

Development of Applications for Simplification of Boolean Functions using Quine-McCluskey Method

Pengembangan Aplikasi untuk Penyederhanaan Fungsi Boolean menggunakan Metode Quine-McCluskey

Eko Dwi Nugroho

Teknik Informatika, Institut Teknologi Sumatera, Indonesia

eko.nugroho@if.itera.ac.id

Informasi Artikel

Received: 21 January 2020

Revised: 31 March 2020

Accepted: 27 January 2021

Published: 28 February 2021

Abstract

Purpose: This research makes an application to simplify the Boolean function using Quine-McCluskey, because length of the Boolean function complicates the digital circuit, so that it can be simplified by finding other functions that are equivalent and more efficient, making digital circuits easier, and less cost.

Design/methodology/approach: The canonical form is Sum-of-Product/Product-of-Sum and is in the form of a file, while the output is in the form of a raw and in the form of a file. Applications can receive the same minterm/maksterm input and do not have to be sequential. The method has been applied by Idempoten, Petrick, Selection Sort, and classification, so that simplification is maximized.

Findings/result: As a result, the application can simplify more optimally than previous studies, can receive the same minterm/maksterm input, Product-of-Sum canonical form, and has been verified by simplifying and calculating manually.

Originality/value/state of the art: Research that applies the petrick method to applications combined with being able to receive the same minterm/maksterm input has never been done before. The calculation is only up to the intermediate stage of the Quine-McCluskey method or has not been able to receive the same minterm/maksterm input.

Abstrak

Tujuan: Penelitian ini membuat aplikasi penyederhanaan fungsi Boolean menggunakan metode Quine-McCluskey untuk menyederhanakan fungsi Boolean, dikarenakan fungsi Boolean terlalu panjang, sehingga rumit untuk membuat rangkaian digital. Fungsi tersebut dapat disederhanakan dengan mencari fungsi lain yang ekuivalen

Keywords: boolean algebra; petrick; pos; quine-mccluskey method; sop
Kata kunci: aljabar boolean; metode quine-mccluskey; metode petrick; pos; sop

dan lebih efisien, sehingga lebih mudah untuk membuat rangkaian digital dan biaya yang dikeluarkan lebih sedikit.

Perancangan/metode/pendekatan: Masukan fungsi Boolean berbentuk kanonik berupa Sum-of-Product atau Product-of-Sum, sedangkan keluarannya berbentuk bakunya. Selain itu, masukan berupa file dan keluaran berupa file. Aplikasi dapat menerima masukan minterm/maksterm yang sama. Pengguna hanya memasukan banyak peubah dan minterm/maksterm ke file. Pada metode tersebut sudah diterapkan hukum Idempoten, metode Petrick, metode Selection Sort, dan klasifikasi, sehingga penyederhanaan fungsi Boolean lebih maksimal.

Hasil: Hasil penelitian ini, aplikasi dapat menyederhanakan fungsi Boolean lebih maksimal dari penelitian terdahulu. Hal tersebut dilihat dari bentuk baku, banyak minterm dan peubah lebih sedikit dibandingkan banyak minterm dan peubah penelitian terdahulu. Selain itu, aplikasi tersebut dapat menerima masukan minterm/maksterm yang sama, serta dapat menerima masukan fungsi Boolean bentuk kanonik Product-of-Sum. Hasil tersebut sudah diverifikasi dengan menyederhanakannya secara manual, serta sudah dilakukan perhitungan manual.

Keaslian/ state of the art: Penelitian yang menerapkan metode petrick pada aplikasi yang dikombinasikan dengan dapat menerima masukan minterm/maksterm yang sama belum pernah dilakukan sebelumnya. Perhitungan hanya sampai tahap menengah dari metode Quine-McCluskey atau belum dapat menerima masukan minterm/maksterm yang sama.

1. Pendahuluan

Rangkaian digital merupakan salah satu representasi dari Aljabar Boolean. Fungsi Boolean adalah pemetaan dari B^n ke B melalui ekspresi Boolean, yang dalam hal ini B^n adalah himpunan yang beranggotakan pasangan terurut ganda- n (ordered n -tuple) di dalam daerah asal B [1]. Terkadang fungsi Boolean terlalu panjang yang mengandung operasi-operasi yang tidak perlu, literatur atau suku-suku yang berlebihan [2], sehingga rumit untuk membuat rangkaian digital. Fungsi tersebut dapat disederhanakan dengan cara mencari fungsi lain yang ekuivalen, tetapi dengan literal atau suku-suku yang lebih sedikit [2], sehingga lebih mudah untuk membuat rangkaian digital dan biaya yang dikeluarkan lebih sedikit. Penyederhanaan fungsi Boolean dapat dilakukan dengan tiga metode, yaitu Aljabar, Peta Karnaugh, dan Metode Quine-McCluskey. Aplikasi penyederhanaan fungsi Boolean sudah ada pada penelitian terdahulu [3] dengan menggunakan metode Quine-McCluskey. Metode Quine-McCluskey merupakan metode dengan cara mengubah peubah fungsi Boolean menjadi bentuk biner, kemudian

sebanyak mungkin peubah dihilangkan secara maksimal, sehingga didapat fungsi Boolean yang paling sederhana [4].

Namun, ada kekurangan pada penelitian tersebut, yaitu aplikasi penyederhanaan fungsi Boolean tidak dapat menerima masukan minterm yang sama, fungsi Boolean dalam bentuk kanonik Product-of-Sum, dan hasil penyederhanaannya kurang maksimal. Misalkan, jika masukan berupa bentuk kanonik Product-of-Sum, maka proses akan terjadi error atau fungsi Boolean yang seharusnya biasa dijadikan lima minterm, namun itu masih tujuh minterm. Oleh sebab itu, penelitian ini membuat aplikasi penyederhanaan fungsi Boolean dengan menggunakan metode Quine-McCluskey untuk menyederhanakan fungsi Boolean dengan perbedaan dan peningkatan, yaitu dapat menerima masukan fungsi Boolean dalam bentuk kanonik Product-of-Sum. Selain itu, aplikasi juga dapat menerima masukan minterm/maksterm yang sama. Pada aplikasi tersebut juga sudah diterapkan hukum Idempoten, metode Quine-McCluskey bagian (tahap 2), metode Petrick, metode Selection Sort, dan klasifikasi yang ke semuanya tidak diterapkan pada penelitian terdahulu, sehingga penyederhanaan fungsi Boolean lebih maksimal.

Rumusan masalah pada penelitian ini, yaitu tidak praktisnya menyederhanakan fungsi Boolean atau dalam artian masih manual dengan menulis, sehingga lama dalam mencari fungsi Boolean yang sederhana, tetapi ekuivalen dan lebih efisien. Selain itu, fungsi Boolean yang dipakai hanya dalam bentuk kanonik Sum-of-Product, tidak dapat untuk dalam bentuk kanonik yang Product-of-Sum. Serta, adanya minterm/maksterm yang sama akibat data tertalu banyak yang mengakibatkan termasuk dua kali. Oleh sebab itu, tujuan pada penelitian ini, yaitu membuat aplikasi penyederhanaan fungsi Boolean yang dapat menerima masukan fungsi Boolean dalam bentuk kanonik Product-of-Sum, serta dapat menerima masukan minterm/maksterm yang sama, dan hasilnya lebih maksimal.

2. Metode/Perancangan

2.1. Landasan Teori

2.1.1. Aljabar Boolean

Aljabar Boolean merupakan aljabar yang dapat digunakan untuk melakukan penyederhanaan atau menguraikan rangkaian logika yang rumit dan kompleks menjadi rangkaian logika yang lebih sederhana [5]. Rangkaian logika merupakan gabungan beberapa gerbang, untuk mempermudah penyelesaian perhitungan secara aljabar dan pengisian tabel kebenaran digunakan sifat-sifat Aljabar Boolean [6]. Aljabar Boolean menyediakan aturan dan operasi untuk bekerja dengan himpunan $\{0,1\}$, sehingga digunakan dua konstanta, yaitu logika 0 dan logika 1 [6]. Operasi dalam Aljabar Boolean yang sebagian besar digunakan, yaitu komplemen, penjumlahan, dan perkalian Boolean, sehingga Aljabar Boolean juga dapat diekspresikan dalam bentuk maksterm ataupun minterm [3]. Komplemen dari elemen, dilambangkan dengan tanda bar ($\bar{\quad}$) atau petik satu ($'$), didefinisikan dengan $\bar{0} = 1$ dan $\bar{1} = 0$. Penjumlahan Boolean, dilambangkan dengan tanda tambah (+) atau OR [7], mempunyai nilai-nilai yaitu $1 + 1 = 1$, $1 + 0 = 1$, $0 + 1 = 1$, $0 + 0 = 0$. Perkalian Boolean, dilambangkan dengan tanda titik di tengah (\cdot) atau AND [7], mempunyai nilai-nilai yaitu $1 \cdot 1 = 1$, $1 \cdot 0 = 0$, $0 \cdot 1 = 0$, $0 \cdot 0 = 0$. Aljabar Boolean mempunyai hukum-hukum persamaan [7], hukum dalam Aljabar Boolean sangat berguna untuk menyederhanakan rangkaian digital [8]. Hukum Idempoten dalam Aljabar Boolean yaitu $x + x = x$ dan $x \cdot x = x$.

2.1.2. Minterm dan Maksterm

Minterm dari peubah Boolean x_1, x_2, \dots, x_n adalah hasil perkalian Boolean $y_1 y_2 \dots y_n$, di mana $y_i = x_i$ atau $y_i = \bar{x}_i$, [8], sehingga minterm dapat diselesaikan dengan mengoperasikan fungsi Boolean dengan operasi AND, kemudian hasilnya dioperasikan dengan operasi OR [3]. Minterm harus berisi peubah yang lengkap [9]. Minterm $y_1 y_2 \dots y_n$ bernilai 1, jika dan hanya jika setiap y_i bernilai 1. Sedangkan maksterm dari peubah Boolean x_1, x_2, \dots, x_n adalah hasil penjumlahan Boolean $y_1 + y_2 + \dots + y_n$, di mana $y_i = x_i$ atau $y_i = \bar{x}_i$, [8], sehingga maksterm dapat diselesaikan dengan mengoperasikan fungsi Boolean dengan operasi OR, kemudian hasilnya dioperasikan dengan operasi AND [3]. Maksterm harus berisi peubah yang lengkap [9]. Maksterm $y_1 + y_2 + \dots + y_n$ bernilai 0, jika dan hanya jika setiap y_i bernilai 0.

2.1.3. Sum-of-Product dan Product-of-Sum

Sum-of-Product adalah penjumlahan dari minterms atau bentuk normal disjungtif dari fungsi Boolean [8] atau penjumlahan dari kali hasil yang dinotasikan: \sum , [10]. Sum-of-Product merupakan bentuk kanonik dari fungsi Boolean [9]. Contoh Sum-of-Product, yaitu $f(x, y, z) = xy\bar{z} + x\bar{y}z + \bar{x}y\bar{z}$. Sedangkan Product-of-Sum adalah perkalian dari maksterms atau bentuk normal konjungtif dari fungsi Boolean [8] atau perkalian dari hasil jumlah yang dinotasikan: \prod [10]. Sum-of-Product merupakan bentuk kanonik dari fungsi Boolean [9]. Contoh Product-of-Sum, yaitu $f(x, y, z) = (x + y + \bar{z})(x + \bar{y} + \bar{z})(\bar{x} + y + \bar{z})$.

2.1.4. Metode Quine-McCluskey dan Metode Petrick

Metode Quine-McCluskey adalah prosedur untuk menyederhanakan fungsi Boolean dalam bentuk Sum-of-Product yang dapat menjadi mekanisme untuk digunakan ketika ada lebih dari empat peubah dan tidak bergantung pada inspeksi visual untuk mengidentifikasi istilah-istilah untuk kelompok [8]. Fungsi Boolean dalam bentuk Product-of-Sum harus diubah terlebih dahulu ke dalam bentuk Sum-of-Product [3]. Penyederhanaan dengan metode Quine-McCluskey sangat sistematis dan sangat cocok diterapkan dengan komputer [3]. Metode Quine-McCluskey menggunakan urutan langkah-langkah untuk menyederhanakan fungsi Boolean dalam bentuk Sum-of-Product [8]. Ada dua tahap (bagian) di metode Quine-McCluskey, yaitu:

1. Menemukan kandidat-kandidat dari fungsi Boolean untuk penyederhanaannya atau disebut juga prime implicant
2. Memilih diantara prime implicant tersebut yang paling minimal dalam mengcover fungsi Boolean

Metode Petrick adalah metode minimisasi dua tingkat yang menyediakan cara sistematis dalam pemilihan prime implicant yang dapat memberikan semua alternatif solusi minimum [11]. Metode Petrick dapat mengatasi kekurangan dari metode Quine-McCluskey, yaitu apabila pada tahap pilihan prime implicant essential tidak dapat mengcover semua minterm. Metode Petrick menggunakan urutan langkah-langkah untuk memilih prime implicant selain prime implicant essential yang dapat mengcover semua minterm. Pada bagian pertama metode Petrick, semua prime implicant dari suatu fungsi Boolean secara sistematis dibentuk dengan menggabungkan minterm-minterm [11].

2.1.5. Metode Selection Sort

Metode selection sort adalah metode pengurutan yang membandingkan elemen yang sekarang dengan elemen berikutnya sampai ke elemen yang terakhir. Jika ditemukan elemen lain yang lebih kecil atau lebih besar dari elemen sekarang, maka dicatat posisinya dan langsung ditukar

[12]. Defenisi lainnya, metode selection sort adalah melakukan pemilihan dari suatu nilai yang terkecil dan kemudian menukarnya dengan elemen paling awal, lalu membandingkan dengan elemen yang sekarang dengan elemen berikutnya sampai dengan elemen terakhir, perbandingan dilakukan terus sampai tidak ada lagi pertukaran data [13].

2.1.6. Klasifikasi

Klasifikasi adalah penyusunan bersistem dalam kelompok atau golongan menurut kaidah atau standar yang ditetapkan [14]. Klasifikasi merupakan metode untuk menyusun data secara sistematis atau menurut beberapa aturan atau kaidah yang telah ditetapkan atau pembagian sesuatu menurut kelas-kelasnya [15].

2.2. Tahapan Penelitian

Metode penelitian yang dilakukan oleh peneliti adalah Studi Literatur. Pada penelitian ini, terdapat lema belas referensi yang terdiri dari sembilan jurnal, empat buku, satu makalah ilmiah, dan satu artikel di internet. Selain itu, pada penelitian ini, terdapat delapan belas data penelitian yang dikumpulkan dari berbagai sumber, data yang dikumpulkan adalah data-data rangkaian digital. Metode pengembangan sistem pada penelitian ini adalah Iteratif atau Inkrement dengan tahapan per perulangan yaitu analisis sistem, desain sistem, kode program, penerapan, dan pengujian sistem. Pada pengujian, penelitian ini menggunakan metode Black Box.

2.3. Rancangan Sistem

Rancangan sistem pada penelitian ini sebagai berikut. Tahap 1, sistem menerima semua masukan bentuk kanonik fungsi Boolean. Masukan berupa nilai desimal dari minterm/maksterm. Media masukan berupa file (.txt), bukan ketikan langsung. Tahap 2, sistem melakukan pengecekan terhadap semua minterm/maksterm. Jika terdapat minterm/maksterm yang sama, maka sistem melakukan proses hukum Idempoten terhadap minterm/maksterm tersebut. Tahap 3, untuk fungsi Boolean berbentuk kanonik Product-of-Sum, sistem melakukan konversi fungsi Boolean dari bentuk kanonik Product-of-Sum ke bentuk kanonik Sum-of-Product.

Tahap 4, sistem dapat menyederhanakan fungsi Boolean dengan menggunakan metode Quine-McCluskey. Tahap 5, jika terdapat minterm yang belum tercover semua oleh prime implicant essential, maka sistem melakukan proses metode Petrik untuk mendapatkan prime implicant yang mengcover minterm yang belum tercover oleh prime implicant essential. Tahap 6, sistem menampilkan hasil penyederhanaan fungsi Boolean. Media keluaran berupa file bukan tampilan langsung. Bentuk baku dari Sum-of-Product, komplemen ditandai dengan huruf non capital, sedangkan non komplemen ditandai dengan huruf kapital. Bentuk baku dari Product-of-Sum komplemen ditandai dengan huruf kapital, sedangkan non komplemen ditandai dengan huruf non kapital. Selain sistem menampilkan hasil penyederhanaan fungsi Boolean, sistem juga menampilkan fungsi Boolean awal. Media keluaran berupa file bukan tampilan langsung. Rancangan sistem pada penelitian ini dapat dilihat pada **Gambar 1**.



Gambar 1 Rancangan Sistem

Pada Gambar 1 memperlihatkan alur dari sistem, yaitu mulai dari masukan fungsi Boolean (tahap 1, sistem menerima semua masukan bentuk kanonik fungsi Boolean) sampai keluar dari fungsi Boolean (tahap 6, menampilkan hasil penyederhanaan fungsi Boolean).

3. Hasil dan Pembahasan

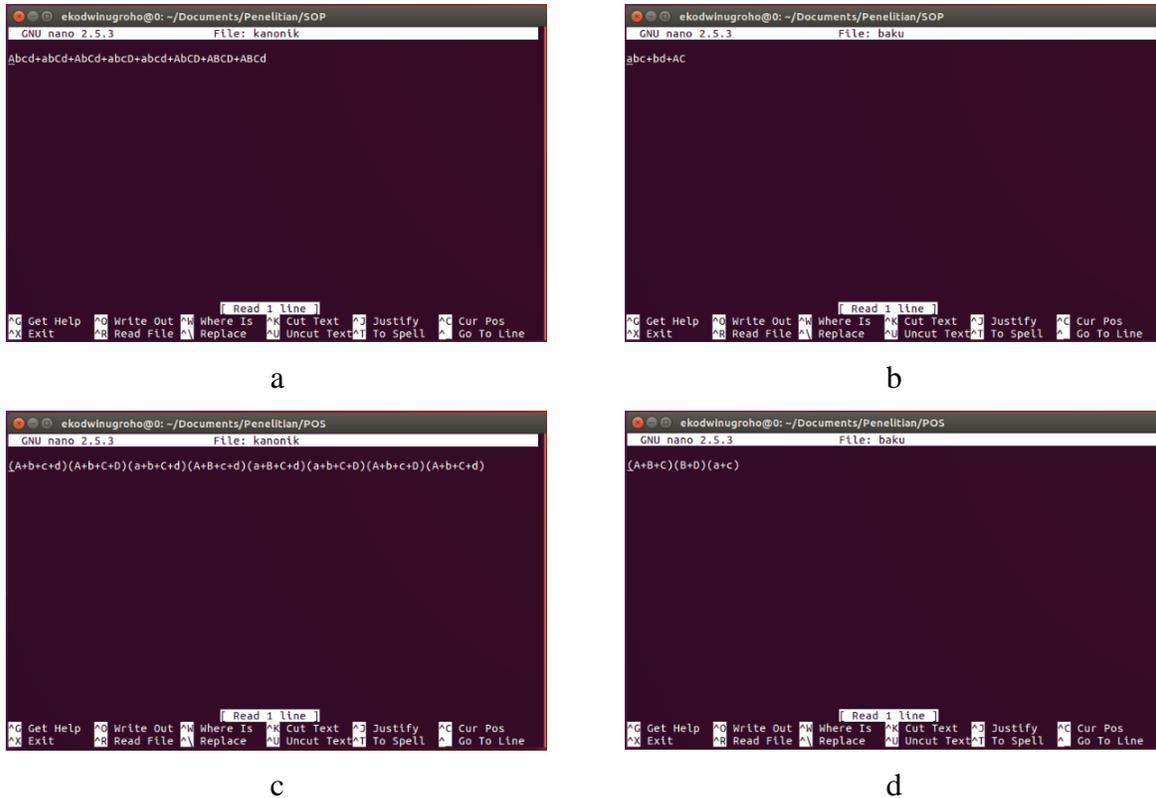
3.1. Hasil

Pengujian dari penelitian ini dilakukan sesudah selesai pembuatan kode-kode program. Pada pengujian, dilakukan tiga kali percobaan, masing-masing dalam bentuk kanonik Sum-of-Product dan Product-of-Sum. Hasil dari penelitian ini sesuai dengan yang diharapkan. Hasil dari penelitian ini sebagai berikut. Variabel non komplemen direpresentasikan dengan huruf kapital, sedangkan untuk variabel komplemen direpresentasikan dengan huruf non kapital. Hal tersebut untuk memudahkan pada komputerisasi.

1. Percobaan ke satu menggunakan data fungsi Boolean bentuk kanonik Sum-of-Product, yaitu $f(A, B, C, D) = \sum m(8, 2, 10, 1, 0, 11, 15, 0, 14)$. Hasil dari percobaan ke satu, yaitu fungsi Boolean bentuk kanonik $f(A, B, C, D) = Abcd + ABCd$ dan bentuk baku: $f(A, B, C, D) = abc + bd + AC$.
2. Percobaan ke dua menggunakan data fungsi Boolean bentuk kanonik Sum-of-Product, yaitu $f(A, B, C, D) = \sum m(6, 4, 7, 1, 15, 11, 8, 10, 9)$. Hasil dari percobaan ke dua, yaitu fungsi Boolean bentuk kanonik $f(A, B, C, D) = aBCd + AbcD$ dan bentuk baku $f(A, B, C, D) = bcd + aBd + BCD + ACD + Ab$.
3. Percobaan ke tiga menggunakan data fungsi Boolean bentuk kanonik Sum-of-Product, yaitu $f(A, B, C, D, E) = \sum m(2, 6, 8, 11, 13, 16, 19, 30, 0, 4, 7, 10, 12, 14, 18, 29)$. Hasil dari percobaan ke tiga, yaitu fungsi Boolean bentuk kanonik $f(A, B, C, D, E) = abcDe + ABCdE$ dan bentuk baku $f(A, B, C, D, E) = abCD + aBCD + BCdE + BCDE + AbcD + bce + ae$.
4. Percobaan ke empat menggunakan data fungsi Boolean bentuk kanonik Product-of-Sum, yaitu $f(A, B, C, D) = \prod M(12, 7, 4, 13, 3, 9, 12, 6, 5)$. Hasil dari percobaan ke empat, yaitu fungsi Boolean bentuk kanonik $f(A, B, C, D) = (A + b + c + d)(A + b + C + d)$ dan bentuk baku $f(A, B, C, D) = (A + B + C)(B + D)(a + c)$.
5. Percobaan ke lima menggunakan data fungsi Boolean bentuk kanonik Product-of-Sum, yaitu $f(A, B, C, D) = \prod M(14, 5, 2, 0, 12, 3, 13)$. Hasil dari percobaan ke lima, yaitu fungsi Boolean bentuk kanonik $f(A, B, C, D) = (a + b + c + D)(A + b + C + d)$ dan bentuk baku $f(A, B, C, D) = (B + C + d)(a + b + D)(a + c + d)(a + c + d)(a + B)$.
6. Percobaan ke enam menggunakan data fungsi Boolean bentuk kanonik Product-of-Sum, yaitu $f(A, B, C, D, E) = \prod M(5, 1, 31, 26, 27, 20, 9, 5, 24, 28, 23, 3, 15, 25, 22, 21, 17)$. Hasil dari percobaan ke enam, yaitu fungsi Boolean bentuk kanonik $f(A, B, C, D, E) =$

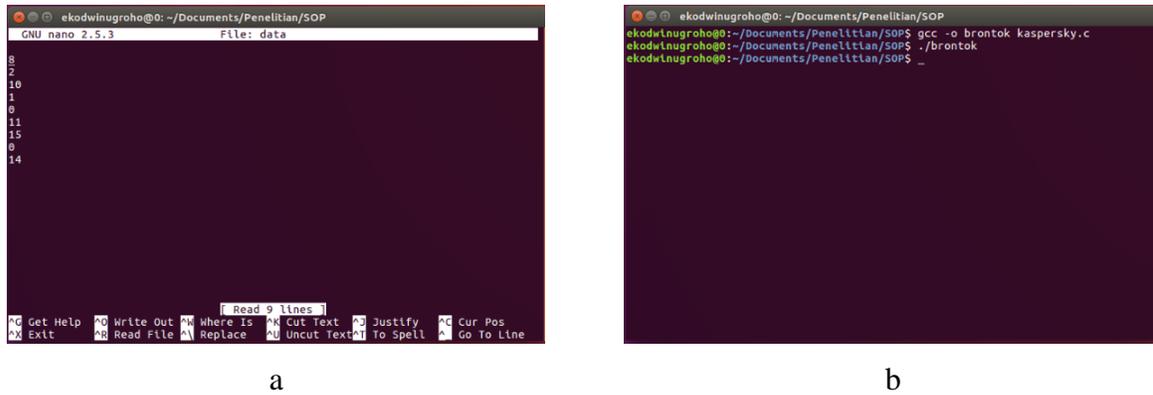
$(A + B + C + D + e)(a + B + C + d + e)$
 dan bentuk baku $f(A, B, C, D, E) = (A + B + c + d)(A + b + C + d)(b + c + D + e)(b + c + d + E)(a + B + C + d)(B + C + E)(A + E)$.

Aplikasi yang dibuat pada penelitian ini dapat dilihat pada **Gambar 2** dan **Gambar 3**.



Gambar 2 Fungsi Boolean Bentuk Kanonik dan Baku Sum-of-Product Percobaan ke Satu dan Empat

Pada Gambar 2a memperlihatkan hasil dari masukan fungsi Boolean dalam bentuk kanonik pada percobaan ke satu. Masukan fungsi Boolean dari nilai desimal diubah ke dalam bentuk fungsi (variabel) dan hasil tersebut dimasukkan ke dalam file. Dari hasil tersebut, $abcd$ adalah minterm dari nilai desimal delapan, $abCd = 2$, $AbCd = 10$, $abcD = 1$, $abcd = 0$, $AbCD = 1$, $ABCD = 15$, dan $ABCd = 14$. Terdapat dua minterm yang sama pada percobaan ke satu, yaitu 0, maka hasilnya hanya diambil satu. Sedangkan pada Gambar 2b memperlihatkan hasil dari keluaran fungsi Boolean dalam bentuk baku pada percobaan ke satu. Dari hasil tersebut, banyak minterm menjadi lebih sedikit jika dibandingkan dengan bentuk kanoniknya setelah disederhanakan dengan menggunakan metode Quine-McCluskey dan Petrick. Pada Gambar 2c dan Gambar 2d memperlihatkan hasil dari masukan fungsi Boolean dalam bentuk kanonik pada percobaan ke empat yang mana banyak maksterm menjadi lebih sedikit.



Gambar 3 Masukan di dalam File Bentuk Kanonik Sum-of-Product, Kompile, dan Menjalankan Aplikasi

Pada Gambar 3a, media masukan fungsi Boolean berupa file dan dituliskan di dalam file tersebut dalam bentuk nilai desimal dari minterm/makstern, serta susunannya ke bawah. Nilai pertama menyatakan banyaknya minterm/makstern pada fungsi Boolean, dan nilai-nilai selanjutnya menyatakan minterm/makstern perbarisnya. Lokasi file statis yaitu file diletakan di folder yang sama dengan aplikasi. Sedangkan pada Gambar 3b, aplikasi dijalankan via terminal/CMD (tidak ada perintah/sintak khusus) dan tidak dibuat GUInya. Penelitian ini hanya berfokus pada metode yang digunakannya. Pada saat menjalankan aplikasi, tidak terjadi tampilan proses apapun, hasil penyederhanaan fungsi Boolean dimasukkan ke dalam file (file yang berbeda dari file masukan) di folder yang sama dengan aplikasi.

3.2. Pembahasan

Hasil dari penelitian ini dibahas sebagai penjelasan. Pembahasan juga menjelaskan tentang hal-hal yang berkaitan dengan penelitian ini. Hasil dari penelitian ini, aplikasi dapat menyederhanakan fungsi Boolean lebih maksimal dari penelitian terdahulu. Hal tersebut dapat dilihat dari bentuk bakunya, yaitu banyak minterm dan peubah lebih sedikit dibandingkan dengan banyak minterm dan peubah pada penelitian terdahulu. Misalkan pada percobaan ke tiga di sub bab Hasil, pada penelitian terdahulu menghasilkan fungsi Boolean bentuk baku $f(A, B, C, D, E) = abCD + aBcD + aBCd + AbcD + BCdE + BCDe + AbcD + bce + ae$, sedangkan pada penelitian ini menghasilkan fungsi Boolean bentuk baku $f(A, B, C, D, E) = abCD + aBcD + BCdE + BCDe + AbcD + bce + ae$ dan jika dihitung efisiensinya menggunakan **Persamaan (1)** [3]:

$$Efisiensi = \frac{(\sum \text{minterm masukan} - \sum \text{minterm keluaran}) + (\sum \text{variabel masukan} - \sum \text{variabel keluaran})}{(\sum \text{minterm masukan} + \sum \text{variabel keluaran})} \times 100 \quad (1)$$

maka pada penelitian terdahulu menghasilkan efisiensi sebesar 62,5%, sedangkan pada penelitian ini mengasilkan efisiensi sebesar 66,667%. Nilai efisiensi didapat dari banyak minterm yang dimasukkan dikurangi banyak minterm yang dihasilkan, kemudian ditambah hasil pengurangan banyak variabel yang dimasukkan dengan banyak variabel yang dihasilkan, setelah itu dibagi dengan penjumlahan antara banyak suku yang dimasukkan dengan banyak variabel yang dimasukkan dan hasil kesemuanya dikalikan seratus persen [3].

Hal tersebut dikarenakan pada penelitian terdahulu, penggunaan metode Quine-McCluskey hanya sampai pada tahap (bagian) ke satu, yaitu pencarian prime implicant. Penyederhaan

fungsi Boolean menggunakan metode Quine-McCluskey sampai pada tahap tersebut sudah menghasilkan fungsi Boolean yang sederhana. Namun, fungsi Boolean akan lebih sederhana jika dilakukan sampai tahap (bagian) ke dua, yaitu pencarian prime implicant essential di mana dipilih prime implicant yang paling minimal dalam mengcover fungsi Boolean.

Pada penelitian ini, penggunaan metode Quine-McCluskey pada sistem yang dibuat sampai pada tahap (bagian) ke dua, bahkan dilanjutkan lagi sampai tahap menggunakan metode Petrick, sehingga hasilnya lebih sederhana lagi. Hasil penyederhanaan fungsi Boolean sebenarnya ditentukan oleh fungsi Boolean bentuk kanonik itu sendiri. Ada kondisi di mana memang fungsi Boolean bentuk kanonik tidak ada bentuk yang lebih sederhananya, sehingga ketika menggunakan metode Quine-McCluskey hasilnya pun sama seperti fungsi Boolean bentuk kanonik.

Penggunaan hukum Idempoten pada penelitian ini berfungsi untuk menangani minterm/maksterm yang sama dalam artian fungsi Boolean sudah lebih sederhana sebelum masuk ke metode Quine-McCluskey. Tidak hanya itu, pada penelitian ini, jika fungsi Boolean berbentuk kanonik Product-of-Sum, maka sistem secara otomatis mengkonversikan dahulu ke bentuk kanonik Sum-of-Product, dikarenakan metode Quine-McCluskey hanya digunakan untuk fungsi Boolean bentuk kanonik Sum-of-Product. Setelah itu, hasilnya dikonversikan kembali ke bentuk kanonik Product-of-Sum. Sedangkan penggunaan metode Selection Sort dan klasifikasi pada penelitian ini berfungsi untuk mempermudah dalam menggunakan metode Quine-McCluskey. Namun, aplikasi penyederhanaan fungsi Boolean dengan menggunakan metode Quine-McCluskey pada penelitian ini hanya dapat menampung maksimal dua puluh tujuh peubah, dikarenakan menggunakan peubah alfabet. Jadi, perbedaan yang penting dari penelitian sebelumnya, yaitu:

1. Di metode Quine-McCluskey ditambahkan tahap (bagian) ke dua
2. Ditambahkan metode Petrick setelah tahap (bagian) kedua metode Quine-McCluskey
3. Dapat menerima masukan fungsi Boolean dalam bentuk Product-of-Sum dan masukan minterm/maksterm yang sama

4. Kesimpulan dan Saran

Kesimpulan pada penelitian ini berdasarkan hasil penelitian dan pembahasan, yaitu aplikasi penyederhanaan fungsi Boolean dengan menggunakan metode Quine-McCluskey berhasil dibuat untuk menyederhanakan fungsi Boolean. Aplikasi berfungsi dengan baik dan sesuai dengan yang diharapkan (di dalamnya sudah termasuk penggunaan metode Quine-McCluskey tahap (bagian) ke dua dan metode Petrick). Aplikasi tersebut dapat menyederhanakan fungsi Boolean lebih maksimal dari penelitian terdahulu, serta dapat menerima masukan fungsi Boolean dalam bentuk kanonik Product-of-Sum dan dapat menerima masukan minterm/maksterm yang sama. Namun, pengembangan aplikasi ini masih hanya sebatas kalkulator untuk menghitung fungsi Boolean, sebagai saran untuk penelitian selanjutnya, sistem ini dapat ditanam atau disertakan di peralatan digital, mikrokontroler, dsb.

Daftar Pustaka

- [1] M. R. Harahap, "Aplikasi Pembelajaran Matematika Diskrit (Karnaugh Map) sebagai Alat Bantu Pengajaran," Manajemen Informatika, Bandung, 2012.
- [2] H. Saputra, "Penerapan Metode Quine-McCluskey untuk Menyederhanakan Fungsi Boolean," *Jurnal Teknologi dan Sistem Informasi*, vol. II, no. 1, pp. 31-41, 2015.
- [3] W. O. D. Endah and S. S. Zahroh, "Pengembangan Aplikasi Penyederhanaan Aljabar Boolean dalam Bentuk Sum-Of-Product dengan Menggunakan Metode Quine Mccluskey," *Jurnal Komputasi*, vol. I, no. 2, pp. 50-58, 2013.
- [4] A. Nasution, "Implementasi Fungsi Boolean Dengan Metode Quine-Mccluskey," *Kultura*, vol. XVII, no. 1, pp. 5786-5792, 2016.
- [5] W. Widjanarka, Teknik Digital, Jakarta: Erlangga, 2006.
- [6] Y. Indrawaty, L. Kristiana and S. Nugraha, "Aplikasi Pembelajaran Rangkaian Kombinasional dengan Menggunakan Skenario Multimedia Interaktif Model Timeline Tree," *Jurnal Informatika*, vol. III, no. 2, pp. 1-11, 2012.
- [7] A. F. Mustika, M. H. Hasyim and S. E. Unas, "Analisa Keterlambatan Proyek menggunakan Fault Tree Analysis (FTA) (Studi Kasus pada Proyek Pembangunan Gedung Program Studi Teknik Industri Tahap II Universitas Brawijaya Malang)," *Jurnal Mahasiswa Jurusan Teknik Sipil*, vol. I, no. 2, pp. 1-9, 2014.
- [8] K. H. Rosen, Discrete Mathematics and Its Applications, NewYork: McGraw-Hill, 2012.
- [9] R. Munir, Matematika Diskrit, 6th ed., Bandung: Informatika, 2016.
- [10] A. Septinez, "Aplikasi Aljabar Boolean dalam Sistem Digital," Teknik Informatika, Bandung, 2015.
- [11] N. Arslan and A. Sertbas, "An Educational Computer Tool for Simplification of Boolean Function's via Petrick's Method," *Journal of Electrical & Electronics*, vol. II, no. 2, pp. 555-561, 2002.
- [12] S. Y. Yahya, "Analisa Perbandingan Algoritma Bubble Sort dan Selection Sort Dengan Metode Perbandingan Eksponensial," *Jurnal Pelita Informatika*, vol. VI, no. 3, pp. 136-142, 2014.
- [13] P. A. Rahayuningsih, "Analisis Perbandingan Kompleksitas Algoritma Pengurutan Nilai (Sorting)," *Jurnal Evolusi*, vol. IV, no. 2, pp. 64-75, 2016.
- [14] B. P. d. P. B. K. P. d. K. Republik Indonesia, "Klasifikasi," 2016. [Online]. Available: <https://kbbi.kemdikbud.go.id/entri/klasifikasi>. [Accessed 19 Mei 2019].
- [15] I. D. Lestari, "Klasifikasi Online dan Google," *Jurnal Iqra'*, vol. X, no. 2, pp. 83-94, 2016.