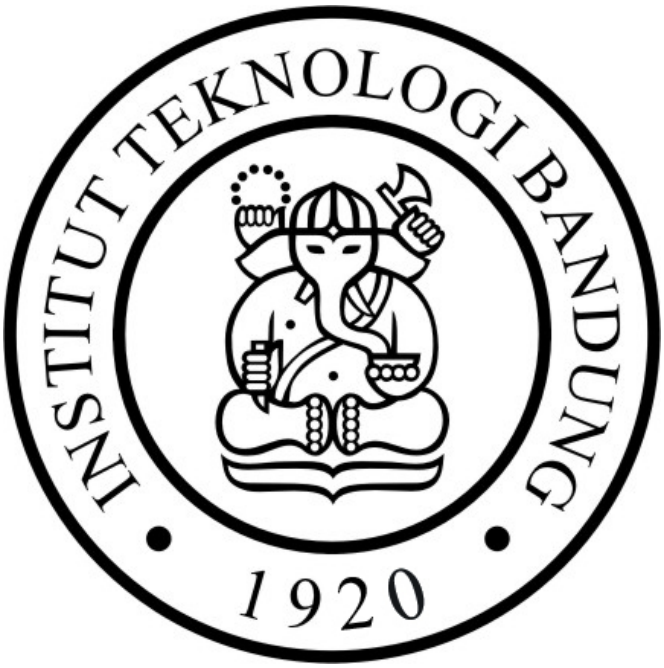
TUGAS BESAR

EL2008 Pemecahan Masalah dengan C

“Eksplorasi Minimisasi Logika”



Disusun oleh:
Fadiah Mumtaz Andevi (18320009)
Tanya Nuhaisy Wulandari (18320017)
Eunike Kristianti (18320019)

Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
2022

Daftar Isi

Laporan Inti

Deskripsi Simulasi

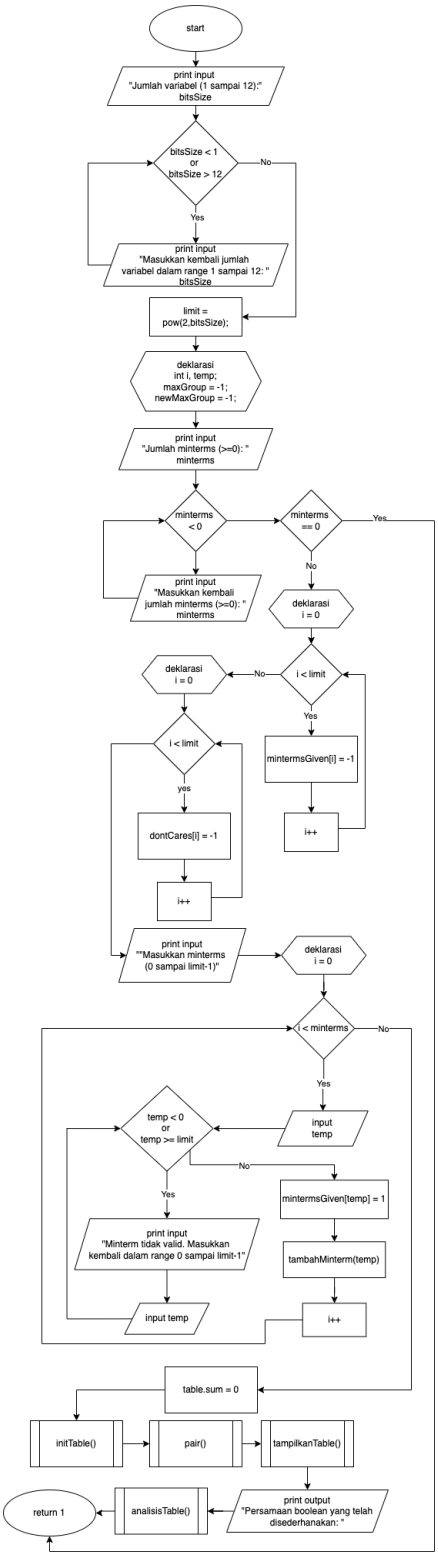
Minimisasi logika merupakan proses untuk menyederhanakan suatu fungsi boolean atau ekspresi algebra [1]. Terdapat tiga cara untuk menyederhanakan fungsi boolean yaitu dengan Teorema Aljabar Boolean, Metode *Karnaugh Map* (*K-Map*), dan dengan Metode Tabular. Setelah melakukan eksplorasi, diputuskan untuk membuat program minimisasi logika menggunakan bahasa C dengan metode tabular.

Program yang akan dibuat adalah sebagai berikut. Program akan menerima input dari pengguna berupa jumlah variabel, jumlah *minterms*, dan *minterms*. Jumlah variabel dari pengguna akan dibatasi dari satu sampai dua belas variabel. Jumlah *minterms* dibatasi dari nol sampai dua pangkat jumlah variabel. Sementara nilai *minterms* dibatasi dari nol sampai dua pangkat jumlah variabel dikurangi satu. Ketiga input tersebut akan diverifikasi program dan pengguna akan diminta input ulang nilainya sampai sesuai dengan ketentuan. Dari ketiga input tersebut, akan dicari *essential prime implicants*-nya yang kemudian akan menjadi solusi sekaligus output dari program ini. Dalam memproses input yang diberikan pengguna, program akan dibagi menjadi 4 langkah. Langkah pertama yaitu program akan mengubah *minterms* dari bentuk desimal menjadi bentuk binernya. Kemudian, bilangan-bilangan biner tersebut akan dikelompokkan ke dalam beberapa grup berdasarkan jumlah angka ‘1’ dalam bilangan binernya. Setelah dikelompokkan, program akan membandingkan seluruh bilangan biner pada grup ke-n dengan seluruh bilangan biner pada grup ke-(n+1). Jika dua bilangan biner yang dibandingkan hanya memiliki satu perbedaan pada nilainya, misal “0001” dengan “0011”, maka kedua bilangan biner tersebut akan dianggap sebagai *matched pair* dan satu angka yang berbeda tersebut akan digantikan menggunakan tanda underscore ‘_’, sehingga menjadi “00_1”. Itu merupakan langkah kedua. Setelah itu, seluruh *matched pair* dari grup ke-n akan dibandingkan dengan seluruh *matched pair* dari grup ke-(n+1). Jika dua *matched pair* yang dibandingkan hanya memiliki satu perbedaan pada nilainya, maka kedua *matched pair* tersebut akan dianggap *matched* dan satu angka yang berbeda tersebut akan digantikan menggunakan tanda underscore ‘_’. Itu merupakan langkah ketiga. Proses membandingkan tersebut akan terus dilakukan sampai tidak ditemukan lagi matched pair. Kemudian, dari angka yang tetap menjadi angka (tidak berubah menjadi underscore), akan ditentukan persamaan aljabar boolean nya untuk kemudian aljabar boolean tersebut akan disebut sebagai *prime implicants*. Pada langkah keempat, akan dibuat tabel *prime implicants* untuk menentukan *essential prime implicants*. *Essential prime implicants* tersebutlah yang kemudian akan ditampilkan menjadi output pada program sebagai *simplified boolean expression* atau persamaan boolean yang telah disederhanakan dari input yang diberikan pengguna. Akan tetapi, untuk mempermudah verifikasi hasil, maka setiap proses yang dilakukan, yang pada program disebut sebagai iterasi, akan ditampilkan juga sebagai output, termasuk dengan tabel *prime implicants* nya.

Jika input jumlah minterm dari pengguna adalah nol, program akan langsung menampilkan hasil “F = 0” dan program berakhir. Sementara itu, jika input jumlah minterm dari pengguna sama dengan dua pangkat jumlah variabel dan input minterms nya merupakan nilai dari nol sampai dua pangkat jumlah variabel dikurangi satu, maka program akan langsung menampilkan hasil “F = 1” dan program berakhir.

Flowchart

1. Main Program



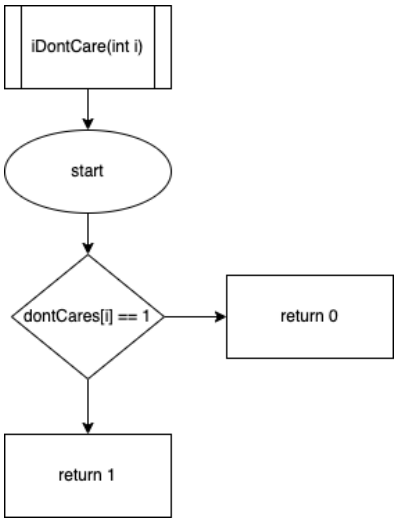
Gambar 1. Flowchart Main Program

Ini merupakan flowchart untuk program utama. Pertama program akan menerima input berupa jumlah variabel. Apabila variabel yang dimasukkan user kurang dari 1 atau lebih dari 12 maka program akan meminta input kembali. Setelah masukkan sesuai syarat maka akan dibuat limit sama dengan 2 pangkat variabel yang diinput tadi. Selanjutnya akan dideklarasi i, temp, maxGroup = -1, dan newMaxGroup = -1.

Setelahnya diminta kembali input user yaitu jumlah minterms dengan syarat minterms harus lebih dari 0. Apabila minterms yang diinput kurang dari 0 maka akan diminta input dari user kembali. Apabila input user minterm = 0 maka program akan selesai. Sedangkan apabila minterms lebih dari 0 program akan melakukan looping yang meminta value untuk minterms.

Setelah memasukkan minterms. Program akan masuk ke fungsi initTable() lalu ke fungsi pair(), tampilkanTable() dimana akan menampilkan output berupa tabel prime implicants, dan juga program akan menampilkan output berupa persamaan boolean yang telah disederhanakan.

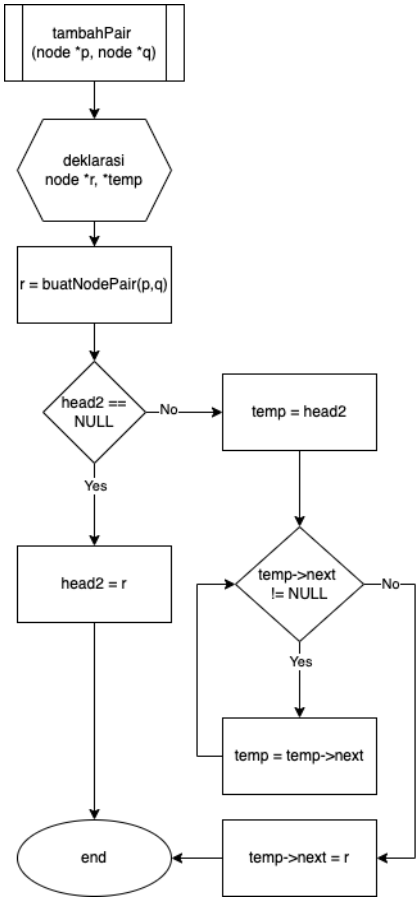
2. Fungsi ifDontCare



Gambar 2. Flowchart Fungsi ifDontCare

Fungsi ini digunakan untuk mengecek apakah minterm merupakan don’t Care. Disini akan dicek minterm satu persatu apakah minterm termasuk dontCare atau tidak.

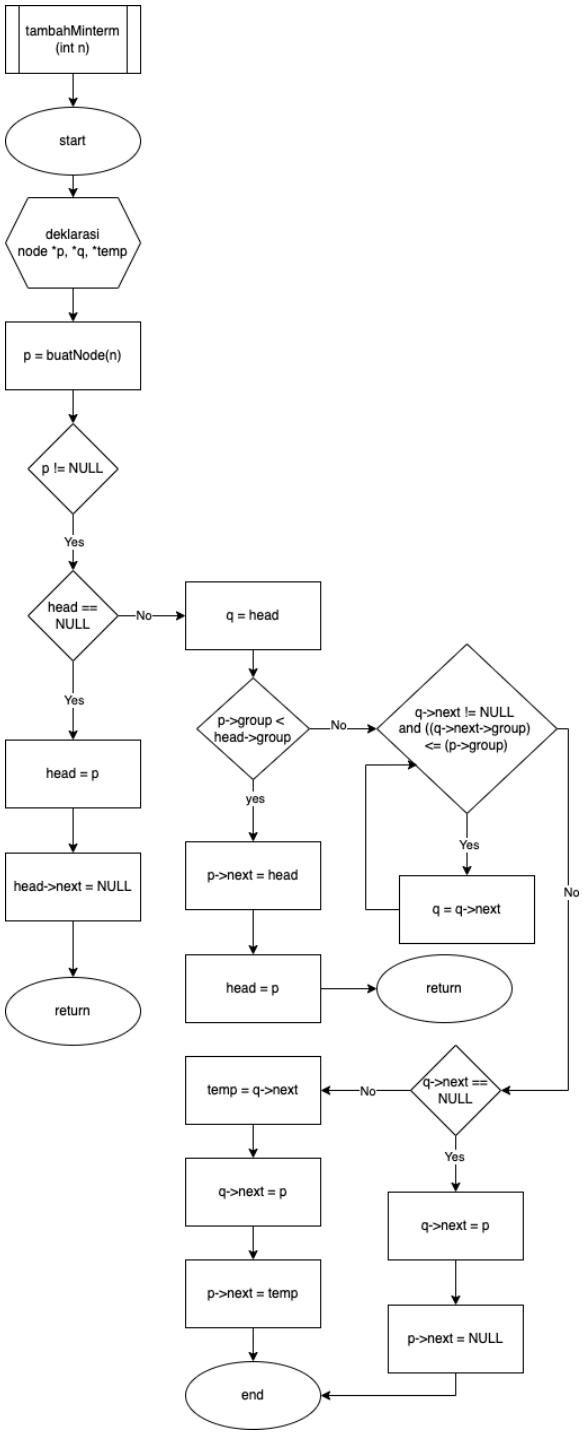
3. Fungsi tambahPair



Gambar 3. Flowchart Fungsi tambahPair

Fungsi ini untuk membuat linked list untuk menyimpan matched pairs. Fungsi ini memiliki proses dimana r akan dimasukkan ke fungsi buatNodePair dengan parameter p dan q. Lalu apabila head2 berupa NULL maka head2 akan menjadi r sedangkan apabila tidak maka akan dicari sampai bertemu dengan NULL.

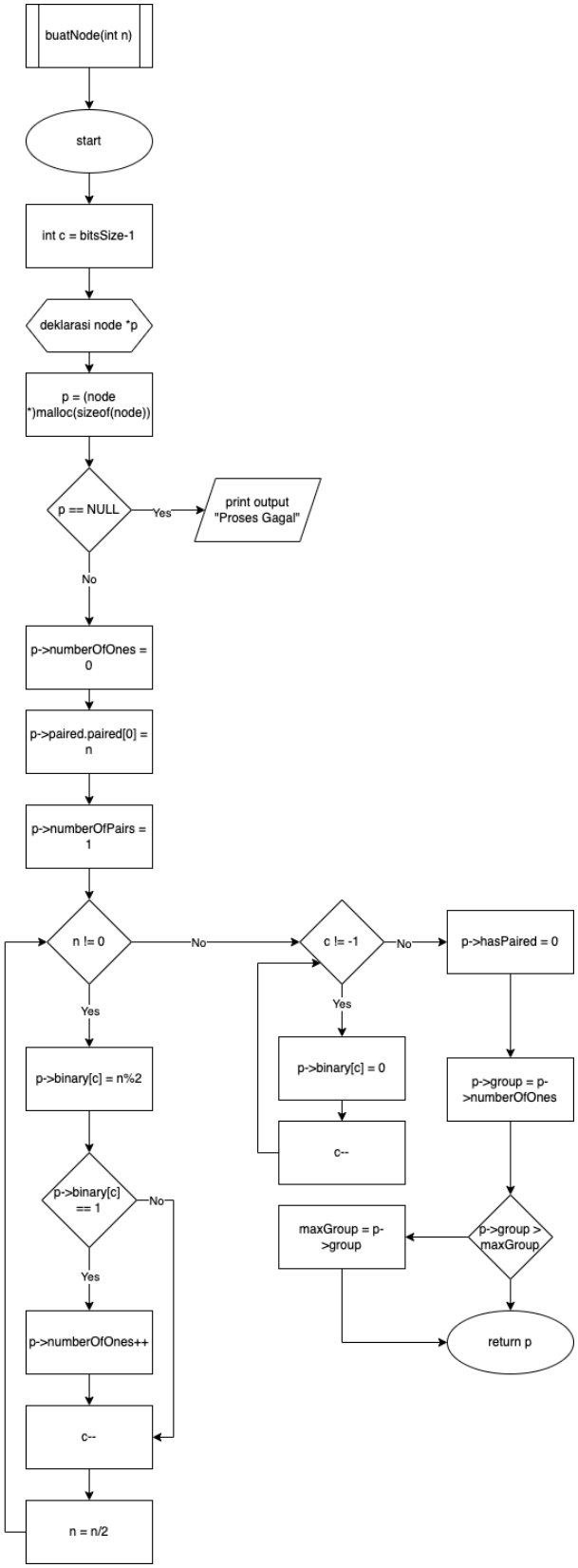
4. Fungsi tambahMinterm



Gambar 4. Flowchart Fungsi tambahMinterm

Fungsi ini untuk membuat linked list untuk menyimpan minterms. Apabila nilai p tidak sama dengan null maka akan dicek apakah nilai head merupakan null. Dimana apabila iya maka akan dibuat head = p, sedangkan apabila tidak akan dicek untuk nilai p selanjutnya.

5. Fungsi buatNode

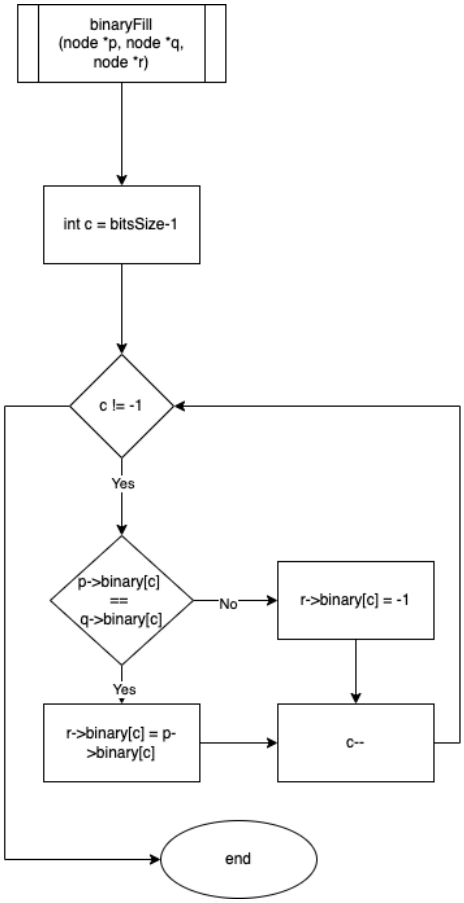


Gambar 5. Flowchart Fungsi buatNode

Fungsi untuk membuat node baru. Pertama akan dicek apakah r sama dengan NULL, dimana apabila iya maka proses akan gagal. Sedangkan apabila r tidak sama dengan NULL maka akan dibuat node baru. Akan dilakukan looping dimana I akan diinisiasi sama dengan 0 dan apabila nilai i kurang dari numberOfPairs p maka nilai paired.paired[i] r sama dengan milik p. Lalu setelahnya nilai paired.paired[i] r akan dibuat menjadi nilai paired.paired[i] p dikali 2.

Pada looping for selanjutnya akan dilakukan inisiasi dimana j sama dengan 0 dan apabila nilai j kurang dari numberOfPairs pada q maka nilai paired.paired[i++] r sama dengan milik q. Setelah keluar dari kedua looping tersebut nilai akan diproses menjadi hasPaired r sama dengan 0, next r sama dengan NULL, group r sama dengan group p dan masuk ke fungsi binaryFill(p, q, r).

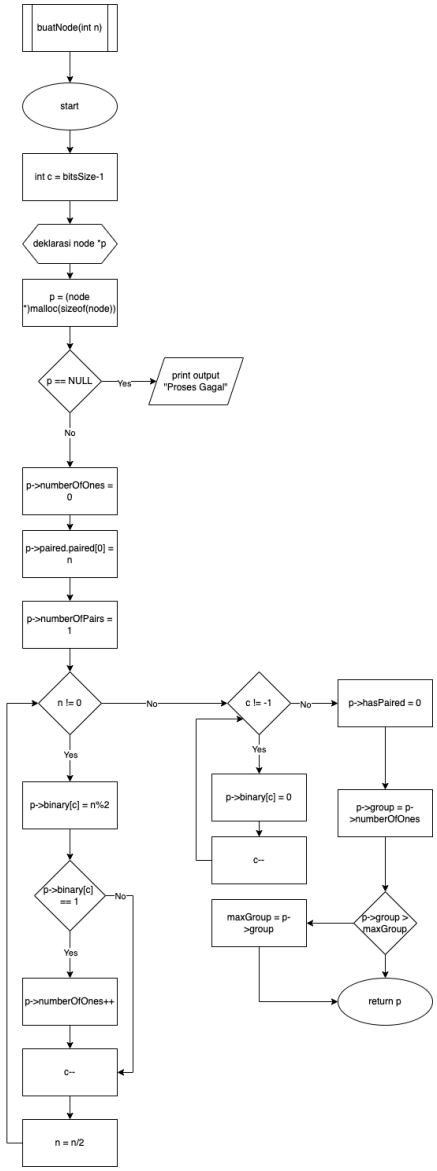
6. Fungsi binaryFill



Gambar 6. Flowchart Fungsi binaryFill

Fungsi untuk mengisi r dengan nilai biner menggunakan parameter p dan q. Fungsi akan memproses nilai c sama dengan nilai variabel - 1. Kemudian fungsi akan melakukan looping while dimana apabila nilai binary pada p sama dengan nilai array binary pada q maka nilai binary r akan sama dengan nilai binary p. Sedangkan apabila nilai binary pada p tidak sama dengan nilai array binary pada q maka nilai binary r akan sama dengan -1. Kemudian akan dilakukan pengurangan pada nilai c untuk melakukan looping while.

7. Fungsi buatNodePair

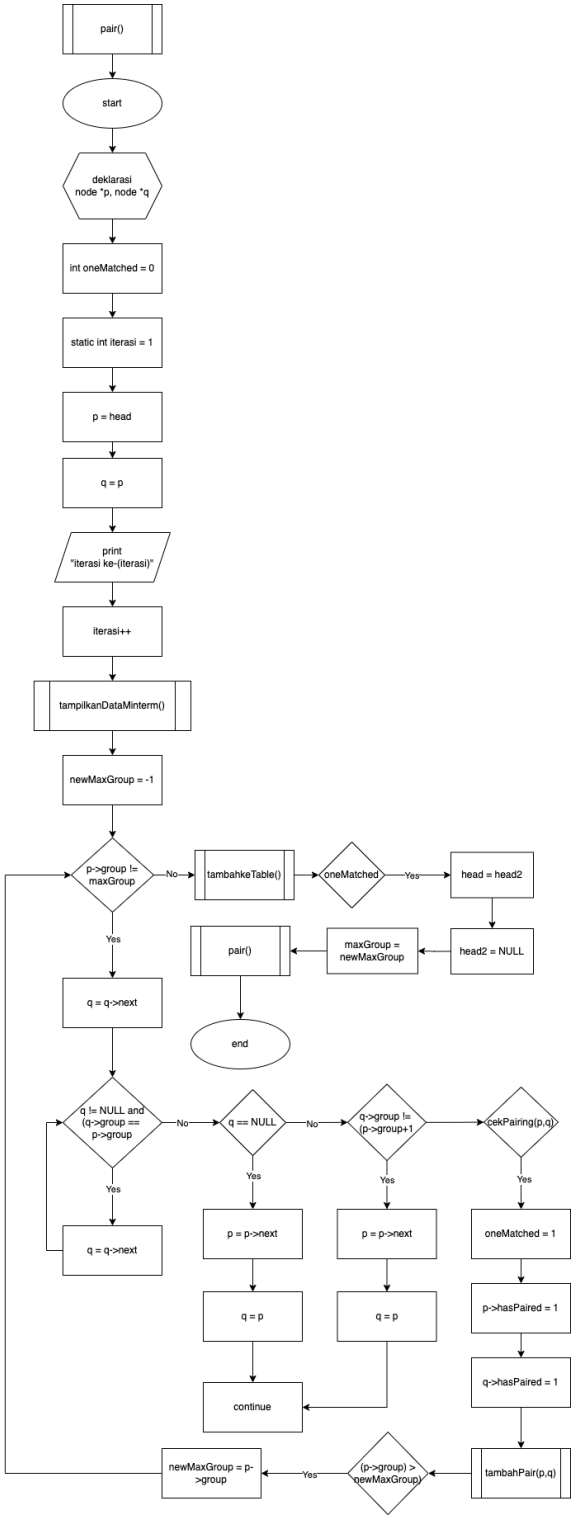


Gambar 7. Flowchart Fungsi buatNodePair

Fungsi untuk membuat node untuk menyimpan data dari minterm. Pertama akan dilakukan alokasi memory pada nilai p. Lalu apabila nilai p sama dengan NULL maka proses pada fungsi buatNode gagal. Sedangkan apabila nilai p tidak sama dengan NULL maka akan dilanjutkan prosesnya. Akan diinisiasi nilai numberOfOnes pada p sama dengan 0, paired.paired[0] p sama dengan nilai n, numberOfPairs pada p yaitu 1. Lalu di lakukan looping while dimana apabila nilai n tidak sama dengan 0 akan diinisiasi nilai binary p sama dengan nilai n dibagi 2 dimana apabila nilai binary sama dengan 1 maka nilai numberOfOnes akan bertambah. Lalu akan dikurang nilai c dan nilai n akan dibagi dua lagi untuk melakukan looping while. Selanjutnya dilakukan kembali looping while dengan parameter apabila nilai c tidak sama dengan -1. Pada looping while kali ini akan diinisiasi nilai binary p sama dengan 0 dan dilakukan pengurangan c terus menerus selama looping berlangsung. Setelahnya akan dibuat nilai numberOfOnes sama dengan

nilai group dimana apabila nilai group lebih besar dari nilai maxGroup maka nilai maxGroup akan sama dengan nilai group.

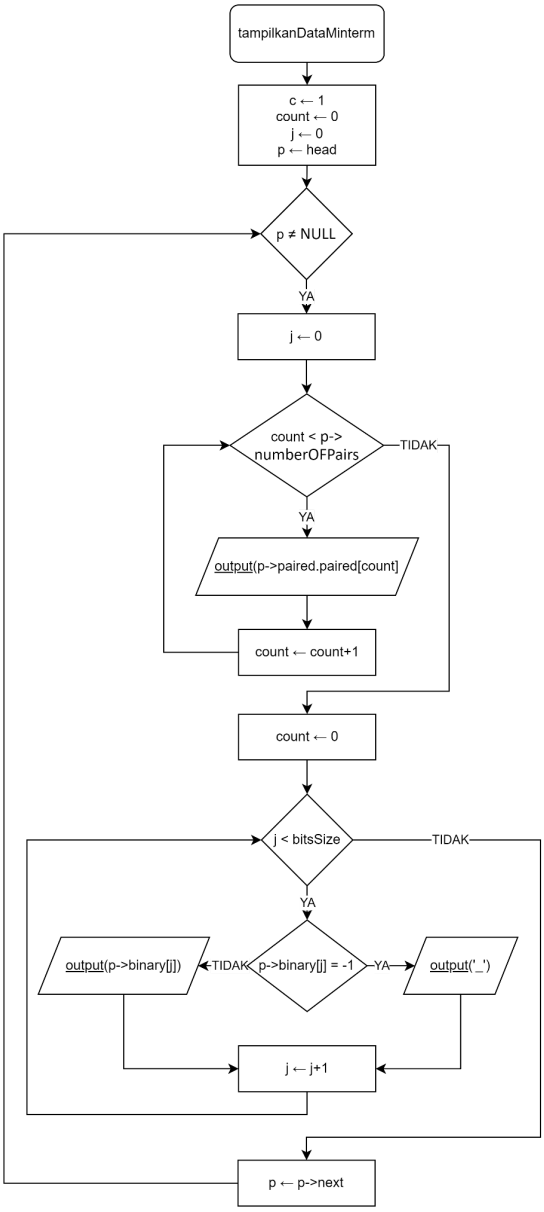
8. Fungsi pair



Gambar 8. Flowchart Fungsi pair

Fungsi pair berguna untuk membandingkan pair. Pertama akan ditampilkan iterasi dan data minterm. Kemudian akan dicek dan dibandingkan nilai p dan q apabila nilai grup p tidak sama dengan maxGroup. Apabila nilai q sama dengan 0 maka nilai q akan sama dengan p, lalu apabila nilai grup q tidak sama dengan nilai grup+1 q, nilai q juga akan sama dengan p, sedangkan apabila pada fungsi cekPairing diketahui nilai p dan q hanya berbeda 1 angka maka akan masuk ke fungsi tambahPair.

9. Fungsi tampilkanDataMinterm

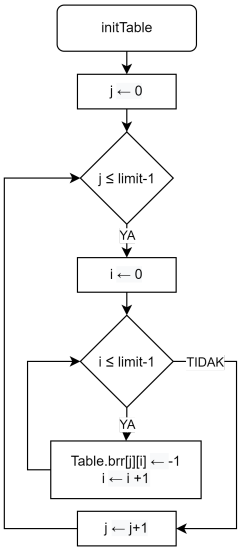


Gambar 9. Flowchart Fungsi tampilkanDataMinterm

Fungsi ini digunakan untuk menampilkan minterms dan datanya. Pertama-tama fungsi menginisiasi p sebagai head, count sebagai 0, c sebagai 1 dan j sebagai 0. Kemudian, akan dilakukan looping selama p tidak sama dengan NULL. Fungsi kemudian akan

menampilkan paired minterm dari data p->paired.paired[count], dimulai dari count sama dengan 0 hingga count sama dengan p->numberOfPairs-1. Kemudian, fungsi akan menampilkan biner dari minterm tersebut dari data p->binary dengan variabel j sebagai letak urutan bit. Jika p->binary[j] menunjukkan nilai -1, akan ditampilkan simbol '_', buka angka 1 atau 0. Simbol ini menunjukkan letak bit yang berbeda nilainya antara paired minterm. Sedangkan bit lainnya akan ditampilkan sebagai angka 0 atau 1 seperti normalnya. Fungsi kemudian akan menampilkan data minterm yang tersimpan pada nodus p berikutnya.

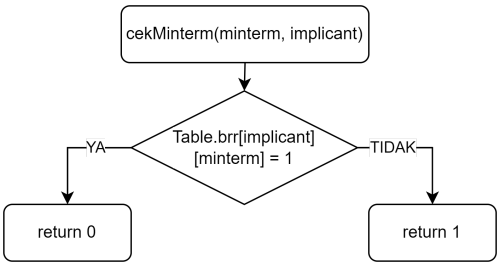
10. Fungsi initTable



Gambar 10. Flowchart Fungsi initTable

Fungsi ini digunakan untuk menginisiasi tabel prime impicants. Fungsi ini akan membuat tabel dengan ukuran limit x limit, dengan limit adalah angka terbesar pada minterm. Kemudian seluruh elemen pada tabel akan diinisiasi dengan angka -1.

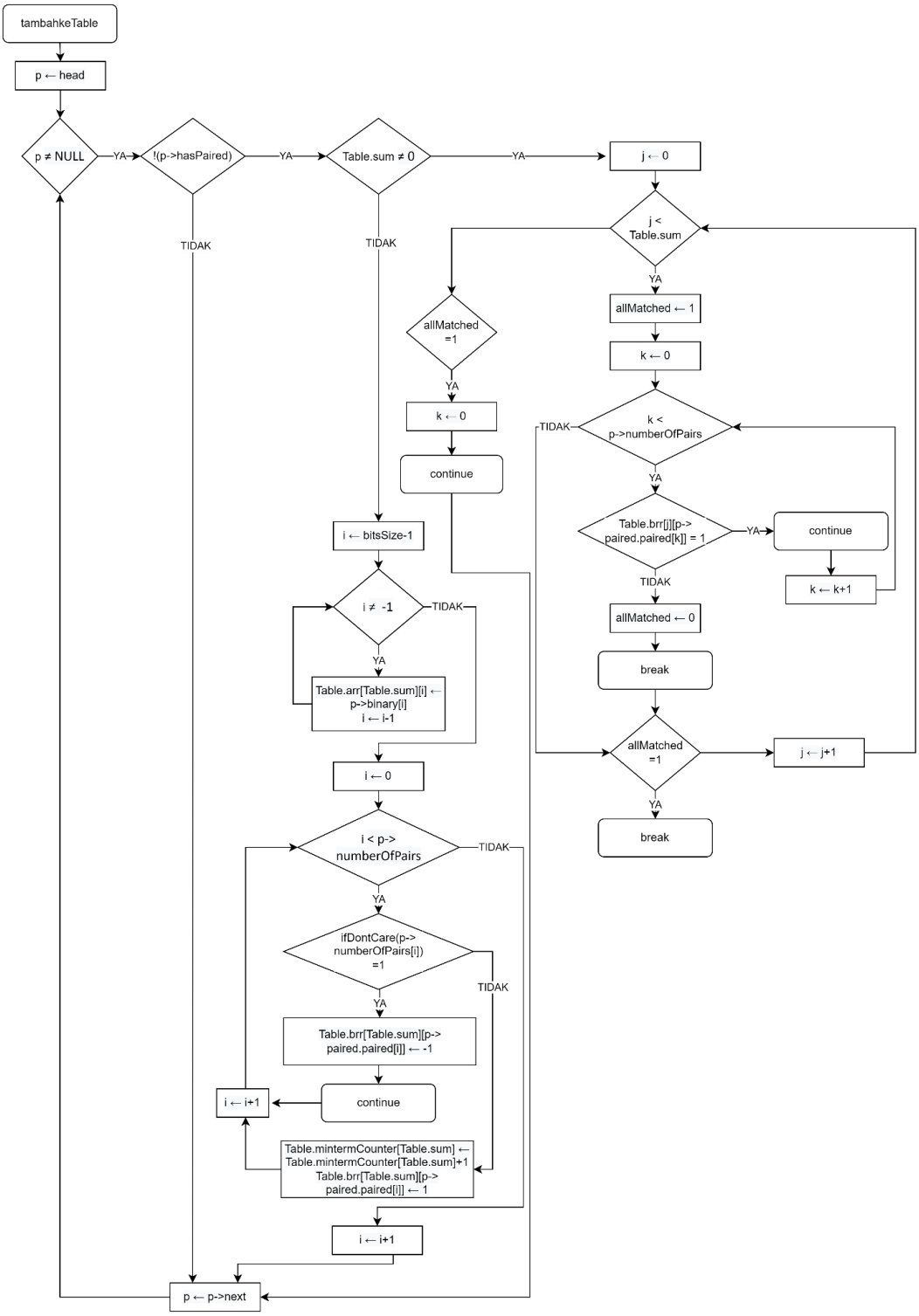
11. Fungsi cekMinterm



Gambar 11. Flowchart Fungsi cekMinterm

Fungsi ini digunakan untuk mengecek apakah minterm ada pada tabel. Fungsi akan mengecek elemen dalam tabel brr baris ke-implicant dan kolom ke-minterm apakah nilai elemen sama dengan 1. Jika iya, maka fungsi akan mengembalikan angka 1, jika tidak maka dikembalikan angka 0.

12. Fungsi TambahkeTable

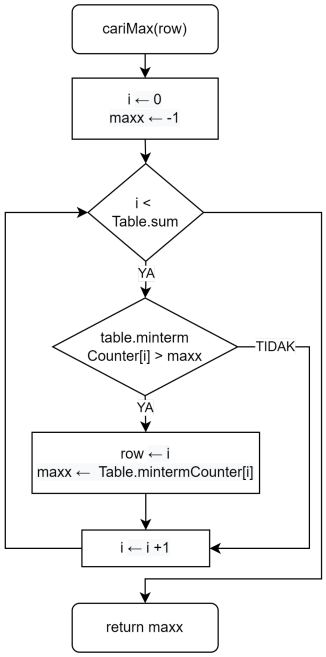


Gambar 12. Flowchart Fungsi TambahkeTable

Fungsi ini digunakan untuk menambahkan prime implicant ke dalam tabel. Fungsi akan menginisiasi variabel p sebagai node head, dan allmatched. Kemudian dilakukan loop

selama p tidak sama dengan NULL. Fungsi akan mengecek apakah p->hasPaired salah. Jika iya, maka akan dicek juga apakah Table.sum tidak sama dengan 0. Jika iya, nilai variabel allMatched diubah menjadi 1 dan dilakukan pengulangan pada tabel brr untuk jumlah baris sebanyak Table.sum. Akan dicek, apakah tabel brr untuk elemen ke p->paired.paired[k] sama dengan 1. Jika iya, looping akan dilanjutkan untuk node p selanjutnya. Nilai k berawal hari 0 hingga p->numberOfPairs-1. Jika tidak, nilai allMatched diubah menjadi 0 dan loop dihentikan. Setelah dicek tabel brr, maka allMatched akan dicek apakah nilainya sama dengan 1. Jika iya, maka loop akan dihentikan. Setiap pengecekan baris tabel brr akan dicek juga apakah allMatched bernilai 1. Jika iya, makan akan dilakukan looping untuk node p berikutnya. Kemudian dilakukan looping sebanyak bitsSize. Tabel.arr[Table.sum][i] diubah nilainya menjadi p->binary[i], dengan i bernilai dari bitsSize-1 hingga 0. Kemudian, dilakukan pengecekan don't care. Jika p->paired.paired[i] adalah don't care, maka nilai Table.brr[Table.sum][p->paired.paired[i]] menjadi -1. Kemudian looping dilanjutkan untuk node p berikutnya. Nilai i berkisar dari 0 hingga p->numberOfPairs-1. Nilai Table.mintermCounter[Table.sum] akan bertambah 1 dan Table.brr[Table.sum][p->paired.paired[i]] = 1. Kemudian, nilai Table.sum bertambah 1. Looping lalu dilanjut untuk node p berikutnya.

13. Fungsi cariMax

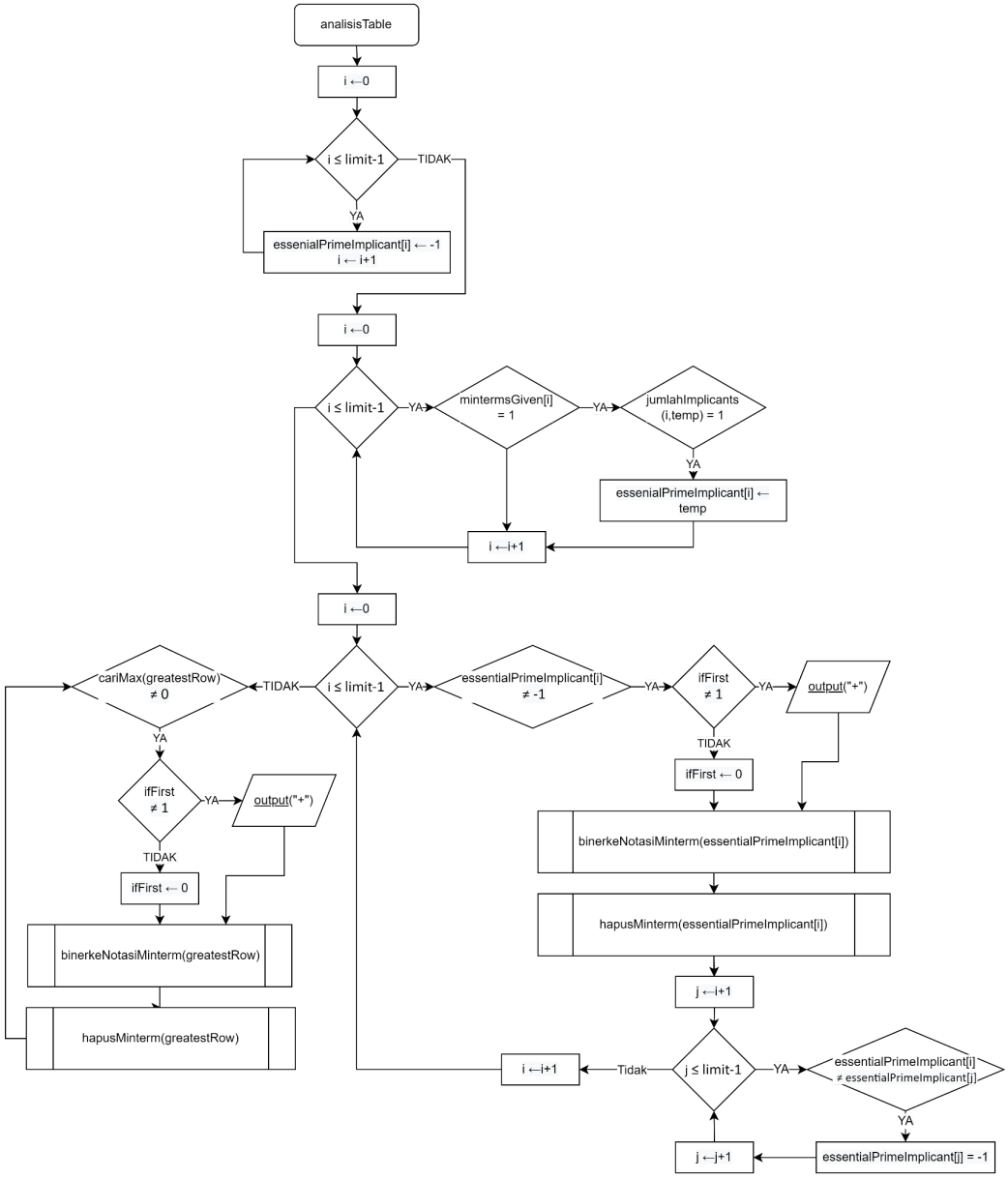


Gambar 13. Flowchart Fungsi cariMax

Fungsi ini digunakan untuk mencari prime implicant dengan unused minterm terbanyak. Fungsi akan menginisiasi nilai maxx dengan -1. Kemudian, nilai maxx ini akan dibandingkan dengan jumlah unused minterm untuk setiap prime implicant. Jika nilai unused minterm suatu prime implicant lebih besar daripada nilai maxx, maka nilai maxx akan diubah menjadi jumlah unused minterm dari prime implicant tersebut. Nilai maxx

akan menyimpan unused minterm terbanyak dan row akan menyimpan prime implicant yang bersangkutan. Fungsi akan mengembalikan nilai maxx.

14. Fungsi analisisTable

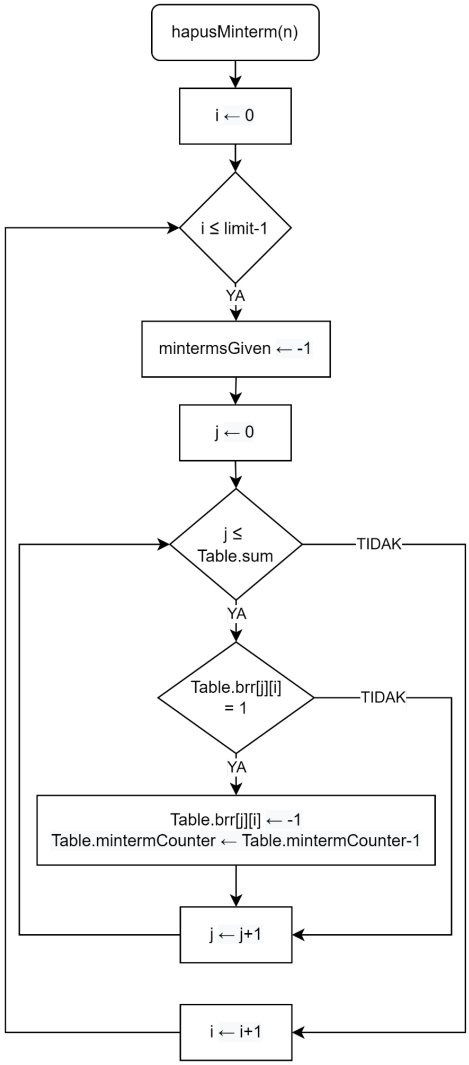


Gambar 14. Flowchart Fungsi analisisTable

Fungsi ini digunakan untuk menentukan essential prime implicant dan menampilkan hasilnya. Pertama-tama, fungsi akan membuat array essentialPrimeImplicant yang elemennya diinisiasi dengan angka -1. Lalu akan dilakukan pengulangan dengan variabel i berkisar dari 0 hingga limit-1. Jika mintermGiven[i] sama dengan 1, maka akan dicek lagi apakah jumlahImplicants(i,temp) sama dengan 1. Jika iya, nilai essentialPrimeImplicant untuk elemen ke-i diubah menjadi nilai temp. Lalu akan dilakukan pengulangan lagi dengan variabel i berkisar dari 0 hingga limit-1. Jika

essentialPrimeImplicant[i] tidak sama dengan -1, maka akan dicek apakah ifFirst tidak sama dengan 1 juga. Jika iya, maka akan diprint simbol '+' pada layar. Jika tidak, maka nilai ifFirst diubah menjadi 0. Kemudian dilakukan pemanggilan fungsi binerkeNotasiMinterm(essentialPrimeImplicant[i]) dan hapusMinterm(essentialPrimeImplicant[i]). Dilakukan lagi pengulangan untuk nilai j sama dengan i+1 hingga limit-1. Akan dicek apakah essentialPrimeImplicant[j] sama dengan essentialPrimeImplicant[i]. Jika iya, maka essentiapPrimeImplicant[j] nilainya menjadi 0. Setelah pengulangan tersebut maka nilai essentialPrimeImplicant [i] diubah menjadi -1. Setelah variabel i mencapai nilai limit-1, maka pengulangan selesai. Setalah itu, dilakukan pengulangan yang berbeda dengan syarat cariMax(&greatestRow) tidak sama dengan 0. Akan dicek apakah ifFirst tidak sama dengan 1. Jika iya, maka akan diprint simbol '+'. Jika tidak, maka variabel ifFirst menjadi 0. Kemudian dilakukan pemanggilan fungsi binerkeNotasiMinterm(greatestRow) dan hapusMinterm(greatestRow).

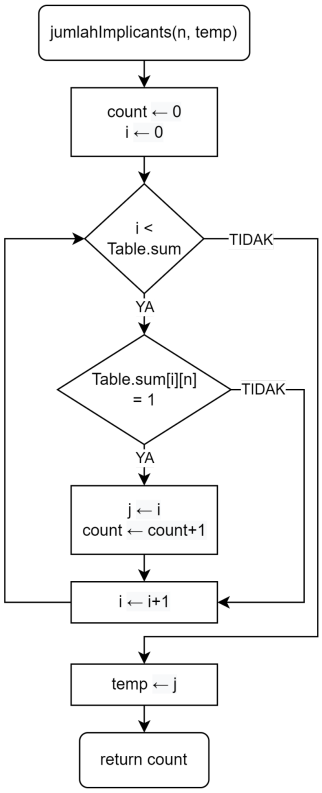
15. Fungsi hapusMinterm



Gambar 15. Flowchart Fungsi hapusMinterm

Fungsi ini digunakan untuk menghapus minterm dari tabel. Fungsi pertama-tama akan mencari baris dari tabel yang akan dihapus (baris ke-n). Jika sudah sampai ke baris tersebut, nilai mintermGiven untuk baris tersebut akan diubah nilainya menjadi -1. Kemudian, pada baris tersebut akan dicek tiap-tiap elemennya. Jika ada elemen dari baris tersebut yang bernilai 1, maka nilai elemen tersebut akan diubah menjadi -1 dan nilai minCounter yang menyimpan jumlah minterm dari baris ke-n akan berkurang 1. Hal ini dilakukan sehingga minCounter bernilai 0 dan semua elemen dari baris ke-n tidak ada yang bernilai 1.

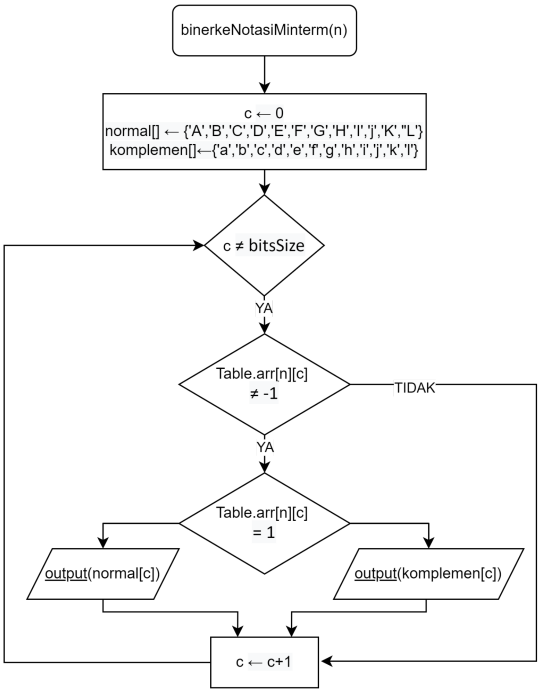
16. Fungsi jumlahImplicants



Gambar 16. Flowchart Fungsi jumlahImplicants

Fungsi ini digunakan untuk mengembalikan jumlah minterm tertentu ada pada berapa implicant. Fungsi ini menerima parameter n yang merupakan minterm yang ingin dicari jumlahnya dan temp. Fungsi menginisiasi variabel count untuk menghitung, i, dan j. Fungsi akan mengecek tiap-tiap baris tabel brr, apakah ada baris i yang elemen dengan ke-n bernilai 1. Jika ada, maka nilai count bertambah 1 dan nilai variabel j menjadi nilai i. Jika semua baris selesai dicek, nilai temp akan diubah menjadi nilai j dan fungsi akan mengembalikan nilai count.

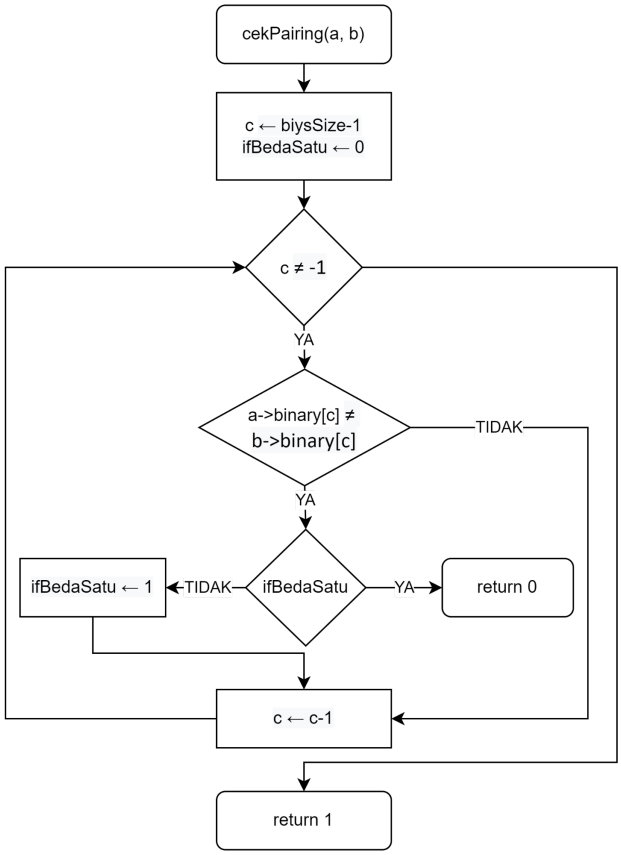
17. Fungsi binerkeNotasiMinterm



Gambar 17. Flowchart Fungsi binerkeNotasiMinterm

Fungsi ini digunakan untuk mengkonversi bilangan biner ke dalam variabel dan menampilkannya. Pertama-tama fungsi akan mendeklarasi array normal dengan alfabet kapital dari A hingga L, array komplemen dengan huruf kecil dari a hingga l dan variabel c yang diinisiasi dengan angka 0. Kemudian fungsi akan mengerjakan loop. Fungsi akan mengecek apakah nilai c tidak sama dengan bitsSize. Jika iya, maka akan dicek lagi apakah tabel arr baris ke-n kolom ke-c tidak bernilai -1. Jika iya, akan dicek sekali lagi, apakah tabel arr baris ke-n kolom ke-c bernilai 1. Jika iya, maka akan ditampilkan ke pada layar elemen array normal ke-c. Jika tidak maka yang ditampilkan adalah elemen dari array komplemen ke-c. Kemudian, loop dilaksanakan untuk nilai c berikutnya (lebih besar 1 dari sebelumnya).

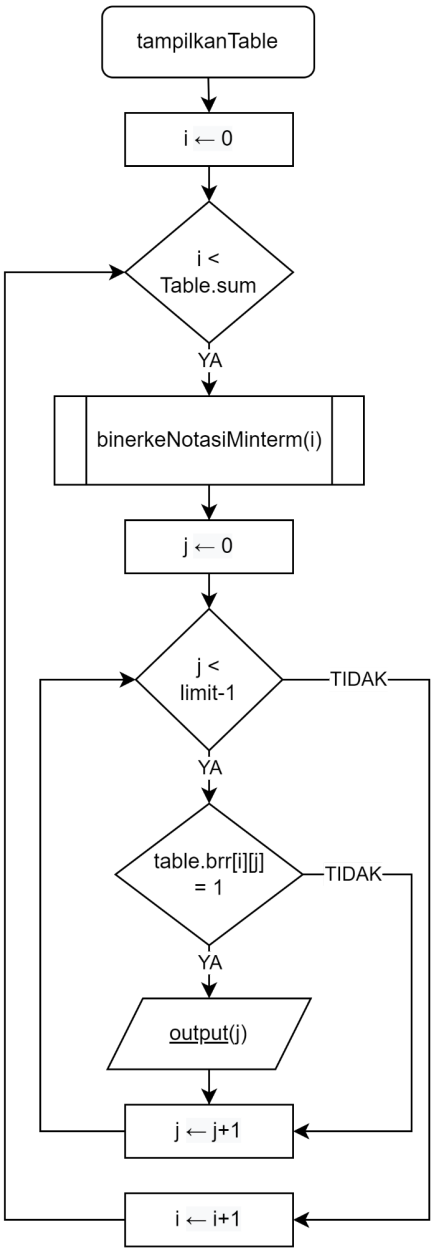
18. Fungsi cekPairing



Gambar 18. Flowchart Fungsi cekPairing

Fungsi ini digunakan untuk mengecek apakah dua bilangan biner hanya berbeda 1 angka. Fungsi akan menginisiasi c dengan nilai bitsSize-1 dan ifBedasatu dengan 0. Fungsi kemudian mengerjakan loop. Pertama-tama fungsi akan mengecek apakah nilai c tidak sama dengan -1. Jika iya, maka fungsi akan mengecek apakah bilangan pada biner a pada urutan ke-c sama dengan bilangan pada biner b urutan ke-c. Jika nilai variabel ifBedaSatu bukanlah 1, maka variabel ifBedaSatu akan diganti nilainya dengan 1. Kemudian, fungsi akan mengecek bilangan pada urutan biner berikutnya. Jika iya, maka fungsi akan mengembalikan 0. Hal ini menunjukkan bahwa perbedaan antara biner a dan b lebih dari 1. Jika benar biner a dan b hanya berbeda 1 bilangan, maka fungsi akan mengembalikan nilai 1.

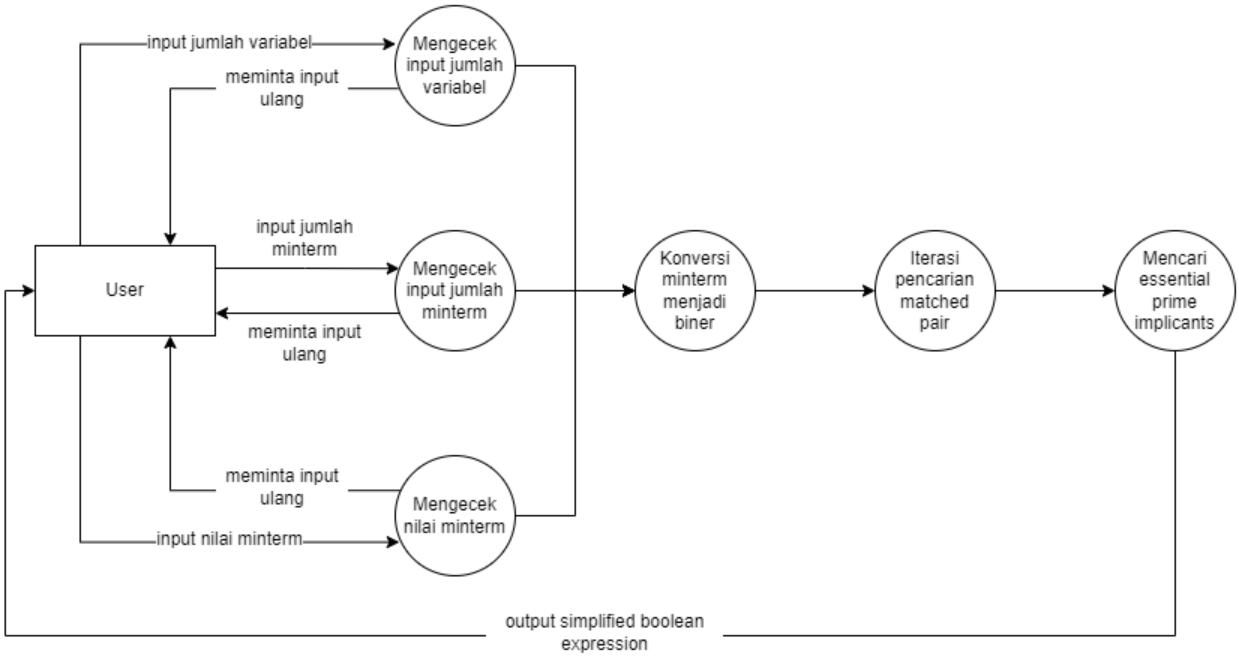
19. Fungsi tampilkanTable



Gambar 19. Flowchart Fungsi tampilkanTable

Fungsi ini digunakan untuk menampilkan tabel prime implicant. Fungsi akan menampilkan minterm untuk setiap prime implicant. Jika nilai dari elemen tabel brr, untuk baris suatu prime implicant dengan kolom ke-j, bernilai 1 maka akan ditampilkan ke layar nilai dari j. Nilai j menunjukkan minterm dari prime implicant tersebut.

Data Flow Diagram (DFD)



Gambar 20. Data Flow Diagram program

Hasil Pengujian

Seluruh test case yang dilakukan, hasilnya akan dicocokan dengan kalkulator online dari https://geeekyboy.github.io/Quine-McCluskey-Solver/#/

Test Case 1

Bagian ini akan memperlihatkan bagaimana program memverifikasi input dari pengguna.

```
PROGRAM MINIMISASI LOGIKA

Jumlah variabel (1 sampai 12): -1
Masukkan kembali jumlah variabel dalam range 1 sampai 12: 4
Jumlah minterms (0 sampai 16): -2
Masukkan kembali jumlah minterms (0 sampai 16): 18
Masukkan kembali jumlah minterms (0 sampai 16): 4
Masukkan minterms (0 sampai 15):
-5
Minterm tidak valid. Masukkan kembali dalam range 0 sampai 15
20
Minterm tidak valid. Masukkan kembali dalam range 0 sampai 15
1
4
```

Gambar 21. Output test case 1

Test Case 2

Bagian ini akan memperlihatkan output program ketika jumlah minterm = 0.

```
PROGRAM MINIMISASI LOGIKA

Jumlah variabel (1 sampai 12): 4
Jumlah minterms (0 sampai 16): 0

F = 0
Process returned 1 (0x1)   execution time : 4.643 s
Press any key to continue.
```

Gambar 22. Output test case 2

Test Case 3

Bagian ini akan memperlihat output program ketika jumlah minterm maksimal dan input minterm dari nol sampai nilai maksimalnya

```
PROGRAM MINIMISASI LOGIKA

Jumlah variabel (1 sampai 12): 2
Jumlah minterms (0 sampai 4): 4
Masukkan minterms (0 sampai 3):
0
1
2
3

F = 1
Process returned 1 (0x1)   execution time : 4.843 s
Press any key to continue.
```

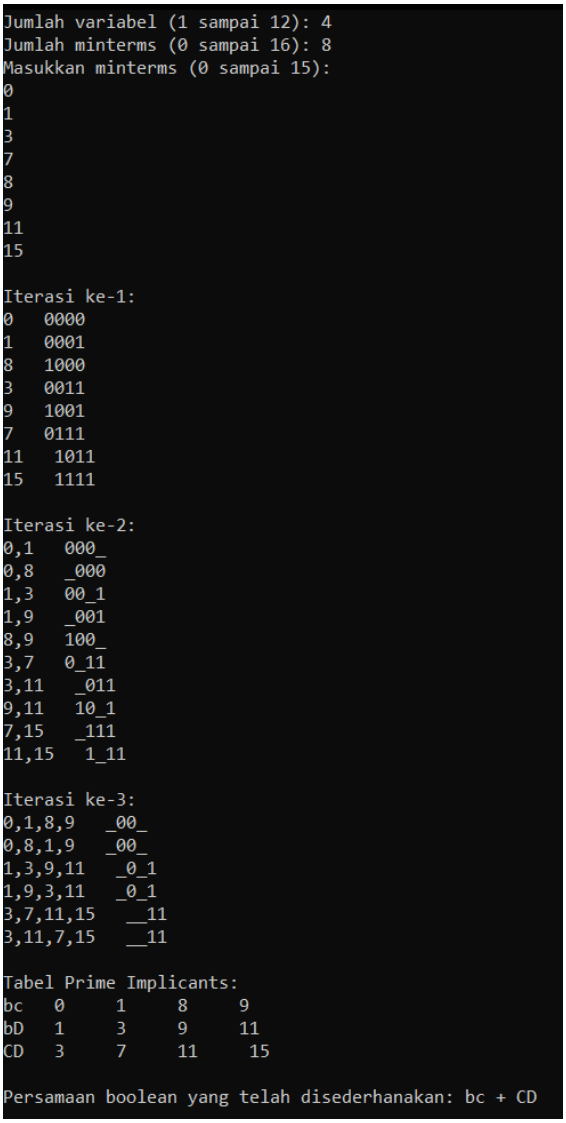
Gambar 23. Output test case 3

Test Case 4

Bagian ini akan memverifikasi output program dengan hasil dari kalkulator online.

Jumlah variabel: 4, jumlah minterm: 8, nilai minterms: 0, 1, 3, 7, 8, 9, 11,15

Hasil kalkulator online: bc + CD



Gambar 24. Output test case 4

Test Case 5

Bagian ini akan memverifikasi output program dengan hasil dari kalkulator online.

Jumlah variabel: 8, jumlah minterm: 6, nilai minterms: 2, 24, 60, 124, 200,254

Hasil kalkulator online: abcdefGh + abcDEfgh + ABcdEfgh + ABCDEFGh + aCDEFgh

```
PROGRAM MINIMISASI LOGIKA

Jumlah variabel (1 sampai 12): 8
Jumlah minterms (0 sampai 256): 6
Masukkan minterms (0 sampai 255):
2
24
60
124
200
254

Iterasi ke-1:
2   00000010
24   00011000
200   11001000
60   00111100
124   01111100
254   11111110

Iterasi ke-2:
60,124   0_111100

Tabel Prime Implicants:
abcdefGh   2
abcDEfgh   24
ABcdEfgh   200
ABCDEFGh   254
aCDEFgh   60     124

Persamaan boolean yang telah disederhanakan: abcdefGh + abcDEfgh + aCDEFgh + ABcdEfgh + ABCDEFGh
```

Gambar 25. Output test case 5

Test Case 6

Bagian ini akan memverifikasi output program dengan hasil dari kalkulator online.

Jumlah variabel: 12, jumlah minterm: 4, nilai minterms: 1, 4000, 4050, 4095

Hasil kalkulator online: abcdefghijkL + ABCDEfGhijkl + ABCDEFgHijKl + ABCDEFGHIJKL

```
PROGRAM MINIMISASI LOGIKA

Jumlah variabel (1 sampai 12): 12
Jumlah minterms (0 sampai 4096): 4
Masukkan minterms (0 sampai 4095):
1
4000
4050
4095

Iterasi ke-1:
1   000000000001
4000   111110100000
4050   111111010010
4095   111111111111

Tabel Prime Implicants:
abcdefghijkL   1
ABCDEfGhijkl   4000
ABCDEFgHijKl   4050
ABCDEFGHIJKL   4095

Persamaan boolean yang telah disederhanakan: abcdefghijkL + ABCDEfGhijkl + ABCDEFgHijKl + ABCDEFGHIJKL
```

Gambar 26. Output test case 6

Analisis

Dari enam test case yang telah dilakukan, dapat dilihat bahwa hasil yang diperoleh telah sesuai dengan output yang diharapkan. Oleh karena itu, program ini dapat dikatakan berhasil memenuhi spesifikasi program dan dapat digunakan untuk menyederhanakan fungsi boolean menggunakan metode Tabular. Selain itu, dari iterasi yang ditampilkan pada output program, pengguna dapat dengan mudah mengetahui *step-by-step* proses minimisasi logika dengan menggunakan metode Tabular dan *prime implicants* dari tabel *prime implicants* yang ditampilkan, sehingga pengguna tidak hanya mendapatkan persamaan boolean yang telah disederhanakan (*essential prime implicants*) nya saja.

Kesimpulan dan Lesson Learned

Metode Tabular merupakan salah satu metode yang digunakan untuk menyederhanakan suatu fungsi boolean. Dari ketiga metode penyederhanaan fungsi boolean, metode ini dapat digunakan untuk fungsi dengan jumlah variabel yang cukup banyak dibanding dengan metode lainnya. Metode Tabular dapat diimplementasikan dalam pemrograman bahasa C. Program yang telah dibuat menerima input berupa jumlah variabel, jumlah minterm dan minterm sedangkan untuk output yaitu berupa tabel *prime implicants* dan fungsi boolean yang telah disederhanakan. Jumlah variabel untuk program ini sendiri sudah dibatasi hanya hingga 12 variabel. Dari pengujian yang telah dilakukan, output yang didapatkan sudah sesuai dengan output yang diharapkan. Salah satu kekurangan dari program ini adalah program menerima input berupa minterm. Sehingga, dari fungsi boolean yang akan disederhanakan, perlu dikonversi secara manual terlebih dahulu ke dalam bentuk binernya.

Pembagian Tugas dalam Kelompok

Fadiah Mumtaz Andevi (18320009)
- Pembuatan Source Code
- Deskripsi Simulasi
- Flowchart
- Kesimpulan dan Lesson Learned

Tanya Nuhaisy Wulandari (18320017)
- Pembuatan Source Code
- Deskripsi Simulasi
- Flowchart
- Kesimpulan dan Lesson Learned

Eunike Kristianti (18320019)
- Pembuatan Source Code
- Deskripsi Simulasi
- Data Flow Diagram (DFD)
- PPT

Daftar Referensi

[1] https://www.geeksforgeeks.org/minimization-of-boolean-functions/, 24 April 2022, 15.30.

[2] http://freesourcecode.net/cprojects/102643/sourcecode/McQuicksy.c#.YlGuFchBw2w, 23 April 2022, 20.00.