BAB II

GERBANG LOGIKA

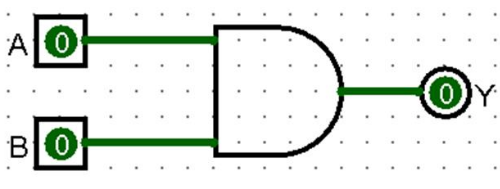
Gerbang logika atau dalam bahasa Inggris disebut dengan *Logic Gate* adalah dasar pembentuk **Sistem Elektronika Digital** yang berfungsi untuk mengubah satu atau beberapa input (masukan) menjadi sebuah sinyal output (keluaran) logis.

Gerbang logika beroperasi berdasarkan sistem bilangan biner yaitu bilangan yang hanya memiliki 2 kode simbol yakni angka 0 dan angka 1 dengan menggunakan teori **Aljabar Boolean**.

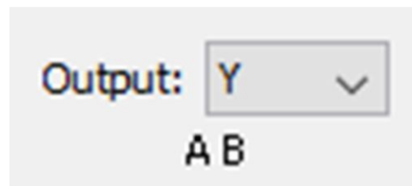
Gerbang logika yang diterapkan dalam sistem elektronika digital pada dasarnya menggunakan komponen–komponen elektronika seperti Integrated Circuit (IC), Dioda, Transistor, Relay, Optik maupun Elemen Mekanikal.

2.1. GERBANG AND

Gerbang AND memerlukan 2 atau lebih masukan (input) untuk menghasilkan hanya 1 keluaran (output). Gerbang AND akan menghasilkan keluaran (output) logika 1 jika semua masukan (input) bernilai logika 1 dan akan menghasilkan keluaran (output) logika 0 jika salah satu dari masukan (input) bernilai logika 0. Simbol yang menandakan operasi gerbang logika AND adalah tanda titik (".") atau tidak memakai tanda sama sekali.



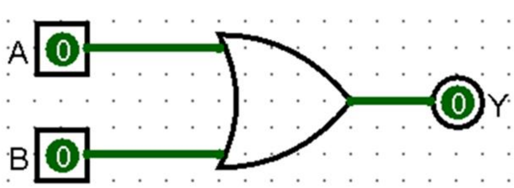
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1



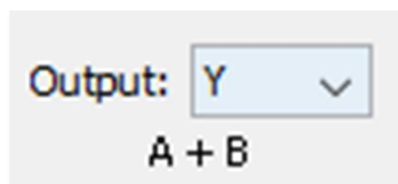
Gambar 2.1. Gerban

2.2. GERBANG OR

Gerbang OR memerlukan 2 atau lebih masukan (input) untuk menghasilkan hanya 1 keluaran (output). Gerbang OR akan menghasilkan keluaran (output) 1 jika salah satu dari masukan (input) bernilai logika 1 dan jika ingin menghasilkan keluaran (output) logika 0, maka semua masukan (input(harus berlogika 0. Simbol yang menandakan operasi gerbang logika OR adalah tanda plus ("+").



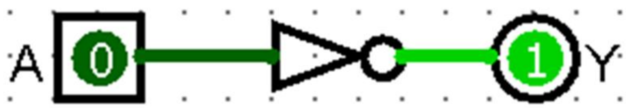
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1



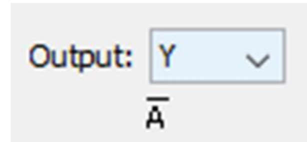
Gambar 2.2. Gerbang **OR**

2.3. GERBANG NOT

Gerbang NOT hanya memerlukan sebuah masukan (input) untuk menghasilkan hanya 1 keluaran (output). Gerbang NOT disebut juga dengan inverter (pembalik) karena menghasilkan keluaran (output) yang berlawanan (kebalikan) dengan masukan atau inputnya. Berarti jika kita ingin mendapatkan keluaran (output) dengan nilai logika 0 maka input atau masukannya harus bernilai logika 1. Gerbang NOT biasanya dilambangkan dengan simbol minus (“-“) di atas variabel inputnya.



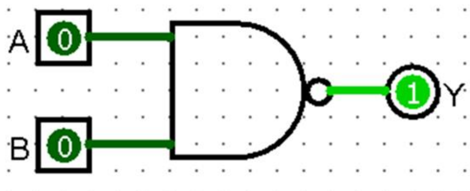
A	Y
0	1
1	0



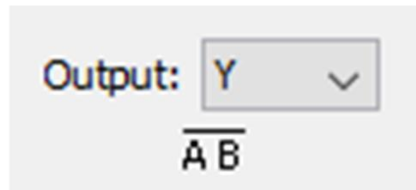
Gambar 2.3. Gerbang **NOT**

2.4. GERBANG NAND

Arti NAND adalah NOT–AND atau BUKAN AND. Gerbang NAND merupakan kombinasi dari gerbang AND dan gerbang NOT yang menghasilkan kebalikan dari keluaran (output) gerbang AND. Gerbang NAND akan menghasilkan keluaran logika 0 apabila semua masukan (input) pada logika 1 dan jika terdapat sebuah input yang bernilai logika 0 maka akan menghasilkan keluaran (output) logika 1.



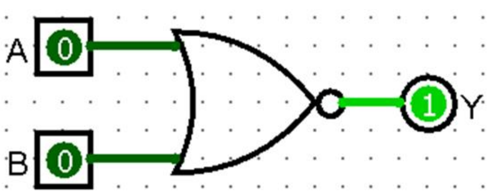
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0



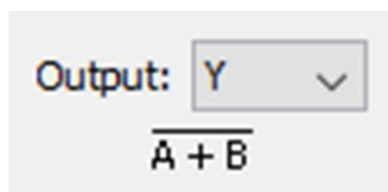
Gambar 2.4. Gerbang **NAND**

2.5. GERBANG NOR

Arti NOR adalah NOT–OR atau BUKAN OR, gerbang NOR merupakan kombinasi dari gerbang OR dan gerbang NOT yang menghasilkan kebalikan dari keluaran (output) gerbang OR. Gerbang NOR akan menghasilkan keluaran logika 0 jika salah satu dari masukan (input) bernilai logika 1 dan jika ingin mendapatkan keluaran logika 1, maka semua masukan (input) harus bernilai logika 0.



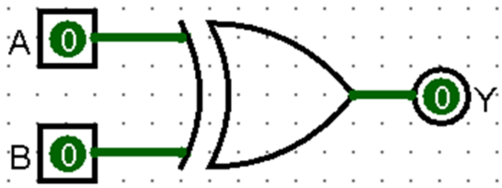
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0



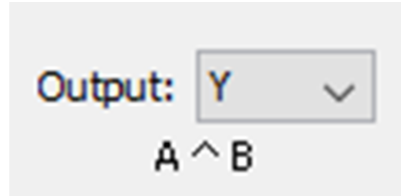
Gambar 2.5. Gerbang **NOR**

2.6. GERBANG XOR

Gerbang XOR adalah singkatan dari Exclusive OR yang terdiri dari 2 masukan (input) dan 1 keluaran (output) logika. Gerbang XOR akan menghasilkan keluaran (output) logika 1 jika semua masukan–masukannya (input) mempunyai nilai logika yang berbeda. Jika nilai logika inputnya sama, maka akan memberikan hasil keluaran logika 0.



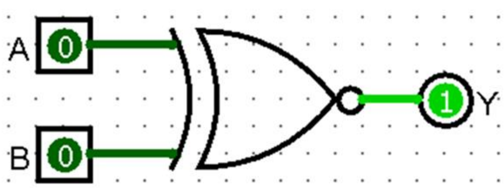
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0



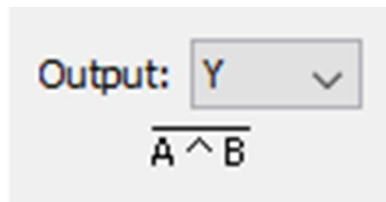
Gambar 2.6. Gerbang **XOR**

2.7. GERBANG XNOR

Seperti halnya gerbang XOR, gerbang XNOR juga terdiri dari 2 masukan (input) dan 1 keluaran (output). XNOR adalah singkatan dari Exlusive NOR dan merupakan kombinasi dari gerbang XOR dan gerbang NOT. Gerbang XNOR akan menghasilkan keluaran (output) logika 1 jika semua masukan atau inputnya bernilai logika yang sama dan akan menghasilkan keluaran (output) logika 0 jika semua masukan atau inputnya bernilai logika yang berbeda. Hal ini merupakan kebalikan dari gerbang XOR (Exlusive OR).



A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1



Gambar 2.7. Gerbang **XNOR**

2.8. SOAL LATIHAN

1. Tunjukkan secara diagram bahwa logika **XOR** dapat diimplementasikan sepenuhnya dengan gerbang **NAND** !
2. Gamabarkan diagram logika untuk setiap elemen dari himpunan **AND, OR, NOT** dapat diimplementasikan dengan menggunakan gerbang **NOR** !
3. Susunlah tabel kebenaran untukgerbang **XOR** dengan 3 masukan !
4. Rancanglah rangkaian logika yang merupakan implementasi dari fungsi Y berikut dengan menggunakan gerbang - gerbang **AND, OR** dan **NOT** !

   $Y = \overline{A}.B.C + \overline{A}.\overline{B}.\overline{C} + A.B.\overline{C}$

5. Rancanglah rangkaian logika yang merupakan implementasi dari fungsi Y berikut dengan menggunakan gerbang –gerbang **AND, OR** dan **NOT**. Jangan berusaha untuk mengubah bentuk persamaannya !

$$Y = A\left(B.C + \overline{B}.\overline{C}\right) + B\left(C.D + \overline{E}\right)$$

6. Apakah kedua fungsi $Y = A.B.C + \overline{A}.B.\overline{C}$ dan $Z = (A \oplus C)B$ tersebut ekuivalen ? Tunjukkan bagaimana Anda mendapatkan penjelasan dan jawabannya !
7. Gunakan hanya gerbang **NOR** untuk merealisasikan operasi **XOR** !
8. Realisasikan fungsi kesamaan dengan 2 masukan menggunakan gerbang – gerbang **NAND** dan **NOR** !

9. Dengan menggunakan tabel kebenaran buktikan bahwa fungsi $\overline{A.B} = \overline{A} + \overline{B}$ dan $A + A.B = A$!
10. Apakah tabel kebenaran dari suatu fungsi dapat digunakan untuk menyederhanakan rancangan gerbang–gerbang logika !

BAB III

ALJABAR BOOLEAN

Aljabar Boolean atau dalam bahasa Inggris disebut Boolean Algebra adalah matematika yang digunakan untuk menganalisis dan menyederhanakan gerbang logika pada rangkaian – rangkaian digital elektronik. Boolean pada dasarnya merupakan tipe data yang hanya terdiri dari dua nilai yaitu **TRUE** dan **FALSE** atau **HIGH** atau **LOW** yang biasanya dilambangkan dengan angka **1** dan angka **0** pada gerbang logika ataupun pemrograman komputer.

Aljabar Boolean ini pertama kali diperkenalkan oleh seorang Matematikawan yang berasal dari Inggris pada tahun 1854. Nama Boolean sendiri diambil dari nama penemunya yaitu George Boole.

3.1. SIFAT– SIFAT KHUSUS ALJABAR BOOLEAN

Gambar 3.1 merangkum sebagian dari sifat–sifat Aljabar Boolean yang dapat diterapkan pada ekspresi logika Boole. **Postulat** (dikenal sebagai **Postulat Huntigton**) merupakan aksioma dasar untuk Aljabar Boolean dan tidak memerlukan pembuktian. Bentuk **Dualisme** ini memungkinkan mengubah bentuk **AND** menjadi **OR** dan sebaliknya bentuk **OR** menjadi **AND**.

	Relasi	Dualisme	Sifat/Teorema
Postulat	$AB = BA$	$A + B\ = B + A$	Sifat Komutatif
	$A(B + C) = AB + AC$	$A + BC = (A + B) + (A + C)$	Sifat Distributif
	$1.A = A$	$0 + A = A$	Sifat Identitas
	$A.\overline{A} = 0$	$A + \overline{A} = 1$	Sifat Komplemen
Teorema	$0.A = 0$	$1 + A = 1$	Teorema nol dan satu
	$A.A = A$	$A + A = A$	Teorema Idempoten
	$A(BC) = (AB)C$	$A + (B + C)$	Teorema Asosiatif
	$\overline{\overline{A}} = A$		Teorema Involusi
	$\overline{AB} = \overline{A} + \overline{B}$	$\overline{A + B} = \overline{AB}$	Teorema DeMorgan
	$AB + \overline{A}C + BC = AB + \overline{A}C$	$(A + B)(\overline{A}C)(B + C) = (A + B)(\overline{A} + C)$	Teorema Konsensus
	$A(A + B) = A$	$A + AB = A$	Teorema absorbsi

Gambar 3.1. Sifat–sifat/Teorema Aljabar Boolean

1. **Sifat Komutatif** menyatakan bahwa urutan kemunculan 2 variabel dalam fungsi **AND** dan **OR** tidak mengakibatkan hasil yang berbeda. Dengan prinsip **Dualisme**, sifat komutatif mempunyai bentuk $\boldsymbol{AND}\ (\boldsymbol{AB = BA})$ dan bentuk $\boldsymbol{OR}\ (\boldsymbol{A + B = B + A})$.
2. **Sifat Distributif** menunjukkan bagaimana variabel didistribusikan melalui operasi **AND**. Karena prinsip **Dualisme** juga maka ada S**ifat Distributf** untuk **OR**.
3. **Teorema Asosiatif** menyatakan bahwa urutan operasi **AND** atau **OR** tidakn mengakibatkan hasil yang berbeda.
4. **Teorema Involusi** menyatakan bahwa komplemen dari komplemen suatu variabel adalah variabel itu sendiri.
5. **Sifat Identitas** menunjukkan bahwa variabel yang di **AND** kan dengan **1** atau di **OR** kan dengan **0**, menghasilkan nilai variabel itu sendiri.
6. **Sifat Komplemen** mengakibatkan bahwa variabel yang dikenakan operasi **AND** terhadap komplemen variabel tersebut, menghasilkan **0** (karena paling tidak pasti ada **1** operan bernilai **0**) dan variabel yang dikenakan operasi **OR** terhadap komplemennya, menghasilkan nilai **1** (karena pasti ada nilai **1** pada operannya.

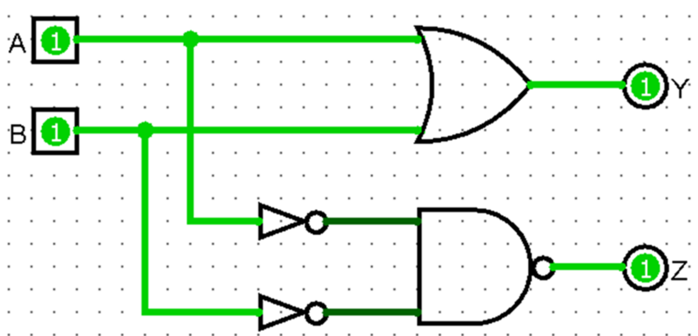
7. **Teorema Nol dan Satu** menyatakan bahwa operasi **AND** antara variabel dengan **0** akan menghasilkan **0** dan operasi **OR** antara variabel dengan **1** akan menghasilkan **1**.
8. **Teorema Idempoten** menyatakan bahwa operasi **AND** atau **OR** antara variabel dengan dirinya sendiri menghasilkan nilai variabel itu sendiri.
9. **Teorema DeMorgan, Konsensus dan Absorbsi** tidak begitu jelas sehingga kita perlu mebuktikannya. Teorema DeMorga dapat dibuktikan dengan induksi yaitu mendaftarkan semua kemungkinan nilai 2 variabel A dan B serta fungsi yang dibuktikan seperti gambar 3.2. Sisi kiri dan kanan dalam ekspresi DeMorgan mempunyai nilai yang sama, inilah buktinya.

A	B	$\overline{AB} = \overline{A} + \overline{B}$		$\overline{A+B} = \overline{A}\,\overline{B}$	
0	0	1	1	1	1
0	1	1	1	0	0
1	0	1	1	0	0
1	1	0	0	0	0

Gambar 3.2. Pembuktian Teorema DeMorgan untuk 2 variabel

Tidak semua gerbang logika dibicarakan secara mendalam karena berdasarkan 3 himpunan gerbang logika yaitu **AND, OR, NOT, NAND** dan **NOR**, satu himpunan dapat disusun dari gerbang – gerbang pada himpunan lainnya.

Sebagai contoh misalnya implementasi **OR** dengan menggunakan himpunan **NAND**. Teorema DeMorgan dapat digunakan untuk menyusun gerbang **OR** dari gerbang **NAND**. Teorema DeMorgan dapat digunakan untuk menyusun gerbang **OR** dari gerbang **NAND** seperti gambar gambar 3.1 dan 3.2.

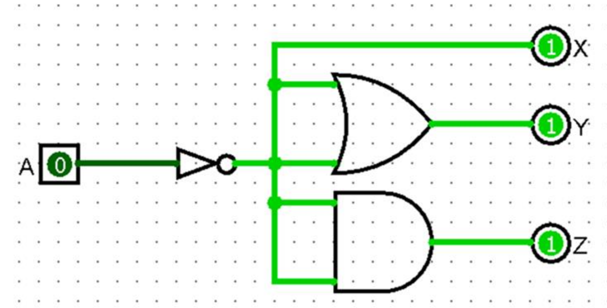


A	B	Y	Z
0	0	0	0
0	1	1	1
1	0	1	1
1	1	1	1

Teorema Involusi : $A + B = \overline{\overline{A + B}}$ *Teorema DeMorgan* : $A + B = \overline{\overline{A}\,\overline{B}}$

Gambar 3.3. Implementasi 2 gerbang **NOT** dan 1 gerbang **NAND** menjadi 1 gerbang **OR**

Ekuivalensi di antara fungsi – fungsi logika menjadi penting dalam praktek, karena suatu jenis gerbang logika kemungkinan mempunyai karakteristik yang lebih baik daripada yang lainnya.



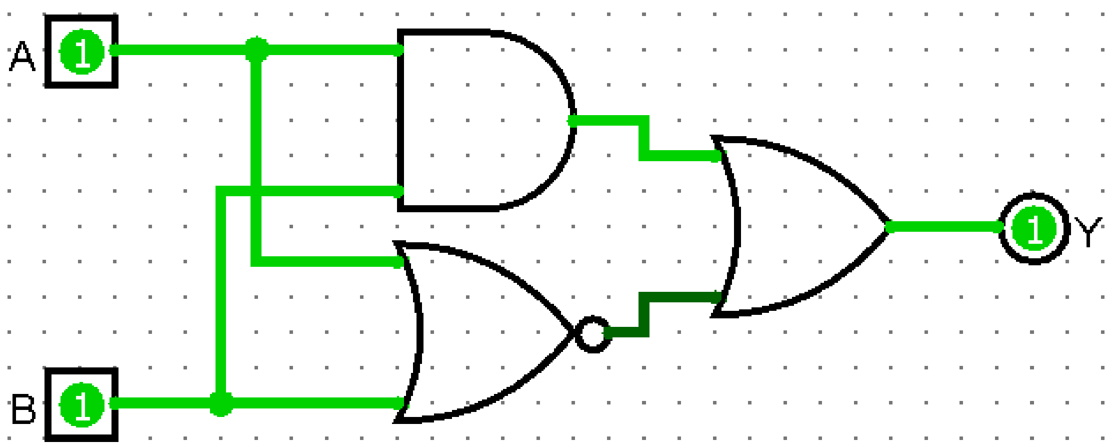
A	X	Y	Z
0	1	1	1
1	0	0	0

$Teorema\ Idempoten:\ \overline{A} = \overline{A} + \overline{A}$ $\quad Teorema\ DeMorgan:\ \overline{A} = \overline{A.A}$

Gambar 3.4. Implementasi gerbang **OR** atau **AND** dengan 2 inputan yang sama

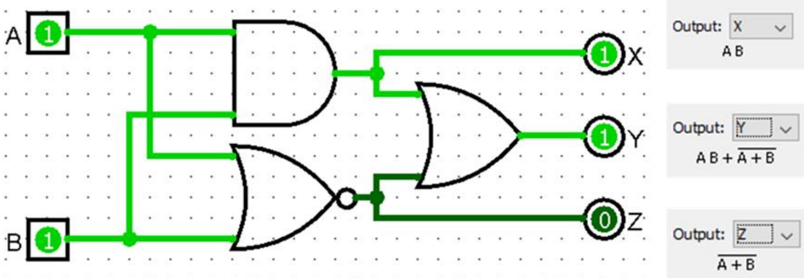
3.2. SOAL–SOAL LATIHAN

1. Sederhanakanlah rangkaian gerbang berikut ini :



Gambar 3.5. Soal latihan 1

Untuk rangkaian gerbang diatas terdiri dari gerbang AND, gerbang NOR dan gerbang OR. Ekspresi untuk AND adalah $A.B$ dan ekspresi untuk gerbang NOR adalah $\overline{A+B}$. Dari kedua ekspresi ini juga memiliki input yang terpisah ke gerbang OR yang didefinisikan sebagai $A+B$. Jadi ekspresi keluaran akhir adalah sebagai berikut:

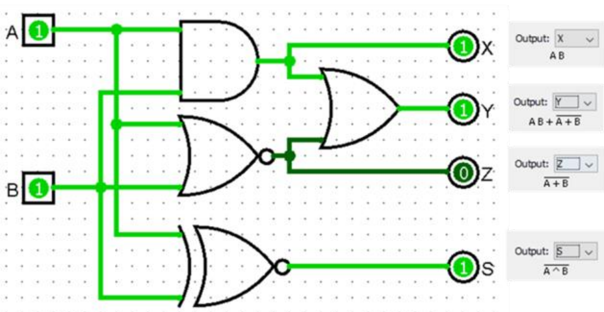


A	B	X	Y	Z
0	0	0	1	1
0	1	0	0	0
1	0	0	0	0
1	1	1	1	0

Gambar 3.6. Penjelasan soal latihan 1

Sehingga output (Y) dari gerbang logika tersebut adalah $\boldsymbol{Y = AB + \overline{A + B}}$, dengan menerapkan teorema De Morgan didapat $\boldsymbol{Y = AB + \overline{A}\,\overline{B}}$ atau $\boldsymbol{Y = \overline{A \oplus B}}$ yang merupakan persamaan dari gerbang XNOR.

Jadi rangkian gerbang diatas bila disederhanakan menjadi sebuah gerbang XNOR.



A	B	X	Y	Z	S
0	0	0	1	1	1
0	1	0	0	0	0
1	0	0	0	0	0
1	1	1	1	0	1

Gambar 3.7. Pembuktian hasil akhir soal latihan 1

2. Sederhanakanlah persamaan logika $\boldsymbol{Y = \overline{A}B\overline{C} + AB\overline{C}}$

   $Y = \overline{A}B\overline{C} + AB\overline{C}$

   Sifat distributif :

   $Y = (\overline{A} + A)B\overline{C}$

   $\boldsymbol{Y = B\overline{C}}$

3. Tentukanlah persamaan logika dari fungsi keluaran pada tabel berikut ini:

A	B	C	Y
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

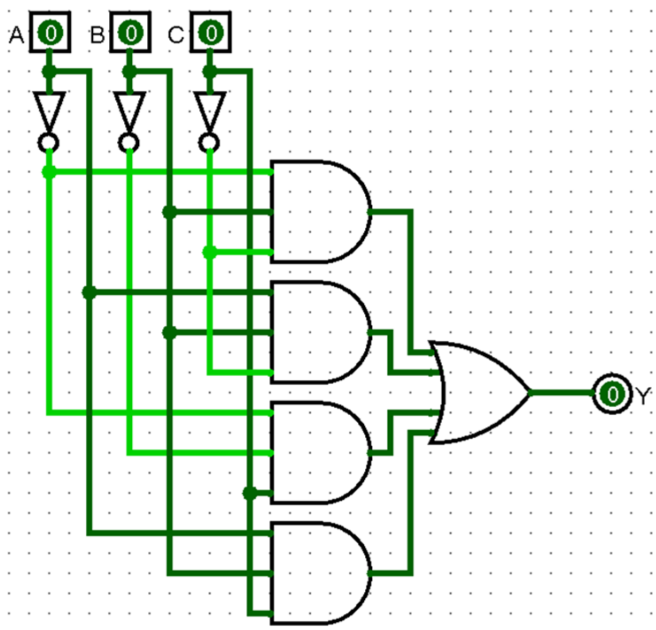
Gambar 3.8. Tabel kebenaran soal latihan 3

Langkah 1, buatlah persamaan dari fungsi pada tabel diatas.

A	B	C	Y	
0	0	0	0	
0	0	1	1	$\overline{A}\,\overline{B}C$
0	1	0	1	$\overline{A}B\overline{C}$
0	1	1	0	
1	0	0	0	
1	0	1	0	
1	1	0	1	$AB\overline{C}$
1	1	1	1	ABC

Gambar 3.9. Tabel pengembangan

Dari tabel diatas kita dapat membuat gambaran mengenai rangkaian gerbang–gerbang logika dan persamaan dari rangkaian gerbang–gerbang logika tersebut.



A	B	C	Y
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

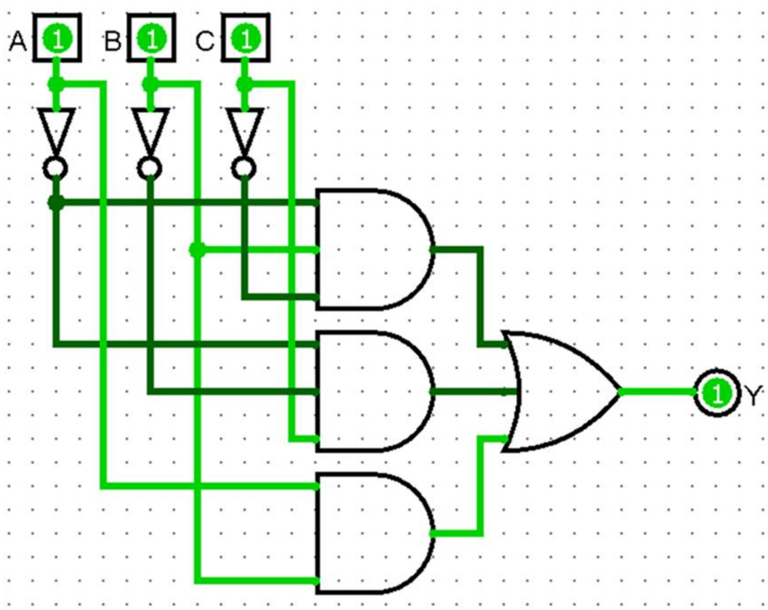
Gambar 3.10. Implementasi tabel kebenaran latihan 3

Langkah 2, menyederhanakan persamaan logika dengan menggunakan Aljabar Boolean.

$$Y = \overline{A}B\overline{C} + \overline{A}\,\overline{B}C + AB\left(\overline{C} + C\right)$$

$$Y = \overline{A}B\overline{C} + \overline{A}\,\overline{B}C + AB$$

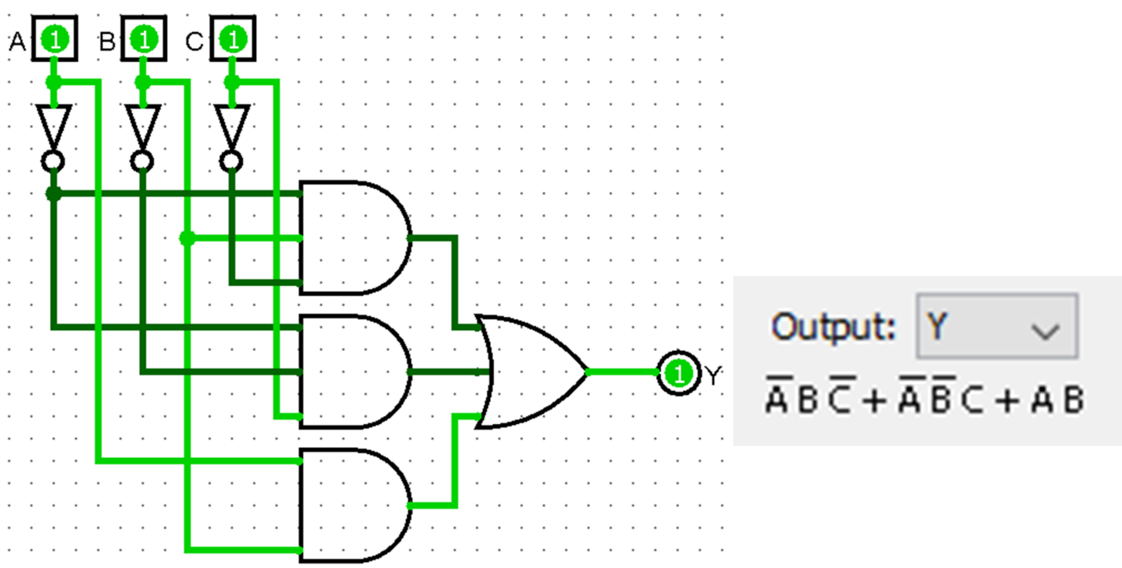
Langkah 3, membuat tabel kebenaran untuk menguji dari penyederhaan persamaan logika tersebut.



A	B	C	Y
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Gambar 3.11. Hasil akhir tabel kebenaran soal latihan 3

Langkah 4, membuat hasil rancangan akhir dari persamaan logika kedalam gerbang–gerbang logika.



Gambar 4.12. Hasil akhir rancangan penyederhanaan gerbang soal latihan 3

4. Selesaikanlah persamaan logika berikut ini :

   a. $Y = A + B(A + B) + A(\bar{A} + B)$

   b. $Y = \bar{A}(BC + AB + B\bar{A})$

   c. $Y = ABC + AB + A$

   d. $Y = (\bar{A} + AB)(\bar{A}B)$

e. $Y = BC + AD + ABCD + ADC + \bar{A}$

5. Selesaikanlah persamaan logika berikut ini :

a. $Z = \bar{A}\bar{B}C + \bar{A}BC + A\bar{B}$

b. $Z = AB + \bar{A}C + BC$

c. $Z = (A + B)(\bar{A} + C)(B + C)$