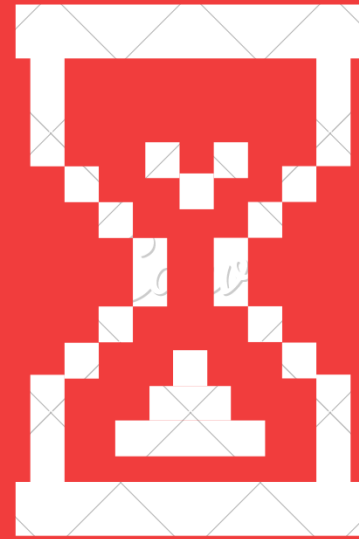


PERMASALAHAN PROGRAM

Metode minimisasi logic yang digunakan program adalah Quine McCluskey



<code>initial_var</code>	<code>compare()</code>
<code>inputData()</code>	<code>initial_implicants()</code>
<code>sort()</code>	<code>reduce()</code>
<code>searchIdxMin()</code>	<code>showImpIi()</code>
<code>delNode()</code>	<code>showTable()</code>
<code>arrMinterms()</code>	<code>implicant()</code>
<code>groupby()</code>	<code>finalNode()</code>
<code>simplificationTab()</code>	<code>changetoVar()</code>
<code>removeLast()</code>	<code>finalResult()</code>



TERIMA KASIH

KELOMPOK 1