



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

FnB Vending Machine

GROUP A5

Darvin	1906300694
Dimas Permana Hamthadi	1906300706
Lazaruslie Karsono	1906300782
Muhammad Wafiyulloh	1906381760

PREFACE

Puji Syukur kami panjatkan kepada Tuhan Yang Maha Esa karena atas berkat dan rahmat-Nya sehingga projek akhir Perancangan sistem digital dengan tema FnB Vending Machine dapat diselesaikan dengan sebatas kemampuan yang dimiliki. Selain itu, kami juga ingin berterima kasih kepada Asisten Laboratorium Digital yang telah memberikan bimbingan untuk menuntun kami dalam menyelesaikan tugas projek yang telah diberikan.

Kelompok kami berharap laporan proyek Vending Machine ini dapat berguna untuk mengembangkan pengetahuan dan ilmu kita dengan mengimplementasikan materi Perancangan sistem digital yang telah dipelajari menjadi sebuah alat yang dapat diterapkan pada kehidupan sehari-hari. Kami juga menyadari bahwa hasil simulasi yang telah dilakukan dan penulisan laporan yang telah dikerjakan tidak sempurna dan masih terdapat kekurangan-kekurangan lainnya. Oleh karena itu, kami berharap adanya kritik, saran, dan usulan untuk projek yang telah kami kerjakan untuk lebih baik kedepannya.

Semoga laporan ini dapat membantu para pembaca laporan projek yang telah kami kerjakan. Akhir kata kami juga ingin mengucapkan terima kasih kepada seluruh anggota kelompok yang sudah bekerja keras dalam mengerjakan untuk menuntaskan projek akhir Perancangan Sistem digital ini serta pihak lainnya yang telah ikut serta membantu.

Depok, Juni 2, 2021

Group A5

TABLE OF CONTENTS

CHAPTER 1	4
INTRODUCTION	4
1.1 Background	4
1.2 Project Description	5
1.3 Objectives	5
1.4 Roles And Responsibilities	6
CHAPTER 2	6
IMPLEMENTATION	6
2.1 Equipment	6
2.2 Implementation	6
CHAPTER 3	18
TESTING AND ANALYSIS	18
3.1 Testing	18
3.2 Result	19
3.3 Analysis	20
CHAPTER 4	20
CONCLUSION	20
REFERENCES	21
APPENDICES	21
Appendix A: Project Schematic	21
Appendix B: Documentation	22

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

Dalam perkembangan ilmu pengetahuan dan teknologi yang semakin maju banyak yang dimanfaatkan untuk memenuhi kebutuhan manusia. Perkembangan teknologi ini dapat membantu kebutuhan manusia yang ditandai dengan alat yang telah diciptakan dan dioperasikan dengan manual atau pun otomatis. Selain itu, perkembangan tersebut juga berdampak pada gaya hidup manusia dalam mengkonsumsi dan memperoleh makanan atau minuman. Perkembangan untuk hal seperti itu juga didukung dengan teknologi saat ini yang dapat membantu manusia. perkembangan teknologi tersebut dapat berupa coffee machine, sistem keamanan rumah, microwave, dll.

Salah satu perkembangan yang membantu manusia untuk hal tersebut adalah vending machine yang dapat mengeluarkan sebuah makanan atau minuman yang tersedia pada mesin tersebut. Dalam mendapatkan sebuah makanan dan minuman dari mesin, pembeli harus memberikan sebuah uang rupiah untuk dapat memilih makanan atau minuman yang diinginkan. ketentuan jumlah uang yang dipenuhi cukuplah beragam pada sebuah vending machine dimana terdapat Rp 5.000, Rp 10.000, Rp 15.000, Rp 20.000, dll. ketentuan tersebut akan diklasifikasikan dengan kode tertentu dengan bilangan biner agar program dapat mengeluarkan output yang dapat memenuhi keinginan pembeli.

Berdasarkan penjelasan diatas, proyek ini akan membuat sistem vending machine dimana program dieksekusi secara sekuensial layaknya pemrograman C, python, dll. Program VHDL ini menggunakan 2 process pada architecture yaitu synchronous process dan combinational process. synchronous process ini menangani dalam signal clock yang dapat mengatur input pada program. selain itu, combinatorial process yang berfungsi untuk mencari next state dan output.

1.2 PROJECT DESCRIPTION

Proyek yang kami kerjakan berupa implementasi FSM yang menjelaskan cara kerja dari sebuah vending machine dengan menggunakan pendeskripsian berupa bahasa VHDL (Very High-Speed Integrated Circuit Hardware Description Language). Vending machine ini memiliki 4 jenis makanan dan minuman yang disediakan dengan memberikan nominal uang rupiah agar dapat bekerja berdasarkan *state* yang tersedia. Setiap makanan, minuman, dan harganya telah diklasifikasikan agar mesin dapat memahami pilihan dari pengguna. Selain itu, sistem Vending machine juga akan mengoperasikan kembalian pada pengguna setelah memilih untuk mengambil uang yang berlebih pada mesin tersebut.

Bahasa pendeskripsian program yang digunakan untuk proyek ini adalah VHDL dan akan disimulasikan dalam modelsim untuk melihat cara kerja berupa proses dan hasil input dan output yang dimasukkan.

Pada program VHDL kami menggunakan 2 process statement yaitu, Synchronous process dan Combinational process. Synchronous process ini akan mengatur input yang diberikan berdasarkan clock signal saat berada pada kondisi rising edge. Sedangkan, combinational process ini akan mencari output dan next state berdasarkan input nominal dan pemilihan produk yang diberikan. Untuk melakukan pengujian terhadap logika Vending machine yang telah dibuat, akan digunakan *software* ModelSim untuk memberikan input berupa logika Digital untuk menguji *output* yang akan diperoleh dari hasil simulasi.

1.3 OBJECTIVES

Tujuan dari proyek tertera di bawah ini:

1. Mengetahui penggunaan FSM pada cara kerja vending machine
2. Mengetahui implementasi FSM pada program VHDL
3. Dapat mendeskripsikan VHDL bersamaan dengan membentuk rangkaian
4. Mengetahui penggunaan process statement pada program VHDL

1.4 ROLES AND RESPONSIBILITIES

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

Roles	Responsibilities	Person
Program	Pendeskripsian VHDL dan Testbench	Darvin & Lazaruslie Karsono
Laporan	Penulisan Laporan	Darvin, Lazaruslie Karsono, Muhammad Wafiyulloh, Dimas Permana Hamthadi
Video	Edit Video	Dimas Permana Hamthadi

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:

- Software Source Code Editor: Notepad++
- Software modelSim - Intel FPGA Starter Edition Model Technology ModelSim
- Software diagram: draw.io (Block Diagram and State diagram)

2.2 IMPLEMENTATION

Dalam implementasi project program Food and Beverage Vending Machine dibutuhkan sejumlah uang untuk dapat menghasilkan makanan yang nantinya akan dikeluarkan oleh vending machine tersebut. Implementasi nilai input dalam program FnB Vending Machine ini, yaitu uang dengan penjelasannya pada tabel “Input Uang” dibawah ini. Terdapat empat buah nilai input uang pada program FnB Vending Machine ini, input uang akan bernilai “Rp 0” jika nilai input pada program yaitu “00”, input uang akan bernilai “Rp 5.000” jika nilai input pada program yaitu “01”, input uang akan bernilai “Rp 10.000” jika

nilai input pada program yaitu “10” dan input uang akan bernilai “Rp 20.000” jika nilai input pada program yaitu “11”.

Sedangkan untuk implementasi nilai output dalam program FnB Vending Machine ini, yaitu makanan dan minuman dengan penjelasannya pada tabel “Tabel Makanan” dan “Tabel Minuman” dibawah ini. Terdapat delapan buah hasil output yang dihasilkan pada program FnB Vending Machine ini yang dibagi menjadi empat buah hasil output berupa makanan dan empat buah hasil output yang berupa minuman.

Hasil output makanan “Permen” akan dikeluarkan jika nilai input pada program yaitu “0001” dengan implementasi harganya yaitu sebesar “Rp 5.000”, hasil output makanan “Chitato” akan dikeluarkan jika nilai input pada program yaitu “0010” dengan implementasi harganya yaitu sebesar “Rp 10.000”, hasil output makanan “Nasi Goreng” akan dikeluarkan jika nilai input pada program yaitu “0011” dengan implementasi harganya yaitu sebesar “Rp 15.000”, dan hasil output makanan “Sushi” akan dikeluarkan jika nilai input pada program yaitu “0100” dengan implementasi harganya yaitu sebesar “Rp 20.000”.

Hasil output minuman “Le Minerale” akan dikeluarkan jika nilai input pada program yaitu “0101” dengan implementasi sebesar “Rp 5.000”, hasil output minuman “Teh Pucuk” akan dikeluarkan nilai input pada program yaitu “0110” dengan implementasi sebesar “Rp 10.000”, hasil output minuman “Matcha” akan dikeluarkan nilai input pada program yaitu “0111” dengan implementasi sebesar “Rp 15.000”, hasil output minuman “Starbucks” akan dikeluarkan nilai input pada program yaitu “1000” dengan implementasi sebesar “Rp 20.000”.

Apabila uang yang diberikan terdapat sisa setelah melakukan transaksi, mesin dapat memberikan kembalian dengan kembali pada state sebelumnya yaitu state Rp 5.000, Rp 10.000, Rp 15.000. Setelah itu, kembali pada state awal. Berikut merupakan beberapa input pilihan yang tersedia pada vending machine:

Input Uang

Uang	
Input	Nominal
00	Rp 0
01	Rp 5.000

10	Rp 10.000
11	Rp 20.000

Table 2. tabel implementasi input uang

Tabel Makanan

Makanan		
Input	Output	Harga
0001	Permen	5.000
0010	Chitato	10.000
0011	Nasi Goreng	15.000
0100	Sushi	20.000

Table 3. tabel implementasi output makanan

Tabel Minuman

Minuman		
Input	Output	Harga
0101	Le Minerale	5.000
0110	Teh Pucuk	10.000
0111	Matcha	15.000
1000	Starbucks	20.000

Table 4. tabel implementasi output minuman

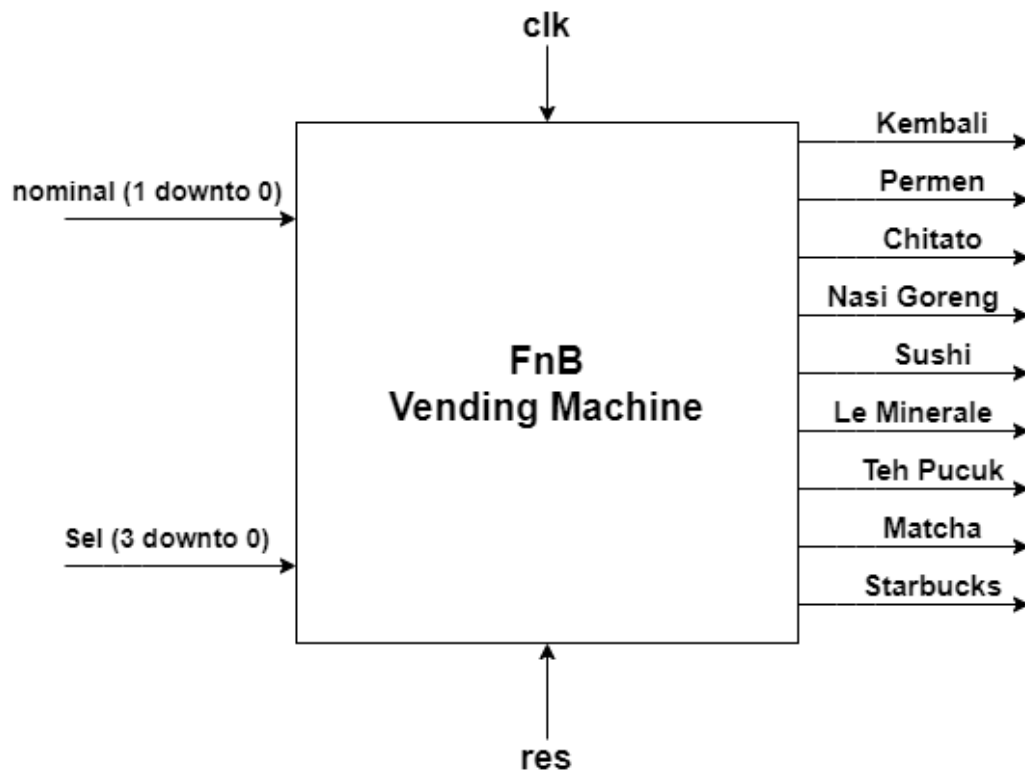


Fig 1. Blackbox FnB Vending Machine

Gambar diatas merupakan blackbox dari program FnB Vending Machine, dimana nilai input terdiri atas “nominal” yang mengidentifikasikan sebagai nilai uang pada vending machine dan “sel” sebagai nilai input untuk memilih produk yang ingin dikeluarkan. Sedangkan untuk produknya itu sendiri merupakan output yang dihasilkan pada program vending machine tersebut, yaitu ada “kembalian”. “Permen”, “Chitato”, “Nasi Goreng”, “Sushi”, “Le Minerale”, Teh Pucuk”, Matcha”, dan “Starbucks”. “clk” berfungsi sebagai clock dan “ambil” berfungsi untuk mengambil kembalian dan state akan menjadi ke state awal atau ke state “start”.

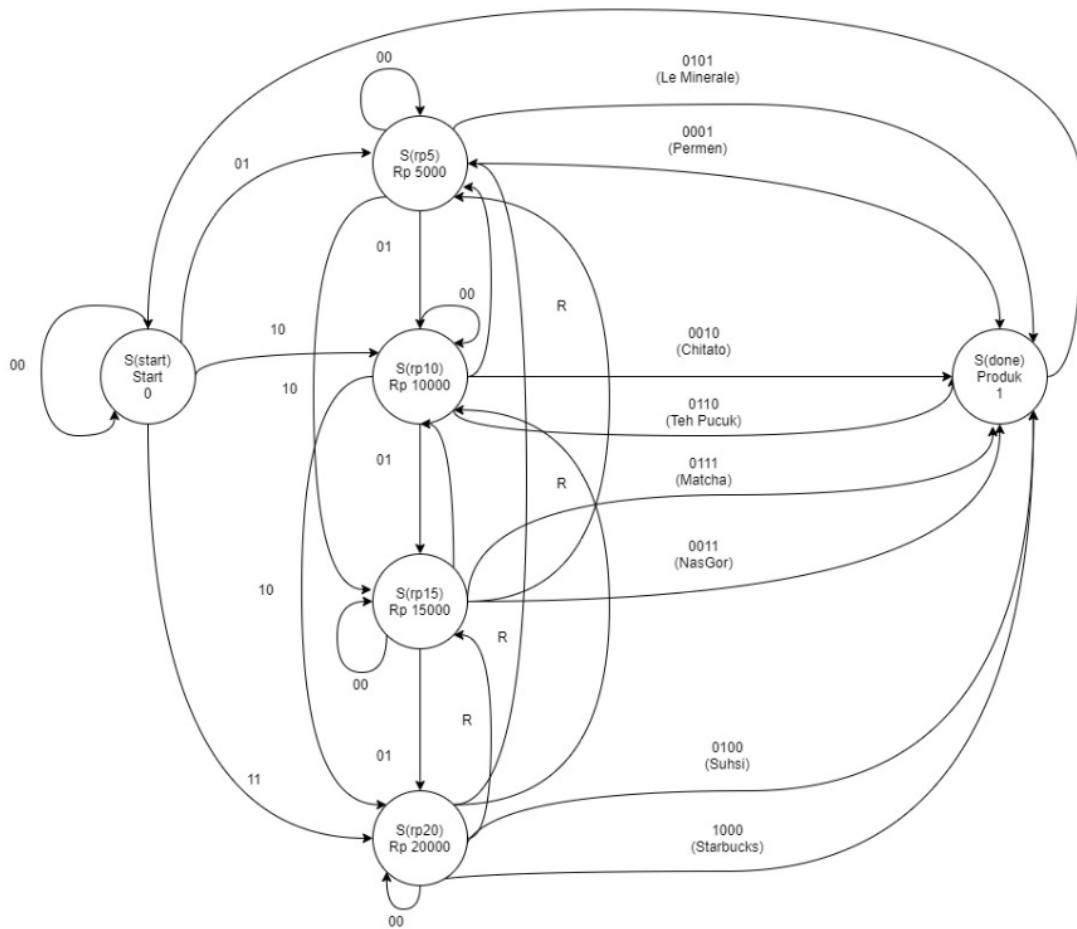


Fig 2. State diagram Moore state Machine FnB Vending Machine

Gambar diatas merupakan state diagram dari program vending machine dengan karakteristiknya yaitu Moore State Machine karena output hanya bergantung pada *state*(keadaan) saja. Pada diagram tersebut keadaan awal akan dimulai pada state *start* yang memiliki output 0 karena belum terdapat produk yang dikeluarkan. Setelah dimasukkan nominal tertentu, maka rangkaian tersebut akan berpindah state sesuai dengan jumlah uang yang dimasukkan. Dalam hal ini jika dimasukkan Rp 5000 maka akan berpindah state menuju rp5, lalu jika dimasukkan nominal yang sama akan menyebabkan berpindah pada state rp10 yang menyatakan bahwa di dalam *vending machine* terdapat jumlah uang sebesar Rp 10.000. Selanjutnya dengan uang yang telah dimasukkan, pengguna dapat memilih makanan yang diinginkan dan jika kondisi memenuhi maka *state* akan berpindah lagi pada *done* yang menandakan *vending machine* akan mengeluarkan produk. Setelah itu *state* akan berpindah lagi menuju keadaan awal jika nominal dalam mesin tersebut sudah menunjukkan 00 atau sudah tidak terdapat uang di dalamnya.

Berikut merupakan pendeskripsian **Kode VHDL** yang telah dibuat berdasarkan Block diagram dan State diagram yang telah didesain:

```
--Deklarasi Library
library IEEE;
use IEEE.std_logic_1164.all;

--entity
entity vend_mch is
    port(
        -- Input Vending Machine
        nominal : in std_logic_vector(1 downto 0) := "00"; --Input duit
        sel      : in std_logic_vector(3 downto 0) := "0000"; -- Pilih
        Produk
        clk      : in std_logic;
        ambil    : in std_logic := '0'; --Mengambil kembalian
        --Output Vending Machine
        kembali   : out string(1 to 9) := "Rp  0.000"; --Kembalian
        --Food
        permen    : out std_logic := '0';
        chitato   : out std_logic := '0';
        nasgor    : out std_logic := '0';
        sushi     : out std_logic := '0';
        --Beverages
        leminerale : out std_logic := '0';
        tehpucek  : out std_logic := '0';
        matcha    : out std_logic := '0';
        starbucks : out std_logic := '0'
    );
end vend_mch;

architecture arch of vend_mch is
    --States
    type state_types is (start, rp5, rp10, rp15, rp20, done);
    signal present_state, next_state : state_types := start;
    --Sinyal pemilihan produk dan produk yang tersedia sesuai nominal
    signal sel_prod : std_logic_vector(3 downto 0) := "0000";
    signal prod5rb : std_logic := '0';
    signal prod10rb : std_logic := '0';
    signal prod15rb : std_logic := '0';
    signal prod20rb : std_logic := '0';
begin
    =====Finite State Machine=====
    -- Synchronous
    sync_proc: process (clk, next_state, ambil)
    begin
```

```

-- Jika mengambil kembalian maka state kembali ke awal
if (ambil = '1') then
    present_state <= start;
-- Memindahkan next state sebagai present state setiap clock
elsif (rising_edge(clk)) then
    present_state <= next_state;
end if;
end process sync_proc;

-- Combinatorial
comb_proc: process (clk, present_state, nominal)
begin
--assign pilihan produk ke signal
sel_prod <= sel;

    case(present_state) is
        when start =>
            --Memindahkan next state sesuai dengan uang yang
dimasukkan

            case(nominal) is
                when "01"    => next_state <= rp5;
                when "10"    => next_state <= rp10;
                when "11"    => next_state <= rp20;
                when others => next_state <= start;
            end case;

-- State ketika dimasukkan 5 ribu
        when rp5 =>
            if (sel_prod = "0000") then
                --Jika tidak memilih produk maka membaca input duit
lagi

                case (nominal) is
                    when "01"    => next_state <= rp10;
                    when "10"    => next_state <= rp15;
                    when others => next_state <= rp5;
                end case;
            else
                --Jika dibeli produk 5ribu maka state vending machine
selesai

                case(sel_prod) is
                    when "0001"|"0101" => next_state <= done;
                    when others => next_state <= rp5;
                end case;
            end if;

-- State ketika dimasukkan 10 ribu

```

```

when rp10=>
    --Jika tidak memilih produk maka membaca input duit
    lagi
        if(sel_prod = "0000") then
            case (nominal) is
                when "01"    => next_state <= rp15;
                when "10"    => next_state <= rp20;
                when others => next_state <= rp10;
            end case;
        else
            --Membeli produk 5 ribu akan berpindah state,
            sedangkan
            --untuk pembelian produk 10 ribu state vend machine
            selesai
                case (sel_prod) is
                    when "0001"|"0101" => next_state <= rp5;
                    when "0010"|"0110" => next_state <= done;
                    when others => next_state <= rp10;
                end case;
            end if;

-- Kondisi ketika dimasukkan 15 ribu
when rp15 =>
    --Jika tidak memilih produk maka membaca input duit
    lagi
        if(sel_prod = "0000") then
            case(nominal) is
                when "01"    => next_state <= rp20;
                when others => next_state <= rp15;
            end case;
        else
            --Mendefinisikan kondisi state berdasarkan pembelian
            produk
                case(sel_prod) is
                    when "0001"|"0101" => next_state <= rp10;
                    when "0010"|"0110" => next_state <= rp5;
                    when "0011"|"0111" => next_state <= done;
                    when others => next_state <= rp15;
                end case;
            end if;

-- Kondisi ketika dimasukkan 20 ribu
when rp20 =>
    case(sel_prod) is
        when "0001"|"0101" => next_state <= rp15;
        when "0010"|"0110" => next_state <= rp10;
    end case;
end if;

```

```

        when "0011"|"0111" => next_state <= rp5;
        when "0100"|"1000" => next_state <= done;
        when others => next_state <= rp20;
    end case;

    when done =>
        next_state <= start;

    end case;
end process comb_proc;

-- Pemberian produk
cekUang: process(clk, next_state, sel_prod, ambil)
begin
    if (clk = '1') then
        --Sebagai signal untuk menyesuaikan produk yang bisa dibeli
        sesuai
        --dengan duit yang dimasukkan
        case (present_state) is
            when rp5 =>
                prod5rb <= '1';
                prod10rb <= '0';
                prod15rb <= '0';
                prod20rb <= '0';
            when rp10 =>
                prod5rb <= '1';
                prod10rb <= '1';
                prod15rb <= '0';
                prod20rb <= '0';
            when rp15 =>
                prod5rb <= '1';
                prod10rb <= '1';
                prod15rb <= '1';
                prod20rb <= '0';
            when rp20 =>
                prod5rb <= '1';
                prod10rb <= '1';
                prod15rb <= '1';
                prod20rb <= '1';
            when others =>
                prod5rb <= '0';
                prod10rb <= '0';
                prod15rb <= '0';
                prod20rb <= '0';
        end case;
    end if;
end process;

```

```

end process cekUang;

--Mengeluarkan produk sesuai yang telah dipilih
permen      <= '1' when sel_prod = "0001" AND prod5rb = '1' else
'0';
chitato     <= '1' when sel_prod = "0010" AND prod10rb = '1' else
'0';
nasgor      <= '1' when sel_prod = "0011" AND prod15rb = '1' else
'0';
sushi       <= '1' when sel_prod = "0100" AND prod20rb = '1' else '0';
leminerale  <= '1' when sel_prod = "0101" AND prod5rb = '1' else '0';
tehpucuk    <= '1' when sel_prod = "0110" AND prod10rb = '1' else '0';
matcha      <= '1' when sel_prod = "0111" AND prod15rb = '1' else
'0';
starbucks   <= '1' when sel_prod = "1000" AND prod20rb = '1' else '0';

--Menampilkan kembalian yang akan diberikan vending machine
kembali     <= "Rp 5.000" when present_state = rp5 else
              "Rp 10.000" when present_state = rp10 else
              "Rp 15.000" when present_state = rp15 else
              "Rp 20.000" when present_state = rp20 else
              "Rp 0.000";

end arch;

```

Pada pemrograman ini, VHDL menggunakan sekuensial sirkuit yang akan mengeksekusi tiap barisnya. Program VHDL menggunakan 2 process statement yaitu synchronous process dan combinatorial process. Pada synchronous process akan mempengaruhi input dengan clock sinyal saat rising edge dan mendeklarasikan kembalian jika pengguna ingin mengambil sisanya. Pada program ini menggunakan state rp5, rp10, rp15, rp20, start, dan done. Selain itu, kami menggunakan 4 input yaitu “nominal” untuk menginput uangnya, “sel” untuk memilih produk, “clk” , dan “ambil” digunakan untuk mengambil kembalian. output pada program yaitu 4 makanan dan 4 minuman yang dideklarasikan menjadi 0, serta terdapat kembalian dalam bentuk string yang menerangkan jumlah kembalian. Program ini menggunakan sinyal present state yang menjelaskan state yang dilakukan saat ini dan sinyal next state adalah state yang akan dijalankan selanjutnya.

Pada Synchronous process, dilakukan untuk mengatur inputnya menggunakan clock dan pada sync process terdapat perintah “ambil” apabila telah diberikan nilai HIGH maka state akan kembali ke start. Selain itu, untuk mengubah next statenya pula. Pada

Combinatorial process akan menjelaskan FSM pada program. Combinational process akan dijalankan berdasarkan kondisi process statement jika terdapat perubahan nilai pada present state dan nominal process statement tersebut.

Jika case pada kondisi start maka hal yang dapat dilakukan, yaitu hanya memasukkan nilai input nominal uang yang nantinya akan dimasukkan ke vending machine tersebut. nilai input nominal akan mempengaruhi state selanjutnya. Pada state rp5,10,15 vending machine dapat melakukan penginputan nominal uang kembali atau pemilihan produk yang akan dikeluarkan. Pada state rp20 vending machine hanya dapat melakukan penginputan pemilihan produk yang akan dikeluarkan, hal ini dikarenakan batas maksimal penginputan produk yaitu sebesar Rp 20.000 saja. setelah produk dikeluarkan kita dapat melakukan pemilihan untuk melakukan penambahan nominal kembali atau pemilihan produk atau mengambil kembalian. Jika memilih untuk mengambil kembalian, maka pada next statenya uang akan menjadi Rp 0.000 yang menandakan uang kembalian telah diberikan. Jika uang pada present statenya sebesar Rp 0.000 maka next state nya akan menjadi done hal ini menandakan vending machine sudah melakukan tugasnya sampai selesai. Jika pada present state nya berada pada kondisi done maka next state nya akan berubah menjadi kondisi start dan vending machine akan melakukan tugasnya dari awal yaitu untuk menunggu adanya nilai input nominal uang yang akan dimasukkan kepada vending machine tersebut.

Untuk menguji logika dari pendeskripsian VHDL, maka akan dibuat sebuah **Kode Testbench** untuk memastikan hasil yang diperoleh sudah tepat yaitu sebagai berikut:

```
--Library
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

--Entity
entity vendMch_tb is
end entity;

--Architecture
architecture vend_arch_tb of vendMch_tb is
    component vend_mch is
    port(
        -- Input Vending Machine
        nominal : in std_logic_vector(1 downto 0) := "00";
        sel      : in std_logic_vector(3 downto 0) := "0000";
```



```

        clk          : in std_logic;
        ambil : in std_logic := '0';

        --Output Vending Machine
        kembali      : out string(1 to 9) := "Rp  0.000";

        --Food
        permen        : out std_logic := '0';
        chitato       : out std_logic := '0';
        nasgor        : out std_logic := '0';
        sushi : out std_logic := '0';

        --Beverages
        leminerale: out std_logic := '0';
        tehpucek      : out std_logic := '0';
        matcha        : out std_logic := '0';
        starbucks: out std_logic := '0'
    );
end component;

--Mendefinisikan signal untuk melakukan mapping sebagai testbench
signal nominal      : std_logic_vector (1 downto 0) := "00";
signal sel          : std_logic_vector (3 downto 0) := "0000";
signal clk          : std_logic;
signal kembali      : string (1 to 9) := "Rp  0.000";
signal ambil, permen, chitato, nasgor, sushi : std_logic := '0';
signal leminerale, tehpucek, matcha, starbucks : std_logic := '0';

--Mendefinisikan banyak clock, looping, dan periodenya
constant T          : time := 20ns;
constant max_clk: integer := 9;
signal i            : integer := 0;

begin
    --Melakukan mapping pada Unit under test secara implisit
    UUT: vend_mch port map (nominal, sel, clk, ambil, kembali,
    permen, chitato, nasgor, sushi, leminerale, tehpucek, matcha,
    starbucks);

    tb:process
        type duit is array (0 to 9) of std_logic_vector (1
downto 0);
        type pil_produk is array (0 to 9) of std_logic_vector
(3 downto 0);
        type exchange is array (0 to 9) of std_logic;
        type kembalian is array (0 to 9) of string (1 to 9);

```

```

--Mendeklarasikan nilai untuk melakukan testbench
constant stream_duit : duit := (0=> "10",
1=>"00",2=>"01", 3=>"00", 4=>"01",5=>"00", 6=>"11", 7=>"00", 8=>"00",
9=>"00");

constant stream_produk : pil_produk := (0=> "0001",
1=>"0000",2=>"0110", 3=>"0000",4=>"0111", 5=>"0000", 6=>"0011",
7=>"0000", 8=>"0000", 9=>"0000");

constant ambil_duit: exchange := (0=> '0',
1=>'0',2=>'0', 3=>'0', 4=>'0',5=>'0', 6=>'0', 7=>'0', 8=>'1', 9=>'0');

constant test_kembali : kembalian := (0=> "Rp 0.000",
1=>"Rp 10.000", 2=>"Rp 5.000", 3=>"Rp 10.000", 4=>"Rp 0.000", 5=>"Rp
0.000", 6=> "Rp 0.000", 7=>"Rp 20.000", 8=>"Rp 5.000", 9=>"Rp
0.000");

begin
--Melakukan transisi clock
clk <= '0';

wait for T/2;
CLK <='1';
wait for T/2;

--Memasukkan nilai testbench sesuai dengan index loop
kepada input vending machine
if(i < max_clk) then i <= i+1;
ambil <= ambil_duit(i);
nominal <= stream_duit(i);
sel <= stream_produk(i);
else wait;
end if;

--Melakukan assert untuk mengecek kesesuaian output
vending machine
assert (kembali = test_kembali(i))
report "Hasil testbench vending machine fail pada loop
ke-" & integer'image(i) severity error;

end process;

end architecture;

```

Dalam hal ini, testbench menggunakan process statement dan *look-up table* dengan arsitektur struktural. Pada simulasi tersebut digunakan process statement dikarenakan instruksi di dalam statement tersebut akan dijalankan secara sekuensial dan akan dilakukan

iterasi sebanyak 8 kali untuk menguji setiap percobaan input nominal dan pemilihan produk pada *vending machine*. Kemudian, penggunaan dari *lookup* table adalah untuk menyimpan beberapa kombinasi *value* yang nantinya akan di assign pada input *vending machine* pada setiap iterasinya sebanyak jumlah *max_clk* yang telah didefinisikan yaitu sebanyak 9 kali. Pada setiap iterasi akan diberikan nilai berbeda pada input nominal, sel, serta ambil berdasarkan index pada langkah iterasi tersebut yang dikontrol dengan menggunakan *if* statement. Bagian akhir dari testbench diberikan construct *assert* dan *report* dengan tingkat severity *error* untuk memberikan informasi apabila ketika simulasi terdapat kesalahan berupa hasil output yang tidak sesuai dengan logikanya namun rangkaian tetap dapat berjalan.

CHAPTER 3

TESTING AND ANALYSIS

3.1 TESTING

Hasil percobaan simulasi pada Modelsim berdasarkan kode VHDL yang telah dibuat:

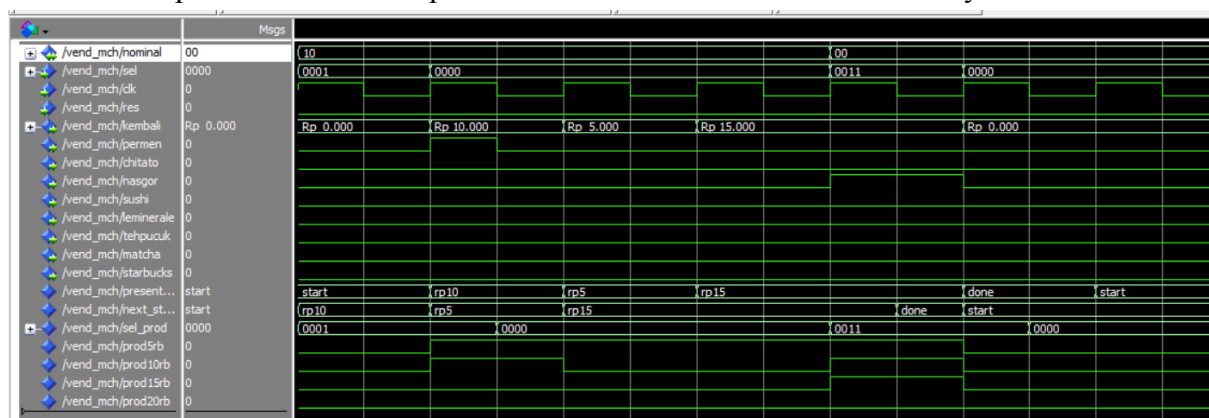


Fig 3. Hasil percobaan simulasi pada Modelsim

Dari hasil percobaan simulasi tersebut, program sudah berjalan dengan benar dan sesuai. Hal pertama yang harus dilakukan dalam melakukan simulasi program, yaitu dengan mengubah nilai input “nominal” sesuai dengan besaran nominal yang diinginkan, dalam simulasi tersebut nominal yang dimasukkan senilai “Rp 10.000” dengan nilai input “10”. Setelah nominal, selanjutnya mengubah nilai input “sel” sesuai dengan pilihan produk yang ingin dikeluarkan, dalam simulasi tersebut produk yang dipilih yaitu “Permen” dengan nilai inputnya yaitu “0001”. Hal terakhir yang harus diperhatikan, yaitu mengubah nilai “clk”

Setelah mengubah nilai input sesuai dengan ketentuan diatas, maka simulasi dapat dijalankan. Setelah dijalankan pertama kali, nilai input “sel” yang pertama-tama bernilai “0001” diubah menjadi “0000” agar produk tidak melakukan dua kali pengeluaran. Setelah itu dapat dilakukan run kembali agar dapat melihat hasil dari pengeluaran output yang dipilih dan melihat nominal uang yang dimiliki yaitu “Rp 10.000”. Setelah melakukan run lagi maka uang kembali akan berubah menjadi “Rp 5.000” hal ini menandakan bahwa uang yang sebelumnya “Rp 10.000” telah menyisakan “Rp 5.000” dalam vending machine tersebut. Karena nilai input “nominal” tidak diubah menjadi “00” maka dari itu nominal yang sebelumnya adalah “Rp 5.000” bertambah menjadi “Rp 15.000”.

3.2 RESULT

Fig 4. Simulasi dengan menggunakan Testbench pada ModelSim

Melalui value yang telah diberikan pada testbench maka *software* ModelSim akan memasukkan seluruh input secara bersamaan dan akan diperoleh output berdasarkan logika input yang telah diberikan yaitu berupa produk yang diperoleh dari hasil pemilihan jika jumlah uang mencukupi.

Berdasarkan instruksi assert dan report yang telah diberikan dengan pengecekan jumlah kembalian, maka diperoleh hasil sebagai berikut:



```
Transcript
# Loading ieee.numeric_std(body)
# Loading work.vendmach_tb(vend_arch_tb)
# Loading work.vend_mch(arch)
# WARNING: No extended dataflow license exists
add wave -position insertpoint sim/vendmach_tb/*
# ** Warning: (vsim-WLF-5000) WLF file currently in use: vsim.wlf
# File in use by: Dar Hostname: LAPTOP-DAR ProcessID: 22160
# Attempting to use alternate WLF file ".\\wlfst20adn".
# ** Warning: (vsim-WLF-5001) Could not open WLF file: vsim.wlf
# Using alternate file: .\\wlfst20adn
VSI90> run -all
VSI91>
```

Fig 5. Transcript pengecekan hasil testbench

3.3 ANALYSIS

Berdasarkan hasil percobaan berupa simulasi pada ModelSim maka dapat diketahui bahwa logika pendeskripsian VHDL yang telah dibuat sudah benar karena tidak terdapat report berupa kesalahan yang terjadi pada simulasi. Hal tersebut dapat juga dibuktikan dengan melakukan analisa terhadap hubungan input dan output yang didapatkan pada simulasi tersebut. Dalam *testbench* ini diberikan *case* sebanyak 9 kali dengan menyesuaikan jumlah batas *clocknya*, sehingga instruksi dalam process statement tersebut akan berjalan sesuai dengan jumlah tersebut juga secara sekuensial.

Pada kondisi awal diberikan value testbench untuk input pada nominal mata uang sebesar Rp 10.000 dengan memberikan logika “10” sehingga pada index pertama terdapat kembalian penuh yang ditandai mesin sedang berada pada state rp10. Kemudian diberikan logika sel 0001 yang akan memilih produk permen seharga Rp 5000, sehingga jumlah nominal yang tersisa berjumlah Rp 5000 juga. Begitu pula dengan kondisi pembelian produk selanjutnya. Sampai pada iterasi ke 5 akan dilakukan pembelian produk Matcha seharga Rp 15.000 namun mata uang yang dimasukkan hanya bernilai Rp 5000 sehingga produk tersebut tidak akan dikeluarkan dari vending machine tersebut. Demikian pula terdapat kondisi

selanjutnya ketika diberikan mata uang Rp 20.000 maka mesin tersebut akan berpindah present state menjadi rp20 dimana pada percobaan ini dilakukan pembelian Nasi goreng yang menyisakan Rp 5000 dan pembeli memutuskan untuk mengambil kembalian dengan memberikan logika 1 pada sinyal ambil yang akan menyebabkan state kembali ke start atau keadaan awal dari mesin sehingga mata uang tersisa Rp 0 dan *vending* machine akan siap untuk menerima input nominal lagi untuk melakukan perpindahan state pada setiap transisi positive edge pada clocknya.

Perlu diketahui juga proyek ini bekerja berdasarkan perpindahan state menggunakan konsep Finite State machine melalui nilai digital sehingga jumlah mata uang yang dapat ditampung terbatas pada jumlah *state* yang didefinisikan. Dalam kasus ini mesin hanya dapat menerima range input nominal mata uang mencapai Rp 20.000 saja yang juga telah disesuaikan dengan harga produk yang dapat dibeli.

CHAPTER 4

CONCLUSION

Konsep dari Finite State machine atau FSM dapat diimplementasikan pada sebuah FPGA yang dapat diterapkan dalam kehidupan sehari-hari untuk merancang suatu rangkaian digital. Salah satu contohnya adalah untuk mendeskripsikan logika dalam sebuah Vending machine menggunakan jenis Moore State machine yang output produknya akan diperoleh berdasarkan perubahan *statenya*. Pada blok diagram dari mesin tersebut terdapat port input berupa nominal untuk memasukkan mata uang, sel untuk pemilihan produk, clock untuk melakukan sinkronisasi rangkaian, dan ambil untuk mendapatkan kembalian dari mata uang yang tersisa. Sedangkan pada output terdiri dari port kembali untuk memberikan kembalian mata uang serta produk-produk yang akan dikeluarkan berdasarkan pilihan dari pengguna. Pada vending machine tersebut juga ditampilkan mata uang yang tersisa melalui penggunaan *conditional signal assignment* untuk mengecek keberadaan present state berdasarkan uang yang tersisa. Maka, cara kerja dari mesin tersebut berdasarkan FSM adalah dengan menggunakan arsitektur *behavioral style* dimana pada awalnya sistem tersebut akan berada pada *state* start yang akan berpindah state berdasarkan nominal yang diberikan melalui

instruksi pada *process statement*. Kemudian, ketika produk telah dipilih dan berhasil dikeluarkan maka mesin akan menuju state *done* dengan output 1 yang menandakan proses telah selesai dan akan kembali pada *state* awal.

Untuk melakukan pengujian terhadap cara kerja dari Vending machine yang telah dibuat maka dapat disimulasikan dengan menggunakan *testbench*. Pendeskripsian kode *testbench* dibuat menggunakan arsitektur Structural dengan menggunakan *lookup* table untuk memberikan beberapa kombinasi nilai terhadap input pada mesin secara sekaligus menggunakan metode iteratif. Dengan demikian dapat dilakukan pengecekan terhadap logika dari rangkaian yang telah dibuat. Dalam hal ini untuk setiap pemilihan produk yang dapat dibeli sesuai dengan jumlah nominal yang telah dimasukan pada mesin akan dikeluarkan dengan representasi logika HIGH atau '1' , sedangkan jika prduk yang dipilih tidak sesuai produk tidak akan dikeluarkan dengan logika digital LOW atau '0'.

Sehingga dapat disimpulkan bahwa pengimplementasian dari Finite State machine dapat digunakan untuk merangkai suatu rangkaian digital dengan prinsip kerja berdasarkan perpindahan *state* untuk menghasilkan logika tertentu. Kemudian untuk memastikan bahwa konsep rangkaian yang telah dirangkai sudah tepat dapat digunakan testbench untuk menguji output yang akan diperoleh berdasarkan beberapa kombinasi input, dimana jika terdapat kesalahan akan disampaikan dalam bentuk *report* pada simulasi.

REFERENCES

- [1] B.Mealy, F.Tappero (2018). Free Range VHDL – Introduction to VHDL
- [2] Pong P. Chu (2008) “FPGA PROTOTYPING BY VHDL” Hoboken, New Jersey; A John Wiley & Sons, Inc
- [3] K. Dbouk, B. Jajou, K. Abbas, S. Nissan “Vending Machine” Electrical and Computer Engineering Department School of Engineering and Computer Science Oakland University, Rochester, MI

- [4] Implementing a Finite State Machine in VHDL (2015), Retrieved 4 June 2021 from <https://www.allaboutcircuits.com/technical-articles/implementing-a-finite-state-machine-in-vhdl/>
- [5] Finite State Machine Mealy and Moore (2015), Retrieved 5 June 201 from <https://www.xilinx.com/support/documentation/university/Vivado-Teaching/HDL-Design/2015x/VHDL/docs-pdf/lab10.pdf>

APPENDICES

Appendix A: Project Schematic

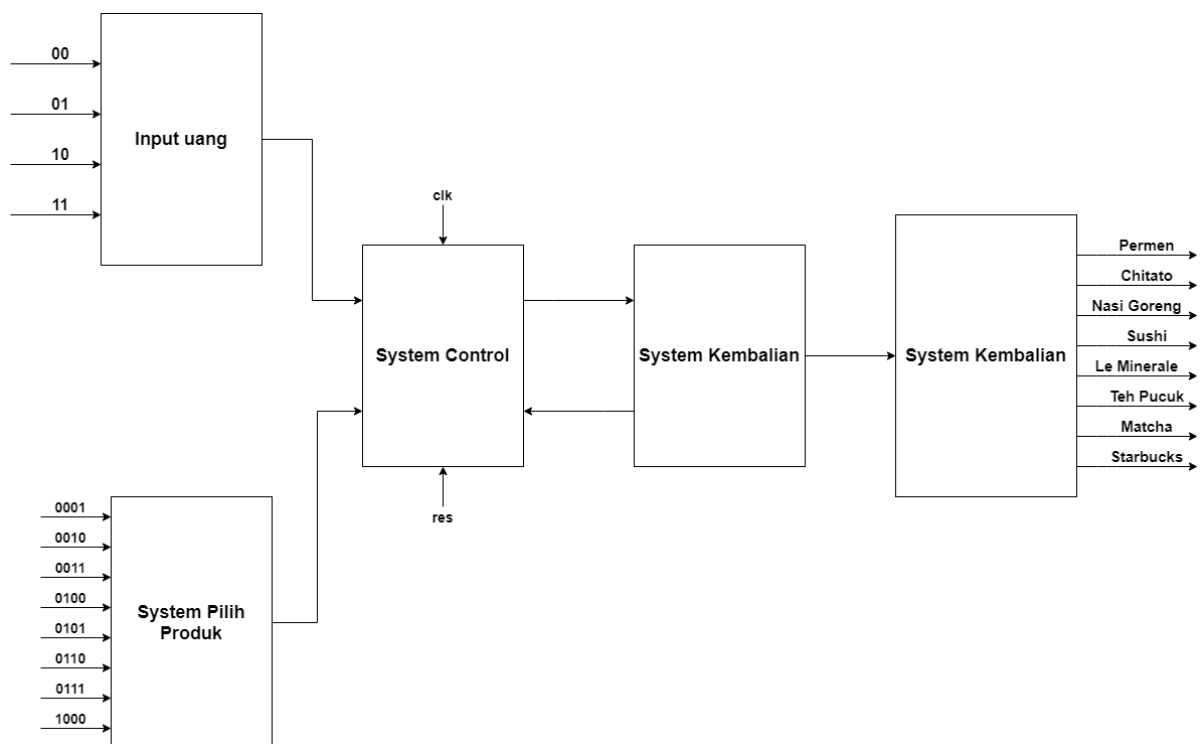


Fig 6. Skema FnB Vending Machine

Appendix B: Documentation

Proses pembuatan State Diagram FnB Vending Machine :

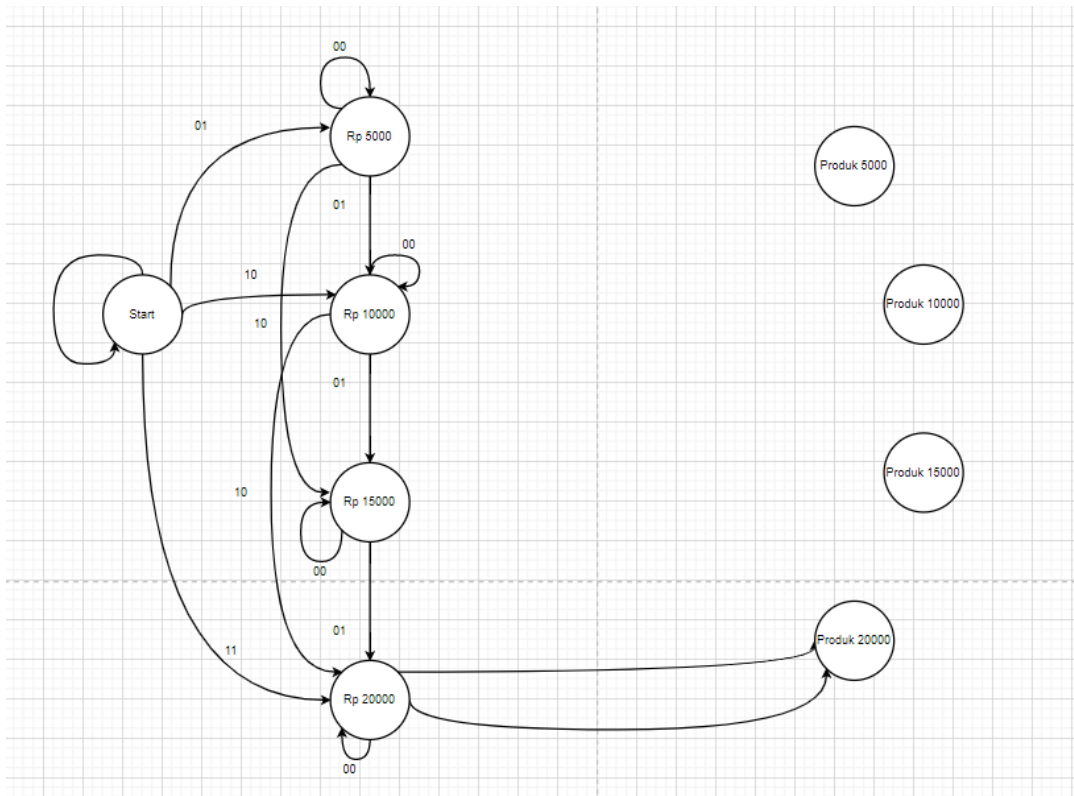


Fig 7. Proses pembuatan state diagram FnB Vending Machine

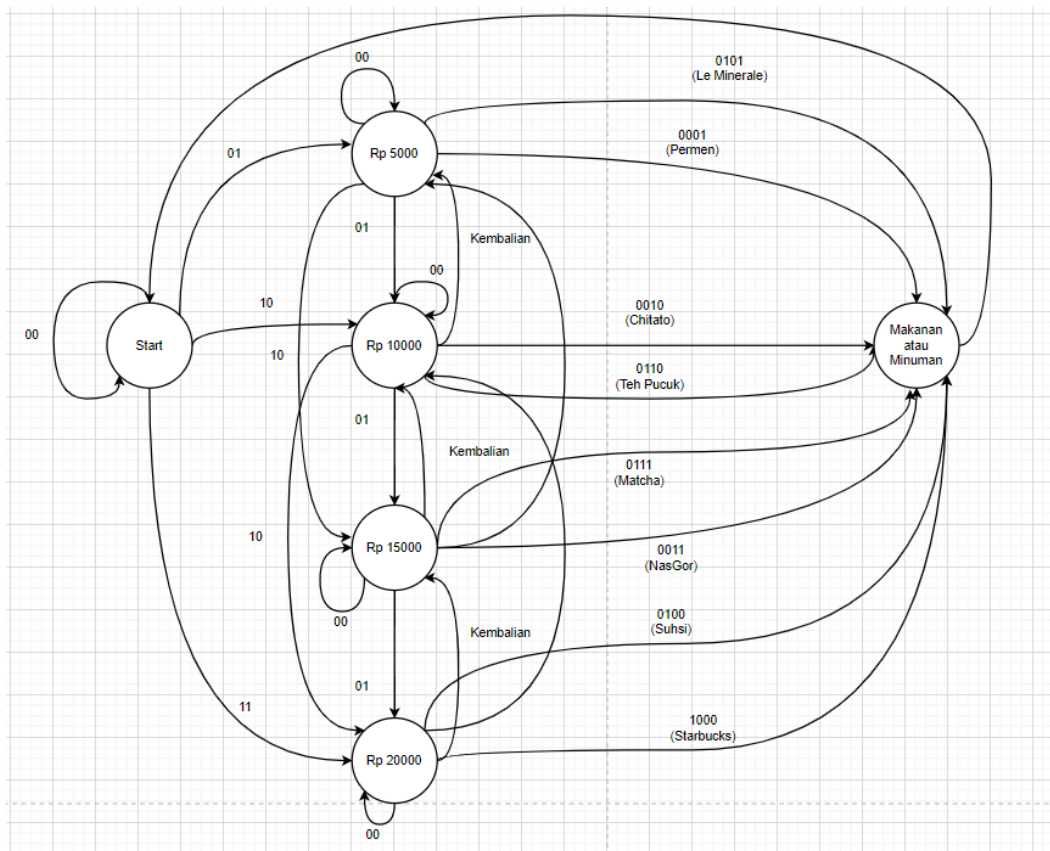


Fig 8. Proses pembuatan state diagram FnB Vending Machine

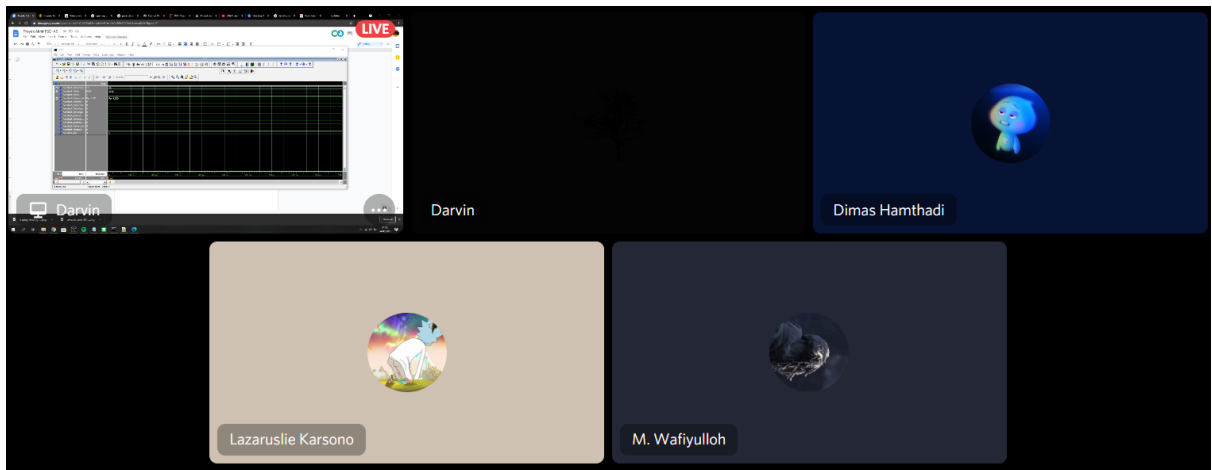


Fig 9. Dokumentasi diskusi mengenai simulasi program VHDL