



Perancangan Sistem Digital

# FOUR LANE JUNCTION TRAFFIC LIGHT CONTROLLER

BINTANG MARSYUMA R.

FAHREZY H

ZALFY PUTRA REZKY

# Latar Belakang

Lampu lalu lintas adalah alat berupa lampu yang memberikan sinyal-sinyal untuk mengendalikan arus lalu lintas. Pada umumnya, lampu lalu lintas terpasang di persimpangan jalan, tempat penyeberangan pejalan kaki (zebra cross), dan tempat arus lalu lintas lainnya. Lampu lalu lintas menginstruksikan pejalan kaki dan pengemudi kendaraan dengan memberikan tiga sinyal lampu dengan periode tertentu yaitu merah, kuning, dan hijau.

Dalam meningkatkan ketertiban lalu lintas dan membantu para pengguna jalan pada keadaan darurat yang dapat menyebabkan terganggunya arus lalu lintas, kami menentukan untuk mengembangkan program Four Lane Junction Traffic Light Controller dalam VHDL agar meningkatkan kenyamanan dan ketertiban lalu lintas bagi para pengguna jalan serta membantu Pihak Kepolisian dalam mengawasi arus lalu lintas.


# Tujuan Proyek



Mengimplementasikan  
VHDL

Membantu kontrol lalu  
lintas di jalan raya

Peningkatan ke-  
Efisien dan  
kenyamanan dalam  
pengguna jalan



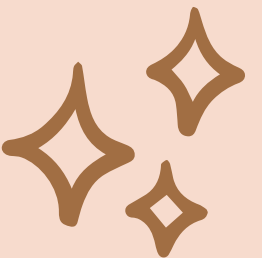
# Implementasi



Dalam kode ini digunakan input clear, clock, dan mode dalam bentuk std\_logic, serta condition dalam bentuk integer. Sedangkan outputnya adalah green, yellow, dan red dalam bentuk std\_logic\_vector. Kode ini terdiri dari 8 state atau kondisi. Dengan state S0 sebagai state pertama dan S8 sebagai state terakhir. Pada masing-masing state dilakukan modifikasi pada output dan juga sinyal nextState.



Rangkaian ini juga terbagi dalam dua bagian besar yaitu proses seq dan proses comb. Pada proses seq dilakukan penentuan mode lampu lalu lintas, yaitu otomatis atau manual. Sedangkan pada proses comb dilakukan penentuan nyala lampu berdasarkan arah mata angin dan arah jarum jam.



# Percobaan

Untuk melakukan percobaan pada program ini, kami menggunakan software ModelSim dan melakukan compiling serta error checking menggunakan Quartus Prime. Pertama, program dijalankan pada mode otomatis dengan memasukkan value '0' pada input "mode". Lalu, memasukkan value '0' pada input "clr" karena belum memerlukan reset dan value '0' juga pada input "condition" karena bukan pada mode manual. Hal ini akan menampilkan urutan lampu traffic light, signal switch, status, nextState, redTime, dan yellowTime. Percobaan diteruskan dengan melakukan run berulang kali untuk setiap state. Untuk melakukan reset atau kembali ke state awal maka dapat memasukkan value '1' pada input "clr" sehingga program masuk ke state S0.

# Percobaan

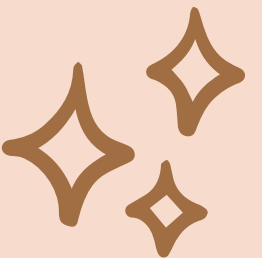


Kedua, program dijalankan pada mode manual dengan memasukkan value '0' pada input "mode". Input "clr" tetap bernilai 0 karena belum memerlukan reset. Mode ini menggunakan lima input "condition" untuk menentukan state yang akan dijalankan oleh program. Berikut algoritma dari condition, State yang dijalankan juga akan menampilkan urutan traffic light sesuai state table, signal switch, status, state, nextState, redTime, dan yellowTime.



## Algoritma Condition:

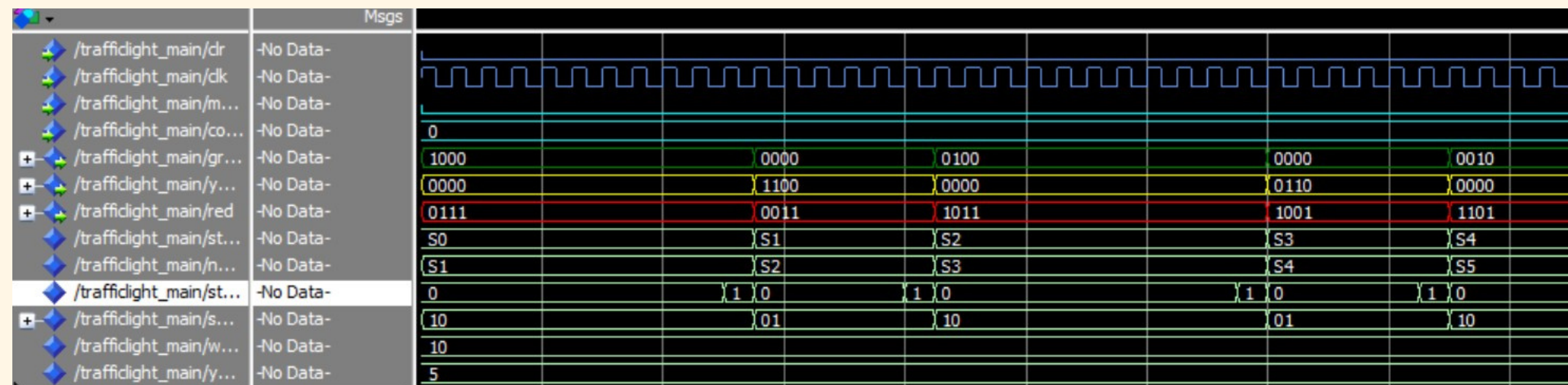
- Condition '0' → menjalankan state S8
- Condition '1' → menjalankan state S0
- Condition '2' → menjalankan state S2
- Condition '3' → menjalankan state S4
- Condition '4' → menjalankan state S5





# Hasil Percobaan 1

Pada percobaan pertama, program dijalankan pada mode otomatis. ModelSim memberikan tampilan seperti berikut.



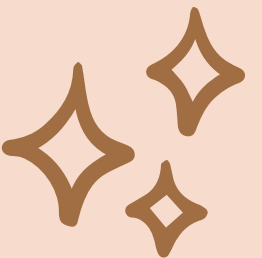
# Hasil Percobaan 1



Dapat terlihat pada hasil simulasi di slide sebelumnya bahwa input "clr" selalu 0 karena tidak melakukan reset dan input clock sebagai clock signal. Selain itu, mode bernilai 0 karena masuk ke mode otomatis. Setiap state memiliki urutan output green, yellow, dan red yang berbeda-beda sesuai dengan state table.



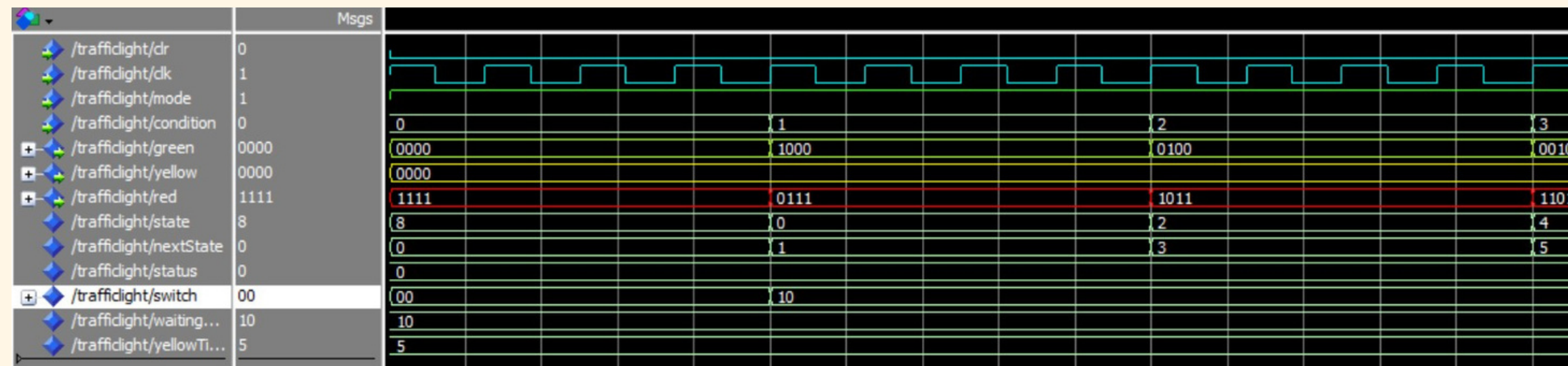
Selain itu, terdapat signal state yang menunjukkan state berapa yang telah dijalankan dan signal nextState yang menunjukkan state berapa yang akan dijalankan berikutnya. Lalu, terdapat signal status bernilai 0 pada semua state karena menggunakan clock rising edge untuk mengeksekusi proses  $state \leq nextState$ . Terdapat juga signal switch untuk menampilkan counter menggunakan dual 7-Segment Display dengan komponen lightcounter.





# Hasil Percobaan 2

Pada percobaan kedua, program dijalankan pada mode manual. ModelSim memberikan tampilan seperti berikut.



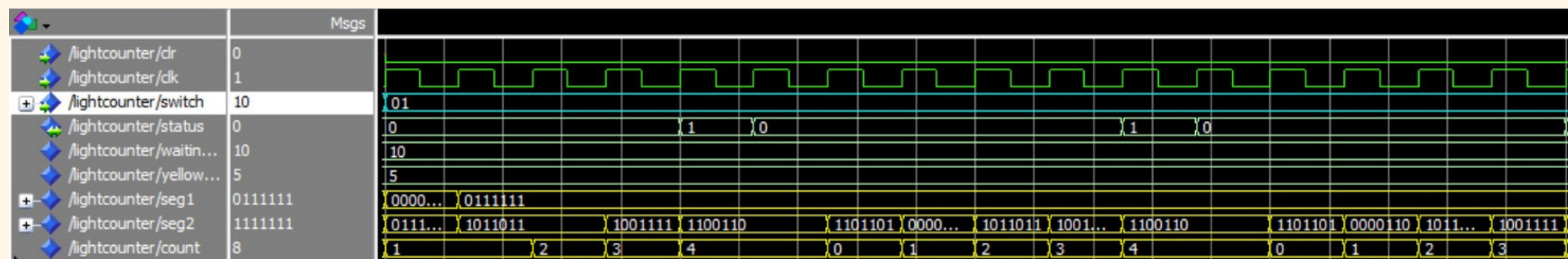
# Hasil Percobaan 2

Dapat terlihat pada hasil simulasi di slide sebelumnya bahwa input "clr" selalu 0 karena tidak melakukan reset dan input clock sebagai clock signal. Selain itu, mode bernilai 1 karena masuk ke mode manual. Setiap condition akan menampilkan state yang berbeda-beda sesuai algoritma.

Setiap state akan menampilkan urutan output green, yellow, red yang sesuai dengan state table. Selebihnya sama seperti pada mode otomatis, di mana pada hasil simulasi juga terdapat signal state, nextState, status, dan switch untuk menggunakan Dual 7-Segment Display.

# Percobaan Komponen Lightcounter

Pada kedua percobaan tersebut, terdapat signal switch dan status yang berbeda-beda. Cara kerja dari kedua signal ini pun ada pada komponen lightcounter. Komponen ini berfungsi untuk menjalankan counter dan dual 7-Segment Display untuk menampilkan counter tersebut. Dengan ModelSim, diperoleh hasil simulasi seperti berikut.



# ▢ Percobaan Komponen Lightcounter

● ○ ○

Pada tampilan di slide sebelumnya, dapat terlihat clr bernilai 0 dan menggunakan clock signal. Value dari signal switch dan status berdasarkan state yang dijalankan. Jika switch bernilai "10", status bernilai 0 dan counter melakukan increment hingga counter = waitingTime di mana akan mengubah status menjadi bernilai 1.

● ○ ○

Jika switch bernilai "01", status bernilai 0 dan counter melakukan increment hingga counter = yellowTime di mana akan mengubah status menjadi bernilai 1. Tiap value count juga akan ditampilkan secara fisik oleh 7-Segment Display.

✧ ✧ ✧



# Analisis

Hasil percobaan yang diperoleh dari simulasi program dapat ini menggunakan software ModelSim kemudian melakukan error checking menggunakan Quartus Prime. untuk program nya sendiri Four Lane Junction Traffic Light Controller ini dijalankan di mode otomatis dan mode manual. Di mode otomatis kita memasukkan value 0 di input mode. ini dibuat untuk membuat urutan lampu traffic nya dan di mode otomatis ini terdapat signal switch yang berguna untuk menampilkan counter menggunakan dual 7-Segment Display tujuanya adalah sebagai komponen lightcounter sehingga dapat menampilkan lampu warna merah, kuning dan hijau secara otomatis. sehingga tidak selalu pihak dari kepolisian melakukan pemantauan secara terus menerus

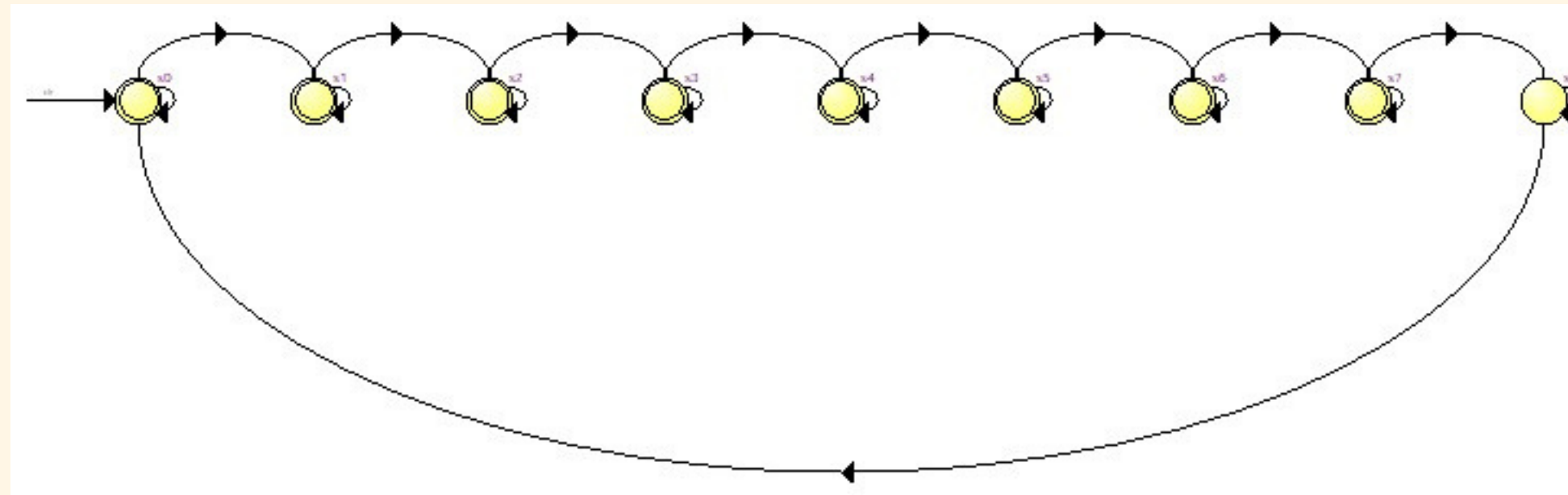


# Analisis

Kedua program ini dijalankan menggunakan mode manual. pada program ini dibuat agar pihak kepolisian dapat memberhentikan lampu nya ketika terjadinya keadaan darurat yang terjadi di lalu lintas sehingga tidak mengganggu pihak pengguna jalan dan juga dapat menertibkan jalan jalan konsepnya sama juga dengan mode manual itu sama saja dengan memasukkan value 0 di input mode. input clr juga tetep bernilai 0 karena belum memerlukan reset. dan mode ini menggunakan lima input condition.condition nya dapat kita lihat disini pada saat condition 1 menjalankan state S8 kemudian condition 2 menjalankan state 0 berlanjut ke state S2 dan State S4 dan terakhir di condition 5 menjalankan di state S5

# Analisis

State Diagram



State yang dijalankan nantinya akan menampilkan sesuai urutan yaitu state table, signal switch, status, state, nextState, redTime dan YellowTime. Jadi terakhir ini dibuat untuk mempermudah pihak kepolisian dan membuat para pengguna jalan lebih aman dan nyaman.

# Kesimpulan

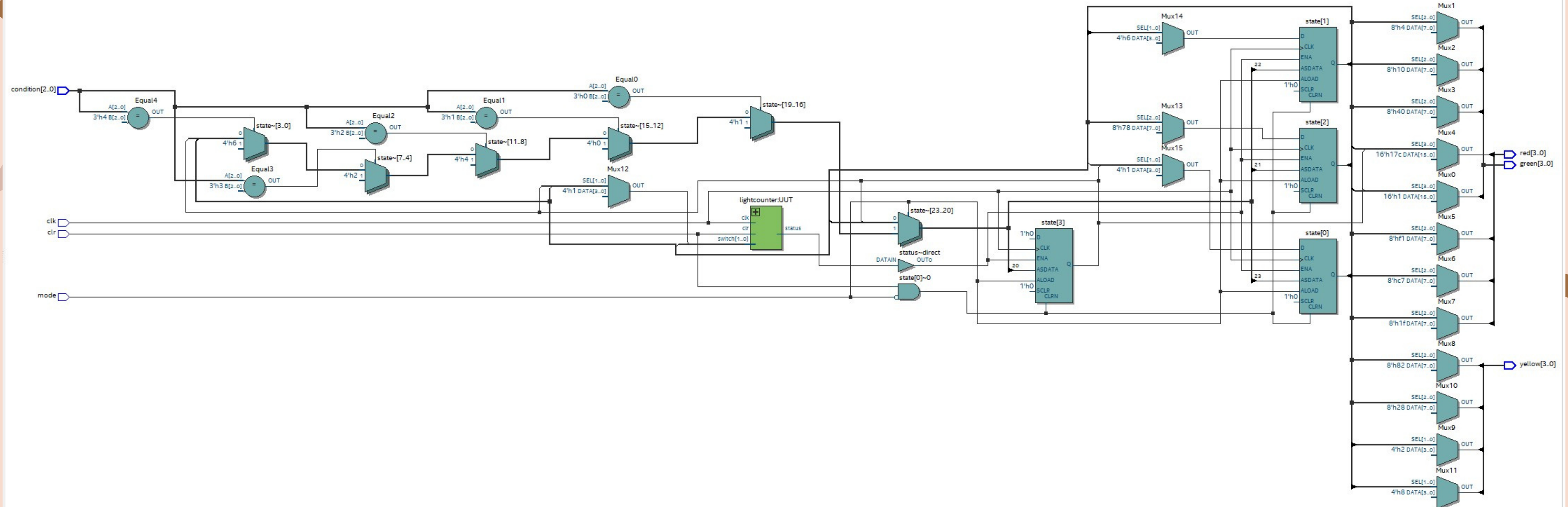
Kelompok kami telah mengembangkan Four Lane Junction Traffic Light Controller dalam VHDL dengan mengimplementasikan logic diagram dari sistem, menerapkan State Table untuk tiap kondisi persimpangan jalan, komponen Counter untuk menentukan interval tiap state, dan 7-Segment Display untuk menampilkan Counter. Selain itu, kelompok kami telah melakukan simulasi terhadap kode VHDL, menghasilkan skematik dari proyek, dan membuat RTL Viewer untuk menampilkan bentuk hardware dari kode VHDL.

# Referensi

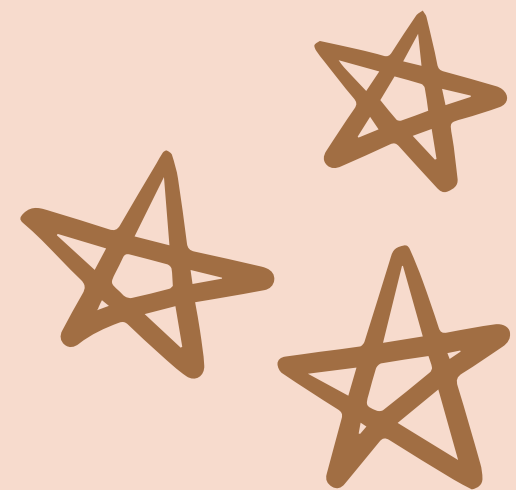
Pringgodigdo. 1973. Ensiklopedia umum. Jakarta: Yayasan Kanisius.

Geeksforgeeks. "What is the meaning and importance of traffic signal lights?". Geeksforgeek.org. Accessed: Dec. 10, 2022. [Online]. Available: <https://www.geeksforgeeks.org/what-is-the-meaning-and-importance-of-traffic-signal-lights/>

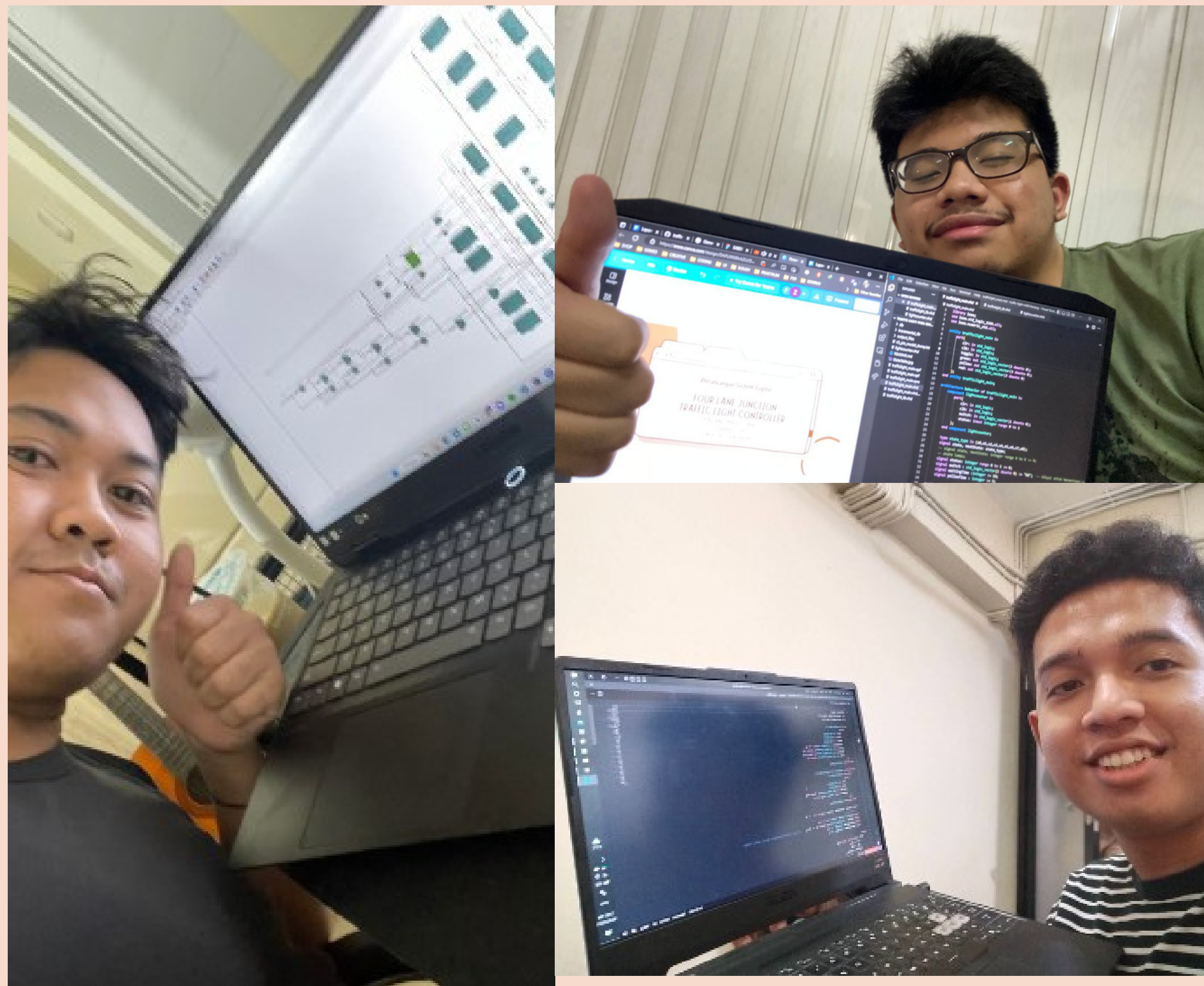
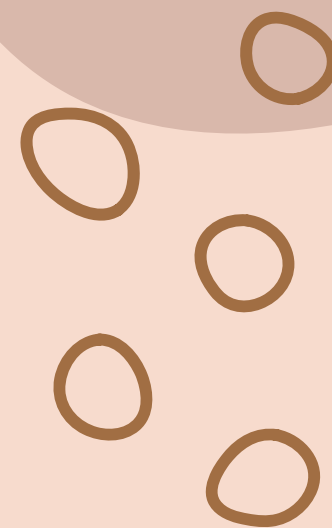
# Project Schematic







# Dokumentasi



Terima Kasih

