

# Rancangan dan Implementasi Dekoder Biner (BCD) ke 7-Segmen (*7-Segment Display*) Menggunakan Aljabar Boolean pada Permainan Minecraft

Arghawisesa Dwinanda Arham - 13524100

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jalan Ganesha 10 Bandung

E-mail: [arghawisesa@gmail.com](mailto:arghawisesa@gmail.com) , [13524100@std.stei.itb.ac.id](mailto:13524100@std.stei.itb.ac.id)

**Abstract**— Dekoder biner (BCD) adalah rangkaian logika yang menerima input berupa kode biner dan mengaktifkan salah satu outputnya sesuai dengan urutan biner tersebut. Dekoder biner biasanya digunakan untuk mengubah sinyal biner menjadi bentuk yang dapat dipahami atau digunakan oleh perangkat lain, seperti *seven segment display*. Implementasi dekoder biner ke *seven segment display* dapat diterapkan dalam permainan Minecraft menggunakan sistem Redstone yang terdapat dalam permainan tersebut.

**Keywords**— *Aljabar Boolean, Logika, Minecraft, Redstone, Dekoder Biner, 7 Segment Display*

## I. PENDAHULUAN

Visualisasi informasi merupakan hal yang tidak dapat terpisahkan dari kehidupan sehari-hari. Mulai dari kalkulator, jam digital, timer digital, serta panel layar lainnya, menerapkan visualisasi dalam menyampaikan informasi yang dimilikinya, dalam hal ini adalah menggunakan *seven segment display*. Di balik tampilan numerik yang dimiliki *seven segment display*, terdapat sirkuit logika yang dapat menerjemahkan sinyal elektronik menjadi suatu hal yang dapat dimengerti oleh manusia.

Di sisi lain, permainan pada zaman sekarang telah berevolusi dan kegunaannya pun dapat sangat beragam. Seperti pada permainan Minecraft, yaitu sebuah *sandbox*, di mana pemain dapat membuat apa pun sesuai dengan kreativitas pemainnya. Sebuah sekolah pada India bernama Sat Paul Mittal School bahkan menggunakan Minecraft: Education Edition untuk mengajarkan peradaban Mesir. Hal ini menunjukkan bahwa Minecraft yang semula hanya digunakan sebagai rekreasi, dapat digunakan juga untuk hal lainnya. Dalam Minecraft, terdapat sistem Redstone, yaitu sebuah sistem 'listrik' di mana pemain dapat membuat rangkaian untuk hal-hal seperti menyala-lam-pu, menggerakkan piston, dan lain-lain. Salah satu yang memungkinkan untuk dilakukan adalah membuat gerbang-gerbang logika dasar seperti AND, OR, dan NOT, yang merupakan fondasi dari semua komputasi digital. Hal ini memungkinkan juga untuk membuat implementasi dekoder biner ke *seven segment display*.

## II. LANDASAN TEORI

### A. Aljabar Boolean

Aljabar Boolean adalah sebuah cabang matematika di mana dalam penerapannya, Aljabar Boolean hanya mengenal dua nilai, yaitu 0 (*false*) dan 1 (*true*). Aljabar Boolean pertama kali ditemukan oleh George Boole pada tahun 1854. Aplikasi dari Aljabar Boolean dalam kehidupan sehari-hari adalah rangkaian pensaklaran, digital, serta *Integrated Circuit* pada komputer. Terdapat beberapa hukum-hukum dalam Aljabar Boolean, berikut rinciannya (negasi ditandai dengan simbol '):

1. Hukum Identitas:
  - i.  $a + 0 = a$
  - ii.  $a \cdot 0 = 0$
2. Hukum Komplemen:
  - i.  $a + a' = 1$
  - ii.  $a \cdot a' = 0$
3. Hukum Involusi:
  - i.  $(a')' = a$
4. Hukum Komutatif:
  - i.  $a + b = b + a$
  - ii.  $a \cdot b = b \cdot a$
5. Hukum Distributif:
  - i.  $a + (b \cdot c) = (a + b) \cdot (a + c)$
  - ii.  $a \cdot (b + c) = a \cdot b + a \cdot c$
6. Hukum 0/1:
  - i.  $0' = 1$
  - ii.  $1' = 0$
7. Hukum Idempoten:
  - i.  $a + a = a$
  - ii.  $a \cdot a = a$
8. Hukum Dominasi:

- i.  $a \cdot 0 = 0$
  - ii.  $a + 1 = 1$
9. Hukum Penyerapan:
- i.  $a + a \cdot b = a$
  - ii.  $a \cdot (a + b) = a$

10. Hukum Asosiatif:

- i.  $a + (b + c) = (a + b) + c$
- ii.  $a \cdot (b \cdot c) = (a \cdot b) \cdot c$

11. Hukum De Morgan:

- i.  $(a + b)' = a' \cdot b'$
- ii.  $(a \cdot b)' = a' + b'$

Di dalam Aljabar Boolean, terdapat sebuah fungsi boolean di mana peubah di dalamnya disebut dengan literal. Sebagai contoh:

$$h(x, y, z) = xyz' \quad (1)$$

Fungsi di atas menunjukkan terdapat tiga buah literal (x, y, dan z').

Bentuk Kanonik dalam Aljabar Boolean menyatakan bahwa suatu fungsi dapat disajikan dalam dua bentuk yang berbeda. Diantaranya adalah SOP (Sum of Product) dan POS (Product of Sum). SOP mengandung penjumlahan dari *minterm* (literal dalam bentuk hasil kali). Sedangkan POS mengandung perkalian dari *maxterm* (literal dalam bentuk hasil jumlah).

Fungsi dalam Aljabar Boolean dapat disederhanakan. Ini berarti mencari fungsi yang equivalen namun jumlah literal ataupun jumlah operasi yang lebih sedikit. Terdapat beberapa metode dalam menyederhanakan fungsi Aljabar Boolean, salah satunya adalah dengan Peta Karnaugh. Peta Karnaugh atau K-Map adalah sebuah diagram yang berbentuk kotak-kotak yang bersisian, ditemukan oleh Maurice Karnaugh pada tahun 1953. Peta Karnaugh direpresentasikan dalam *minterm*.

Cara menyederhanakan dalam Peta Karnaugh adalah dengan mengisikan nilai 1 pada kotak yang menyatakan *minterm*, sisanya dengan 0 (namun dalam beberapa kasus diisikan dengan X atau *don't care*). Kemudian dilakukan penggabungan pada kotak-kotak yang bernilai 1 dan saling bersisian dengan urutan pengelompokan dilakukan pada pasangan yang oktet, kemudian kuad, dan terakhir duplet.

w	x	y	z	00	01	11	10
0	0	0	0	0	1	1	0
1	0	1	0	0	1	1	1
2	1	0	0	1	0	0	1
3	1	0	1	1	0	1	0
4	1	1	0	0	1	1	1
5	1	1	0	1	0	0	0
6	1	1	1	0	1	0	0

Hasil penyederhanaan:  $f(w, x, y, z) = z + xy + wx'y'$

**Gambar 2.1** Contohn penyederhanaan dengan Peta Karnaugh

(Sumber: Aljabar Boolean (Bag. 2) oleh Rinaldi Munir)

Pada beberapa kasus, terdapat kondisi *don't care*, yaitu sebuah kondisi di mana nilai peubah tidak diperhitungkan. Ketika ingin melakukan penyederhanaan, X dapat dilibatkan jika terdapat 1 diantaranya. Hal ini dapat dilakukan agar memiliki pengelompokan semaksimal mungkin (yaitu dari oktet - kuad - duplet).

**Contoh:** Minimisasi fungsi Boolean berikut (dalam bentuk baku SOP dan bentuk baku POS):  $f(w, x, y, z) = \Sigma (1, 3, 7, 11, 15)$

dengan kondisi *don't care* adalah  $d(w, x, y, z) = \Sigma (0, 2, 5)$ .

**Penyelesaian:**

w	x	y	z	00	01	11	10
0	0	0	0	1	1	X	X
1	0	1	0	0	X	1	0
2	1	0	0	0	0	0	1
3	1	0	1	1	0	1	0
4	1	1	0	0	1	1	1
5	1	1	0	1	0	0	0
6	1	1	1	0	1	0	0

Hasil penyederhanaan:

SOP:  $f(w, x, y, z) = y' + w'z$

(kelompok garis penuh)

POS:  $f(w, x, y, z) = z(w' + y)$

(kelompok garis putus-putus)

**Gambar 2.2** Contoh penyederhanaan dengan Peta Karnaugh yang terdapat *don't care*

(Sumber: Aljabar Boolean (Bag. 2) oleh Rinaldi Munir)

Fungsi dalam Aljabar Boolean juga dapat direpresentasikan dalam bentuk lain, yaitu rangkaian logika. Berikut merupakan bentuk dan artinya :

No.	FUNGSI	SIMBOL	TABEL
1	AND	A  B → F	A   B   F 0   0   0 0   1   0 1   0   0 1   1   1
2	OR	A  B → F	A   B   F 0   0   0 0   1   1 1   0   1 1   1   1
3	NOT	A  → F	A   F 0   1 1   0
4	NAND	A  B → F	A   B   F 0   0   1 0   1   1 1   0   1 1   1   0
5	NOR	A  B → F	A   B   F 0   0   1 0   1   0 1   0   0 1   1   0
6	X-OR	A  B → F	A   B   F 0   0   0 0   1   1 1   0   1 1   1   0
7	X-NOR	A  B → F	A   B   F 0   0   1 0   1   0 1   0   0 1   1   1

**Gambar 2.3** Tabel dari simbol dan tabel kebenaran

(Sumber:

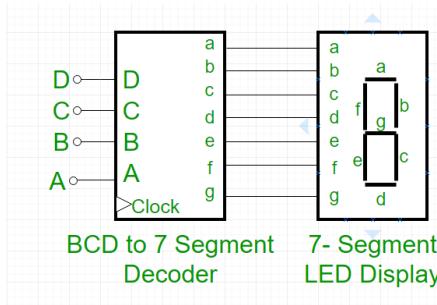
<https://ekomartantoh.net/artikel/2023/09/30/rangkaian-logika/>

**B. Dekoder Biner (BCD) ke Seven Segment Display**

BCD (*Binary Coded Decimal*) adalah sebuah desimal yang dikodekan menjadi bentuk biner. Dekoder BCD ke *seven segment display* adalah sebuah rangkaian yang berfungsi untuk mengonversi BCD menjadi data tampilan yang digunakan untuk penampil pada *seven segment display*. Hal ini memanfaatkan ilmu dari Aljabar Boolean.

*Seven segment display* adalah sebuah visualisasi dalam menampilkan informasi (angka) di mana dalam prosesnya akan dibagi tujuh segmen berbentuk garis dan akan dinyalakan sesuai dengan angka yang ingin ditampilkan. Masukkan pada *seven*

*segment display* harus menggunakan dekoder BCD ke *seven segment display*.



**Gambar 2.4** Ilustrasi proses dari dekoder ke *seven segment display*

(Sumber: <https://www.geeksforgeeks.org/digital-logic/bcd-to-7-segment-decoder/>)

### C. Minecraft

Minecraft adalah sebuah permainan yang dibuat oleh Markus "Notch" Persson pada tahun 2009. Permainan ini menggunakan unsur simplistik dan berbasis balok. ini termasuk ke dalam *sandbox*, yang berarti ini tidak memiliki batasan dalam pemain untuk melakukan segala hal, menuang kreativitas pemain ke dalam .

Di dalam Minecraft, terdapat sistem Redstone yang memungkinkan pemain untuk membuat seperti rangkaian listrik dalam tersebut. Terdapat beberapa komponen atau item yang dapat digunakan untuk sistem Redstone ini, yaitu sebagai berikut:

1. Block of Redstone: Memberikan power yang konstan
2. Button: Membuat denyut (untuk mengaktifkan sesuatu)
3. Lever: Untuk membuat sirkuit ON atau OFF
4. Redstone Torch: Memberikan power dan dapat digunakan untuk membalikkan sinyal
5. Redstone Dust: Mengalirkan power
6. Redstone Repeater: Mengalirkan power, memperkuat sinyal dari Redstone Dust, dapat membuat sinyal delay maupun mengubah arah sinyal.
7. Redstone Comparator: Membandingkan atau mengurangi dua sinyal
8. Piston: Menggerakkan balok
9. Redstone Lamp: Memancarkan cahaya
10. Command Block: Mengeksekusi command yang diberikan pemain
11. Dan beberapa komponen lainnya



**Gambar 2.5** Item yang dapat digunakan untuk Redstone System

(Sumber: Arsip pengguna)



**Gambar 2.6** Item yang dapat digunakan untuk Redstone System

(Sumber: Arsip pengguna)

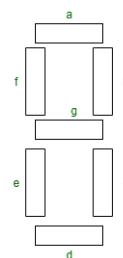
Dalam makalah ini, digunakan Minecraft 1.21.6 dengan *texture pack* Mattpack. Pemilihan *texture pack* ini dilakukan karena pada *texture pack* tersebut dapat dilihat kekuatan Redstone Dust sehingga dapat memudahkan dalam membuat rangkaian nantinya.

## III. PERHITUNGAN DAN IMPLEMENTASI KE DALAM MINECRAFT

Langkah yang akan dilakukan adalah mencari fungsi dari setiap segmen pada *seven segment display* berdasarkan input (4 bit). Setiap fungsi tersebut akan berguna untuk menentukan rangkaian pada Redstone Minecraft per segmen nantinya. Setelah fungsi telah ditentukan, maka langkah selanjutnya adalah menimplementasikannya dalam Minecraft.

### A. Perhitungan

Pertama-tama yang dilakukan adalah membuat *Truth Table* untuk dekoder BCD ke *seven segment*. Tata letak segmen adalah sebagai berikut:



**Gambar 3.1** Letak segmen pada *seven segment display*

(Sumber: <https://www.geeksforgeeks.org/digital-logic/seven-segment-displays/>)

Kemudian didapatkan *Truth Table* dengan input biner 4 bit (A,B,C, dan D) dan output *seven segment display* (a, b, c, d, e, f, dan g) sebagai berikut (*don't care* direpresentasikan dengan X):

Angka Desimal	Input				Output						
	A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	0	1	1	
10	1	0	1	0	X	X	X	X	X	X	X
11	1	0	1	1	X	X	X	X	X	X	X
12	1	1	0	0	X	X	X	X	X	X	X
13	1	1	0	1	X	X	X	X	X	X	X
14	1	1	1	0	X	X	X	X	X	X	X
15	1	1	1	1	X	X	X	X	X	X	X

Tabel 2.1 Tabel kebenaran input biner dan output *seven segment display*

Kemudian didapatkan fungsi-fungsi sebagai berikut.

$$\begin{aligned} a &= A'B'C'D' + A'B'CD' + A'B'CD + A'BC'D \\ &\quad + A'BCD' + A'BCD + AB'C'D' \\ &\quad + AB'C'D \end{aligned} \quad (2)$$

$$\begin{aligned} b &= A'B'C'D' + A'B'C'D + A'B'CD' + \\ &\quad A'B'CD + A'BC'D' + A'BCD + AB'C'D' + \\ &\quad AB'C'D \end{aligned} \quad (3)$$

$$\begin{aligned} c &= A'B'C'D' + A'B'C'D + A'B'CD + A'BC'D' \\ &\quad + A'BC'D + A'BCD' + A'BCD \\ &\quad + AB'C'D' + AB'C'D \end{aligned} \quad (4)$$

$$\begin{aligned} d &= A'B'C'D' + A'B'CD' + A'B'CD + A'BC'D \\ &\quad + A'BCD' + AB'C'D' + AB'C'D \end{aligned} \quad (5)$$

$$e = A'B'C'D' + A'B'CD' + A'BCD' + AB'C'D' \quad (6)$$

$$\begin{aligned} f &= A'B'C'D' + A'BC'D' + A'BC'D + A'BCD' \\ &\quad + AB'C'D' + AB'C'D \end{aligned} \quad (7)$$

$$\begin{aligned} g &= A'B'CD' + A'B'CD + A'BC'D' + A'BC'D \\ &\quad + A'BCD' + AB'C'D' + AB'C'D \end{aligned} \quad (8)$$

Kemudian dilakukan penyederhanaan setiap fungsi. Penyederhanaan akan dilakukan dengan Peta Karnaugh. Berikut adalah hasil Peta Karnaugh serta hasil penyederhananya:

AB	CD			
	00	01	11	10
00	1	0	1	1
01	0	1	1	1
11	X	X	X	X
10	1	1	X	X

Tabel 2.2 Peta Karnaugh & Penyederhanaan persamaan (2)

AB	CD			
	00	01	11	10
00	1	1	1	1
01	1	0	1	0
11	X	X	X	X
10	1	1	X	X

Tabel 2.3 Peta Karnaugh & Penyederhanaan Persamaan (3)

AB	CD			
	00	01	11	10
00	1	1	1	0
01	1	1	1	1
11	X	X	X	X
10	1	1	X	X

Tabel 2.4 Peta Karnaugh & Penyederhanaan Persamaan (4)

AB	CD			
	00	01	11	10
00	1	1	1	1
01	0	1	0	1
11	X	X	X	X
10	1	1	X	X

Tabel 2.5 Peta Karnaugh & Penyederhanaan Persamaan (5)

AB	CD			
	00	01	11	10
00	1	0	1	1
01	0	0	0	1
11	X	X	X	X
10	1	0	X	X

Tabel 2.6 Peta Karnaugh & Penyederhanaan Persamaan (6)

AB	CD			
	00	01	11	10
00	1	0	0	0
01	1	1	0	1
11	X	X	X	X
10	1	1	X	X

Tabel 2.7 Peta Karnaugh & Penyederhanaan Persamaan (7)

AB	CD			
	00	01	11	10
00	0	0	1	1
01	1	1	0	1
11	X	X	X	X
10	1	1	X	X

Tabel 2.8 Peta Karnaugh & Penyederhanaan Persamaan (8)

Maka didapatkan fungsi dengan yang paling sederhana adalah sebagai berikut:

$$a = A + C + BD + B'D' \quad (9)$$

$$b = B' + CD + C'D' \quad (10)$$

$$c = B + D + C' \quad (11)$$

$$d = A + B'C + B'D' + CD' + BC'D \quad (12)$$

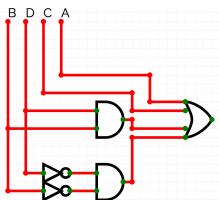
$$e = B'D' + CD' \quad (13)$$

$$f = A + BC' + BD' + C'D' \quad (14)$$

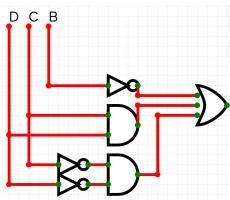
$$g = A + BC' + B'C + CD' \quad (15)$$

Dari hal ini, maka didapatkan bahwa fungsi untuk setiap segmen, ini seperti sebuah syarat bahwa segmen tersebut akan menyala jika sesuai dengan fungsi Aljabar Boolean yang telah ditentukan di atas.

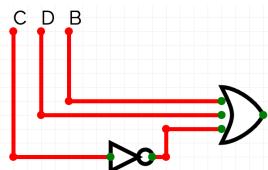
Berikut divisualisasikan juga fungsi dalam bentuk rangkaian logika agar memudahkan dalam implementasi ke dalam Minecraft:



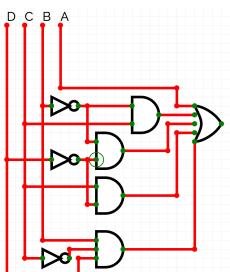
**Gambar 3.2** Rangkaian logika fungsi (9)  
(Sumber: Arsip pengguna)



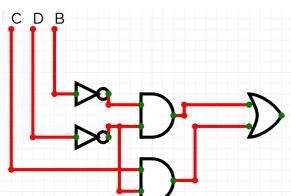
**Gambar 3.3** Rangkaian logika fungsi (10)  
(Sumber: Arsip pengguna)



**Gambar 3.4** Rangkaian logika fungsi (11)  
(Sumber: Arsip pengguna)

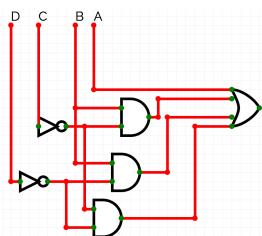


**Gambar 3.5** Rangkaian logika fungsi (12)  
(Sumber: Arsip pengguna)

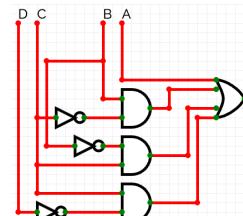


**Gambar 3.6** Rangkaian logika fungsi (13)

(Sumber: Arsip pengguna)



**Gambar 3.7** Rangkaian logika fungsi (14)  
(Sumber: Arsip pengguna)



**Gambar 3.8** Rangkaian logika fungsi (15)  
(Sumber: Arsip pengguna)

#### B. Implementasi dalam Minecraft

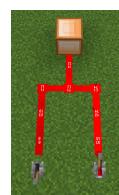
Di dalam permainan Minecraft, terdapat beberapa skema gate yang dapat dilakukan dengan Redstone yang akan sangat berpengaruh dan memudahkan dalam implementasi dekoder BCD ke *seven segment display* kali ini, yaitu:

1. NOT Gate: Dibuat dengan item Redstone Dust, Redstone Torch, dan Block yang dibebaskan



**Gambar 3.9** NOT Gate pada Minecraft (kiri jika input 0 maka output 1 dan kanan jika input 1 maka output 0). Lever sebagai item dari contoh inputnya.  
(Sumber: Arsip pengguna)

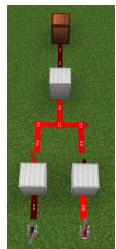
2. OR Gate: Dibuat dengan item Redstone Dust



**Gambar 4.1** AND Gate pada Minecraft dengan contoh input pertama 0 dan input kedua 1 menghasilkan output 1. Lever sebagai item dari contoh input dan Redstone Lamp sebagai item dari contoh output  
(Sumber: Arsip pengguna)

3. AND Gate: Dibuat dengan Redstone Dust, Redstone Torch, dan Block yang dibebaskan. AND Gate dibuat

dengan menerapkan hukum de morgan, yaitu  $(a \cdot b)' = a' + b'$  yang dapat diubah menjadi  $a \cdot b = '(a' + b')$ .



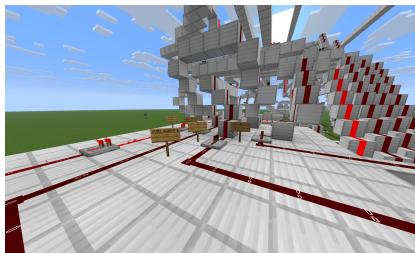
**Gambar 4.2** AND Gate pada Minecraft dengan contoh input pertama 0 dan input kedua 1 menghasilkan output 0. Lever sebagai item dari contoh inputnya dan Redstone Lamp sebagai item dari contoh outputnya.  
(Sumber: Arsip pengguna)

Untuk *seven segment display*, digunakan Redstone Lamp, Redstone Dust, Redstone Repeater, Target block, dan Glass untuk membuat *seven segment display* bekerja. Kemudian untuk ukuran *seven segment display* yang digunakan adalah 10x20 blok, untuk segmen horizontal memiliki ukuran 6x2 blok dan untuk segmen vertikal memiliki ukuran 2x7 blok.

Kemudian dengan menggabungkan skema gate yang telah dibuat dan rangkaian logika yang telah dibuat sebelumnya, maka didapatkan rangkaian Redstone sebagai berikut:



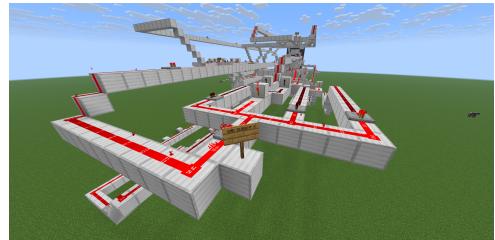
**Gambar 4.3** Rangkaian Redstone keseluruhan  
(Sumber: Arsip pengguna)



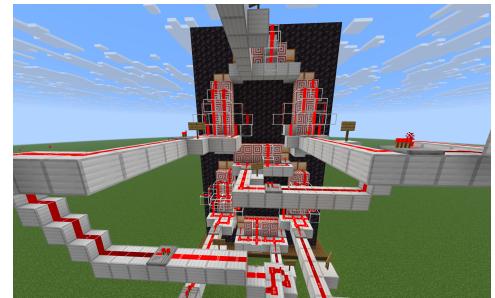
**Gambar 4.4** Sebagian rangkaian  
(Sumber: Arsip pengguna)



**Gambar 4.5** Sebagian rangkaian  
(Sumber: Arsip pengguna)



**Gambar 4.5** Sebagian rangkaian  
(Sumber: Arsip pengguna)



**Gambar 4.6** Sebagian rangkaian *seven segment display*  
(Sumber: Arsip pengguna)



**Gambar 4.7** Tampilan depan, Lever sebagai input (4 bit biner) dan Redstone Lamp sebagai *seven segment display*  
(Sumber: Arsip pengguna)

Secara keseluruhan, penulis menggunakan Redstone Dust untuk menyambungkan bagian-bagian, serta Redstone Repeater untuk memperkuat aliran 'listrik' yang ada karena semakin panjang Redstone Dust yang digunakan, maka aliran 'listrik' pun akan semakin lemah.

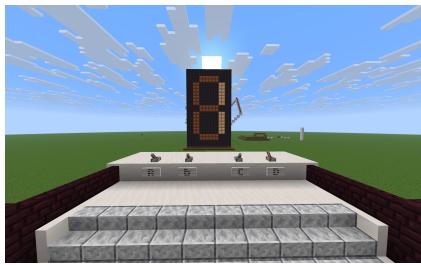
Kemudian berikut adalah hasil dari percobaan setiap angka dari 0 sampai dengan 9:

1. Input = 0 (0000)



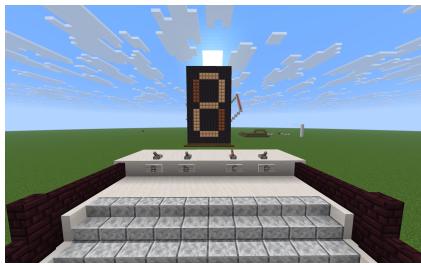
**Gambar 4.8** Tampilan input = 0 atau 0000  
(Sumber: Arsip pengguna)

2. Input = 1 (0001)



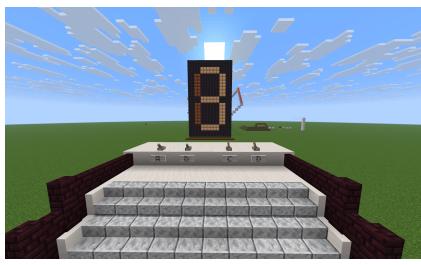
**Gambar 4.9** Tampilan input = 1 atau 0001  
(Sumber: Arsip pengguna)

3. Input = 2 (0010)



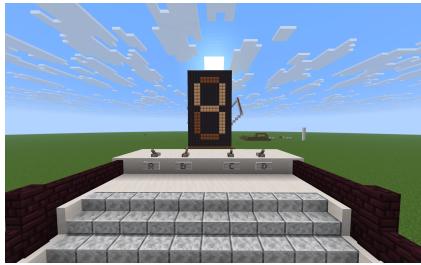
**Gambar 5.1** Tampilan input = 2 atau 0010  
(Sumber: Arsip pengguna)

4. Input = 3 (0011)



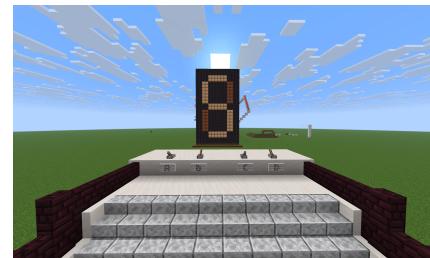
**Gambar 5.2** Tampilan input = 3 atau 0011  
(Sumber: Arsip pengguna)

5. Input = 4 (0100)



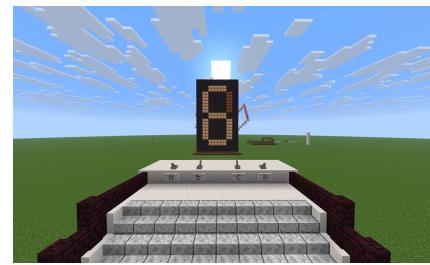
**Gambar 5.3** Tampilan input = 4 atau 0100  
(Sumber: Arsip pengguna)

6. Input = 5 (0101)



**Gambar 5.4** Tampilan input = 5 atau 0101  
(Sumber: Arsip pengguna)

7. Input = 6 (0110)



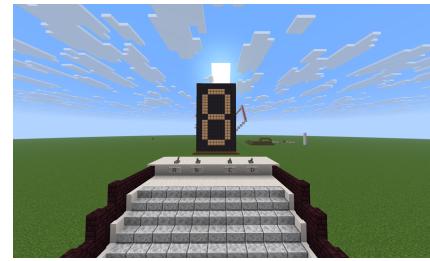
**Gambar 5.5** Tampilan input = 6 atau 0110  
(Sumber: Arsip pengguna)

8. Input = 7 (0111)



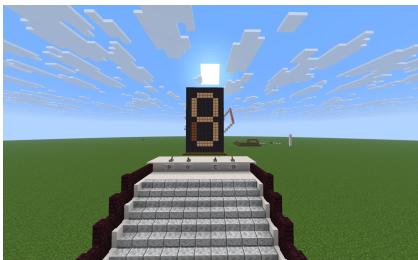
**Gambar 5.6** Tampilan input = 7 atau 0111  
(Sumber: Arsip pengguna)

9. Input = 8 (1000)



**Gambar 5.7** Tampilan input = 8 atau 1000  
(Sumber: Arsip pengguna)

10. Input = 9 (0101)



**Gambar 5.8** Tampilan input = 9 atau 0101  
(Sumber: Arsip pengguna)

Ketika melakukan proses input, penulis merasakan terdapat jeda dalam pergantian angka pada *seven segment display*. Hal ini mungkin terjadi karena pada Redstone Repeater, terdapat *delay* apabila

#### IV. KESIMPULAN

Implementasi dekoder biner (BCD) ke 7 segmen (*seven segment display*) dapat dilakukan pada permainan Minecraft. Langkah-langkah yang dilakukan adalah membuat *truth table*, menentukan fungsi setiap segmen pada *seven segment display*, menyederhanakan fungsi yang telah dibuat, membuat rangkaian logika, dan mengimplementasikannya dalam permainan Minecraft.

Dalam permainan Minecraft, inti dalam pembuatan dekoder BCD ke *seven segment display* adalah sistem Redstone. Redstone merupakan sistem yang meyerupai listrik pada permainan Minecraft. Item Redstone yang digunakan antara lain adalah Redstone Lamp, Redstone Dust, Redstone Torch, Redstone Repeater, Target block, dan Lever.

Hasil dari makalah ini menunjukkan bahwa permainan Minecraft tidak terbatas sebagai media untuk rekreasi, namun juga dapat digunakan untuk banyak hal. Seperti pada makalah ini, permainan Minecraft digunakan untuk media pembelajaran serta eksperimen yang implementasi lanjutnya dapat digunakan pada dunia nyata.

#### V. SARAN

Saran apabila terdapat orang yang ingin melakukan implementasi ini adalah dengan membuat rangkaian Redstone pada Minecraft dengan lebih efisien sehingga menggunakan Redstone Dust dan ruangan yang lebih efisien.

#### VI. UCAPAN TERIMA KASIH

Penulis tidak dapat menyelesaikan makalah ini tanpa bantuan dari banyak pihak. Maka dari itu, penulis mengucapkan terima kasih kepada:

1. Allah SWT dan Nabi Muhammad SAW
2. Kedua orang tua penulis
3. Kedua dosen pengampu mata kuliah IF1220 Matematika Diskrit Semester II 2024/2025
4. Teman-teman penulis
5. Keluarga ACATRIX (STEI-K angkatan 2024).

#### REFERENCES

- [1] Raihana, Fatima. 2019. *Encoder dan Decoder*. Tangerang Selatan. Universitas Pembangunan Jaya.
- [2] Anhar, Muh. B Ogi Saputra. Yapet Tayan. 2017. *Mikrokontroler Seven*. Makassar. STMIK Professional Makassar.
- [3] Landin, Per. 2023. *What is Minecraft*. <https://www.minecraft.net/en-us/article/what-minecraft>. Diakses pada 17 Juni 2025.
- [4] Munir, Rinaldi. 2023. *Aljabar Boolean (Bag. 1)*. [https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/12-Aljabar-Boolean-\(2024\)-bagian1.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/12-Aljabar-Boolean-(2024)-bagian1.pdf). Diakses pada 17 Juni 2025.
- [5] Munir, Rinaldi. 2024. *Aljabar Boolean (Bag. 2)*. [https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/13-Aljabar-Boolean-\(2024\)-bagian2.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/13-Aljabar-Boolean-(2024)-bagian2.pdf). Diakses pada 17 Juni 2025.
- [6] Zakaria, Muchammad. *Pengertian Decoder Beserta Fungsi dan Tabel Kebenaran Decoder*. <https://www.nesabamedia.com/pengertian-decoder/>. Diakses pada 17 Juni 2025
- [7] Fikriansyah, Ilham. 2024. Apa Itu Seven Segment Display? Ini Fungsi dan Jenis-jenisnya. <https://www.detik.com/edu/detikpedia/d-7121226/apa-itu-seven-segment-display-ini-fungsi-dan-jenis-jenisnya>. Diakses pada 17 Juni 2025.
- [8] <https://education.minecraft.net/en-us/blog/creative-learning-and-leadership-minecraft-education-edition-at-indias-sat-paul-mittal-school>. Diakses pada 17 Juni 2025.
- [9] Trivusi. 2022. *Prinsip-Prinsip Hukum Aljabar Boolean dalam Sistem Digital*. <https://www.trivusi.web.id/2022/08/aljabar-boolean.html>. Diakses pada 17 Juni 2025.
- [10] Admin. 2023. *Rangkaian Logika*. <https://ekomartantoh.net/artikel/2023/09/30/rangkaian-logika/>. Diakses pada 17 Juni 2025
- [11] *BCD to 7 Segment Decoder*. 2024. <https://www.geeksforgeeks.org/digital-logic/bcd-to-7-segment-decoder/>. Diakses pada 17 Juni 2025.
- [12] *Seven Segment Displays*. 2024. <https://www.geeksforgeeks.org/digital-logic/seven-segment-displays/>. Diakses pada 17 Juni 2025.
- [13] Sheposh, Richard. 2024. <https://www.ebsco.com/research-starters/history/minecraft-game#:~:text=Minecraft%20is%20a%20popular%20video,to%20the%20public%20in%202009>. Diakses pada 17 Juni 2025.
- [14] Mattbatwings. 2023. *Logical Redstone Reloaded*. <https://www.youtube.com/watch?v=BH0j4qQORqE&list=PL5LiOvrbV08keeEWRZVaHfprU4zQTCSv4>. Diakses pada 18 Juni 2025.

#### LAMPIRAN

Berikut penulis lampirkan github yang berisi save file Minecraft yang digunakan dalam makalah ini serta video penjelasan mengenai makalah ini <https://github.com/arghaarham/MakalahMatdis1324100>

#### 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.

Sumedang, 19 Juni 2025

Arghawisesa Dwinanda Arham, 135224100

