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

PASSCODE

GROUP AP03

DANIEL NIKO MARDJAJA	**2206026183**
RIFQI RAMADHAN	**2206062964**
FAIRUZ MUHAMMAD	**2206814324**
NICHOLAS SAMOSIR	**2206059396**

PREFACE

Segala puji dan syukur kami panjatkan ke hadirat Tuhan Yang Maha Esa, karena atas berkat dan rahmat-Nya, kami dapat menyelesaikan proyek Perancangan Sistem Digital 2023. Karena dengan bantuan-Nya yang tidak terhingga ini memungkinkan kami untuk menuntaskan proyek inovatif ini dengan sukses.

Di era yang serba digital ini, keamanan informasi menjadi sangat krusial. Berbagai teknologi terkini telah mempermudah pekerjaan kita dan meningkatkan efisiensi. Oleh karena itu, kami terinspirasi untuk mengembangkan sebuah sistem passcode yang memberikan lapisan keamanan tambahan dalam menjaga data dan informasi penting. Kami berharap bahwa dengan terbitnya proyek ini, akan memberikan kontribusi signifikan dalam meningkatkan keamanan digital dan membantu banyak orang dalam melindungi data pribadi mereka.

Kami mengucapkan terima kasih yang sebesar-besarnya kepada semua pihak yang telah berkontribusi dalam pembuatan proyek ini. Mulai dari orang tua kami yang selalu mendukung, hingga kepada Asisten Laboratorium kami, Aldrian Raffi Wicaksono, yang telah memberikan bimbingan dan arahan yang berharga sehingga proyek ini dapat terlaksana dengan lancar. Tak lupa, ucapan terima kasih juga kami sampaikan kepada kelompok AP03, yang telah berkolaborasi dengan penuh semangat dalam menyusun laporan dan mengembangkan proyek ini.

Kami berharap, melalui proyek ini, pembaca dapat memperoleh wawasan baru dan menemukan solusi keamanan digital yang efektif. Kami juga berharap agar kelompok AP03 akan terus mengembangkan kemampuan mereka dalam bidang sistem digital, serta memahami aplikasi praktis dari algoritma enkripsi dan dekripsi. Akhir kata, kami memohon maaf apabila terdapat kekurangan dalam penulisan laporan ini. Terima Kasih

Depok, December 23, 2023

Group AP03

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

Dengan semakin berkembangnya kemajuan teknologi dan semakin meningkatnya ketergantungan kegiatan sehari-hari kita terhadap aplikasi dan website-website tertentu, keamanan informasi-informasi penting yang kita miliki juga menjadi semakin rentan. Salah satu metode yang digunakan untuk meningkatkan keamanan dan mengurangi kerentanan adalah dengan menggunakan sebuah *password* yang bertugas sebagai kunci yang bertugas untuk menjaga informasi agar tidak dapat diakses oleh pihak-pihak yang tidak kita inginkan. Namun, sudah terjadi terlalu banyak kasus dimana *password* berhasil tertembus. Hal tersebut terjadi dikarenakan banyak hal seperti *password* yang kita miliki terlalu simpel, data yang menyimpan *password* secara lokal berhasil dibobol, dan lain-lainnya.

Untuk menanggulangi problema tersebut, solusi yang kami terapkan adalah dengan cara menambahkan lapisan pembatas antara data-data penting dan pihak yang tidak kita inginkan. Untuk melakukan hal tersebut, kami memutuskan untuk membuat sebuah program yang *passcode* yang dapat bekerja sebagai *password* database yang akan melakukan proses enkripsi dan dekripsi untuk menambahkan lapisan penjaga untuk meningkatkan keamanan.

1.2 PROJECT DESCRIPTION

Passcode ini merupakan sistem keamanan digital yang dirancang untuk melindungi akses ke informasi atau ruang tertentu. Sistem ini menggunakan kombinasi tombol untuk mengatur dan mereset password, dimana password tersebut disimpan dalam file dengan proteksi algoritma enkripsi dan dekripsi. Setiap kali pengguna menekan tombol ‘Enter’, sistem akan membandingkan input dengan password yang telah disimpan. Jika cocok, akses akan diberikan, sedangkan jika tidak, akan muncul pesan kesalahan.

Fitur unggulan dari sistem ini adalah kemampuannya dalam menyediakan proteksi enkripsi yang kuat, sehingga menjaga keamanan password dari resiko peretasan atau akses tidak sah. Hal ini sangat penting dalam melindungi informasi sensitif di berbagai lingkungan, seperti perusahaan, lembaga pemerintahan, atau bahkan untuk penggunaan pribadi di rumah.

Sistem ini juga dirancang dengan interface yang mudah digunakan, memungkinkan pengguna untuk dengan mudah mengatur dan mereset password mereka. Mode reset password ini sangat berguna dalam situasi di mana pengguna lupa password mereka atau ketika ada kebutuhan untuk mengganti password secara berkala sebagai langkah keamanan tambahan.

Selain itu, sistem ini dirancang untuk memungkinkan kontrol manual penuh oleh pengguna, tanpa ada fitur operasi otomatis. Hal ini memberikan fleksibilitas bagi pengguna untuk mengontrol sistem sesuai dengan kebutuhan spesifik mereka, meningkatkan keamanan dan personalisasi pengalaman pengguna.

Secara keseluruhan, Passcode ini menawarkan solusi keamanan yang tangguh dan mudah digunakan, menjadikannya pilihan ideal untuk siapa saja yang membutuhkan sistem keamanan yang dapat diandalkan untuk melindungi data dan ruang pribadi mereka.

1.3 OBJECTIVES

Tujuan dari proyek ini adalah sebagai berikut,

1. Mengimplementasikan VHDL dalam Kehidupan Sehari - Hari.
2. Mengintegrasikan Teknologi Enkripsi dalam Keamanan Digital.
3. Pengembangan sistem keamanan yang dapat diandalkan dan Mudah Digunakan.
4. Memberikan Solusi Keamanan yang Fleksibel dan Personalisasi.
5. Meningkatkan Kesadaran dan Penggunaan Keamanan Digital yang Efektif.

1.4 ROLES AND RESPONSIBILITIES

Peran dan tanggung jawab yang ditugaskan untuk setiap anggota kelompok adalah sebagai berikut:

Roles	Responsibilities	Person
Desain Skematik dan Programming di VHDL	Merancang skematik rangkaian digital yang dibutuhkan untuk proyek.	Nicholas Samosir Daniel Niko Mardjaja

	Melakukan simulasi dan verifikasi desain untuk memastikan semua fungsi bekerja sesuai dengan spesifikasi.	
Laporan Proyek	Mengumpulkan data dan hasil pengujian dari proyek. Menulis laporan detail mengenai proses desain, dan hasil yang dicapai.	Rifqi Ramadhan Fairuz Muhammad Daniel Niko Mardjaja
Pembuatan PPT Presentasi	Mendesain slide presentasi yang menarik untuk mendukung penyampaian informasi tentang proyek. Menyusun konten presentasi yang mencakup latar belakang proyek, proses desain, pengembangan, dan hasil.	Daniel Niko Mardjaja Fairuz Muhammad Rifqi Ramadhan

Table 1. Roles and Responsibilities

CHAPTER 2

IMPLEMENTATION

2.1 EQUIPMENT

Perlengkapan yang digunakan dalam membuat proyek ini adalah sebagai berikut:

- Visual Studio Code
- ModelSim
- Quartus Prime

2.2 IMPLEMENTATION

Passcode merupakan sebuah module *password* yang akan meningkatkan keamanan terhadap data dalam kehidupan sehari-hari. Hal tersebut dapat dicapai dengan melakukan enkripsi, dekripsi, dan menyimpan data - data tersebut kedalam sebuah file. Dimana dengan kapabilitasnya untuk menyimpan *password*, module ini juga memiliki kapabilitas untuk berfungsi sebagai sebuah "pintu" baik secara kiasan maupun harfiah.

Alur jalan program ini dimulai dengan pengguna memilih fitur apa yang diinginkannya. Terdapat 2 fitur yang disediakan yaitu untuk me-*register* atau menambahkan sebuah password dan melakukan sebuah *login* atau mendapatkan akses. Apabila user ingin menambahkan sebuah *password*, maka program akan menerima input tersebut dan menjalankan algoritma enkripsi. Algoritma enkripsi ini akan menggunakan input *user* dengan sebuah vektor yang akan di *generate* secara *random* oleh program sehingga setiap *password* yang berhasil dienkripsi akan memiliki "kunci" yang berbeda. Hal tersebut dilakukan untuk meningkatkan keamanan algoritma yang kami gunakan. Setelah input berhasil di enkripsi, program akan menyimpan data tersebut kedalam sebuah file bernama "Pass.txt". File ini akan menyimpan *password* dalam bentuk sudah terenkripsi, sehingga jika terjadi *breach* terhadap database, *password-password* yang tersimpan masih dalam bentuk abstrak. Bukan hanya dapat menyimpan dan melakukan enkripsi terhadap sebuah *password*, program kami juga dapat melakukan operasi dekripsi. Proses ini dimulai ketika *user* menginput *password* dan memulai proses *login*. Pada proses *login*, program akan mengambil data dari file dan melakukan proses dekripsi. Setelah melakukan proses tersebut, maka program akan

membandingkan input *user* dengan hasil dekripsi. Apabila menghasilkan sebuah kesamaan, maka akan menampilkan pesan bahwa berhasil dan berpindah kepada *state* selanjutnya.

Kode ini akan menggunakan 4 buah input yaitu *password* berupa string, sebuah *clock*, *start*,*mode*,dan *reset* dimana 4 input terakhir berbentuk *std_logic*. Kode akan terdiri dari 10 State yang dimulai dari *state* IDLE. Alur *state* dari program kami terbagi menjadi 3 dimana input *mode* dan *reset* akan menjadi penentu akan alur tersebut. Apabila mode bernilai "1", maka urutan *state* adalah IDLE, LOGIN, CHECK. Pada *state* CHECK ini, program akan menentukan apakah password yang dimasukan user terbukti benar atau salah. Apabila password terbukti validitasnya, maka *state* berikutnya adalah CORRECT dan apabila validitas password salah, maka *state* berikutnya adalah WRONG. Alur program kedua adalah apabila input mode bernilai "0". Disini, alur *state* adalah IDLE, REG, ENCRYPT, dan SAVE. Apabila *reset* bernilai "1", maka program akan berjalan dari *state* IDLE menuju *state* CLR.

CHAPTER 3

TESTING AND ANALYSIS

3.1 TESTING

Untuk melakukan percobaan pada program, kami menggunakan software ModelSim dan juga Quartus Prime. Untuk percobaan pertama, kita melakukan register, dimana kita akan membuat sebuah password yang akan dienkripsi dan disimpan dalam sebuah file. Untuk menjalankan program, kita wajib memasukkan input ‘1’ pada input “start”, jika input ‘0’ pada input “start” program tidak akan berjalan. Untuk memilih mode register, kita memasukkan value ‘0’ pada input “mode”. Lalu, kita memasukkan value ‘0’ pada input “reset” dikarenakan kita belum memerlukan reset untuk sekarang. Kemudian kita memberikan input berupa string sepanjang 8 huruf yang akan diregister sebagai password pada “input_password”.

Untuk percobaan kedua, kita akan melakukan login, dimana kita akan memasukkan sebuah password untuk mencoba masuk. Percobaan ini akan dilakukan 2 kali dimana kita akan mencoba memasukkan password yang benar dan password yang salah. Untuk memilih mode login, kita memasukkan value ‘1’ pada input “mode”. Lalu, kita memasukkan value ‘0’ pada input “reset” dikarenakan kita belum memerlukan reset untuk sekarang. Kemudian kita memberikan input berupa string sepanjang 8 huruf pada “input_password”. Input password ini akan dibandingkan dengan password yang ada dalam file Pass.txt.

Untuk percobaan terakhir, kita akan melakukan reset, dimana kita akan menghapus semua password yang ada di dalam file Pass.txt. Untuk melakukan reset, kita hanya perlu memberikan value ‘1’ pada input “reset”.

3.2 RESULT

Hasil Percobaan pertama, program akan melakukan register dengan test bench. Hasil wave dari Modelsim seperti berikut.

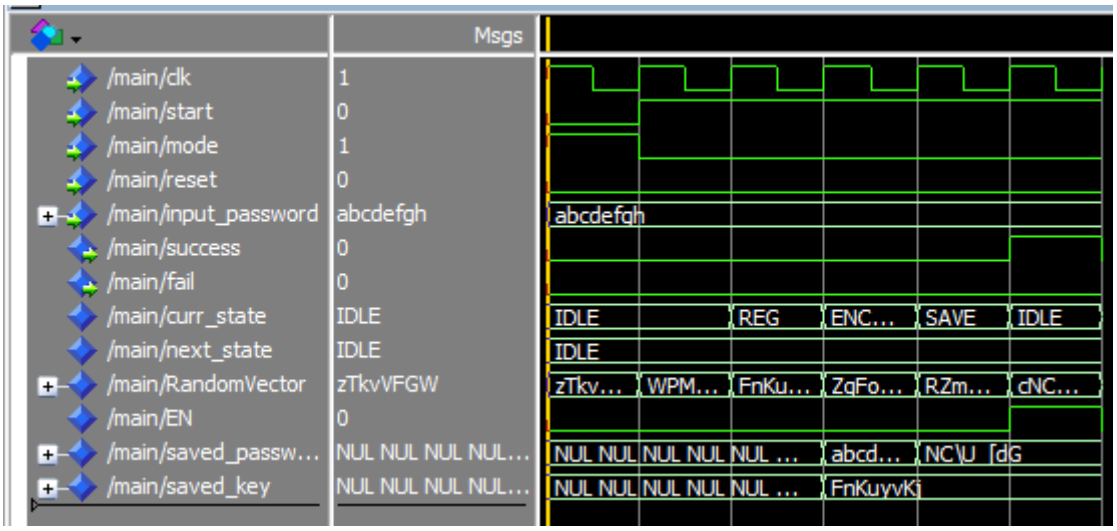


Fig 1. Hasil Simulasi Register

Dimana dapat kita lihat bahwa *state* berhasil berpindah dari IDLE dan berakhir pada SAVE. Dapat terlihat juga sebuah *RandomVector* yang merupakan sebuah KEY yang di generate secara random. Didapatkan juga sebuah perubahan yang terjadi kepada password database seperti berikut.

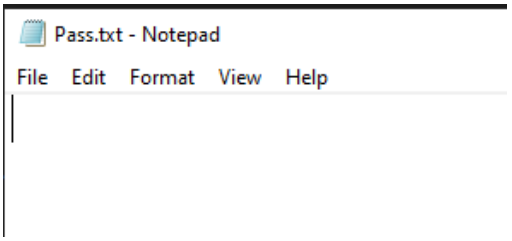


Fig 2. Database Before

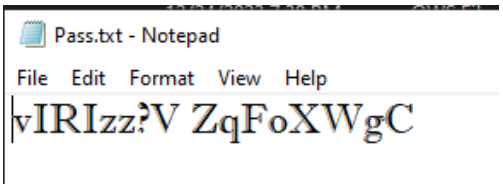


Fig 3. Database After

Pada percobaan kedua, program akan dijalankan untuk melakukan *login* dengan password yang sudah kita isi. Hasil wave pada modelsim akan menjadi seperti berikut.

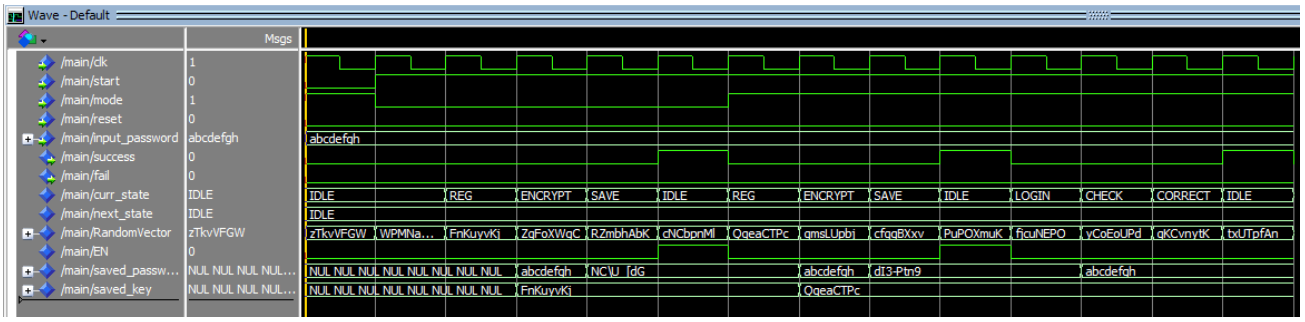


Fig 4. Hasil Simulasi Login

Dapat terlihat pada hasil simulasi bahwa *state* akan dimulai dari IDLE dan berakhir pada CORRECT. Perbedaan pada input apabila dibandingkan pada percobaan sebelumnya terletak pada *mode* dimana, kali ini, nilai dari input tersebut adalah 1 untuk menandakan bahwa user ingin melakukan proses LOGIN. Sama seperti proses sebelumnya, terdapat juga sebuah *RandomVector* yang akan di generate, namun pada kali ini, *RandomVector* tersebut tidak akan digunakan. Terlihat juga sebuah signal success dan fail yang menandakan keberhasilan berjalannya program, dimana keduanya akan menghasilkan output pada *state* terakhir dan pada kali ini, signal *success* bernilai "1" yang menandakan keberhasilan LOGIN.

Percobaan ketiga adalah menguji fitur "Reset" yang akan menghapus database yang digunakan. Hasil Wave pada Modelsim sebagai berikut.

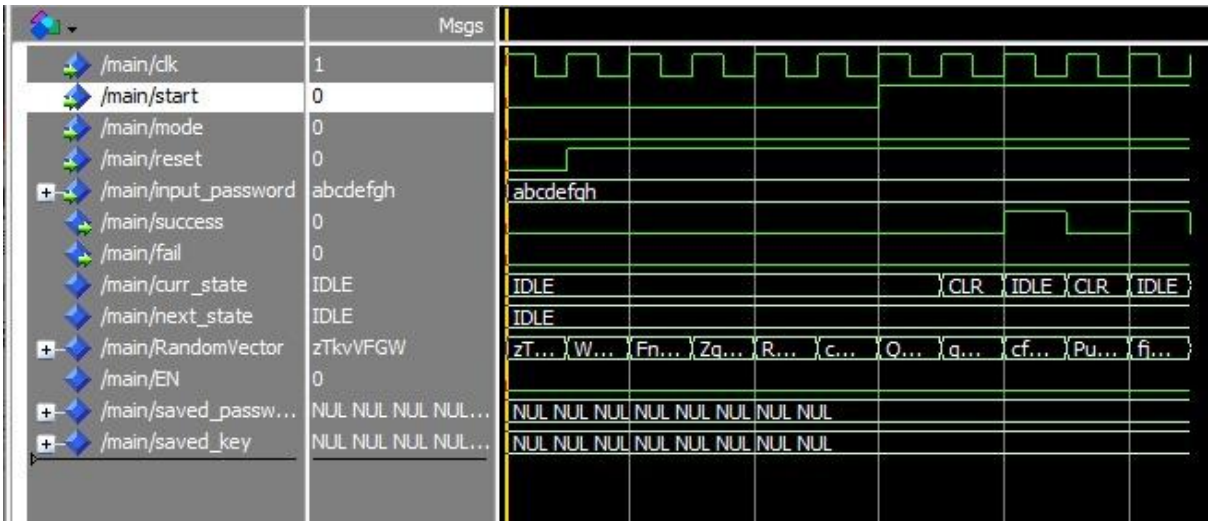


Fig 5. Hasil Simulasi Reset

Terlihat pada kali ini, nilai untuk input *reset* bernilai “1” sehingga *state* yang akan dilalui oleh program hanya IDLE dan CLR. Terdapat juga hasil pada database password sebagai berikut.

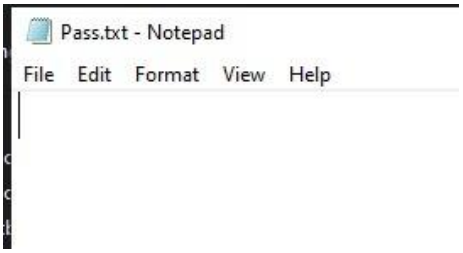


Fig 6. Kondisi Database setelah Reset

Yang menunjukan bahwa semua password yang terdapat pada file sebelumnya, telah hilang setelah menggunakan fitur RESET.

Percobaan terakhir adalah dengan menggunakan TestBench yang menghasilkan simulasi ModelSim sebagai berikut :

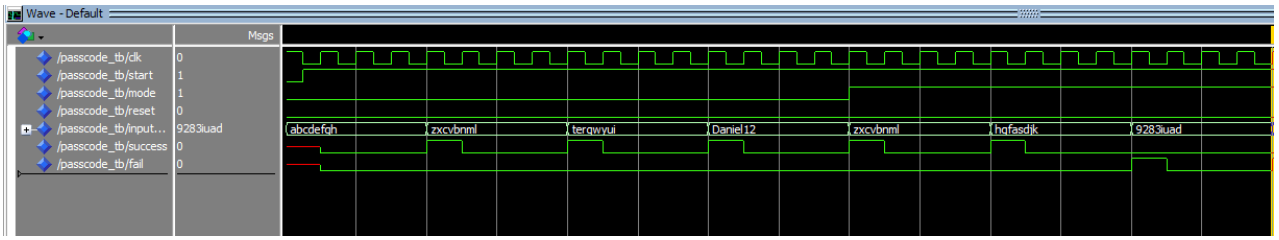


Fig 7. Hasil Simulasi TestBench

Kali ini, Testbench digunakan untuk melakukan REGISTER dan juga melakukan LOGIN. Terlihat bawa pertama - tama, program akan meng-input 4 buah password ke dalam database. Setelah itu, nilai pada input *mode* akan berubah menjadi “1” dimana program akan melakukan fase kedua Testbench yaitu Login. Pada fase Login, 2 password pertama yang kita coba berhasil, ditandakan dengan signal *success* bernilai “1”, sedangkan password terakhir gagal, ditandakan dengan signal *fail* bernilai “1”, hal tersebut dikarenakan password yang digunakan pada fase tersebut, tidak terdaftar pada database, sehingga hasil LOGIN adalah gagal.

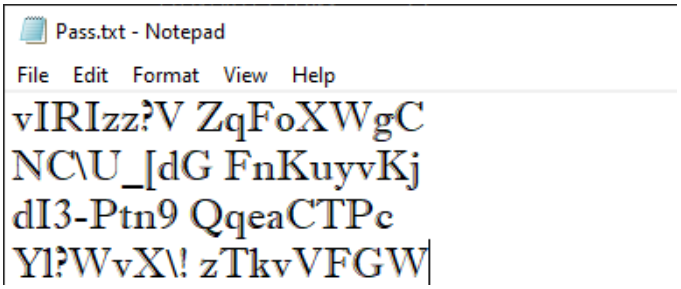


Fig 8. Isi Database setelah TestBench

3.3 ANALYSIS

Pada percobaan pertama, state yang dilalui adalah: IDLE ⇒ REG ⇒ ENCRYPT ⇒ SAVE ⇒ IDLE. Pada state ENCRYPT, program akan menjalankan procedure “encrypt_pass” dimana procedure ini akan melakukan enkripsi pada password tersebut. Kemudian pada state SAVE, password yang telah di enkripsi akan disimpan pada file Pass.txt. Setelah password disimpan, program akan kembali ke state awal.

Untuk percobaan kedua, state yang akan dilewati adalah: IDLE ⇒ LOGIN ⇒ CHECK ⇒ CORRECT / WRONG ⇒ IDLE. Pada state CHECK, program akan menjalankan procedure “check_valid_password” dimana input password akan dibandingkan dengan password yang ada di dalam file Pass.txt. Jika password benar, program akan melanjutkan ke state CORRECT dan jika salah program masuk ke state WRONG. Setelah selesai, program akan kembali ke state awal.

Untuk percobaan ketiga, state yang akan dilewati pada percobaan adalah: IDLE ⇒ CLR ⇒ IDLE. Pada state CLR, program akan membersihkan semua isi Pass.txt. Setelah pembersihan file Pass.txt selesai, program akan kembali ke state awal.

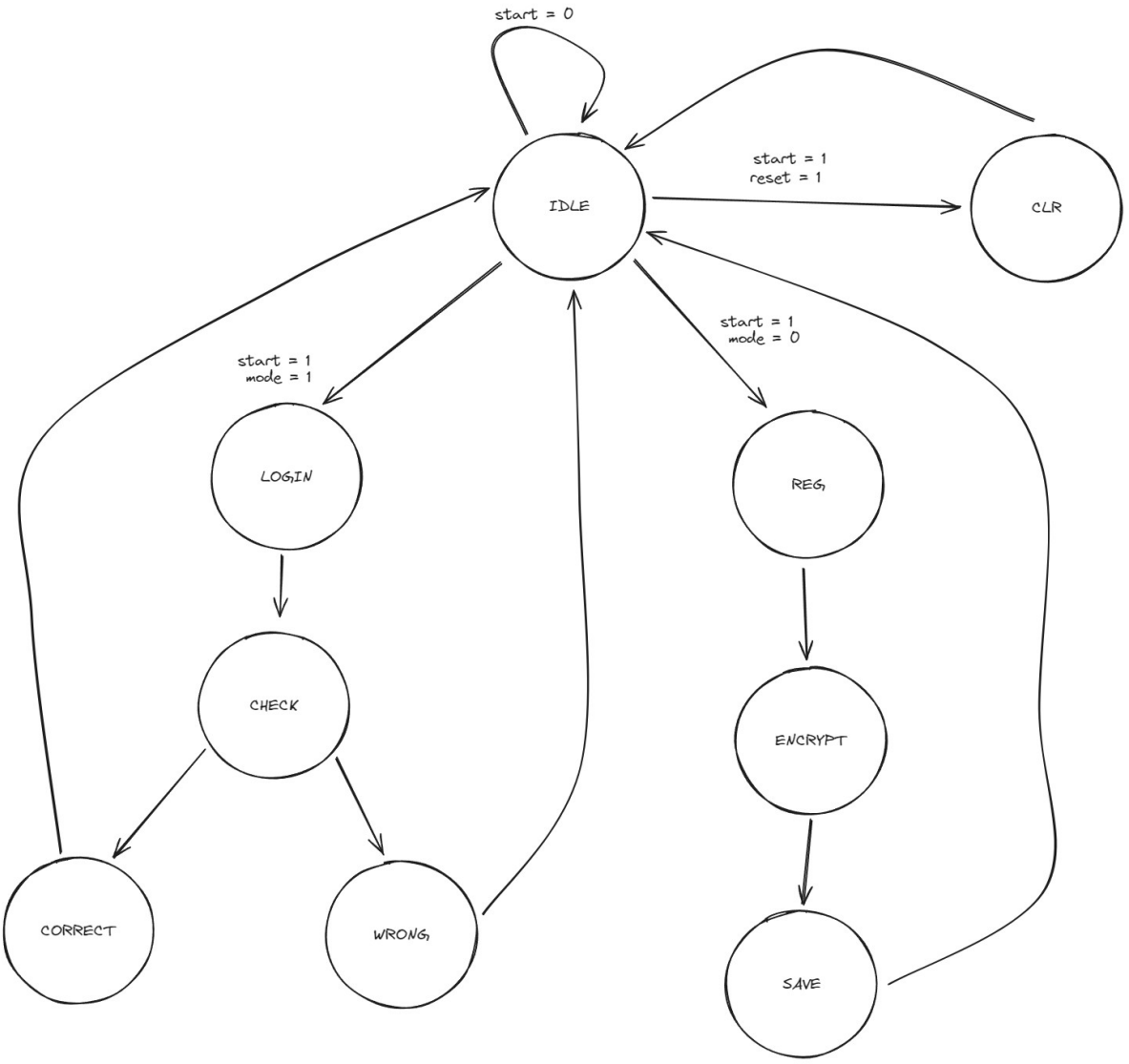


Fig 9. State Diagram

CHAPTER 4

CONCLUSION

Kelompok kami telah berhasil merancang dan mengembangkan sistem Passcode dengan enkripsi dan dekripsi. Proyek ini diwujudkan dengan mengimplementasikan algoritma enkripsi yang kuat untuk melindungi integritas password. Kami telah menciptakan mekanisme untuk pengaturan dan penghapusan password yang efektif, yang memungkinkan interaksi yang aman dan pribadi untuk pengguna. Melalui simulasi, kami telah memverifikasi fungsionalitas kode VHDL dan memastikan bahwa sistem beroperasi dengan tepat, termasuk respons yang akurat terhadap input password yang benar dan salah. Sebagai hasil dari upaya ini, kami telah mencapai solusi keamanan digital yang tidak hanya memenuhi kebutuhan keamanan tetapi juga mudah digunakan dan dikelola.

REFERENCES

[1] S. Team et al., “VHDL component and Port Map tutorial,” Invent Logics, https://allaboutfpga.com/vhdl-component-port-map-tutorial/ (accessed Dec. 18, 2023).

[2] J. J. Jensen, “How to create a finite-state machine in VHDL,” VHDLwhiz, https://vhdlwhiz.com/finite-state-machine/ (accessed Dec. 18, 2023).

[3] “String,” VHDL - String, https://peterfab.com/ref/vhdl/vhdl_renerta/mobile/source/vhd00070.htm (accessed Dec. 18, 2023).

[4] J. J. Jensen, “How to use a for loop in VHDL,” VHDLwhiz, https://vhdlwhiz.com/for-loop/ (accessed Dec. 18, 2023).

[5] Russell, “VHDL Example Code of file Io,” Nandland, https://nandland.com/file-input-output/ (accessed Dec. 18, 2023).

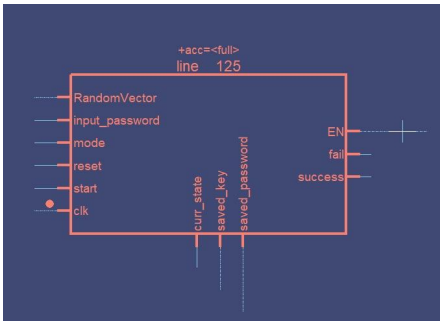
[6] P. Loshin and M. Cobb, “What is encryption and how does it work? - techtarget,” Security, https://www.techtarget.com/searchsecurity/definition/encryption (accessed Dec. 17, 2023).

[7] J. J. Jensen, “How to generate random numbers in VHDL,” VHDLwhiz, https://vhdlwhiz.com/random-numbers/ (accessed Dec. 21, 2023).

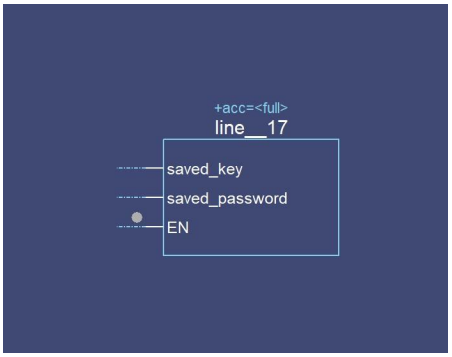
APPENDICES

Appendix A: Project Schematic

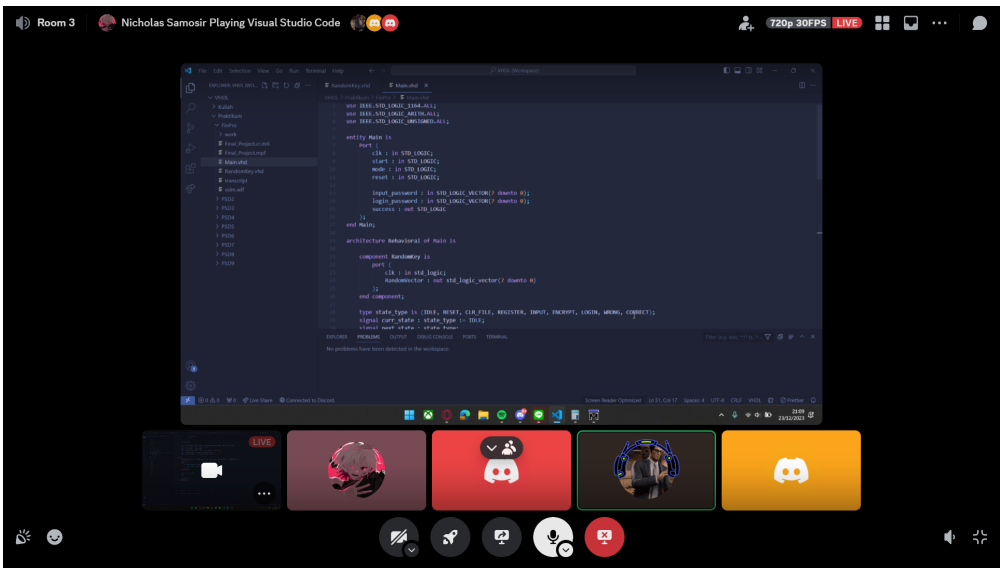
a. Main

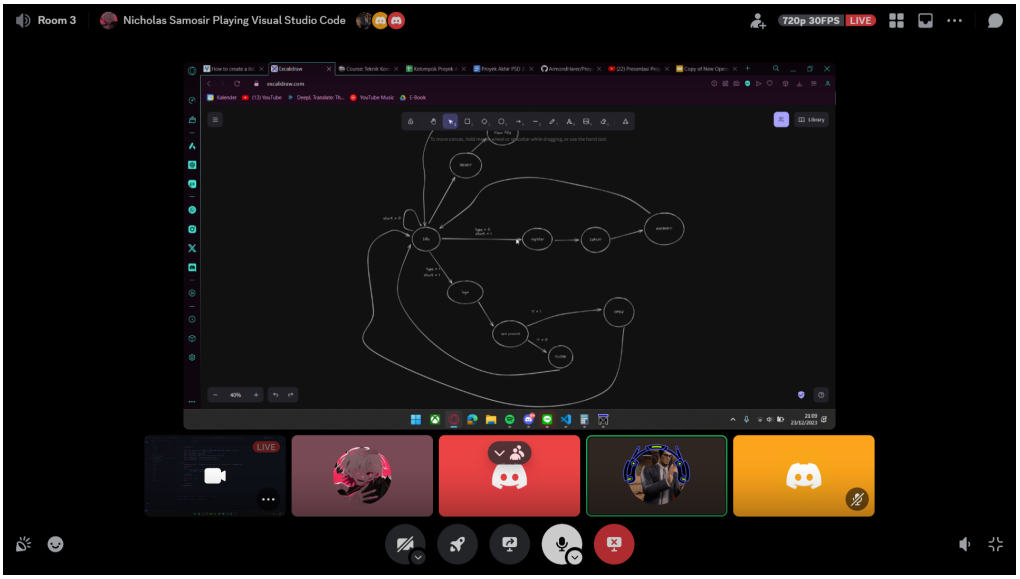


b. File Handling



Appendix B: Documentation




Room 3
Nicholas Samosir Playing Visual Studio Code
720p 30FPS
LIVE