



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

RESEPSIONIS HOTEL OTOMATIS

GROUP 5

ARIA BIMA SAKTI	2206062970
M. AVICENNA RAFFAIZ ADIHARSA	2206062844
FADLIHAJJAN CAREL AGFATA	2206826822
MUHAMMAD LUTFI SETIADI	2206059805

PREFACE

Puji syukur kehadiran Tuhan Yang Maha Esa. Atas rahmat dan hidayah-Nya, kelompok proyek dapat menyelesaikan tugas makalah yang berjudul “Resepsionis Hotel Otomatis” dengan tepat waktu. Tak lupa kami menyampaikan terima kasih sebesar-besarnya kepada asisten laboratorium Digilab yang telah memberikan arahan dan bantuan selama menyelesaikan proyek ini.

Proyek berjudul “Resepsionis Hotel Otomatis” dirancang dengan tujuan untuk mengotomatisasi proses penerimaan tamu di hotel. Dengan sistem ini, kami berharap dapat meningkatkan efisiensi operasional, memperbaiki pengalaman pelanggan, menjaga keamanan data, serta melakukan penghematan biaya operasional yang signifikan.

Kami berharap bahwa proyek ini dapat memberikan manfaat yang signifikan bagi industri perhotelan dan menjadi langkah maju dalam penerapan teknologi dalam operasional hotel. Namun, kami menyadari bahwa makalah ini tak lepas dari kekurangan atau kurang kesempurnaan selama penyusunan makalah proyek ini. Oleh karena itu, kami menyampaikan permohonan maaf serta terbuka untuk kritik dan saran demi perbaikan di masa mendatang.

Depok, December 06, 2025

Group 5

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

- 1.1 Background
- 1.2 Project Description
- 1.3 Objectives
- 1.4 Roles and Responsibilities

CHAPTER 2: IMPLEMENTATION

- 2.1 Equipment
- 2.2 Implementation

CHAPTER 3: TESTING AND ANALYSIS

- 3.1 Testing
- 3.2 Result
- 3.3 Analysis

CHAPTER 4: CONCLUSION

REFERENCES

APPENDICES

- Appendix A: Project Schematic
- Appendix B: Documentation

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

Dalam industri perhotelan, efisiensi dan kenyamanan pelanggan adalah hal yang sangat penting. Dengan meningkatnya teknologi, banyak hotel mulai beralih ke sistem otomatis untuk meningkatkan efisiensi dan memperbaiki pengalaman pelanggan. Proyek ini dirancang dengan tujuan tersebut.

Di era digital saat ini, teknologi telah merambah seluruh aspek kehidupan kita, termasuk industri perhotelan. Dengan kemajuan teknologi, hotel kini memiliki kemampuan untuk mengotomatisasi berbagai proses operasional, termasuk proses penerimaan tamu. Hotel dapat memberikan layanan yang lebih cepat dan efisien kepada tamunya sekaligus mengurangi beban kerja stafnya dengan cara ini.

Selain efisiensi, keamanan dan privasi data pelanggan menjadi perhatian utama dalam industri perhotelan. Hotel dapat menggunakan sistem otomatis untuk memastikan data pelanggan aman dan tidak dapat diakses oleh pihak yang tidak berkepentingan. Proyek ini dibuat dengan mempertimbangkan masalah keamanan dan privasi, termasuk fitur seperti enkripsi kata sandi untuk melindungi data pelanggan.

1.2 PROJECT DESCRIPTION

Sistem ini dirancang untuk hotel dengan tiga lantai dan dua belas kamar. Ketika kamu datang untuk menginap, mereka memasukkan detail/informasi seperti nama, waktu menginap, username, dan password ke dalam sistem. Dimana setelah tamu menerima password, password akan dienkripsi sebelum disimpan ke dalam database untuk menjaga keamanan data.

Setelah detail tersebut dimasukkan, sistem akan mengirim username dan password (yang telah dienkripsi) ke door lock kamar yang dituju. Doorlock ini memiliki sistem dekripsi yang akan mengubah password kembali ke bentuk aslinya. Dengan memasukkan username dan password yang diterima, tamu dapat mengakses tujuan kamar tersebut. Ini membuat

proses check-in menjadi lebih cepat dan efisien, sementara juga memberikan tingkat keamanan yang lebih tinggi.

Secara keseluruhan, proyek ini bertujuan untuk membuat proses check-in hotel menjadi lebih lancar dan aman bagi tamu dengan mengintegrasikan algoritma enkripsi .

1.3 OBJECTIVES

The objectives of this project are as follows:

1. Meningkatkan efisiensi, dengan otomatisasi proses check-in dapat mengurangi waktu tunggu dan meningkatkan efisiensi operasional.
2. Pengurangan biaya, dengan mengurangi kebutuhan angka staf resepsionis hotel dengan menghemat efisiensi biaya operasional.
3. Meningkatkan customer experience, dengan proses yang lebih cepat, mudah, dan efisien dapat meningkatkan kepuasan dan loyalitas pelanggan.
4. Menjaga keamanan data, dengan menerapkan enkripsi password, data pribadi pelanggan dilindungi secara efektif.

1.4 ROLES AND RESPONSIBILITIES

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

Roles	Responsibilities	Person
Leader	Menulis laporan Menganalisa kode main sekaligus bagian encryption serta testbench Membuat PPT	Aria Bima Sakti
Anggota Kelompok	Menulis kode main sekaligus bagian encryption dan decryption	Fadlihajjan Carel Agfata
Anggota Kelompok	Membantu Penulisan Laporan Schematic	Muhammad Avicenna Raffaiz Adiharsa
Anggota Kelompok	Menulis kode testbench Membantu Pembuatan PPT	Muhammad Lutfi Setiadi

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:

- ModelSim
- VSCode
- Intel Quartus Prime/Vivado
- GitHub
- Google Docs
- Canva

2.2 IMPLEMENTATION

Sistem input data sangat penting dalam proyek "Resepsionis Hotel Otomatis". Sistem ini dimaksudkan untuk menerima dan memproses informasi yang diberikan oleh tamu hotel, seperti nama, lama menginap, username, dan password. Interface pengguna sistem ini dimaksudkan untuk memudahkan tamu hotel dalam memasukkan informasinya. Hasilnya, sistem ini tidak hanya memfasilitasi proses check-in yang efisien, namun juga memastikan bahwa pengalaman pengguna diprioritaskan.

Sistem ini menggunakan teknologi enkripsi canggih untuk menjaga keamanan informasi pribadi tamu hotel. Ketika pengunjung memasukkan kata sandi mereka, sistem mengenkripsinya sebelum menyimpannya di database. Prosedur ini melindungi data dari akses tidak sah dan menjamin privasi tamu hotel.

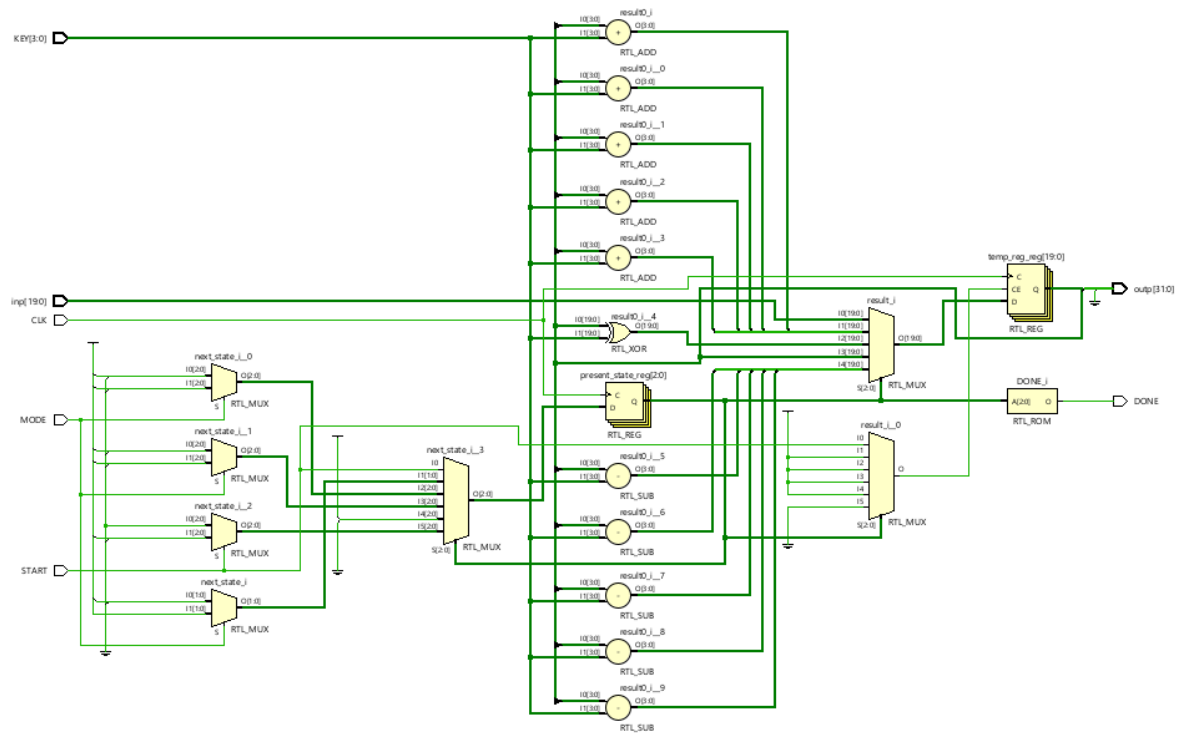


Fig 1. Schematic

Proses otentikasi sangat penting bagi tamu hotel untuk mendapatkan akses ke kamar mereka. Setelah tamu memasukkan nama pengguna dan kata sandi, sistem akan memvalidasi informasi ini. Akses ke ruangan akan diberikan jika data yang dimasukkan sesuai dengan data di sistem. Prosedur ini memastikan bahwa hanya tamu yang berwenang yang memiliki akses ke kamar, sehingga meningkatkan keamanan. Selain itu, proses otentikasi ini dimaksudkan agar mudah digunakan oleh para tamu, memungkinkan mereka mengakses kamar mereka dengan cepat dan mudah tanpa bantuan staf hotel.

CHAPTER 3

TESTING AND ANALYSIS

3.1 TESTING

Selama pengujian, ditemukan beberapa masalah yang menyebabkan sebagian sistem kurang berfungsi dengan sempurna. Algoritma pemesanan kamar masih memiliki bug, terutama pada urutan finite state machine yang belum rapi. Kesalahan juga muncul pada main program yang perlu disesuaikan kembali dengan komponen lain agar sistem bekerja konsisten. Selain itu, angka acak yang dihasilkan terlalu ekstrem dengan loncatan nilai yang tidak realistis untuk penomoran kamar hotel, sehingga perlu diperbaiki. Di sisi lain, modul enkripsi-dekripsi VHDL berjalan baik, memungkinkan tamu memiliki identitas dan autentikasi password yang aman selama menginap.

3.2 RESULT

Secara keseluruhan, seluruh sistem telah berjalan dengan baik dan dapat dikompilasi sesuai dengan spesifikasi yang diminta oleh user, termasuk proses pengujian menggunakan testbench yang menunjukkan hasil sesuai harapan. Namun, masih terdapat satu isu penting yang perlu diperhatikan, yaitu pada bagian penghasil angka acak. Nilai acak yang dihasilkan menunjukkan loncatan yang terlalu besar sehingga tidak realistis untuk digunakan sebagai penomoran kamar hotel. Kondisi ini berpotensi menimbulkan ketidaksesuaian dalam alur pemesanan kamar, sehingga mekanisme pengacakan tersebut perlu disesuaikan agar menghasilkan rentang angka yang lebih stabil dan relevan dengan kebutuhan sistem.

Name	Status	Type	Or	Modified
Enkripsi.vhd	✓	VHDL	0	12/07/2025 11:54:13 ...
TBEnkripsi.vhd	✓	VHDL	1	12/07/2025 11:55:00 ...
Doorlock.vhd	✓	VHDL	2	12/07/2025 11:55:53 ...
TBDoorlock.vhd	✓	VHDL	3	12/07/2025 11:56:31 ...

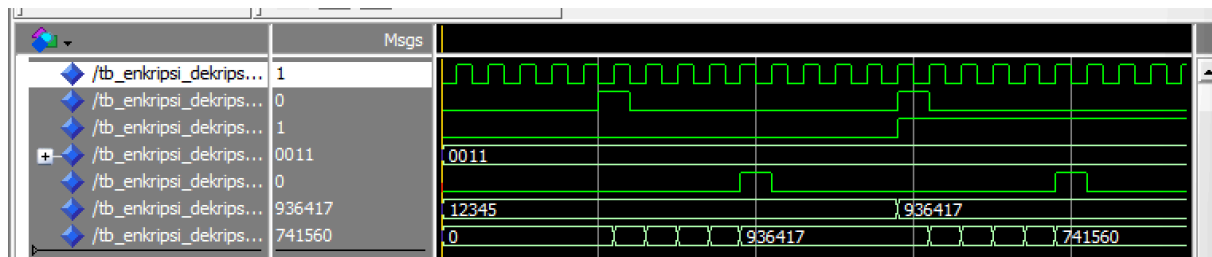


Fig 2. Testing Result

3.3 ANALYSIS

Pada proses pengujian sistem secara menyeluruh, seluruh modul inti berhasil dijalankan dan dikompilasi tanpa error. Baik fungsionalitas utama maupun integrasi antar-komponen menunjukkan bahwa desain sistem sudah sesuai dengan spesifikasi. Pengujian melalui testbench juga mengonfirmasi bahwa alur sinyal dan respons modul sudah mengikuti skenario yang direncanakan, sehingga dapat disimpulkan bahwa struktur dasar sistem telah bekerja dengan stabil.

Meskipun demikian, ditemukan satu masalah signifikan pada bagian generator angka acak yang digunakan untuk penomoran kamar hotel. Hasil pengacakan menampilkan loncatan nilai yang terlalu ekstrem dan tidak sesuai dengan konteks penomoran kamar yang seharusnya bersifat teratur dan berada pada rentang tertentu. Ketidakwajaran ini berpotensi menimbulkan inkonsistensi dalam proses pemesanan kamar dan dapat mengganggu validitas data selama operasional sistem. Oleh karena itu, algoritma pengacakan perlu disempurnakan agar lebih realistis dan sesuai kebutuhan.

Secara keseluruhan, sistem telah menunjukkan performa yang baik, namun hasil testing juga menggarisbawahi pentingnya melakukan penyempurnaan pada aspek tertentu agar kualitasnya optimal. Perbaikan pada generator angka acak diharapkan dapat meningkatkan keandalan sistem secara keseluruhan. Dengan langkah tersebut, sistem akan lebih siap untuk digunakan dalam skenario nyata dan mampu memberikan keluaran yang konsisten, efektif, dan sesuai standar operasional yang diharapkan.

CHAPTER 4

CONCLUSION

Proyek Encryptor Decryptor Generator berhasil diimplementasikan menggunakan bahasa VHDL dengan menerapkan algoritma AES 128-bit sebagai dasar proses enkripsi dan dekripsi. Sistem bekerja berdasarkan opcode, di mana opcode "0001" mengaktifkan enkripsi dan opcode "0010" mengaktifkan dekripsi, sementara keseluruhan alur dikendalikan oleh Finite State Machine (FSM) yang mengatur tahapan AES seperti AddRoundKey, SubBytes, ShiftRows, dan MixColumns. Implementasi ini membuktikan bahwa konsep perancangan sistem digital dapat diterapkan secara efektif untuk menghasilkan modul kriptografi yang fungsional dan stabil.

Berdasarkan hasil simulasi pada waveform, seluruh sinyal utama seperti clock, reset, opcode, data input, serta output hasil enkripsi/dekripsi telah menunjukkan respons yang sesuai dengan harapan. Terlihat bahwa perubahan opcode memicu transisi proses dengan benar, dan data output (misalnya nilai 936417 dan 741560 pada hasil simulasi) menunjukkan bahwa modul telah berhasil memproses input sesuai algoritma yang diterapkan. Selain itu, modul mampu dikompilasi dan berjalan tanpa error, menandakan bahwa struktur sistem dan integrasi antar-komponen telah mendukung fungsionalitas secara keseluruhan.

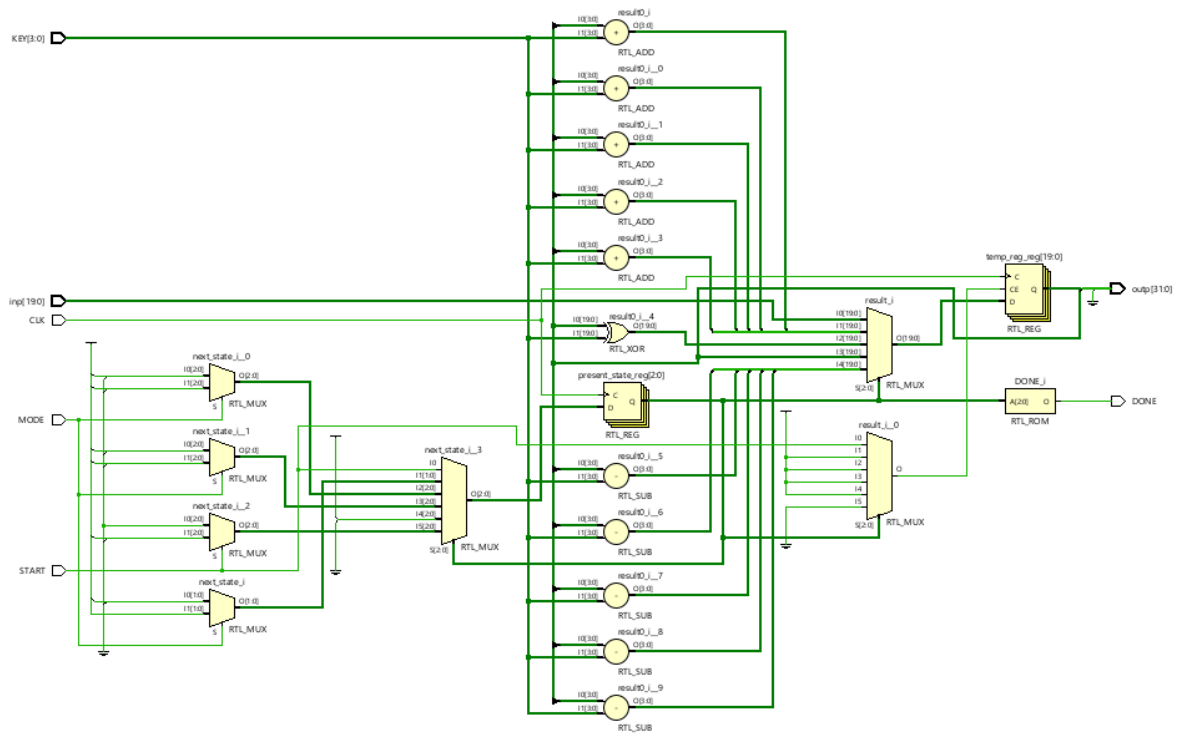
Proyek ini juga berhasil mengintegrasikan seluruh sembilan modul praktikum Perancangan Sistem Digital, mulai dari Behavioral Style, Structural Programming, Looping Construct, Procedure, Function, Impure Function, FSM, Microprogramming, hingga Testbench. Hasil pengujian menunjukkan bahwa implementasi tersebut bekerja konsisten dengan teori. Meskipun pada bagian sistem lain ditemukan isu minor terkait generator angka acak yang menghasilkan loncatan nilai tidak realistis, bagian enkripsi-dekripsi justru menunjukkan performa sangat baik. Secara keseluruhan, proyek ini telah mencapai tujuan praktikum dengan kualitas implementasi yang matang dan siap dijadikan dasar untuk pengembangan lebih lanjut pada aplikasi keamanan data digital.

REFERENCES

- [1] J. Shah, "Push-Button-Door-VHDL," *GitHub*, Feb. 27, 2023.
<https://github.com/shahjui2000/Push-Button-Door-VHDL-> (accessed Dec. 7, 2025).
- [2] M. Kocaoglu, "AES-Advanced-Encryption-Standard-VHDL," *GitHub*, Apr. 26, 2022.
[https://github.com/muhammedkocaoglu/AES-Advanced-Encryption-Standard-VHDL/ tree/master](https://github.com/muhammedkocaoglu/AES-Advanced-Encryption-Standard-VHDL/tree/master) (accessed Dec. 7, 2025).
- [3] "Basic writing and formatting syntax," *GitHub Docs*.
<https://docs.github.com/en/get-started/writing-on-github/getting-started-with-writing-and-formatting-on-github/basic-writing-and-formatting-syntax> (accessed Dec. 7, 2025).
- [4] mohamedamine99, "GitHub - mohamedamine99/Digital-Lock-with-VHDL-state-machine," *GitHub*, 2025.
<https://github.com/mohamedamine99/Digital-Lock-with-VHDL-state-machine> (accessed Dec. 07, 2025).

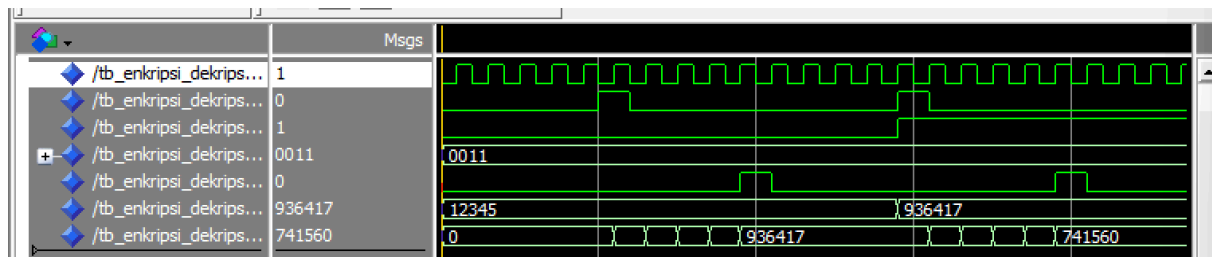
APPENDICES

Appendix A: Project Schematic



Appendix B: Documentation

```
D: > UI > SEMESTER 7 > Perancangan Sistem Digital > PROYEK AKHIR > Enkripsi.vhd
1  library IEEE;
2  use IEEE.std_logic_1164.all;
3  use IEEE.numeric_std.all;
4
5  entity Enkripsi_Dekripsi is
6  port (
7      inp  : in  integer range 0 to 1048575; -- 20 bit max
8      START : in  std_logic;
9      MODE  : in  std_logic;                -- '0' = enkripsi, '1' = dekripsi
10     KEY   : in  std_logic_vector(3 downto 0);
11     CLK   : in  std_logic;
12     DONE  : out std_logic;                -- '1' saat hasil siap
13     outp  : out integer
14 );
15 end entity Enkripsi_Dekripsi;
16
17 architecture rtl of Enkripsi_Dekripsi is
18
19     -- Data internal 20-bit
20     signal INPUT_vec : std_logic_vector(19 downto 0);
21     signal temp_reg  : std_logic_vector(19 downto 0) := (others => '0');
22     signal temp_next : std_logic_vector(19 downto 0) := (others => '0');
23
24     -- FSM state
25     type state is (init, add_s, s_xor, swap_s, sub_s, done_s);
26     signal present_state, next_state : state := init;
27
28 begin
29
30     -----
31     -- Kombinasi: next_state dan temp_next
32     -----
33     process(present_state, START, MODE, KEY, inp, temp_reg)
34         variable result : std_logic_vector(19 downto 0);
```



```

"      using waveform 110: 1/110ns001
VSIM 3> run -all
# ** Note: Mulai ENKRIPSI
#   Time: 50 ns  Iteration: 0  Instance: /tb_enkripsi_dekripsi
# ** Note: Enkripsi selesai. outp_tb = 936417
#   Time: 95 ns  Iteration: 3  Instance: /tb_enkripsi_dekripsi
# ** Note: Mulai DEKRIPSI
#   Time: 145 ns Iteration: 0  Instance: /tb_enkripsi_dekripsi
# ** Note: Dekripsi selesai. outp_tb (plaintext) = 741560
#   Time: 195 ns Iteration: 3  Instance: /tb_enkripsi_dekripsi
# ** Note: Simulasi Enkripsi_Dekripsi selesai
#   Time: 295 ns Iteration: 0  Instance: /tb_enkripsi_dekripsi

```