



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

**HOTEL BOOKING SYSTEM**

**GROUP AP04**

<b>BINTANG SIAHAAN</b>	<b>2206024322</b>
<b>MUHAMMAD ABRISAM CAHYO</b>	<b>2206026050</b>
<b>ACHMAD ZAIDAN LAZUARDY</b>	<b>2206059793</b>
<b>AULIA ANUGRAH AZIZ</b>	<b>2206059364</b>

## **PREFACE**

Puji dan syukur kita panjatkan kehadiran Tuhan YME yang telah rahmatnya sehingga kami sebagai kelompok AP4 dapat menyelesaikan tugas akhir Praktikum Perancangan Sistem Digital yang berjudul “Hotel Booking System”.

Pada zaman ini, perkembangan teknologi informasi dan komunikasi semakin pesat dan tidak bisa dilepaskan dari kehidupan sehari-hari, termasuk dalam dunia perhotelan. Sistem pemesanan kamar hotel secara online menjadi salah satu kebutuhan bagi para wisatawan untuk memudahkan mereka dalam melakukan pemesanan kamar hotel.

Kami kelompok AP4 mengucapkan terima kasih kepada para anggota kelompok yang telah saling bekerja sama dalam membuat program, simulasi, hingga menyusun laporan, sehingga proyek dapat selesai dengan baik. Kami juga berterima kasih kepada asisten lab yang telah membimbing kami. Kami menyadari bahwa tugas akhir ini masih jauh dari sempurna. Oleh karena itu, kami mengharapkan kritik dan saran yang membangun dari pembaca. Semoga tugas akhir ini dapat bermanfaat bagi para pembaca.

Depok, December 15, 2023

Group AP4

## TABLE OF CONTENTS

CHAPTER 1.....	4
INTRODUCTION.....	4
1.1 BACKGROUND.....	4
1.2 PROJECT DESCRIPTION.....	6
1.3 OBJECTIVES.....	6
1.4 ROLES AND RESPONSIBILITIES.....	7
CHAPTER 2.....	9
IMPLEMENTATION.....	9
2.1 EQUIPMENT.....	9
2.2 IMPLEMENTATION.....	9
CHAPTER 3.....	10
TESTING AND ANALYSIS.....	10
3.1 TESTING.....	10
3.2 RESULT.....	15
3.3 ANALYSIS.....	19
CHAPTER 4.....	21
CONCLUSION.....	21
REFERENCES.....	22

# CHAPTER 1

## INTRODUCTION

### 1.1 BACKGROUND

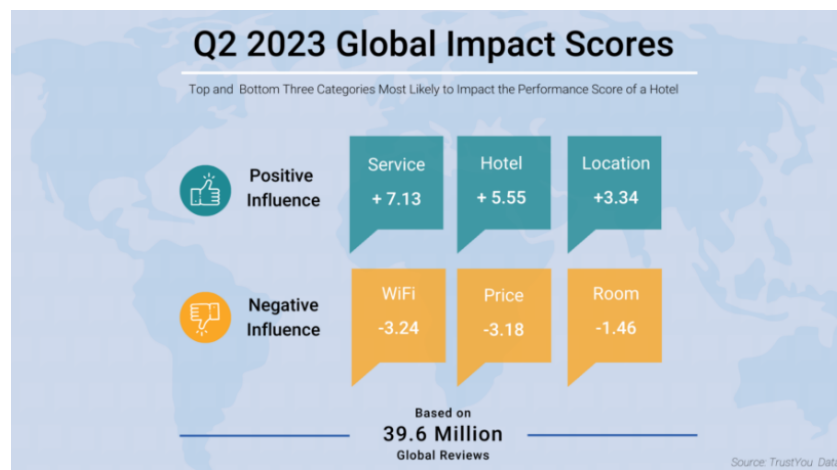
Dalam era modern ini, pengembangan desain perangkat keras memegang peranan penting dalam mendukung kemajuan teknologi. Salah satu bahasa deskripsi perangkat keras yang menjadi landasan dalam merancang, memodelkan, dan mensimulasikan sirkuit digital adalah Very High-Speed Integrated Circuit Hardware Description Language (VHDL). Dalam upaya memahami kompleksitas desain perangkat keras, empat aspek utama muncul sebagai poin fokus, yaitu Interfaces, Modeling (Behavior, Dataflow, Structure), Test Benches, dan proses Analisis, Elaborasi, Simulasi, serta Sintesis.

Interfaces, sebagai fondasi interaksi antar komponen, menuntut pemahaman mendalam terkait pembuatan antarmuka, protokol komunikasi, dan integrasi komponen. Pemodelan (Modeling), dilakukan melalui pendekatan Behavior, Dataflow, dan Structure, menjadi kunci dalam merinci fungsi sirkuit, aliran data, dan struktur fisik. Test Benches, sebagai metode validasi, memiliki peran sentral dalam memastikan kebenaran fungsi desain sebelum langkah implementasi. Sementara itu, proses Analisis, Elaborasi, Simulasi, dan Sintesis membentuk siklus integral dalam pengembangan desain perangkat keras. Dari pemeriksaan karakteristik hingga implementasi sirkuit pada perangkat keras target, setiap langkah memiliki dampak signifikan dalam keberhasilan proyek pengembangan.

Dalam era digital dan globalisasi, industri perhotelan mengalami transformasi signifikan dalam menyajikan layanan kepada pelanggan. Keberadaan sistem pemesanan hotel telah menjadi pilar utama dalam meningkatkan efisiensi operasional dan memberikan pengalaman yang lebih baik kepada para tamu. Sistem pemesanan hotel modern tidak hanya menghadirkan kemudahan dalam memesan kamar, tetapi juga memberikan keunggulan dalam manajemen inventaris, pelacakan reservasi, dan personalisasi layanan untuk memenuhi harapan tamu.

Laporan terbaru dari Hospitality Planet yang mengamati perilaku utama tamu dari sebelum hingga sesudah menginap menyajikan temuan yang signifikan. Menurut laporan tersebut, terlihat bahwa 49% tamu lebih memilih check-in melalui staf, sementara 32% lebih memilih menggunakan perangkat pribadi untuk check-in mandiri, dan 15% memilih check-in

mandiri melalui kios hotel. Dalam hal penyimpanan informasi, hanya 37% tamu yang kembali menyatakan bahwa data mereka disimpan di sistem hotel. Ketika tamu mempertimbangkan untuk memesan ulang, diskon (48%) menjadi insentif paling penting, diikuti oleh poin loyalitas (40%). Temuan ini menggambarkan harapan tamu masa kini yang cenderung mencari pengalaman hotel yang terhubung, efisien, dan memberikan nilai tambah melalui diskon dan program loyalitas.



Gambar 1. Kategori yang paling mungkin untuk memberi dampak terhadap skor performa hotel

Berdasarkan penelitian Q3 2023 Global Impact Scores, terungkap bahwa pelayanan yang baik memiliki dampak positif terhadap ulasan tamu. Implementasi hotel system booking menjadi solusi strategis untuk meningkatkan kualitas pelayanan. Dengan sistem ini, hotel dapat menyediakan layanan check-in yang efisien, memungkinkan tamu untuk memilih antara check-in oleh staf, check-in mandiri menggunakan perangkat pribadi, atau melalui kios hotel. Penggunaan teknologi dalam proses pemesanan juga dapat membantu hotel menyimpan informasi tamu secara terorganisir, menciptakan pengalaman yang mulus dan terhubung. Melalui sistem booking ini, hotel dapat memberikan layanan yang responsif, menyediakan diskon dan program loyalitas yang meningkatkan kepuasan tamu, sebagaimana diindikasikan dalam penelitian Global Impact Scores.

## **1.2 PROJECT DESCRIPTION**

FPGA yang telah kami rancang adalah Hotel System Booking yang mendefinisikan inovasi dalam layanan reservasi kamar hotel. Sistem ini dirancang untuk memberikan pengalaman yang mulus dan terhubung bagi para tamu, tidak hanya memudahkan proses pemesanan tetapi juga mengintegrasikan berbagai aspek, termasuk perhitungan biaya menginap yang akurat. Dalam operasinya, FPGA Hotel System Booking mempertimbangkan sejumlah faktor penting yang mencakup jenis kamar yang dipilih oleh tamu, jumlah orang yang akan menginap, dan waktu menginap. Melalui pemrosesan yang canggih, sistem secara otomatis menghasilkan total biaya yang harus dibayarkan, memberikan kemudahan dan kejelasan bagi tamu terkait dengan komponen keuangan perjalanan mereka.

Tidak hanya sebatas pada pemesanan dan perhitungan biaya, sistem ini juga memastikan bahwa tamu memiliki akses transparan terhadap seluruh proses. Sebagai contoh, jika tamu memberikan jumlah uang yang melebihi total harga kamar, FPGA secara cerdas memberikan uang kembalian sesuai dengan selisihnya. Kelebihan lainnya dari FPGA Hotel System Booking adalah kemampuannya untuk memberikan fleksibilitas dan kejelasan dalam mengelola reservasi. Tamu dapat dengan mudah mengubah jenis kamar, jumlah orang, atau durasi menginap, dan sistem akan langsung menghitung kembali biaya yang sesuai. Dengan demikian, FPGA Hotel System Booking tidak hanya menjadi solusi praktis untuk manajemen reservasi hotel tetapi juga menciptakan pengalaman yang nyaman, transparan, dan terkini bagi para tamu yang menginginkan lebih dari sekadar tempat menginap.

## **1.3 OBJECTIVES**

Tujuan dari proyek ini dapat dijabarkan sebagai berikut:

1. Menerapkan bahasa VHDL untuk meningkatkan fungsionalitas dan efisiensi dalam kehidupan sehari-hari
2. Memperdalam pemahaman mengenai VHDL dan bagaimana penerapannya dalam pemrograman.
3. Mengembangkan desain FSM yang inovatif dengan memanfaatkan kemampuan bahasa VHDL untuk kontrol sistem yang efisien.
4. Mengidentifikasi dan mengimplementasikan strategi untuk meningkatkan keberlanjutan sistem melalui pemahaman dalam VHDL.

5. Memperkuat keterampilan kolaborasi tim dalam pengembangan menggunakan bahasa VHDL.

#### 1.4 ROLES AND RESPONSIBILITIES

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

Roles	Responsibilities	Person
Penentu Ide	Menentukan ide yang ingin diterapkan.	Aulia Anugrah Aziz
Pembuat FSM	Mendesain dan mengimplementasikan FSM berdasarkan ide yang telah ditentukan	Aulia Anugrah Aziz
Pembuat Testbench	Membuat testbench untuk menguji fungsi dan kinerja	Aulia Anugrah Aziz
Pembuat Function	Mengembangkan fungsi-fungsi yang diperlukan	Achmad Zaidan Lazuardy
Pembuat Seven Segment	Mengimplementasikan tampilan Seven Segment sesuai spesifikasi yang dibutuhkan	Bintang Siahaan
Pembuat README	Menyiapkan README yang jelas dan informatif mengenai proyek yang dibuat	Muhammad Abrisam Cahyo Juhartono
Pembuat Powerpoint Presentation	Membuat presentasi PowerPoint yang merangkum proyek dengan baik	Muhammad Abrisam Cahyo Juhartono

Pembuat Laporan	Menyiapkan laporan yang mencakup detail dari proses pembuatan, hasil pengujian, dan evaluasi keseluruhan proyek	Bintang Siahaan
-----------------	---	-----------------

Table 1. Roles and Responsibilities



## **CHAPTER 2**

### **IMPLEMENTATION**

#### **2.1 EQUIPMENT**

Tools yang digunakan pada project ini adalah sebagai berikut:

- VS Code
- ModelSim
- Quartus Prime
- Github

#### **2.2 IMPLEMENTATION**

Pendekatan desain sistem yang telah disebutkan mencakup beberapa gaya dan teknik yang berbeda dalam pengembangan proyek. Dataflow Style digunakan untuk menangani ketergantungan data langsung dalam proyek, sementara Behavioral Style membantu dalam deskripsi fungsi-fungsi kontroler tanpa perlu terlalu terperinci pada implementasi perangkat keras. Penggunaan Testbench sangat penting untuk menguji skenario-skenario berbeda seperti reservasi nomor kamar hotel dalam rentang tertentu, pembayaran, serta waktu check-in/check-out.

Di samping itu, Structural Style digunakan untuk menggambarkan keterkaitan antar modul dalam proyek, sementara Looping Construct memfasilitasi iterasi dalam akses data. Selain itu, penggunaan Procedure, Function, dan Impure Function sangat berguna dalam menjalankan tugas-tugas seperti reservasi, pembatalan, dan notifikasi. Teknik Finite State Machine (FSM) digunakan untuk mewakili keadaan-keadaan seperti BOOKING, PAYMENT, dan CHECK\_IN pada kontroler booking, sementara Microprogramming memanfaatkan micro instruction untuk berbagai operasi mulai dari aritmatika, logika, hingga fitur tambahan lainnya dalam proyek.

## CHAPTER 3

### TESTING AND ANALYSIS

#### 3.1 TESTING

Program untuk Hotel Booking System ini terdiri dari FSM state yang berfungsi sebagai inti atau pusat kontrol, dan juga mencakup komponen seven segment yang digunakan untuk menampilkan informasi terkait harga sewa kamar dalam bentuk seven segment. Hotel Booking System ini memiliki total 8 input, 1 inout (port yang sudah dideklarasikan), dan 6 output. Input melibatkan sinyal-sinyal seperti CLK , start, reset, jml\_orang, jml\_malam, no\_kamar, serta input\_uang . Sementara itu, total\_harga digunakan sebagai inout port yang menggambarkan total biaya transaksi, dan kembalian, done, step, ratus\_ribuan, dan jutaan adalah output yang memberikan informasi mengenai hasil transaksi, status sistem, langkah proses, serta nilai-nilai untuk menampilkan harga di seven segment. Berikut ini adalah potongan kode bagian entity nya:

```
ENTITY HotelSystem IS
  PORT (
    CLK, start, besok : IN STD_LOGIC;
    reset : IN STD_LOGIC;
    -- jml_orang & jml_malam maksimum adalah 31
    jml_orang, jml_malam : IN STD_LOGIC_VECTOR (4 downto 0);
    -- jumlah kamar di hotel ada 32 - 1 = 31 kamar
    no_kamar : IN STD_LOGIC_VECTOR (4 downto 0);
    -- input_uang dlm satuan ratusan ribu. contoh: 0101 = 500k.
    input_uang : IN STD_LOGIC_VECTOR (13 downto 0);
    total_harga : INOUT STD_LOGIC_VECTOR (13 downto 0);
    kembalian : OUT STD_LOGIC_VECTOR (13 downto 0);
    done : OUT STD_LOGIC;
    -- menunjukkan current state
    step : OUT STD_LOGIC_VECTOR (2 downto 0);

    -- ada 2 sevseg untuk display harga 1 kamar
    ratus_ribuan : OUT STD_LOGIC_VECTOR (6 DOWNTO 0);
    jutaan : OUT STD_LOGIC_VECTOR (6 DOWNT0 0);

    -- menunjukkan reservasi keberapa
    test_out : OUT integer
  );
END ENTITY HotelSystem;
```

Gambar 2. Deklarasi entity Hotel System

Pada code di atas, digunakan port input clk, reset, dan start yang berfungsi sebagai kontrol dari program ini. CLK digunakan sebagai referensi waktu dengan sinyal rising edge, reset untuk mengembalikan state ke kondisi awal, dan start untuk memulai proses state. Selanjutnya, akan dimasukkan input berupa jumlah orang, jumlah malam, nomor kamar, dan input uang yang dideklarasikan sebagai "1111111111111" atau setara dengan Rp6.553.500.000. Hal ini dikarenakan input dan output uang yang ada dalam program ini direpresentasikan dalam satuan ratusan ribu. Untuk outputnya, terdapat enam sinyal, yaitu total\_harga untuk menyimpan nilai total biaya transaksi, kembalian untuk menyimpan nilai kembalian dari transaksi, done untuk menandakan bahwa proses booking telah berhasil, step untuk menunjukkan langkah state pada suatu waktu untuk membantu pemantauan proses, ratus\_ribuan untuk menyimpan nilai untuk menampilkan digit ratus ribuan pada seven segment, jutaan untuk menyimpan nilai untuk menampilkan digit jutaan pada seven segment, dan test\_out yang merupakan sinyal untuk menguji atau memantau keluaran tambahan yang mungkin diperlukan selama proses simulasi atau debugging. Kemudian untuk state pada program ini ada 6 state yang akan dihasilkan state machine viewer di Quartus. Berikut ini adalah potongan kode state yang ada pada modul HotelSystem.

```
-- deklarasi state
type stateType IS (idle, booking, loading, payment, check_in, booking_success, next_day);
```

Gambar 3. Deklarasi state FSM

Untuk melakukan pengetesan dari code-code tersebut, dapat digunakan metode yang disebut testbench. Testbench ini merupakan sekelompok kode yang memuat komponen dari file VHD yang ingin diuji, dan melalui testbench ini, kita dapat menyediakan input untuk menguji program serta mempermudah simulasi di ModelSim. Ciri khas dari testbench adalah tidak memiliki entity / entity nya kosong dan komponen akan berisi entity file VHD yang ingin diuji. Untuk program ini, kami membuat dua kode testbench yang telah kami buat, seperti pada modul HotelSystem\_tb untuk testbench dari HotelSystem. Dari testbench tersebut, kita dapat mengamati input dan output dari program, serta melihat current state dan next state dari program. Berikut adalah kode testbench dari HotelSystem\_tb:

```
library IEEE;
```

```

use IEEE.std_logic_1164.all;
use IEEE.numeric_std.all;

entity HotelSystem_tb is
end entity HotelSystem_tb;

architecture rtl of HotelSystem_tb is
    signal CLK, reset, besok : std_logic := '0';
    signal start : std_logic;
    signal jml_orang, jml_malam, no_kamar: std_logic_vector(4 downto 0)
:= (others => '0');
    signal input_uang, total_harga, kembalian : std_logic_vector(13
downto 0) := (others => '0');
    signal done : std_logic := '0';
    signal step : std_logic_vector (2 downto 0) := "000";
    signal ratus_ribuan, jutaan : std_logic_vector(6 downto 0);
    signal test_out : integer;

    COMPONENT HotelSystem
        PORT (
            CLK, start, besok, reset : IN STD_LOGIC;
            jml_orang, jml_malam : IN STD_LOGIC_VECTOR (4 downto 0);
            no_kamar : IN STD_LOGIC_VECTOR (4 downto 0);
            input_uang, total_harga : INOUT STD_LOGIC_VECTOR (13 downto
0);
            kembalian : OUT STD_LOGIC_VECTOR (13 downto 0);
            done : OUT STD_LOGIC;
            step : OUT STD_LOGIC_VECTOR (2 downto 0);
            ratus_ribuan, jutaan : OUT STD_LOGIC_VECTOR (6 DOWNT0 0);
            test_out : OUT integer
        );
    END COMPONENT HotelSystem;

    constant waktu : time := 10 ps;
    signal stimulus_done : boolean := false;
begin

```

```

UUT : HotelSystem port map (
    CLK => CLK,
    start => start,
    besok => besok,
    reset => reset,
    jml_orang => jml_orang,
    jml_malam => jml_malam,
    no_kamar => no_kamar,
    input_uang => input_uang,
    total_harga => total_harga,
    kembalian => kembalian,
    done => done,
    step => step,
    ratus_ribuan => ratus_ribuan,
    jutaan => jutaan,
    test_out => test_out
);

process
begin
    while not stimulus_done loop
        CLK <= '0';
        wait for waktu;
        CLK <= '1';
        wait for waktu;
    end loop;
    wait;
end process;

process
    variable counter : integer := 0;
begin
    start <= '1';

```

```

-----
-- Test case 1 : kasus di mana uang cukup --
-----

input_uang <= "11111111111111";
for i in 0 to 31 loop
    for j in 0 to 31 loop
        for k in 0 to 31 loop
            no_kamar <= STD_LOGIC_VECTOR(to_unsigned(k, 5));
            jml_malam <= STD_LOGIC_VECTOR(to_unsigned(j, 5));
            jml_orang <= STD_LOGIC_VECTOR(to_unsigned(i, 5));

            wait for waktu * 8;

            counter := counter + 1;
        end loop;
    end loop;
end loop;
reset <= '1';
wait for waktu;
reset <= '0';

-----

-- Test case 2: kasus untuk decreaseNight --
-----

besok <= '1';
wait for waktu * 8;

reset <= '1';
besok <= '0';
wait for waktu;
reset <= '0';
start <= '1';

-----

-- Test case 3 : kasus di mana uang tidak cukup --

```

```

-----
input_uang <= "0000000000000000";
for i in 0 to 31 loop
    for j in 0 to 31 loop
        for k in 0 to 31 loop
            no_kamar <= STD_LOGIC_VECTOR(to_unsigned(k, 5));
            jml_malam <= STD_LOGIC_VECTOR(to_unsigned(j, 5));
            jml_orang <= STD_LOGIC_VECTOR(to_unsigned(i, 5));

            wait for waktu * 8;

            counter := counter + 1;
            if counter = 32*32*32*2 then
                stimulus_done <= TRUE;
            end if;
        end loop;
    end loop;
end loop;
reset <= '1';
start <= '0';
wait;

end process;

end architecture rtl;

```

Gambar 3. Testbench HotelSystem

Selain testbench HotelSystem, ada pula sevseg\_modul\_tb untuk testbench dari sevseg\_modul. Testbench ini digunakan untuk menguji masing-masing input dan memastikan bahwa output yang dihasilkan sesuai dengan yang telah dideklarasikan. Berikut adalah kode testbench dari sevseg\_modul\_tb:

```

LIBRARY IEEE;
USE IEEE.std_logic_1164.ALL;
USE IEEE.numeric_std.ALL;

ENTITY SevSegModul_tb IS
END ENTITY SevSegModul_tb;

ARCHITECTURE rtl OF SevSegModul_tb IS
    COMPONENT SevSegModul IS
        PORT (
            sev_in : IN STD_LOGIC_VECTOR(3 DOWNTO 0);
            sev_out : OUT STD_LOGIC_VECTOR(6 DOWNTO 0)
        );
    END COMPONENT SevSegModul;

    SIGNAL input : STD_LOGIC_VECTOR(3 DOWNTO 0);
    SIGNAL output : STD_LOGIC_VECTOR(6 DOWNTO 0);
BEGIN
    UUT : SevSegModul PORT MAP(sev_in => input, sev_out => output);

    PROCESS
        CONSTANT waktu : TIME := 100 ps;
        VARIABLE counter : INTEGER := 0;
    BEGIN
        FOR i IN 0 TO 9 LOOP
            input <= STD_LOGIC_VECTOR(to_unsigned(counter, 4));
            counter := counter + 1;
            WAIT FOR waktu;
        END LOOP;
        WAIT;
    END PROCESS;
END ARCHITECTURE rtl;

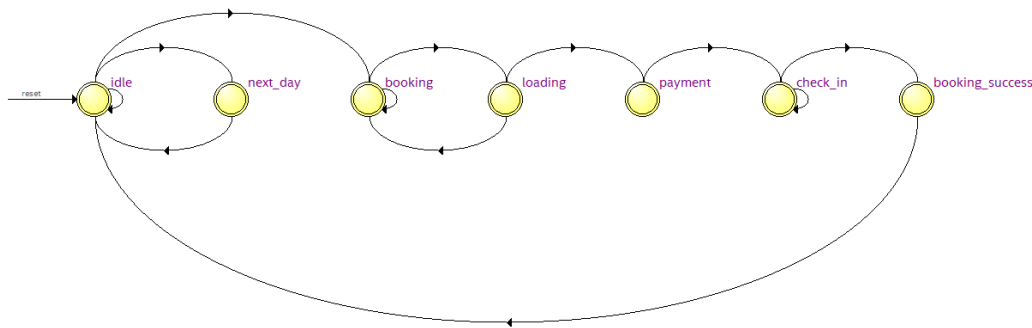
```

Gambar 4. Testbench 7 segment module



Hasil pengetesan kedua testbench ini melalui ModelSim akan ditunjukkan pada subbab berikutnya yaitu result. Dengan menggunakan testbench, kita dapat memverifikasi bahwa logika dan fungsionalitas dari modul-modul tersebut berjalan sebagaimana yang diharapkan, sehingga memastikan keandalan program dalam menjalankan tugasnya.

### 3.2 RESULT



Gambar 5. State diagram dari state machine dalam program

Gambar di atas merupakan hasil dari State Machine Viewer pada Quartus setelah melakukan sintesis. Dari visualisasi tersebut, dapat dilihat bahwa semua state saling berhubungan, dimulai dari state idle dan diakhiri pada booking\_success. Logika yang digunakan untuk State Machine ini dimulai dari state idle. Jika nilai start bernilai 1, maka akan menuju state booking. Jika nilai start bernilai 0, sistem akan kembali ke state idle.

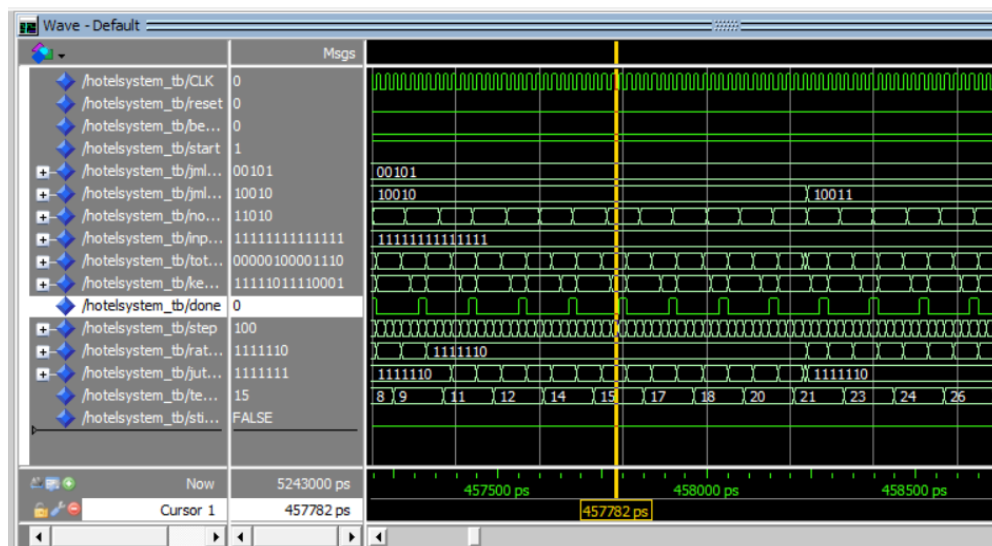
Selanjutnya, pada state booking, nilai isBooked masih false. Di sini, pengguna akan diminta untuk memasukkan nomor kamar, jumlah malam, dan jumlah orang. Jika input yang dimasukkan valid (lebih dari 0), maka sistem akan melanjutkan ke state loading. Namun, jika tidak valid, sistem akan tetap berada di state booking sampai input yang dimasukkan valid.

Pada state loading, sistem akan memeriksa ketersediaan kamar. Jika kamar tidak tersedia, nilai isBooked berubah menjadi TRUE, dan pesan laporan akan ditampilkan bahwa booking gagal. Namun, jika kamar tersedia, sistem akan menyimpan nomor kamar ke dalam daftar dan beralih ke state payment.

Selanjutnya, pada state payment, sistem akan menentukan total harga kamar berdasarkan nomor kamar, jumlah orang, dan jumlah malam. Harga kamar akan ditampilkan di seven segment bcd1 dan bcd2, setelah itu, sistem akan beralih ke state check\_in.

Pada state check\_in, pengguna akan diminta untuk memasukkan uang pembayaran. Jika jumlah uang cukup, maka check-in akan berhasil, dan sistem akan menuju state booking\_success. Namun, jika jumlah uang kurang, sistem akan kembali ke state check\_in.

Pada state booking\_success, nilai done diatur menjadi 1, dan sistem beralih ke state idle. Ini menandakan bahwa proses booking telah berhasil dan sistem siap menerima booking selanjutnya.



Gambar 6. Waveform simulasi HotelSystem\_tb

Gambar di atas adalah hasil simulasi dari HotelSystem\_tb. Pada simulasi ini, clock berjalan pada rising edge, tombol reset diatur ke 0, dan start diatur ke 1, menandakan bahwa program akan dimulai. Selanjutnya, input jumlah orang diatur menjadi 00010, menunjukkan bahwa terdapat 2 orang yang akan menginap. Input jumlah malam diatur menjadi 00011, yang artinya ada pesanan untuk menginap selama 3 malam. Input uang diatur sebagai "11111111111111". Setelah wave dijalankan, langkah-langkah berjalan sesuai dengan urutan dalam state machine.

Pada state 000 (idle), seven-segment masih menunjukkan nilai 0, dan pada state 001 (booking), input jumlah orang, jumlah malam, dan nomor kamar dicek apakah valid ( $> 0$ ). Jika valid, sistem berpindah ke state 010 (loading) untuk memeriksa ketersediaan kamar. Jika kamar sudah dipesan, akan muncul laporan bahwa booking gagal. Jika kamar tersedia, informasi kamar akan dimasukkan ke dalam daftarKamar. Setelah loading berhasil, sistem berpindah ke state 011 (payment), di mana total pembayaran dihitung.

Pada state 011 (payment) akan dihitung total pembayaran yang akan dibayar. Untuk output ratus ribuan dan jutaan akan keluar sesuai seven segmen yang berlangsung. seperti ketika di state idle, seven segment masih bernilai 0, yang pada modul seven segmen 0 menghasilkan output 1111110, kemudian ketika pada state payment, yang akan dihitung harga kamar dari no kamar yang dipilih. disini dipilih no kamar 00101 (dec 5) yang dimana sesuai syarat pada codingan, no kamar yang kurang dari 01011 (dec 11) akan menampilkan seven segmen ratusan bernilai 0101 (dec 5) dan jutaan bernilai 0000. Disini artinya jika nomor kamar yang dipilih berada di range 1-10 maka harga 500k untuk 2 orang. Untuk syarat no kamar 11-20 harga kamar adalah 800k orang untuk 3 orang dan untuk syarat no kamar 21 -31 harga kamar adalah 1.5 juta dengan kapasitas 4 orang per kamar. Jadi pada wave ini dapat dilihat karena yang dipesan nomor kamar 00101 (dec 5) maka sev seg ratusan yang keluar adalah 1011011 (dec 5). kemudian pada state berikutnya terdapat output total pembayaran yang disesuaikan dari jumlah orang (disini 2 orang), jumlah malam (3 malam) dan no kamar (ke 5), kemudian akan ditotalkan yaitu 00000000001111. kemudian berikutnya pada state 100 yaitu state check in, dimana disini kita memasukkan uang dan akan mengeluarkan jumlah kembalian. disini kembalian bernilai 1111111110000 (input uang - total harga). Karena jumlah uang sudah dibayar maka akan berhasil check in dan akan lanjut ke state 101 yaitu state booking\_success dimana disini akan merubah output done dari 0 menjadi 1.

Sealain itu, sinyal besok akan bertindak sebagai pemicu untuk memajukan simulasi ke hari berikutnya. Ini memungkinkan sistem untuk memperbarui status kamar yang dipesan, mengurangi malam yang tersisa, dan membersihkan reservasi yang telah berakhir. Pada testbench, sinyal besok akan menjadi '1' untuk mensimulasikan kondisi ketika sudah hari berikutnya dan mengamati perilaku sistem selama transisi dari idle ke next\_day.



Gambar di atas adalah hasil dari RTL viewer setelah melakukan sintesis pada program. Dalam gambar tersebut, terdapat banyak state yang digunakan, dan sebagai akibatnya, jumlah komponen yang muncul di RTL juga menjadi banyak. Meskipun sebenarnya ada 4 halaman RTL, namun hanya halaman 1 dan halaman ke-4 yang ditampilkan di sini.

Pada blok kuning yang terlihat di RTL, itu adalah representasi dari current state, dan kabel yang keluar dari current state ini menuju ke destinasi current state berikutnya. Blok persegi menunjukkan latch, yang berfungsi untuk menyimpan informasi yang telah diberikan. Adapun persegi yang berwarna hijau melambangkan komponen seven-segment, di mana seven-segment di bagian atas menampilkan ratusan, dan yang di bagian bawah menampilkan jutaan. Input dan output dari RTL viewer ini sesuai dengan input dan output pada port entity, yaitu terdapat 7 input dan 6 output.

### **3.3 ANALYSIS**

Pada testbench hotelsystem\_tb, terdapat berbagai state yang mencerminkan berbagai kondisi dan proses yang terjadi dalam sistem Hotel Booking. Saat program dalam keadaan siap dan menunggu input, sistem akan berada pada kondisi idle. Pada kondisi idle, pengguna diminta untuk memasukkan input berupa nomor kamar, jumlah malam, dan jumlah orang. Semua sinyal dan port pada saat ini diatur ke nilai 0.

Ketika input dimasukkan, sistem akan memproses input sesuai dengan kondisi state yang ada. Pada kondisi-kondisi tertentu, seven segment akan menampilkan informasi yang sesuai dengan input yang diberikan. Sebagai contoh, ketika nomor kamar yang dimasukkan berada dalam rentang 1-10, seven segment yang menunjukkan ratus ribuan akan memiliki nilai 5, yang mengindikasikan harga 500 ribu untuk tipe kamar dengan kapasitas 2 orang. Jika nomor kamar berada dalam rentang 11-20, nilai seven segment pada ratus ribuan akan menjadi 8, menandakan harga 800 ribu untuk tipe kamar dengan kapasitas 3 orang. Selanjutnya, jika nomor kamar berada dalam rentang 21-31, seven segment pada jutaan akan

menampilkan nilai 15, yang mengindikasikan harga 1,5 juta untuk tipe kamar dengan kapasitas 4 orang.

Pada state 000 (idle), seven-segment menampilkan nilai 0. Kemudian, pada state 001 (booking), sistem melakukan validasi input orang, malam, dan nomor kamar. Jika valid, proses berlanjut ke state 010 (loading) untuk memeriksa ketersediaan kamar. Setelah booking berhasil, sistem berpindah ke state 011 (payment), di mana total pembayaran dihitung dan harga kamar ditampilkan pada seven-segment. State 100 (check\_in) memproses pembayaran, menghitung kembalian, dan berpindah ke state 101 (booking\_success) jika pembayaran mencukupi.

Testbench untuk seven-segment menguji outputnya, memastikan bahwa setiap digit menghasilkan keluaran yang sesuai. Selanjutnya, hasil dari State Machine Viewer pada Quartus setelah melakukan sintesis menunjukkan bahwa semua state saling terhubung, dengan logika yang dimulai dari state idle dan kembali ke idle jika nilai start adalah 0.

Dalam analisis sintesis RTL Viewer, terlihat banyaknya state yang direpresentasikan oleh blok kuning, dan komponen-komponen seperti latch untuk menyimpan informasi dan blok hijau sebagai komponen seven-segment. Seven-segment ini digunakan untuk menampilkan nilai ratusan dan jutaan.

Secara keseluruhan, simulasi dan sintesis sistem Hotel Booking System ini berhasil mencapai state booking\_success dan menunjukkan pembentukan komponen yang sesuai pada RTL Viewer. Dengan penggunaan seven-segment, sistem dapat memvisualisasikan informasi harga kamar dan output lainnya dengan jelas selama simulasi.

## **CHAPTER 4**

### **CONCLUSION**

Proyek Hotel Booking System ini secara keseluruhan dirancang untuk mengotomatiskan dan mengelola proses pemesanan kamar hotel. Sistem ini menggunakan state machine sebagai dasar pengelolaan alur kerja, dengan beberapa state seperti idle, booking, loading, payment, check-in, dan booking success. Pada kondisi idle, pengguna diminta untuk memasukkan informasi seperti nomor kamar, jumlah malam, dan jumlah orang. Logika state machine kemudian memproses input tersebut, melakukan verifikasi, dan mengarahkan sistem ke state berikutnya.

Komponen seven-segment digunakan sebagai antarmuka untuk menampilkan informasi kepada pengguna, termasuk harga kamar dan output lainnya. Setiap digit seven-segment dikendalikan oleh entitas terpisah yang mengonversi input ke format yang dapat ditampilkan pada tujuh segmen tersebut. Proyek ini melibatkan berbagai macam input, seperti jumlah orang, jumlah malam, nomor kamar, dan input uang, yang kemudian diolah oleh state machine untuk menghasilkan output yang diinginkan.

Simulasi menggunakan testbench memberikan pemahaman visual tentang bagaimana sistem berperilaku dalam berbagai kondisi dan input yang mungkin. Selain itu, RTL Viewer dan sintesis memberikan pandangan yang lebih mendalam tentang struktur logika dan hubungan antar state dalam proyek ini. Proses sintesis memvisualisasikan hubungan antar komponen dan kabel-kabel yang menghubungkan state, memperlihatkan kompleksitas dari implementasi keseluruhan.

Analisis terhadap proyek ini menunjukkan bahwa sistem berhasil mengimplementasikan fungsi dasar pemesanan hotel dengan baik, dengan logika program

yang memadai seperti pembagian harga berdasarkan nomor kamar, jumlah orang, dan jumlah malam. Keseluruhan proyek memberikan gambaran yang komprehensif tentang bagaimana aplikasi pemesanan hotel dapat diwujudkan dalam konteks FPGA.

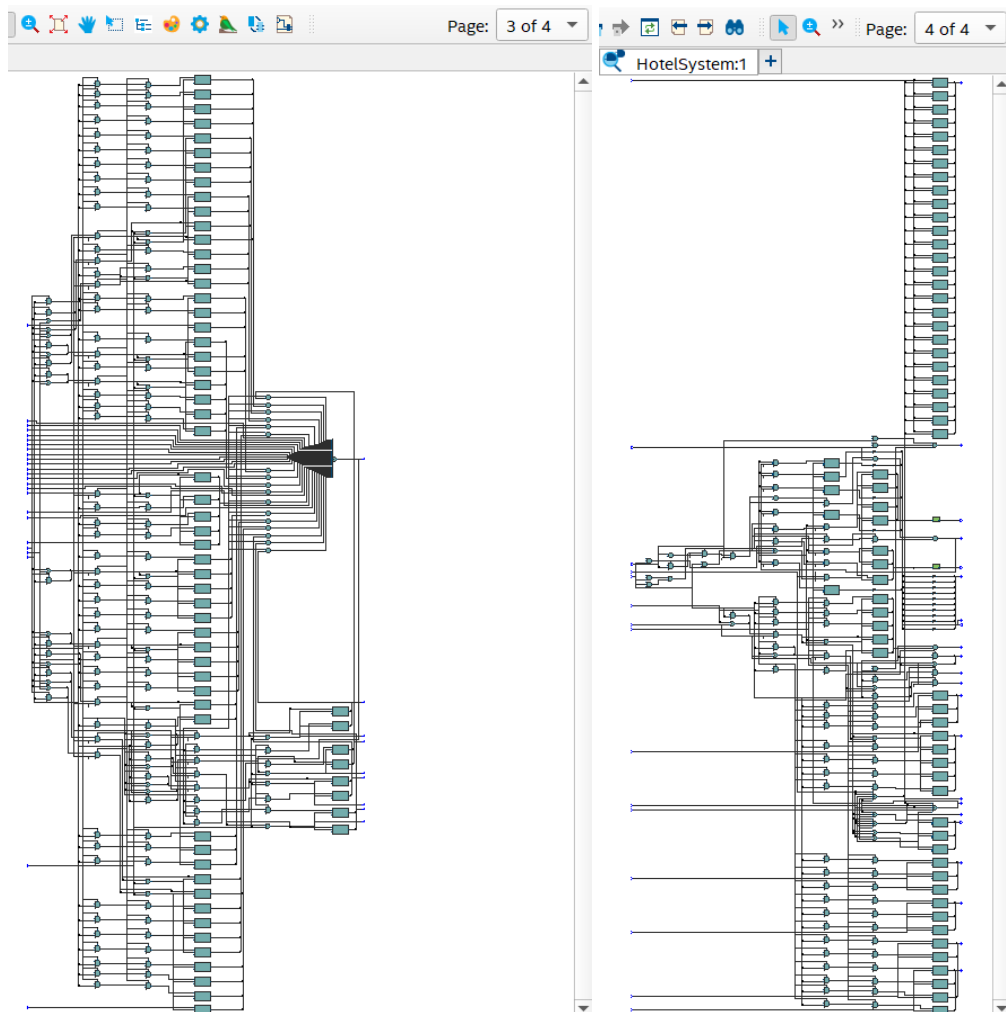
## REFERENCES

- [1] C. Brinza, "The Latest Global Hospitality Statistics", TrustYou, (9 Nov. 2023) [Online]. Available: <https://www.trustyou.com/blog/research/global-hospitality-statistics-2023>.
- [2] "Hotel Booking Engine Market Size In 2023 : Growth Opportunities and Future Outlook 2030", (no date.) [Online]. Available: <https://www.linkedin.com/pulse/hotel-booking-engine-market-size-2023-growth-opportunities>.
- [3] Tim Asisten Digilab 2023, Modul 2 hingga 9, in Modul Praktikum Perancangan Sistem Digital, Laboratorium Digilab, Universitas Indonesia, 2023
- [4] D. Williams, "Implementing a Finite State Machine in VHDL", (23 Dec. 2015) [Online]. Available: <https://www.allaboutcircuits.com/technical-articles/implementing-a-finite-state-machine-in-vhdl/>.
- [5] Ltd., Arm. "What is FPGA?" Arm | The Architecture for the Digital World. Accessed December 24, 2023. <https://www.arm.com/glossary/fpga>.



## APPENDICES

### Appendix A: Project Schematic



## Appendix B: Documentation

