



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

DIGITAL LOCK SYSTEM

GROUP A2

Fatima Khairunnisa	2106651515
Albertus Timothy Gunawan	2106639472
Eldisja Hadasa	2106640133
Muhammad Fajri Alqomaril	2106651635

KATA PENGANTAR

Puji syukur kami panjatkan kepada Tuhan Yang Maha Esa karena karunia dan rahmat-Nya sehingga kami, kelompok A2 dapat menyelesaikan proyek akhir praktikum Perancangan Sistem Digital dengan judul “*Digital Lock System*” tepat pada waktunya. Kami kelompok A2 berusaha semaksimal mungkin agar laporan yang kami buat dapat bermanfaat dalam bidang keamanan dan pertahanan.

Adapun tujuan dari penulisan laporan ini, yakni untuk memenuhi tugas akhir praktikum Perancangan Sistem Digital (PSD). Kami ingin mengucapkan terima kasih kepada Bapak Dr. Ruki Harwahu, ST. MT. MSc. dan Bapak Yan Maraden, ST. MT yang telah membimbing mata kuliah Perancangan Sistem Digital, serta para asisten laboratorium digital yang telah membimbing kami selama praktikum hingga proyek akhir ini. Kami juga mengucapkan terima kasih kepada Kak Naufal Faza yang telah membantu kami dalam mengerjakan proyek akhir ini.

Kami menyadari laporan yang kami susun ini masih jauh dari kata sempurna. Oleh karena itu, kritik serta saran kami butuhkan untuk menyempurnakan laporan ini.

Depok, 10 Desember, 2022

Group A2

DAFTAR ISI

BAB 1: PENDAHULUAN

- 1.1 LATAR BELAKANG
- 1.2 DESKRIPSI PROYEK
- 1.3 TUJUAN
- 1.4 TUGAS DAN TANGGUNG JAWAB

BAB 2: IMPLEMENTASI

- 2.1 PERALATAN
- 2.2 IMPLEMENTASI

CHAPTER 3: PENGUJIAN DAN ANALISIS

- 3.1 PENGUJIAN
- 3.2 HASIL
- 3.3 ANALISIS

BAB 4: KESIMPULAN

REFERENSI

LAMPIRAN

- Lampiran A: Project Schematic
- Lampiran B: Dokumentasi

BAB 1

PENDAHULUAN

1.1 LATAR BELAKANG

Seiring dengan perkembangan zaman, banyak inovasi yang berkembang pada bidang teknologi. Dimana, perkembangan ini memudahkan masyarakat bahkan menggantikan pekerjaan manusia saat ini. Salah satunya adalah teknologi sistem digital yang digunakan untuk pemrosesan sinyal atau data sehingga dapat menghasilkan output yang merupakan bukti dari perkembangan teknologi yang sangat pesat.

Pada sistem digital yang kami rancang, kami memanfaatkan mikrokontroler merupakan salah satu contoh dari inovasi pada bidang sistem digital. Dalam makalah ini, terdapat penjelasan tentang perancangan sistem digital dengan studi kasusnya yaitu perancangan *Digital Lock System* yang menerapkan *finite state machine* pada metodenya.

Finite State Machine adalah metode yang berguna untuk memudahkan dalam membuat VHDL melalui pengkonversian *state diagram*-nya ke bahasa VHDL. FSM digunakan juga untuk membuat *digital sequential circuit*. Pada rangkaian digital sekuensial, outputnya bergantung dengan *present state* dan juga bergantung dengan kondisi pada input yang sebelumnya. Dalam hal ini, rangkaian *Digital Lock System* memerlukan memori atau register yang berguna untuk penyimpanan informasi biner.

Berangkat dari hal tersebut, *Digital Lock System* tentunya akan menjadi salah satu solusi yang akan memudahkan manusia dan tentunya akan meningkatkan rasa kepercayaan banyak dalam hal keamanan.

1.2 DESKRIPSI PROYEK

Digital Lock System merupakan locking system yang dapat menerima 6 input angka password serta dapat melakukan enkripsi input password. Dimana, prinsip kerjanya adalah password yang telah terenkripsi akan disimpan ke dalam memory. Digital system lock ini memiliki dua buah metode pemasukkan password. Pada metode pertama, memasukkan seluruh input password secara berurutan, sedangkan cara kedua adalah dengan memasukkan beberapa input password sesuai dengan urutan digit permintaan device. Dimana, pada *Digital*

Lock System ini, input password akan dilakukan proses enkripsi dengan metode *simple hashing* yang kami buat dan akan disimpan dalam *register*. Jika password yang diinput sudah benar, maka seven segment akan menghasilkan output "O", kebalikannya jika inputnya salah maka outputnya akan menjadi "X".

1.3 TUJUAN

Tujuan dari proyek ini adalah sebagai berikut:

1. Dapat membuat *Digital Lock System* yang baik dan benar
2. Dapat menerapkan *finite state machine* dalam perancangan sistem digital
3. Dapat memanfaatkan FSM dalam bahasa pemrograman VHDL

1.4 TUGAS DAN TANGGUNG JAWAB

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

Peran	Tanggung Jawab	Person
Membuat laporan	Membuat isi laporan	Fatima Khairunnisa, Albertus Timothy Gunawan, Eldisja Hadasa
Membuat Presentasi	Membuat PPT	Fatima Khairunnisa, Muhammad Fajri Alqomaril
State diagram	Membuat state diagram sesuai rangkaian yang dibuat	Fatima Khairunnisa
Membuat code state	Membuat kode VHDL state pada proyek	Eldisja Hadasa, Albertus Timothy Gunawan
Coder test bench VHDL	Membuat kode VHDL test bench pada proyek	Fatima Khairunnisa, Albertus Timothy Gunawan

Pembuatan hasil simulasi	Menjalankan program dan mendokumentasikan hasilnya	Fatima Khairunnisa, Muhammad Fajri Alqomaril, Albertus Timothy Gunawan
Coder main source VHDL	Membuat kode VHDL utama pada proyek	Albertus Timothy Gunawan
Membuat read.me	Membuat isi read.me dengan Markdown	Eldisja Hadasa, Muhammad Fajri Alqomaril
Membuat Project Schematic	Membuat block diagram	Eldisja Hadasa

Tabel 1. Tugas dan Tanggung Jawab

BAB 2

IMPLEMENTASI

2.1 PERALATAN

Pada proyek ini, kami memanfaatkan beberapa perlengkapan yaitu:

- Github
- VScode
- Modelsim
- Quartus Prime
- 7 segment display
- Basic logic gate
- LED
- Register
- Counter
- Switch

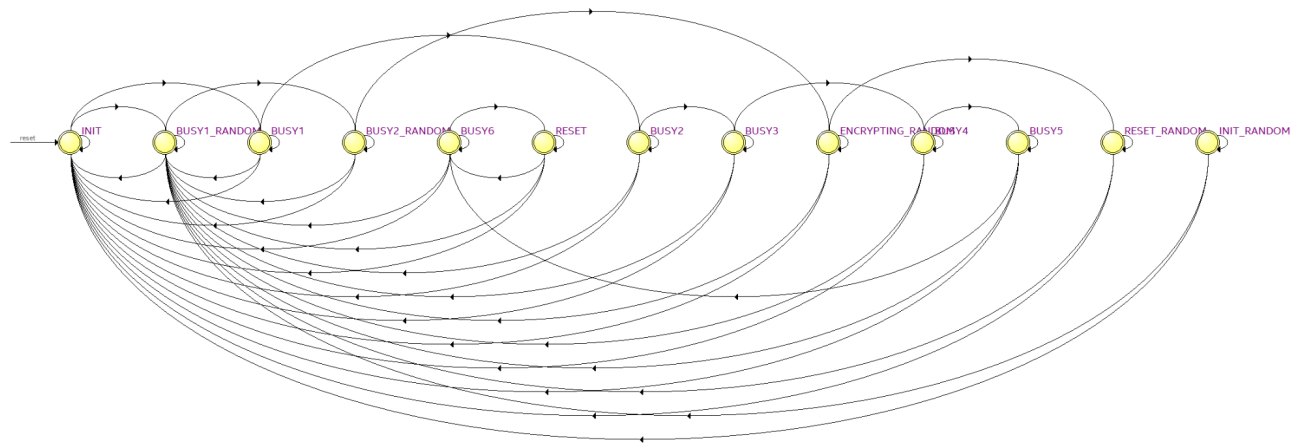
2.2 IMPLEMENTASI

Pada rangkaian *Digital Lock System* yang kami buat terdapat `main.vhd` yang merupakan *top level* dari rangkaian ini. Selain itu, kami membuat `main_function.vhd` sebagai *package* yang menyimpan fungsi-fungsi yang akan dipakai di `main`. Dalam `main.vhd`, menerima input `RESET_BTN`, `NEXT_BTN`, `RESET2_BTN`, `CLOCK`, dan `SWITCH`. `RESET_BTN` dan `RESET2_BTN` digunakan untuk mereset register input password sehingga bernilai 0. `NEXT_BTN` digunakan agar state dapat mengubah `DEBOUNCE_INIT` menjadi `SHIFT_STATE`. Dalam `SHIFT_STATE` ini, program akan menerima input dari `SWITCH` setelah itu input `SWITCH` akan dimasukkan kedalam register input password.

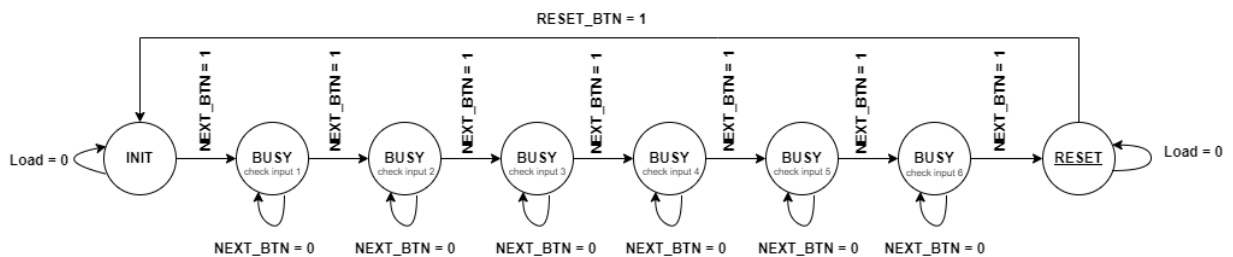
Setelah seluruh input password telah diisi atau user sudah mengisi 6 input password, akan dilakukan proses enkripsi password menggunakan function `simple_hash` atau `random_simple_hash`. Kemudian, password yang sudah di enkripsi tersebut akan dimasukkan kedalam register. Selanjutnya, nilai dari register tersebut akan dibandingkan dengan answer key. Jika hasil perbandingan password dengan register bernilai *true*, maka seven segment paling kanan akan menampilkan “O”. Sebaliknya, jika perbandingan password dengan register bernilai *false*, maka tampilan seven segment nya akan menampilkan “X”.

Dalam `main_function.vhd`, kami membuat 4 buah function yang terdiri dari `bcd_to_7segment`, `integer_to_4bits`, `simple_hash`, dan `random_simple_hash`. Pada `bcd_to_7segment`, fungsi ini berfungsi untuk mengubah `STD_LOGIC_VECTOR` yang berukuran 4 bit ke input yang dapat diterima seven segment. Sedangkan, `integer_to_4bits` berfungsi untuk mengubah nilai integer menjadi `STD_LOGIC_VECTOR` berukuran 4 bit. Kemudian, `simple_hash` merupakan proses enkripsi dari input password metode pertama yang berukuran 24 bit dan akan mereturn `STD_LOGIC_VECTOR` berukuran 32 bit. Sedangkan pada `simple_hash` dari input password metode kedua yang berukuran 8 bit dan akan mereturn `STD_LOGIC_VECTOR` berukuran 12 bit.

Selain itu, dalam pembuatan diagram state pada proyek ini akan dibuat dengan mengimplementasikan *mealy state diagram*. Hal ini dikarenakan proyek yang dibuat memakai FSM dan next state nya dipengaruhi oleh present state dan juga inputnya. Dimana, inputnya merupakan nilai dari switch dan diagram state yang dibuat adalah berupa rangkaian yang diilustrasikan sebagai berikut:

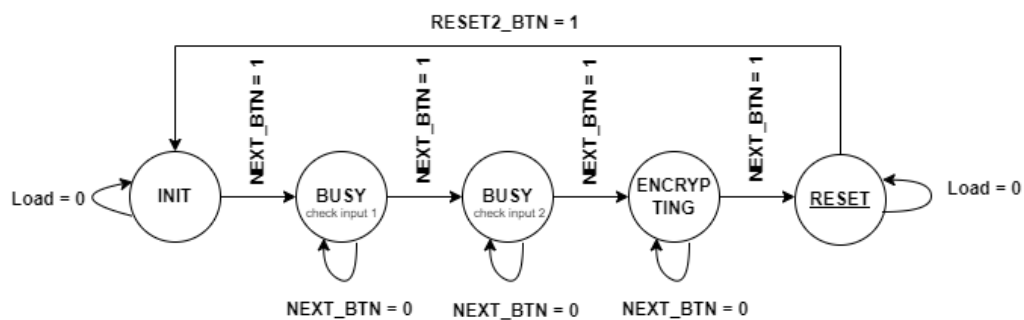


Gambar 1. Finite State Machine after Synthesis (*Quartus*)



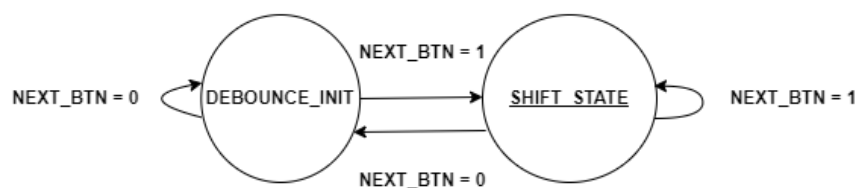
Gambar 2. State Diagram untuk metode 1

Pada diagram metode pertama, rangkaian ini akan diimplementasikan dengan 8 state yang berbeda, yakni INIT, BUSY1, BUSY2, BUSY3, BUSY4, BUSY5, BUSY6, RESET. Dengan mekanisme, rangkaian akan di-reset ketika tombol RESET_BTN ditekan. Pada saat user memasukkan input, state akan berpindah secara otomatis selama NEXT_BTN bernilai 1 menuju state reset. Dalam reset ini, akan dilakukan proses pengecekan password.



Gambar 3. State Diagram untuk metode 2

Sedangkan, pada diagram metode kedua, rangkaian ini akan diimplementasikan dengan 5 state yang berbeda, yakni INIT_RANDOM, BUSY1_RANDOM, BUSY2_RANDOM, ENCRYPTING_RANDOM, RESET_RANDOM. Dengan mekanisme, rangkaian akan di-reset ketika tombol RESET_BTN ditekan. Pada saat user memasukkan input, state akan berpindah secara otomatis selama NEXT_BTN bernilai 1 menuju state hash. Dalam state hash akan dilakukan peng-enkripsian dan setelah enkripsi selesai akan berpindah ke state reset. Dalam reset ini, akan dilakukan proses pengecekan password.



Gambar 4. State Diagram untuk perpindahan state pada kedua method

Yang terakhir, pada diagram ketiga ini menunjukkan pada saat perpindahan state dalam kedua method, rangkaian ini akan diimplementasikan dengan 2 state yang berbeda, yakni DEBOUNCE_INIT dan SHIFT_STATE. Dalam state DEBOUNCE_INIT, state pada state diagram pada metode 1 dan 2 tidak akan berpindah. sedangkan pada SHIFT_STATE, state pada diagram 1 dan 2 akan berpindah.

Dalam interface-nya, rangkaian ini menggunakan seven-segment display, LED, dan button. Pada saat user memasukkan input, urutan dari password yang diinput akan ditampilkan pada 7-segment display. Sebelumnya, input yang dimasukkan user akan dikonversi ke dalam bilangan biner menggunakan basic logic gate untuk metode 1. sementara, untuk metode 2, rangkaian akan meminta input digit tertentu dari user sebanyak 2 angka. Password tersebut akan dienkripsi kemudian akan dilakukan proses pengecekan.

BAB 3

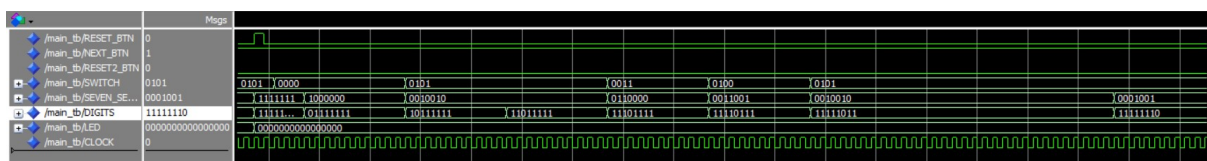
PENGUJIAN DAN ANALISIS

3.1 PENGUJIAN

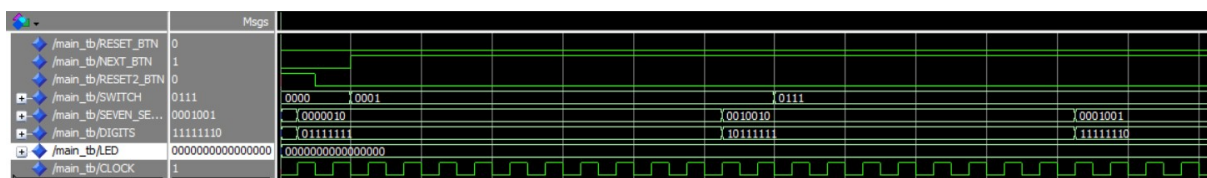
Dalam pengujian, kelompok kami memanfaatkan *test bench* pada VHDL guna mensimulasikan situasi dan kondisi yang akan terjadi pada saat user menggunakan sistemnya. Dimana, pengujian ini kami lakukan pada ModelSim. Pada pengujian, kami melakukannya dengan 4 kondisi input yang berbeda:

1. Pengujian dengan memasukkan 6 input password yang benar untuk metode pertama dengan menggunakan input test bench
2. Pengujian dengan memasukkan 6 input password yang salah untuk metode pertama dengan menggunakan input test bench
3. Pengujian dengan memasukkan password sesuai dengan urutan digit yang diminta oleh device dengan input manual
4. Pengujian dengan memasukkan password yang tidak sesuai dengan urutan digit yang diminta oleh device dengan menggunakan input test bench

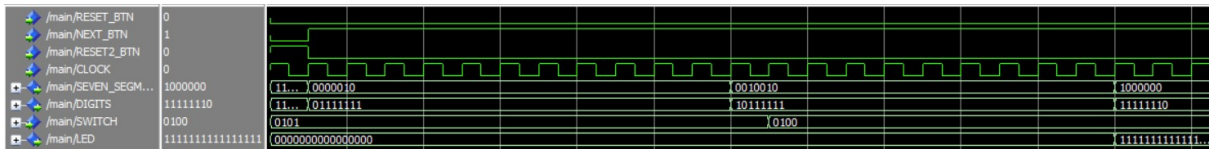
3.2 HASIL



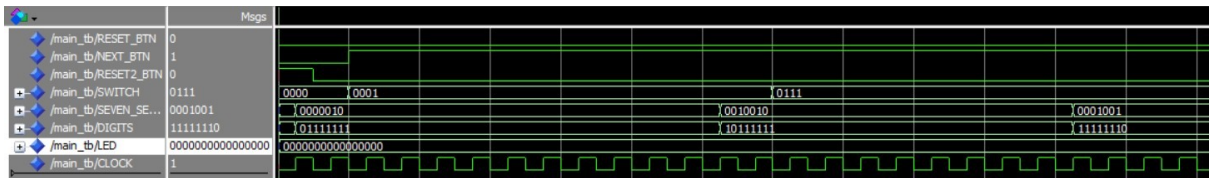
Gambar 5. Pengujian dengan memasukkan 6 input password yang benar untuk metode pertama dengan menggunakan input test bench



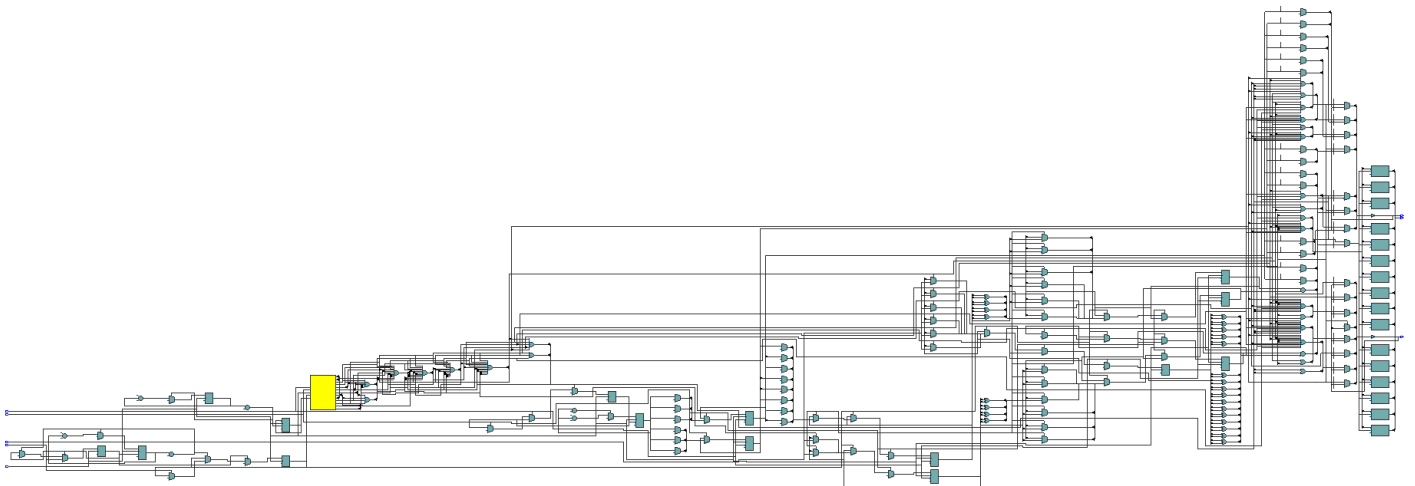
Gambar 6. Pengujian dengan memasukkan 6 input password yang salah untuk metode pertama dengan menggunakan input test bench



Gambar 7. Pengujian dengan memasukkan password sesuai dengan urutan digit yang diminta oleh device dengan menggunakan input manual



Gambar 8. Pengujian dengan memasukkan password tidak sesuai dengan urutan digit yang diminta oleh device dengan menggunakan input test bench



Gambar 8. RTL viewer main.vhd

3.3 ANALISIS

Seperti pada Gambar 2, saat tombol RESET ditekan ($\text{RESET_BTN} = 1$) rangkaian akan direset - kembali ke keadaan semula - pada *state* INIT (inisialisasi rangkaian). Saat tombol RESET2 di tekan ($\text{RESET2_BTN} = 1$) password dalam rangkaian akan direset - dengan memasukkan angka *random* ke dalam *register* - pada *state* BUSY1_RANDOM. Karena tombol termasuk saklar manual sehingga sinyal yang dihasilkan tidak stabil, digunakan algoritma *debounce* sebagai penstabil sinyal tombol. Pada metode 1, Saat tombol LOAD ditekan ($\text{debounce_signal} = 1$) *input* dari *user* di masukkan ke dalam *memory* dengan *shift left* dan setiap kali melakukan *shift*, maka *state* berbeda juga, misalnya *shift input* password 1 *state* yang berjalan BUSY1, *shift input* 2 *state* yang berjalan BUSY2, dan seterusnya.

Pada metode 2, ketika RESET1_BTN ditekan maka rangkaian akan memasuki *state* BUSY1_RANDOM dimana rangkaian meminta *input* yang sesuai dengan *password* dengan digit sesuai keinginan device (digit *random*) pada *range* 1 sampai 6, sebanyak dua angka atau dua digit. Setelah *user* memasukkan *input password* yang tepat, *state* akan berubah ke BUSY_RANDOM2, yakni keadaan saat *input password* digit ke-2 dari *user* disimpan ke *memory*. Setelah itu, *state* akan berubah menjadi RESET_RANDOM yang menandakan rangkaian sedang melakukan *generate* angka acak. Rangkaian akan menampilkan output 7-segment display yang berkedip (*flickering*) saat *state* BUSY6 yang kemudian mematikan *flick* tersebut pada *state* RESET dan kembali lagi ke BUSY 6.

Bagian *output* akan diproses sesuai *state* yang sedang berjalan. Pada bagian *state* INIT *input* belum dimasukkan oleh *user*, maka 7-segment display akan menampilkan angka 0. Pada *state* BUSY1, *input* password digit ke-1 dari *user* diproses agar dapat ditampilkan pada 7-segment display. Pada *state* BUSY2, *input* password digit ke-2 dari *user* diproses agar dapat ditampilkan pada 7-segment display, dan seterusnya sampai BUSY6. Setiap digit dikonversi berdasarkan *state* BUSY kesekian yang sedang berjalan dengan menggunakan *counter* (*display_counter*) sebagai sinyal *select* sampai angka tertentu. Setelah *user* berhasil memasukkan semua password yang sesuai rangkaian akan di-reset (kembali ke sebelum dimasukkan password) dan LED akan menyala ($\text{LED} = 1111111111111111$) begitu juga dengan 7-segment display sesuai dengan *select* yang ada.

Pada metode 1, hasil *test bench* untuk kasus *user* memasukkan *password* yang benar tercantum pada Gambar 3. Saat *user* memasukkan digit ke-2 SWITCH (*input user*) bernilai

0111 dan DIGITS bernilai 10111111 yang menunjukkan *state* BUSY2 yang sedang berjalan - *input* digit ke-2 sedang diproses - yang dapat dilihat pada posisi angka 0 dalam DIGITS, begitu juga dengan *state* berikutnya. Saat user selesai memasukkan semua digit password (sampai BUSY6), LED akan bernilai 1111111111111111 (LED menyala) yang menandakan *lock system* terbuka serta rangkaian berada pada *state* RESET yang menandakan *input password* dari *user* akan dihapus (rangkaiannya kembali ke *state* INIT). Pada Gambar 4, masih pada metode 1 namun dengan *input* yang salah, dapat dilihat bahwa rangkaian tidak berubah dari *state* BUSY6 ke *state* selanjutnya, yakni RESET. Pada gambar terlihat bahwa lampu LED tetap bernilai 0000000000000000 (LED mati).

Pada pengujian Metode 2, untuk *input* password yang benar tidak bisa menggunakan *test bench* karena harus menyesuaikan dengan keinginan *device*, dimana *device* meminta digit *random* sehingga pengujian dilakukan secara manual. Dapat dilihat pada Gambar 5, pada pengujian diminta digit ke-1 dan ke-2 dari password. Ketika user berhasil menginput digit password ke-1, rangkaian akan berubah *state* dari BUSY1_RANDOM ke BUSY2_RANDOM. Begitu juga, pada BUSY2_RANDOM saat user berhasil memasukkan password sesuai yang diminta, *state* akan berubah ke RESET yang ditandai dengan menyala LED. Pada Gambar 6, pengujian metode 2 dilakukan untuk *input* yang salah dengan *testbench*. Dapat dilihat saat *input* user (SWITCH) tidak sesuai dengan yang diminta rangkaian, rangkaian tidak akan dapat mencapai *state* RESET, yakni saat *lock* bisa terbuka dan LED menyala (LED = 1111111111111111).

BAB 4

KESIMPULAN

Pada proyek akhir dalam mata kuliah Perancangan Sistem Digital ini kami merancang *Digital Lock System* sebagai perangkat yang dapat membantu manusia dalam menjamin keamanan. Dalam proyek ini, kami juga menerapkan pemrograman VHDL pada rangkaian yang dibuat agar rangkaian tersebut dapat dilakukan uji coba pada ModelSim dan Quartus. Saat pengujian dilakukan, rangkaian yang sudah dirancang sebaik mungkin dapat menjalankan prosesnya dengan baik.

Pada rangkaian, kami membuatnya dengan rangkaian yang sekuensial. Dimana, rangkaian akan berjalan secara berurut, kami juga melakukan penerapan *behaviorial style* yang memanfaatkan statement process.

Rangkaian ini juga dilengkapi dengan *test bench* yang berfungsi untuk memudahkan pada saat pengujian dalam menelusuri kemungkinan permasalahan pada rangkaian. Selain itu, pada pembuatan kode, kami mengimplementasikan prinsip modularitas karena komponen yang digunakan tidak sedikit. Tidak lupa juga, rangkaian yang kami buat adalah rangkaian *mealy state machine*, karena next state nya dipengaruhi oleh present state dan juga inputnya. Oleh karena itu, input yang dimasukkan oleh user pada rangkaian akan mempengaruhi output selanjutnya.

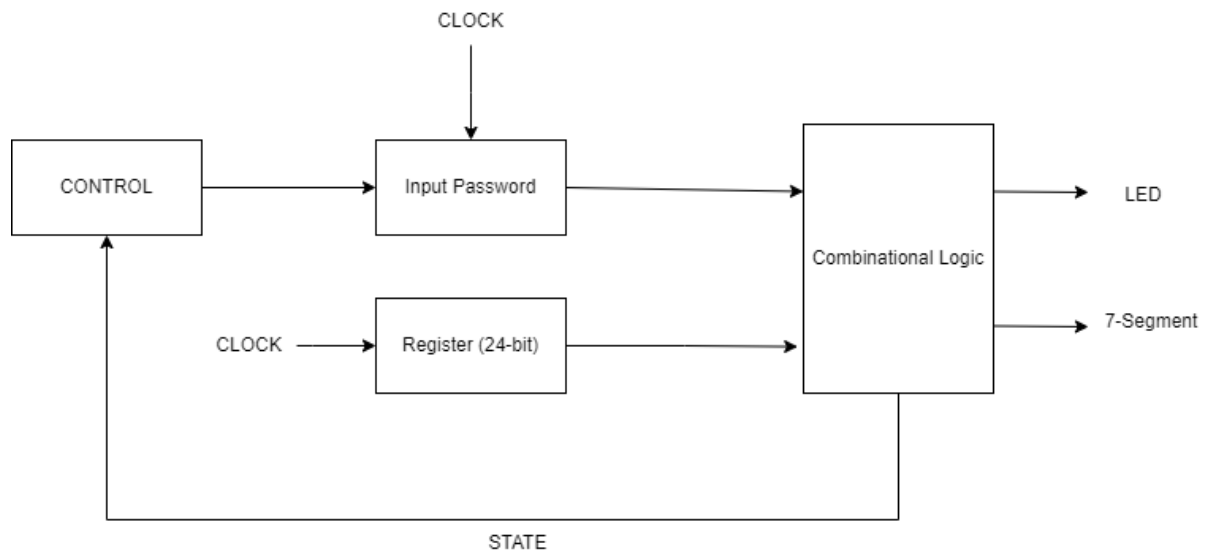
REFERENSI

- [1] FPGA Tutorial. (2020). Writing Reusable VHDL Code using Generics and Generate Statements. [Online]. Available: [Writing Reusable VHDL Code using Generics and Generate Statements \(fpgatutorial.com\)](https://fpgatutorial.com/writing-reusable-vhdl-code-using-generics-and-generate-statements) [Dec 1, 2022].
- [2] VHDLGuide. (n.d). 6.Procedures, functions and packages [Online]. Available: [6. Procedures, functions and packages — FPGA designs with VHDL documentation \(vhdlguide.readthedocs.io\)](https://vhdlguide.readthedocs.io/en/latest/procedures-functions-packages/) [Dec 1, 2022].
- [3] T. R. Kuphaldt, “Simple Combination Lock,” *Allaboutcircuits.com*, Feb. 18, 2015. [Online]. Available: <https://www.allaboutcircuits.com/textbook/experiments/chpt-7/simple-combination-lock/>. [Accessed: Dec. 2, 2022]
- [4] Mealy, B. & Tappero, F., 2018. Free Range VHDL
- [5] Digital Laboratory Universitas Indonesia. (2021). Perancangan Sistem Digital dan Praktikum Modul 2: VHDL Introduction. [Online]. Available: [Modul2_PSD.pdf \(ui.ac.id\)](https://ui.ac.id/modul2-psd/) [Dev 1, 2022].
- [6] Digital Laboratory Universitas Indonesia. (2021). Perancangan Sistem Digital dan Praktikum Modul 3: Concurrent Circuit Design. [Online]. Available: [Modul3_PSD.pdf \(ui.ac.id\)](https://ui.ac.id/modul3-psd/)[Dev 1, 2022].
- [7] Digital Laboratory Universitas Indonesia. (2021). Perancangan Sistem Digital dan Praktikum Modul 4: Sequential Circuit Design. [Online]. Available: [Modul4_PSD.pdf \(ui.ac.id\)](https://ui.ac.id/modul4-psd/)[Dev 1, 2022].
- [8] Digital Laboratory Universitas Indonesia. (2021). Perancangan Sistem Digital dan Praktikum Modul 5: Structural Circuit Design. [Online]. Available: [Modul5_PSD.pdf \(ui.ac.id\)](https://ui.ac.id/modul5-psd/)[Dev 1, 2022].
- [9] Digital Laboratory Universitas Indonesia. (2021). Perancangan Sistem Digital dan Praktikum Modul 6: Test Bench. [Online]. Available: [Modul6_PSD.pdf \(ui.ac.id\)](https://ui.ac.id/modul6-psd/)[Dev 1, 2022].
- [10] Digital Laboratory Universitas Indonesia. (2021). Perancangan Sistem Digital dan Praktikum Modul 7: Implementasi VHDL. [Online]. Available: [Modul7_PSD.pdf \(ui.ac.id\)](https://ui.ac.id/modul7-psd/)[Dev 1, 2022].
- [11] Digital Laboratory Universitas Indonesia. (2021). Perancangan Sistem Digital dan Praktikum Modul 8: Looping Constructs. [Online]. Available: [Modul8_PSD.pdf \(ui.ac.id\)](https://ui.ac.id/modul8-psd/) [Dev 1, 2022].

- [12] Digital Laboratory Universitas Indonesia. (2021). Perancangan Sistem Digital dan Praktikum Modul 9: Finite State Machine. [Online]. Available: [Modul9_PSD.pdf \(ui.ac.id\)](#)[Dev 1, 2022].

LAMPIRAN

Lampiran A: Project Schematic



Lampiran B: Dokumentasi

Put the documentation (photos) during the making of the project

