Perbandingan antara Penerapan Operator NOR dengan Operator AND, OR, dan NOT pada Representasi Ekspresi Boolean dalam Permainan Geometry Dash (Studi Kasus Representasi Biner 4-Bit dari Fungsi Collatz)

M. Abdi Haryadi. H (13519156)[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]*abdiharyadi.ah@gmail.com*

***Abstrak*—Pada makalah ini dijelaskan perbandingan penerapan operator NOR dengan operator AND, OR, dan NOT dalam ekspresi boolean. Ekspresi boolean direpresentasikan dalam permainan Geometry Dash versi 2.111 dengan menggunakan tiga jenis *trigger*, yaitu *toggle trigger* (komponen utama), *spawn trigger*, dan *color trigger*. Kasus yang digunakan untuk penentuan fungsi boolean yang ditinjau adalah fungsi Collatz dalam representasi biner 4-bit. Hasil perbandingannya adalah operator AND, OR, NOT, dan NOR dapat direpresentasikan dalam permainan Geometry Dash dengan banyak *trigger* dan banyak ID grup yang beragam. Selain itu, ekspresi boolean hanya dengan operator NOR lebih hemat dalam penggunaan *trigger* dan ID grup, tetapi menggunakan lebih banyak proses.**

***Kata Kunci*—aljabar boolean, fungsi Collatz, Geometry Dash, NOR**

I. Pendahuluan

A. Latar Belakang

Dalam membangun ekspresi boolean, AND, OR, dan NOT adalah operator logika yang paling umum digunakan. Misalkanlah terdapat ekspresi boolean A dan ekspresi boolean B. Dengan dua ekspresi tersebut, jika dibentuk ekspresi baru dengan operator AND, ekspresi baru tersebut bernilai *true* (benar) jika dan hanya jika keduanya bernilai *true*. Jika menggunakan operator OR, ekspresi baru bernilai *true* jika dan hanya jika A atau B, atau keduanya, bernilai *true*. Untuk operator NOT yang membentuk ekspresi baru menggunakan satu ekspresi saja, misalkan A, ekspresi baru bernilai *false* (salah) jika dan hanya jika ekspresi A bernilai *true*. Tabel I menunjukkan ringkasan dari penjelasan tersebut.

Tabel I Tabel Kebenaran AND, OR, dan NOT

	AND		OR		NOT
	true	***false***	***true***	***false***	
true	*true*	*false*	*true*	*true*	*false*
false	*false*	*false*	*true*	*false*	*true*

Ekspresi boolean ini dapat diimplementasikan ke dalam salah satu permainan yang dapat ditemukan di Google Play bernama Geometry Dash. Permainan yang dibuat oleh Robert Topala (dikenal dengan nama RobTop) ini adalah salah satu permainan yang memberikan kebebasan pemain untuk memainkan dan menyelesaikan sekumpulan rintangan yang disebut level. Level dapat berasal dari pembuat permainan ini sendiri ataupun dari pemain-pemain lain. Gambar 1 menunjukkan tangkapan layar level yang dibuat oleh Serponge (nama dalam permainan) dan kawan-kawan dengan nama CastleMania. Pemain dapat menuangkan kreativitasnya melalui level yang dibuat, salah satunya dalam hal mekanisme. Contohnya adalah kondisi suatu objek rintangan muncul yang dapat dikaitkan dengan ekspresi boolean. Operator-operator di dalamnya dapat dibuat dengan kumpulan *trigger* (picu) yang disusun sedemikian sehingga menghasilkan suatu prosedur yang dapat merepresentasikan operator AND, OR, dan NOT.

Gambar 1 CastleMania—contoh level dalam permainan Geometry Dash

Akan tetapi, ada juga operator logika bernama NOR. Operator ini, sesuai namanya, adalah gabungan dari operator NOT dan OR. Tabel II menunjukkan tabel kebenaran dari operasi NOR. Perhatikan bahwa hasilnya berlawanan dengan operator OR yang ditunjukkan oleh Tabel I. Uniknya, ekspresi boolean yang mengandung operator AND, OR, dan/atau NOT dapat digantikan dengan satu jenis operator saja, yaitu NOR.

Tabel II Tabel Kebenaran NOR

	NOR	
	true	***false***
true	*false*	*false*
false	*false*	*true*

B. Rumusan Masalah

Dari latar belakang tersebut, timbul dua pertanyaan:

1) Bagaimanakah cara merepresentasikan operator AND, OR, NOT, dan NOR dalam permainan Geometry Dash?
2) Bagaimanakah kelebihan dan kekurangan antara penggunaan operator NOR saja dengan penggunaan operator AND, OR, dan NOT saja dalam permainan Geometry Dash?

C. Tujuan

Tujuan pembuatan makalah ini adalah sebagai berikut:

1) mengetahui representasi operator AND, OR, NOT, dan NOR dalam permainan Geometry Dash; serta
2) mengetahui kelebihan dan kekurangan antara penggunaan operator NOR saja dengan penggunaan operator AND, OR, dan NOT saja dalam permainan Geometry Dash.

II. TEORI DASAR

A. Aljabar Boolean

Jika berbicara tentang hal-hal yang menyatakan dua hal saja seperti benar atau salah, itu bisa dikaitkan dengan aljabar boolean. Aljabar boolean menggunakan operator-operator boolean, termasuk operator logika, untuk mengoperasikan hal-hal yang memiliki dua nilai yang saling komplementer [2:847]. Dalam aljabar boolean, untuk penyederhanaan, *true* dianggap bernilai 1, sedangkan *false* dianggap bernilai 0.

Terdapat tiga operasi yang sering digunakan dalam aljabar boolean: penjumlahan boolean (+), perkalian boolean (·), dan komplemen (ditandai dengan garis atas) [2:847]. Operator OR ekuivalen dengan penjumlahan boolean, AND dengan perkalian boolean, dan komplemen dengan NOT. Hasil dari tiga keterkaitan tersebut sama. Misalnya, 0 · 1 = 0 ekuivalen dengan "*false* AND *true* bernilai *false*" (lihat tabel I). Oleh karena itu, untuk penyederhanaan, notasi aljabar booleanlah yang lebih banyak digunakan dalam makalah ini. Untuk selanjutnya, operator untuk penjumlahan boolean lebih sering disebut operator OR, operator untuk perkalian boolean disebut operator AND, dan operator untuk komplemen disebut operator NOT.

Secara informal, gabungan antara operan-operan, baik 0 dan 1 maupun variabel boolean (nilainya hanya bisa 0 atau 1), dengan operator-operator disebut sebagai ekspresi boolean. Misalkan a dan b adalah variabel boolean, contoh ekspresi boolean adalah $ab + \bar{c}$ (operator untuk perkalian kadang-kadang tidak diperlihatkan untuk penyederhanaan), $\bar{b} \cdot \overline{(a + b + c)}$, dan $0 + 1 + \bar{\bar{a}}$ (dua garis atas untuk a menunjukkan komplemen dari komplemen-dari-a).

Ekspresi boolean dapat membentuk fungsi boolean, yaitu seperti fungsi pada umumnya, namun menggunakan variabel-variabel boolean. Contohnya adalah $f(x, y, z) = xy + \bar{z}$ dengan x, y, dan z adalah variabel boolean. Jika semua kemungkinan nilai-nilai variabelnya disertai nilai hasil pemetaannya diketahui, fungsi boolean dapat dibentuk dengan cara yang ditunjukkan oleh [2:855]. Misalkanlah fungsi boolean $g(0, 1,0) = g(1,0,1) = 1$, dan bernilai 0 untuk hasil pemetaannya. Tinjau $g(0, 1,0)$ terlebih dahulu. Perhatikan bahwa jika $x = 0$, $y = 1$, dan $z = 0$, hasil pemetannya bernilai 1. Maka, dengan operasi komplemen dan perkalian boolean, $\bar{x}y\bar{z} = 1$. Dengan cara yang sama untuk $g(1, 0,1)$, didapatkan $x\bar{y}z = 1$. Untuk menggabungkan nilainya, gunakan penjumlahan boolean dan didapatkanlah fungsi boolean $g(x, y, z) = \bar{x}y\bar{z} + x\bar{y}z$.

Tentunya dalam membentuk fungsi boolean seperti itu, ekspresinya akan semakin rumit jika terdapat banyak kemungkinan sehingga nilai hasil pemetaannya bernilai 1. Untuk menyederhanakan hal tersebut, tabel III menunjukkan identitas-identitas dalam aljabar boolean yang diperoleh dari [2:851]. Identitas-identitas ini juga akan membantu dalam mengonversi operator-operator lain ke operator NOR yang akan dibahas pada bagian berikutnya.

Tabel III Identitas-Identitas Boolean

Ekspresi	**Nama Identitas**
$\bar{\bar{x}} = x$	Hukum komplemen ganda
$x + x = x$ $x \cdot x = x$	Hukum idempoten
$x + 0 = x$ $x \cdot 1 = x$	Hukum identitas
$x + 1 = 1$ $x \cdot 0 = 0$	Hukum dominasi
$x + y = y + x$ $xy = yx$	Hukum komutatif
$x + y = y + x$ $xy = yx$	Hukum asosiatif
$x(y + z) = xy + xz$ $x + yz = (x + y)(x + z)$	Hukum distributif
$\overline{(x + y)} = \bar{x}\bar{y}$ $\overline{(xy)} = \bar{x} + \bar{y}$	Hukum De Morgan
$x + xy = x$ $x(x + y) = x$	Hukum penyerapan
$x + \bar{x} = 1$ $x\bar{x} = 0$	Sifat komplementer

B. NOR

NOR adalah operator logika yang menggabungkan operator OR dan NOT. Dalam aljabar boolean, NOR dapat dinotasikan dengan tanda panah bawah (↓) [2:857] dan didefinisikan sebagai berikut:

$$x \downarrow y = \overline{(x + y)}. \;....... (1)$$

Dengan hukum De Morgan, ekspresi tersebut dapat berubah menjadi

$$x \downarrow y = \bar{x}\bar{y}. \;....... (2)$$

Operator AND, OR, dan NOT yang sering digunakan dalam aljabar boolean dapat digantikan oleh satu jenis operator saja, yaitu NOR. Akibatnya, setiap ekspresi boolean dapat dibuat hanya dengan operasi NOR saja.

Penurunan dari operasi komplemen adalah sebagai berikut:

$$\begin{aligned} \bar{x} &= \overline{(x+0)} && \text{(hukum identitas)} \\ &= x \downarrow 0 && \text{(definisi NOR)} \end{aligned}$$

Penurunan dari operasi penjumlahan boolean adalah sebagai berikut:

$$\begin{aligned} x+y &= \overline{\overline{(x+y)}} && \text{(hukum komplemen ganda)} \\ &= \overline{(x \downarrow y)} && \text{(definisi NOR)} \\ &= (x \downarrow y) \downarrow 0 && \text{(konversi komplemen ke NOR)} \end{aligned}$$

Penurunan dari operasi perkalian boolean adalah sebagai berikut:

$$\begin{aligned} xy &= \overline{\overline{(xy)}} && \text{(hukum komplemen ganda)} \\ &= \overline{(\bar{x}+\bar{y})} && \text{(hukum De Morgan)} \\ &= \bar{x} \downarrow \bar{y} && \text{(definisi NOR)} \\ &= (x \downarrow 0) \downarrow (y \downarrow 0) && \text{(konversi komplemen ke NOR)} \end{aligned}$$

Untuk membuatnya lebih lengkap, terdapat alternatif lain untuk konversi dari operasi komplemen. Alternatifnya adalah menggunakan hukum idempoten sehingga seperti ini:

$$\begin{aligned} \bar{x} &= \overline{(x+x)} && \text{(hukum idempoten)} \\ &= x \downarrow x && \text{(definisi NOR)} \end{aligned}$$

Akan tetapi, alternatif pertama (dengan hukum identitas) lebih digunakan dalam makalah ini dengan alasan minimalisasi. Tabel IV menyimpulkan tiga konversi ke NOR yang telah diturunkan.

Tabel IV Konversi ke NOR

Ekspresi	Konversi
$\bar{x} = x \downarrow 0$ $\bar{x} = x \downarrow x$	Komplemen ke NOR
$x+y = (x \downarrow y) \downarrow 0$ $x+y = (x \downarrow y) \downarrow (x \downarrow y)$	Penjumlahan boolean ke NOR
$xy = (x \downarrow 0) \downarrow (y \downarrow 0)$ $xy = (x \downarrow x) \downarrow (y \downarrow y)$	Perkalian boolean ke NOR

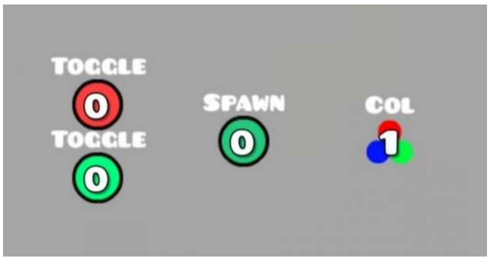
Perlu diperhatikan bahwa operator NOR bersifat komutatif mengingat definisinya dari persamaan (1), penjumlahan boolean bersifat komutatif. Akan tetapi, operator NOR tidak bersifat asosiatif karena terdapat pasangan nilai x, y, dan z yang tidak memenuhi $(x \downarrow y) \downarrow z = x \downarrow (y \downarrow z)$. Misalnya, $(1 \downarrow 1) \downarrow 0$ bernilai 1, tetapi $1 \downarrow (1 \downarrow 0)$ bernilai 0.

C. Trigger-Trigger Geometry Dash

Trigger adalah komponen penting dalam permainan Geometry Dash untuk melakukan suatu aksi khusus seperti menggerakkan objek, memutar objek, mengganti warna objek, dan lain-lain. Objek yang dipengaruhi oleh *trigger* dapat ditentukan berdasarkan ID (nilai unik) yang dimilikinya.

Dalam makalah ini, ID yang digunakan hanyalah dua jenis: ID grup dan ID warna. ID grup adalah jenis ID yang digunakan untuk mengelompokkan beberapa objek yang ingin dipengaruhi. Setiap objek dapat memiliki ID grup bukan hanya satu, tetapi juga dapat lebih dari satu. Namun, untuk setiap objek, banyak ID grup maksimum yang dapat disimpan adalah sepuluh. Di sisi lain, ID warna adalah jenis ID yang menyimpan informasi warna. Misalnya, objek A dengan ID warna 99 akan memiliki informasi warna yang sama dengan objek B yang memiliki ID warna 99 juga dengan syarat tidak ada modifikasi warna yang berbeda satu sama lain. Banyak ID warna bergantung pada jenis objek. Modifikasi warna tidak akan dijelaskan dalam makalah ini, dan banyak ID warna yang dapat dipengaruhi untuk setiap objek dianggap hanya satu.

Terdapat tiga jenis *trigger* yang digunakan dalam menyimulasikan operator-operator logika: *toggle trigger, spawn trigger,* dan *color trigger*. Gambar 2 menunjukkan tampak tiga *trigger* ini dalam mode editior Geometry Dash. Secara garis besar, *toggle trigger* akan berguna dalam menentukan keaktifan (kemunculan) suatu objek dengan ID grup tertentu. *Trigger* ini dapat diatur sehingga dapat menonaktifkan (melenyapkan) objek yang sedang aktif yang ditandai oleh warna merah, atau sebaliknya yang ditandai oleh warna hijau. Kemudian, *spawn trigger* berperan dalam memicu kumpulan *trigger* dengan ID grup yang sama sehingga dapat melakukan aksi masing-masing. Lalu, *color trigger* dipicu untuk mengubah warna yang menandakan aktif tidaknya *trigger* tersebut. Dalam implementasi ekspresi boolean, jika *color trigger* aktif, ekspresi boolean bernilai 1. Selain itu, angka yang ditunjukkan dalam setiap *trigger* menunjukkan ID target, yaitu ID yang akan dipengaruhinya. Untuk tiga jenis *trigger* yang ditunjukkan pada gambar 2, *color trigger* memengaruhi ID warna, sedangkan yang lainnya memengaruhi ID grup.



Gambar 2 Tampak *toggle trigger* (kiri), *spawn trigger* (tengah), dan *color trigger* (kanan)

D. Representasi Biner

Suatu bilangan yang sering digunakan terdiri dari sepuluh jenis digit angka, yaitu 0 sampai 9, yang biasanya disebut bilangan desimal. Namun, untuk kasus yang melibatkan fungsi boolean, hanya dua jenis digit saja yang terlibat, yaitu 0 dan 1. Maka, untuk mengubah suatu bilangan desimal, misalkan n, menjadi dua jenis digit saja, misalkan x, nilai x_k, x_{k-1}, hingga x_0 yang hanya dapat bernilai 0 atau 1, perlu ditemukan hingga memenuhi persamaan berikut: [2:260]

$$n = 2^k x_k + 2^{k-1} x_{k-1} + \cdots + 2x_1 + x_0, \text{.......} \; (3)$$

nilai k adalah bilangan bulat nonnegatif, x_k tidak sama dengan 0. Maka, $x_k x_{k-1} \dots x_1 x_0$ adalah nilai dari x yang dapat disebut representasi biner dari n. Sebagai contoh, untuk $n = 11$, representasi binernya adalah 1011. Agar dapat dibedakan antara bilangan desimal dengan bilangan biner, representasinya dapat ditulis sebagai $(1011)_2$. Sebagai pelengkap, kasus khusus untuk $n = 0$, reprsentasi binernya, dapat ditebak, adalah $(0)_2$. Ini akan diterapkan dalam penentuan fungsi boolean pada bagian III-B.

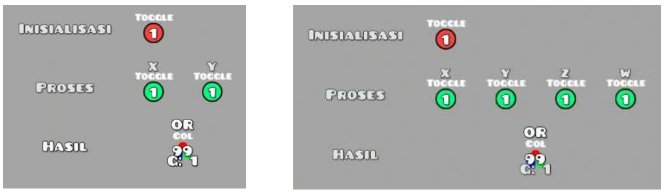
III. HASIL DAN PEMBAHASAN

A. Representasi Operator-Operator Logika dalam Permainan Geometry Dash

Sebelum menunjukkan hasil, penulis menetapkan suatu konvensi untuk setiap gambar pada bagian ini. Ada tiga kelompok *trigger* yang ditunjukkan: tahap inisialisasi, tahap proses, dan tahap hasil. Jika prosesnya lebih dari satu, bagian proses dipecah menjadi proses 1, proses 2, dan seterusnya. Hanya satu *trigger* yang berada dalam tahap hasil, yaitu *color* trigger dengan target ID warna 99. Pada setiap *trigger* yang diberikan ID grup, terdapat label untuk menunjukkan ID grup yang ada pada *trigger* tersebut. Misalnya pada gambar 3 bagian (a), *color trigger* driberi label "G: 1" yang menunjukkan *color trigger* memiliki ID grup 1 sehingga dapat dipengaruhi oleh *trigger* dengan ID target grup 1. Selain label ID grup, ada juga label untuk menunjukkan operan dan operator pada bagian atas setiap *trigger*, kecuali pada tahap inisialisasi.

Untuk menentukan nilai dari suatu variabel, setiap *trigger* dengan label nama variabel tersebut dihapus untuk nilai 0, dibiarkan ada untuk nilai 1. Hal tersebut juga berlaku jika ingin dinonaktifkan dengan *toggle trigger*. Intinya, tampak *trigger* untuk nilai 0 adalah tidak ada *trigger*.

Gambar 3 bagian (a) menunjukkan operator OR dalam permainan Geometry Dash dengan ekspresi boolean $x + y$. Mula-mula operator OR diinisialisasi dengan menonaktifkan ID grup yang diterapkan pada objek hasil operasi. Dari gambar tersebut, ID grup yang dimaksud adalah 1. Dalam tahap proses, dua *toggle trigger* mengaktifkan objek dengan ID grup yang sama, yaitu 1. Hal tersebut masuk akal mengingat objek hanya dengan ID grup 1 dapat diaktifkan dengan *toggle trigger* satu kali saja. Jika diaktifkan dua kali, tidak ada pengaruh untuk pengaktifan kedua sehingga tetap aktif. Jika tidak ada *toggle trigger* yang mengaktifkannya, objek dengan ID grup 1 nonaktif karena hanya dipengaruhi oleh tahap inisialisasi. Uniknya, untuk operasi yang melibatkan penjumlahan boolean saja seperti $x + y + z + w$, cukup tambahkan dua *trigger toggle* lagi dengan sifat yang sama dengan label yang berbeda. Hal itu ditunjukkan dalam gambar 3 bagian (b).



(a) (b)

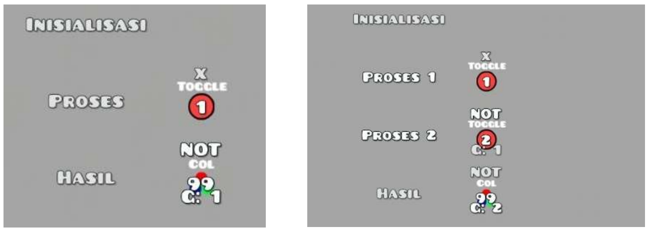
Gambar 3 Operator OR dalam permainan Geometry Dash—bagian (a) untuk ekspresi boolean $x + y$; bagian (b) untuk ekspresi boolean $x + y + z + w$

Gambar 4 bagian (a) menunjukkan operator AND dalam permainan Geometry Dash dengan ekspresi boolean xy. Tahap inisialisasi untuk operator AND hampir sama dengan OR, namun ada dua ID grup yang dinonaktifkan. Dari gambar tersebut, ID grup yang dimaksud adalah 1 dan 2. Kedua ID grup ini diterapkan pada objek hasil operasi. Akibatnya, untuk mengaktifkan kembali objek tersebut, kedua ID grup tersebut harus diaktifkan karena salah satunya saja tidak cukup. Hal tersebut mereprentasikan operasi AND mengingat semuanya harus bernilai 1 (*true*) untuk menghasilkan 1. Hampir sama dengan OR, untuk operasi yang melibatkan perkalian boolean saja seperti $xyzw$, cukup tambahkan dua *trigger toggle* lagi, tetapi ID targetnya harus berbeda satu sama lain. ID grup baru yang digunakan sebagai target perlu dimiliki juga oleh objek hasil operasi. Hal itu ditunjukkan dalam gambar 4 bagian (b).

(a) (b)

Gambar 4 Operator AND dalam permainan Geometry Dash—bagian (a) untuk ekspresi boolean xy; bagian (b) untuk ekspresi boolean $xyzw$

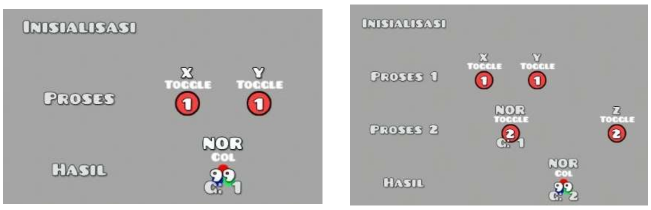
Gambar 5 bagian (a) menunjukkan operator NOT dalam permainan Geometry Dash dengan ekspresi boolean $\bar{x}$. Perhatikan bahwa tidak ada *trigger* untuk inisialisasi sehingga jika tidak ada proses, objek hasil operasi akan tetap aktif. Namun, pada tahap proses terdapat *toggle trigger* untuk menonaktifkan objek hasil operasi sehingga jika *trigger* tersebut aktif, hasil operasi tentunya akan nonaktif. Sudah jelas bahwa cara ini merepresentasian operator NOT. Akan tetapi, tidak seperti operator AND dan OR, mengingat NOT hanya menerima satu operan, melakukan operasi komplemen satu kali lagi menambah proses baru. Hal itu ditunjukkan oleh gambar 5 bagian (b) untuk ekspresi boolean $\bar{\bar{x}}$. Perhatikan bahwa terdapat ID grup baru yang berbeda untuk proses selanjutnya. Beruntungnya, mengingat adanya hukum komplemen ganda pada tabel III, ekspresi boolean ini bisa diubah menjadi x saja.



(a) (b)

Gambar 5 Operator NOT dalam permainan Geometry Dash—bagian (a) untuk ekspresi boolean $\bar{x}$; bagian (b) untuk ekspresi boolean $\bar{\bar{x}}$

Gambar 6 bagian (a) menunjukkan operator NOR dalam permainan Geometry Dash dengan ekspresi boolean $x \downarrow y$. Sama seperti operator NOT, tidak ada *trigger* dalam tahap inisialisasi. Perhatikan bahwa pada tahap proses, satu saja *trigger* yang aktif akan menonaktifkan objek hasil operasi. Hal itu dapat merepresentasikan NOR mengingat objek hasil operasi bisa tetap aktif jika dan hanya jika tidak ada *toggle trigger* yang menonaktifkannya. Perhatikan juga bahwa jika salah satu *trigger* dinonaktifkan, atau dengan kata lain ada variabel yang nilainya 0 antara x dan y. Jika itu terjadi, strukturnya akan seperti pada gambar 5 bagian (a) yang menunjukkan ekspresi boolean $\bar{x}$. Hal itu sesuai mengingat adanya persamaan konversi komplemen dari ke NOR, yaitu $\bar{x} = x \downarrow 0$. Selain itu, operator NOR tidak bisa digabungkan dengan operator lain mengingat operator NOR tidak bersifat asosiatif. Oleh karena itu, untuk ekspresi boolean $(x \downarrow y) \downarrow z$, ada dua proses yang digunakan seperti pada gambar 6 bagian (b).



(a) (b)

Gambar 6 Operator NOR dalam permainan Geometry Dash—bagian (a) untuk ekspresi boolean $x \downarrow y$; bagian (b) untuk ekspresi boolean $(x \downarrow y) \downarrow z$

Dari keempat representasi operator logika dalam permainan Geometry Dash, tabel V menunjukkan banyak *trigger* pada tahap inisialisasi dan tahap proses, serta banyak ID grup yang digunakan. Jika ditinjau untuk satu operator (tunggal), terlihat bahwa operator OR dan AND direpresentasikan dengan banyak *trigger* yang besar. Namun, operator-operator OR dapat digabungkan dalam satu proses sehingga dapat menghemat objek sehingga tidak perlu menggunakan objek tambahan untuk menyimpan nilai sementara dari hasil operasi dari operator OR. Operator AND juga berlaku demikian, tetapi dengan konsekuensi banyak ID grup juga bertambah sehingga banyak *trigger* dalam tahap inisialisasi juga bertambah.

Tabel V Banyak *Trigger* dan Banyak ID Grup untuk Representasi Operator Logika dalam Permainan Geometry Dash

Operator	**Banyak *Trigger***	**Banyak ID Grup**
OR (tunggal)	3	1
AND (tunggal)	4	2
NOT	1	1
NOR	2	1
OR (n buah dalam satu proses)	n+2	1
AND (n buah dalam satu proses)	2n+2	n+1

B. Fungsi Boolean yang Digunakan

Dinyatakan kembali bahwa perbandingan yang ditinjau adalah perbandingan antara penerapan operator NOR dengan operator AND, OR, dan NOT pada representasi ekspresi boolean. Perbandingan tersebut didapatkan dengan metode eksperimen dalam permainan Geometry Dash. Versi yang digunakan adalah 2.111.

Fungsi boolean yang digunakan melibatkan fungsi Collatz, yaitu fungsi yang berasal dari Lothar Collatz [1]. Misalkanlah bilangan asli n dan fungsi f yang terdefinisi sebagai berikut:

$$f(n) = \begin{cases} 3n+1, n \text{ bilangan ganjil} \\ \frac{n}{2}, n \text{ bilangan genap} \end{cases} \quad \text{....... (4)}$$

Agar fungsi Collatz dapat digunakan untuk fungsi boolean, fungsi tersebut direpresentasikan dalam biner 4-bit. Akibatnya, empat fungsi boolean digunakan, yaitu y_3, y_2, y_1, dan y_0. Masing-masing fungsi tersebut mengguakan empat variabel sebagai masukan, yaitu x_3, x_2, x_1, dan x_0. Untuk $f(n) \geq 16$, yang bukan empat digit terakhir dari representasi biner $f(n)$ diabaikan. Untuk kasus khusus $n = 0$, $f(0) = 0$. Tabel VI menunjukkan hasil pemetaan setiap fungsi dengan variabel-variabel tersebut. Sebagai contoh, $y_2(0,0,1,1) = 0$.

Tabel VI Hasil Pemetaan y_3, y_2, y_1, dan y_0 Berdasarkan Nilai x_3, x_2, x_1, dan x_0

n	x_3	x_2	x_1	x_0	$f(n)$	y_3	y_2	y_1	y_0
0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	4	0	1	0	0
2	0	0	1	0	1	0	0	0	1
3	0	0	1	1	10	1	0	1	0
4	0	1	0	0	2	0	0	1	0
5	0	1	0	1	16	0	0	0	0
6	0	1	1	0	3	0	0	1	1
7	0	1	1	1	22	0	1	1	0
8	1	0	0	0	4	0	1	0	0
9	1	0	0	1	28	1	1	0	0
10	1	0	1	0	5	0	1	0	1
11	1	0	1	1	34	0	0	1	0
12	1	1	0	0	6	0	1	1	0
13	1	1	0	1	40	1	0	0	0
14	1	1	1	0	7	0	1	1	1
15	1	1	1	1	46	1	1	1	0

Dari tabel di atas, fungsi y_3, y_2, y_1, dan y_0 dapat didefinisikan dengan ekspresi booean. Fungsi y_3 didefinisikan sebagai

$$y_3(x_3, x_2, x_1, x_0) = \overline{x_3}\,\overline{x_2}\,x_1\,x_0 + x_3\,\overline{x_2}\,\overline{x_1}\,x_0 + x_3\,x_2\,\overline{x_1}\,x_0 + x_3\,x_2\,x_1\,x_0 \text{ ...... (5)}$$

yang dapat disederhanakan sebagai

$$y_3(x_3, x_2, x_1, x_0) = (\overline{x_3}\,\overline{x_2}\,x_1 + x_3(\overline{x_1} + x_2))\,x_0 \text{ .... (6)}$$

dan operator-operatornya dapat dikonversi menjadi NOR sehingga menjadi

$$y_3(x_3, x_2, x_1, x_0) = (\Big(((x_3 \downarrow x_2) \downarrow 0) \downarrow (x_1 \downarrow 0)\Big) \downarrow ((x_3 \downarrow 0) \downarrow ((x_1 \downarrow 0) \downarrow x_2))) \downarrow (x_0 \downarrow 0). \text{ .... (7)}$$

Fungsi y_2 didefinisikan sebagai

$$y_2(x_3, x_2, x_1, x_0) = \overline{x_3}\,\overline{x_2}\,\overline{x_1}\,x_0 + \overline{x_3}\,x_2\,x_1\,x_0 + x_3\,\overline{x_2}\,\overline{x_1}\,\overline{x_0} + x_3\,\overline{x_2}\,\overline{x_1}\,x_0 + x_3\,\overline{x_2}\,x_1\,\overline{x_0} + x_3\,x_2\,\overline{x_1}\,\overline{x_0} + x_3\,x_2\,x_1\,\overline{x_0} + x_3\,x_2\,x_1\,x_0 \text{ ...... (8)}$$

yang dapat disederhanakan sebagai

$$y_2(x_3, x_2, x_1, x_0) = (\overline{x_2}\,\overline{x_1} + x_2\,x_1)\,x_0 + x_3\,\overline{x_0} \text{ .... (9)}$$

dan operator-operatornya dapat dikonversi menjadi NOR sehingga menjadi

$$y_2(x_3, x_2, x_1, x_0) = \Big(\Big(\Big(\big((x_2 \downarrow x_1) \downarrow ((x_2 \downarrow 0) \downarrow (x_1 \downarrow 0))\big) \downarrow (x_0 \downarrow 0)\Big) \downarrow ((x_3 \downarrow 0) \downarrow x_0)\Big) \downarrow 0. \;....... (10)$$

Fungsi y_1 didefinisikan sebagai

$$y_1(x_3, x_2, x_1, x_0) = \overline{x_3}\,\overline{x_2}\,x_1\,x_0 + \overline{x_3}\,x_2\,\overline{x_1}\,\overline{x_0} + \overline{x_3}\,x_2\,x_1\,\overline{x_0} + \overline{x_3}\,x_2\,x_1\,x_0 + x_3\,\overline{x_2}\,x_1\,x_0 + x_3\,x_2\,\overline{x_1}\,\overline{x_0} + x_3\,x_2\,x_1\,\overline{x_0} + x_3\,x_2\,x_1\,x_0 \;...... (11)$$

yang dapat disederhanakan sebagai

$$y_1(x_3, x_2, x_1, x_0) = x_1\,x_0 + \,x_2\,\overline{x_0} \;....... (12)$$

dan operator-operatornya dapat dikonversi menjadi NOR sehingga menjadi

$$y_1(x_3, x_2, x_1, x_0) = (((x_1 \downarrow 0) \downarrow (x_0 \downarrow 0)) \downarrow ((x_2 \downarrow 0) \downarrow x_0)) \downarrow 0. \;....... (13)$$

Fungsi y_0 didefinisikan sebagai

$$y_0(x_3, x_2, x_1, x_0) = \overline{x_3}\,\overline{x_2}\,x_1\,\overline{x_0} + \overline{x_3}\,x_2\,x_1\,\overline{x_0} + x_3\,\overline{x_2}\,x_1\,\overline{x_0} + x_3\,x_2\,x_1\,\overline{x_0} \;...... (14)$$

yang dapat disederhanakan sebagai

$$y_0(x_3, x_2, x_1, x_0) = x_1\,\overline{x_0} \;....... (15)$$

dan operator-operatornya dapat dikonversi menjadi NOR sehingga menjadi

$$y_0(x_3, x_2, x_1, x_0) = (x_1 \downarrow 0) \downarrow x_0. \;....... (16)$$

Semua penurunannya dapat dilihat pada lampiran A.

C. Analisis Representasi Ekspresi Boolean dalam Permainan Geometry Dash

Gambar-gambar representasi ekspresi boolean dari kasus yang digunakan dalam permainan Geometry Dash berada pada lampiran B. Banyak objek, banyak (bagian) proses yang digunakan (tidak termasuk hasil), serta banyak ID grup yang digunakan untuk tahap inisialisasi, proses, dan hasil yang digunakan oleh *toggle trigger* (tidak termasuk *toggle trigger* untuk input) disimpulkan pada tabel VII. Berdasarkan tabel tersebut, representasi ekspresi boolean hanya dengan operator NOR lebih hemat dalam penggunaan *trigger* dan ID grup daripada representasi ekspresi boolean tanpa NOR. Akan tetapi, proses yang diperlukan untuk representasi ekspresi boolean hanya dengan NOR lebih banyak. Hal ini dapat didukung mengingat representasi operator NOR yang banyak tidak dapat digabungkan seperti operator OR (lihat gambar 3 bagian b) dan AND (lihat gambar 4 bagian b) dalam permainan Geometry Dash. Untuk ekspresi boolean tanpa NOR, persamaannya dapat disederhanakan lagi sehingga operasinya berkurang, serta untuk representasi dalam permainan Geometry Dash, penjumlahan boolean lebih diutamakan. Contohnya pada persamaan (6), $\overline{x_3}\,\overline{x_2}$ dapat diganti menjadi $\overline{(x_3 + x_2)}$ mengingat adanya hukum De Morgan.

Tabel VII Banyak *Trigger*, Banyak ID Grup, dan Banyak Proses untuk Representasi Ekspresi Boolean dari Kasus Fungsi Collatz 4-Bit dalam Permainan Geometry Dash

Fungsi	Tanpa NOR			Hanya dengan NOR		
	Banyak *Trigger*	Banyak ID Grup	Banyak Proses	Banyak *Trigger*	Banyak ID Grup	Banyak Proses
y_3	24	12	5	18	11	5
y_2	26	13	5	18	11	6
y_1	13	6	3	11	6	4
y_0	6	3	2	4	2	2
Total	69	34	15	51	30	17

V. KESIMPULAN

Operator OR, AND, NOT, dan NOR dapat direpresentasikan dalam permainan Geometry Dash dengan banyak *trigger* dan banyak ID grup yang beragam. Dalam permainan tersebut, representasi ekspresi boolean yang hanya menggunakan operator NOR lebih menghemat penggunaan *trigger* dan ID grup, tetapi menggunakan lebih banyak proses jika dibandingkan dengan representasi ekspresi boolean tanpa operator NOR. Hasil ini diharapkan dapat membantu dalam pembuatan level Geometry Dash yang berkaitan dengan operator logika. Hasil ini juga diharapkan dapat dikaitkan dengan penerapan di luar permainan Geometry Dash sehingga keistimewaan dari operator NOR dapat berlaku dalam jangkauan yang lebih luas.

VI. LAMPIRAN

A. Penyederhanaan dan Konversi Operator ke NOR dari Fungsi Boolean

Pada bagian ini, untuk membuat tulisan menjadi lebih sederhana, hukum komplemen ganda, hukum komutatif, hukum asosiatif, hukum identitas, dan operasi-operasi yang tidak terlalu penting tidak ditunjukkan.

Fungsi boolean dari persamaan (5) dapat disederhanakan menjadi persamaan (6) dengan cara sebagai berikut:

$$y_3(x_3, x_2, x_1, x_0) = \overline{x_3}\,\overline{x_2}\,x_1\,x_0 + x_3\,\overline{x_2}\,\overline{x_1}\,x_0 + x_3\,x_2\,\overline{x_1}\,x_0 + x_3\,x_2\,x_1\,x_0$$

$= (\overline{x_3}\,\overline{x_2}\,x_1 + x_3\,\overline{x_2}\,\overline{x_1} + x_3\,x_2\,\overline{x_1} + x_3\,x_2\,x_1)\,x_0$ (hukum distributif)

$= (\overline{x_3}\,\overline{x_2}\,x_1 + x_3\,\overline{x_2}\,\overline{x_1} + x_3\,x_2\,\overline{x_1} + x_3\,x_2\,\overline{x_1} + x_3\,x_2\,x_1)\,x_0$ (hukum idempoten)

$= (\overline{x_3}\,\overline{x_2}\,x_1 + x_3\,\overline{x_1}(\overline{x_2} + x_2) + x_3\,x_2\,(\overline{x_1} + x_1))\,x_0$ (hukum distributif)

$= (\overline{x_3}\,\overline{x_2}\,x_1 + x_3\,\overline{x_1} + x_3\,x_2)\,x_0$ (sifat komplementer)

$= (\overline{x_3}\,\overline{x_2}\,x_1 + x_3(\overline{x_1} + x_2))\,x_0$ (hukum distributif)

Operator-operator fungsi boolean dari persamaan (6) dapat dikonversi menjadi operator NOR seperti persamaan (7) dengan cara sebagai berikut:

$$y_3(x_3, x_2, x_1, x_0) = (\overline{x_3}\,\overline{x_2}\,x_1 + x_3(\overline{x_1} + x_2))\,x_0$$

$= ((x_3 \downarrow x_2)\,x_1 + x_3\overline{(\overline{x_1} \downarrow x_2)})\,x_0$ (definisi NOR)

$= ((\overline{(x_3 \downarrow x_2)} \downarrow \overline{x_1}) + (\overline{x_3} \downarrow (\overline{x_1} \downarrow x_2)))\,x_0$ (definisi NOR)

$= \overline{(((x_3 \downarrow x_2) \downarrow \overline{x_1}) \downarrow (\overline{x_3} \downarrow (\overline{x_1} \downarrow x_2)))}\; x_0$ (definisi NOR)
$= (((x_3 \downarrow x_2) \downarrow \overline{x_1}) \downarrow (\overline{x_3} \downarrow (\overline{x_1} \downarrow x_2))) \downarrow \overline{x_0}$ (definisi NOR)
$= ((((x_3 \downarrow x_2) \downarrow 0) \downarrow (x_1 \downarrow 0)) \downarrow ((x_3 \downarrow 0) \downarrow ((x_1 \downarrow 0) \downarrow x_2))) \downarrow (x_0 \downarrow 0)$ (konversi komplemen ke NOR)

Fungsi boolean dari persamaan (8) dapat disederhanakan menjadi persamaan (9) dengan cara sebagai berikut:

$$y_2(x_3, x_2, x_1, x_0) = \overline{x_3}\,\overline{x_2}\,\overline{x_1}\,x_0 + \overline{x_3}\,x_2\,x_1\,x_0 + x_3\,\overline{x_2}\,\overline{x_1}\,\overline{x_0} + x_3\,\overline{x_2}\,\overline{x_1}\,x_0 + x_3\,\overline{x_2}\,x_1\,\overline{x_0} + x_3\,x_2\,\overline{x_1}\,\overline{x_0} + x_3\,x_2\,x_1\,\overline{x_0} + x_3\,x_2\,x_1\,x_0$$

$= \overline{x_2}\,\overline{x_1}\,x_0(\overline{x_3} + x_3) + x_2\,x_1\,x_0(\overline{x_3} + x_3) + x_3\,\overline{x_0}(\overline{x_2}\,\overline{x_1} + \overline{x_2}\,x_1 + x_2\,\overline{x_1} + x_2\,x_1)$ (hukum distributif)
$= \overline{x_2}\,\overline{x_1}\,x_0 + x_2\,x_1\,x_0 + x_3\,\overline{x_0}$ (sifat komplementer)
$= (\overline{x_2}\,\overline{x_1} + x_2\,x_1)\,x_0 + x_3\,\overline{x_0}$ (hukum distributif)

Operator-operator fungsi boolean dari persamaan (9) dapat dikonversi menjadi operator NOR seperti persamaan (10) dengan cara sebagai berikut:

$$y_2(x_3, x_2, x_1, x_0) = (\overline{x_2}\,\overline{x_1} + x_2\,x_1)\,x_0 + x_3\,\overline{x_0}$$

$= \overline{((x_2 \downarrow x_1) + (\overline{x_2} \downarrow \overline{x_1}))}x_0 + (\overline{x_3} \downarrow x_0)$ (definisi NOR)
$= ((x_2 \downarrow x_1) \downarrow (\overline{x_2} \downarrow \overline{x_1}))x_0 + (\overline{x_3} \downarrow x_0)$ (definisi NOR)
$= (((x_2 \downarrow x_1) \downarrow (\overline{x_2} \downarrow \overline{x_1})) \downarrow \overline{x_0}) + (\overline{x_3} \downarrow x_0)$ (definisi NOR)
$= \overline{(((x_2 \downarrow x_1) \downarrow (\overline{x_2} \downarrow \overline{x_1})) \downarrow \overline{x_0}) \downarrow (\overline{x_3} \downarrow x_0)}$ (definisi NOR)
$= ((((x_2 \downarrow x_1) \downarrow ((x_2 \downarrow 0) \downarrow (x_1 \downarrow 0))) \downarrow (x_0 \downarrow 0)) \downarrow ((x_3 \downarrow 0) \downarrow x_0)) \downarrow 0$ (konversi komplemen ke NOR)

Fungsi boolean dari persamaan (11) dapat disederhanakan menjadi persamaan (12) dengan cara sebagai berikut:

$$y_1(x_3, x_2, x_1, x_0) = \overline{x_3}\,\overline{x_2}\,x_1\,x_0 + \overline{x_3}\,x_2\,\overline{x_1}\,\overline{x_0} + \overline{x_3}\,x_2\,x_1\,\overline{x_0} + \overline{x_3}\,x_2\,x_1\,x_0 + x_3\,\overline{x_2}\,x_1\,x_0 + x_3\,x_2\,\overline{x_1}\,\overline{x_0} + x_3\,x_2\,x_1\,\overline{x_0} + x_3\,x_2\,x_1\,x_0$$

$= x_1\,x_0(\overline{x_3}\,\overline{x_2} + \overline{x_3}\,x_2 + x_3\,\overline{x_2} + x_3\,x_2) + x_2\,\overline{x_0}(\overline{x_3}\,\overline{x_1} + \overline{x_3}\,x_1 + x_3\,\overline{x_1} + x_3\,x_1)$ (hukum distributif)
$= x_1\,x_0 + x_2\,\overline{x_0}$ (sifat komplementer)

Operator-operator fungsi boolean dari persamaan (12) dapat dikonversi menjadi operator NOR seperti persamaan (13) dengan cara sebagai berikut:

$$y_1(x_3, x_2, x_1, x_0) = x_1\,x_0 + x_2\,\overline{x_0}$$

$= \overline{(\overline{x_1} \downarrow \overline{x_0}) + (\overline{x_2} \downarrow x_0)}$ (definisi NOR)
$= ((\overline{x_1} \downarrow \overline{x_0}) \downarrow (\overline{x_2} \downarrow x_0))$ (definisi NOR)
$= (((x_1 \downarrow 0) \downarrow (x_0 \downarrow 0)) \downarrow ((x_2 \downarrow 0) \downarrow x_0)) \downarrow 0$ (konversi komplemen ke NOR)

Fungsi boolean dari persamaan (14) dapat disederhanakan menjadi persamaan (15) dengan cara sebagai berikut:

$$y_0(x_3, x_2, x_1, x_0) = \overline{x_3}\,\overline{x_2}\,x_1\,\overline{x_0} + \overline{x_3}\,x_2\,x_1\,\overline{x_0} + x_3\,\overline{x_2}\,x_1\,\overline{x_0} + x_3\,x_2\,x_1\,\overline{x_0}$$

$= (\overline{x_3}\,\overline{x_2} + \overline{x_3}\,x_2 + x_3\,\overline{x_2} + x_3\,x_2)x_1\,\overline{x_0}$ (hukum distributif)
$= x_1\,\overline{x_0}$ (sifat komplementer)

Operator-operator fungsi boolean dari persamaan (15) dapat dikonversi menjadi operator NOR seperti persamaan (16) dengan cara sebagai berikut:

$$y_0(x_3, x_2, x_1, x_0) = x_1\,\overline{x_0}$$

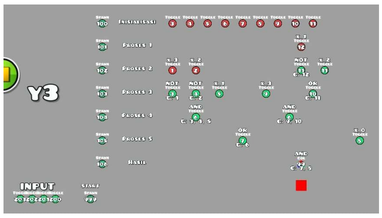
$= (x_1 \downarrow 0) \downarrow (\overline{x_0} \downarrow 0)$ (konversi perkalian boolean ke NOR)
$= (x_1 \downarrow 0) \downarrow x_0$ (konversi NOR ke komplemen)

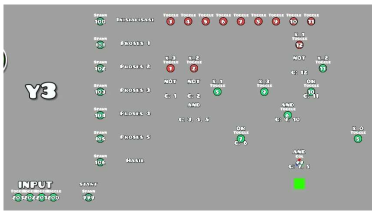
B. Gambar Representasi Ekspresi Boolean dalam Permainan Geometry Dash

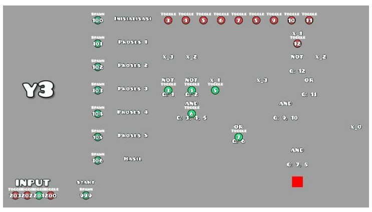
Pada bagian ini, semua persamaan mengacu pada bagian III-B. Terdapat delapan ekspresi boolean yang masing-masing memiliki tiga gambar: kondisi awal, kondisi setelah memproses masukan (1, 1, 1, 1), dan kondisi setelah memproses masukan (0, 0, 1, 0). Dua masukan ini dipilih karena hasilnya untuk setiap fungsi saling komplementer. Selain itu, konvensi pada gambar sama dengan pada bagian III-A dengan beberapa penambahan:

1) Label ID grup hanya menampilkan ID grup di bawah 100, yaitu ID grup untuk tahap tinisialisasi, proses, dan hasil.
2) *Spawn trigger* dengan ID target grup 999 berfungsi untuk memicu proses inisialisasi.
3) *Spawn trigger* dengan ID target 10X, X adalah digit desimal, digunakan untuk memicu *trigger* dalam thap inisialisasi (100), tahap proses X (10X), dan tahap hasil secara berurutan.
4) *Toggle trigger* dengan ID target grup 20N, N dapat bernilai 0, 1, 2, dan 3, yang dilabeli dengan kata "input" digunakan untuk menentukan masukan x_N—merah untuk nilai 0, hijau untuk nilai 1.
5) Terdapat kotak warna di bawah *color trigger* untuk menandakan hasil akhir—merah untuk nilai 0, hijau untuk nilai 1.

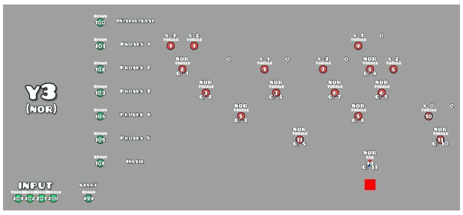
Fungsi boolean dari persamaan (6) direpresentasikan dengan tiga gambar berikut ini.

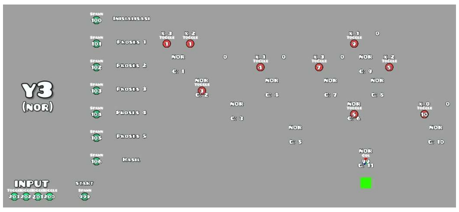


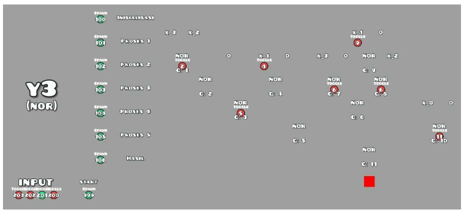




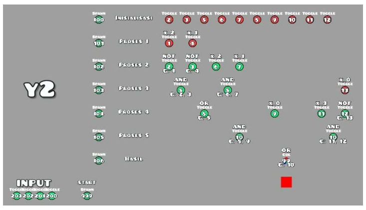
Fungsi boolean dari persamaan (7) direpresentasikan dengan tiga gambar berikut ini.

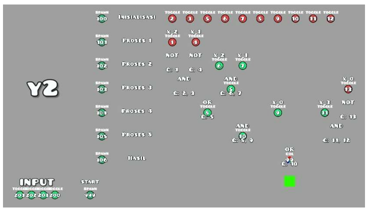


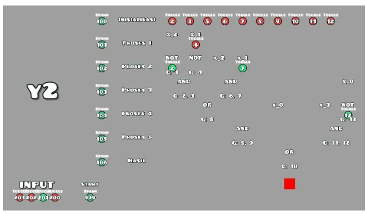




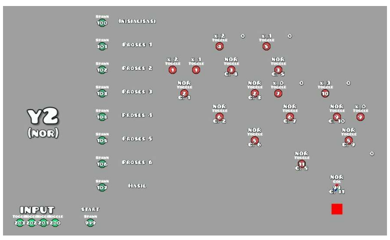
Fungsi boolean dari persamaan (9) direpresentasikan dengan tiga gambar berikut ini.

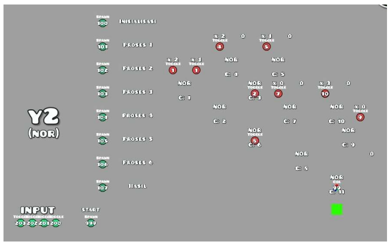


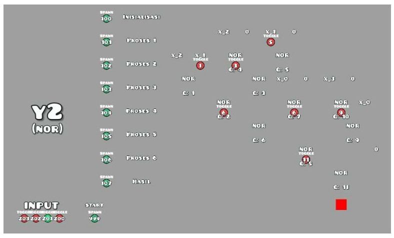




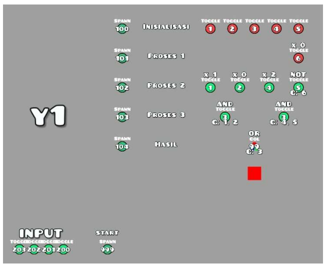
Fungsi boolean dari persamaan (10) direpresentasikan dengan tiga gambar berikut ini.

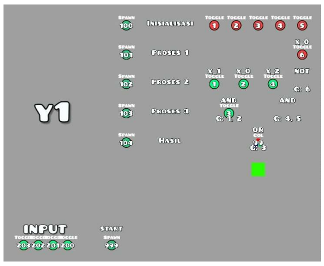


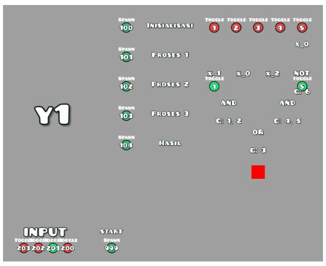




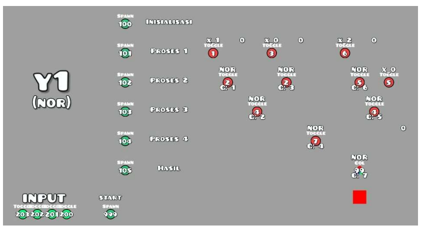
Fungsi boolean dari persamaan (12) direpresentasikan dengan tiga gambar berikut ini.

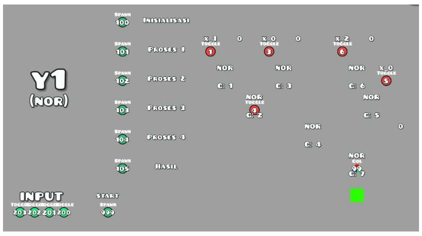


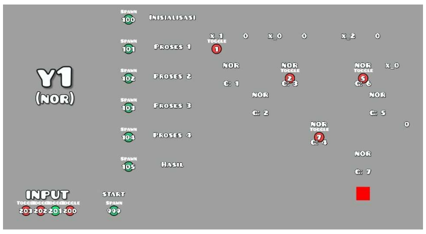




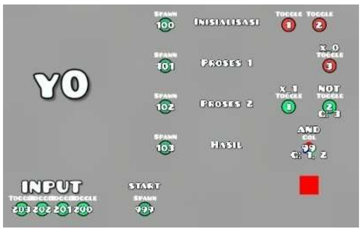
Fungsi boolean dari persamaan (13) direpresentasikan dengan tiga gambar di bawah ini.



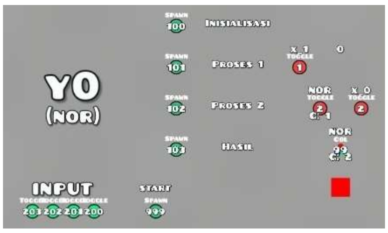


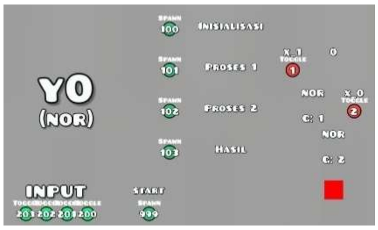


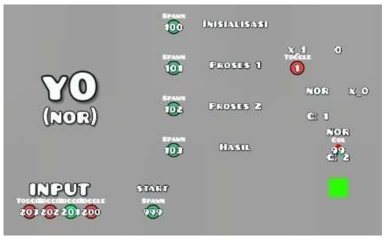
Fungsi boolean dari persamaan (15) direpresentasikan dengan tiga gambar berikut ini.



Fungsi boolean dari persamaan (16) direpresentasikan dengan tiga gambar berikut ini.







VII. UCAPAN TERIMA KASIH

Saya sebagai penulis berterima kasih kepada dosen-dosen pengampu mata kuliah IF2120 Matematika Diskrit Semester 1 Tahun 2020/2021, terutama Ibu Nur Ulfa Maulidevi yang telah memberikan bimbingan selama satu semester di kelas K-4. Saya juga berterima kasih kepada keluarga dan teman-teman yang telah memberikan umpan balik dan dukungan sehingga saya dapat menyelesaikan makalah ini dengan baik.

DAFTAR PUSTAKA

[1] J.J. O'Connor dan E.F. Robertson, "Lothar Collatz," *MacTutor*, 2006, diperoleh dari https://mathshistory.st-andrews.ac.uk/Biographies/Collatz/ pada tanggal 7 Desember 2020, 20.04 WITA

[2] Kenneth H. Rosen, *Discrete Mathemtics and Its Application*, edisi kedelapan, New York: McGraw-Hill Education, 2019, bab 4 dan 12

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, 3 Desember 2020

M. Abdi Haryadi. H (13519156)