

EL2008 PEMECAHAN MASALAH DENGAN C  
TUGAS BESAR  
MEMBUAT PROGRAM MINIMISASI LOGIKA



Muhammad Zakie Shahab (13220019)  
Muhammad Hanif Hibatullah (13220051)  
Radithya Arisaputra (13220053)

TEKNIK ELEKTRO  
INSTITUT TEKNOLOGI BANDUNG  
BANDUNG  
2022

# DAFTAR ISI

<b>DAFTAR ISI</b>	2
<b>PENDAHULUAN</b>	3
<b>DASAR TEORI</b>	4
Sum Of Product (SOP)	4
Minimisasi Logika	4
Metode Tabular	4
<b>RANCANGAN PROGRAM</b>	6
Input dan Output	6
Algoritma	6
<b>FLOWCHART</b>	7
<b>DATA FLOW DIAGRAM</b>	9
<b>DEMONSTRASI</b>	10
Demo 1	10
Demo 2	10
Demo 3	11
<b>KESIMPULAN DAN LESSON LEARNED</b>	13
Kesimpulan	13
Lesson Learned	13
<b>PEMBAGIAN TUGAS</b>	14
<b>REFERENSI</b>	15
<b>LAMPIRAN</b>	16

# PENDAHULUAN

Pada tugas besar ini, diminta untuk membuat program yang dapat melakukan minimisasi logika. Minimisasi logika merupakan proses menyederhanakan ekspresi aljabar dari fungsi boolean. Minimisasi logika dapat dilakukan dengan berbagai cara, beberapa diantaranya adalah aljabar boolean, Karnaugh Map, dan Metode Tabular (Quine-McCluskey).

Pada tugas besar ini, akan dibuat sebuah program yang melakukan minimisasi logika dengan menggunakan Metode Tabular dan juga program minimisasi ini terbatas pada SOP (Sum of Product). Input yang diterima oleh program adalah jumlah variabel, input minterm, dan don't care dalam bentuk indeks angka.. Kemudian akan memberikan output berupa persamaan aljabar boolean dari input yang sudah disederhanakan.

Pada dasarnya, aljabar boolean merupakan sebuah fungsi yang berdasar pada *trial and error* untuk mencari sebuah nilai pada tiap variabelnya. Disini datanglah pentingnya melakukan minimisasi, yaitu agar pencarian nilai-nilai pada variabel yang ditujukan untuk mencari sebuah hasil yang diinginkan menjadi lebih simpel dan mudah.

Lalu, penyederhanaan ini juga akan memberi efek pada implementasinya ke rangkaian logika (biasanya menggunakan CMOS). Semakin sederhana fungsi yang kita buat maka akan semakin sederhana pula rangkaian logikanya. Hal ini akan memberi dampak positif pada pengeluaran, besar rangkaian, dan juga efektivitas rangkaian. Oleh karena itu, dilakukan pembuatan program untuk melakukan minimisasi pada aljabar boolean dengan standar SOP (Sum of Product).

# DASAR TEORI

## 1. Sum Of Product (SOP)

SOP (Sum of Product) juga sering disebut dengan minterm merupakan sebuah teknik penulisan fungsi dari sebuah aljabar boolean logic gates dimana fungsi yang ditulis merupakan penjumlahan dari perkalian-perkalian variabel yang menjadi input. SOP ditulis dengan lambang sigma (sum) diikuti dengan rentetan angka sesuai yang ingin dihitung. Pembagian variabel hitung ditulis dengan huruf 'm' dan diikuti angka urutan nilai variabelnya mulai dari yang terkecil. Berikut merupakan contoh penjabaran dari SOP.

x	y	Lambang	Nilai
0	0	m0	$x'y'$
0	1	m1	$xy'$
1	0	m2	$x'y$
1	1	m3	$xy$

## 2. Minimisasi Logika

Minimisasi logika merupakan proses meminimalkan suatu persamaan boolean. Dalam melakukan minimisasi, dapat dilakukan berbagai cara diantaranya adalah

- Aljabar Boolean, yaitu dengan menerapkan teorema teorema yang berlaku pada persamaan tersebut. Teorema yang sering digunakan di antaranya adalah teorema de'Morgan, Absorption dan sejenisnya.
- Karnaugh Map, yaitu dengan mengubah persamaan boolean yang ingin diminimisasi ke dalam bentuk mintermnya. Dari nilai minterm tersebut dibuat tabel yang disebut karnaugh map dan dilakukan operasi operasi yang berlaku pada karnaugh map. Umumnya, penggunaan karnaugh map lebih mudah jika dibandingkan metode lain untuk jumlah variabel yang sedikit (<5).
- Metode Tabular (*Quine McCluskey*), metode ini digunakan juga dengan memanfaatkan minterm dari persamaan boolean yang ingin diminimisasi. Minterm tersebut diambil bentuk binarynya dan dilakukan operasi yang berlaku pada metode tabular.

## 3. Metode Tabular

Metode tabular merupakan salah satu metode atau cara untuk melakukan minimisasi pada suatu persamaan boolean. Metode ini memanfaatkan minterm dan don't care pada persamaan boolean dan dilakukan minimisasi dari minterm dan don't care tersebut. Minterm sendiri merupakan suatu istilah yang digunakan ketika output dari suatu persamaan bernilai True atau 1 ketika diberikan suatu input. Selain minterm, ada juga istilah lain yang disebut dengan Don't Care. Dimana don't care memiliki arti bahwa

output pada kondisi tersebut bisa memiliki nilai 1 atau 0. Nilai don't care ini dapat membantu dalam melakukan minimisasi persamaan boolean karena kita dapat memasangkan / grouping suatu minterm dengan don't care tersebut.

Dari index minterm dan don't care tersebut, kita dapat melakukan minimisasi dengan memasangkan minterm berdasarkan angka 1 pada bentuk binary dari indexnya, contohnya untuk minterm dengan index 3 (memiliki bentuk binary 011) maka digroup dengan minterm yang jumlah angka satunya ada 2. Lalu selanjutnya dilakukan minimisasi dengan melihat jumlah minterm yang berbeda tepat 1 dan dipasangkan. Dilakukan seterusnya sampai tidak dapat lagi di minimisasi. Selanjutnya dicari *prime implicants* dan *essential prime implicants* dari hasil akhir sebelumnya.

# RANCANGAN PROGRAM

## Input dan Output

Input : jumlah variabel, jumlah minterm, jumlah don't care, indeks minterm, indeks don't care

Output : persamaan aljabar boolean dari hasil minimisasi input

## Algoritma

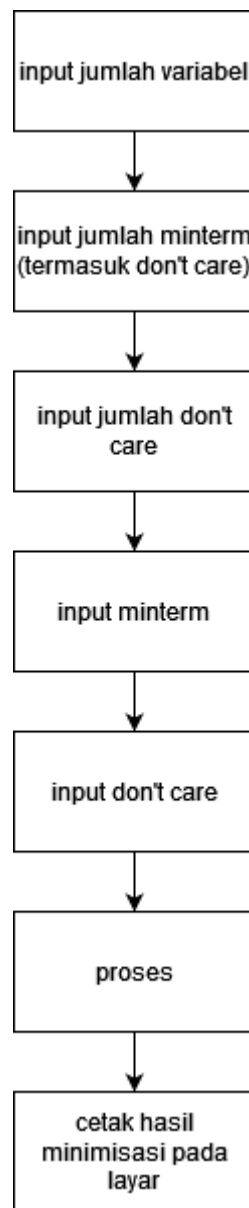
1. Menerima input jumlah variabel, jumlah minterm, jumlah don't care, indeks minterm, dan indeks don't care
2. Mengelompokkan input berdasarkan jumlah '1' pada bentuk binary minterm
3. Minimisasi input sampai tidak bisa diminimisasi
4. Tentukan prime implicant (PI) dari hasil minimisasi
5. Tentukan essential prime implicant (EPI) dari PI
6. Ambil PI yang mengandung EPI
7. Ambil PI lain sampai semua elemen tercakup
8. Mengeluarkan output berupa persamaan boolean yang telah diminimisasi

Algoritma minimisasi input:

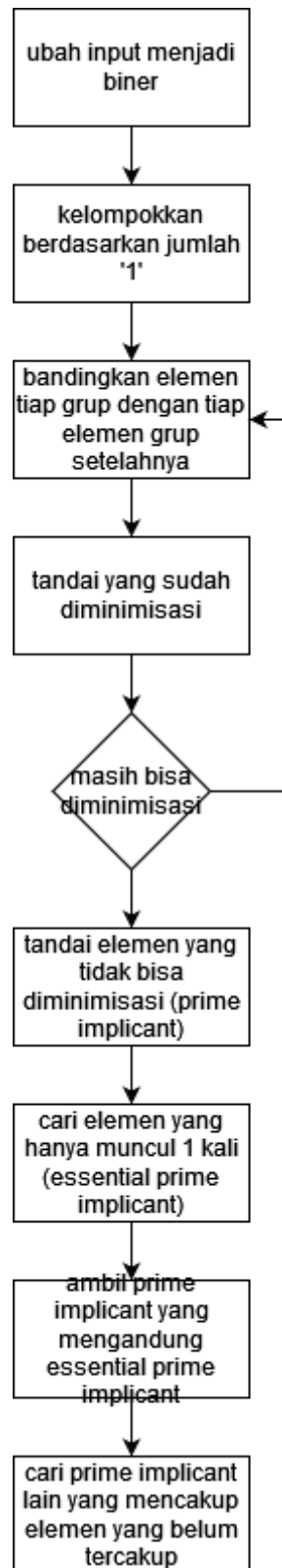
1. Membandingkan biner dari 2 indeks dari kelompok (jumlah '1') yang berbeda
2. Apabila hanya ada 1 biner yang berbeda, indeks digabung menjadi 1 indeks
3. Hasil dikelompokkan kembali berdasarkan jumlah '1' pada biner
4. Ulangi dari langkah 1, pengulangan terus dilakukan sampai langkah 2 tidak dapat berlangsung (Tidak tersisa indeks yang berbeda hanya 1 biner)

# FLOWCHART

Flowchart main program

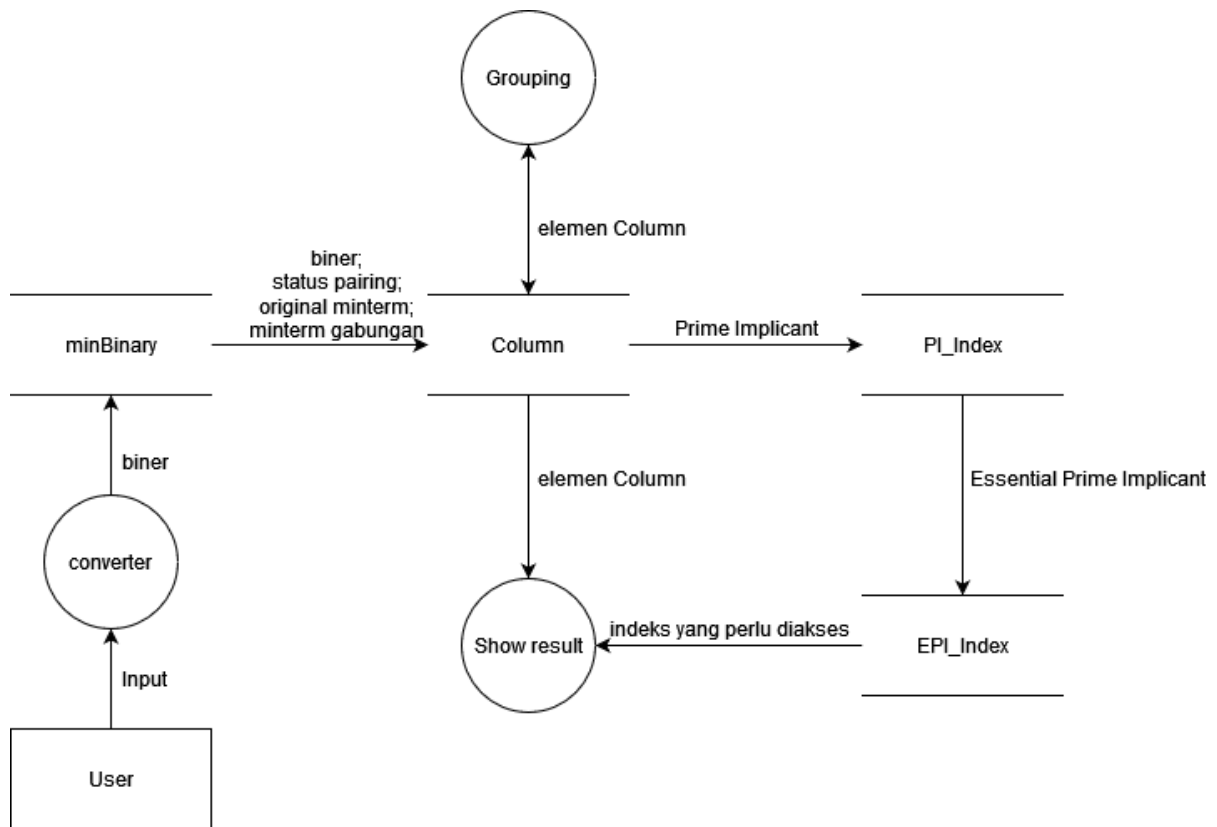


# Flowchart blok proses





# DATA FLOW DIAGRAM



# DEMONSTRASI

Hasil demonstrasi akan dikonfirmasi menggunakan  
<https://www.charlie-coleman.com/experiments/kmap/>

## Demo 1

Jumlah variabel = 4

Jumlah minterm = 12

Minterm = 4,5,6,9,11,12,13,14

Don't care = 0,1,3,7

```
Program untuk menyederhanakan persamaan boolean.
Masukkan jumlah variabel persamaan: 4
Masukkan jumlah minterm (termasuk don't care) pada persamaan: 12
Masukkan jumlah Don't Care pada persamaan: 4
Masukkan minterm ke-1 (berurut dari rendah):0
Masukkan minterm ke-2 (berurut dari rendah):1
Masukkan minterm ke-3 (berurut dari rendah):3
Masukkan minterm ke-4 (berurut dari rendah):4
Masukkan minterm ke-5 (berurut dari rendah):5
Masukkan minterm ke-6 (berurut dari rendah):6
Masukkan minterm ke-7 (berurut dari rendah):7
Masukkan minterm ke-8 (berurut dari rendah):9
Masukkan minterm ke-9 (berurut dari rendah):11
Masukkan minterm ke-10 (berurut dari rendah):12
Masukkan minterm ke-11 (berurut dari rendah):13
Masukkan minterm ke-12 (berurut dari rendah):14
Masukkan Don't Care ke-1 (berurut dari rendah):0
Masukkan Don't Care ke-2 (berurut dari rendah):1
Masukkan Don't Care ke-3 (berurut dari rendah):3
Masukkan Don't Care ke-4 (berurut dari rendah):7
```

Persamaan boolean yang telah disederhanakan adalah:

$B'D + BD' + C'D$

Press any key to exit...

Output Name:

One string for function result

f

Minterms:

Comma separated list of numbers

4,5,6,9,11,12,13,14

Generic:

$f(a, b, c, d) = b'd + bd' + bc'$

Input Names:

Comma separated list of variable names

a, b, c, d

Don't Cares:

Comma separated list of numbers

0,1,3,7

## Demo 2

Jumlah variabel = 8

Jumlah minterm = 4

Minterm = 19,51,53,132

Don't care =

```

Program untuk menyederhanakan persamaan boolean.
Masukkan jumlah variabel persamaan: 8
Masukkan jumlah minterm (termasuk don't care) pada persamaan: 4
Masukkan jumlah Don't Care pada persamaan: 0
Masukkan minterm ke-1 (berurut dari rendah):19
Masukkan minterm ke-2 (berurut dari rendah):51
Masukkan minterm ke-3 (berurut dari rendah):53
Masukkan minterm ke-4 (berurut dari rendah):132

Persamaan boolean yang telah disederhanakan adalah:

A'B'DE'F'GH+A'B'CDE'FG'H+AB'C'D'E'FG'H'

Press any key to exit...

```

Function Info	Terms	Solutions:
<b>Output Name:</b> <small>One string for function result</small> <input type="text" value="f"/>	<b>Minterms:</b> <small>Comma separated list of numbers</small> <input type="text" value="19,51,53,132"/>	<b>Generic:</b> <input type="text" value="f(a, b, c, d, e, f, g, h) = a'b'de'fgh + a'b'cde'fg'h + ab'c'd'e'fg'h'"/>
<b>Input Names:</b> <small>Comma separated list of variable names</small> <input type="text" value="a, b, c, d, e, f, g, h"/>	<b>Don't Cares:</b> <small>Comma separated list of numbers</small> <input type="text" value="0"/>	

### Demo 3

Jumlah variabel = 10

Jumlah minterm = 10

Minterm = 26,132,208,246,264,420,911

Don't care = 19,51,53

```

Program untuk menyederhanakan persamaan boolean.
Masukkan jumlah variabel persamaan: 10
Masukkan jumlah minterm (termasuk don't care) pada persamaan: 10
Masukkan jumlah Don't Care pada persamaan: 3
Masukkan minterm ke-1 (berurut dari rendah):19
Masukkan minterm ke-2 (berurut dari rendah):26
Masukkan minterm ke-3 (berurut dari rendah):51
Masukkan minterm ke-4 (berurut dari rendah):53
Masukkan minterm ke-5 (berurut dari rendah):132
Masukkan minterm ke-6 (berurut dari rendah):208
Masukkan minterm ke-7 (berurut dari rendah):246
Masukkan minterm ke-8 (berurut dari rendah):264
Masukkan minterm ke-9 (berurut dari rendah):420
Masukkan minterm ke-10 (berurut dari rendah):911
Masukkan Don't Care ke-1 (berurut dari rendah):19
Masukkan Don't Care ke-2 (berurut dari rendah):51
Masukkan Don't Care ke-3 (berurut dari rendah):53

Persamaan boolean yang telah disederhanakan adalah:

A'B'C'D'E'FGH'I'J'+A'B'CD'E'F'G'HI'J'+A'B'CDE'FG'H'I'J'+A'B'CDEFG'HI'J'+A'BC'D'E'F'GH'I'J'+A'BCD'EF'G'HI'J'+ABCD'E'F'GHI'J'

Press any key to exit...

```

Function Info	Terms	Solutions:
<b>Output Name:</b> One string for function result f	<b>Minterms:</b> Comma separated list of numbers 26,132,208,246,264,420,911	<b>Generic:</b> $f(a, b, c, d, e, f, g, h, i, j) = a'b'c'd'e'fgh'ij' + a'b'cd'e'fgh'i'j' + a'b'cde'fg'h'i'j' + a'b'cdefg'hij' + a'bc'd'e'fgh'i'j' + a'bcd'efg'hi'j' + abcd'e'fghij$
<b>Input Names:</b> Comma separated list of variable names a, b, c, d, e, f, g, h, i, j	<b>Don't Cares:</b> Comma separated list of numbers 19,51,53	

# KESIMPULAN DAN LESSON LEARNED

## Kesimpulan

Program dapat berfungsi dengan baik dan sesuai spesifikasi yang diminta. Pertama-tama program menerima jumlah input-input yang diperlukan, yaitu jumlah variabel, jumlah minterm, jumlah don't care. Kemudian menerima nilai nilai minterm dan don't care. Setelah itu program melakukan proses seperti pada flowchart blok proses. Kemudian menampilkan output pada layar.

## Lesson Learned

### 1. Muhammad Zakie Shahab (13220019)

Pada tugas besar ini, saya mempelajari kembali mengenai minimisasi pada sebuah aljabar boolean dan mendapatkan ilmu yang lebih lagi daripada yang sudah pernah dipelajari sebelumnya. Lalu, saya juga mendapat informasi lebih mengenai penggunaan-penggunaan fungsi bawaan yang ada pada bahasa C. Sebagai tambahan, saya juga belajar lebih banyak mengenai cara bekerja sama sebagai sebuah tim untuk menyelesaikan permasalahan yang ada

### 2. Muhammad Hanif Hibatullah (13220051)

Hal yang saya pelajari dari tugas besar ini adalah mengenai penerapan atau penggunaan ilmu yang saya dapatkan di mata kuliah lain (untuk Tugas Besar kali ini adalah mata kuliah Sistem Digital) untuk menyelesaikan suatu permasalahan dan diimplementasikan ke dalam bahasa C. Selain itu, saya juga belajar untuk bekerja sama dalam mengerjakan suatu permasalahan yang diberikan.

### 3. Radithya Arisaputra (13220053)

Dari tugas besar ini, saya belajar cara melakukan minimisasi terhadap suatu ekspresi logika. Selain itu juga saya belajar cara bekerjasama dengan orang lain sebagai tim untuk menyelesaikan suatu permasalahan.

## PEMBAGIAN TUGAS

Berkas	13220019	13220051	13220053
File ppt	•	•	•
Laporan	•	•	•
Flowchart	•	•	•
DFD	•	•	•
Kode	•	•	•

## REFERENSI

Huang, J. (2014). Programing implementation of the Quine-McCluskey method for minimization of Boolean expression. *arXiv preprint arXiv:1410.1059*.

# LAMPIRAN

Percakapan yang terjadi sekitar 11 jam 29 menit sebelum presentasi:

Read 2  
7:31 PM

gw takut ditanyain source code anjir

Muhammad Hanif H

harusnya ga sih

7:31 PM

Read 2  
7:31 PM

"bagian ini untuk apa ya?"

Read 2  
7:32 PM

"hanya Tuhan yang tau pak"

Muhammad Hanif H

wkowkaowk

7:32 PM

Zakie

"Itu dimasuk2in aja pak, soalnya kalo gak dimasukin gak jalan kodenya"

7:32 PM