Tugas Besar EL2008 Pemecahan Masalah dengan C

# Algoritma Minimisasi Persamaan Logika dengan Menggunakan Metode Quine-McCluskey (QM)



Oleh:

Kadhan Dalilurahman (13220001)

Elmo Ryaner Panggabean (13220012)

Maheswara Apta Adiyatma (18320005)

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung

Jl. Ganesha No. 10 Bandung, Indonesia, 40132

2022

# Daftar Isi

[Algoritma Minimisasi Persamaan Logika dengan Menggunakan Metode Quine-McCluskey (QM) 0](#_Toc104933332)

[Daftar Isi 1](#_Toc104933333)

[Abstrak 2](#_Toc104933334)

[Pendahuluan 3](#_Toc104933335)

[Latar Belakang 3](#_Toc104933336)

[Tujuan 3](#_Toc104933337)

[Dasar Teori 4](#_Toc104933338)

[Langkah 1: Menerima Minterm dan Mengelompokkannya 4](#_Toc104933339)

[Langkah 2: Memasangkan Minterm yang Berbeda Kelompok 4](#_Toc104933340)

[Langkah 3: Melakukan Penghilangan Duplikat 6](#_Toc104933341)

[Langkah 4: Membuat PI Chart 6](#_Toc104933342)

[Langkah 5: Mencari Essential Prime Implicant 7](#_Toc104933343)

[Langkah 6: Menurunkan PI Chart dan Mentransformasi EPI Menjadi Boolean 7](#_Toc104933344)

[Penjelasan Program 9](#_Toc104933345)

[Struktur Data yang Digunakan 9](#_Toc104933346)

[Daftar Variabel dan Fungsi/Prosedur 9](#_Toc104933347)

[Daftar Variabel 9](#_Toc104933348)

[Daftar Fungsi 9](#_Toc104933349)

[Alur Program 10](#_Toc104933350)

[Input 11](#_Toc104933351)

[Proses dan Output 11](#_Toc104933352)

[Data Flow Diagram 12](#_Toc104933353)

[Pengujian Program 13](#_Toc104933354)

[*Lesson Learned* 15](#_Toc104933355)

[Kesimpulan 16](#_Toc104933356)

[Pembagian Tugas 17](#_Toc104933357)

[Daftar Pustaka 18](#_Toc104933358)

# Abstrak

Quine-McCluskey Method atau metode Tabular merupakan metode minimisasi persamaan logika yang paling umum digunakan dalam membuat algoritma penyederhanaan persamaan logika dan dapat diimplementasikan dengan software (perangkat lunak) secara algoritmik. Pada laporan ini akan membahas tentang pembuatan program minimisasi logika dalam bahasa pemrograman C. Metode yang digunakan adalah Quine-McCluskey Method atau metode Tabular. Hasil dari tugas besar ini adalah bahwa minimisasi logika dapat disimulasikan dalam program dengan menggunakan metode Quine-McCluskey atau Metodee Tabular.

# Pendahuluan

## Latar Belakang

Minimisasi logika (logic minimization) adalah proses menemukan bentuk persamaan logika paling sederhana dengan mengidentifikasi dan menghapus semua input yang berlebihan. Beberapa metode telah dikembangkan untuk membantu proses minimisasi, salah satunya yaitu Quine-McCluskey Method, atau metode Tabular. Metode ini merupakan metode minimisasi persamaan logika yang paling umum digunakan dalam membuat algoritma penyederhanaan persamaan logika. Keuntungan utama dari metode ini adalah dapat menerima lebih banyak input daripada K-map dan dapat diimplementasikan dengan software (perangkat lunak) secara algoritmik, sedangkan kelemahannya adalah kompleksitas komputasinya masih tetap tinggi.

## Tujuan

* Mahasiswa dapat melakukan pemecahan masalah dengan C,
* Mahasiswa bekerja sama dan berkomunikasi dalam kelompok serta mempresentasikan hasil kerja kelompok.

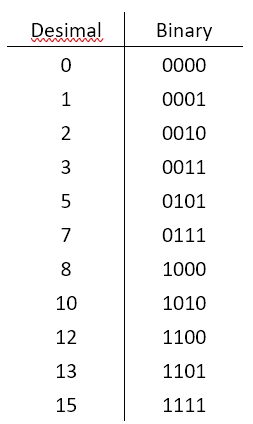
# Dasar Teori

## Langkah 1: Menerima Minterm dan Mengelompokkannya

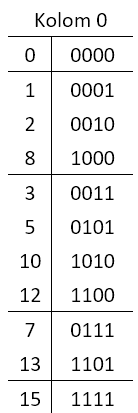
Misalkan terdapat fungsi logika sebagai berikut:

**F (A, B, C, D) = Σ (0, 1, 2, 3, 5, 7, 8, 10, 12, 13, 15)**

Minterm-minterm dari fungsi tersebut ditransformasi menjadi biner lalu dikelompokkan berdasarkan banyaknya angka ‘1’. Minterm yang sudah dikelompokkan dimasukkan kedalam kolom 0.



Gambar 3.1.1. Transformasi Minterm Menjadi Biner



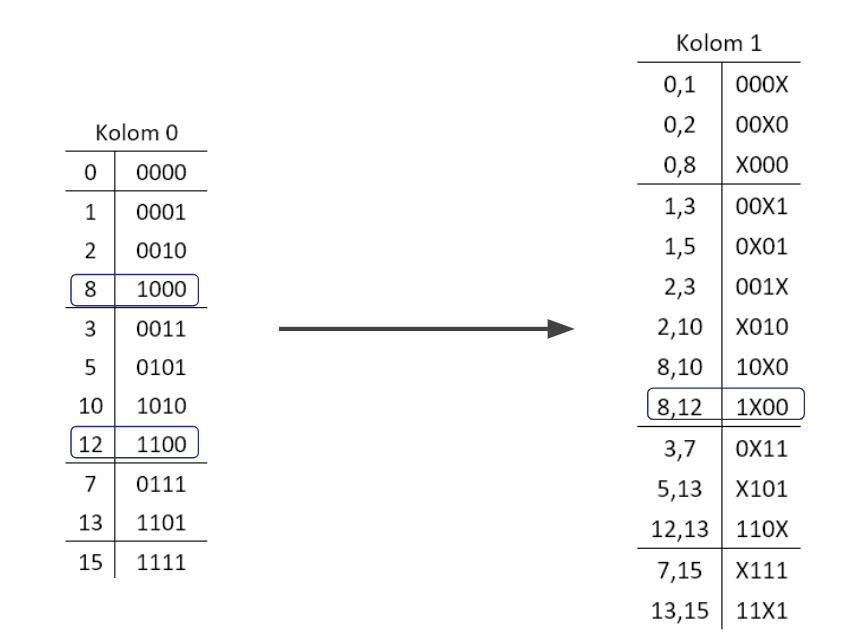
Gambar 3.1.2. Minterm Dikelompokkan Berdasarkan Banyaknya “1”

## Langkah 2: Memasangkan Minterm yang Berbeda Kelompok

Pada Langkah ini, kita akan memasangkan dua minterm yang berbeda kelompok (memiliki jumlah “1” yang berbeda). Pemasangan dapat terjadi jika kedua minterm yang akan dipasangkan hanya memiliki satu variabel yang berbeda.

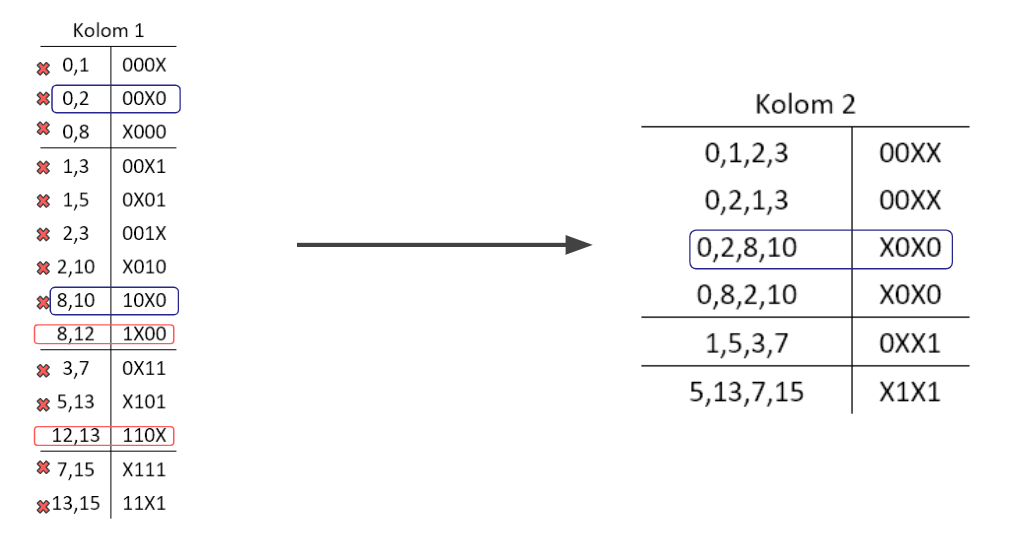
Contohnya pada gambar 3.1.2. kita akan mengambil satu minterm acak dari kelompok dengan “1” berjumlah dua, didapatkan minterm 8 yakni 1000. Lalu kita mengambil minterm kedua dari kelompok dengan “1” berjumlah tiga. Pada kelompok minterm dengan “1” berjumlah tiga, kita cari minterm yang hanya memiliki satu variabel berbeda dengan 1000, didapatkan minterm 12 yakni 1100. Maka dari itu, 8 dan 12 dapat dipasangkan.

Setelah dilakukan pemasangan, variabel yang berbeda dari kedua minterm diganti menjadi tanda X. Dalam contoh diatas, 1000 dan 1100 akan menghasilkan 1X00. Lalu hasil pasangan minterm dimasukkan ke dalam tabel yang baru. Pada Gambar 3.2.1. ditunjukkan bahwa hasil dari pemasangan pada kolom 0 akan dimasukkan ke dalam kolom baru yakni kolom 1. Hal tersebut dilakukan berulang sampai tidak ada lagi minterm yang dapat dipasangkan.



Gambar 3.2.1. Proses Pemasangan Minterm 8 dan 12.

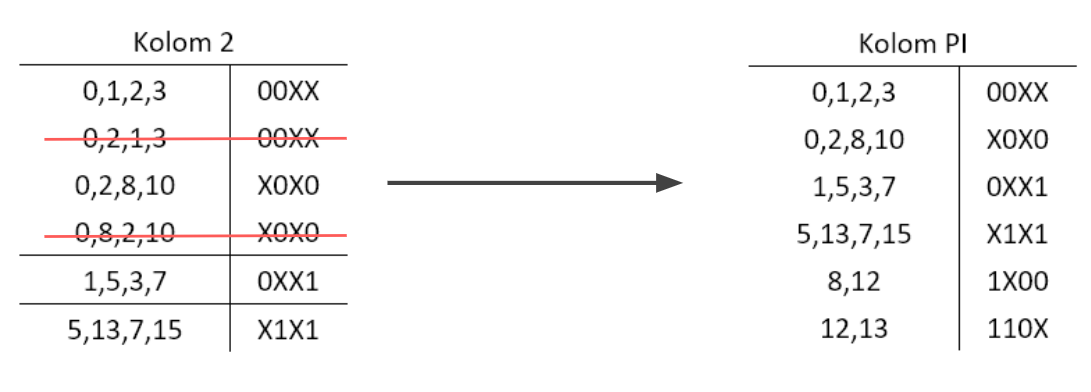
Jika pada saat proses pemasangan di tabel baru terdapat beberapa minterm yang tidak memiliki pasangan, maka minterm tersebut merupakan Prime Implicant (PI). Pada Gambar 3.2.2. 8,12 dan 12,13 adalah PI, karena mereka tidak bisa dipasangkan oleh siapapun. Hasil dari pemasangan pada kolom 1 ditaruh di kolom baru yakni kolom 2. PI tidak perlu dimasukkan ke dalam kolom 2.



Gambar 3.2.2. Lanjutan Pemasangan Minterm yang Menghasilkan Prime Implicant.

## Langkah 3: Melakukan Penghilangan Duplikat

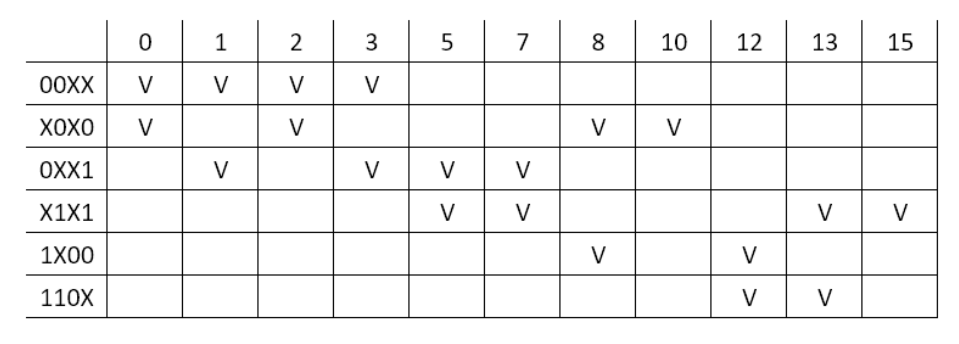
Setelah didapatkan kolom yang berisikan minterm-minterm yang tidak dapat dipasangkan lagi, maka akan dilakukan penyederhanaan dengan menghilangkan duplikat. Term dengan lebih dari satu “X” akan memiliki duplikat. Maka dari itu, kita harus menghilangkan duplikat tersebut agar didapatkan Prime Implicant yang baru. Hasil dari term yang tidak terhilangkan adalah PI. Kita masukkan ke dalam kolom baru, yakni kolom PI. Pada kolom PI, kita tambahkan dua PI yang sudah kita dapatkan dari kolom 1 pada gambar 3.2.2.



Gambar 3.3.1. Proses Penghilangan Duplikat dan Membuat Kolom PI

## Langkah 4: Membuat PI Chart

Kolom PI yang telah didapatkan merupakan informasi yang kita gunakan untuk membuat PI Chart. PI chart ini terdiri dari kolom dan baris. Kolom dari PI Chart adalah minterm-minterm yang diberikan pada soal. Dan baris-nya diisi oleh PI. Setiap baris akan dilakukan pencontrengan jika PI tersebut memiliki anggota minterm pada kolom.

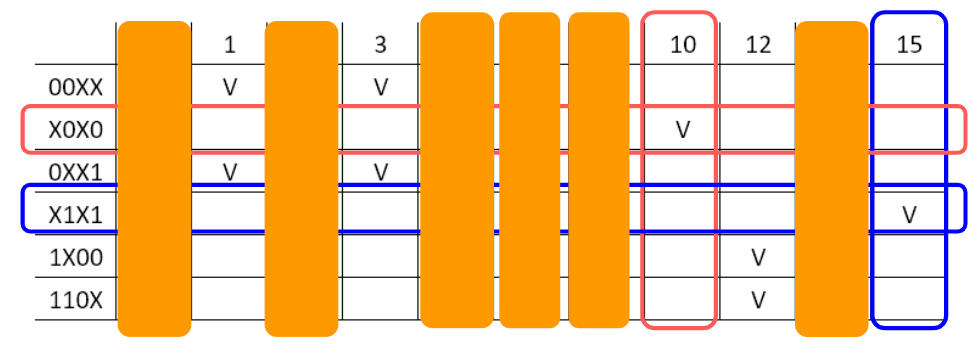


Gambar 3.4.1. PI Chart

## Langkah 5: Mencari Essential Prime Implicant

PI Chart akan digunakan untuk mencari Essential Prime Implicant. Hal ini dapat dilakukan dengan memilih minterm yang hanya terdapat pada satu PI. Pada kasus kita, minterm 10 dan 15 merupakan minterm yang hanya terdapat pada satu PI. Setelah ditandai, hapus semua minterm yang anggotanya terdapat pada kolom yang bersangkutan dengan minterm 10 dan 15. Tak lupa, hapus juga minterm 10 dan 15 juga kolom yang bersangkutan dengan kedua minterm teresebut.

Dari proses tersebut, PI yang bersangkutan dengan minterm 10 dan 15 adalah Essential Prime Implicant (EPI). X0X0 dan X1X1 dimasukkan ke dalam list EPI

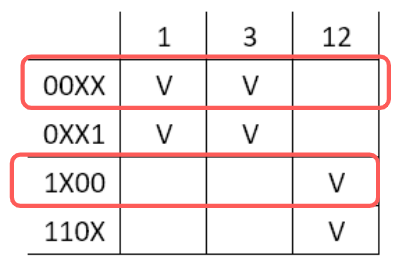


Gambar 3.5.1. Mencari Essential Prime Implicant.

Pada gambar 3.5.1 dapat dilihat bahwa kita menghapus kolom 10 dan 15 karena mereka berdua hanya memiliki satu anggota. Kita juga menghapus baris X0X0 dan X1X1 dikarenakan mereka memiliki anggota minterm 10 dan 15. Dan yang terakhir, kita menghapus semua kolom yang bersangkutan dengan baris X0X0 dan X1X1 (yang ditutupi oleh warna oranye).

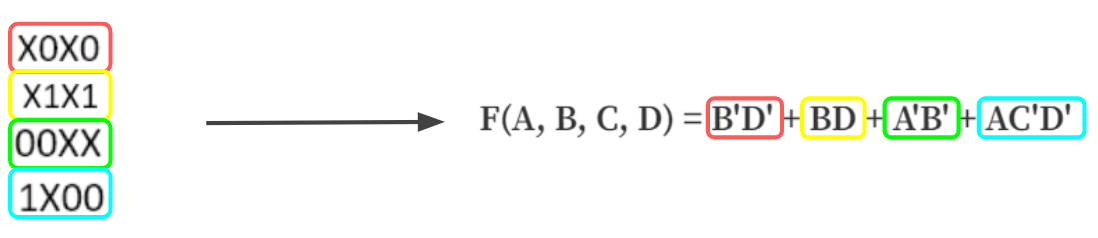
## Langkah 6: Menurunkan PI Chart dan Mentransformasi EPI Menjadi Boolean

Setelah melakukan penghapusan, didapatkan PI Chart yang telah disederhanakan. Dari PI Chart yang telah disederhanakan tersebut, kita dapat mengambil EPI yang tersisa. Yakni dengan cara mencari PI yang menutupi semua minterm yang tersisa. Jika tidak ada, maka pilihlah PI yang paling banyak menutupi minterm, lalu hapus minterm dan PI tersebut dan masukkan PI tersebut ke dalam list EPI. Jika terdapat PI yang memiliki jumlah minterm sama, maka pilihlah PI dengan besaran terkecil.



Gambar 3.6.1. PI Chart yang Telah Disederhanakan

Setelah semua EPI terkumpul, maka akan dilakukan Langkah terakhir yakni menurunkan EPI menjadi bentuk Boolean. Hal tersebut dapat dilakukan dengan cara melihat bilangan pada EPI, jika bilangan adalah “X” maka hiraukan, jika “0” berarti variabel tersebut di-*not*-kan, dan “1” berarti variabel tersebut tidak di-*not­­*-kan.



Gambar 3.6.2. Mentransformasi EPI Menjadi Boolean

Maka dari contoh kasus diatas, didapatkan hasil penyederhanaan persamaan fungsi Boolean adalah

**F(A, B, C, D) = B’D’ + BD + A’B’ + AC’D**

# Penjelasan Program

## Struktur Data yang Digunakan

* Array
* Integer
* Double
* Char

## Daftar Variabel dan Fungsi/Prosedur

Pada program minimasi logika yang telah dibuat, terdapat 6 fungsi yang telah dibuat dan beberapa variabel yang dideklarasikan di luar fungsi main. Hal tersebut dilakukan agar memudahkan pemanggilan variabel di antara fungsi, sehingga efisien dalam kompleksitas ruang.

### Daftar Variabel

Terdapat 24 variabel yang dideklarasi di luar fungsi main, di antaranya yaitu, 11 variabel berupa tipe data integer, terdiri dari variabel NumberOfVariable, NumberOfAllMinterm, NumberOfDontCare, NumberOfRemainingMT, NumberOfRemainingPI, NumberOfPI, NumberOfEPI, PotEPINo, NumberOfPossibleEPI, MinimumNo, dan Groupable. 7 variabel berupa tipe data pointer of integer, terdiri dari variable MintermIndicesDecimal, MintermIndicesDecimal\_DontCare, NumberCounter, ReducedPIChart\_X, For, dan NoOfPIForEPI. 6 variabel berupa tipe data pointer of pointer of integer, yaitu Minterm\_Binary, PI\_Index, EPI\_Index, ReducedPIChart\_Y, ReducedPIChart, dan Potential\_EPI. Kemudian ada satu variable pointer of pointer of pointer of pointer of integer, yaitu variable Column.

### Daftar Fungsi

Terdapat 6 fungsi pada program yang telah dibuat, yaitu

* + DecimalToBinary

DecimalToBinary adalah fungsi untuk mengubah bilangan desimal ke binary. Fungsi ini tidak meminta input apapun. Fungsi ini melakukan proses dengan variabel yang sebelumnya telah dideklarasikan di luar fungsi main dan juga fungsi DecimalToBinary. Pada program ini, fungsi ini dipanggil hanya sekali untuk memasukkan data ke variabel Minterm\_Binary.

* + IsDontCare

IsDontCare adalah fungsi untuk mengecek apakah minterm merupakan Don’t Care atau tidak. Fungsi ini akan me-*return* TRUE jika input yang dimasukkan merupakan Don’t Care dan akan memberikan hasil FALSE jika inputnya bukan Don’t Care.

* + OneCounter

OneCounter adalah fungsi untuk menghitung satu pada binary. Fungsi ini meminta 2 input berupa array of integer dan integer, kemudian fungsi ini akan me-*return* banyaknya angka 1 pada angka binary yang diinput.

* + Combination

Combination adalah fungsi untuk melakukan kombinasi pada kolom. Fungsi ini meminta 3 input berupa integer dan akan me-*return* nilai kemungkinan kombinasi yang memungkinkan.

* + IsPowerOfTwo

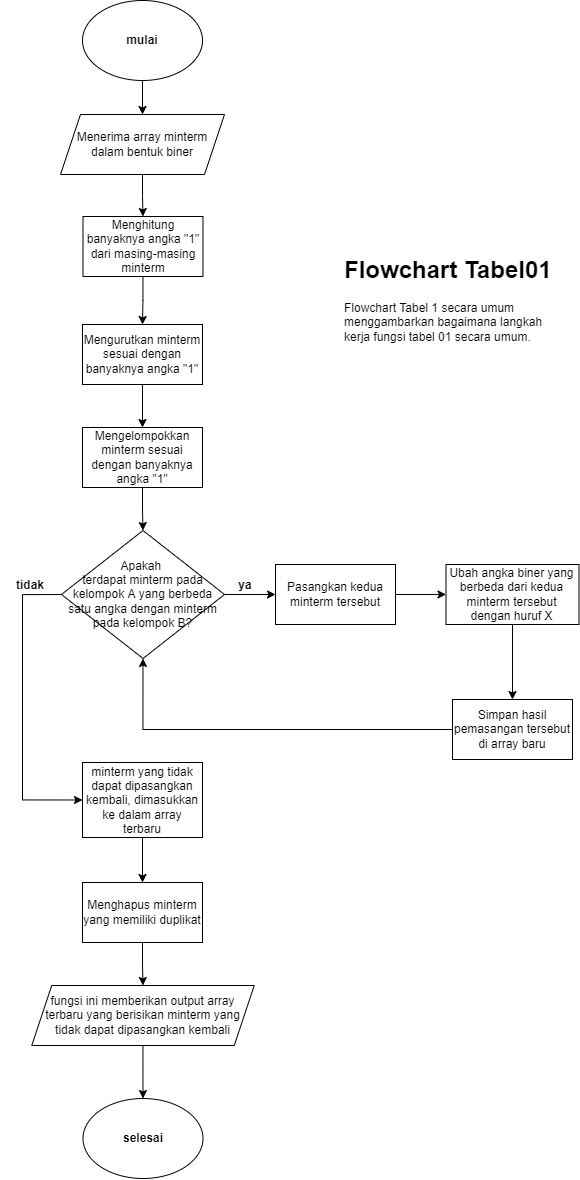
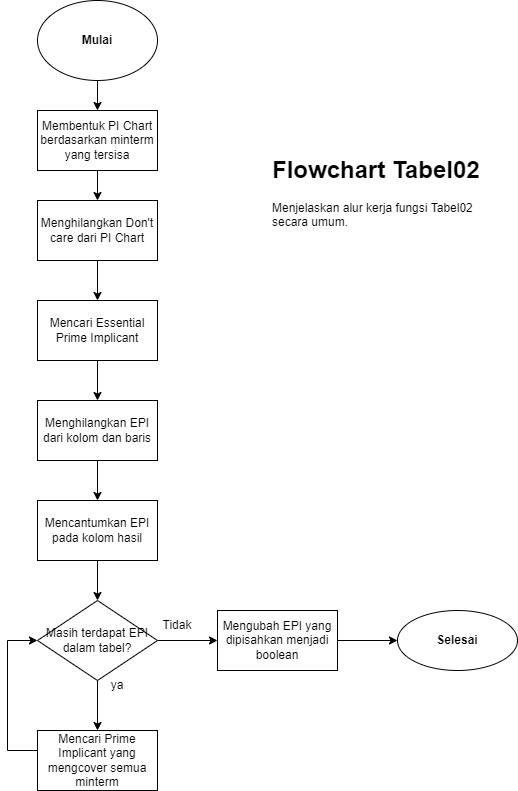
IsPowerOfTwo adalah fungsi untuk memeriksa apakah bilangan merupakan 2 pangkat dari bilangan yang diberikan. Fungsi ini menerima satu input berupa integer dan akan me-*return* TRUE jika benar dan FALSE jika salah.

* + Recursion\_For\_Loop

Recursion\_For\_Loop adalah fungsi untuk membuat rekursi dalam for loop. Fungsi ini meminta satu input berupa integer dan akan melakukan proses rekursif sesuai dengan input integer yang dimasukkan.

## Alur Program

Seperti pada alur program pada biasanya, program minimasi logika yang telah dibuat melalui tahapan input, proses, dan output. Flowchart pada program minimasi logika yang telah dibuat seperti pada gambar berikut.

### Input

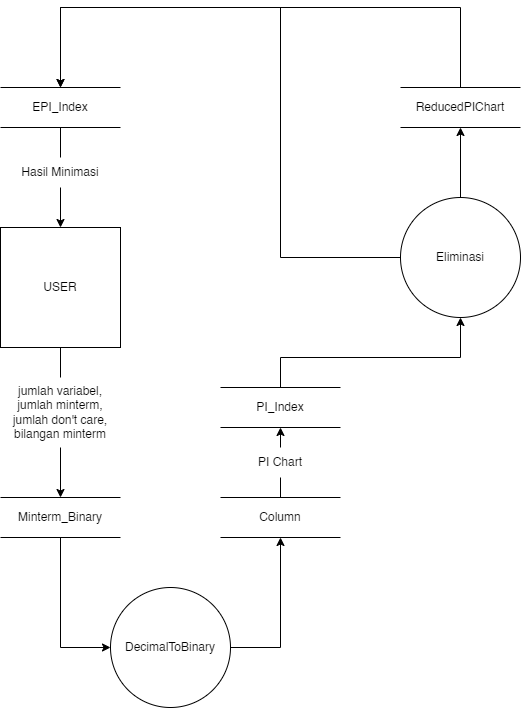
Program pertama dimulai dengan meminta input untuk memasukkan jumlah variabel, banyaknya minterm, banyaknya jumlah minterm Don’t Care, bilangan minterm-nya, dan bilangan minterm yang don’t care-nya.

### Proses dan Output

Setelah mendapatkan data yang diperlukan, semua minterm yang telah diinput akan diubah menjadi bilangan biner terlebih dahulu, dalam konteks program akan disimpan ke dalam variable Minterm\_Binary. Setelah minterm diubah menjadi bilangan biner, masing-masing minterm yang bersebelahan akan dicocokkan untuk proses mencari Prime Implicant (PI). Pada saat melakukan proses ini, data-data yang berhasil diproses dimasukkan ke dalam variabel Column, jika terdapat data yang sama atau duplikat maka akan dihapus. Kemudian dibuat PI Chart dan mengeliminasi kolom dan baris, pada saat melakukan proses ini program menyimpan data pada variabel PI\_Index dan EPI\_Index. Setelah dilakukan eliminasi, PI Chart sisanya disimpan ke dalam variabel ReducedPIChart. Kemudian, dari variabel ReducedPIChart diambil lagi datanya dan disimpan ke dalam EPI\_Index. Dari variabel inilah hasil minimasinya ditampilkan.

## Data Flow Diagram

Data Flow Diagram pada program minimasi logika yang telah dibuat adalah seperti pada gambar berikut.



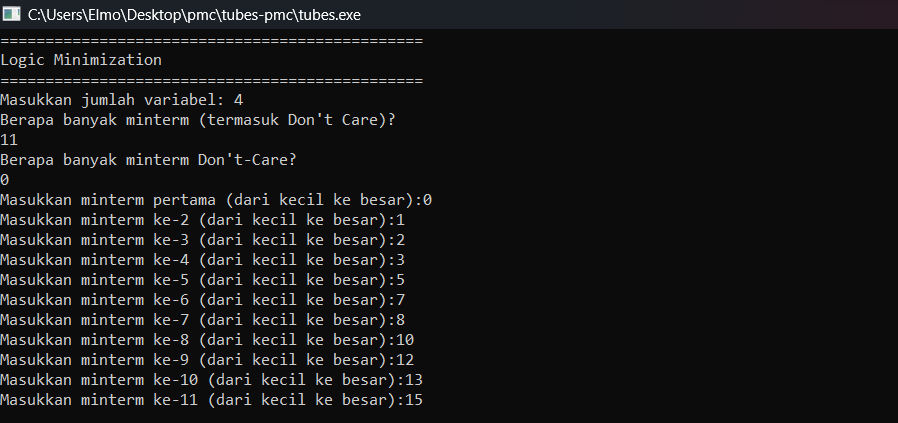
# Pengujian Program

Untuk Tugas Besar EL2008 – Pemecahan Masalah dengan C, kelompok ini membuat program minimisasi logika dalam bahasa pemrograman C dengan aplikasi CodeBlocks, yaitu sebuah IDE untuk C/C++. Selain itu, digunakan juga Visual Studio Code sebagai aplikasi tambahan.

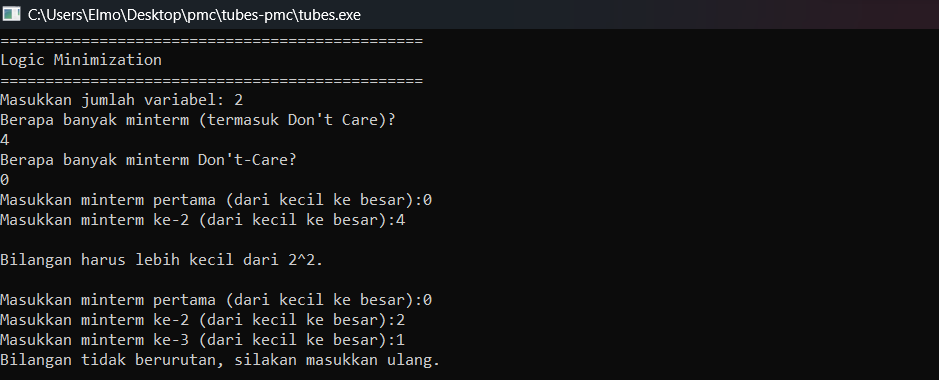
Misalkan digunakan fungsi logika:

**F (A, B, C, D) = Σ (0, 1, 2, 3, 5, 7, 8, 10, 12, 13, 15)**

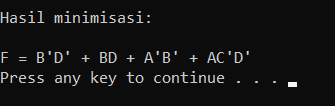
Program akan menerima input berupa banyak variabel, banyak minterm (termasuk Don’t-Care), dan minterm yang Don’t-Care jika ada. Program ini dapat menerima input variabel sebanyak mungkin, namun semakin banyak variabel yang ada, semakin lama program akan memproses fungsi tersebut. Hal ini disebabkan oleh kompleksitas algoritma yang masih cukup besar.



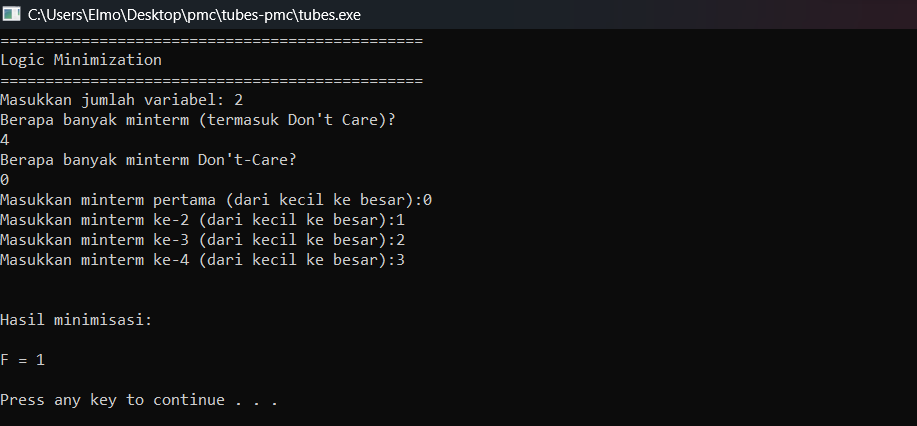
Program akan menerima input minterm yang sudah termasuk Don’t Care. Jika input minterm tidak sesuai (contohnya minterm tidak berurutan dari kecil sampai besar atau melebihi dua pangkat jumlah variabel), maka program akan memberi pesan peringatan lalu akan menerima input lagi sampai benar.



Program lalu akan menerima banyak minterm yang Don’t Care. Jika input 0, maka program tidak akan menanyakan minterm yang Don’t Care. Penerimaan minterm akan berulang sampai sesuai dengan jumlah minterm yang ada. Setelah setiap input sudah dilakukan, program lalu akan memproses input-input tersebut untuk melakukan minimisasi logika, Setelah proses tersebut, program akan mengoutput hasil minimisasi.



Jika jumlah minterm sama dengan dua pangkat jumlah variabel, program akan mengoutput:



Hal ini disebabkan karena fungsi tidak perlu disederhanakan lagi.

Program akan diterminasi setelah user memencet tombol apa saja pada keyboard.

# *Lesson Learned*

1. Penulis dapat memahami lebih lanjut tentang pemecahan masalah dengan bahasa C melalui tugas besar yang telah diberikan. Minimisasi logika yang sudah dipelajari pada semester lalu dapat diimplementasikan ke dalam suatu program dengan menggunakan salah satu metode optimisasi logika, yaitu Quine-McCluskey Method atau metode Tabular.
2. Penulis mendapatkan pengalaman dalam aspek kerja sama. Pengerjaan tugas besar dapat dilakukan secara efisien dengan adanya komunikasi antar anggota kelompok. Selain itu, setiap anggota kelompok dapat berkontribusi dengan baik dengan adanya kerja sama.
3. Agar pekerjaan terasa lebih ringan, maka pekerjaan kelompok harus dikerjakan dengan melakukan pembagian tugas. Selain itu jika terdapat materi yang kurang dipahami atau terdapat kendala, alangkah baiknya untuk meminta bantuan orang yang lebih mengerti. Di sini penulis belajar bagaimana sebaiknya kerja secara kelompok itu dilakukan. Terarah, transparan, dan saling membantu.
4. Mempelajari konsep dari suatu materi pada perkuliahan akan membantu penulis jika suatu saat membutuhkan suatu ide akan penyelesaian suatu masalah. Pada kasus ini, penulis sebelumnya sudah mempelajari materi minimisasi gerbang logika dari mata kuliah Sistem Digital di semester 3 teknik elektro. Karena sudah mempelajari hal tersebut, ketika tugas ini diberikan penulis pun langsung terpikir akan materi yang terdapat di mata kuliah tersebut, meskipun sudah lupa detil materinya.
5. Pada sebuah pengerjaan tugas, pekerjaan yang diberikan sebaiknya dicicil. Jangan sampai pekerjaan baru dikerjakan mendekati deadline. Jika pekerjaan baru dikerjakan saat deadline, maka akan menyulitkan kita sendiri dan hasil tugasnya pun tidak maksimal.

# Kesimpulan

Pada Pengujian Program, dapat terlihat bahwa program yang dijalankan sudah sesuai dengan dasar teori minimisasi tabular dengan menggunakan metode QM. Dari pengujian tersebut, penulis dapat menyimpulkan bahwa salah satu algoritma minimisasi persamaan logika adalah dengan metode Quine McCluskey, atau yang bisa juga disebut metode Tabular. Setiap langkah metode QM diilustrasikan secara paralel dengan struktur data dan algoritma yang sesuai yang digunakan dalam program agar lebih mudah dipahami. Program ini juga dapat digunakan sebagai contoh yang baik untuk mempraktekkan program implementasi metode minimisasi logika.

# Pembagian Tugas

Pembagian tugas kelompok ini dalam mengerjakan Tugas Besar EL2008 Pemecahan Masalah dengan C adalah sebagai berikut:

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Anggota** | **Tugas** |
| 1. | Kadhan Dalilurahman (13220001) | File Presentasi, Tester, Laporan (Daftar Isi, Dasar Teori, Lesson Learned) |
| 2. | Elmo Ryaner Panggabean (13220012) | File Presentasi, Koding, Laporan (Abstrak, Pendahuluan, Pengujian Program) |
| 3. | Maheswara Apta Adiyatma (18320005) | File Presentasi, Tester, Laporan (Penjelasan Program, Kesimpulan) |

# Daftar Pustaka

Zvonko G. Vranesic dan Stephen D. Brown. *Fundamentals of Digital Logic with VHDL Design*. United States, McGraw-Hill, 2009.

<https://arxiv.org/abs/1410.1059>

<https://steemit.com/logic/@drifter1/logic-design-quine-mccluskey-tabular-simplification-method>