Penerapan Aljabar Boolean dalam Rangkaian Digital

Eugene Yap Jin Quan - 13521074[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]*13521074@std.stei.itb.ac.id*

***Abstract*—Rangkaian digital adalah rangkaian listrik yang terdiri atas komponen dan sistem digital. Aljabar Boolean dapat digunakan dalam proses analisis rangkaian digital. Dengan prinsip aljabar Boolean, penulis mencontohkan proses perancangan sebuah rangkaian digital melalui perhitungan, perancangan rangkaian gerbang logika, dan simulasi. Berdasarkan pembahasan yang penulis lakukan, penulis menyimpulkan bahwa aljabar Boolean dapat diterapkan secara luas dalam konteks rangkaian digital dan elektronika.**

***Kata kunci*—aljabar Boolean, gerbang logika, rangkaian digital.**

I. Pendahuluan

Rangkaian digital adalah rangkaian listrik yang terdiri atas komponen dan sistem digital. Rangkaian digital menerima masukan dan mengeluarkan keluaran yang berupa sinyal digital (diskrit). Umumnya, rangkaian ini menggunakan sinyal digital biner [6].

Rangkaian digital dapat ditemukan pada berbagai perangkat elektronik. Perangkat-perangkat seperti saklar lampu rumah, sistem sensor pada sebuah kendaraan, perangkat manufaktur industri, dan perangkat komputer dapat dijelaskan menggunakan konsep rangkaian digital.

Salah satu konsep dalam matematika diskrit adalah aljabar Boolean. Aljabar Boolean adalah aljabar yang mengoperasikan nilai biner. Dalam bidang elektronika, aljabar Boolean digunakan sebagai metode untuk melakukan abstraksi terhadap sebuah rangkaian digital.

Berdasarkan uraian di atas, pertanyaan yang menjadi rumusan masalah adalah bagaimana contoh penggunaan aljabar Boolean dalam rangkaian digital? Untuk menjawab itu, penulis melakukan eksplorasi terhadap topik matematika diskrit aljabar Boolean, serta topik elektronika sederhana. Pada makalah ini, penulis mencoba menjawab masalah tersebut dengan melakukan studi literatur, penyelesaian masalah rangkaian digital sederhana, serta melakukan eksperimentasi. Eksperimentasi ini menggunakan perangkat simulasi *Digital Logic Sim* oleh Sebastian Lague [1].

II. Teori Dasar

A. Aljabar Boolean

Aljabar Boolean adalah aljabar yang dicetuskan oleh George Boole pada 1854. Aljabar Boolean dicetuskan sebagai superset dari aljabar himpunan dan aljabar logika proposisi. Aljabar Boolean menggunakan nilai biner. Nilai biner ini umumnya dinyatakan sebagai ‘0’ dan ‘1’, atau ‘True’ dan ‘False’.

Secara formal, aljabar Boolean didefinisikan sebagai sebuah tupel dari himpunan nilai biner dan operator-operator Boolean. Misalkan *B* adalah himpunan yang terdiri atas nilai 0 dan 1, dan *B* terdefinisi pada operator biner + dan ·, dan operator uner ’. Tupel $< B, +, \cdot, ', 0, 1 >$ adalah aljabar Boolean jika untuk setiap $a, b, c \in B$ memenuhi aksioma aljabar Boolean. Aksioma aljabar Boolean adalah sebagai berikut.

1. Aksioma Identitas
   i. $a + 0 = a$
   ii. $a \cdot 1 = a$
2. Aksioma Komutatif
   i. $a + b = b + a$
   ii. $a \cdot b = b \cdot a$
3. Aksioma Distributif
   i. $a \cdot (b + c) = a \cdot b + a \cdot c$
   ii. $a + (b \cdot c) = (a + b) \cdot (a + c)$
4. Aksioma Komplemen
   i. $a + a' = 1$
   ii. $a \cdot a' = 0$

Aljabar Boolean umumnya dinyatakan dalam bentuk ekspresi dan fungsi Boolean. Ekspresi Boolean dapat berbentuk 0, 1, $a + b$, $a \cdot b$, $a \cdot b' + c \cdot d$, dan sebagainya. Fungsi Boolean dapat memiliki sejumlah peubah. Dalam konteks aljabar Boolean, peubah fungsi Boolean disebut literal. Contoh dari fungsi Boolean adalah $f(x) = x'$, $f(x, y) = xy + x'$, dan sebagainya.

B. Hukum-Hukum Aljabar Boolean

Dalam Aljabar Boolean berlaku 11 hukum dasar. Hukum dasar tersebut adalah sebagai berikut.

1. Hukum Identitas
   i. $a + 0 = a$
   ii. $a \cdot 1 = a$
2. Hukum Idempoten
   i. $a + a = a$
   ii. $a \cdot a = a$
3. Hukum Komplemen
   i. $a + a' = 1$
   ii. $aa' = 0$
4. Hukum Dominasi
   i. $a \cdot 0 = 0$
   ii. $a + 1 = 1$
5. Hukum Involusi
   i. $(a')' = a$

6. Hukum Penyerapan
   i. $a + ab = a$
   ii. $a(a + b) = a$
7. Hukum Komutatif
   i. $a + b = b + a$
   ii. $ab = ba$
8. Hukum Asosiatif
   i. $a + (b + c) = (a + b) + c$
   ii. $a(bc) = (ab)c$
9. Hukum Distributif
   i. $a + (bc) = (a + b)(a + c)$
   ii. $a(b + c) = ab + ac$
10. Hukum De Morgan
   i. $(a + b)' = a'b'$
   ii. $(ab)' = a' + b'$
11. Hukum 0-1
   i. $0' = 1$
   ii. $1' = 0$

C. Bentuk Kanonik dari Ekspresi Boolean

Sebuah ekspresi Boolean dapat dinyatakan dalam dua bentuk kanonik. Bentuk pertama adalah penjumlahan dari suku *minterm* (*Sum of Products*/SOP). Sebuah suku *minterm* adalah ekspresi Boolean yang mengandung perkalian dari literal lengkap. Untuk bentuk *minterm*, literal dengan nilai 0 dinyatakan dalam bentuk komplemen, sedangkan nilai 1 tanpa komplemen. Contoh penulisan sebuah fungsi Boolean dalam bentuk kanonik SOP adalah $f(x, y, z) = x'y'z + xy'z' + xyz$. [3]

Bentuk kedua adalah bentuk perkalian dari suku *maxterm* (*Product of Sums*/POS). Kebalikan dengan suku *minterm*, sebuah suku *maxterm* merupakan ekspresi Boolean yang mengandung penjumlahan dari literal lengkap. Untuk bentuk *maxterm*, literal dengan nilai 1 dinyatakan dalam bentuk komplemen, sedangkan nilai 0 tanpa komplemen. Contoh penulisan sebuah fungsi Boolean dalam bentuk kanonik POS adalah $g(x, y, z) = (x + y + z)(x + y' + z)(x + y' + z')(x' + y + z')(x' + y' + z)$. [3]

Bentuk kanonik (SOP/POS) dapat diperoleh dari tabel kebenaran sebuah ekspresi. Untuk menyusun bentuk kanonik SOP, hal yang perlu dilakukan adalah mengambil suku *minterm* dari tabel. Suku *minterm* diperoleh dari nilai 1 pada fungsi tabel kebenaran. Sebaliknya, bentuk kanonik POS disusun dengan mengambil suku *maxterm* dari tabel. Suku *maxterm* diperoleh dari nilai 0 pada fungsi tabel kebenaran.

Berikut adalah contoh konversi ekspresi Boolean ke bentuk kanonik.

x	*y*	*z*	*f*(*x*, *y*, *z*)
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Tabel I. Contoh Tabel Kebenaran dari sebuah Fungsi Boolean (Munir, R., 2020) [3]

Pada tabel di atas, bentuk kanonik SOP dari fungsi Boolean adalah

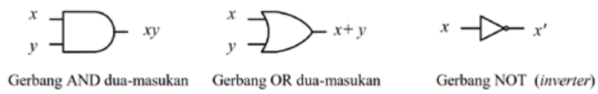
$$f(x, y, z) = x'y'z + xy'z' + xyz = m_1 + m_4 + m_7$$
$$f(x, y, z) = \Sigma(1,4,7)$$

Sementara itu, bentuk kanonik POS dari fungsi Boolean di atas adalah

$$f(x, y, z) = (x + y + z)(x + y' + z)(x + y' + z')(x' + y + z')(x' + y' + z)$$
$$f(x, y, z) = M_0M_2M_3M_5M_6 = \Pi(0,2,3,5,6)$$

D. Gerbang Logika

Gerbang logika adalah salah satu metode merepresentasikan fungsi Boolean. Terdapat tiga gerbang logika dasar, yaitu gerbang NOT, gerbang AND, dan gerbang OR. Ketiga gerbang logika tersebut masing-masing menyatakan operasi komplemen, perkalian, dan penjumlahan Boolean. Ketiga gerbang logika tersebut disimbolkan sebagai berikut.



Gambar 1. Simbol Gerbang Logika AND, OR dan NOT (Munir, R., 2020) [3]

Keluaran dari ketiga gerbang logika tersebut dinyatakan dalam tabel kebenaran berikut.

x	*y*	NOT *x*	*x* AND *y*	*x* OR *y*
0	0	1	0	0
0	1	1	0	1
1	0	0	0	1
1	1	0	1	1

Tabel II. Tabel Kebenaran Gerbang Logika NOT, AND dan OR

E. Peta Karnaugh

Peta Karnaugh (K-map) adalah teknik yang digunakan untuk menyederhanakan ekspresi dari fungsi Boolean. K-map ditemukan oleh Maurice Karnaugh pada 1953. K-map berbentuk diagram yang terdiri atas beberapa sel. Setiap sel merepresentasikan suku *minterm*. Dalam penyusunan K-map, setiap sel yang bertetangga hanya berbeda dalam 1 buah literal. Contoh K-map adalah sebagai berikut.

		y 0	1
x	0	*x'y'*	*x'y*
	1	*xy'*	*xy*

Gambar 2. Susunan K-map dari Fungsi Boolean Dua Literal [4]

		yz 00	01	11	10
x	0	*x'y'z'*	*x'y'z*	*x'yz*	*x'yz'*
	1	*xy'z'*	*xy'z*	*xyz*	*xyz'*

Gambar 3. Susunan K-map dari Fungsi Boolean Tiga Literal [4]

wx \ yz	00	01	11	10
00	w'x'y'z'	w'x'y'z	w'x'yz	w'x'yz'
01	w'xy'z'	w'xy'z	w'xyz	w'xyz'
11	wxy'z'	wxy'z	wxyz	wxyz'
10	wx'y'z'	wx'y'z	wx'yz	wx'yz'

Gambar 4. Susunan K-map dari Fungsi Boolean Empat Literal [4]

Setelah membentuk K-map, penyederhanaan dilakukan dengan membentuk kelompok sel bersisian yang bernilai 1. Pembentukan kelompok dilakukan berdasarkan pasangan, kuad, atau oktet. Kaidah ini dijelaskan lebih detail pada gambar di bawah.

wx \ yz	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	0	0	1	1
10	0	0	0	0

Gambar 5. Pengelompokan Pasangan pada K-map [4]

wx \ yz	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	1	1	1
10	0	0	0	0

Gambar 6. Pengelompokan Kuad (1) pada K-map [4]

wx \ yz	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	1	0	0
10	1	1	0	0

Gambar 7. Pengelompokan Kuad (2) pada K-map [4]

wx \ yz	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	1	1	1
10	1	1	1	1

Gambar 8. Pengelompokan Oktet pada K-map [4]

Setelah pengelompokan dilakukan, sebuah fungsi Boolean dapat dinyatakan dalam bentuk yang lebih sederhana. Sebagai contoh, fungsi Boolean pada Gambar 6. berbentuk

$$f(w,x,y,z) = wxy'z' + wxy'z + wxyz + wxyz'$$

setelah dilakukan pengelompokan pada K-map, fungsi f disederhanakan menjadi $f(w,x,y,z) = wx$.

Pengelompokkan pada K-map mempunyai dua aturan khusus. Aturan khusus pertama adalah penggulungan K-map. Pada aturan ini, sekelompok sel *minterm* pada ujung K-map (misal ujung kiri) dapat dikelompokkan dengan sekelompok sel pada ujung lain (dalam hal ini ujung kanan) apabila memenuhi syarat pengelompokan.

xy \ zt	00	01	11	10
00	1	1	0	0
01	0	0	0	1
11	0	0	1	1
10	1	1	1	1

Gambar 9. Contoh Pengelompokan dengan Aturan Penggulungan [4]

Aturan khusus kedua adalah kondisi *don't care*. Aturan khusus ini berlaku untuk suku-suku *minterm* yang tidak dianggap tidak penting dalam sebuah fungsi Boolean. Menurut aturan ini, sel-sel dari suku *minterm* tersebut dapat digunakan dalam pengelompokan sel secara bebas.

wx \ yz	00	01	11	10
00	1	0	1	0
01	1	1	1	0
11	X	X	X	X
10	X	X	X	X

Gambar 10. Contoh Pengelompokan dengan Aturan *Don't Care* (X adalah suku yang diabaikan) [4]

F. Aljabar Boolean dalam Rangkaian Digital

Aljabar Boolean dapat digunakan dalam merancang rangkaian digital. Proses merancang rangkaian digital dengan aljabar Boolean adalah: mendeskripsikan fungsi rangkaian, menuliskan dan menyederhanakan persamaan fungsi rangkaian, dan mengimplementasikannya sebagai rangkaian gerbang logika [6]. Dengan rangkaian gerbang logika, perilaku dari sebuah rangkaian digital dapat dengan mudah dipahami. Selain itu, sebuah gerbang logika dapat disubstitusi dengan rangkaian transistor yang sesuai.

III. PEMBAHASAN

A. Contoh Perancangan Rangkaian Digital – Pengecekan Bilangan Prima pada Bilangan 4-bit

Pada contoh ini terdapat 4 masukan rangkaian digital. Keempat masukan tersebut merepresentasikan sebuah bilangan biner 4-bit. Keluaran dari rangkaian digital adalah 1 jika bilangan tersebut merupakan bilangan prima dalam desimal, dan 0 jika tidak (diadaptasi dari *Ex. 2.58*, *Digital Design 2nd Ed.* [6]).

Berdasarkan deskripsi permasalahan, sebuah tabel kebenaran dapat disusun berdasarkan keempat masukan dan keluaran. Misal keempat input dilabeli w, x, y, z, dengan *w* sebagai *most significant bit*, dan *z* sebagai *least significant bit*. Apabila keluaran diberikan label $f(w,x,y,z)$, maka tabel kebenaran akan berbentuk seperti berikut.

Desimal	*w*	*x*	*y*	*z*	*f(w,x,y,z)*
0	0	0	0	0	0
1	0	0	0	1	0
2	**0**	**0**	**1**	**0**	**1**
3	**0**	**0**	**1**	**1**	**1**
4	0	1	0	0	0
5	**0**	**1**	**0**	**1**	**1**
6	0	1	1	0	0
7	**0**	**1**	**1**	**1**	**1**
8	1	0	0	0	0
9	1	0	0	1	0
10	1	0	1	0	0
11	**1**	**0**	**1**	**1**	**1**
12	1	1	0	0	0
13	**1**	**1**	**0**	**1**	**1**
14	1	1	1	0	0
15	1	1	1	1	0

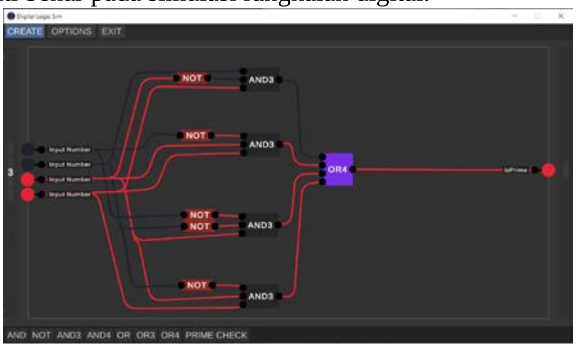
Tabel III. Tabel Kebenaran Rangkaian Pengecekan Bilangan Prima

Berdasarkan Tabel III., salah satu konfigurasi K-map yang dapat disusun adalah sebagai berikut.

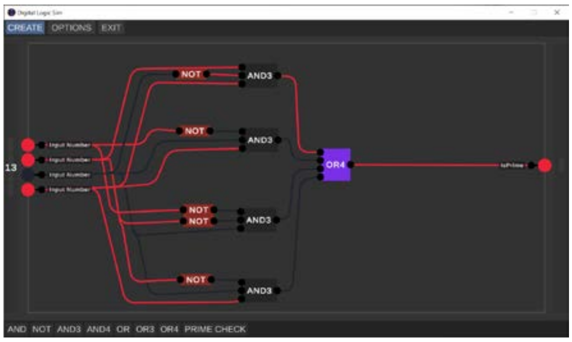
wx \ yz	00	01	11	10
00	0	0	1	1
01	0	1	1	0
11	0	1	0	0
10	0	0	1	0

Gambar 11. K-map untuk Rangkaian Pengecekan Bilangan Prima

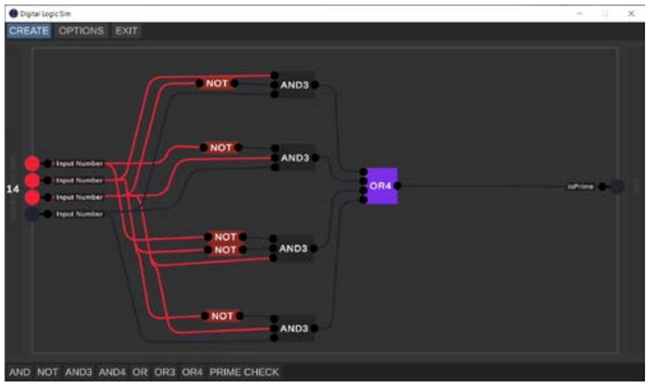
Hasil penyederhanaan adalah sebanyak empat suku *minterm*, yaitu $f(w,x,y,z) = xy'z + w'yz + w'x'y + x'yz$. Fungsi ini terbukti benar pada simulasi rangkaian digital.



Gambar 12. Simulasi Rangkaian Pengecekan Bilangan Prima, Input 0011 (desimal 3), Output 1



Gambar 13. Simulasi Rangkaian Pengecekan Bilangan Prima, Input 1101 (desimal 13), Output 1



Gambar 14. Simulasi Rangkaian Pengecekan Bilangan Prima, Input 1110 (desimal 14), Output 0

B. Contoh Perancangan Rangkaian Digital – Sensor Tangki Bensin

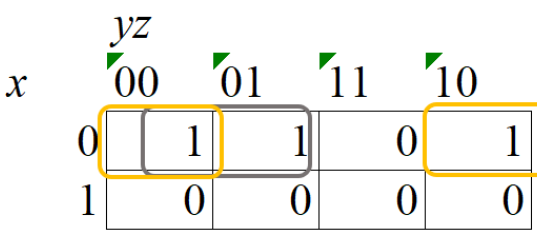
Pada contoh ini, tangki bensin memiliki sensor pendeteksi jumlah bensin. Sensor ini menghasilkan keluaran berupa bilangan biner 3-bit. Keluaran 111 berarti tangki penuh, sedangkan 000 berarti tangki kosong. Dalam masalah ini, rangkaian digital yang dirancang adalah sensor pendeteksi kondisi *low fuel*. Dalam contoh ini, kondisi *low fuel* tercapai ketika jumlah bensin berada di bawah level 3 (desimal). Apabila memasuki kondisi *low fuel*, sebuah lampu akan menyala (diadaptasi dari *Ex. 2.59*, *Digital Design 2nd Ed.* [6]).

Berdasarkan deskripsi permasalahan, rangkaian digital sensor *low fuel* menerima 3 masukan, dan mengeluarkan 1 keluaran (status lampu). Misal ketiga input diberikan label x, y, z, dan keluaran adalah $g(x, y, z)$. Tabel kebeneran dari rangkaian yang dibuat adalah sebagai berikut.

Desimal	*x*	*y*	*z*	*g(x,y,z)*
0	**0**	**0**	**0**	**1**
1	**0**	**0**	**1**	**1**
2	**0**	**1**	**0**	**1**
3	0	1	1	0
4	1	0	0	0
5	1	0	1	0
6	1	1	0	0
7	1	1	1	0

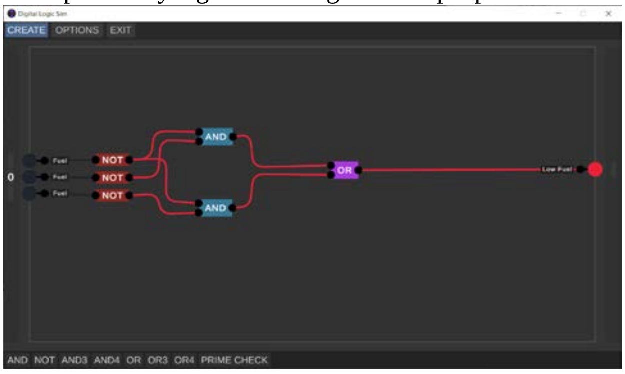
Tabel IV. Tabel Kebenaran Rangkaian Sensor *Low Fuel*

Konfigurasi K-map yang dapat disusun dari Tabel IV. adalah sebagai berikut.

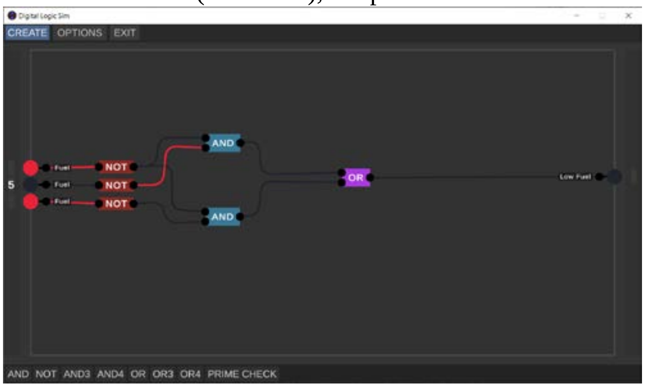


Gambar 15. K-map untuk Rangkaian Sensor *Low Fuel*

Hasil penyederhanaan adalah sebanyak dua suku *minterm*, yaitu $g(x, y, z) = x'y' + x'z'$. Rangkaian dari fungsi ini memiliki perilaku yang sesuai dengan deskripsi permasalahan.



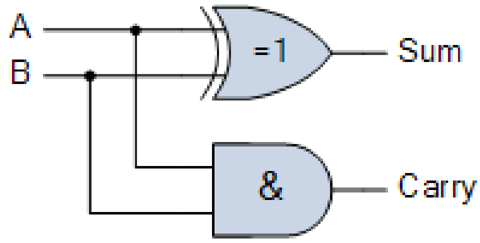
Gambar 16. Simulasi Rangkaian Sensor *Low Fuel*, Input 000 (desimal 0), Output 1



Gambar 17. Simulasi Rangkaian Sensor *Low Fuel*, Input 101 (desimal 5), Output 0

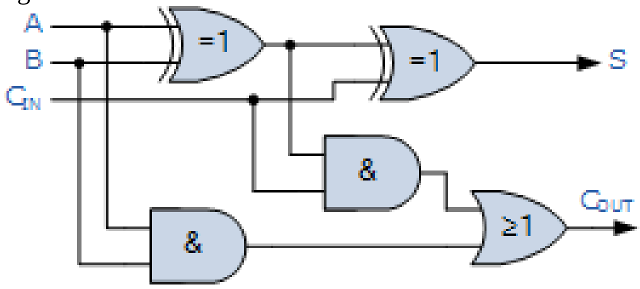
C. Rangkaian Penjumlahan Bilangan Biner

Rangkaian penjumlahan bilangan biner menggunakan sistem *sum* dan *carry* [5]. Nilai *sum* didefinisikan sebagai penjumlahan dari dua bit. Nilai *sum* diperoleh dengan melakukan operasi XOR (XOR menghasilkan 1 jika kedua input berbeda). Nilai *carry* adalah nilai bit yang dibawa pada penjumlahan bit selanjutnya. Nilai *carry* diperoleh dengan melakukan operasi AND.



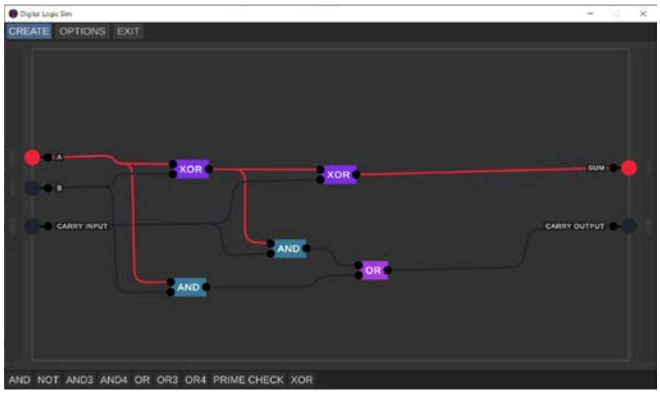
Gambar 18. Rangkaian Logika Penjumlahan Bit Biner [5]

Penjumlahan bit biner umumnya menggunakan *full adder*. *Full adder* menerima dua bit yang akan dijumlahkan, serta satu masukan bit *carry* yang dijumlahkan pada hasil penjumlahan. Keluaran dari rangkaian ini adalah satu bit *sum* dan satu bit *carry*. Rangkaian logika dari penjumlahan *full adder* adalah sebagai berikut.



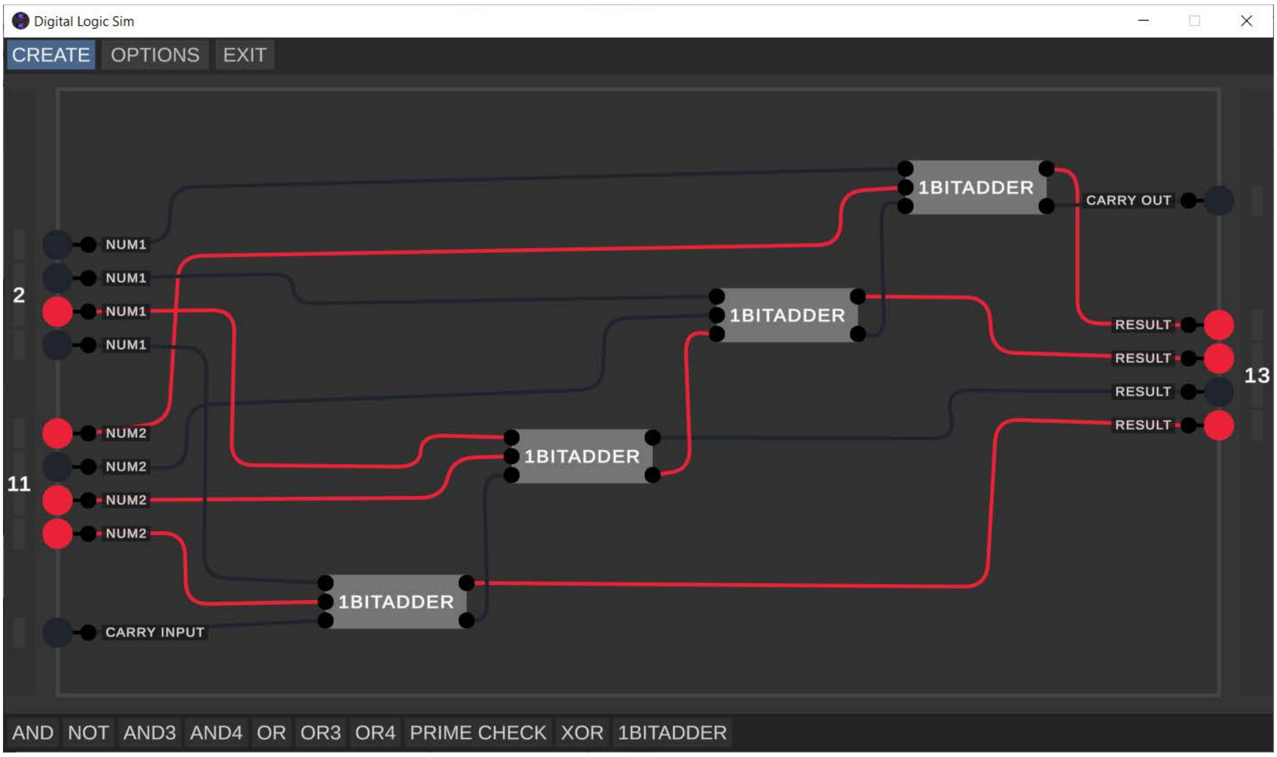
Gambar 19. Rangkaian Logika Penjumlahan *Full Adder* [5]

Dalam program simulasi, *full adder* dapat dibuat seperti berikut.

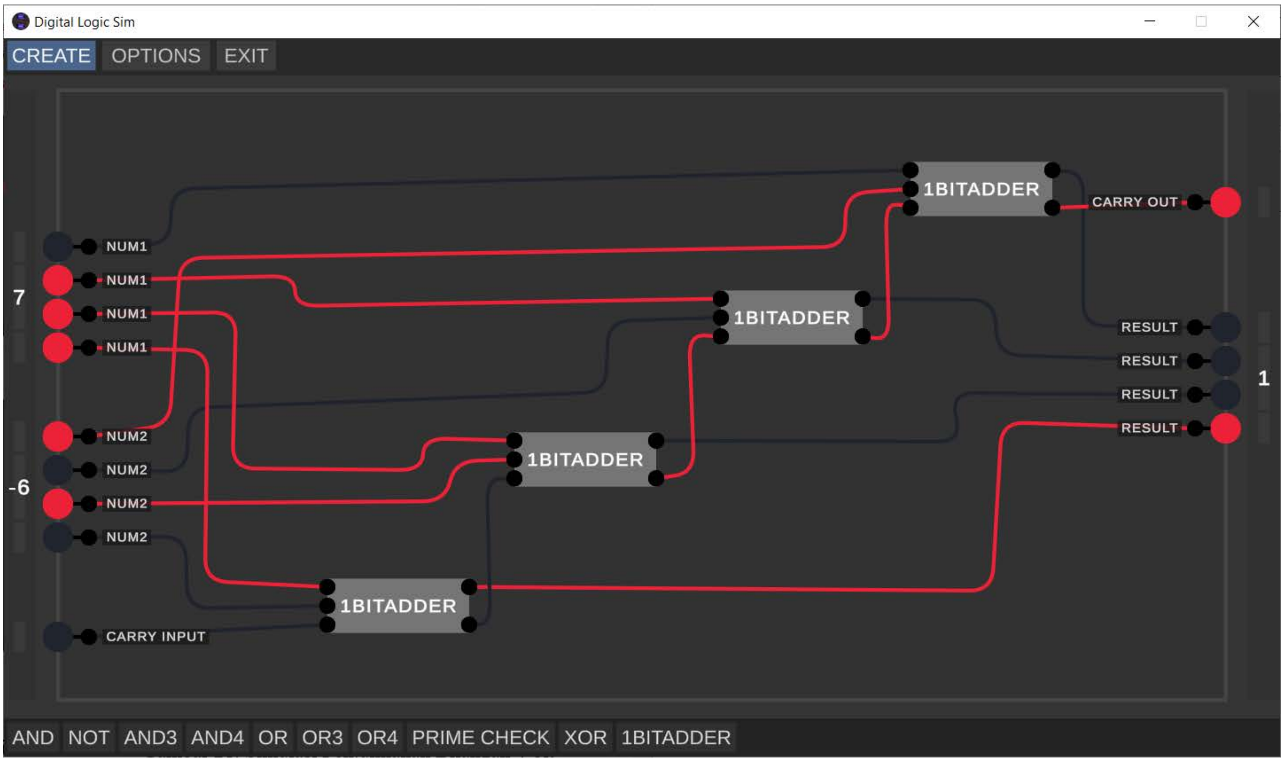


Gambar 20. Simulasi Rangkaian *Full Bit Adder*

Komponen *full bit adder* ini kemudian dapat digunakan untuk melakukan penjumlahan bilangan biner. Sebagai contoh, berikut adalah rangkaian penjumlahan bilangan biner 4-bit.



Gambar 21. Simulasi Penjumlahan Bilangan 4-bit (*unsigned integer*)



Gambar 22. Simulasi Penjumlahan Bilangan 4-bit (*signed integer*)

D. Aljabar Boolean pada Sequential Circuit

Ketiga contoh di atas merupakan contoh penerapan aljabar Boolean pada rangkaian digital. Ketiga tersebut merupakan contoh *combinational circuit*. *Combinational circuit* adalah rangkaian digital yang merupakan fungsi terhadap kombinasi masukan rangkaian. Rangkaian jenis ini merupakan rangkaian sederhana tanpa penyimpanan informasi [6].

Untuk menyimpan informasi, rangkaian digital memanfaatkan konsep *sequential logic*. Rangkaian yang memakai konsep ini mampu menyimpan memori terhadap operasi sebelumnya. Rangkaian jenis ini bergantung pada *state* yang sedang dialami oleh rangkaian. Rangkaian yang menggunakan konsep ini di antaranya adalah *latch, clock, flip-flop,* dan *register/memory* [6].

IV. Simpulan

Aljabar Boolean dapat digunakan dalam penyusunan dan analisis rangkaian digital. Dengan aljabar Boolean, sebuah rangkaian digital kompleks dapat ditinjau secara sederhana menggunakan rangkaian gerbang logika yang merepresentasikan perilaku rangkaian. Berdasarkan contoh dan pembahasan yang penulis berikan, penulis juga dapat menyimpulkan bahwa penggunaan aljabar Boolean sangat luas dalam bidang elektronika.

V. Ucapan Terima Kasih

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa karena dengan berkat-Nya, penulis dapat menyelesaikan makalah ini dengan baik dan tepat waktu. Penulis ingin mengucapkan kepada Dr. Nur Ulfa Maulidevi, S.T., M.Sc. selaku dosen Kelas 1 Mata Kuliah IF2120 Matematika Diskrit karena telah membimbing dan mengajari penulis tentang konsep dasar aljabar Boolean. Penulis ingin mengucapkan terima kasih kepada pencipta referensi yang penulis gunakan dalam penyelesaian makalah ini. Penulis juga ingin mengucapkan terima kasih kepada kedua orang tua penulis yang telah mendukung penulis dalam proses penyusunan makalah.

Penulis bersyukur karena mendapatkan kesempatan untuk mengeksplorasi topik matematika diskrit dan elektronika. Penulis berharap bahwa makalah yang tidak sempurna ini dapat bermanfaat bagi pembaca.

Daftar Referensi

[1] Lague, S. (2020). *Digital logic sim by Sebastian Lague*. itch.io. Diakses Desember 10, 2022, dari https://sebastian.itch.io/digital-logic-sim

[2] Lague, S. (2020). *Exploring how computers work*. YouTube. Diakses Desember 10, 2022, dari https://youtu.be/QZwneRb-zqA

[3] Munir, R. (2020). *Aljabar Boolean (bag. 1)*. IF2120 - Matematika Diskrit. Diakses Desember 9, 2022, dari https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Aljabar-Boolean-(2020)-bagian1.pdf

[4] Munir, R. (2020). *Aljabar Boolean (bag.2)*. IF2120 Matematika Diskrit. Diakses Desember 9, 2022, dari https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Aljabar-Boolean-(2020)-bagian2.pdf

[5] Storr, W. (2022). *Binary Adder and Binary Addition using Ex-OR Gates*. Basic Electronics Tutorials. Diakses Desember 10, 2022, dari https://www.electronics-tutorials.ws/combination/comb_7.html

[6] Vahid, F. (2010). *Digital Design: With RTL Design, VHDL, and Verilog*. Wiley-Blackwell.

Pernyataan

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 11 Desember 2022

Eugene Yap Jin Quan 13521074