

# MERANCANG DESAIN MESIN PENJUAL PULSA ELEKTRIK OTOMATIS BERBASIS FPGA

*Priska Restu Utami*

*Jurusan Teknik Elektro, Universitas Gunadarma  
priska@staff.gunadarma.ac.id*

## Abstrak

*Sebagai inovasi, mesin penjual otomatis bisa dibuat untuk pengisian pulsa elektrik. Mesin penjual otomatis ini berbasis FPGA. Hal ini dapat dibuat fleksibel untuk memodifikasi dan mengkonfigurasi ulang data yang ada dalam modul. Konsep yang digunakan dalam merancang mesin penjual ini adalah dengan menggunakan konsep finite state machine yang berhingga dan logika pembentuk keluaran. Setelah disain, kemudian dilakukan simulasi dengan menggunakan software Xilinx ISE 9.2i yang berisi simulasi perilaku dan Post Route simulasi. Hasil simulasi menunjukkan sistem mesin penjual otomatis mampu berjalan seperti skenario. Penulisan ini telah dirancang sistem pengisian pulsa elektronik otomatis dengan 10 skenario. Skenario tersebut berisi kemungkinan yang terjadi saat sistem mesin penjual otomatis beroperasi. Dari masing-masing skenario, hasil simulasi dibandingkan dengan output yang ditampilkan bersamaan dengan proses tersebut. Dari hasil simulasi semua skenario dapat digunakan untuk berbagai jenis operator seluler dan dengan nilai nominal pulsa yang berbeda.*

**Kata kunci:** *finite state machine, FPGA, output forming logic, vending machine*

## DESIGNING ELECTRICAL PULSES (MOBILE-PAYMENT) VENDING MACHINE BASED ON FPGA

### Abstract

*As an innovation, vending machines can be builded for charging electrical pulses. This vending machine is FPGA based. It can be builded flexible to modify and reconfigure existing data in the module. The concept used in designing this vending machine is to use finite state machine finite concept and output-forming logic. After designing, then simulate using Xilinx ISE 9.2i software that contains behavioral simulation and Post Route simulation. The simulation results show that the vending machine system is capable of running like a scenario. This writing has been designed automatic electronic charging system with 10 scenarios. The scenario contains the possibilities that occur when the vending machine system operates. From each scenario, the simulation results compared with the output shown along with the process. From the simulation results, all scenarios can be use for different types of mobile operators and with different pulse nominal values.*

**Keywords:** *finite state machine, FPGA, output forming logic, vending machine*

## PENDAHULUAN

Di era modern, segala aktivitas manusia didunia dipermudah, tentunya dengan bantuan teknologi yang selalu

berkembang. Salah satunya adalah aktivitas jual beli. Kegiatan menjual sekarang sudah bisa digantikan oleh mesin penjual otomatis atau disebut juga *vending machine*. Dengan mesin penjual otomatis

pembeli dapat melakukan transaksi sendiri tanpa harus dilayani oleh tenaga operator. Selain menjual kebutuhan manusia seperti makanan, minuman dan rokok, *vending machine* juga bisa untuk menjual tiket atau pun pulsa elektrik. Penjualan pulsa elektrik konvensional dilakukan interaksi antara penjual dan pembeli. Salah satu kendala dari transaksi tersebut adalah jika pengisian pulsa dibutuhkan di waktu yang tidak memungkinkan penjual dapat melayani pembeli. Oleh sebab itu dengan adanya mesin penjual pulsa otomatis, pembeli dapat dengan mudah melakukan transaksi di waktu kapanpun.

Secara umum, sistem kontrol mesin penjual otomatis masih menggunakan mikrokontroler sehingga cenderung statis dan sangat sulit untuk berubah, baik itu masukan nominal uangnya, maupun jumlah dan harga produknya [Wini, 2007]. Oleh sebab itu, diciptakan sebuah sistem kontrol yang memungkinkan adanya kemudahan konfigurasi ulang oleh pengguna agar dapat diaplikasikan sesuai kebutuhan yang bersifat dinamis. Untuk itu dirancang sistem kontrol mesin penjual otomatis berbasis FPGA (*Field Programmable Gate Array*). FPGA adalah alat yang cocok untuk disain rangkaian digital prototip, terutama dengan skala kompleksitas yang besar. Selain bisa dikonfigurasi ulang, jumlah gerbang dan flip-flop di FPGA juga sangat banyak.

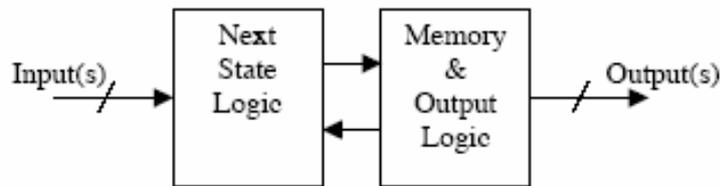
Penelitian ini membahas bagaimana mendisain mesin penjual pulsa elektrik secara otomatis yang berbasis FPGA. Penggunaan FPGA dalam sistem ini untuk menggantikan fungsi mikrokontroler sebagai sistem kontrol. Sehingga sistem pada *vending machine* dapat dengan mudah diatur ulang atau ditambah konfigurasinya.

## METODE PENELITIAN

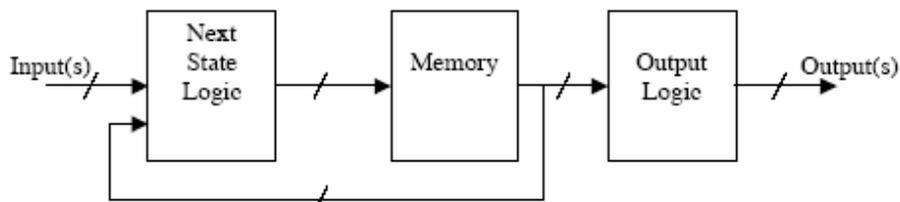
Penelitian terkait *vending machine* telah dilakukan oleh beberapa peneliti sebelumnya, diantaranya adalah "*Micro-controller based sms micro e-load vending machine*". Penelitian tersebut dilakukan karena di bawah masalah yang ada di negara mereka. Adanya kebutuhan untuk mengotomatisasi sistem pemuatan otomatis saat ini yang diterapkan di Filipina. Seperti yang terlihat di kebanyakan toko, layanan reload otomatis prabayar sedang dilakukan secara manual dengan vendor menggunakan telepon genggamnya untuk melakukan transaksi [Joshua, 2006]. Pada tahun yang sama, peneliti lainnya melakukan penelitian tentang merancang dan membangun *vending machine* untuk kantor dengan menggunakan mikrokontroler. Proyeknya bertujuan untuk merancang dan menerapkan mesin penjual otomatis untuk penggunaan kantor [Faizal, 2006]. Dalam penulisan lainnya, menjelaskan bahwa, mesin penjual otomatis dengan sistem control FPGA memiliki cenderung tidak statis dan tidak sulit untuk memodifikasi input nominal uang serta harga dan variasi produk [Wini, 2007].

### *Finite State Machine*

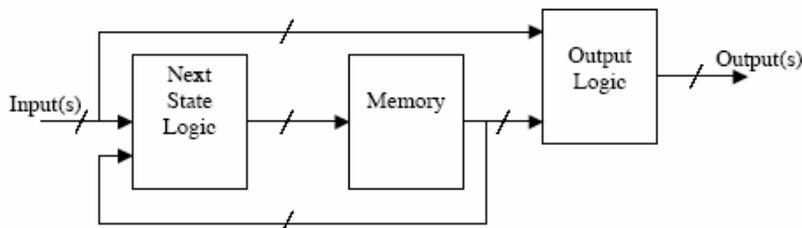
Untuk merancang *vending machine* dibutuhkan konstruksi logika yang digunakan untuk mendefinisikan sifat dari kerja sistem *vending machine*. Konsep itu dikenal dengan *Finite State Machine* (FSM). Dalam teori dijelaskan *Finite State Machine* menggunakan kombinasi latch dan gerbang logika untuk sistem yang mempunyai memori. *State machine* digambarkan dengan *state diagram* yang didefinisikan dari *state transition table*. *State machine* diperlihatkan pada Gambar 1.



Gambar 1. State Machine



Gambar 2. Moore machine



Gambar 3. Mealy machine

State machine dapat diklasifikasikan menjadi 2 tipe:

- Outputnya tergantung pada present state yang dikenal dengan *Moore machine* yang diperlihatkan pada Gambar 2.
- Outputnya merupakan kombinasi present state dan input yang dikenal dengan *Mealy machine* yang diperlihatkan pada Gambar 3.

Present state didefinisikan pada output memori. Beberapa contoh aplikasi state machine dapat dilihat pada *embedded computing system*, dimana komputer dirancang untuk menunjukkan beberapa tugas tertentu, seperti lampu merah, elevator dan juga *vending machine* yang bisa dirancang dengan menggunakan pendekatan finite state machine.

### FPGA (*Field Programmable Gate Array*)

FPGA merupakan sebuah IC digital yang sering digunakan untuk mengimplementasikan rangkaian digital. Bila dilihat dari segi namanya, *Field Programmable* dapat diartikan bahwa FPGA ini bersifat dapat dirancang sesuai dengan keinginan dan kebutuhan user atau pemakai tanpa melalui tahap “burn” di laboratorium atau di “hardwire” oleh pabrik piranti. Sedangkan *Gate Array* artinya bahwa FPGA ini terdiri atas gerbang-gerbang digital dimana interkoneksi masing-masing gerbang tersebut dapat dikonfigurasi antara satu sama lainnya.

FPGA merupakan sebuah IC digital yang bersifat programmable. User dapat memakai IC digital ini secara berulang-ulang untuk menyesuaikan program

apa yang akan didownload ke dalam FPGA ini. Program tersebut nantinya akan dibuat oleh user menggunakan *software* yang ada untuk kemudian disimulasikan. Setelah simulasi berjalan lancar dan berhasil, program tersebut siap untuk didownload ke dalam FPGA.

Spartan 3E merupakan salah satu keluarga FPGA yang diproduksi oleh Xilinx. Spartan 3E tipe Xc3S500E yang digunakan dalam penelitian ini memiliki 500000 gerbang logika. IC Xilinx ini dapat diprogram dan dihapus dengan waktu yang tidak terbatas, dengan VHDL sebagai bahasa pemrogramannya pada perangkat lunak Xilinx *Development System*.

Secara umum arsitektur FPGA Spartan-3E terlihat pada Gambar 4, terdiri dari *Configurable Logic Blocks (CLBs)*, *Input/Output Blocks (IOBs)*, Block RAM, *Multiplier Blocks* dan *DigitalClock Manager (DCM) Blocks, Interkoneksi*.

1. *Configurable logic blocks*: melakukan fungsi logika, look up table based complex structure, implement the sequential circuit, dan *Input/Output Blocks (IOBs)* sebagai interface antara external package pin dari device dan internal user logic

2. *Multiplier Blocks* melakukan tugas utama yaitu perkalian numerik *two's complement* tapi bisa juga melakukan beberapa aplikasi seperti menyimpan data dan *barrel shifting*.

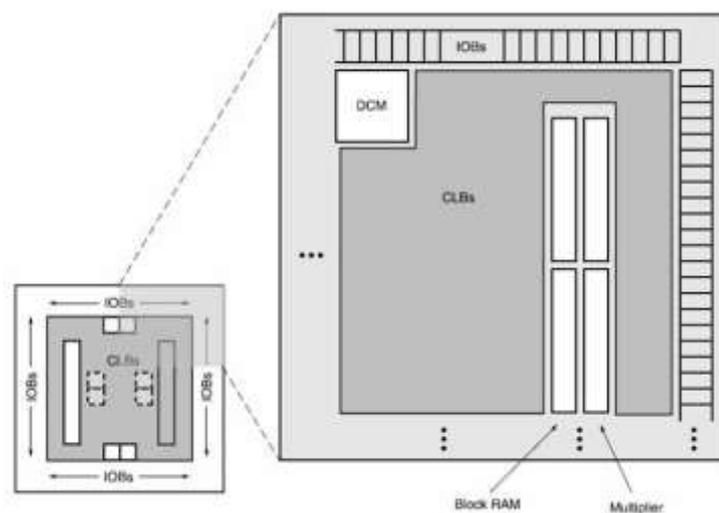
3. *Digital Clock manager (DCM)* memberikan fleksibilitas kelengkapan kontrol *over clock* frekuensi, *phase shift* dan *skew*.

4. *Programmable Interconnect* : berisi *wire segments* dan *programmable switches* dan menghubungkan antar *Configurable Logic Blocks* yang berbeda.

### Output Forming Logic

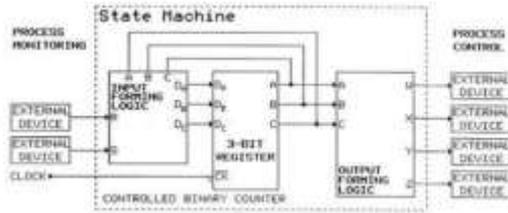
*Output forming logic* hampir sama dengan translasi pada counter akan tetapi *Output forming logic* mempunyai fungsi lebih untuk perancangan yang *glitch-free* dan di gunakan untuk masukan yang lebih kompleks ke suatu dekoder yang merupakan basis rangkaian digital berdasarkan output.

Untuk mendesain *state machine* dibutuhkan 3 bagian, yaitu register biner berikut tabel kebenarannya, *input forming logic*, *output forming logic*.

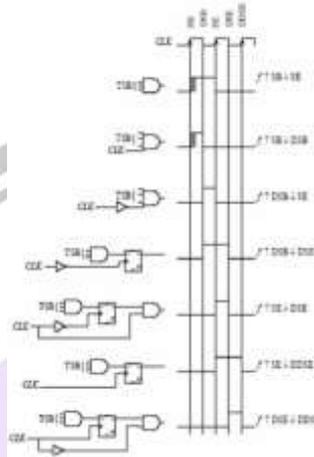


Gambar 4. Arsitektur FPGA Spartan-3E (Xilinx, 2011)

SB : State beginning  
 SE : State End  
 DSB : Delayed State Beginning  
 DSE : Delayed State End  
 DDSE : Double Delayed State End



Gambar 5. Diagram *State machine*



Gambar 6. Rangkaian Pembangkit Pulsa Output *Forming Logic*

Beberapa rangkaian untuk membangkitkan pulsa untuk output forming logic dapat dilihat pada Gambar 6. Terdapat beberapa tahap atau proses dalam perancangan sistem penjual otomatis yang harus dilakukan, pertama dilakukan pengidentifikasi masalah. Dimana ditentukan input dan output dari rangkaian. Kemudian dilakukan pengidentifikasi keadaan dilanjutkan dengan penggambaran *state diagram*. Dari *state diagram* dapat ditentukan tabel kebenaran. Dengan bantuan *Karnough Map*(K-Map) dapatlah dibentuk persamaan aljabar Boolean sederhana untuk sebagai masukan pada D Flip-Flop atau disebut sebagai *Input Forming Logic*(IFL). D Flip-flop digunakan karena lebih mudah diimplementasikan dalam merancang sistem.

Sebagai *Output Forming Logic* (OFL) dipilih gelombang  $\uparrow$ DSB  $\downarrow$ SE karena rangkaian yang dihasilkan *Output Forming Logic* ini sederhana, yaitu hanya mempergunakan gerbang AND. Selanjutnya menentukan persamaan rangkaian *Output Forming Logic* berdasarkan hasil

dari tabel kebenaran. Proses selanjutnya menggambarkan hasil persamaan rangkaian digital yang diperoleh dan mensimulasikan rangkaian tersebut. Hasil simulasi dianalisa, apakah rangkaian yang telah dibuat sesuai dengan yang diharapkan. Dalam proses pembuatan disain ini dibantu dengan menggunakan Design tool Xilinx ISE 9.2i. Setelah di disain, sistem tersebut dapat dilakukan simulasi sebelum diimplementasikan ke dalam FPGA Xilinx Spartan 3-E.

Pada Gambar 7 diperlihatkan rangkaian dari sistem *vending machine* secara keseluruhan. Rangkaian tersebut terdiri dari modul-modul untuk pembentuk rangkaian menjadi satu kesatuan sistem. Masing-masing modul berperan aktif untuk menjalankan sistem mesin penjual otomatis. Jika salah satu modul diabaikan maka akan berdampak pada hasil output dari sistem mesin penjual otomatis tersebut.

Modul-modul ini dibentuk berdasarkan hasil perumusan dengan *state diagram* yang kemudian di translasi ke state

translasi dan dibuat dengan menggunakan tabel kebenaran.

1. Modul 'Menu Utama'

Pada proses transaksi pembelian, menu awal yang akan muncul pada mesin penjual otomatis yaitu menu utama. Modul ini akan menghasilkan output menu utama disimbolkan dengan 'mu', jika input yang diberikan pada a,b,c,d,e dan clock bernilai 0. Berdasarkan konsep dari FSM yang telah dibuat kondisi tersebut pada saat state a dengan *statenumber* '00000'.

2. Modul 'IFL'

Modul 'IFL' merupakan modul yang berisikan rangkaian logika sebagai *input forming logic*. Rangkaian IFL ini dirancang sesuai dengan hasil yang didapat dari tabel kebenaran dan dari penyederhanaan persamaan-persamaan dengan menggunakan *Karnough Map*. Dalam modul ini menyimpan semua input yang digunakan pada sistem mesin penjual otomatis diantaranya perintah memilih jenis operator seluler, besar nominal pulsa, memberi inputan uang, memasukkan nomor hp, pengontrol uang yang dimasukkan, dan tombol menu seperti clear, cancel, enter, dan lain-lain.

3. Modul 'Jenis Operator seluler'

Dari modul 'jenis operator seluler' berisikan 10 sub modul lagi, yakni penjabaran dari jenis operator seluler itu sendiri beserta 5 jenis nominal pulsa. Modul ini menjelaskan logika dalam pemilihan nominal pulsa sesuai dengan input yang diberikan. Jika sudah dipilih salah satu jenis operator seluler, maka dalam sub modul tersebut diberikan 5 pilihan yang dijadikan sebagai input yaitu 'in\_1' untuk memilih nominal pulsa 5000, 'in\_2' untuk memilih nominal pulsa 10.000, 'in\_3' untuk memilih nominal pulsa 20.000, 'in\_4' untuk memilih nominal pulsa 50.000, dan 'in\_5' untuk memilih nominal pulsa 100.000. Selain itu terdapat pula input 'menu' yang

dapat digunakan untuk membatalkan transaksi dan output yang dihasilkan yaitu kembali ke menu utama.

4. Modul 'Mengeluarkan uang'

Pada proses transaksi pembelian, jika pembeli membatalkan transaksinya setelah terlanjur memasukkan uang, maka uang tersebut akan kembali keluar. Kondisi tersebut dirancang pada modul 'mengeluarkan uang'. Modul ini akan menghasilkan output mengeluarkan uang(dalam simulasi disimbolkan dengan 'v') jika diberikan input 1 pada a, input 0 pada b,c,d,e dan clock. Berdasarkan konsep dari FSM yang telah dibuat kondisi tersebut terjadi pada state r dengan *statenumber* '10000'.

5. Modul 'Mengirim Pulsa'

Pada proses akhir transaksi jika dilakukan proses transaksi yang normal, terdapat proses pengiriman pulsa. Kondisi tersebut dirancang pada modul 'Mengirim Pulsa'. Modul ini akan menghasilkan output mengirimkan pulsa (dalam simulasi disimbolkan dengan 'mp') jika pada a dan c diberikan input 1 sedangkan pada b,d,e dan clock diberikan input 0. Berdasarkan konsep dari FSM yang telah dibuat kondisi tersebut terjadi pada state u dengan *statenumber* '10100'.

6. Modul 'Mengeluarkan Struk'

Modul ini berfungsi untuk melakukan proses mengeluarkan struk jika proses transaksi telah disetujui oleh pembeli setelah memasukkan uang. Modul ini akan menghasilkan output mengeluarkan struk(dalam simulasi disimbolkan dengan 's') jika pada a,d dan e diberikan input 1 sedangkan pada b,c dan clock diberikan input 0. Berdasarkan konsep dari FSM yang telah dibuat kondisi tersebut terjadi pada state t dengan *statenumber* '10011'.

7. Modul 'Menu Operator'

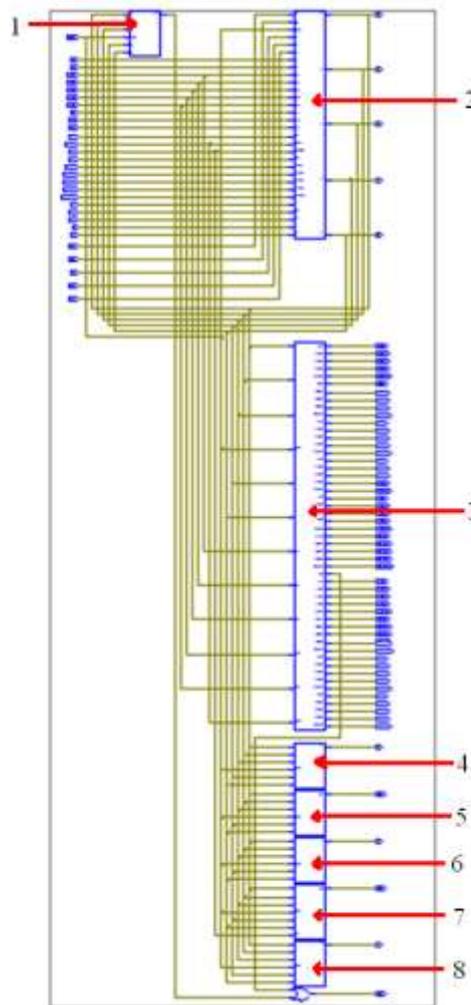
Pada proses transaksi pembelian, setelah muncul menu utama maka

selanjutnya akan tampil ‘menu operator’. Modul ini akan menghasilkan output menu operator (dalam simulasi disimbolkan dengan ‘mo’) jika input yang diberikan pada a,b,c,d,e dan clock bernilai 0 sedangkan pada MA diberikan input 1. Berdasarkan konsep dari FSM yang telah dibuat kondisi tersebut terjadi pada transisi antara state a dan state b dengan input ‘MA’.

8. Modul ‘Mengeluarkan uang kembalian’ Pada proses transaksi pembelian, jika uang yang dimasukkan oleh pembeli lebih besar dari harga jual maka dari modul ini akan mengeluarkan uang kembalian. Dari modul ini akan mengeluarkan uang kembalian

(dalam simulasi disimbolkan dengan ‘o’) jika pada a dan d diberikan input 1 sedangkan pada b,c,e dan clock diberikan input 0. Berdasarkan konsep dari FSM yang telah dibuat kondisi tersebut terjadi pada state s dengan statenumber ‘10010’

Skenario pengujian dalam sistem ini terdiri dari 10 skenario. Skenario tersebut berisikan kemungkinan-kemungkinan yang terjadi jika sistem penjual otomatis sedang beroperasi. Dari 10 skenario ini dapat digunakan untuk setiap pemilihan jenis operator dan nominal operator yang berbeda.

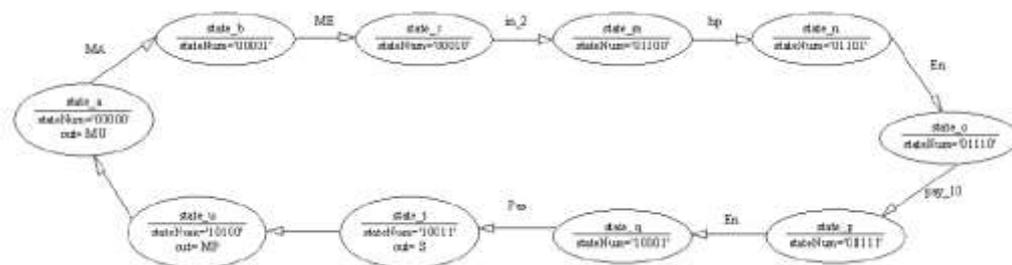


Gambar 7. Skematik RTL sistem mesin penjual pulsa elektrik otomatis

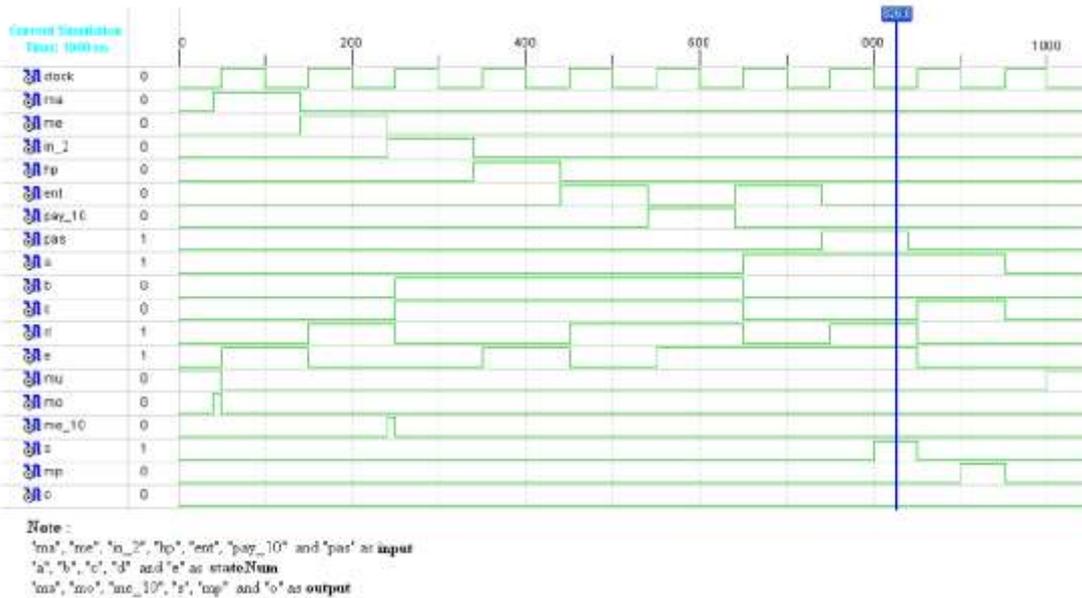
## HASIL DAN PEMBAHASAN

Berdasarkan percobaan sesuai dengan skenario 1 pada gambar 8, didapat hasil simulasi yang ditampilkan pada gambar 9. Pada gambar tersebut diperlihatkan simbol-simbol yang dijadikan sebagai input yaitu 'MA', 'ME', 'in\_2', 'hp', 'ent', 'pay\_10' dan 'pas'. Sedangkan simbol untuk output yaitu 'MU', 'MO', 'ME\_10', 'S' dan 'MP'. Simbol a,b,c,d dan e merupakan simbol untuk menghasilkan nilai dari state yang dilalui(dalam skenario disebut dengan statenum). Pertama diawali dengan statenum '00000' yang mana dalam skenario merupakan state a. Dari state a langsung muncul output MU(menu utama). Jadi walaupun tidak terjadi kegiatan, output tersebut selalu muncul. Output MU muncul pada saat clock bernilai '0'. Maka dalam sistem ini menggunakan sistem aktif low. Maksudnya adalah output yang dihasilkan akan aktif jika nilai clocknya '0' dan akan nonaktif jika clock bernilai '1'. Selanjutnya sistem diberi input dengan nilai 1 pada 'MA', maka dari simulasi akan ditampilkan output 'MO'. Maksudnya adalah jika dalam sistem *vending machine* tombol MA(Menu Awal) ditekan maka akan muncul pada layar *vending machine* berupa Menu Operator(MO) yang berisi jenis-jenis menu operator seluler yang dapat dipilih. Setelah itu akan berlanjut dengan statenum '00001'(dalam skenario merupakan state b). Di state b ini kemudian

diberi input dengan nilai '1' pada 'ME' (pilihan operator Mentari). Kemudian pada simulasi akan berlanjut ke statenum '00010'(dalam skenario yaitu state c). Dari state c diberi input dengan nilai '1' pada symbol 'in\_2' dan menghasilkan output 'ME\_10'(pulsa operator Mentari dengan nominal 10000). Lalu berlanjut ke statenum '01100'(pada skenario merupakan state m). Dari state m di beri input hp dengan nilai '1' kemudian berlanjut ke statenum '01101'(pada skenario merupakan state n). Kemudian pada system diberi input 'En' dengan nilai '1' maka akan berlanjut ke statenum '01110' (dalam skenario yaitu state o). Dari state o berlanjut dengan diberi input 'pay\_10' dengan diberi nilai '1' kemudian berlanjut ke statenum '01111' (sesuai dengan skenario yaitu state p). Dari state p diberi input lagi dengan nilai '1' pada input 'En' dan berlanjut ke statenum '10001'(dalam skenario merupakan state q). Setelah itu diberi input 'pas' dengan nilai '1' dan berlanjut ke statenum '10011'(dalam skenario merupakan state t). Dari state t ini dihasilkan output berupa tanda bukti transaksi/struk(disimbolkan s dalam simulasi). Kemudian berlanjut ke statenum '10100'(state u dalam skenario). Pada state u ini dihasilkan output mengirim pulsa(dalam simulasi disimbolkan dengan MP). Setelah proses selesai, maka kembali lagi ke statenum '00000'(state a) dan muncul output seperti pada awal yang sudah dijelaskan yaitu Menu utama (disimbolkan dengan MU).



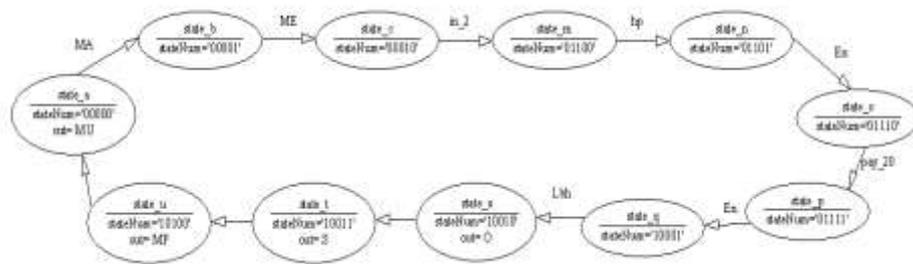
Gambar 8. State diagram skenario 1



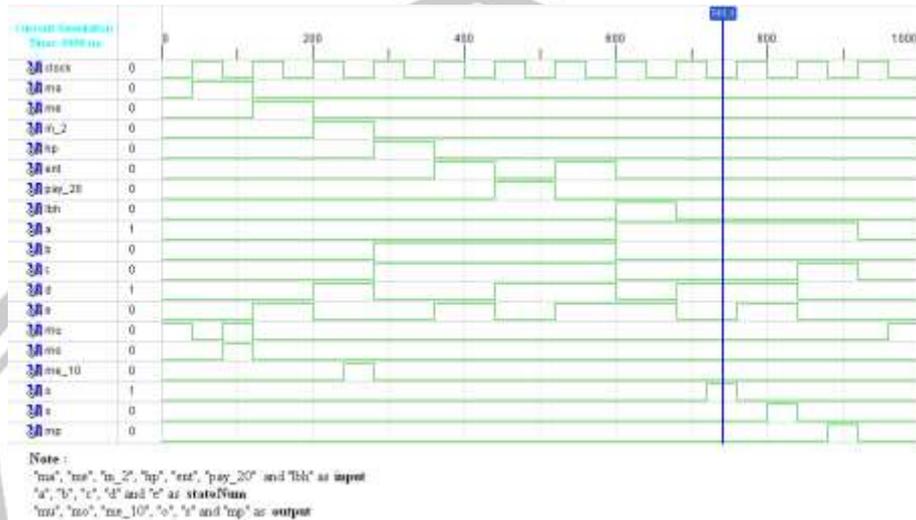
Gambar 9. Hasil Simulasi Skenario 1

Berdasarkan percobaan sesuai dengan state diagram skenario 2 pada gambar 10, didapat hasil simulasi yang ditampilkan pada gambar 11. Pada gambar tersebut diperlihatkan simbol-simbol yang dijadikan sebagai input yaitu 'MA', 'ME', 'in\_2', 'hp', 'ent', 'pay\_20' dan 'lbh'. Sedangkan simbol untuk output yaitu 'MU', 'MO', 'ME\_10', 'O', 'S' dan 'MP'. Simbol a,b,c,d dan e merupakan symbol untuk menghasilkan nilai dari state yang dilalui(dalam scenario disebut dengan statenum). Simulasi yang dihasilkan dari state a dengan statenum '00000' sampai pada statenum '01110'(state o dalam scenario) adalah sama seperti simulasi skenario 1, yang membedakan adalah input yang diberikan. Pada state o tersebut diberi input dengan nilai '1' pada simbol 'pay\_20' kemudian berlanjut ke statenum '01111' (dalam scenario merupakan state p). Setelah itu diberi input 'ent' dengan diberi nilai '1'. Kemudian berlanjut ke statenum '10001'(state q).

lalu diberi input dengan nilai '1' pada simbol 'lbh' dan berlanjut ke statenum '10010'(dalam scenario merupakan state s). Pada state s ini dihasilkan output uang kembalian(dalam simulasi disimbolkan dengan o). Kemudian berlanjut ke statenum '10011' (dalam scenario merupakan state t). Pada state ini langsung mengeluarkan output tanda bukti transaksi (dalam simulasi disimbolkan dengan s). Setelah itu berlanjut ke statenum '10100' (dalam scenario merupakan state u) yang menghasilkan output mengirim pulsa(dalam simulasi disimbolkan dengan MP). Dan akhirnya kembali lagi ke statenum '00000'(state a) dengan output Menu Utama. Setiap output yang muncul terjadi pada saat clock bernilai '0'. Sedangkan pada saat clock bernilai '1' output tidak ada. Hal ini disebabkan system menggunakan aktif low karena pada saat mendesign clock diberi tambahan inverter.



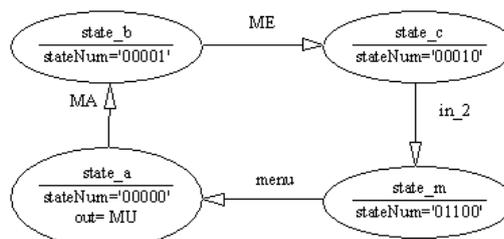
Gambar 10. State Diagram Skenario 2



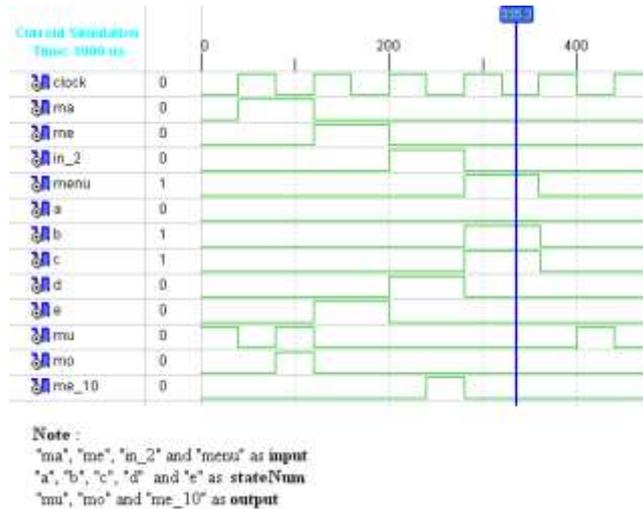
Gambar 11. Hasil Simulasi Skenario 2

Berdasarkan percobaan sesuai dengan state diagram skenario 5 pada Gambar 12, didapat hasil simulasi yang ditampilkan pada Gambar 13. Pada gambar tersebut diperlihatkan simbol-simbol yang dijadikan sebagai input yaitu 'MA', 'ME', 'in\_2' dan 'menu'. Sedangkan simbol untuk output yaitu 'MU', 'MO' dan 'ME\_10'. Simbol a,b,c,d dan e merupakan simbol untuk menghasilkan nilai dari state yang dilalui(dalam skenario disebut dengan statenum). Hasil

simulasi yang dimulai dari statenum '00000'(state a) sampai pada statenum '00010'(state c) sama seperti simulasi skenario 4. Kemudian dari state c diberi nilai '1' pada input 'in\_2' maka dihasilkan output 'ME\_10'. Lalu berlanjut ke statenum '01100' dan diberi input pada 'menu dengan nilai '1'. Karena dipilih menu maka kembali ke statenum '00000'(dalam skenario merupakan state a) dengan output Menu Utama.



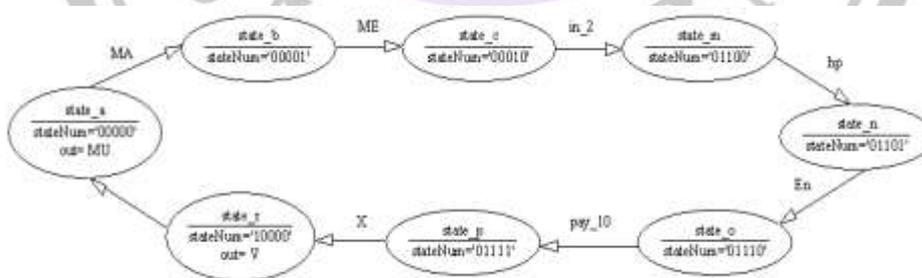
Gambar 12. State Diagram Skenario 5



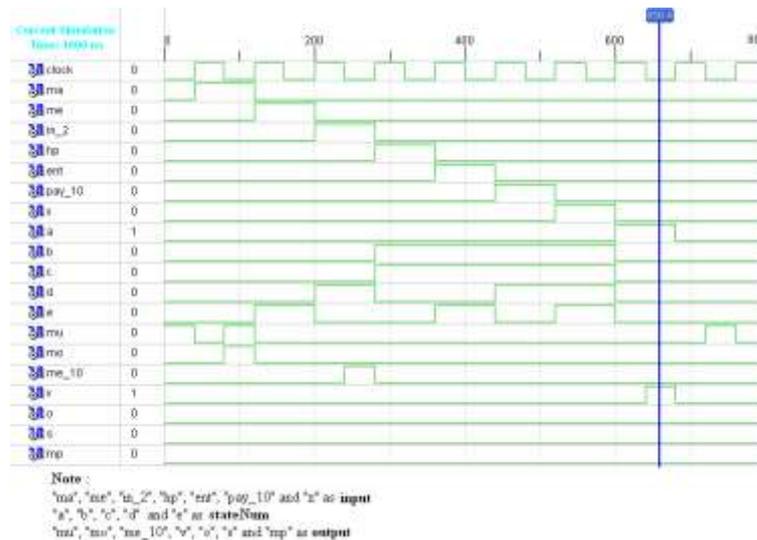
Gambar 13. Hasil simulasi skenario 5

Berdasarkan percobaan sesuai dengan state diagram skenario 9 pada Gambar 14, didapat hasil simulasi yang ditampilkan pada Gambar 15. Pada gambar tersebut diperlihatkan simbol-simbol yang dijadikan sebagai input yaitu 'MA', 'ME', 'in\_2', 'hp', 'ent', 'pay\_10' dan 'x'. Sedangkan simbol untuk output yaitu 'MU', 'MO', 'ME\_10' dan v. Simbol a,b,c,d dan e merupakan simbol untuk menghasilkan nilai dari state yang dilalui (dalam scenario disebut dengan statenum). Pada simulasi yang ditampilkan Gambar 15 juga memiliki kesa-

maan dengan simulasi yang ditampilkan Gambar 9 yaitu mulai dari statenum '00000' (dalam scenario merupakan state a) sampai dengan statenum '01111' (dalam skenario merupakan state p). Selanjutnya pada state ini diberi nilai '1' pada input 'x' maka akan berlanjut ke statenum '10000' dan dihasilkan output berupa keluar uang karena batal transaksi (dalam simulasi disimbolkan dengan v). Dan akhirnya kembali lagi ke awal statenum yaitu '00000'(state a) dan menampilkan output Menu Utama.



Gambar 14. State Diagram Skenario 9



Gambar 15. Hasil Simulasi Skenario 9

## SIMPULAN DAN SARAN

Mesin vending sistem yang dirancang telah berhasil dan disimulasikan dengan Xilinx ISE (Project Navigator) berbasis FPGA. Hasil simulasi menunjukkan bahwa sistem mesin penjual pulsa elektrik otomatis mampu berjalan seperti skenario yang telah dibuat. Dari setiap skenario, hasil simulasi tersebut sesuai dengan output yang ditampilkan dan proses *flow*. Dengan jumlah 4 input LUTs adalah 143, jumlah *slice* yang ditempati adalah 73, jumlah IOBs *bonded* adalah 91 dan jumlah gerbang ekuivalen untuk desain adalah 901.

Sistem kontrol mesin penjual otomatis ini menggunakan FPGA sebagai alternatif mikrokontroler yang memudahkan dalam memodifikasi data dalam modul. Dapat disimpulkan bahwa FPGA lebih baik daripada mikrokontroler untuk merancang sistem mesin penjual otomatis.

Sistem yang dibuat hanya sampai tahap simulasi saja, belum sampai tahap implementasi alat. Oleh sebab itu untuk penelitian selanjutnya dapat dikembangkan ke tingkat *hardware*. Selain itu menu-menu atau intruksi-intruksi yang digunakan dalam sistem penjual otomatis ini masih dapat dikembangkan lagi untuk

menyempurnakan proses kerja penjual otomatis. Hal tersebut dapat dilakukan dengan cara membuat modul baru dan ditambahkan ke dalam sistem ataupun mengubah konfigurasi pada modul yang telah dibuat.

## DAFTAR PUSTAKA

- Andeski, Hendrasto. Juli 2008. *Pengisian pulsa elektronik dengan menggunakan vending machine*.
- Brown, Stephen and Rose, Jonathan. 1996. *Architecture of fpgas and cplds: A tutorial*. Tech, rep. University of Toronto.
- C, Joshua and T, Jeckson. August 2006. *Microcontroller based sms micro e-load vending machine*. De La Salle University. Manila.
- F, Elvien Ricky. Juli 2006. *Designing and constructing a vending machine for offices using microcontroller*. Swiss German University, Bumi Serpong Damai, Island of Java, Indonesia.
- Golson, Steve. September 2000. *State machine design techniques for verilog and vhdl*. High-Level Design.
- Jenkins and Mazumder. 2000. *Finite state machine design a vending machine*. tech. rep., University of Michigan.

- EECS 270: Introduction to Logic Design.
- Monga, A., Singh, B. (2012). Finite State Machine based Vending Machine Controller with Auto-Billing Features. *International Journal of VLSI design & Communication Systems (VLSICS)*. Vol.3, No.2, Pp 19-28
- Pedroni, Volnei A. 2004. *Circuit Design with VHDL*. MIT Press. London,England.
- Reichardt, J. *Synchronous finite state machines; design and behavior*. tech. rep., university of applied sciences hamburg. Digital Systems.
- Rizkiningayu, Wini and Putra,Agfianto Eko. 2007.*Vending Machine Control System Prototype Using Altera FPGA EPF 10K10*. Gadjah Mada University. Bulaksumur, Yogyakarta.
- Roth, Charles H and John, Lizy Kurian. 2008. *Digital Systems Design Using VHDL*. Australia: Thomson Learning, second edition.
- Rodriguez, Ridwan. November 2011. *Output Forming Logic*.
- Sahin, Suhap. Becerikli, Yasar and Yazici, Suleyman. *Neural network implementation in hardware using fpgas*.
- Sajid, Abdul Hafeez. 2008. *A Seminar Report on FPGA Based Design and Development of Distributed Arithmetic Control System*. Shivaji University. Kolhapur.
- Siswono, Hartono. 2005. *Perancangan sistem digital menggunakan simulator xilinx*. Gunadarma University. Depok.
- Ulfah, Silmina. Januari 2011. *Designing backscattering control logic based on fpga using vhdl*. Gunadarma University.Jakarta.
- Xilinx. November 2011.*Fpga architecture of spartan 3E*.
- Yalamanchili, Sudhakar . 2001. *Introductory VHDL From Simulation to Synthesis*. Xilinx Design, Prentice-Hall.