

## Laporan Tugas 3 : Lampu Geser

Nama : Elkhani Julian Brilianshah

NIM : 13219059

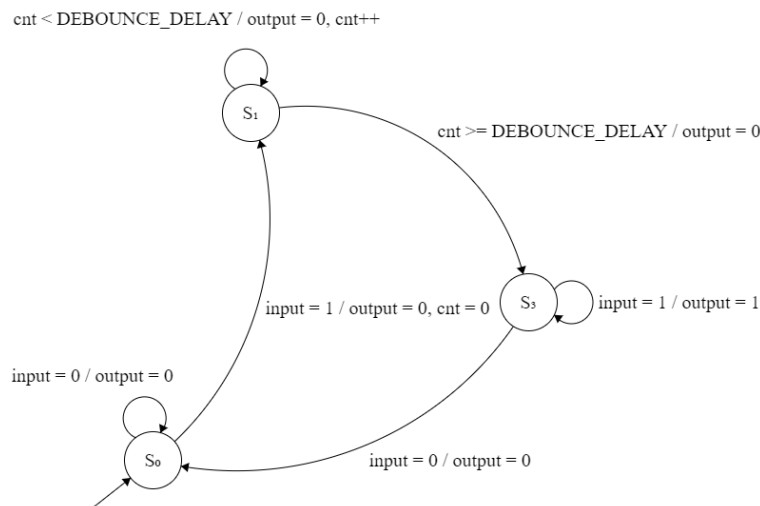
### Daftar Gambar

Gambar 1 Diagram FSM Debounce.....	1
Gambar 2 Simulasi FSM Debounce.....	2
Gambar 3 Diagram FSM Rising Edge.....	2
Gambar 4 Simulasi FSM Rising Edge .....	3
Gambar 5 Diagram FSM Slide .....	3
Gambar 6 Simulasi FSM Slide.....	4
Gambar 7 Skema FSM Kaskade.....	4
Gambar 8 Skematik Rangkaian .....	5
Gambar 9 Bentuk Fisik Rangkaian.....	5

## 1 Desain FSM

### 1.1 FSM Debounce

Untuk melakukan *debouncing*, diperlukan sebuah FSM yang dapat mengabaikan beberapa sampel masukan dari tombol setelah tombol ditekan untuk pertama kalinya. Digaram untuk FSM tersebut adalah seperti pada gambar berikut.

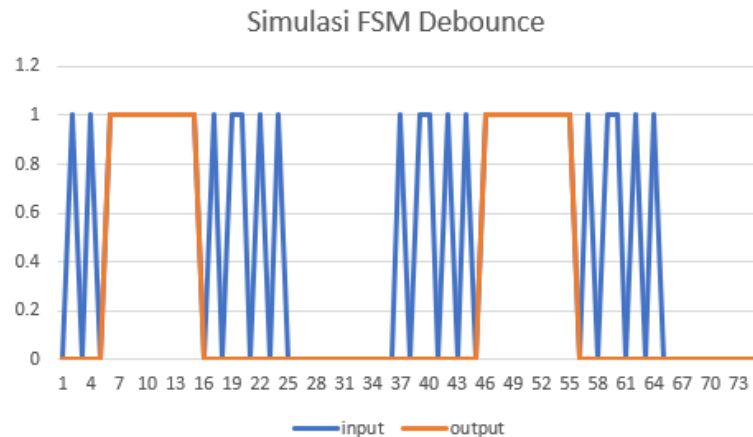


Gambar 1 Diagram FSM Debounce

*Debounce* FSM memiliki 3 state yaitu  $S_0$ ,  $S_1$ , dan  $S_2$  yang secara berurutan memiliki alias *IDLE\_STATE*, *WAITING\_STATE*, dan *READING\_STATE*. *IDLE\_STATE* adalah *state* yang merepresentasikan keadaan awal dan keadaan ketika tidak ada tekanan tombol dari pengguna. *WAITING\_STATE* adalah *state* yang merepresentasikan keadaan ketika tombol baru saja ditekan dan menghasilkan sinyal *bouncing*. *READING\_STATE* adalah *state* yang merepresentasikan keadaan setelah sinyal *bouncing* sudah tidak ada dan pembacaan keadaan tombol dapat dilakukan. Selain itu, FSM ini memiliki satu masukan yaitu input,

satu keluaran yaitu output, dan satu variabel yaitu cnt. Masukkan input memodelkan sinyal dari tombol yang menjadi masukan FSM sedangkan keluaran output adalah sinyal yang merepresentasikan hasil *debouncing* dari sinyal input. Variabel cnt digunakan untuk merepresentasikan *state-state* lain yang digunakan untuk membuat jeda yang dibutuhkan untuk menunggu *bouncing* tombol.

Simulasi FSM dilakukan menggunakan sebuah program C yang menghasilkan file csv yang berisikan nilai masukan dan keluaran dari FSM. Data pada file csv tersebut kemudian dapat digambar menggunakan aplikasi seperti MS Excel untuk mendapatkan gambaran yang lebih jelas tentang masukan dan keluaran FSM. Hasil simulasi adalah seperti pada gambar berikut.

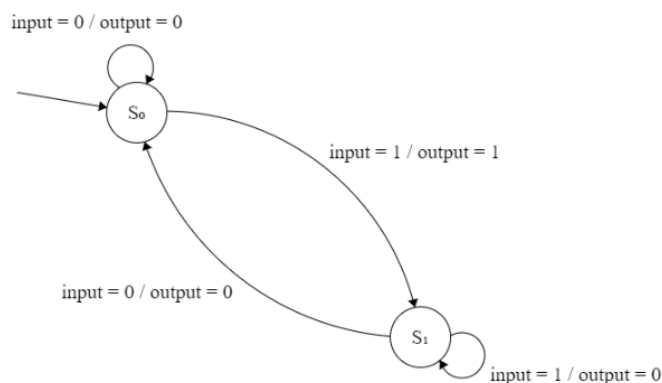


Gambar 2 Simulasi FSM Debounce

Dapat diamati bahwa, berdasarkan simulasi, *Debounce* FSM mampu memberikan keluaran yang menghilangkan efek *bouncing* pada tombol.

## 1.2 FSM Deteksi Rising Edge

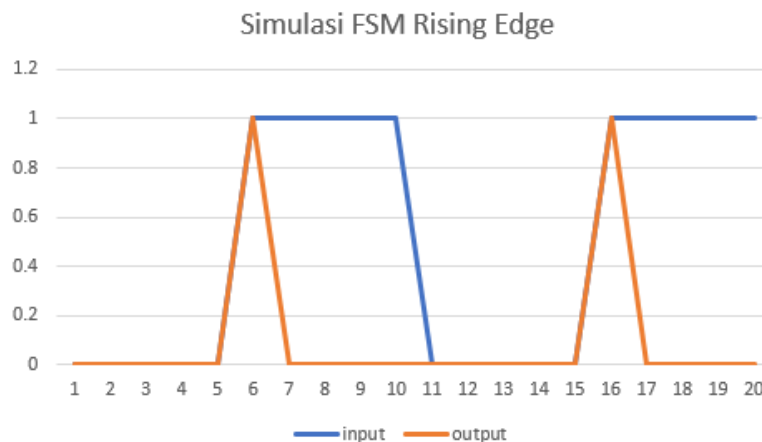
Untuk mendeteksi *rising-edge* pada tombol, diperlukan sebuah FSM yang dapat memberikan keluaran yang bernilai 1 jika sampel terbaru masukan bernilai satu dan sampel sebelumnya bernilai 0. Digaram untuk FSM tersebut adalah seperti pada gambar berikut.



Gambar 3 Diagram FSM Rising Edge

*Rising Edge* FSM memiliki 2 state yaitu  $S_0$  dan  $S_2$  yang secara berurutan memiliki alias LOW\_STATE dan HIGH\_STATE. LOW\_STATE adalah *state* yang merepresentasikan keadaan ketika tombol sedang tidak ditekan sedangkan HIGH\_STATE adalah *state* yang merepresentasikan keadaan ketika tombol sedang ditekan. Masukkan input memodelkan sinyal dari tombol yang menjadi masukan FSM sedangkan keluaran output adalah sinyal yang merepresentasikan hasil deteksi *rising edge* dari sinyal input.

Simulasi FSM dilakukan menggunakan sebuah program C yang menghasilkan file csv yang berisikan nilai masukan dan keluaran dari FSM. Data pada file csv tersebut kemudian dapat digambar menggunakan aplikasi seperti MS Excel untuk mendapatkan gambaran yang lebih jelas tentang masukan dan keluaran FSM. Hasil simulasi adalah seperti pada gambar berikut.

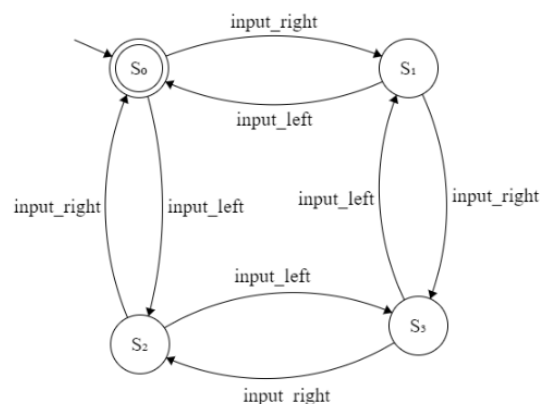


Gambar 4 Simulasi FSM Rising Edge

Dapat diamati bahwa, berdasarkan simulasi, *Rising Edge* FSM mampu mendeteksi *rising edge* masukan.

### 1.3 FSM Slide

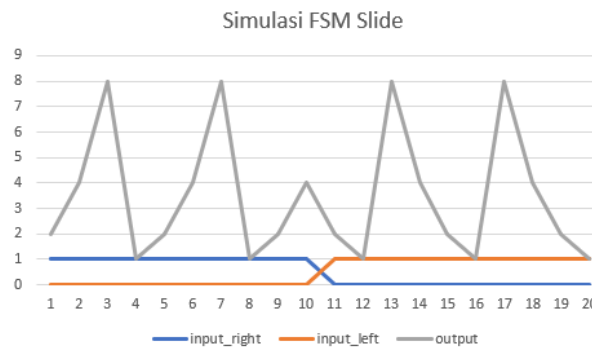
Untuk melakukan pergeseran lampu, diperlukan sebuah FSM yang dapat memberikan keluaran berupa penanda posisi LED yang menyala berdasarkan masukan yang diberikan kepada FSM. Diagram untuk FSM tersebut adalah seperti pada gambar berikut.



Gambar 5 Diagram FSM Slide

*Slide* FSM memiliki 4 state yaitu  $S_0$ ,  $S_1$ ,  $S_2$ , dan  $S_3$  yang secara berurutan merepresentasikan nyala LED paling kiri hingga nyala LED paling kanan. FSM ini memiliki dua masukan yaitu `input_left` yang merepresentasikan perintah untuk menggeser LED yang menyala ke kiri dan `input_right` yang merepresentasikan perintah untuk menggeser LED yang menyala ke kanan. Keluaran FSM ini adalah sama dengan *state* dari FSM ini sehingga FSM ini merupakan FSM Moore. Setiap *state* pada FSM ini memiliki nilai yang dalam biner akan menentukan nyala LED.

Simulasi FSM dilakukan menggunakan sebuah program C yang menghasilkan file csv yang berisikan nilai masukan dan keluaran dari FSM. Data pada file csv tersebut kemudian dapat digambar menggunakan aplikasi seperti MS Excel untuk mendapatkan gambaran yang lebih jelas tentang masukan dan keluaran FSM. Hasil simulasi adalah seperti pada gambar berikut.

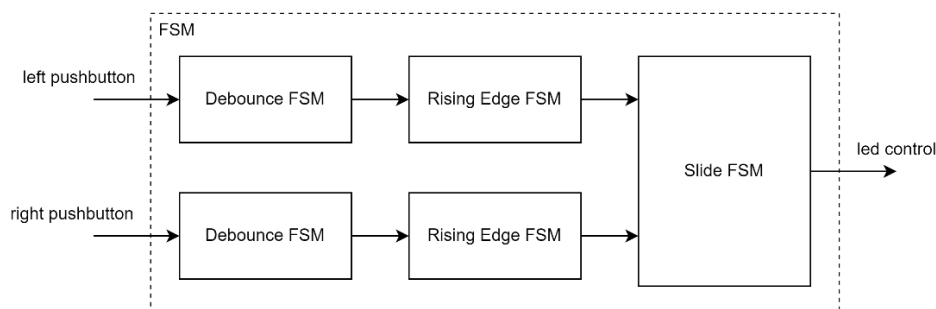


Gambar 6 Simulasi FSM Slide

Dapat diamati bahwa, berdasarkan simulasi, *Slide* FSM mampu memberikan keluaran yang sesuai berdasarkan masukan yang diberikan.

## 2 Desain Software

Pembacaan masukan dan penampilan LED pada tugas ini dilakukan secara periodik setiap 10ms dalam bentuk sebuah task. Pada task tersebut, program akan membaca masukan tombol dan memasukkan hasil pembacaan tersebut ke sebuah FSM. *State* dari FSM tersebut kemudian akan dijadikan sebagai sebuah nilai yang menentukan nyala LED. FSM yang dimaksud merupakan gabungan dari FSM-FSM yang telah dibahas pada bab sebelumnya dengan konfigurasi kaskade seperti pada gambar berikut.

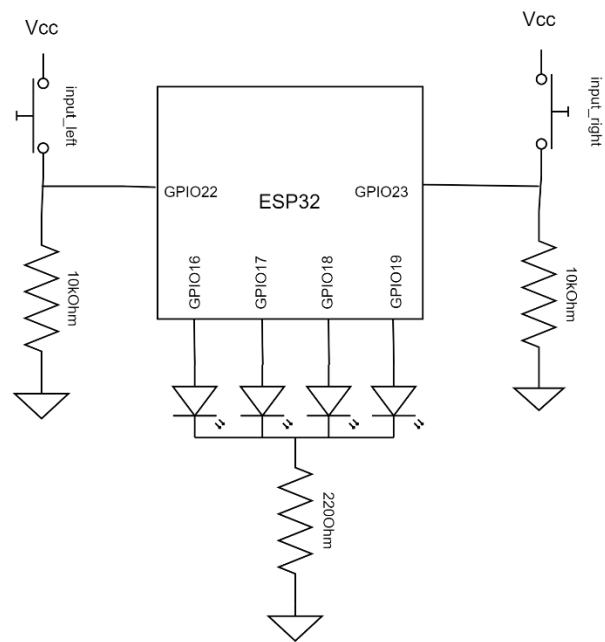


Gambar 7 Skema FSM Kaskade

Source Code tugas dapat dilihat di <https://github.com/ubbeg2000/tugas3-el4121>

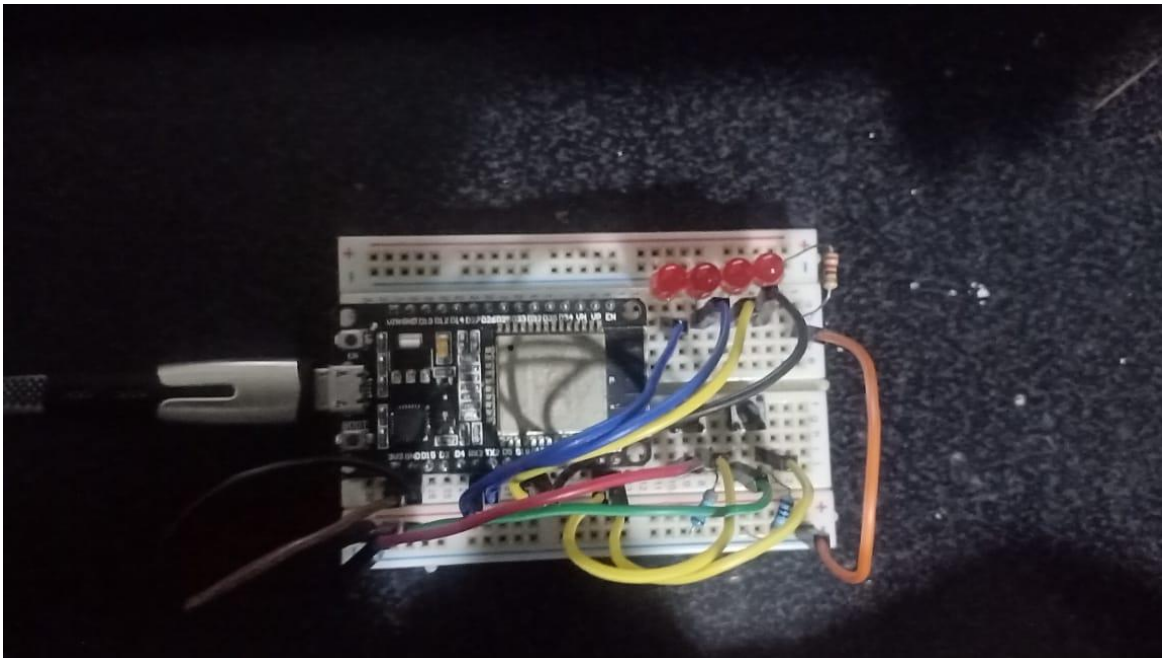
### 3 Desain Hardware

Skematik rangkaian adalah seperti pada gambar berikut.



Gambar 8 Skematik Rangkaian

Gambar rangkaian adalah seperti pada gambar berikut.



Gambar 9 Bentuk Fisik Rangkaian

Video demonstrasi dapat dilihat di <https://www.youtube.com/watch?v=TaR-ZnYPZ2I>