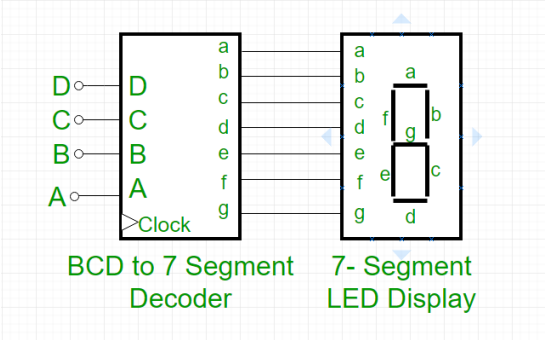
Penjelasan Implementasi Logic Gate
"Minekarnaugh"

Potret penulis saat mengerjakan spek

Penjelasan:

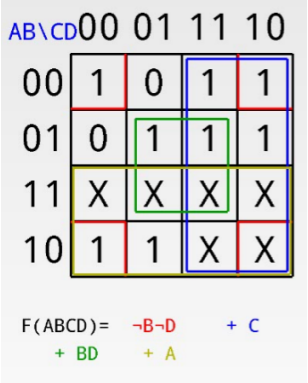
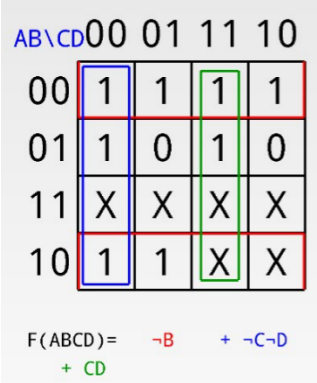
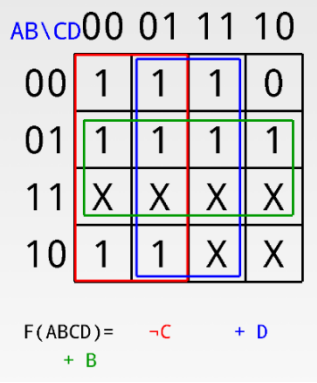
Rangkaian *redstone* yang dibuat bertujuan untuk menyalakan sebuah lampu 7-*segment* yang dibuat menggunakan *redstone lamp* berdasarkan *input* berupa angka biner (yang direpresentasikan melalui empat buah *lever*, dengan OFF sebagai 0 dan ON sebagai 1) dari 0 sampai dengan 9 (dalam biner, 0000 sampai dengan 1001).

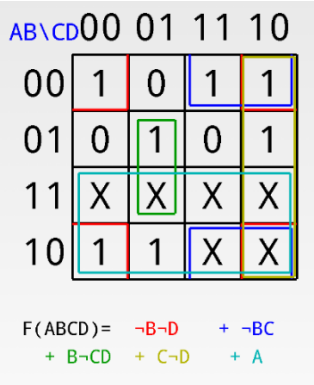
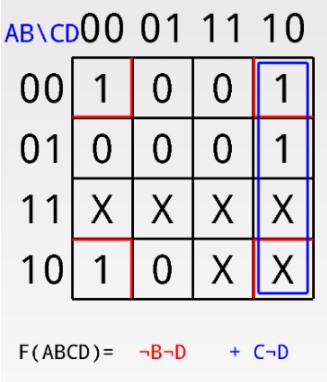
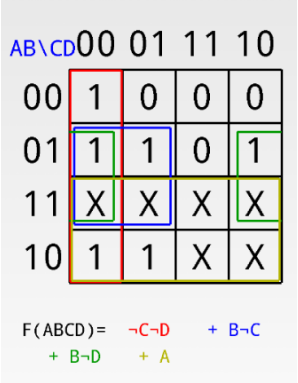
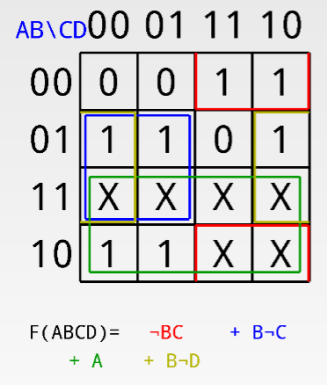
Terdapat 7 buah *output*, yang masing-masing nilainya bergantung pada 4 buah *input*. Jika digambarkan, *black box* dari rangkaian berbentuk sebagai berikut:



Sumber gambar: https://www.geeksforgeeks.org/bcd-to-7-segment-decoder/

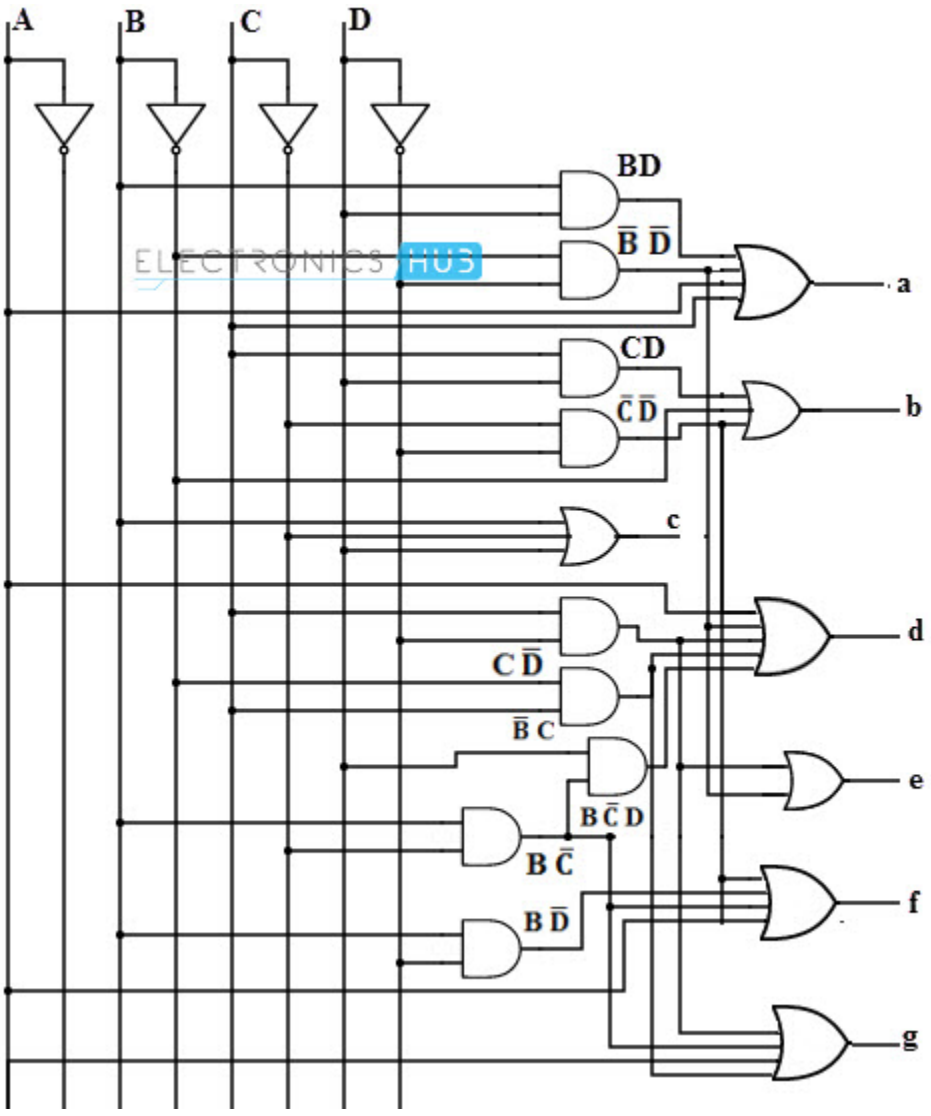
Untuk memudahkan pembuatan rangkaian, digunakan peta Karnaugh (K-map) yang menunjukkan kapan *segment* tertentu harus menyala dan kapan harus mati berdasarkan input yang diberikan. Hal tersebut dilakukan untuk setiap *segment*. Berikut peta-peta Karnaugh yang digunakan sebagai referensi dalam pembuatan, beserta dengan fungsi-fungsi untuk setiap *segment*:

Segment	Fungsi	Peta
a	F(ABCD) = ~B~D + C + BD + A	
b	F(ABCD) = ~B + ~C~D + CD	
c	F(ABCD) = ~C + D + B	

d	F(ABCD) = ~B~D + ~BC + B~CD + C~D + A	
e	F(ABCD) = ~B~D + C~D	
f	F(ABCD) = ~C~D + B~C + B~D + A	
g	F(ABCD) = ~BC + B~C + A + B~D	

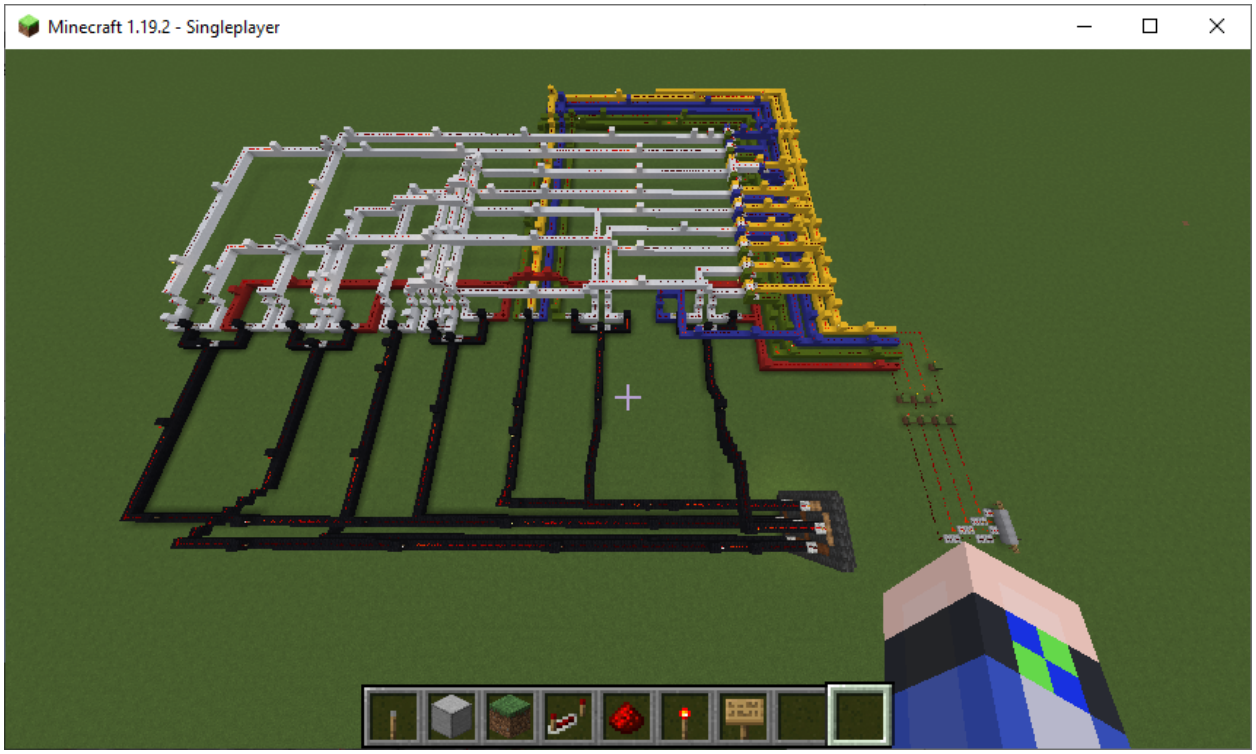
Sebagai contoh, *segment* a harus menyala untuk setiap kotak di peta Karnaugh yang bernilai 1, dan mati untuk setiap kotak yang bernilai 0 (untuk kotak X, dapat diabaikan). Dengan mengelompokkan kotak-kotak tersebut, kita dapat mendapatkan fungsi yang dibutuhkan, misalnya dengan mengelompokkan kita mendapatkan bahwa a selalu menyala ketika C dan A bernilai 1, dan seterusnya. Begitu juga untuk *segment-segment* lainnya.

Untuk beberapa *segment*, terdapat beberapa nilai produk yang sama, misal nilai ~B~D dibutuhkan di *segment* a, d, dan e. Supaya mempermudah pengerjaan, kita dapat membuat *logic gate* (tipe AND) khusus untuk setiap nilai produk ini (supaya tidak harus menarik dari setiap sumber awal untuk setiap output). Melakukan hal tersebut, diperoleh rangkaian berikut:



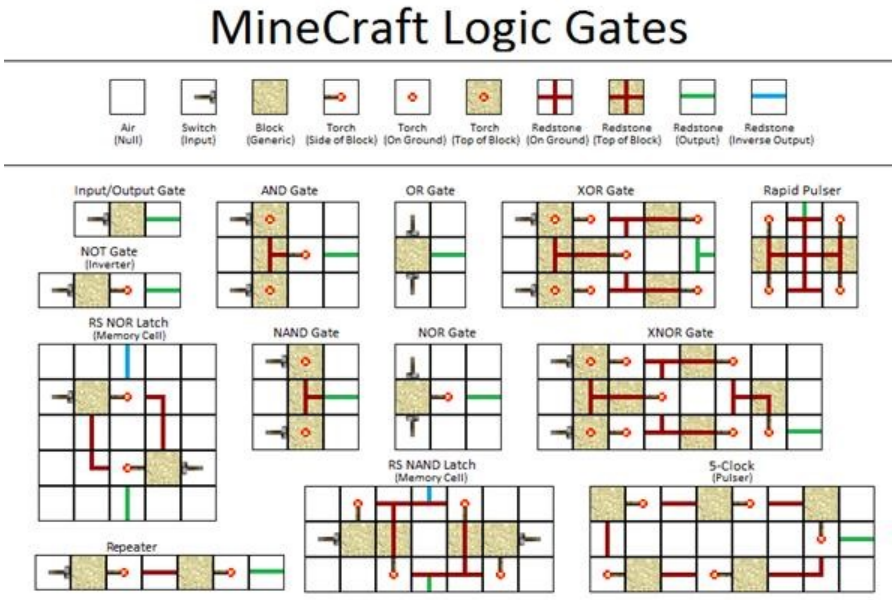
Sumber gambar: https://www.electronicshub.org/bcd-7-segment-led-display-decoder-circuit/

Setelah mendapatkan gambar tersebut, implementasi dilakukan di dalam Minecraft. Mayoritas kepusingan datang dari sini, karena sistem *redstone* itu Naudzubillah Min Dzalik.

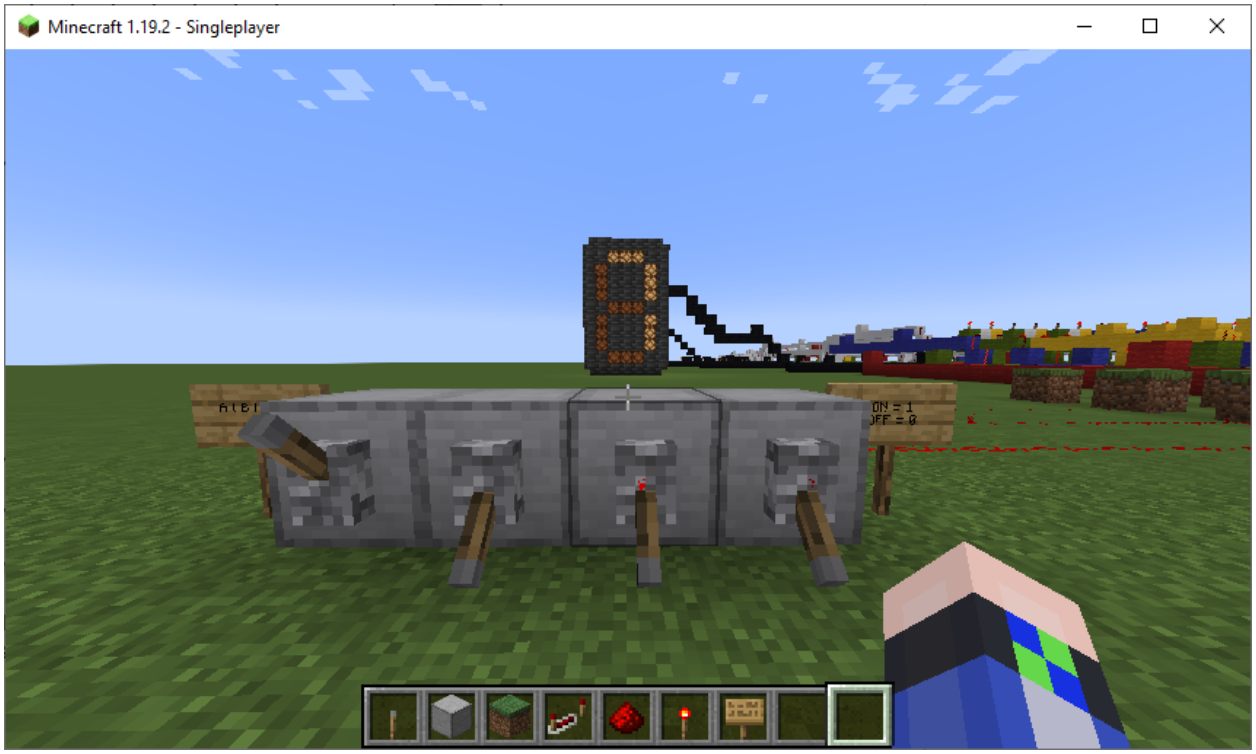


Nggak lagi-lagi.

Dalam melakukan translasi dari diagram rangkaian ke rangkaian *redstone*, digunakan referensi berikut:



Sumber gambar:
https://minecraft.wonderhowto.com/news/redstone-logic-gates-mastering-fundamental-building-blocks-for-creating-game-machines-0135063/



Hasil akhir; ditampilkan adalah kondisi 7-*segment* untuk input 7 (0111).

World dapat dimainkan dengan cara menaruh folder yang telah di-*unzip* di folder .minecraft/saves/ (lokasi folder berbeda-beda berdasarkan pengaturan user serta sistem operasi yang digunakan).

Catatan:

Layout *logic gate* produk di world menggunakan diagram lain yang tidak lagi digunakan karena berbeda dengan peta-peta Karnaugh yang digunakan.

(https://www.electricaltechnology.org/2018/05/bcd-to-7-segment-display-decoder.html)