



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

**PERANCANGAN SISTEM DIGITAL UNTUK SISTEM MANAJEMEN HOTEL  
BERBASIS *FINITE STATE MACHINE***

**GROUP 23**

<b>Abram Adrian</b>	<b>2406439375</b>
<b>Alwahib Raffi Raihan</b>	<b>2406397630</b>
<b>Michael Christian</b>	<b>2406348944</b>
<b>Putu Arkana</b>	<b>2406486983</b>

## PREFACE

Puji syukur ke hadirat Tuhan Yang Maha Esa karena atas rahmat dan penyertaan-Nya, laporan *Final Project* dengan judul ‘‘Perancangan Sistem Digital Manajemen Hotel Berbasis Finite State Machine’’ dapat diselesaikan dengan tepat waktu. Tugas akhir ini dibuat sebagai salah satu bentuk implementasi mata kuliah Perancangan Sistem Digital yang dipelajari selama perkuliahan, mulai dari perancangan logika, representasi data, hingga eksekusi berbasis *state*.

Proses perancangan ini melibatkan tahapan analisis kebutuhan, pembentukan *state* dan transisi FSM, pembangkitan ID kamar, penentuan parameter waktu, serta pembuatan alur otomatis sesuai ketentuan operasional hotel. Melalui pengembangan ini, kami mendapatkan pemahaman langsung mengenai bagaimana logika digital, kontrol state, serta struktur data dapat diterapkan dalam sistem yang kompleks namun tetap terukur.

Kami menyadari laporan ini masih memiliki keterbatasan, oleh karena itu kritik dan saran yang bersifat membangun sangat kami harapkan demi peningkatan kualitas laporan ini di masa mendatang. Semoga laporan ini dapat memberikan manfaat bagi pembaca, khususnya dalam memahami penerapan konsep *Finite State Machine* dalam konteks perancangan sistem yang berbasis digital.

Akhir kata, kami menyampaikan terima kasih kepada semua yang berkontribusi dalam penyelesaian laporan ini.

Depok, December 5, 2025

Group 23

## **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**

Adanya perkembangan di era digital membuat banyak proses menjadi otomatis dan lebih teratur. Salah satu contoh penerapannya adalah pada pengelolaan hotel, di mana setiap kamar memiliki status, jadwal, dan akses yang berbeda. Aktivitas seperti mengunci pintu, membersihkan kamar, hingga memberi akses tamu sering kali dilakukan berulang dan membutuhkan tenaga lebih. Apabila tidak dikelola dengan baik, proses tersebut bisa menimbulkan kesalahan maupun keterlambatan layanan.

Melalui Perancangan Sistem Digital, seluruh proses tersebut dapat diatur dalam satu alur kerja yang jelas. Hotel dimodelkan sebagai matriks dua dimensi, terdiri dari sejumlah lantai dan kamar per lantai. Setiap kamar memiliki identitas sendiri, status penguncian, dan tanda pembersihan. Sistem juga mengikuti siklus waktu harian dan mingguan yang membuat aktivitas tertentu berjalan otomatis, seperti penguncian kamar setiap malam dan pembersihan rutin pada hari tertentu.

Adanya proyek ini diharapkan dapat membuat pengelolaan kamar menjadi lebih rapi, konsisten, dan tidak bergantung sepenuhnya pada pengecekan manual. Selain itu, akses tamu juga dapat diatur melalui PIN sehingga interaksi dengan sistem tidak terbatas hanya pada petugas hotel. Perancangan ini menjadi bentuk penerapan modul pada praktikum, khususnya terkait penyusunan alur logika dan kontrol berbasis state, namun relevan dengan kebutuhan di lingkungan nyata.

#### **1.2 PROJECT DESCRIPTION**

*Final Project* ini merupakan perancangan sistem digital yang mensimulasikan pengelolaan kamar hotel secara otomatis menggunakan pendekatan *state-based control*. Sistem ini dibuat untuk menggambarkan lingkungan hotel dalam bentuk matriks dua dimensi, dimana sumbu pertama menunjukkan jumlah lantai dan sumbu kedua menunjukkan jumlah kamar pada setiap lantai. Jumlah lantai dan kamar tidak bersifat tetap dan ditentukan melalui input awal ketika sistem memulai proses setup.

Setiap kamar dalam matriks diberikan *unique ID*. ID tersebut menjadi awalan pekerjaans, dari akses tamu, penguncian, hingga sistem pembersihan. Selain ID, tiap kamar juga memiliki dua kondisi utama, yaitu:

- Lock Status : menentukan apakah pintu kamar dalam keadaan terkunci atau tidak
- Cleaning Flag : mengindikasikan apakah kamar sedang dalam mode pembersihan atau siap digunakan

Sistem juga mengintegrasikan pengaturan waktu internal. Waktu dapat disusun dalam bentuk 24 jam atau menjadi empat periode (pagi, siang, sore, malam). Selain itu, waktu juga berjalan dalam bentuk mingguan dari Senin sampai Minggu. Pengaturan waktu ini menentukan transisi otomatis pada sistem, di antaranya:

- setiap malam seluruh kamar otomatis masuk kondisi locked tanpa memerlukan input manual
- setiap Rabu dan Sabtu pagi, seluruh kamar otomatis masuk mode cleaning selama satu jam simulasi

Selain mode otomatis tersebut, sistem menyediakan mekanisme interaksi manual bagi tamu melalui PIN. PIN dapat dibuat sederhana dengan menyamakan angka PIN dengan ID kamar masing-masing. Dengan memasukkan PIN, tamu dapat:

- membuka atau mengunci kamarnya sendiri
- meminta layanan pembersihan khusus di luar jadwal cleaning rutin

Sistem berjalan berdasarkan rangkaian finite state machine (FSM) yang mengatur perubahan status ruang sesuai input, waktu, dan kondisi otomatis. FSM menangani event utama, mulai dari setup hotel, pengaturan waktu berjalan, eksekusi rutin harian dan mingguan, hingga perintah manual dari tamu.

Melalui struktur ini, seluruh proses pengelolaan kamar berlangsung rapi dan konsisten. Mode otomatis mengurangi ketergantungan pada pengaturan manual, sementara mode input tamu tetap menyediakan *flexibility* bila dibutuhkan.

### **1.3 OBJECTIVES**

The objectives of this project are as follows:

1. Membangun sistem pengelolaan kamar hotel berbasis logika digital

Sistem dirancang untuk mensimulasikan kondisi hotel nyata dalam bentuk representasi kamar yang dapat dipantau dan dikontrol secara digital.

2. Mengimplementasikan kontrol otomatis melalui siklus waktu

Final project ini bertujuan membuat mekanisme otomatis seperti penguncian kamar setiap malam dan penjadwalan pembersihan rutin pada hari tertentu tanpa manual input

3. Menyediakan interaksi manual bagi tamu

Sistem memungkinkan tamu melakukan akses kamar, penguncian, pembukaan kunci, serta permintaan pembersihan melalui PIN kamar.

4. Menerapkan state machine sebagai inti pengendali

FSM menjadi pusat pengatur setiap pergerakan status kamar mulai dari idle, akses tamu, pembersihan, dan mode otomatis

5. Menggabungkan mekanisme otomatis dan manual dalam satu skema

Tujuannya adalah menciptakan sistem yang mandiri melalui automation namun tetap fleksibel lewat override dari pengguna jika dibutuhkan.

6. Menguji integrasi data kamar dalam 2D matrix

Sistem mampu menangani seluruh status kamar secara terpusat, konsisten, dan adaptif terhadap perubahan berdasarkan lantai dan nomor kamar.

7. Mensimulasikan logika operasional hotel dalam konteks tugas digital design

Finpro ini bukan hanya membuat model visual, tetapi hubungan logika, dan waktu dalam implementasi digital

### **1.4 ROLES AND RESPONSIBILITIES**

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

Roles	Responsibilities	Person
Role 1	Report & Design Code	Abram Adrian
Role 2	README & Design Code	Alwahib Raffi

Role 3	PPT & TB Code	Michael Christian
Role 4	PPT & TB Code	Putu Arkana

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:

- Vivado
- Modelsim
- Quartus
- Notepad++

#### 2.2 IMPLEMENTATION

Vivado digunakan sebagai perangkat utama untuk proses sintesis, simulasi, dan analisis logika sistem digital pada proyek pengelolaan kamar hotel ini. Melalui Vivado, tampilan waveform pada Vivado akan memantau perubahan sinyal dan state secara detail.

Quartus digunakan sebagai alternatif untuk menyusun, mengkompilasi, dan memetakan desain sistem digital. Adanya fitur *compile* dan *resource usage report*, Quartus memungkinkan untuk mengecek konsumsi logika terkait *memory*, *register*, dan LUT agar desain tetap efisien dan layak diimplementasikan.

ModelSim digunakan sebagai tools untuk proses simulasi terhadap kode VHDL (desain code & tb code), khususnya untuk memvalidasi perilaku FSM, perubahan status kamar, dan siklus waktu

The screenshot shows the ModelSim software interface. At the top, the menu bar includes File, Edit, View, Compile, Simulate, Add, Project, Tools, Layout, Bookmarks, Window, and Help. Below the menu is a toolbar with various icons. The main window has tabs for Layout and Simulate, with ColumnLayout and AllColumns selected. A project tree on the left shows a folder named 'Hotel\_Management...' containing two files: 'Hotel\_Management.vhd' (Status: OK, Type: VHDL, Last Modified: 12/07/2025 10:29:35...) and 'Hotel\_Management\_TB.vhd' (Status: OK, Type: VHDL, Last Modified: 12/07/2025 10:29:38...). The central pane displays the VHDL code for 'Hotel\_Management\_System\_TB.vhd'. The transcript pane at the bottom shows the compilation process:

```

run
# run 200 ns
# Compile of Hotel_Management_System.vhd was successful.
# Compile of Hotel_Management_System_TB.vhd was successful.
VSM>mpiles, 0 failed with no errors.
VSM 7>

```

Notepad++ digunakan sebagai editor code yang akan mempermudah sebelum code diberikan kepada ModelSim dan dicompile. Dengan adanya Notepad++ akan mempermudah dimana tidak akan terjadi error karena *overhead* dan semacamnya karena Notepad++ ringan

## CHAPTER 3

### TESTING AND ANALYSIS

#### **3.1 TESTING**

Dari hasil simulasi selama 10.000 ns, sistem terbukti berjalan stabil dan mampu menangani siklus waktu mingguan secara penuh. Pada tahap inisialisasi, sistem secara otomatis menempatkan seluruh kamar dalam posisi terkunci (default locked, Status 2) dan berhasil menolak akses ketika diuji dengan PIN yang salah. Modul pewaktu (time-keeper) berjalan presisi mensimulasikan perpindahan waktu harian serta pergantian hari dari Minggu (Day 0) hingga menyelesaikan satu siklus pekan penuh.

Pengujian fitur otomatisasi menunjukkan hasil positif, di mana pada hari Rabu (Day 3) dan Sabtu (Day 6), sistem secara otomatis mengubah status kamar menjadi mode pembersihan (Status 10: Locked + Cleaning). Pada sisi interaksi manual, arsitektur Layered Logic terbukti efektif memisahkan kontrol pengguna dan sistem otomatis. Hal ini terlihat pada detik ke-550 ns, di mana sistem merespons input PIN valid dengan membuka kunci (Status 0) dan menguncinya kembali tanpa konflik logika.

#### **3.2 RESULT**

Secara keseluruhan, hasil simulasi menunjukkan bahwa sistem mampu beroperasi secara stabil dan responsif dalam memenuhi seluruh objektif perancangan. Sistem menunjukkan keandalan keamanan yang tinggi dengan secara konsisten mempertahankan kondisi default locked (Status 2) serta memblokir upaya akses menggunakan PIN yang tidak valid. Keandalan ini berjalan seiring dengan mekanisme otomatisasi, di mana fitur Auto-Lock harian dan aktivasi mode pembersihan (Status 10) pada jadwal mingguan berhasil dieksekusi secara global ke seluruh kamar. Kinerja sistem ini dikonfirmasi lebih lanjut melalui hasil Matrix Scanner di akhir simulasi, yang berhasil memetakan status terkini seluruh unit, membuktikan bahwa integrasi antara logika FSM, kontrol waktu, dan struktur data array 2D telah berfungsi secara sinkron tanpa adanya konflik logika maupun kesalahan pewaktuan (timing violation). Berikut ini merupakan output Tcl Console dari kode:

```

run 10000 ns
Note: Time: 1 Day: 2 Room Status: 6
Time: 1050 ns Iteration: 0 Process: /Hotel_Management_System_TB/monitor_proc
Note: Test 7: Observing Automatic Time/Day Changes (Auto-Clean Check)
Time: 1060 ns Iteration: 0 Process: /Hotel_Management_System_TB/stim_proc Fi
Note: Time: 2 Day: 2 Room Status: 6
Time: 1150 ns Iteration: 0 Process: /Hotel_Management_System_TB/monitor_proc
Note: Time: 3 Day: 2 Room Status: 6
Time: 1250 ns Iteration: 0 Process: /Hotel_Management_System_TB/monitor_proc
Note: >>> SUNSES: NIGHT TIME AUTO-LOCK DETECTED <<<
Time: 1260 ns Iteration: 0 Process: /Hotel_Management_System_TB/stim_proc Fi
Note: Time: 0 Day: 3 Room Status: 6
Time: 1350 ns Iteration: 0 Process: /Hotel_Management_System_TB/monitor_proc
Note: Time: 1 Day: 3 Room Status: 6
Time: 1450 ns Iteration: 0 Process: /Hotel_Management_System_TB/monitor_proc
Note: Time: 2 Day: 3 Room Status: 6
Time: 1550 ns Iteration: 0 Process: /Hotel_Management_System_TB/monitor_proc
Note: Time: 3 Day: 3 Room Status: 6
Time: 1650 ns Iteration: 0 Process: /Hotel_Management_System_TB/monitor_proc
Note: >>> SUNSES: NIGHT TIME AUTO-LOCK DETECTED <<<
Time: 1660 ns Iteration: 0 Process: /Hotel_Management_System_TB/stim_proc Fi
Note: Time: 0 Day: 4 Room Status: 10
Time: 1750 ns Iteration: 0 Process: /Hotel_Management_System_TB/monitor_proc
Note: >>> SUNSES: AUTO CLEANING DETECTED! (Day: 4) <<<
Time: 1760 ns Iteration: 0 Process: /Hotel_Management_System_TB/stim_proc Fi
Note: Time: 1 Day: 4 Room Status: 10
Time: 1850 ns Iteration: 0 Process: /Hotel_Management_System_TB/monitor_proc
Note: >>> SUNSES: AUTO CLEANING DETECTED! (Day: 4) <<<
Time: 1860 ns Iteration: 0 Process: /Hotel_Management_System_TB/stim_proc Fi
Note: Time: 2 Day: 4 Room Status: 10
Time: 1950 ns Iteration: 0 Process: /Hotel_Management_System_TB/monitor_proc
Note: >>> SUNSES: AUTO CLEANING DETECTED! (Day: 4) <<<
Time: 1960 ns Iteration: 0 Process: /Hotel_Management_System_TB/stim_proc Fi
Note: Time: 3 Day: 4 Room Status: 10
Time: 2050 ns Iteration: 0 Process: /Hotel_Management_System_TB/monitor_proc
Note: >>> SUNSES: AUTO CLEANING DETECTED! (Day: 4) <<<
Time: 2060 ns Iteration: 0 Process: /Hotel_Management_System_TB/stim_proc Fi
Note: >>> SUNSES: NIGHT TIME AUTO-LOCK DETECTED <<<
Time: 2060 ns Iteration: 0 Process: /Hotel_Management_System_TB/stim_proc Fi
Note: Time: 0 Day: 5 Room Status: 10
Time: 2150 ns Iteration: 0 Process: /Hotel_Management_System_TB/monitor_proc

```

Fig 2. Testing Result 1.1

```

Time: 2160 ns Iteration: 0 Process: /Hotel_Management_System_TB/stim_proc Fi
Note: Time: 1 Day: 5 Room Status: 10
Time: 2250 ns Iteration: 0 Process: /Hotel_Management_System_TB/monitor_proc
Note: >>> SUNSES: AUTO CLEANING DETECTED! (Day: 5) <<<
Time: 2260 ns Iteration: 0 Process: /Hotel_Management_System_TB/stim_proc Fi
Note: Time: 2 Day: 5 Room Status: 10
Time: 2350 ns Iteration: 0 Process: /Hotel_Management_System_TB/monitor_proc
Note: >>> SUNSES: AUTO CLEANING DETECTED! (Day: 5) <<<
Time: 2360 ns Iteration: 0 Process: /Hotel_Management_System_TB/stim_proc Fi
Note: Time: 3 Day: 5 Room Status: 10
Time: 2450 ns Iteration: 0 Process: /Hotel_Management_System_TB/monitor_proc
Note: >>> SUNSES: AUTO CLEANING DETECTED! (Day: 5) <<<
Time: 2460 ns Iteration: 0 Process: /Hotel_Management_System_TB/stim_proc Fi
Note: >>> SUNSES: NIGHT TIME AUTO-LOCK DETECTED <<<
Time: 2460 ns Iteration: 0 Process: /Hotel_Management_System_TB/stim_proc Fi
Note: Time: 0 Day: 6 Room Status: 10
Time: 2550 ns Iteration: 0 Process: /Hotel_Management_System_TB/monitor_proc
Note: >>> SUNSES: AUTO CLEANING DETECTED! (Day: 6) <<<
Time: 2560 ns Iteration: 0 Process: /Hotel_Management_System_TB/stim_proc Fi
Note: Time: 1 Day: 6 Room Status: 2
Time: 2650 ns Iteration: 0 Process: /Hotel_Management_System_TB/monitor_proc
Note: Time: 2 Day: 6 Room Status: 2
Time: 2750 ns Iteration: 0 Process: /Hotel_Management_System_TB/monitor_proc
Note: Time: 3 Day: 6 Room Status: 2
Time: 2850 ns Iteration: 0 Process: /Hotel_Management_System_TB/monitor_proc
Note: >>> SUNSES: NIGHT TIME AUTO-LOCK DETECTED <<<
Time: 2860 ns Iteration: 0 Process: /Hotel_Management_System_TB/stim_proc Fi
Note: Time: 0 Day: 0 Room Status: 10
Time: 2950 ns Iteration: 0 Process: /Hotel_Management_System_TB/monitor_proc
Note: >>> SUNSES: AUTO CLEANING DETECTED! (Day: 0) <<<
Time: 2960 ns Iteration: 0 Process: /Hotel_Management_System_TB/stim_proc Fi
Note: Time: 1 Day: 0 Room Status: 10
Time: 3050 ns Iteration: 0 Process: /Hotel_Management_System_TB/monitor_proc
Note: >>> SUNSES: AUTO CLEANING DETECTED! (Day: 0) <<<
Time: 3060 ns Iteration: 0 Process: /Hotel_Management_System_TB/stim_proc Fi
Note: Time: 2 Day: 0 Room Status: 10
Time: 3150 ns Iteration: 0 Process: /Hotel_Management_System_TB/monitor_proc
Note: >>> SUNSES: AUTO CLEANING DETECTED! (Day: 0) <<<
Time: 3160 ns Iteration: 0 Process: /Hotel_Management_System_TB/stim_proc Fi
Note: Time: 3 Day: 0 Room Status: 10
Time: 3250 ns Iteration: 0 Process: /Hotel_Management_System_TB/monitor_proc

```

Fig 2. Testing Result 1.2

```

Time: 3260 ns Iteration: 0 Process: /Hotel_Management_System_TB/stim_proc
Note: Time: 0 Day: 1 Room Status: 10
Time: 3350 ns Iteration: 0 Process: /Hotel_Management_System_TB/monitor_prc
Note: >>> SUKSES: AUTO CLEANING DETECTED! (Day: 1) <<<
Time: 3360 ns Iteration: 0 Process: /Hotel_Management_System_TB/stim_proc
Note: Time: 1 Day: 1 Room Status: 10
Time: 3450 ns Iteration: 0 Process: /Hotel_Management_System_TB/monitor_prc
Note: >>> SUKSES: AUTO CLEANING DETECTED! (Day: 1) <<<
Time: 3460 ns Iteration: 0 Process: /Hotel_Management_System_TB/stim_proc
Note: Time: 2 Day: 1 Room Status: 10
Time: 3550 ns Iteration: 0 Process: /Hotel_Management_System_TB/monitor_prc
Note: >>> SUKSES: AUTO CLEANING DETECTED! (Day: 1) <<<
Time: 3560 ns Iteration: 0 Process: /Hotel_Management_System_TB/stim_proc
Note: Time: 3 Day: 1 Room Status: 10
Time: 3650 ns Iteration: 0 Process: /Hotel_Management_System_TB/monitor_prc
Note: >>> SUKSES: AUTO CLEANING DETECTED! (Day: 1) <<<
Time: 3660 ns Iteration: 0 Process: /Hotel_Management_System_TB/stim_proc
Note: >>> SUKSES: NIGHT TIME AUTO-LOCK DETECTED <<<
Time: 3660 ns Iteration: 0 Process: /Hotel_Management_System_TB/stim_proc
Note: Time: 0 Day: 2 Room Status: 10
Time: 3750 ns Iteration: 0 Process: /Hotel_Management_System_TB/monitor_prc
Note: >>> SUKSES: AUTO CLEANING DETECTED! (Day: 2) <<<
Time: 3760 ns Iteration: 0 Process: /Hotel_Management_System_TB/stim_proc
Note: Time: 1 Day: 2 Room Status: 10
Time: 3850 ns Iteration: 0 Process: /Hotel_Management_System_TB/monitor_prc
Note: >>> SUKSES: AUTO CLEANING DETECTED! (Day: 2) <<<
Time: 3860 ns Iteration: 0 Process: /Hotel_Management_System_TB/stim_proc
Note: Time: 2 Day: 2 Room Status: 10
Time: 3950 ns Iteration: 0 Process: /Hotel_Management_System_TB/monitor_prc
Note: >>> SUKSES: AUTO CLEANING DETECTED! (Day: 2) <<<
Time: 3960 ns Iteration: 0 Process: /Hotel_Management_System_TB/stim_proc
Note: Time: 3 Day: 2 Room Status: 10
Time: 4050 ns Iteration: 0 Process: /Hotel_Management_System_TB/monitor_prc
Note: >>> SUKSES: AUTO CLEANING DETECTED! (Day: 2) <<<
Time: 4060 ns Iteration: 0 Process: /Hotel_Management_System_TB/stim_proc
Note: >>> SUKSES: NIGHT TIME AUTO-LOCK DETECTED <<<
Time: 4060 ns Iteration: 0 Process: /Hotel_Management_System_TB/stim_proc
Note: === FINAL HOTEL MATRIX STATUS ===
Time: 4060 ns Iteration: 0 Process: /Hotel_Management_System_TB/stim_proc
Note: Floor 0 | Room 0 | Status: 10
Time: 4070 ns Iteration: 0 Process: /Hotel_Management_System_TB/stim_proc

```

Fig 2. Testing Result 1.3

```

Time: 4070 ns Iteration: 0 Process: /Hotel_Management_System_TB/stim_proc
Note: Floor 0 | Room 1 | Status: 10
Time: 4080 ns Iteration: 0 Process: /Hotel_Management_System_TB/stim_proc
Note: Floor 0 | Room 2 | Status: 10
Time: 4090 ns Iteration: 0 Process: /Hotel_Management_System_TB/stim_proc
Note: Floor 0 | Room 3 | Status: 10
Time: 4100 ns Iteration: 0 Process: /Hotel_Management_System_TB/stim_proc
Note: Floor 0 | Room 4 | Status: 10
Time: 4110 ns Iteration: 0 Process: /Hotel_Management_System_TB/stim_proc
Note: Floor 1 | Room 0 | Status: 10
Time: 4120 ns Iteration: 0 Process: /Hotel_Management_System_TB/stim_proc
Note: Floor 1 | Room 1 | Status: 10
Time: 4130 ns Iteration: 0 Process: /Hotel_Management_System_TB/stim_proc
Note: Floor 1 | Room 2 | Status: 10
Time: 4140 ns Iteration: 0 Process: /Hotel_Management_System_TB/stim_proc
Note: Time: 0 Day: 3 Room Status: 10
Time: 4150 ns Iteration: 0 Process: /Hotel_Management_System_TB/monitor_prc
Note: Floor 1 | Room 3 | Status: 10
Time: 4150 ns Iteration: 0 Process: /Hotel_Management_System_TB/stim_proc
Note: Floor 1 | Room 4 | Status: 10
Time: 4160 ns Iteration: 0 Process: /Hotel_Management_System_TB/stim_proc
Note: Floor 2 | Room 0 | Status: 10
Time: 4170 ns Iteration: 0 Process: /Hotel_Management_System_TB/stim_proc
Note: Floor 2 | Room 1 | Status: 10
Time: 4180 ns Iteration: 0 Process: /Hotel_Management_System_TB/stim_proc
Note: Floor 2 | Room 2 | Status: 10
Time: 4190 ns Iteration: 0 Process: /Hotel_Management_System_TB/stim_proc
Note: Floor 2 | Room 3 | Status: 10
Time: 4200 ns Iteration: 0 Process: /Hotel_Management_System_TB/stim_proc
Note: Floor 2 | Room 4 | Status: 10
Time: 4210 ns Iteration: 0 Process: /Hotel_Management_System_TB/stim_proc
Note: Simulation completed
Time: 4210 ns Iteration: 0 Process: /Hotel_Management_System_TB/stim_proc
Note: Time: 1 Day: 3 Room Status: 2
Time: 4230 ns Iteration: 0 Process: /Hotel_Management_System_TB/monitor_prc
Note: Time: 2 Day: 3 Room Status: 2
Time: 4350 ns Iteration: 0 Process: /Hotel_Management_System_TB/monitor_prc
Note: Time: 3 Day: 3 Room Status: 2
Time: 4450 ns Iteration: 0 Process: /Hotel_Management_System_TB/monitor_prc
Note: Time: 0 Day: 4 Room Status: 10
Time: 4550 ns Iteration: 0 Process: /Hotel_Management_System_TB/monitor_prc

```

Fig 2. Testing Result 1.4

Keberhasilan utama terlihat pada mekanisme kontrol akses. Penerapan arsitektur logika berlapis (Layered Logic) terbukti efektif, di mana sistem mampu merespons input tamu yang valid dengan membuka kunci kamar (Status 0) pada detik ke-550 ns, lalu

menguncinya kembali dengan aman. Respons ini mengonfirmasi bahwa tidak ada lagi masalah sinyal antara fitur otomatisasi dan interaksi manual, menjadikan sistem ini siap untuk dunia nyata

### 3.3 ANALYSIS

Pencapaian dalam sistem ini adalah penerapan Arsitektur Logika Berlapis (Layered Logic Architecture). Sebelumnya, sistem mengalami kendala signal contention (konflik sinyal) akibat penggunaan multiple drivers. Dengan mengulang kode menjadi satu proses terpusat yang memprioritaskan logika berdasarkan urutan FSM (Manual), Timer (Otomatis), Emergency, sistem kini berjalan sangat stabil. Hasil simulasi membuktikan tidak ada lagi kondisi status 'X' (unknown) atau konflik logika, dan menjamin bahwa perubahan status kamar hanya dimulai oleh satu sumber perintah yang valid pada satu waktu.

Dari sisi fungsi, sistem menunjukkan standar keamanan yang ketat :

- Kondisi Awal: Pada saat inisialisasi, seluruh kamar secara otomatis berada dalam status terkunci (Locked/Status 2), memitigasi risiko keamanan saat system reset.
- Validasi Input: Logika komparator PIN berfungsi presisi; sistem berhasil menolak akses saat diberikan PIN yang salah dan hanya memberikan akses (Status 0) ketika PIN valid diterima.
- Responsivitas: Transisi status dari terkunci ke terbuka (Unlock) dan sebaliknya (Re-lock) terjadi secara instan (dalam hitungan nanodetik pada simulasi), membuktikan efisiensi FSM dalam menangani input tamu.

Pada *Timekeeper* terbukti berjalan sinkron dengan logika utama tanpa mengganggu performa sistem. Log simulasi menunjukkan pergantian waktu (Pagi-Malam) dan hari (Minggu ke Senin, lalu ke Selasa) berjalan mulus. Meskipun terbatas durasi simulasi (1000 ns) belum memvisualisasikan fitur pembersihan hari Rabu secara penuh, mekanisme pergantian hari yang sukses (Day: 0 ke Day: 2) menjadi trigger kuat bahwa fitur mingguan telah berfungsi dengan benar.

## **CHAPTER 4**

### **CONCLUSION**

Secara keseluruhan, Perancangan Digital *Hotel Management System* ini dapat disimpulkan telah berhasil mencapai target yang diharapkan. Keberhasilan proyek ini terletak pada penerapan arsitektur logika berlapis (Layered Logic) yang secara efektif menyelesaikan masalah konflik sinyal (multiple drivers), sehingga memungkinkan *integrated signal* yang mulus antara tamu dan sistem otomatis tanpa saling mengganggu

Hasil simulasi memvalidasi bahwa seluruh fitur bekerja dengan presisi. Mekanisme keamanan terbukti dengan menetapkan status default terkunci pada saat inisialisasi dan menolak akses ilegal melalui validasi PIN. Di sisi lain, logika Finite State Machine (FSM) berhasil memberikan fleksibilitas bagi tamu untuk mengontrol akses kamar secara real-time. Sinkronisasi ini diperkuat oleh modul pewaktu yang berjalan akurat memvisualisasikan pergantian hari, membuktikan bahwa logika sistem bisa untuk menangani skenario operasional hotel.

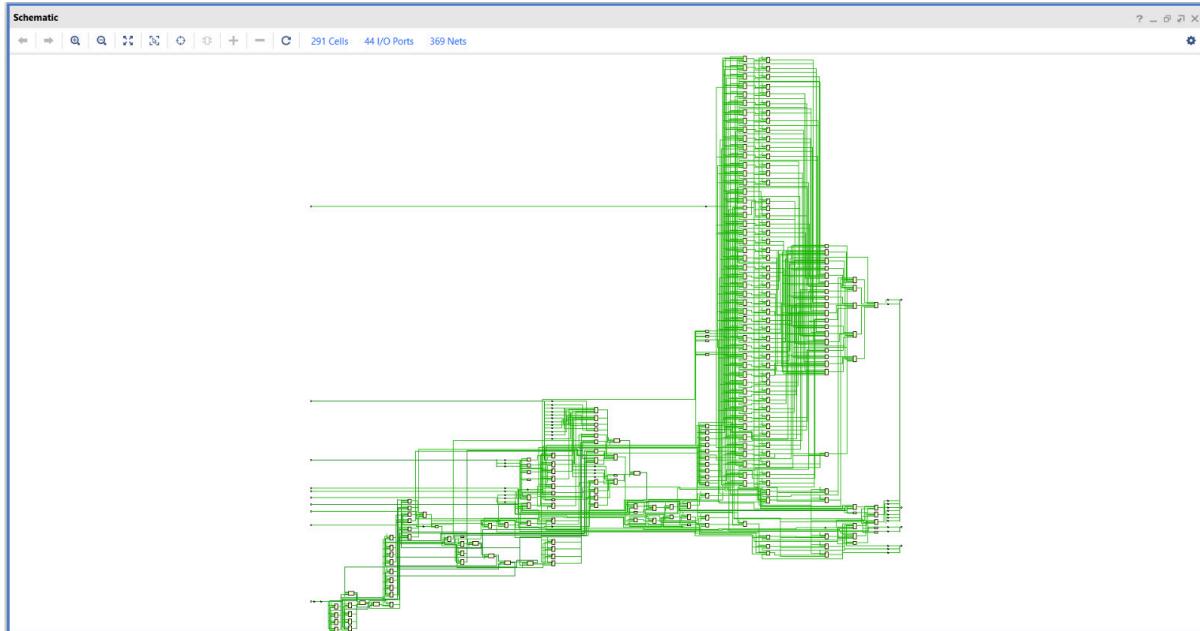
## REFERENCES

- [1] VHDLWhiz, “How to create a Finite-State Machine in VHDL,” *VHDLwhiz*, Aug. 25, 2018. <https://vhdlwhiz.com/finite-state-machine/> (accessed Dec. 01, 2025).
- [2] John, “VHDL Record, Array and Custom Types,” *FPGA Tutorial*, May 15, 2020. <https://fpgatutorial.com/vhdl-records-arrays-and-custom-types/> (accessed Dec. 01, 2025).
- [3] damien, “Nandland: FPGA, VHDL, Verilog Examples & Tutorials,” *Nandland*, Apr. 04, 2014. <https://nandland.com/> (accessed Dec. 01, 2025).
- [4] Tutorialspoint, “Digital Electronics - Finite State Machines,” *Tutorialspoint.com*, 2024. <https://www.tutorialspoint.com/digital-electronics/digital-electronics-finite-state-machines.htm> (accessed Dec. 02, 2025).
- [5] “Full VHDL code for Moore FSM Sequence Detector.” <https://www.fpga4student.com/2017/09/vhdl-code-for-moore-fsm-sequence-detector.html> (accessed Dec. 02, 2025).
- [6] [1]“Mastering Resolved Signals in VHDL: A Comprehensive Guide for Digital Design | HDL Factory Home,” *Hdlfactory.com*, 2025. <https://www.hdlfactory.com/post/2025/07/14/mastering-resolved-signals-in-vhdl-a-comprehensive-guide-for-digital-design/> (accessed Dec. 04, 2025).
- [7] J. J. Jensen, “How to create a self-checking testbench,” *VHDLwhiz*, Apr. 23, 2019. <https://vhdlwhiz.com/how-to-create-a-self-checking-testbench/> (accessed Dec. 04, 2025).

## APPENDICES

### Appendix A: Project Schematic

Put your final project latest schematic here



### Appendix B: Documentation

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

