

TUGAS BESAR EL2008 – PEMECAHAN MASALAH DENGAN C

Program Minimisasi Logic

Farhan Hakim Iskandar (13220007)

Fitra Nurindra (13220011)

Muhammad Daffa Daniswara (13220043)

EL2008-Pemecahan Masalah dengan C

Teknik Elektro - Sekolah Teknik Elektro dan Informatika ITB

Abstrak

Pada tugas besar PMC ini, mahasiswa dibagi berkelompok untuk mengeksplor metode-metode dalam minimisasi logic dan mengimplementasikannya pada program dengan bahasa C. Spesifikasi dibebaskan berkelompok.

Kata kunci: minimisasi, boolean, tabular, quine-mccluskey.

1. PENDAHULUAN

Logic minimization adalah proses menyederhanakan ekspresi boolean. Tujuan dari logic minimization ialah untuk memudahkan serta meningkatkan efisiensi suatu sistem.

Pada saat eksplorasi metode, kami mencoba beberapa algoritma logic minimization, yakni aljabar boolean, Karnaugh map, dan Quine-McCluskey atau metode tabular. Kami setuju untuk menggunakan algoritma Quine-McCluskey untuk diimplementasikan ke program.

Algoritma Quine-McCluskey ini dipilih karena dibanding dengan metode lain, metode ini lebih mudah diimplementasikan ke dalam program dan direalisasikan secara komputasional. Hal ini karena metode ini dilakukan banyak pengulangan atau looping. Selain itu, algoritma ini juga cocok untuk penyederhanaan logic dengan variabel yang banyak. Tahapan metode tabular ini adalah penentuan prime implicant dengan mencari matched pairs dari minterm. Selanjutnya penentuan minimum cover, yakni memilih prime implicant yang telah didapat untuk mencakup semua suku dan menghasilkan fungsi yang paling sederhana.

2. SPESEFIKASI

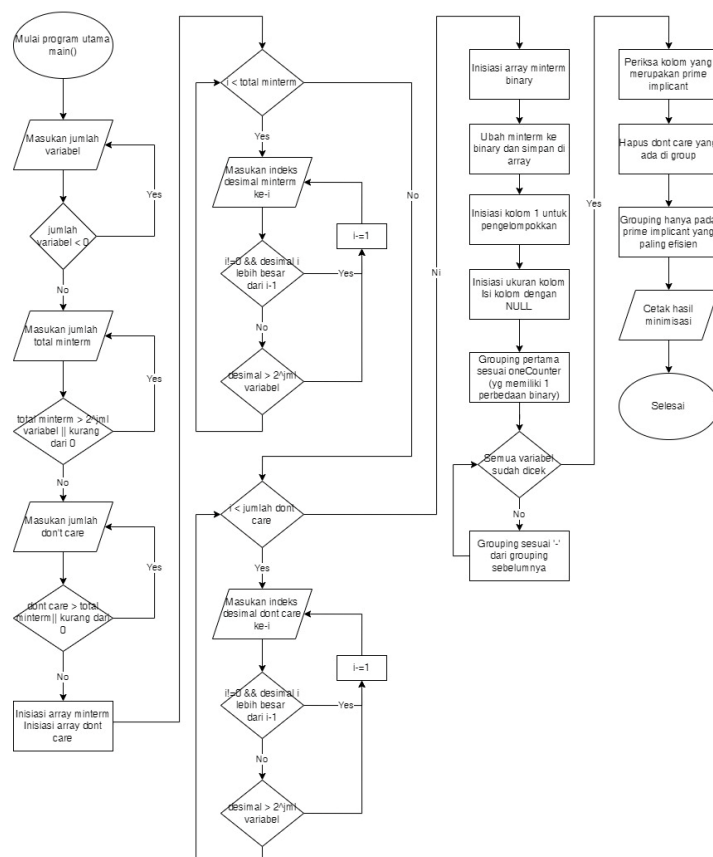
Spesifikasi dari program yang telah dibuat adalah sebagai berikut :

1. Jumlah variabel input maksimum 10 variabel

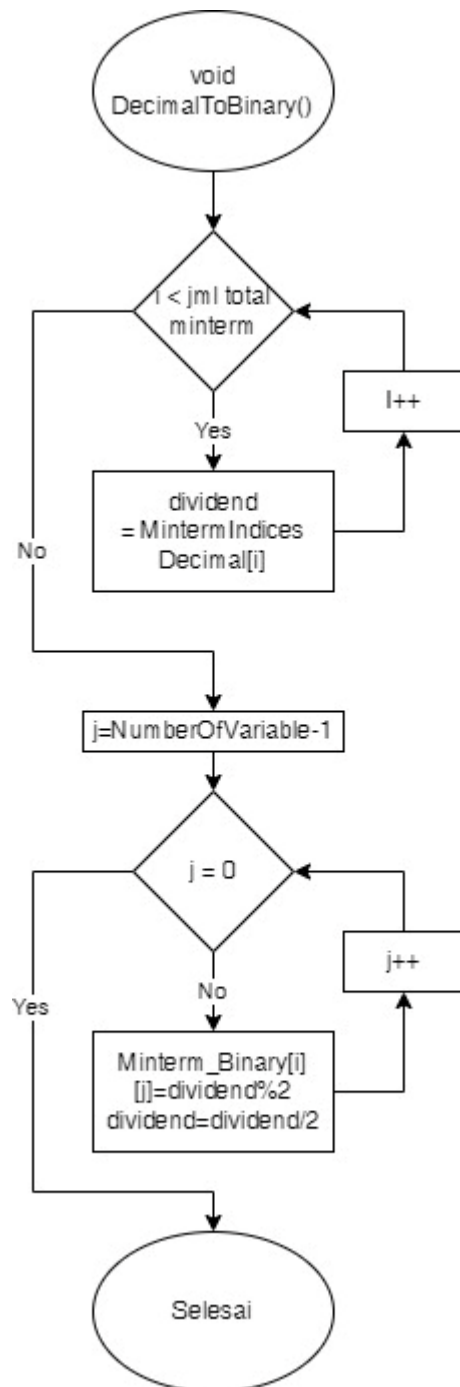
2. Input program adalah jumlah variabel, total jumlah minterm, jumlah don't care, indeks minterm, dan indeks don't care
3. Output dari program adalah hasil minimisasi

3. FLOWCHART

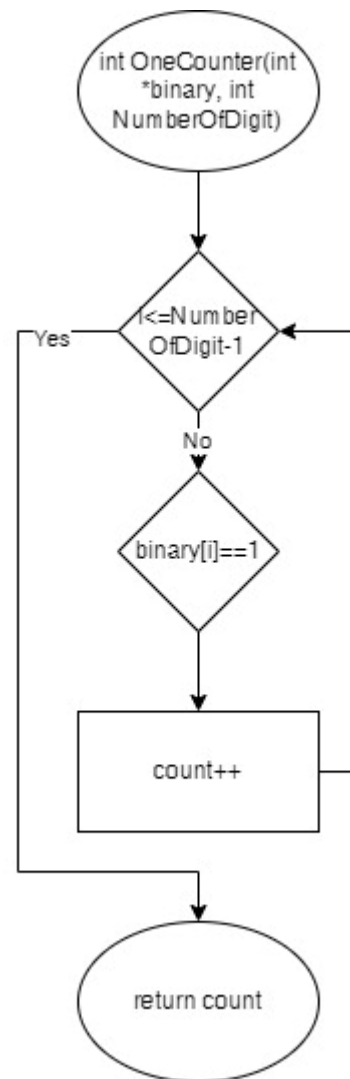
3.1 MAIN



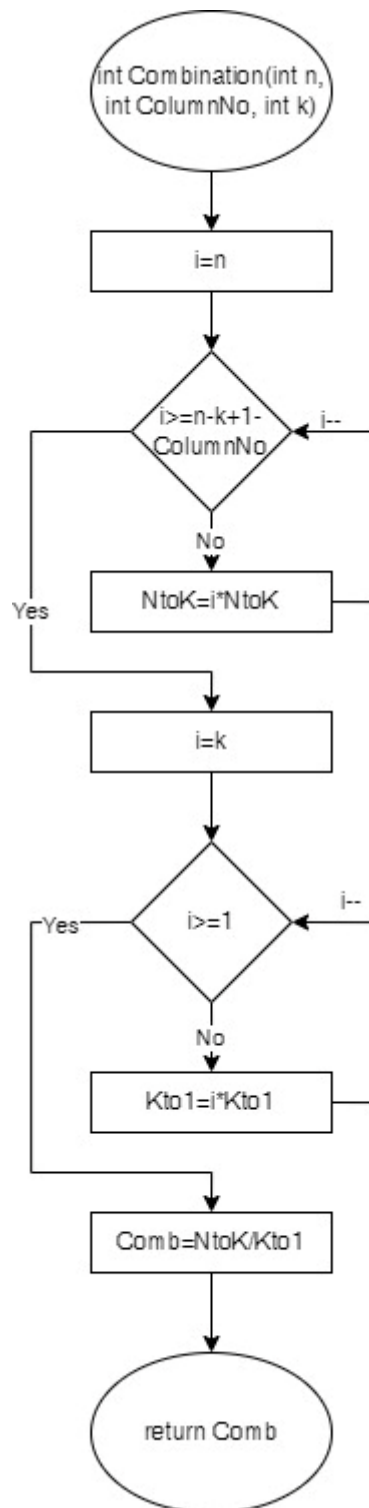
3.2 DECIMALTOBINARY()



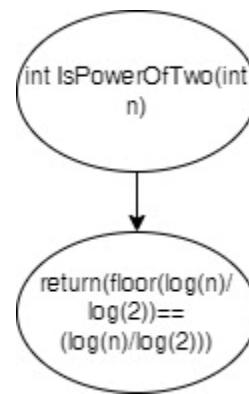
3.3 ONECOUNTER()



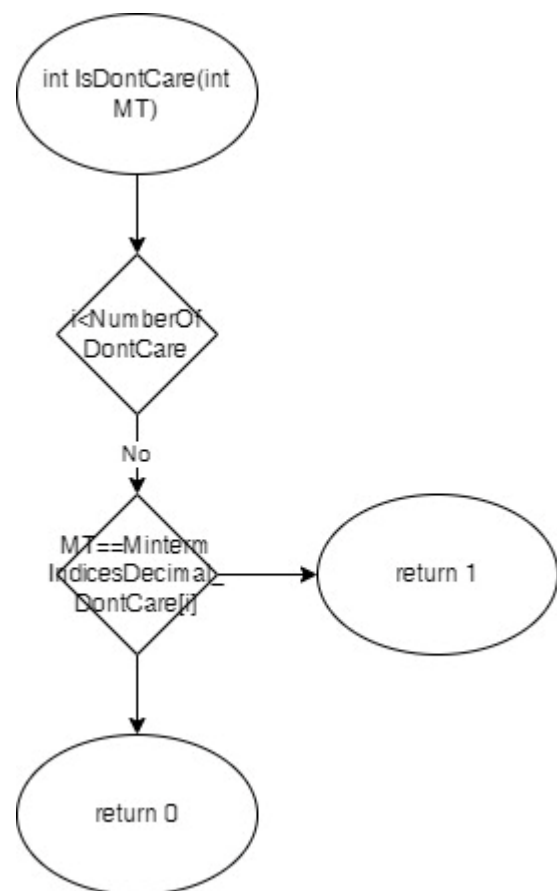
3.4 COMBINATION()



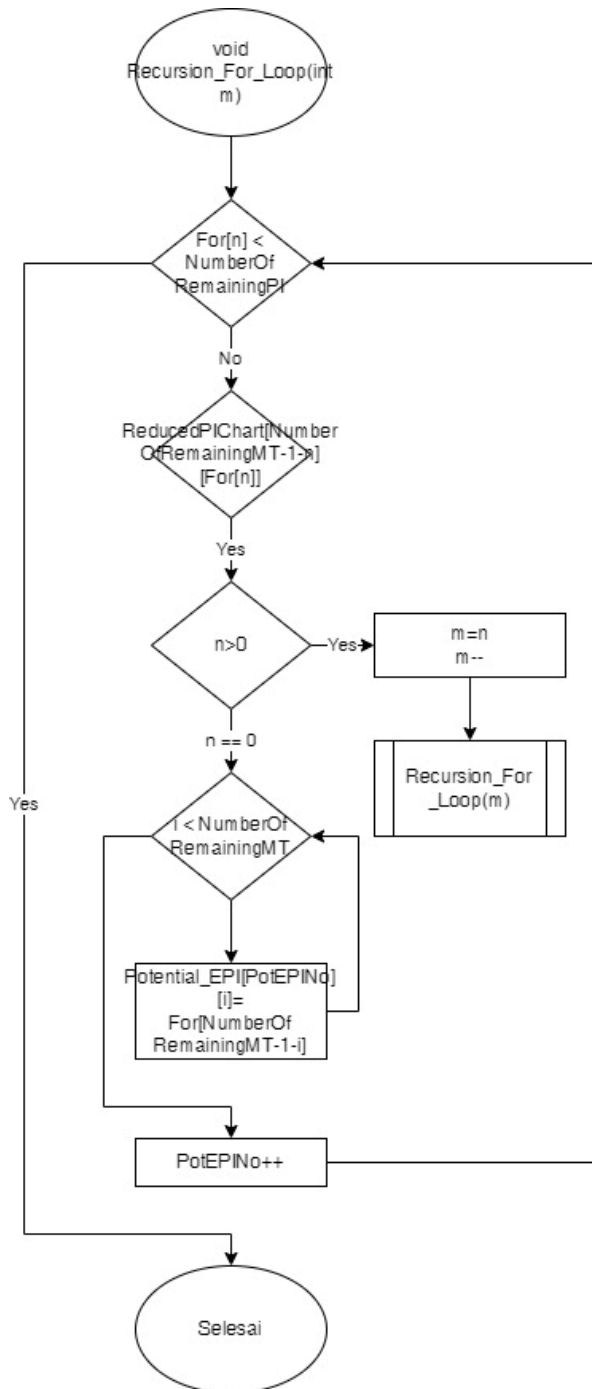
3.5 ISPOWEROFTWO()



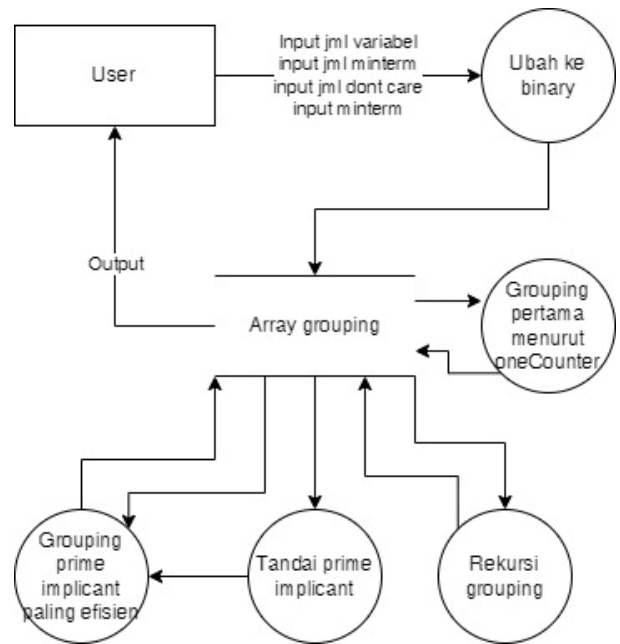
3.6 ISDONTCARE()



3.7 RECURSION_FOR_LOOP()



4. DATA FLOW DIAGRAM



5. RENCANA PENGUJIAN

Untuk mengukur keakuratan program, dilakukan benchmarking atau perbandingan hasil minimisasi logic dari program yang telah dibuat dengan logic minimization calculator dari internet.

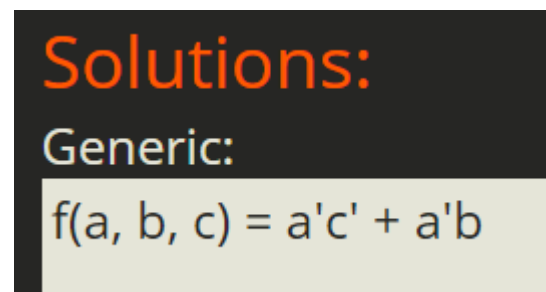
Pengujian dibatasi dengan input maksimum sebanyak 10 variabel serta dapat menggunakan don't care.

6. HASIL PENGUJIAN

Kasus 1

$$F(a, b, c) = m(0, 3) + d(2)$$

Hasil dari logic minimization calculator internet :



Hasil program :

```
Selamat Datang di Program Minimisasi Logic.
Silakan Masukkan Informasi Mengenai Ekspresi Logic yang Ingin Diminimisasi

Masukkan Banyak Variabel: 3
Masukkan Banyak Minterm Keseluruhan (Termasuk Don't Care Minterm): 3
Masukkan Banyak Don't Care Minterm: 1

Masukkan Minterm Keseluruhan (Termasuk Don't Care Minterm)!
Masukkan Minterm ke-1 (Dalam Urutan Meningkat): 0
Masukkan Minterm ke-2 (Dalam Urutan Meningkat): 2
Masukkan Minterm ke-3 (Dalam Urutan Meningkat): 3

Masukkan Minterm yang Merupakan Don't Care Minterm!
Masukkan Don't Care Minterm ke-1 (Dalam Urutan Meningkat): 2
Fungsi Logika Setelah Minimisasi Dalam Bentuk SOP:

A'C' + A'B
Tekan Tombol Apapun Untuk Keluar!
```

$$A'C' + A'B$$

Kasus 2

$$F(a, b, c) = m(0, 1, 4, 7) + d(3)$$

Hasil dari logic minimization calculator internet :

Solutions:
Generic:
$$f(a, b, c) = b'c' + bc + a'b'$$

Hasil program :

```
Selamat Datang di Program Minimisasi Logic.
Silakan Masukkan Informasi Mengenai Ekspresi Logic yang Ingin Diminimisasi

Masukkan Banyak Variabel: 3
Masukkan Banyak Minterm Keseluruhan (Termasuk Don't Care Minterm): 5
Masukkan Banyak Don't Care Minterm: 1

Masukkan Minterm Keseluruhan (Termasuk Don't Care Minterm)!
Masukkan Minterm ke-1 (Dalam Urutan Meningkat): 0
Masukkan Minterm ke-2 (Dalam Urutan Meningkat): 1
Masukkan Minterm ke-3 (Dalam Urutan Meningkat): 3
Masukkan Minterm ke-4 (Dalam Urutan Meningkat): 4
Masukkan Minterm ke-5 (Dalam Urutan Meningkat): 7

Masukkan Minterm yang Merupakan Don't Care Minterm!
Masukkan Don't Care Minterm ke-1 (Dalam Urutan Meningkat): 3
Fungsi Logika Setelah Minimisasi Dalam Bentuk SOP:

B'C' + BC + A'B'
Tekan Tombol Apapun Untuk Keluar!
```

$$B'C' + BC + A'B'$$

Kasus 3

$$F(a, b, c, d) = m(1, 3, 7, 12, 13, 14, 15)$$

Hasil dari logic minimization calculator internet :

Solutions:
Generic:
$$f(a, b, c, d) = a'b'd + ab + a'cd$$

Hasil program :

```
Selamat Datang di Program Minimisasi Logic.
Silakan Masukkan Informasi Mengenai Ekspresi Logic yang Ingin Diminimisasi

Masukkan Banyak Variabel: 4
Masukkan Banyak Minterm Keseluruhan (Termasuk Don't Care Minterm): 7
Masukkan Banyak Don't Care Minterm: 0

Masukkan Minterm Keseluruhan (Termasuk Don't Care Minterm)!
Masukkan Minterm ke-1 (Dalam Urutan Meningkat): 1
Masukkan Minterm ke-2 (Dalam Urutan Meningkat): 3
Masukkan Minterm ke-3 (Dalam Urutan Meningkat): 7
Masukkan Minterm ke-4 (Dalam Urutan Meningkat): 12
Masukkan Minterm ke-5 (Dalam Urutan Meningkat): 13
Masukkan Minterm ke-6 (Dalam Urutan Meningkat): 14
Masukkan Minterm ke-7 (Dalam Urutan Meningkat): 15

Fungsi Logika Setelah Minimisasi Dalam Bentuk SOP:

A'B'D + AB + A'CD
Tekan Tombol Apapun Untuk Keluar!
```

$$A'B'D + AB + A'CD$$

Kasus 4

$$F(a, b, c, d) = m(7, 13, 14, 15) + d(5, 11)$$

Hasil dari logic minimization calculator internet :

Solutions:
Generic:
$$f(a, b, c, d) = bd + abc$$

Hasil program :

```
Selamat Datang di Program Minimisasi Logic.
Silakan Masukkan Informasi Mengenai Ekspresi Logic yang Ingin Diminimisasi

Masukkan Banyak Variabel: 4
Masukkan Banyak Minterm Keseluruhan (Termasuk Don't Care Minterm): 6
Masukkan Banyak Don't Care Minterm: 2

Masukkan Minterm Keseluruhan (Termasuk Don't Care Minterm)!
Masukkan Minterm ke-1 (Dalam Urutan Meningkat): 5
Masukkan Minterm ke-2 (Dalam Urutan Meningkat): 7
Masukkan Minterm ke-3 (Dalam Urutan Meningkat): 11
Masukkan Minterm ke-4 (Dalam Urutan Meningkat): 13
Masukkan Minterm ke-5 (Dalam Urutan Meningkat): 14
Masukkan Minterm ke-6 (Dalam Urutan Meningkat): 15

Masukkan Minterm yang Merupakan Don't Care Minterm!
Masukkan Don't Care Minterm ke-1 (Dalam Urutan Meningkat): 5
Masukkan Don't Care Minterm ke-2 (Dalam Urutan Meningkat): 11
Fungsi Logika Setelah Minimisasi Dalam Bentuk SOP:

BD + ABC
Tekan Tombol Apapun Untuk Keluar!
```

$$BD + ABC$$

7. ANALISIS

Bagian ini akan melakukan analisis terhadap kompleksitas waktu dan ruang dari program logic minimization yang telah dibuat.

Kompleksitas waktu dari program yang dibuat adalah $O(n^2)$, dimana n adalah banyaknya minterm. Lalu kompleksitas ruangnya juga $O(n^2)$.

8. KESIMPULAN

Dari pengerjaan yang telah dilakukan, dapat ditarik kesimpulan sebagai berikut.

- Minimisasi adalah proses penyederhanaan ekspresi boolean, yang mana algoritmanya banyak dan beragam
- Algoritma Quine-McCluskey adalah salah satu algoritma logic minimization yang mudah diimplementasikan ke program karena prosesnya dilakukan berulang serta efektif jika ekspresi boolean input memiliki banyak variabel
- Pada kasus tertentu, program yang telah kami buat tidak dapat meminimisasi logika sesederhana mungkin setelah prime implicant sudah ditemukan. Hal ini bisa terjadi karena proses pencarian minimum cover yang tidak optimal sehingga masih ada redundant group.
- Secara umum, program minimization logic sudah terealisasi sesuai dengan spesifikasi.

DAFTAR PUSTAKA

- [1] https://github.com/Es1chUbJyan9/32bit_Quine-McCluskey_and_Petricks_Method_in_C
- [2] <https://www.incdmtm.ro/editura/documente/pag.%2095-99.%20Software%20for%20The%20Minimization%20of%20The%20Combinational%20Logic%20Functions.pdf>
- [3] <https://www.geeksforgeeks.org/logic-functions-and-minimization/>
- [4] <https://www.sciencedirect.com/topics/computer-science/logic-minimization>
- [5] <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.697.3580&rep=rep1&type=pdf>