Mengatur Palang Pintu Kereta Api Menggunakan Aplikasi Aljabar Boolean

Ridho Novan Imanda 202151101

Program Studi Teknik Informatika

Universitas Muria Kudus

202151101@std.umk.ac.id

*Abstract*—Beruntungnya manusia lahir di zaman era modern ini kenapa tidak, semakin maju perkembangan zaman semakin mudah dan praktis. Salah satu kemajuan zaman pada era modern adalah alat transportasi. Seperti kereta api adalah kemajuan transportasi darat, tetapi masih ada bebrapa kendala pada keamanan setiap alat transportasi. Banyak dan sering di Indonesia kecelakaan kereta api, entah diakibatkan oleh pengaman lintasan atau pengguna jalan. Oleh sebab itu dalam makalah ini akan membahas Aljabar Boolean untuk mengatur palang pintu kereta api. Agar mengurangi tragedi penerobosan palang dan menyebabkan kecelakaan.

*Keywords*—Era modern,Transportasi,Palang pintu kereta api, Aljabar Boolean.

# I. PENDAHULUAN

Transportasi pertama diciptakan oleh penjelajah Australia untuk melintasi air, yaitu perahu. Dan orang pertama yang menyebrangi lautan, meskipun ada beberapa bukti pelayaran yang dilakukan 90.000 tahun yang lalu.

Perahu sederhana adalah perahu pertama yang dikenal, dibuat dengan cara melubangi batang pohon besar. Artefak yang membuktikan perahu pertama sekitae 7000-10.000 tahun yang lalu.

Semakin berkembangnya zaman dari beberapa ribu tahun akhirnya manusia mulai menjinakan kuda digunakan sebagai alat transportasi untuk mengangkut barang sekitar pada zaman itu juga manusiia mengenal roda untuk memudahkan transportasi. Pada 3500 SM catatan arkeologi menunjukan bahwa kendaraan beroda pertama kali digunakan, dengan bukti adanya alat yang ditemukan di Mesopotamina, Kaukus Utara, dan Eropa Tengah.

Lokomotif jalam raya ditemukan oleh penemu inggris Richard Trevithick pada tahun 1801, dengan memanfaatkan mesin uap yang telah ditemukan pada tahun 1769. 3 tahun kemudian Trevithick mendesain ulang lokomotif yang berjalan di atas rel dan dapat mengangkut 10 ton besi dengan kecepatan 4 mph.

Seiring perkembangan zaman, penemuan-penamuan semakin hebat dan praktis. Maka dari itu transportasi seperti kereta api harus memberikan rasa aman dan nyaman. Untuk itu maka harus dibangun sebuah gerbang pintu di setiap persimpangan kereta api. Agar mengurangi terjadinya kecelakaan antara kereta api dengan kendaraan lain atau pejalan kaki yang sedang melintas.

# II. ALJABAR BOOLEAN

Aljabar Boolean ditemukan oleh George Boole, pada tahun 1854. Boole melihat bahwa himpunan logika proposisi mempunyai sifat sifat yang serupa pada kemiripan antara hukum-hukum aljabar logika dan hukum-hukum aljabar himpunan. Boole juga menerapkan aturan aturan dasar logikanya di buku The Laws of Thought. Aljabar Boolean merupakan aljabar yang berhubungan dengan variabel-variabel biner dan operasi logika. Variabel-variabel diperlihatkan dengan huruf-huruf alfabet dan tiga operasi dasar dengan AND, OR, dan NOT.

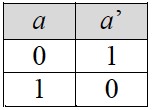
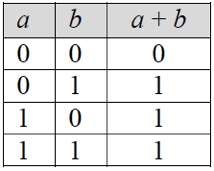
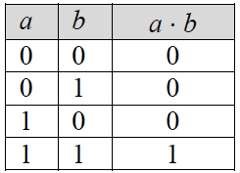
1. *PENGERTIAN*

Aljabar boolean adalah struktur aljabar yang memiliki basis biner(0 dan 1), sesuai dengan tipe data Boolean, 1 jika true dan 0 jika false. Berdasarkan definisi: misalkan B adalah himpunan yang didefinisikan pada dua operator biner + dan, sebuah operator uner, misalkan 0 dan 1 adalah dua elemen yang berbeda dari B. Maka tupel <B,+,0,1> disebut aljabar Boolean jika untuk setiap a,b,c B berlaku aksioma berikut:

1. Closure
2. a + b B
3. a b B
4. Identitas
5. a + 0 = a
6. a 1 = a
7. Komutatif
8. a + b = b + a
9. a b = b a
10. Distributif
11. a (b + c) = (a b) + (a c)
12. a + (b c) = (a + b) (a + c)
13. Komplemen
14. a + a’ = 1
15. a a’ = 0
16. *Hukum-hukum aljabar boolean*
17. Hukum Identitas
18. a + 0 = a
19. a 1 = a
20. Hukum Komplemen
21. a + a’ = 1
22. a a’ = 0
23. Hukum Idempoten
24. a + a = a
25. a a = a
26. Hukum Dominasi
27. a + 1 = 1
28. a 0 = 0
29. Hukum Involusi
30. (a’)’ = a
31. Hukum Penyerapan
32. a + ab = a
33. a(a + b) = a
34. Hukum Asosiatif
35. a + (b + c) = (a + b ) + c
36. a (b . c) = (a . b) c
37. Hukum Komukatif
38. a + b = b + a
39. a . b = b . a
40. Hukum Distributif
41. a (b + c = (a b) + (a c)
42. a + (b . c) = (a + b) (a + c)
43. Hukum De Morgan
44. (a + b)’ = a’ b’
45. (a b)’ = a’ + b’
46. Hukum 0/1
47. 0’ = 1
48. 1’ = 0
49. Aljabar Boolean dua nilai

Aljabar Boolean dua nilai merupakan himpunan B yang memenuhi kaidah aksioma aljabar Boolean yang memiliki dua elemen yaitu 0 dan 1 (B = {0,1}), memiliki operator binner + dan ,serta operator uner’. Kaidah operator uner adalah sebagai berikut:

Tabel 2.1 Operator binner dan Operator uner



1. *Fungsi Boolean*

Suatu fungsi yang dibentuk dari beberapa variable boolean. Setiap perubah di dalam fungsi boolean, termasuk dalam bentuk komplemennya, disebut literal.

Contoh-contoh fungsi boolean :

f(x) = x

f(x, y) = x’y + xy’ + y’

f(x, y) = x’y’

f(x, y) = (x + y)’

f(x, y, z) = xyz’

Fungsi h (x, y, z) = xyz’ terdiri dari 3 buah literal, yaitu x, y, dan z.

Jika diberikan x = 1, y = 1, z = 0, maka nilai fungsinya:

h(1, 1, 0) = 1. 1. 0’ = (1. 1) . 1 = 1.1= 1

1. *Bentuk Kanokik*

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.

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’ )

dua contoh diatas adalah dua buah fungsi yang sama.

Bentuk kanokik yang dapat dibagi menjadi dua jenis, yaitu:

1. SOP (Sum of Product)

Merupakan perkalian dari hasil jumlah. Dapat disebut juga *maxterm* dengan lambing M.

Bentuknya : X1 + X2 + …. + Xn

Notasi : ∑

1. POS (Product of Sum)

Merupakan perkalian dari hasil jumlah. Dapat disebut juga *minterm* dengan lambing m.

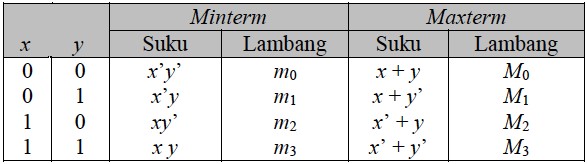
Bentuknya : X1 X2 …..Xn

Notasi : ∏

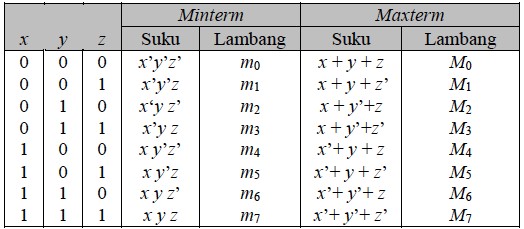
Maxterm : Suku (term) di dalam ekspresi Boolean mengandung literal yang lengkap dalam bentuk hasil jumlah

Minterm : Suku (term) di dalam ekspresi Boolean mengandung literal yang lengkap dalam bentuk hasil kali

Cara membentuk Minterm dan Maxterm dari tabel kebenaran untuk dua perubah

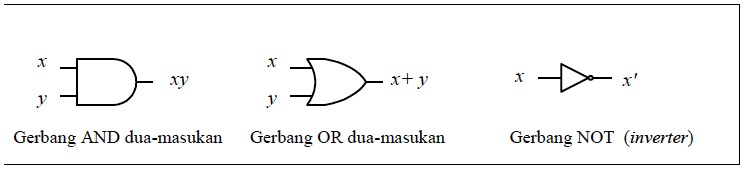


Cara membentuk minterm dan maxterm dari tabel kebenaran intuk tiga perubah

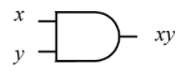


1. *Rangkaian Logika*

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



1. Gerbang AND



Gambar 2.1 Gerbang AND

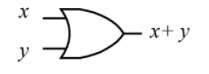
Gerbang AND mempunyai dua atau lebih input dan menghasilkan satu output. Untuk mendapatkan output yang bernilai 1 atau *true*, juga sebaliknya jika ada salah satu input yang bernilai 0 atau false.

Berikut adalah tabel kebenaran gerbang AND:

Tabel 2.5 Gerbang AND dua input



1. Gerbang OR



Gambar 2.2 Gerbang OR

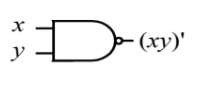
Gerbang OR mempunyai dua atau lebih input dan menghasilkan satu output. Untuk mendapatkan output yang bernilai 1 atau *true*, maka salah satu inputnya harus bernilai 1 atau *true* juga, sebaliknya jika semua inputnya yang bernilai 0 atau *false*, maka outputnya akan bernilai 0 atau *false.*

Berikut adalah tabel kebenaran gerbang OR:

Tabel 2.6 Gerbang OR dua input



1. Gerbang NAND



Gambar 2.3 Gerbang NAND

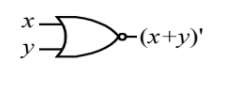
Pada gerbang NAND jika salah satu atau semua inputnya bernilai false, maka akan menghasilkan output yang bernilai 1 atau *true*, sebaliknya jika semua inputnya bernilai true, maka outputnya akan bernilai 0 atau *false*.

Berikut adalah tabel kebenaran gerbang NAND:

Tabel 2.7 Gerbang NAND

|  |  |  |
| --- | --- | --- |
| *x* | *y* | *f(x,y)* |
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

1. Gerbang NOR



Gambar 2.4 Gerbang NOR

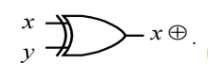
Pada gerbang NOR jika salah satu atau semua inputnya bernilai *true*, maka akan menghasilkan output yang bernilai 0 atau *false*, sebaliknya jika semua inputnya bernilai *false,* maka outputnya akan bernilai 1 atau *true*.

Berikut adalah tabel kebenaran gerbang NOR:

Tabel 2.8 Gerbang NOR

|  |  |  |
| --- | --- | --- |
| *x* | *y* | *f(x,y)* |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

1. Gerbang XOR



Gambar 2.5 Gerbang XOR

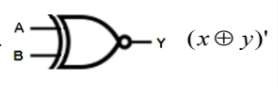
Gerbang XOR akan menghasilkan output *true* jika kedua unput memiliki nilai yang berbeda.

Berikut adalah tabel kebenaran gerbang XOR:

Tabel 2.9 Gerbang XOR

|  |  |  |
| --- | --- | --- |
| *x* | *y* | *f(x,y)* |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

1. Gerbang XNOR



Gambar 2.6 Gerbang XNOR

Gerbang XNOR akan menghasilkan output *true* jika kedua input memilikai nilai yang sama.

Berikut adalah tabel kebenaran gerbang XNOR:

Tabel 2.10 Gerbang XNOR

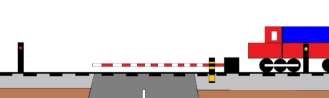
|  |  |  |
| --- | --- | --- |
| *x* | *y* | *f(x,y)* |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

# III. PENERAPAN ALJABAR BOOLEAN PADA PALANG PINTU KERETA API

## 

## Gambar 4.1 Ilustrasi

## Pada gambar diatas merupakan ilustrasi dua sensor yang menangkap datangnya kereta. Sensor tersebut dapat berupa nilai true dan menunjukan warna hijau atau bernilai false menunjukan warna merah. Kedua sensor tersebut (S1 dan S2) hanya dapat merespon jika yang lewat didepannya adalah kereta api.



Gambar 4.2 Ilustrasi dengan kereta

Ada bebrapa cara penerapan aljabar Boolean yang dapat digunakan untuk mengatur palang pintu kereta api.

1. Menggunakan gerbang OR

Gerbang OR menghasilkan output yang bernilai true jika salah satu atau kedua sensor bernilai *true,* maka dapat dimisalkan kedua sensor tersebut di inisialisasikan dengan nilai false dan lampu sensor akan berwarna merah. Palang pintu dapat ditutup jika bernilai *true* dan menggunakan gerbang OR. Jadi jika kereta melewati sensor pertama sensor akan bernilai *true* sehingga outputnya gerbang OR maka palang pintu akan tertutup secara otomatis. Sebaliknya jika kereta bagian belakang meninggalkan sensor outputnya false maka palang pintu akan terbuka.

Berikut adalah tabel kebenaran penggunaan gerbang OR pada palang pintu kereta api

Tabel 4.1 Kondisi palang pintu OR.

|  |  |  |  |
| --- | --- | --- | --- |
| Kondisi kereta | S1 | S2 | Kondisi pintu (S1 + S2) |
| 1 | *false* | *false* | Membuka |
| *2* | *true* | *false* | Menutup |
| *3* | *true* | *true* | Menutup |
| *4* | *false* | *true* | Menutup |
| *5* | *false* | *false* | Membuka |

1. Menggunakan gerbang AND

Jika kedua inputnya bernilai *true* maka kedua sensor menunjukan warna hijau. Jika hasil dari S1 dan S2 menggunakan gerbang AND bernilai false maka palang pintu akan menutup. Jadi jila kereta api melewati sensor pertama sensor akan bernilai *false* sehingga outputnya *false* dan palang pintu akan menutup secara otomatis. Sebaliknya jika kereta bagian belakang meninggalkan sensor outputnya *true* maka palang pintu akan terbuka.

Tabel 4.2 Kondisi palang pintu AND

|  |  |  |  |
| --- | --- | --- | --- |
| Kondisi kereta | S1 | S2 | Kondisi pintu (S1 ∙ S2) |
| 1 | *true* | *true* | Membuka |
| *2* | *false* | *true* | Menutup |
| *3* | *false* | *false* | Menutup |
| *4* | *true* | *false* | Menutup |
| *5* | *true* | *true* | Membuka |

1. Menggunakan gerbang NOR

Jika kedua inputnya bernilai *false* outputnya akan true maka kedua sensor menunjukan warna merah. Jika hasil dari S1 dan S2 menggunakan gerbang NOR bernilai *false* maka palang pintu akan menutup. Jadi jika kereta melewati sensor pertama, sensor akan bernilai true sehingga outputnya *false* dan palang pintu akan menutup otomatis. Sebaliknya jika kereta bagian belakang meninggalkan sensor, outputnya *true* maka palang pintu akan terbuka. Berikut adalah tabel kebenaran penggunaan gerbang NOR pada palang pintu kereta api

Tabel 4.3 Kondisi palang pintu gerbang NOR

|  |  |  |  |
| --- | --- | --- | --- |
| Kondisi kereta | S1 | S2 | Kondisi pintu (S1 *NOR S2)* |
| 1 | *false* | *false* | Membuka |
| *2* | *true* | *false* | Menutup |
| *3* | *true* | *true* | Menutup |
| *4* | *false* | *true* | Menutup |
| *5* | *false* | *false* | Membuka |

1. Menggunakan gerbang NAND

Jika kedua inputnya bernilai *false* outputnya akan true maka kedua sensor menunjukan warna hijau. Jika hasil dari S1 dan S2 menggunakan gerbang NOR bernilai *true* maka palang pintu akan menutup. Jadi jika kereta melewati sensor pertama, sensor akan bernilai *false* sehingga outputnya *true* dan palang pintu akan menutup otomatis. Sebaliknya jika kereta bagian belakang meninggalkan sensor, outputnya *false* maka palang pintu akan terbuka. Berikut adalah tabel kebenaran penggunaan gerbang NAND pada palang pintu kereta api

Tabel 4.4 Kondisi palang pintu gerbang NAND

|  |  |  |  |
| --- | --- | --- | --- |
| Kondisi kereta | S1 | S2 | Kondisi pintu  (S1 NAND S2) |
| 1 | *true* | *true* | Membuka |
| *2* | *false* | *true* | Menutup |
| *3* | *false* | *false* | Menutup |
| *4* | *true* | *false* | Menutup |
| *5* | *true* | *true* | Membuka |

1. Menggunakan gerbang XOR

Jika kedua inputnya memiliki nilai berbeda maka outputnya akan true maka kedua sensor menunjukan warna merah. Jika hasil dari S1 dan S2 menggunakan gerbang XOR bernilai *true* maka palang pintu akan menutup. Jadi jika kereta melewati sensor pertama, sensor akan bernilai *true* sehingga outputnya *true* dan palang pintu akan menutup otomatis. Sebaliknya jika kereta bagian belakang meninggalkan sensor, outputnya *false* maka palang pintu akan terbuka. Berikut adalah tabel kebenaran penggunaan gerbang XOR pada palang pintu kereta api.

Tabel 4.5 Kondisi palang pintu gerbang XOR

|  |  |  |  |
| --- | --- | --- | --- |
| Kondisi kereta | S1 | S2 | Kondisi pintu  (S1 *S2) OR*  ⊕  (S1 ∙ S2) |
| 1 | *false* | *false* | Membuka |
| *2* | *true* | *false* | Menutup |
| *3* | *true* | *true* | Menutup |
| *4* | *false* | *true* | Menutup |
| *5* | *false* | *false* | Membuka |

1. Menggunakan gerbang XNOR

Jika kedua inputnya memiliki nilai yang sama maka outputnya akan true maka kedua sensor menunjukan warna hijau. Jika hasil dari S1 dan S2 menggunakan gerbang XNOR bernilai *false* maka palang pintu akan menutup. Jadi jika kereta melewati sensor pertama, sensor akan bernilai *false* sehingga outputnya *false* dan palang pintu akan menutup otomatis. Sebaliknya jika kereta bagian belakang meninggalkan sensor, outputnya *true* maka palang pintu akan terbuka. Berikut adalah tabel kebenaran penggunaan gerbang XNOR pada palang pintu kereta api.

Tabel 4.6 Kondisi palang pintu gerbang XNOR

|  |  |  |  |
| --- | --- | --- | --- |
| Kondisi kereta | S1 | S2 | Kondisi pintu  (S1 *S2)’ OR*  (S1’ ∙ S2’) |
| 1 | *true* | *true* | Membuka |
| *2* | *false* | *true* | Menutup |
| *3* | *false* | *false* | Menutup |
| *4* | *true* | *false* | Menutup |
| *5* | *true* | *true* | Membuka |

# IV. KERETA API

The word “data” is plural, not singular. The subscript for the permeability of vacuum µ0 is zero, not a lowercase letter “o.” The term for residual magnetization is “remanence”; the adjective is “remanent”; do not write “remnance” or “remnant.” Use the word “micrometer” instead of “micron.” A graph within a graph is an “inset,” not an “insert.” The word “alternatively” is preferred to the word “alternately” (unless you really mean something that alternates). Use the word “whereas” instead of “while” (unless you are referring to simultaneous events). Do not use the word “essentially” to mean “approximately” or “effectively.” Do not use the word “issue” as a euphemism for “problem.” When compositions are not specified, separate chemical symbols by en-dashes; for example, “NiMn” indicates the intermetallic compound Ni0.5Mn0.5 whereas “Ni–Mn” indicates an alloy of some composition NixMn1-x.

Be aware of the different meanings of the homophones “affect” (usually a verb) and “effect” (usually a noun), “complement” and “compliment,” “discreet” and “discrete,” “principal” (e.g., “principal investigator”) and “principle” (e.g., “principle of measurement”). Do not confuse “imply” and “infer.”

Prefixes such as “non,” “sub,” “micro,” “multi,” and “"ultra” are not independent words; they should be joined to the words they modify, usually without a hyphen. There is no period after the “et” in the Latin abbreviation “*et al.*” (it is also italicized). The abbreviation “i.e.,” means “that is,” and the abbreviation “e.g.,” means “for example” (these abbreviations are not italicized).

# V. Conclusion

A conclusion section is not required. Although a conclusion may review the main points of the paper, do not replicate the abstract as the conclusion. A conclusion might elaborate on the importance of the work or suggest applications and extensions.

# VI. Appendix

Appendixes, if needed, appear before the acknowledgment.

# VII. Acknowledgment

The preferred spelling of the word “acknowledgment” in American English is without an “e” after the “g.” Use the singular heading even if you have many acknowledgments. Avoid expressions such as “One of us (S.B.A.) would like to thank ... .” Instead, write “F. A. Author thanks ... .” Sponsor and financial support acknowledgments are placed in the unnumbered footnote on the first page.

# References

1. G. O. Young, “Synthetic structure of industrial plastics (Book style with paper title and editor),” in *Plastics*, 2nd ed. vol. 3, J. Peters, Ed. New York: McGraw-Hill, 1964, pp. 15–64.
2. W.-K. Chen, *Linear Networks and Systems* (Book style)*.* Belmont, CA: Wadsworth, 1993, pp. 123–135.
3. H. Poor, *An Introduction to Signal Detection and Estimation*. New York: Springer-Verlag, 1985, ch. 4.
4. B. Smith, “An approach to graphs of linear forms (Unpublished work style),” unpublished.
5. E. H. Miller, “A note on reflector arrays (Periodical style—Accepted for publication),” *IEEE Trans. Antennas Propagat.*, to be published.
6. J. Wang, “Fundamentals of erbium-doped fiber amplifiers arrays (Periodical style—Submitted for publication),” *IEEE J. Quantum Electron.*, submitted for publication.
7. C. J. Kaufman, Rocky Mountain Research Lab., Boulder, CO, private communication, May 1995.

# 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.

Kudus, 26 Desember 2021

Ttd (scan atau foto ttd)

Nama dan NIM