



**FUNDAMENTAL OF DIGITAL SYSTEM FINAL PROJECT REPORT
DEPARTMENT OF ELECTRICAL ENGINEERING
UNIVERSITAS INDONESIA**

Soda Dispensing Machine

GROUP A8

Adrien Ardra Ramadhan	2106731485
Andikha Wisanggeni	2106731503
Jeremy Ganda Pandapotan	2106731573
Muhammad Farrel Mirawan	2106731554

PREFACE

Puji dan syukur kita panjatkan kehadiran Tuhan YME yang telah rahmatnya sehingga kami sebagai kelompok A-8 dapat menyelesaikan tugas akhir Praktikum Perancangan Sistem Digital yang berjudul “Soda Dispensing Machine”.

Pada zaman ini, perkembangan teknologi sudah semakin maju dan tidak bisa terlepas dari kehidupan sehari-hari, mulai dari kita bangun di pagi hari hingga kita tidur untuk mengakhiri hari di malam hari. Dengan proyek ini, kami berharap dapat belajar sekaligus menerapkan perkembangan teknologi, dan nanti di masa depan dapat dikembangkan menjadi hal yang lebih bermanfaat.

Kami kelompok A-8 mengucapkan terima kasih kepada para anggota kelompok yang telah saling bekerja sama dalam membuat program, simulasi, hingga menyusun laporan, sehingga proyek dapat selesai dengan baik. Kami juga berterima kasih kepada asisten lab yang telah membimbing kami.

Depok, December 10, 2022

Group A-8

TABLE OF CONTENTS

CHAPTER 1	4
INTRODUCTION	4
1.1 BACKGROUND	4
1.3 OBJECTIVES	5
1.4 ROLES AND RESPONSIBILITIES	6
CHAPTER 2	7
IMPLEMENTATION	7
2.1 EQUIPMENT	7
2.2 IMPLEMENTATION	7
CHAPTER 3	8
TESTING AND ANALYSIS	8
3.1 TESTING	8
3.2 RESULT	12
3.3 ANALYSIS	14
CHAPTER 4	16
CONCLUSION	16
REFERENCES	17
APPENDICES	18
Appendix A: Project Schematic	18
Appendix B: Documentation	18

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

FPGA(Field Programmable Gate Arrays) merupakan sebuah IC digital yang sering digunakan untuk mengimplementasikan rangkaian digital. FPGA berisikan configurable logic blocks (CLBs) dan sekumpulan interkoneksi yang dapat diprogram yang memungkinkan designer untuk mengkonfigurasi blok dan menghubungkannya sehingga dapat memiliki fungsi yang kompleks.

FPGA dapat dikatakan memiliki banyak kegunaan. FPGA memungkinkan developer untuk melakukan tes terhadap sejumlah variable ketika sebuah rangkaian telah dibuat. selain itu, FPGA memiliki banyak kelebihan dibandingkan dengan programmable logic device (PLD). kelebihan tersebut yaitu FPGA dapat berisikan gerbang logika yang dapat mencapai ribuan. Pemrograman FPGA dapat dilakukan dengan mengimplementasikan Very High Speed Integrated Circuit Hardware Description Language (VHDL).

Dalam dunia industri makanan dan minuman, tentunya memerlukan berbagai macam inovasi untuk dapat berjalan sesuai dengan majunya teknologi. Maka dari itu, kami dari kelompok A8 melakukan inovasi dalam industri ini, yaitu kami membuat Soda Dispensing Machine.



Menurut transparencymarketresearch, penjualan dari dispensing machine akan terus meningkat hingga tahun 2027 dengan segmentasi market berada pada tempat-tempat umum seperti hotel, sekolah, bandara, stasiun kereta, kantor, dan masih banyak lagi. Dengan melihat

pasar yang besar itu, maka diharapkan Teknologi yang kami rancang dapat berkontribusi dalam peningkatan penjualan soda dispensing machine. Dimana Soda Dispensing Machine ini akan melakukan perhitungan harga setelah user memilih soda yang diinginkan lalu menampilkannya dalam seven segment display. Selain itu juga memungkinkan pengguna untuk memiliki akun sehingga dapat mempunyai balance pada akun tersebut.

1.2 PROJECT DESCRIPTION

FPGA yang kami buat adalah Soda Dispensing Machine yang akan menerima pesanan dari pelanggan dan memberitahu pelanggan berapa banyak uang yang harus dibayar. Mesin kami memiliki beberapa jenis minuman dengan ukuran gelas yang bisa dipilih oleh pelanggan. Harga dari Minuman ini tentunya akan berbeda untuk jenis soda dan ukuran yang dipilih oleh pelanggan. Jenis dari sodanya pun ada 2, yaitu pepsi dan coke. Harga dari masing-masing soda untuk pepsi yaitu seharga 5 ribu rupiah dan coke seharga 10 ribu rupiah. Ukuran dari soda nya juga akan menentukan harga akhir, jika memilih medium maka harga soda akan ditambah 5 ribu rupiah dan big akan ditambah 10 ribu rupiah.

Ketika user telah memilih soda dan ukuran yang akan dibeli, Selanjutnya user akan ditanya apakah memiliki akun atau tidak. Jika user tidak memiliki akun, maka akan masuk ke mode guest dimana user harus memasukan uang yang akan dideteksi oleh sensor. Nantinya mesin akan menampilkan harga dari soda dan menghitung uang yang masuk sekaligus memberikan uang kembalian jika uang yang dimasukkan oleh user berlebih. Namun jika user memiliki akun, maka user tidak perlu memasukan uang namun balance dari akun tersebut akan berkurang dengan sendirinya.

1.3 OBJECTIVES

The objectives of this project are as follows:

1. Mengimplementasikan bahasa VHDL ke barang yang biasa kita temukan sehari-hari.
2. Dapat membuat desain Finite State Machine menggunakan bahasa VHDL.
3. Bekerja sama dengan baik dalam membuat mesin dengan bahasa pemrograman VHDL.
4. Mengasah ilmu VHDL untuk bekal masa depan nanti.

1.4 ROLES AND RESPONSIBILITIES

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

Roles	Responsibilities	Person
Penentu Ide	Brainstorming dan menentukan ide yang ingin dibuat	Adrien Ardra Ramadhan, Andikha Wisanggeni, Jeremy Ganda Pandapotan, Muhammad Farrel Mirawan
Pembuat component	Membuat code untuk component-component yang nanti akan digunakan	Adrien Ardra Ramadhan, Andikha Wisanggeni, Jeremy Ganda Pandapotan, Muhammad Farrel Mirawan
Pembuat testbench	Membuat code testbench	Adrien Ardra Ramadhan, Andikha Wisanggeni, Jeremy Ganda Pandapotan, Muhammad Farrel Mirawan
Pembuat Laporan	Mengedit dan menyelesaikan laporan	Adrien Ardra Ramadhan, Andikha Wisanggeni, Jeremy Ganda Pandapotan, Muhammad Farrel Mirawan

Table 1. Roles and Responsibilities

CHAPTER 2

IMPLEMENTATION

2.1 EQUIPMENT

Tools yang digunakan pada project ini adalah :

- VSCode
- ModelSim
- Draw.io
- Github
- Quartus Prime

2.2 IMPLEMENTATION

Soda Dispensing Machine memiliki state awal *idle* dimana pada state ini akan diminta input untuk memilih antara 2 jenis soda yaitu pepsi atau coke. jika salah satunya dipilih oleh user maka yang terjadi selanjutnya adalah mesin akan berubah state menjadi state jenis soda yang dipilih sebelumnya. harga yang harus dibayar oleh user juga sudah diset oleh mesin dimana harga ini akan berbeda untuk jenis minuman yang dipilih. Pada state ini user akan diminta lagi untuk memasukan input untuk memilih ukuran gelas, terdapat tiga ukuran gelas yaitu small, medium, dan big. ukuran minuman ini akan menentukan total uang yang harus user bayar.

Setelah ukuran soda dipilih maka mesin akan menampilkan harga yang harus dibayarkan oleh user pada seven segment. User memiliki dua pilihan pada state ini yaitu untuk membayar dalam mode member atau dalam tunai. jika user memilih member maka pembayaran akan dilakukan secara cashless dan state mesin berubah menjadi dispense yang artinya minuman langsung disajikan. Namun jika user memilih tunai maka mesin akan menunggu user memasukan uang. setiap uang dimasukan mesin akan mengecek nominal dari uang tersebut dan mengurangi angka yang berada pada seven segment sesuai dengan uang yang dimasukan. Ketika total uang sudah terbayarkan maka state mesin berubah menjadi state dispense dan minuman akan disajikan. ketika minuman sudah disajikan mesin akan berubah state lagi menjadi statenya yang awal yaitu Idle

CHAPTER 3

TESTING AND ANALYSIS

3.1 TESTING

Program untuk Soda-Dispensing-Machine ini terdiri dari FSM state yang dijadikan sebagai main dan komponen seven segment yang berguna untuk menampilkan informasi mengenai harga dari produk yang ingin dibeli dalam bentuk seven segment. Soda-Dispensing-Machine ini juga memiliki total 8 input dan 5 output yang dapat dilihat lewat potongan kode bagian entity dibawah ini :

```
entity FSMDispenser is
  port (
    --Input
    clk      : in std_logic;
    reset    : in std_logic;
    drink    : in std_logic;
    account  : in std_logic;
    done     : in std_logic;
    money    : in std_logic_vector(3 downto 0);
    size     : in std_logic_vector(1 downto 0);
    sensor   : in std_logic;
    --Output
    sodaOut  : out std_logic;
    change   : out std_logic_vector(4 downto 0);
    kembali  : out std_logic;
    display1 : out std_logic_vector(6 downto 0);
    display2 : out std_logic_vector(6 downto 0)
  );
end entity;
```

Diatas ini merupakan port yang digunakan dengan port input yang terdiri dari clk dan reset yang digunakan sebagai control dari program ini dengan clk sebagai waktu yang bersifat rising edge dan reset sebagai yang membuat state dari program kembali seperti semula, untuk input-input lain adalah input yang akan dipilih oleh user. Dan sensor pada input akan menandakan apakah ada uang yang masuk atau tidak, jumlah nominal uang yang bisa dimasukkan adalah Rp.5000 dan Rp.10.000 dan maksimal uang yang masuk adalah Rp.20.000 Untuk output sendiri terdiri dari 5 yaitu sodaOut menandakan bahwa soda sudah keluar, change menandakan nominal dari kembalian yang diberikan apabila ada, kembali menandakan apakah ada kembalian atau tidak, dan display1 dan display2 menandakan seven segment, dengan display1 sebagai puluhan dan display2 sebagai satuan. Kemudian untuk state pada program ini ada 15 state yang dapat dilihat dari potongan kode berikut :


```

type state_type is (idle, pepsi, coke, small, medium, big, guest,
member, money_0, money_5,
money_10, money_15, money_20, check, dispense);

```

Gambar diatas merupakan state yang digunakan dan untuk program state tersebut juga sudah kami buat dan kami coba di Quartus untuk menghasilkan state machine viewer dan hasil dari state machine viewer akan dicantumkan pada bagian result.

Untuk melakukan pengetesan dari code-code tersebut dapat digunakan metode yang namanya test bench. Test bench ini sendiri merupakan barisan kode yang berisi komponen dari file vhd yang ingin dites, yang nantinya lewat test bench ini kita dapat memasukkan input untuk melakukan pengetesan program dan bisa memudahkan dalam melakukan simulasi di modelsim, untuk program ini kami membuat 2 kode test bench yaitu test bench untuk FSMDispenser dan test bench untuk seven segmen. Kode test bench yang telah kami buat adalah sebagai berikut :

```

library ieee;
use ieee.std_logic_1164.all;

entity FSMDispenser_tb is
end entity;

architecture tb of FSMDispenser_tb is

    component FSMDispenser is
        port (
            clk      : in std_logic;
            reset    : in std_logic;
            drink    : in std_logic;
            account  : in std_logic;
            done     : in std_logic;
            money    : in std_logic_vector(3 downto 0);
            size     : in std_logic_vector(1 downto 0);
            sensor   : in std_logic;
            sodaOut  : out std_logic;
            change   : out std_logic_vector(4 downto 0);
            kembali  : out std_logic;
            display1 : out std_logic_vector(6 downto 0);
            display2 : out std_logic_vector(6 downto 0)
        );
    end component;

    signal clk : std_logic := '0';
    signal reset : std_logic := '0';
    signal drink : std_logic := '0';
    signal account : std_logic := '0';

```

```

signal done : std_logic := '0';
signal money : std_logic_vector(3 downto 0) := (others => '0');
signal size : std_logic_vector(1 downto 0) := (others => '0');
signal sensor : std_logic := '0';
signal sodaOut : std_logic;
signal change : std_logic_vector(4 downto 0);
signal kembali : std_logic;
signal display1 : std_logic_vector(6 downto 0);
signal display2 : std_logic_vector(6 downto 0);
begin
    DUT: FSMDispenser
        port map (
            clk => clk,
            reset => reset,
            drink => drink,
            account => account,
            done => done,
            money => money,
            size => size,
            sensor => sensor,
            sodaOut => sodaOut,
            change => change,
            kembali => kembali,
            display1 => display1,
            display2 => display2
        );
    clk_process: process
    begin
        clk <= '0';
        wait for 10 ns;
        clk <= '1';
        wait for 10 ns;
    end process;
    test_vector_1: process is
    begin
        reset <= '0';
        wait for 20 ns;
        drink <= '0';
        wait for 20 ns;
        size <= "00";
        wait for 20 ns;
        account <= '0';
        wait for 20 ns;
        sensor <= '1';
        wait for 20 ns;
        money <= "1010";
        wait for 20 ns;
        done <= '1';
        wait for 100 ns;
        assert sodaOut = '1'

```

```

        report "Test vector 1 failed: sodaOut"
        severity failure;
    assert change = "00101"
        report "Test vector 1 failed: change"
        severity failure;
    assert kembali = '1'
        report "Test vector 1 failed: kembali "
        severity failure;
    reset <= '1';
end process;
end architecture;

```

Diatas ini merupakan kode test bench untuk FSMDDispenser, ciri khas dari test bench ini adalah tidak memiliki entity atau entity nya kosong dan komponen akan berisi entity file vhd yang ingin dites pada kasus ini adalah entity dari FSMDDispenser. Dari test bench diatas dapat melihat input dan output dari program juga dapat melihat curren_state dan next_state dari program. Selain test bench dari FSMDDispenser kami juga membuat test bench untuk sevensegment, kode nya adalah sebagai berikut :

```

library ieee;
use ieee.std_logic_1164.all;

entity sevseg_tb is
end entity;

architecture testbench of sevseg_tb is
    -- Instantiate the sevseg component
    component sevseg is
        port (
            sev_in    : in std_logic_vector(3 downto 0);
            sev_out   : out std_logic_vector(6 downto 0)
        );
    end component;

    -- Clock period is 10 ns
    constant clk_period: time := 10 ns;
    signal sev_in    : std_logic_vector(3 downto 0);
    signal sev_out   : std_logic_vector(6 downto 0);

begin
    -- Instantiate the sevseg component
    DUT: sevseg port map (
        sev_in => sev_in,
        sev_out => sev_out
    );

    -- Generate a clock signal with a period of 10 ns
    clk_gen: process

```

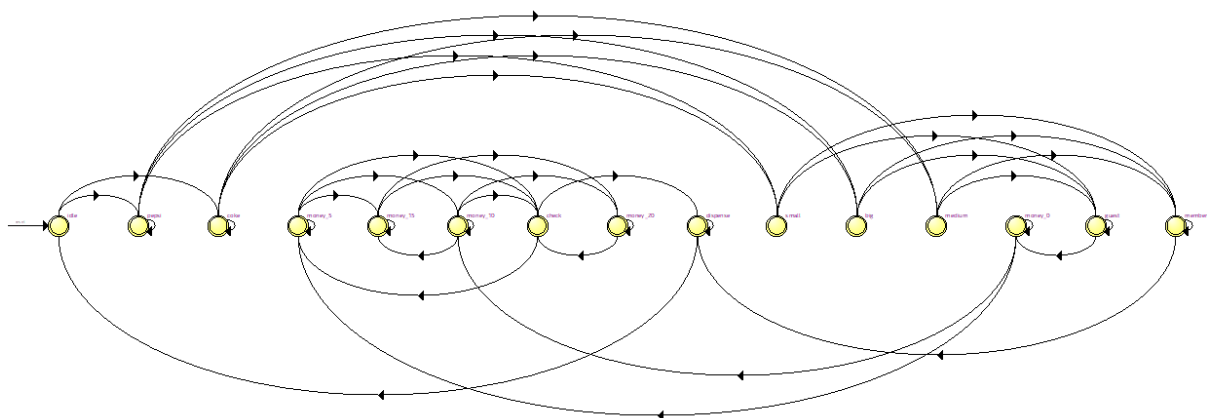
```

begin
    sev_in <= "0000";
    wait for clk_period/2;
    sev_in <= "0001";
    wait for clk_period/2;
    sev_in <= "0010";
    wait for clk_period/2;
    sev_in <= "0011";
    wait for clk_period/2;
    sev_in <= "0100";
    wait for clk_period/2;
    sev_in <= "0101";
    wait for clk_period/2;
    sev_in <= "0110";
    wait for clk_period/2;
    sev_in <= "0111";
    wait for clk_period/2;
    sev_in <= "1000";
    wait for clk_period/2;
    sev_in <= "1001";
    wait for clk_period/2;
    sev_in <= "0000";
    wait for clk_period/2;
    sev_in <= "0001";
    wait for clk_period/2;
end process;
end architecture;

```

Diatas merupakan kode testbench dari sevensegmen, kode diatas akan melakukan pengetesan untuk masing-masing input dan akan dapat dilihat apakah output yang akan dikeluarkan akan sesuai atau tidak dengan output yang sudah dideklarasikan. Untuk hasil pengetesan kedua test bench lewat modelsim akan ditunjukkan pada bagian berikutnya yaitu result.

3.2 RESULT



idle. Pada kondisi Idle ini lah user akan memasukkan input berupa pilihan minuman antara pepsi maupun coke. Semua signal maupun port akan di set dari 0. Kemudian jika dimasukkan input bernilai 1 maka akan pindah ke state coke, Jika input bernilai 0 maka akan pindah ke state pepsi, dan jika tidak keduanya maka akan tidak berpindah state.

Kemudian ketika pada FSM present state nya pepsi, maka otomatis seven segment akan menampilkan nilai 5 ribu. next state nya akan bergantung dari input pada size, mulai dari small hingga big yang tentunya akan terdiri dari harga yang berbeda-beda juga. Sama hal nya ketika present state nya coke, awalnya secara otomatis seven segment akan menampilkan nilai 10 ribu, kemudian next state nya akan berupa size yang akan bergantung pada inputnya mulai dari small hingga big. Jika input size bernilai 00 maka size nya akan small, jika input bernilai 01 maka size akan medium, jika input 10 maka size akan big.

Pada tahap size, seven segment akan menampilkan harga namun akan tergantung dengan pilihan minuman yang dipilih oleh pengguna. Semakin besar size maka seven segment akan menampilkan nilai harga yang semakin besar juga. Pada size small, sevseg akan menampilkan nilai 5 ribu jika drink =0 (pepsi) dan menampilkan nilai 10 ribu jika drink=1 (coke). Pada state medium, sevseg akan menampilkan nilai 10 ribu jika user memilih drink =0 (pepsi). Kemudian sevseg akan menampilkan nilai 15 ribu jika user memilih drink =1 (coke). Pada state big, sevseg akan menampilkan nilai 15 ribu jika user memilih drink =0(pepsi) dan sevseg akan menampilkan nilai 20 ribu jika user memilih drink=1(coke). Kemudian pada masing-masing size akan melanjutkan next state dengan state member jika input =1 dan guest jika input =0.

Jika pada state member, maka sevseg akan menampilkan harga dari masing-masing minuman sesuai dengan jenis minuman dan ukurannya lalu pindah ke state dispense dimana mesin akan mengeluarkan minuman. Kemudian jika terdapat pada state guest, maka sevseg akan menampilkan harga dari minumannya lalu akan mengaktifkan sensor yang mendeteksi ada atau tidaknya uang. Kemudian akan lanjut kepada state money_0

Pada state money_0, maka akan menerima input money berupa 5 maupun 10 ribu sehingga nilai dari accumulate akan terus bertambah sesuai dengan jumlah uang yang masuk. Mesin akan terus menerima input money hingga nilai dari done akan bernilai 1. Ketika done =1 maka state akan berpindah menjadi check. Dimana pada state check akan menghitung nilai accumulate dan mengcompare dengan price, operasi ini bertujuan untuk menentukan ada atau tidaknya kembalian dari uang yang dimasukkan oleh user. Selain itu, pada state ini akan

ke state dispense. Pada state dispense, maka terdapat kondisi jika account =1 maka soda akan keluar dan tidak perlu mengeluarkan uang kembalian. kemudian jika tidak memiliki akun, maka akan mengeluarkan uang kembalian jika terdapat kembalian.

CHAPTER 4

CONCLUSION

Soda Dispensing Machine merupakan mesin atau sistem yang akan memberikan jenis soda yang kita inginkan dengan ukuran yang kita inginkan juga. Soda Dispensing Machine akan menerima perintah dan uang yang harus dibayarkan dan uang tersebut akan dicek untuk menentukan jenisnya.

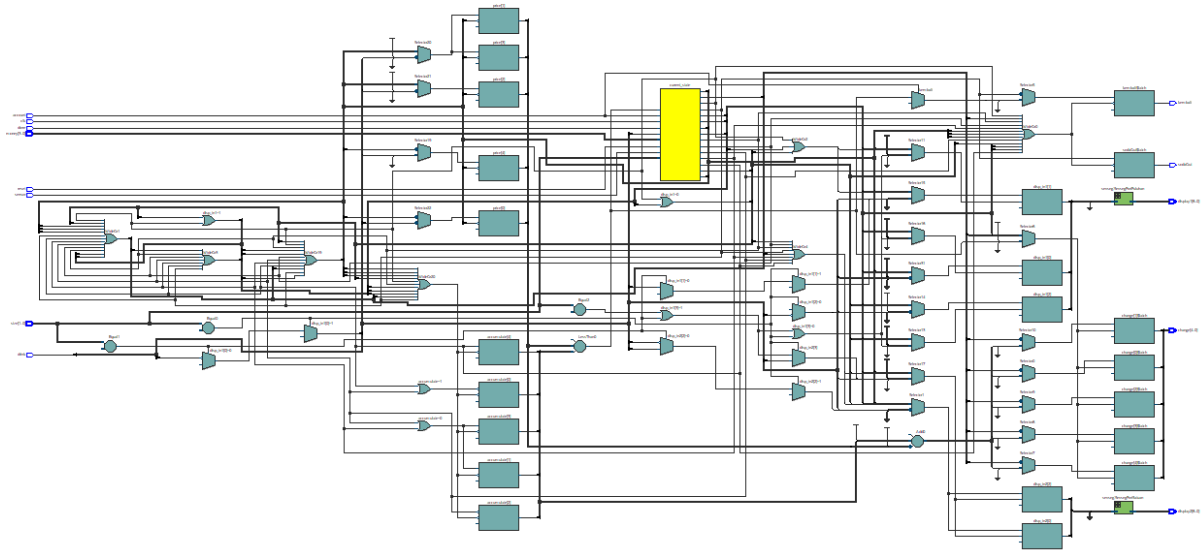
Cara kerja dari sistem ini adalah sistem akan dimulai di mode idle dan diakhiri di mode dispense. Pada mode idle user akan memilih jenis minuman, yaitu pepsi (1) dan coke (2), juga akan memasukkan ukuran minumannya, yaitu Small (00), Medium (01), dan Big (10). Setelah itu, user akan memilih apakah user memiliki member atau tidak. Jika iya, maka user harus memilih 1 (Member), dengan begitu user tidak harus memasukkan uang. Namun jika tidak, maka user harus memilih 0 (Guest), kemudian user akan memasukkan uang, dan mesin akan memprosesnya menjadi state yang sesuai dengan uang yang diinput. Kemudian sistem akan mengecek uang yang telah diinput. Jika uang yang diinput lebih besar dari harga minuman, maka uang tersebut akan dikurangi dengan harga soda dan akan mengeluarkan kembaliannya. Namun jika uang yang diinput lebih kecil dari harga minuman, maka akan diminta untuk menambahkan uang. Setelah selesai cek uangnya dan hasilnya cukup, maka sistem akan masuk ke state dispense dan akan mengeluarkan minumannya.

REFERENCES

- [1] *Soft drink dispensing machines market*. Transparency Market Research. [Online] Available at: <https://www.transparencymarketresearch.com/soft-drink-dispensing-machines-market.html> (Accessed: December 8, 2022).
- [2] Ltd., A. *What is FPGA?*, Arm. Available at: <https://www.arm.com/glossary/fpga> (Accessed: December 8, 2022).
- [3] Modul 2 hingga 9 Praktikum Perancangan Sistem Digital. [Online]. Available at: <https://emas.ui.ac.id/course/view.php?id=23115> (Accessed: December 8, 2022)
- [4] *Finite State Machine*. [Online]. Available at : https://isaaccomputerscience.org/concepts/dsa_toc_fsm?examBoard=all&stage=all. (Accessed : December 3, 2022)
- [5] CPLD Development Software for PC Features. [Online]. Available at : <https://www.datasheetarchive.com/pdf/download.php?id=b0b64e7a12a46525ede945010afb4749e94667&type=P&term=vhdl%2520code%2520for%2520vending%2520machine%2520with%25207%2520segment%2520disk> (Accessed : December 3, 2022)

APPENDICES

Appendix A: Project Schematic



Appendix B: Documentation

