Aplikasi Aljabar Boolean dalam Komparator Digital

Ade Yusuf Rahardian / 13514079[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]*13514079@std.stei.itb.ac.id*

***Abstract*—Makalah ini menjelaskan bagaimana aljabar boolean digunakan di dua jenis komparator digital, yaitu *equality comparator* dan *magnitude comparator*. Komparator adalah komponen elektronik yang berfungsi membandingkan dua nilai kemudian memberikan hasilnya. Komparator banyak digunakan misalnya pada mesin penyeleksi surat, baik ukuran dimensinya, berat surat, kode area (berdasarkan bar-code), dan sebagainya.**

***Keywords*—Aljabar Boolean, Digital, Komparator, Rangkaian Logika**

I. PENDAHULUAN

Dalam sebuah sistem digital, membandingkan dua buah nilai (data digital) merupakan operasi aritmatika yang paling dasar. Hasil dari perbandingan itu adalah sama dengan atau tidak sama dengan (lebih kecil atau lebih besar). Data digital yang dibandingkan dimulai dari yang paling kecil (1-bit) hingga ke data yang lebih besar lagi.

Di zaman modern ini, semua kebutuhan manusia pastilah ingin seefisien dan sehemat mungkin. Termasuk dalam komponen sebuah komparator digital. Semakin sedikit komponen yang dipakai, semakin hemat pula biaya yang dikeluarkan. Dengan adanya fungsi aljabar boolean yang dapat disederhanakan, diharapkan komparator yang mengeluarkan *output* yang sama dapat diperoleh rangkaian dengan komponen yang dibutuhkan seminimal mungkin.

II. DASAR TEORI

A. Definisi Aljabar Boolean

Misalkan B adalah himpunan yang didefinisikan pada dua operator biner, + dan ·, dan sebuah operator uner, ‘. Misalkan 0 dan 1 adalah dua elemen yang berbeda dari B. Maka, tupel <B, +, ·, ‘, 0, 1> disebuh aljabar Boolean jika untuk setiap a, b, c ∈ B berlaku aksioma berikut:

1. Identitas
   (i) a + 0 = a
   (ii) a · 1 = a
2. Komutatif
   (i) a + b = b + a
   (ii) a · b = b · a
3. Distributif
   (i) a · (b + c) = (a · b) + (a · c)
   (ii) a + (b · c) = (a + b) · (a + c)
4. Komplemen
   Untuk setiap a ∈ B terdapat elemen unik a’ ∈ B sehingga
   (i) a + a’ = 1
   (ii) a · a’ = 0

Berhubung elemen-elemen B tidak didefinisikan nilainya (kita bebas menentukan anggota-anggota B), maka terdapat banyak sekali aljabar boolean. Untuk mempunyai sebuah aljabar Boolean, orang harus memperlihatkan:

1. elemen-elemen himpunan B,
2. kaidah/aturan operasi untuk dua operator biner dan operator uner,
3. himpunan B, bersama-sama dengan dua operator tersebut, memenuhi keempat aksioma di atas.

B. Fungsi Boolean

Fungsi boolean adalah pemetaan dari B^n ke B melalui ekspresi Boolean, kita menuliskannya sebagai :

$f : B^n \rightarrow B$

yang dalam hal ini B^n adalah himpunan yang beranggotakan pasangan terurut ganda-n di dalam daerah asal B.

Contoh-contoh fungsi Boolean:

1. *f(x) = x*
2. *f(x, y) = x’y + xy’+ y’*
3. *f(x, y) = x’ y’*
4. *f(x, y) = (x + y)’*
5. *f(x, y, z) = xyz’*

Setiap peubah di dalam fungsi Boolean, termasuk dalam bentuk komplemennya disebut literal.

C. Bentuk Kanonik

Ekspresi Boolean yang menspesifikasikan suatu fungsi dapat disajikan dalam dua bentuk berbeda. Pertama, sebagai penjumlahan dari hasil kali dan kedua sebagai perkalian dari hasil jumlah. Misalnya,

f(x,y,z) = x’y’z + xy’z’ + xyz

dan

g(x,y,z) = (x + y + z) (x + y’ +z) (x + y’ + z’) (x’ + y + z’) (x’ + y’ +z)

adalah dua buah fungsi yang sama. Fungsi yang pertama, *f*, muncul dalam bentuk penjumlahan dari hasil kali, sedangkan fungsi yang kedua, *g*, muncul dalam bentuk perkalian dari hasil jumlah. Setiap suku (*term)* di dalam ekspresi mengandung literal yang lengkap dalam peubah x, y, dan z, baik peubahnya tanpa komplemen maupun dengan komplemen.

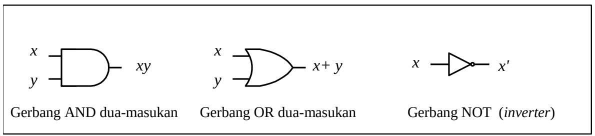
Ada dua macam bentuk *term*, yaitu *minterm* dan *maxterm*. *Minterm* yaitu apabila suku di dalam ekspresi boolean mengadung literal yang lengkap dalam bentuk hasil kali. Sedangkan *maxterm* yaitu apabila suku di dalam ekspresi boolean mengandung literal yang lengkap dalam bentuk hasil jumlah.

Ekspresi Boolean yang dinyatakan sebagai penjumlahan dari satu atau lebih *minterm* atau perkalian dari satu atau lebih *maxterm* disebut dalam bentuk kanonik. Jadi, ada dua macam bentuk kanonik :

1. Penjumlahan dari hasil kali (*sum-of-product* atau SOP)
2. Perkalian dari hasil jumlah (*product-of-sum* atau POS)

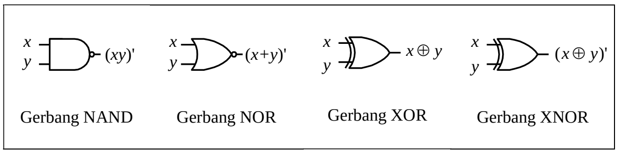
D. Rangkaian Logika

Fungsi Boolean dapat juga direpresentasikan dalam bentuk rangkaian logika. Ada tiga gerbang logika dasar: gerbang AND, gerbang OR, dan gerbang NOT.



Gambar 1. Gerbang logika dasar[1]

Dan gerbang logika turunan.



Gambar 2. Gerbang logika turunan[1]

E. Peta Karnaugh

Peta Karnaugh (atau *K-map*) merupakan metode grafis untuk menyederhanakan fungsi Boolean. Metode ini ditemukan oleh Maurice Karnaugh pada tahun 1953. Peta Karnaugh adalah sebuah diagram/peta yang terbentuk dari kotak-kotak (berbentuk bujursangkar) yang bersisian. Tiap kotak merepresentasikan sebuah *minterm.* Tiap kotak dikatakan bertetangga jika *minterm-minterm* yang merepresentasikannya berbeda hanya 1 buah literal.

Contoh penyajian Peta Karnaugh ditunjukkan seperti gambar di bawah ini.

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

Gambar 3. Peta Karnaugh dengan dua peubah[1]

		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 4. Peta Karnaugh dengan tiga peubah[1]

	yz			
	00	01	11	10
wx 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 5. Peta Karnaugh dengan empat peubah[1]

Cara mengisi peta Karnaugh adalah kotak yang menyatakan *minterm* diisi "1", sisanya diisi 0. Contoh :

f(x, y, z) = x'yz' + xyz' + xyz

		yz			
		00	01	11	10
x	0	0	0	0	1
	1	0	0	1	1

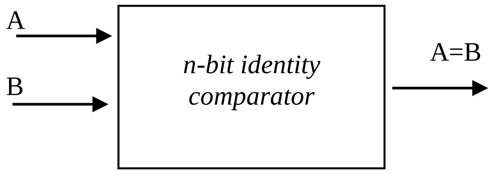
Gambar 6. Pengisian Peta Karnaugh[1]

Penggunaan Peta Karnaugh dalam penyederhanaan fungsi Boolean dilakukan dengan cara menggabungkan kotak-kotak yang bernilai 1 dan saling bersisian. Kelompok kotak yang bernilai 1 dapat membentuk pasangan (dua), kuad (empat), dan oktet (delapan).

III. PRINSIP KERJA KOMPARATOR

Prinsip kerja komparator adalah untuk membandingkan dua *n-bit binary words* (misal A dan B). Terdapat dua jenis *digital comparator*, yaitu

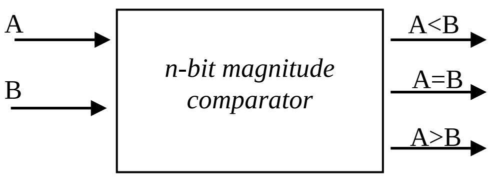
1. *Identity/Equality Comparator*



Gambar 7. *Identity/Equality Comparator*

Sebuah *digital comparator* yang hanya memiliki satu terminal *output* (A = B). Komparator ini berfungsi untuk mendeteksi apakah dua buah data n-bit nilainya sama atau tidak. Implementasinya dengan gerbang XNOR untuk setiap bit. Karena gerbang XNOR akan bernilai ‘1’ apabila kedua input sama.

2. *Magnitude Comparator*



Gambar 8. *Magnitude Comparator*

Sebuah *digital comparator* yang memiliki tiga terminal output, yaitu A < B, A = B, dan A > B. Komparator ini tidak hanya mengecek apakah dua buah data n-bit nilainya sama atau tidak, komparator ini bisa mendeteksi data mana yang lebih besar maupun yang lebih kecil.

Komparator banyak digunakan misalnya pada mesin penyeleksi surat, baik ukuran dimensinya, berat surat, kode area (berdasarkan bar-code), dan sebagainya.

IV. PENGGUNAAN ALJABAR BOOLEAN DALAM KOMPARATOR

Dalam bab ini akan dijelaskan penggunaan aljabar boolean dalam komparator 1-bit dan 2-bit pada *equality comparator* dan *magnitude comparator.*

4.1. 1-bit Equality Comparator

4.1.1. Tabel Kebenaran

Desimal		Biner		A=B
A	B	a_0	b_0	
0	0	0	0	1
0	1	0	1	0
1	0	1	0	0
1	1	1	1	1

Tabel I. Tabel Kebenaran untuk *1-bit Equality Comparator*

4.1.2. Peta Karnaugh

	b_0	0	1
a_0	0	1	0
	1	0	1

Gambar 9. Peta Karnaugh untuk *1-bit Equality Comparator*

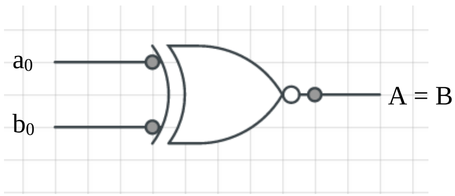
$f(a_0, b_0) = a_0'b_0' + a_0b_0$

Namun, fungsi ini dapat disederhanakan menjadi

$f(a_0, b_0) = (a_0 \oplus b_0)'$

Penyederhanaan fungsi ini perlu karena dalam mendesain sistem, kita ingin berusaha untuk menggunakan komponen yang diperlukan sesedikit mungkin.

4.1.3. Rangkaian Logika



Gambar 10. Rangkaian logika *1-bit Equality Comparator*

4.2. 2-bit Equality Comparator

4.2.1. Tabel Kebenaran

Desimal		Biner				A=B
A	B	a_1	a_0	b_1	b_0	
0	0	0	0	0	0	1
0	1	0	0	0	1	0
0	2	0	0	1	0	0
0	3	0	0	1	1	0
1	0	0	1	0	0	0
1	1	0	1	0	1	1
1	2	0	1	1	0	0
1	3	0	1	1	1	0
2	0	1	0	0	0	0
2	1	1	0	0	1	0
2	2	1	0	1	0	1
2	3	1	0	1	1	0
3	0	1	1	0	0	0
3	1	1	1	0	1	0
3	2	1	1	1	0	0
3	3	1	1	1	1	1

Tabel II. Tabel Kebenaran untuk *2-bit Equality Comparator*

4.2.2. Peta Karnaugh

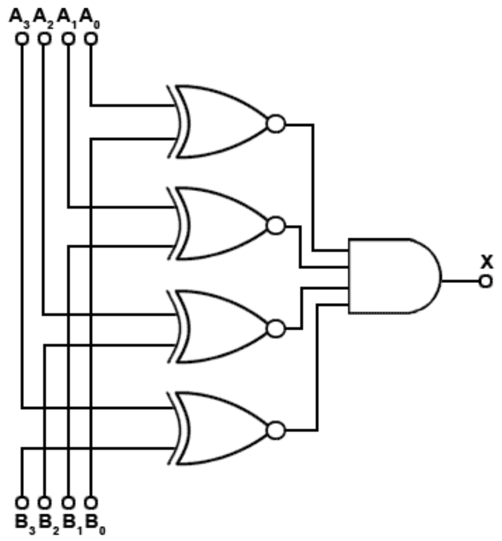
	b_1b_0	00	01	11	10
a_1a_0	00	1	0	0	0
	01	0	1	0	0
	11	0	0	1	0
	10	0	0	0	1

Gambar 11. Peta Karnaugh untuk *2-bit Equality Comparator*

$f(a_1, a_0, b_1, b_0) = a_1'a_0'b_1'b_0' + a_1'a_0b_1'b_0 + a_1a_0b_1b_0 + a_1a_0'b_1b_0'$

4.2.3. Rangkaian Logika

Setelah dilakukan penyederhanaan, digunakan empat gerbang XNOR dan satu gerbang AND, sehingga diperoleh rangkaian logika seperti di bawah ini



Gambar 12. Rangkaian Logika *2-bit Equality Comparator*[2]

4.3. 1-bit Magnitude Comparator

4.3.1. Tabel Kebenaran

Desimal		Biner		A<B	A=B	A>B
A	B	a_0	b_0			
0	0	0	0	0	1	0
0	1	0	1	1	0	0
1	0	1	0	0	0	1
1	1	1	1	0	1	0

Tabel IV. Tabel Kebenaran untuk *1-bit Magnitude Comparator*

4.3.2. Peta Karnaugh

a)

		b_0	
		0	1
a_0	0	0	1
	1	0	0

b)

		b_0	
		0	1
a_0	0	1	0
	1	0	1

c)

		b_0	
		0	1
a_0	0	0	0
	1	1	0

Gambar 13. Peta Karnaugh untuk *1-bit Magnitude Comparator.* (a) A<B, (b) A=B, (c) A>B

Untuk A<B (Gbr 13(a)) berlaku fungsi boolean

$f(a_0,b_0) = a_0'b_0$

Untuk A=B (Gbr 13(b)) berlaku fungsi boolean

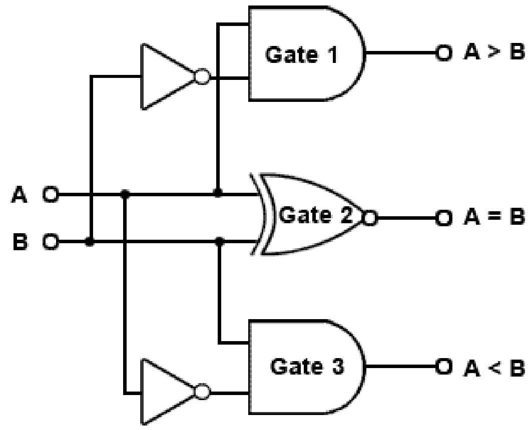
$f(a_0,b_0) = a_0'b_0' + a_0b_0$

Untuk A>B (Gbr 13(c)) berlaku fungsi boolean

$f(a_0,b_0) = a_0b_0'$

4.3.3. Rangkaian Logika

Setelah dilakukan penyederhanaan, diperoleh rangkaian logika seperti di bawah ini



Gambar 14. Rangkaian Logika *1-bit Magnitude Comparator*[2]

4.4. 2-bit Magnitude Comparator

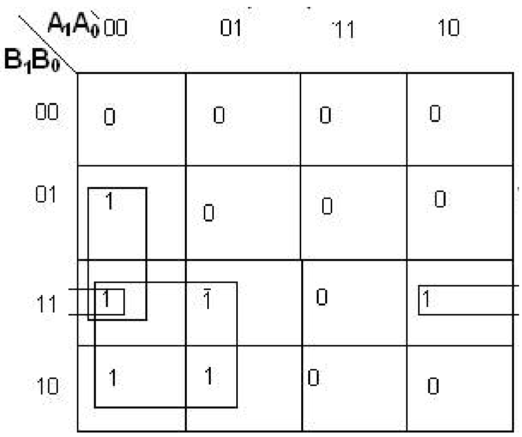
4.4.1. Tabel Kebenaran

Desimal		Biner				A<B	A=B	A>B
A	B	a_1	a_0	b_1	b_0			
0	0	0	0	0	0	0	1	0
0	1	0	0	0	1	1	0	0
0	2	0	0	1	0	1	0	0
0	3	0	0	1	1	1	0	0
1	0	0	1	0	0	0	0	1
1	1	0	1	0	1	0	1	0
1	2	0	1	1	0	1	0	0
1	3	0	1	1	1	1	0	0
2	0	1	0	0	0	0	0	1
2	1	1	0	0	1	0	0	1
2	2	1	0	1	0	0	1	0
2	3	1	0	1	1	1	0	0
3	0	1	1	0	0	0	0	1
3	1	1	1	0	1	0	0	1
3	2	1	1	1	0	0	0	1
3	3	1	1	1	1	0	1	0

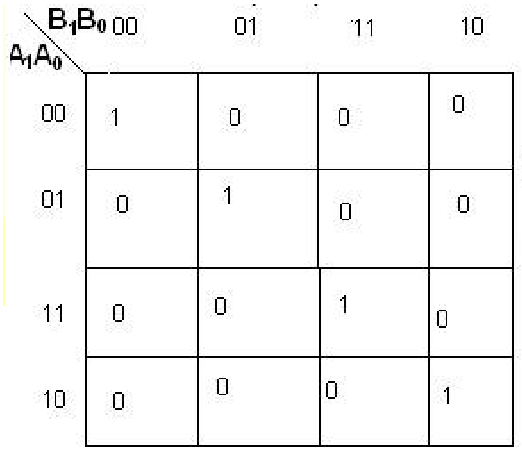
Tabel II. Tabel Kebenaran untuk *2-bit Equality Comparator*

4.4.2. Peta Karnaugh

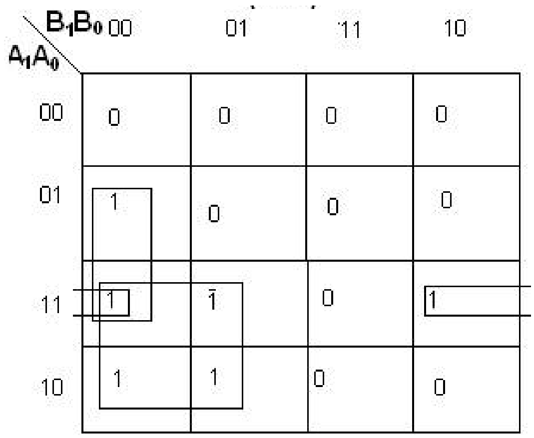
a)



b)



c)



Gambar 15. Peta Karnaugh untuk *2-bit Magnitude Comparator.* (a) A<B, (b) A=B, (c) A>B[4]

Untuk A<B (Gbr. 15(a)) berlaku fungsi boolean

$f(a_1, a_0, b_1, b_0) = a_1'b_1 + a_0'b_1b_0 + a_1'a_0'b_0$

Untuk A=B (Gbr. 15(b)) berlaku fungsi boolean

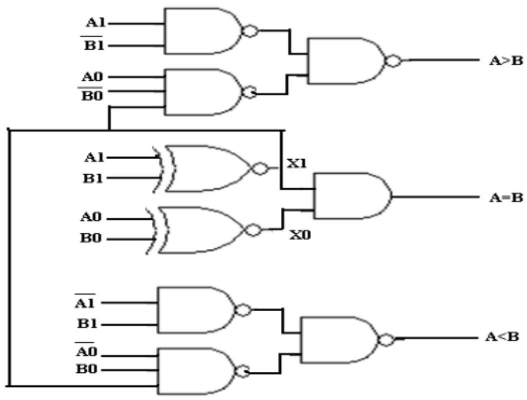
$f(a_1, a_0, b_1, b_0) = a_1'a_0'b_1'b_0' + a_1'a_0b_1'b_0 + a_1a_0b_1b_0 + a_1a_0'b_1b_0'$

Untuk A>B (Gbr. 15(c)) berlaku fungsi boolean

$f(a_1, a_0, b_1, b_0) = a_1b_1' + a_0b_1'b_0' + a_1a_0b_0'$

4.4.3. Rangkaian Logika

Setelah dilakukan penyederhanaan, diperoleh rangkaian logika seperti di bawah ini



Gambar 14. Rangkaian Logika *2-bit Magnitude Comparator*[5]

V. KESIMPULAN

Gerbang logika merupakan bagian utama dari sebuah komparator digital. Untuk mendapatkan komparator digital dengan menggunakan komponen sesedikit mungkin, diperlukan penyederhanaan fungsi. Selain itu penghematan komponen dapat diperoleh dengan mengamati fungsi boolean yang memiliki bentuk kanonik yang sama, sehingga tidak ada komponen ganda yang sebenarnya memiliki fungsi yang sama dalam satu rangkaian.

VI. UCAPAN TERIMA KASIH

Penulis memanjatkan puji syukur kepada Tuhan Yang Maha Esa atas segala kenikmatan sehingga dapat menyelesaikan makalah ini. Penulis juga mengucapkan terima kasih kepada Ibu Dra. Harlili S., M.Sc. dan Bapak Dr.Ir. Rinaldi Munir, MT. selaku dosen pengajar mata kuliah Matematika Diskrit atas segala bimbingan serta ilmu yang telah diberikan kepada penulis. Penulis juga menyampaikan rasa terima kasih kepada rekan-rekan penulis yang mendukung dan mendorong penulis dalam menyelesaikan makalah ini. Terakhir, penulis juga menyampaikan terima kasih kepada seluruh pihak yang ikut membantu pembuatan makalah ini baik yang secara langsung maupun tidak langsung.

REFERENSI

[1] Munir, Rinaldi, *Matematika Diskrit,* Bandung: Penerbit Informatika Bandung.

[2] http://www.learnabout-electronics.org/Digital/dig43.php, diakses pada 8 Desember 2015 pukul 18.00 WIB

[3] http://www.electronics-tutorials.ws/combination/comb_8.html, diakses pada 8 Desember 2015 pukul 18.00 WIB

[4] http://www.exploreroots.com/dc18.html, diakses pada 8 Desember 2015 pukul 18.40 WIB

[5] http://www.ijesi.org/papers/Vol(2)1%20(version%202)/C211324.pdf, diakses pada 8 Desember 2015 pukul 21.20 WIB

[6] http://nainysari.lecture.ub.ac.id/files/2013/09/Rangkaian-Kombinasional-Komparator.pptx, diakses pada 8 Desember 2015 pukul 21.45 WIB

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, 8 Desember 2015

Ade Yusuf Rahardian (13514079)