FUNDAMENDTAL OF DIGITAL SYSTEM FINAL PROJECT REPORT
DEPARTMENT OF ELECTRICAL ENGINEERING
UNIVERSITAS INDONESIA

ADJUSTABLE CROSSWALK LAMP

GROUP B-1

DIVA HANA PRILIA	**2006529543**
GEMILANG BAGAS R	**2006535205**
MUHAMMAD IRSYAD F	**2006468850**
RAIN ELGRATIO S H L G	**2006577574**

PREFACE

Pertama-tama kami ingin berterimakasih kepada para asisten praktikum yang sudah membimbing kami selama praktikum Perancangan Sistem Digital sehinga kami bisa membuat proyek akhir ini. Kami membuat proyek yang Bernama *Adjustable Crosswalk Lamp* untuk proyek akhir praktikum. *Adjustable Crosswalk Lamp* ini kami buat sebagai pengaplikasian ilmu-ilmu yang sudah kami dapatkan dari materi kelas serta kelas praktikum Perancangan Sistem Digital.

Dalam penyusunan laporan ini, penulis sangat berterima kasih kepada Muhammad Naufal Faza selaku pembimbing kelompok kami yang telah senantiasa memberikan saran dan bimbingannya kepada kelompok B-1 selama pengerjaan proyek akhir berlangsung. Kami menyadari bahwa laporan ini tidak terlepas dari kekurangan, sehingga kami sangat terbuka terhadap kritik dan saran dari berbagai pihak demi perbaikan kedepannya. kami berharap dengan adanya laporan ini dapat meningkatkan pemahaman pembaca mengenai proyek akhir yang kami buat serta dapat memberikan manfaat bagi dunia pendidikan dan teknologi.

Depok, December 10, 2022

Group B-1

TABLE OF CONTENTS

CHAPTER 1: INRODUCTION1

1.1 Background

1.2 Project Description

1.3 Objectives

1.4 Roles and Responsibilities

CHAPTER 2: IMPLEMENTATION4

2.1 Equipment

2.2 Implementation

CHAPTER 3: TESTING AND ANALYSIS4

3.1 Testing

3.2 Result

3.3 Analysis

CHAPTER 4: CONCLUSION

REFERENCES

APPENDICES

Appendix A: Project Schematic

Appendix B: Documentation

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

Tempat penyeberangan pada jalan atau yang kerap disebut dengan *zebra cross* merupakan sebuah sarana yang disediakan guna memudahkan perjalanan dari para pejalan kaki. Area ini menyediakan sebuah tempat dimana orang-orang sekitar diharapkan akan lebih berhati-hati, beberapa zebra cross bahkan sudah dilengkapi dengan tombol yang akan mengeluarkan suara tanda akan ada yang menyebrang disertai dengan lampu lalu lintas yang akan menjadi merah bagi pengguna jalan utama dan hijau untuk pengguna jalan penyebrangan. Hal ini tentunya bagus karena dengan adanya *crosswalk lamp* ini proses penyeberangan bagi pejalan kaki dibuat menjadi lebih nyaman, namun mode yang ditawarkan *crosswalk lamp* pada umumnya lumayan terbatas.

Untuk proyek akhir PSD, kelompok kami memutuskan untuk membuat sebuah *crosswalk lamp* dimana *crosswalk* ini memiliki fitur yang dapat diatur sesuai dengan kebutuhan pengguna jalan penyebrangan. Beda dengan *crosswalk* lamp pada umumnya yang hanya memiliki 1 mode untuk tiap pengguna yang kurang inklusif, *crosswalk* yang kami rancang menyediakan beberapa pilihan yang kami harap dapat memudahkan menyesuaikan mode sesuai dengan si pengguna.

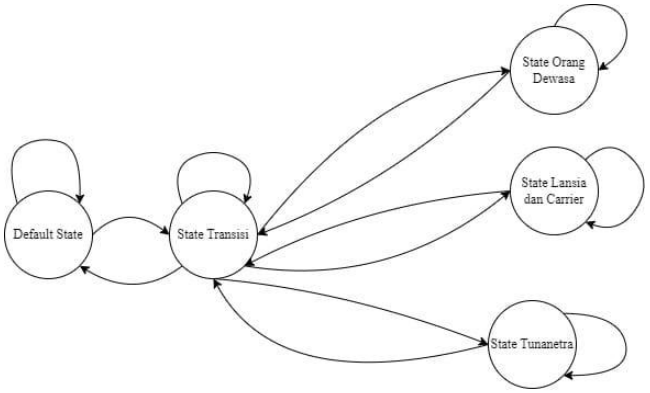
1.2 PROJECT DESCRIPTION

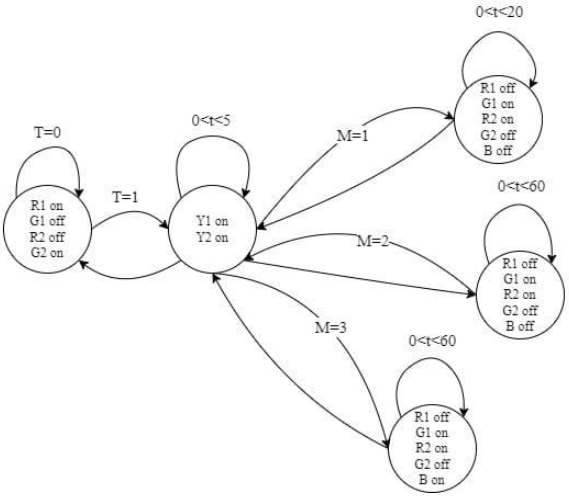
Adjustable crosswalk lamp yang kami rancang menggunakan model komputasi *Finite-state machine* , mealy machine yang menghasilkan output dari input baru dan input sebelumnya.Hal ini diperlukan untuk proyek kami karena jauh lebih efektif untuk diterapkan dibanding moore mempertimbangkan output untuk pengguna akan dipengaruhi oleh beberapa state yang bekerja seperti domino.

Proyek Akhir ini dideskripsikan dengan Bahasa VHDL dan kami mendesain 3 file yang terdiri dari program utama, decoder, dan *testbench*.

Adjustable crosswalk lamp akan menyediakan 4 tombol dimana guna dari masing-masing tombol tersebut dapat disesuaikan dengan kebutuhan pengguna. Tersedia mode untuk umum yang akan menyediakan waktu menyeberang selama 20 detik, mode untuk lansia yang akan menyediakan waktu menyeberang selama 60 detik, mode untuk tunanetra yang akan menyeberang selama 60 detik dan mode reset. Tentunya untuk kenyamanan pengguna tunanetra, tombol-tombol akan dilengkapi

dengan alfabet braille yang akan membantu proses penggunaan dari *Adjustable crosswalk lamp.*





The objectives of this project are as follows:

- Memenuhi ketentuan proyek akhir
- Menyediakan lampu penyeberangan yang memenuhi kebutuhan pengguna
- Mengatur lalu lintas agar lebih tertata rapi
- Membuat sebuah sistem digital yang bermanfaat dengan mengimplementasikannya pada VHDL.

1.3 ROLES AND RESPONSIBILITIES

The roles and responsibilities assigned to the group members are as follows:

Roles	Responsibilities	Person
Ketua	Manajemen proyek	Gemilang Bagas R
VHDL Programmer	Mengimplementasikan rangkaian dan testbench pada VHDL	Diva Hana Gemilang Bagas R M Irsyad Fakhruddin Rain Elgratio Sion H L G
Laporan	Menyusun laporan &PPT	Diva Hana M Irsyad Fakhruddin

Table 1. Roles and Responsibilities

CHAPTER 2

IMPLEMENTATION

2.1 EQUIPMENT

The tools that are going to be used in this project are as follows:

- Notepad ++
- Modelsim
- Google Slides
- Microsoft Word
- Github
- Draw.io

2.2 IMPLEMENTATION

Dalam kehidupan sehari-hari, tentunya mobilitas tidak dapat dihindari. Karena hal itu, manusia terus mengembangkan struktur dari perencanaan perkotaan termasuk sarana penyeberangan saat ini. Ada 2 alternatif yang saat ini digunakan oleh pejalan kaki pada umumnya untuk menyebrang , entah zebra cross maupun jembatan penyeberangan. Untuk *zebra cross* tersendiri, biasanya akan disediakan rambu di sekitar area agar pengguna jalan utama dapat berhati-hati ketika akan melewati zona tersebut, bahkan sekarang sudah banyak *zebra cross* dengan buzzer yang akan memberikan suara ketika akan ada yang menyebrang dan mengubah lampu lalu lintas jalan utama menjadi merah guna menambah keamanan bagi pengguna *zebra cross*. Adjustable Crosswalk Lamp kami dapat membuat pengalaman menyebrang para pejalan kaki makin nyaman karena lampu penyeberangan ini disertai mode-mode yang dapat disesuaikan dengan kebutuhan pejalan kaki.

Untuk mengimplementasikan proyek ini, kami menggunakan FSM yang dapat diatur oleh 4 mode yang sudah disediakan dengan sebuah default state yang menunjukan hijau untuk jalan utama. Apabila salah satu mode ditekan , maka ia akan pindah ke state tersebut dan bila tombol reset ditekan, maka timer akan langsung mengembalikan keadaan ke state semula.

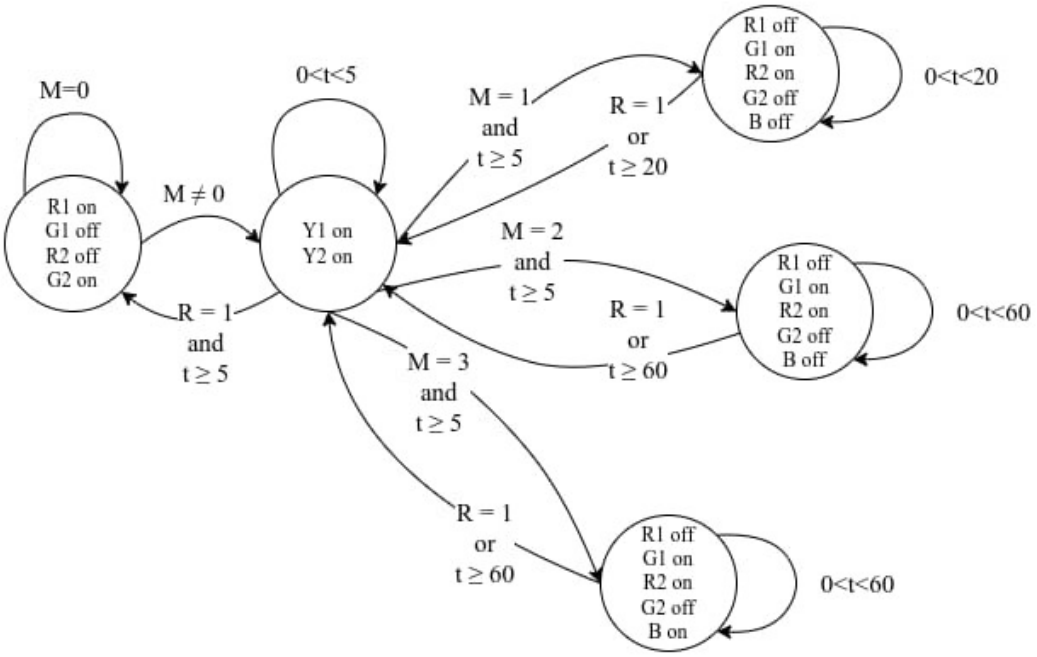


Fig 1. State Diagram

CHAPTER 3

TESTING AND ANALYSIS

3.1 TESTING

Setelah pembuatan program atau source code dari Crosswalk Lamp serta testbench selesai, selanjutnya adalah pengujian program. Pengujian ini bertujuan untuk mengetahui apakah program berjalan sesuai dengan yang diharapkan dan apakah output dari program sesuai dengan input yang diberikan. Pengujian dilakukan dengan menggunakan testbench. Pada pengujian ini program diberikan input berupa 4 buah tombol yang dipilih alat ini. Kemudian output yang akan didapatkan adalah sebuah timer, buzzer dan juga visual untuk menyebrang jalan. Source code program yang diuji adalah sebagai berikut :

```
1  library IEEE;
2  use IEEE.std_logic_1164.all;
3  use IEEE.numeric_std.all;
4
5  entity main_program is
6      port (
7      clock   : in std_logic;
8      tombol  : in std_logic_vector (1 downto 0); -- 00 untuk reset
9      mode    : out std_logic_vector (1 downto 0) -- 00 untuk tidak menyebrang (default)
10                                                 -- 01 untuk orang dewasa,
11                                                 -- 10 untuk lansia dan carrier,
12                                                 -- 11 untuk tunanetra
13     );
14 end main_program;
15
16 architecture behave of main_program is
17     --Component Low Lever Decoder BCD to 7-Segment
18     component Decoder is
19         port (
20             clk : in std_logic;
21             bcd : in std_logic_vector(3 downto 0);  --BCD input
22             segment7 : out std_logic_vector(6 downto 0)  -- 7 bit decoded output.
23         );
24     end component;
25
26     constant waktu_transisi : integer := 5; --limit waktu transisi sesaat ada yang menekan tombol = 4 detik
27     constant waktu_menyebrang_dewasa : integer := 20; --limit waktu saat menyebrang = 20 detik
28     constant waktu_menyebrang_lainnya : integer := 60; --limit waktu saat menyebrang = 20 detik
29
30     signal trigger_5 : std_logic := '0'; --trigger untuk memulai counter 5 detik
31     signal trigger_20 : std_logic := '0'; --trigger untuk memulai counter 20 detik
32     signal trigger_60 : std_logic := '0'; -- trigger untuk memulai counter 60
33     signal buzzer : std_logic := '0'; -- trigger untuk menyalakan buzzer
34     signal trigger_STM : std_logic := '0';
35
36     type state is (   --Define State kondisi jalan raya
37         state_default,
38         menyebrang_dewasa,
39         menyebrang_lansia,
40         menyebrang_tunanetra,
```

```
41          transisi
42          );
43      signal PS, NS: state;
44
45      type lampu is (merah, kuning, hijau);                   --Define warna yang akan menyala pada lampu
46      signal lampu_jalanRaya, lampu_penyebrang : lampu;
47
48      signal clock_count : integer := 0;                      --counter berdasarkan clock cycle
49      signal STM_counter1 : integer := 0;
50      signal inputPuluhan : std_logic_vector (3 downto 0);    --Input decoder untuk 7-segment puluhan
51      signal inputSatuan : std_logic_vector (3 downto 0);     --Input decoder untuk 7 segement satuan
52      signal outputPuluhan : std_logic_vector (6 downto 0);   --Output decoder untuk 7-segment puluhan
53      signal outputSatuan : std_logic_vector (6 downto 0);    --Output decoder untuk 7-segemnt satuan
54
55
56
57  begin
58      --define 2 decoder untuk 7-segment puluhan dan satuan
59      decode_puluhan : Decoder port map (clock, inputPuluhan, outputPuluhan);
60      decode_satuan : Decoder port map (clock, inputSatuan, outputSatuan);
61
62      --Synchronous process (Nerepresentasikan Flip FLop (terdapat Memory))
63      sync_proc : process (clock, NS) --Variable control ada clock, dan Next State (NS)
64      begin
65          if rising_edge(clock) then
66              PS <= NS;
67          end if;
68      end process sync_proc;
69
70      --Combinational process
71      comb_proc : process (STM_counter1,clock_count,tombol, PS) --Variable control ada clock-couter , tombol penyebrang (input 1 saat ada yang menekan tombol), Present State (PS)
72      begin
73          trigger_20 <= '0';
74          trigger_5 <= '0';
75
76          case PS is
77              when state_default =>               --Kondisi saat tidak ada pejalan kaki yang ingin menyebrang
78                  mode <= "00";
79                  lampu_jalanRaya <= hijau;
80                  lampu_penyebrang <= merah;
81                  trigger_STM <= '1';
82                  if (tombol = "00") then             --Jika tidak ada pejalan kaki yang menekan tombol
```

```
83                      NS <= state_default;
84                  else        --Saat ada pejalan kaki yang menekan tombol ingin menyebrang jalan
85                      NS <= transisi;         --Next State tetap ke kondisi state_default
86                  end if;
87
88              when transisi =>                        --Kondisi sesaat setelah ada pejalan kaki yang menekan tombol
89                  lampu_jalanRaya <= kuning;
90                  lampu_penyebrang <= kuning;
91                  trigger_5 <= '1';                                   --Trigger counter 5 detik aktif
92                  if((clock_count = waktu_transisi) and (tombol = "01")) then NS <= menyebrang_dewasa;
93                  elsif((clock_count = waktu_transisi) and (tombol = "10")) then NS <= menyebrang_lansia
94                  elsif((clock_count = waktu_transisi) and (tombol = "11")) then NS <= menyebrang_tunanetra
95                  elsif((clock_count = waktu_transisi) and (tombol = "00")) then NS <= state_default   --Setelah counter bernilai sama dengan waktu transisi yaitu 5 maka NS menjadi ko
96                  end if;
97
98              when menyebrang_dewasa =>                  --Kondisi saat pejalan kaki boleh menyebrang dan kendaraan wajib berhenti
99                  mode <= "01";
100                 lampu_jalanRaya <= merah;
101                 lampu_penyebrang <= hijau;
102                 trigger_20 <= '1';                                          --Trigger counter 20 detik aktif
103                 if ((clock_count = waktu_menyebrang_dewasa) or (tombol = "00")) then
104                     NS <= transisi;
105                 end if;
106
107             when menyebrang_lansia =>                  --Kondisi saat pejalan kaki boleh menyebrang dan kendaraan wajib berhenti
108                 mode <= "10";
109                 lampu_jalanRaya <= merah;
110                 lampu_penyebrang <= hijau;
111                 trigger_60 <= '1';                                          --Trigger counter 60 detik aktif
112                 if ((clock_count = waktu_menyebrang_lainnya) or (tombol = "00")) then
113                     NS <= transisi;
114                 end if;
115
116                 when menyebrang_tunanetra =>                --Kondisi saat pejalan kaki boleh menyebrang dan kendaraan wajib berhenti
117                 mode <= "11";
118                 lampu_jalanRaya <= merah;
119                 lampu_penyebrang <= hijau;
120                 buzzer <= '1';
121                 trigger_60 <= '1';                                          --Trigger counter 60 detik aktif
122                 if ((clock_count = waktu_menyebrang_lainnya) or (tombol = "00")) then
```

```
                    NS <= transisi;
                end if;

            end case;
            end process comb_proc;

        --Prosess counter waktu
        timer : process (trigger_STN, trigger_20, trigger_5, clock)
        begin
            if trigger_5 = '1' then
                STN_counter1 <= 1;                                          --Jika trigger counter 5 detik menyala, maka counter akan menghitung 5 kali clock cycle
                if rising_edge(clock) then                                  --Saat clock kondisi naik
                                    clock_count <= clock_count + 1;                             --Counter akan bertamabh 1
                                    if (clock_count = waktu_transisi) then                        --Saat counter bernilai sama dengan waktu transisi atau 5 detik
                        clock_count <= 0;                                   -- clock_count akan direset
                                    end if;
                            end if;

            elsif trigger_20 = '1' then                                     --Jika trigger counter 20 detik menyala, maka counter akan menghitung 20 kali clock cycle
                if rising_edge(clock) then                                  -- saat clock kondisi naik
                                    clock_count <= clock_count + 1;                             -- counter akan bertambah 1
                                    if (clock_count = waktu_menyebrang_dewasa) then                     --Saat counter bernilai sama dengan waktu menyebrang atau 20
                        clock_count <= 0;                                   --Counter akan direset
                                    end if;
                            end if;

            elsif trigger_60 = '1' then                                     --Jika trigger counter 60 detik menyala, maka counter akan menghitung 60 kali clock cycle
                if rising_edge(clock) then                                  -- saat clock kondisi naik
                                    clock_count <= clock_count + 1;                             -- counter akan bertambah 1
                                    if (clock_count = waktu_menyebrang_lainnya) then                        --Saat counter bernilai sama dengan waktu menyebrang atau 60
                        clock_count <= 0;                                   --Counter akan direset
                                    end if;
                            end if;

            elsif trigger_STN = '1' then
                if rising_edge(clock) then
                    STN_counter1 <= STN_counter1 + 1;
                end if;
            end if;
```

```
            end if;
            end process timer;

        --proses input control untuk decoder puluhan dan satuan berdasarkan counter clock cycle
        decControl : process (clock_count, trigger_5, trigger_20, trigger_60)
        begin
            --Saat kondisi transisi atau sesaat setelah pejalan kaki menekan tombol
            --7 segment akan menampilkan hitung mundur 5 detik
            --Karena counter merupakan count up maka input decoder dibuat terbalik dengan counter
            if(trigger_5 = '1') then
                inputPuluhan <= "1111";
                case clock_count is
                    when 1 => inputSatuan <= "0101"; --5
                    when 2 => inputSatuan <= "0100"; --4
                    when 3 => inputSatuan <= "0011"; --3
                    when 4 => inputSatuan <= "0010"; --2
                    when 5 => inputSatuan <= "0001"; --1
                    when others => inputSatuan <= "1111"; --Mati
                end case;

                --Saat kondisi pejalan kaki menyebrang dan kendaraan wajib berhenti
                --7 segment akan menampilkan hitung mundur 20 detik
                --Karena counter merupakan count up, maka input decoder dibuat terbalik dengan counter
                elsif(trigger_20 = '1') then
                    case clock_count is
                        when 1 =>
                            inputPuluhan <= "0010"; --20
                            inputSatuan <= "0000";
                        when 2 =>
                            inputPuluhan <= "0001"; --19
                            inputSatuan <= "1001";
                        when 3 =>
                            inputPuluhan <= "0001"; --18
                            inputSatuan <= "1000";
                        when 4 =>
                            inputPuluhan <= "0001"; --17
                            inputSatuan <= "0111";
                        when 5 =>
                            inputPuluhan <= "0001"; --16
                            inputSatuan <= "0110";
                        when 6 =>
```

```
                        inputPuluhan <= "0001"; --15
                        inputSatuan <= "0101";
                    when 7 =>
                        inputPuluhan <= "0001"; --14
                        inputSatuan <= "0100";
                    when 8 =>
                        inputPuluhan <= "0001"; --13
                        inputSatuan <= "0011";
                    when 9 =>
                        inputPuluhan <= "0001"; --12
                        inputSatuan <= "0010";
                    when 10 =>
                        inputPuluhan <= "0001"; --11
                        inputSatuan <= "0001";
                    when 11 =>
                        inputPuluhan <= "0001"; --10
                        inputSatuan <= "0000";
                    when 12 =>
                        inputPuluhan <= "0000"; --9
                        inputSatuan <= "1001";
                    when 13 =>
                        inputPuluhan <= "0000"; --8
                        inputSatuan <= "1000";
                    when 14 =>
                        inputPuluhan <= "0000"; --7
                        inputSatuan <= "0111";
                    when 15 =>
                        inputPuluhan <= "0000"; --6
                        inputSatuan <= "0110";
                    when 16 =>
                        inputPuluhan <= "0000"; --5
                        inputSatuan <= "0101";
                    when 17 =>
                        inputPuluhan <= "0000"; --4
                        inputSatuan <= "0100";
                    when 18 =>
                        inputPuluhan <= "0000"; --3
                        inputSatuan <= "0011";
```

```
                    when 19 =>
                        inputPuluhan <= "0000"; --2
                        inputSatuan <= "0010";
                    when 20 =>
                        inputPuluhan <= "0000"; --1
                        inputSatuan <= "0001";
                    when others =>
                        inputPuluhan <= "1111"; --Mati
                        inputSatuan <= "1111"; --Mati
                end case;
            end if;
        end process ;


end architecture ;
```

Sourch code dari dari testbench yang diuji adalah sebagai berikut :

```vhdl
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity TestBench is
end TestBench;

architecture rtl of TestBench is
    component main_program is
        port
        (
            clock : in std_logic;
            tombol : in std_logic_vector (1 downto 0);
            mode : out std_logic_vector (1 downto 0)
        );
    end component;
    signal clock : std_logic;
    constant clock_period : time := 0.1 ns;
    signal clock_counter : integer := 0;
    signal tombol : std_logic_vector := '00';

    -- 00 untuk tidak menyebrang (default)
    -- 01 untuk orang dewasa,
    -- 10 untuk lansia dan carrier,
    -- 11 untuk tunanetra

begin
    test : main_program port map (clock, tombol, mode);

    uut : process
        begin
            clock <= '0';
            wait for clock_period;
            clock <= '1';
            clock_counter <= clock_counter + 1;
            wait for clock_period;

            -- membiarkan mode idle selama 15 detik
            if(clock_counter = 15) then
                tombol <= '00'
```

```vhdl
                severity note;

            elsif(clock_counter = 20) then
                tombol <= '10' -- mencoba mode 2

            elsif(clock_counter = 35) then
                tombol <= '11' -- melakukan interupsi dengan mencoba mode 11
            assert mode = '10'
                report "Tidak dapat melakukan mode tunanetra, silahkan reset terlebih dahulu"
                severity warning;

            elsif(clock_counter = 42) then
                tombol <= '01'

            elsif(clock_counter = 40) then
                tombol <= '00' -- melakukan reset sebelum selesai

            elsif(clock_counter = 45) then
                tombol <= '11' -- mode tunanetra

            elsif(clock_counter = 85) then
                tombol <= '00' -- melakukan reset sebelum selesai

            elsif(clock_counter = '90') then
                tombol <= '01'

            elsif(clock_counter = 120) then
                tombol <= '11'

            elsif(counter_clock = 190) then
                wait;
            end if;
        end process;

end architecture;
```

3.2 RESULT

Berdasarkan pengujian yang dilakukan didapatkan dua hasil yaitu hasil pengujian secara langsung dan hasil pengujian menggunakan testbench. Hasil yang didapatkan menggunakan metode secara langsung adalah sebagai berikut :

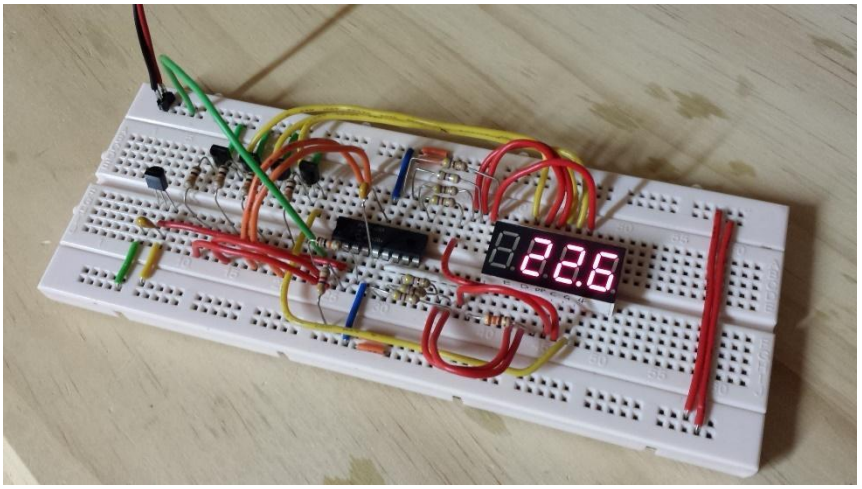


Fig 2. Testing Result

Donec at iaculis leo. Integer congue sed lacus suscipit iaculis. Nulla a augue ut sapien rutrum consectetur. Sed ac dignissim lorem. Maecenas hendrerit nisl a metus posuere, vel vehicula metus eleifend. Mauris blandit, dolor nec malesuada tempor, purus nibh aliquet nibh, at faucibus leo felis a nisi. Donec pharetra leo risus, in vestibulum dui laoreet in. Nulla facilisi. Etiam nec consequat justo. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam erat volutpat. Etiam pharetra eleifend hendrerit.

3.3 ANALYSIS

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi ornare accumsan nisl sit amet sodales. Suspendisse sed dictum velit, in suscipit sem. Vestibulum egestas neque vel velit tristique, id venenatis nunc fringilla. Mauris condimentum diam consequat egestas tincidunt. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Vivamus semper pharetra commodo. Integer hendrerit ultricies lacus. Nullam id magna sed risus placerat luctus sed at mauris. Curabitur ligula urna, pharetra eget mi sit amet, sagittis feugiat magna. Curabitur ex nisl, eleifend et mattis sit amet, condimentum non nisi.

Donec at iaculis leo. Integer congue sed lacus suscipit iaculis. Nulla a augue ut sapien rutrum consectetur. Sed ac dignissim lorem. Maecenas hendrerit nisl a metus posuere, vel vehicula metus eleifend. Mauris blandit, dolor nec malesuada tempor, purus nibh aliquet nibh, at faucibus leo felis a nisi. Donec pharetra leo risus, in vestibulum dui laoreet in. Nulla facilisi. Etiam nec consequat justo. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam erat volutpat. Etiam pharetra eleifend hendrerit.

CHAPTER 4

CONCLUSION

Setelah mengerjakan serangkaian proses pembuatan proyek akhir ini, kami berhasil membuat suatu program Crosswalk Lamp yang menggunakan Finite State Machine (FSM) sebagai penggambaran sekuensial dari cara Crosswalk Lamp tersebut bekerja. Present state dan juga next state dari FSM berfungsi dengan baik untuk menentukan mode yang akan digunakan untuk menyebrang mulai dari mode umum, lansia, hingga tunanetra, dengan menekan salah satu dari mode tersebut maka lampu akan berubah menjadi hijau pada pejalan kaki dan menjadi merah pada para pengendara, dan juga terdapat timer yang berbeda di setiap modenya setelah sampai pada ujung jalan maka penyebrang dapat menekan tombol reset agar kembali seperti semula yaitu hijau di pengendara dan merah pada penyebrang. Pada program test bench perlu diperhatikan agar FSM dapat melewati tiap state yang dibutuhkan untuk mengeluarkan lampu dan waktu yang tepat yang diinginkan

REFERENCES

[1] Reference 1

[2] Reference 2

[3] Reference 3

[4] Reference 4

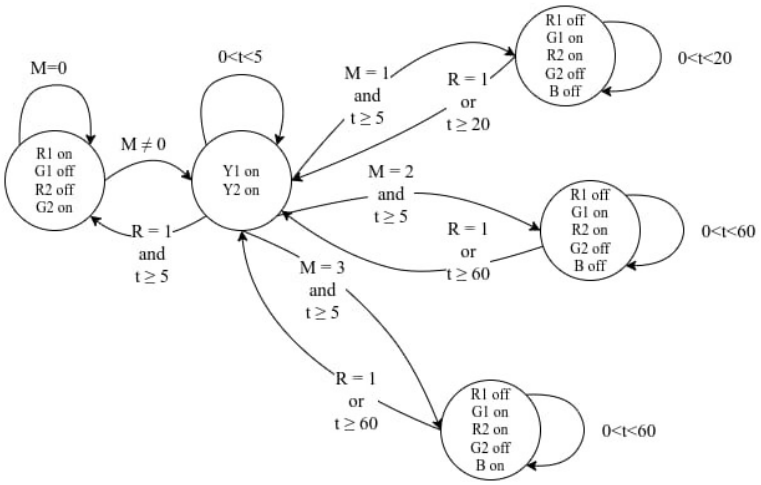
[5] Reference 5

[6] Reference 6

[7] And so on

APPENDICES

Appendix A: Project Schematic



Appendix B: Documentation