

Nama : Dzaky Fattan Rizqullah

NIM : 13520003

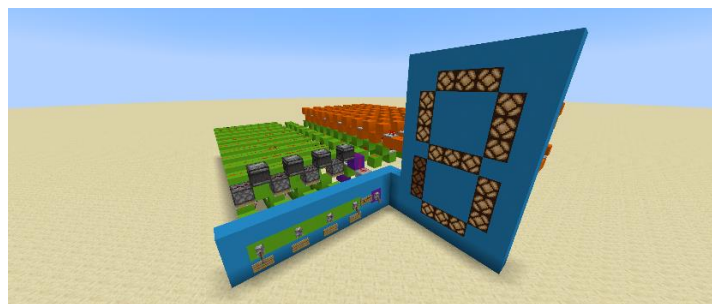
12. Turing Machine 101 – Solution

Perintah dari soal ini adalah untuk membuat simulasi *7-segment* menggunakan *video game Minecraft*. Implementasi yang dihasilkan dapat menerima nilai 0 hingga 9 sebagai masukan dan dapat menampilkan keluaran berupa angka yang sesuai dengan masukan. Format masukannya menggunakan *radix binary* yang direpresentasikan menggunakan *lever*. Implementasi logic gate tidak terlalu rumit, dan tidak menggunakan *logic gate adder* karena akan memakan tempat yang sangat besar, sehingga alternatif solusinya dengan menggunakan beberapa *logic gate AND* yang dimodifikasi.

Lever digunakan karena kedua *state* pada *lever* (mati dan hidup) dapat digunakan untuk merepresentasikan 0 dan 1. Empat buah *lever* diletakkan bersampingan secara horizontal, masing-masing merepresentasikan tiap bit pada *biner*. Setiap representasi *biner* dapat dianggap sebagai suatu *state lever* yang unik. Di sini lah *AND gate* berfungsi. *AND* sendiri akan memberikan *output* jika seluruh *input*-nya dinyalakan, dalam hal ini keseluruhan *input* harus dalam *state* hidup. *Gate* ini dapat dimodifikasi, sedemikian sehingga, untuk merepresentasikan suatu bilangan desimal sebagai *output*, maka pasti ada representasi *biner* yang unik untuk masing-masing bilangan desimal, untuk di-*input*-kan dengan suatu kombinasi *lever* yang unik pula namun bersesuaian dengan representasi bilangan *biner*-nya.

AND gate modifikasi ini akan memiliki empat buah *lever* sebagai *input*. Kemudian *gate* ini diduplikasi menjadi sepuluh buah, masing-masing merepresentasikan bilangan desimal 0 hingga 9, yang tentu saja membuat masing-masing *AND gate* akan memiliki susunan kombinasi *input* yang berbeda satu sama lain. Misalkan kombinasi *lever* dari kiri ke kanan adalah “mati – hidup – mati – hidup”, representasi *biner*-nya menjadi “0101” dan akan mengaktifkan salah satu *AND gate*, namun tidak akan mengaktifkan *AND gate* yang lain karena memerlukan kombinasi *lever* yang berbeda.

Ke-sepuluh *output* unik yang merepresentasikan 0 hingga 9 ini dikonversi menjadi *output* untuk *7-segment*. Implementasi *converter* ini sederhana. Setiap *segment* akan menerima *input* yang terpisah, sehingga akan ada 7 input. Setiap *output* 0 hingga 9 akan mengaktifkan satu atau lebih dari ketujuh *segment* tersebut sedemikian sehingga *7-segment* akan menampilkan angka yang bersesuaian dengan *input*-nya.



Gambar 1. Redstone *7-segment* yang menampilkan angka 9 dengan *input* “1001”.