



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

**IOT REMOTE WEATHER STATION AND
CENTRALIZED CONTROL CENTER**

GROUP PA18

Javana Muhammad Dzaki	2306161826
Benedict Aurelius	2306209095
Syahmi Hamdani	2306220532
Muhamad Rey Kafaka Fadlan	2306250573

PREFACE

Puji syukur kehadirat Tuhan Yang Maha Esa, atas segala rahmat dan karunia-Nya sehingga proyek "IoT Remote Weather Station & Centralized Control Center" ini dapat diselesaikan dengan baik. Proyek ini dirancang sebagai solusi berbasis teknologi Internet of Things (IoT) yang mampu memberikan kemudahan dalam pengamatan dan pengelolaan data cuaca secara efisien dan akurat.

Proyek ini mengintegrasikan berbagai sensor seperti sensor kelembaban, suhu, dan intensitas cahaya untuk membaca data lingkungan secara real-time. Data yang diperoleh diproses menggunakan sistem berbasis *Finite State Machine* (FSM) sebelum dikirim dalam format ke pusat kontrol. Selanjutnya, hasil pengamatan secara otomatis dicatat dalam format CSV, sehingga mempermudah pengolahan dan analisis data.

Proyek ini diharapkan dapat memberikan manfaat, baik sebagai langkah awal dalam pengembangan sistem IoT sebagai referensi untuk proyek serupa di masa mendatang. Semoga proyek ini dapat memberikan manfaat dan menjadi kontribusi yang positif dalam pengembangan teknologi IoT di berbagai bidang.

Depok, 7 Desember, 2024

Group PA18

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
- 2.3 Schematics
- 2.4 RTL Designs

CHAPTER 3: TESTING AND ANALYSIS

- 3.1 Testing
- 3.2 Result - Weather Station
- 3.3 Result - Station Controller
- 3.4 Result - Control Center
- 3.5 Result - Control Center Reporter (TB)
- 3.6 Analysis

CHAPTER 4: CONCLUSION

REFERENCES

APPENDICES

- Appendix A: Project Progress & Documentation

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

Indonesia, sebagai negara kepulauan yang berada pada Cincin Api Pasifik, sangat rentan terhadap berbagai bencana alam seperti banjir, tanah longsor, gempa bumi, dan badai. Berdasarkan data, hingga september 2024, telah terjadi 1.300 bencana alam di indonesia. Tingginya jumlah bencana ini menekankan perlunya sistem yang lebih canggih dan responsif untuk memantau serta mengelola bencana guna mengurangi dampak yang ditimbulkan. Dalam hal ini, teknologi memainkan peran yang penting dalam meningkatkan kesiapan dan mitigasi bencana.

Pengembangan Weather Station yang berbasis IoT yang terintegrasi dengan control center. Teknologi ini memberikan kemampuan untuk memantau secara real-time dengan tingkat akurasi yang cukup tinggi, sehingga dapat mendukung upaya untuk memanajemen bencana yang lebih efektif dan efisien di berbagai wilayah.

Weather Station dapat dilengkapi dengan berbagai sensor untuk mendeteksi parameter seperti suhu, kelembapan, dan intensitas cahaya. Dengan IoT, data dari stasiun ini dapat ditransmisikan secara langsung ke sistem pusat. Hal ini memungkinkan pemantauan kondisi cuaca secara berkelanjutan.

Sementara itu, control center berfungsi sebagai pusat pengelolaan data dan pengambilan kesimpulan. Control center dapat mendeteksi pola cuaca, memprediksi bencana, dan memberikan peringatan yang lebih awal.

Untuk menghadapi 1.300 bencana yang tercatat pada 2024, sistem yang mampu menyediakan data secara cepat dan akurat sangat dibutuhkan. Seperti, deteksi yang lebih awal untuk peningkatan curah hujan dapat membantu mencegah terjadinya banjir dan lainnya. Di sisi lain, data historis yang dikumpulkan melalui weather station IoT juga dapat digunakan untuk meningkatkan akurasi prediksi bencana, yang berguna untuk upaya mitigasi bencana dan lainnya.

Dengan penerapan teknologi berbasis IoT dan sistem kendali terpusat, Indonesia dapat membangun ekosistem manajemen bencana yang lebih kuat.

Pendekatan ini tidak hanya membantu mengurangi risiko dan kerugian ekonomi, tetapi juga menyelamatkan lebih banyak nyawa dalam menghadapi tantangan bencana di masa depan.

1.2 PROJECT DESCRIPTION

Proyek Weather Station IoT dirancang untuk memonitor parameter cuaca seperti kelembapan (humidity), suhu (temperature), dan intensitas cahaya (daylight). Sensor kelembaban akan menghitung tingkat kelembapan udara dari hasil bacaan massa air dan volume yang dihasilkan. Sensor suhu akan mengukur suhu lingkungan dalam Celcius. Sensor daylight akan memastikan apakah sedang siang atau malam. Data ini dikumpulkan secara otomatis melalui sensor, diproses, dan dikirimkan ke pusat kontrol menggunakan mekanisme Finite State Machine (FSM). Sistem ini memiliki kemampuan membaca data dari file I/O, mengolahnya, dan menyimpan hasil observasi dalam format CSV untuk mempermudah analisis dan pelaporan.

1.3 OBJECTIVES

Objektif dari proyek ini adalah sebagai berikut:

1. Mengembangkan Sistem pemantauan cuaca berbasis FSM.
2. Mengembangkan mekanisme encoding yang mampu mengkonversi data sensor menjadi paket data 64-bit.
3. Merancang sistem yang dapat membaca dan mengeksekusi instruksi dari pusat kontrol menggunakan port instruksi.
4. Mengimplementasikan teori dan konsep perancangan sistem digital untuk menyelesaikan real-world problems

1.4 ROLES AND RESPONSIBILITIES

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

Roles	Responsibilities	Person

Role 1	Membuat Opcode, Menentukan Format Package Encoding, Membuat Weather Station, PPT	Javana Muhammad Dzaki
Role 2	Membuat Station Controller dan Metode CSV Reading, PPT	Benedict Aurelius
Role 3	Membuat Control Center dan Control Center Testbench, Control Center Report, PPT	Syahmi Hamdani
Role 4	Membuat Laporan dan Readme.md, PPT	Muhamad Rey Kafaka Fadlan

Table 1. Roles and Responsibilities

CHAPTER 2

IMPLEMENTATION

2.1 EQUIPMENT

Alat yang akan kami gunakan dalam project ini adalah:

- VHDL (VHSIC Hardware Description Language)
- ModelSim
- Github:
- Quartus Prime 21.1
- Visual Studio Code

Project Repository

<https://github.com/javendzk/FinalProjectPSD-PA18>

2.2 IMPLEMENTATION

Komponen Utama:

- Control Center, yaitu komponen yang bertugas sebagai pusat monitoring station controller 1 sampai dengan 3 dan mampu memberikan instruksi untuk station. Sebagai kontrol pusat, control center akan men-decode package report yang diterima dari station controller, diteruskan pada viewing signal, dan dilaporkan pada final report CSV. Pengontrol Control Center yang berfungsi menerima instruction dan reporting adalah Control Center Reporter, dimana akan menjadi run-all testbench untuk memantau pergerakan sebuah control center. Kemudian pada awal pengembangan, kami sembat membuat penguji Control Center bernama Control Center Testbench untuk memastikan Control Center memiliki hasil yang sesuai dengan guideline yang ditetapkan. Control Center menggunakan sistem decoding packet report queue-dequeue untuk menangani packet report flow yang masuk secara bersamaan.
- Terdapat tiga Entity yang mengontrol Station, yaitu station_controller_1, station_controller_2, dan station_controller_3. Setiap station bertugas untuk

membaca data sensor dari CSV (sensor_1, sensor_2, dan sensor_3). Lalu data ini kemudian diteruskan ke Weather_Station untuk diproses.

- Weather Station, yaitu komponen yang berfungsi sebagai state machine yang memproses data dari sensor. Weather station meng-encode data sensor, memproses instruksi operasional, dan menghasilkan paket laporan yang mencakup informasi seperti sumber data, status sistem, opcode, timestamp, serta data sensor.

Implementasi Pada Setiap Modul

1. Behavioral Style Modelling

Project ini menerapkan modul Behavioral Style Modelling untuk semua komponennya, yang berinteraksi dan membentuk sistem pelaporan cuaca yang terpusat. Beberapa penggunaan behavioral style antara lain:

- Membaca data sensor dari file CSV.
- Memproses dan mengonversikan pembacaan sensor.
- Menghasilkan laporan berdasarkan instruksi, dan
- Mengelola aliran data dan sinkronisasi antar komponen

2. Finite State Machine (FSM)

Pada Salah satu komponen bernama weather_station, terdapat FSM yang terdiri dari empat state utama:

- Idle_State : Keadaan default ketidak ada instruksi yang aktif.
- Read_Inst : Membaca dan menginterpretasikan instruksi yang masuk.
- Encode : Menyiapkan data sensor untuk laporan.
- Generate_Report : Membuat laporan dalam format paket 64-bit.

FSM ini mengelola dekode instruksi, aktivitas sensor, pengkodean data, dan meng-generate laporan.

3. Looping

- Proses membaca file secara baris ke baris lainnya.
- Transisi state berbasis clock.
- Penghitungan Cycle untuk sinkronisasi.
- Kemajuan state berdasarkan kondisi tertentu

4. Function

Terdapat beberapa function yang diimplementasikan, sebagai contoh:

- Funsi `is_valid_opcode()` : Memvalidasi opcode instruksi agar berada dalam rentang tertentu.
- Fungsi `calculated_kelembapan()` : Menghitung kelembaban dengan memastikan tidak ada pembagian dengan nol.
- Fungsi `encode_sensor_data()` : Mengkodekan data sensor berdasarkan tipe, status, dan data.

5. Structural Architecture

- Control Center Report : Menjalankan Control Center dan menulis pada final CSV Report
- Control Center : Pusat kontrol sentral yang membawahi 3 controller melalui proses instantiasi.
- Individual Controllers : tiga komponen `station_controller_1`, `station_controller_2`, dan `station_controller_3` yang meng-instantiate Weather Station dan mengendalikannya.
- Weather_Station : Komponen pemrosesan data dan packaging report.
- Semua komponen diintegrasikan melalui port mapping

6. Testbench

- Menyediakan interface input dan output yang jelas dari sebuah kode.
- Penanganan instruksi yang sudah terdefinisi
- Modularitas mencakup komponen seperti Individual Controller seperti `station_controller_1`, `station_controller_2`, dan `station_controller_3`, yang dapat diuji satu-persatu untuk memastikan fungsionalitasnya.
- Control Center testbench berfungsi untuk memastikan kinerja Control Center, serta menjalankan Control Center dengan Otomatis

2.3 SCHEMATICS

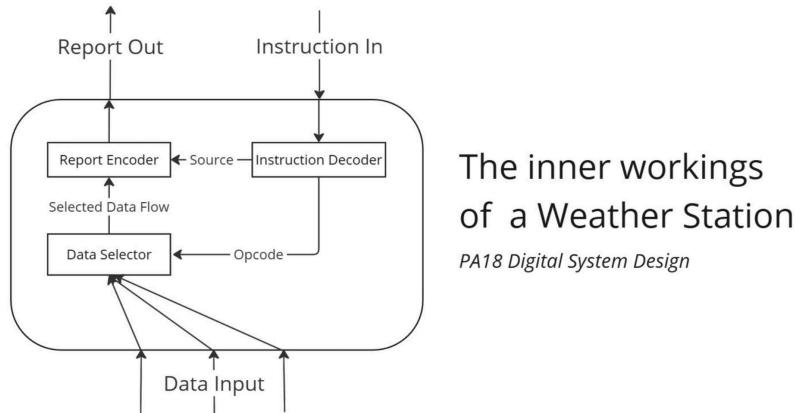


Fig. 1. Weather Station Schematic

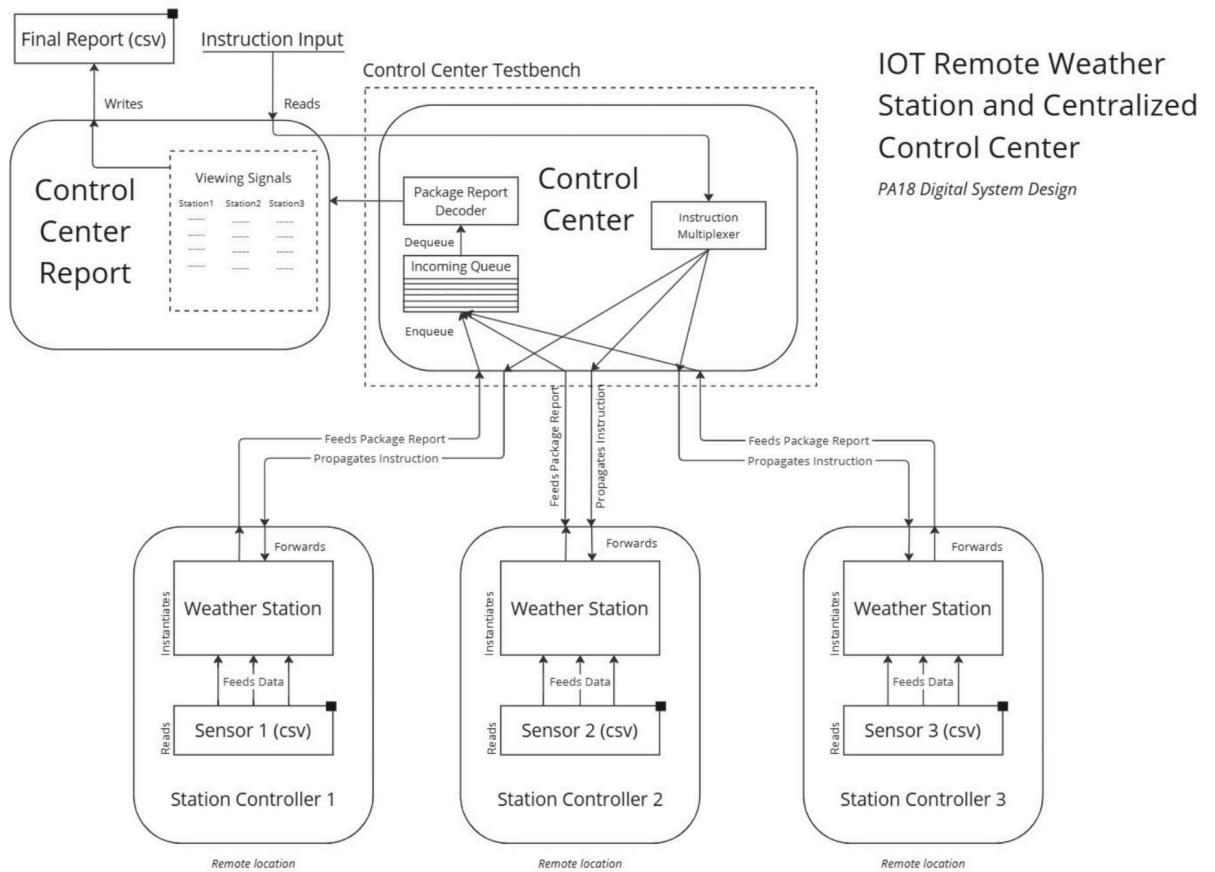


Fig. 2. System Schematic

2.4 RTL DESIGNS

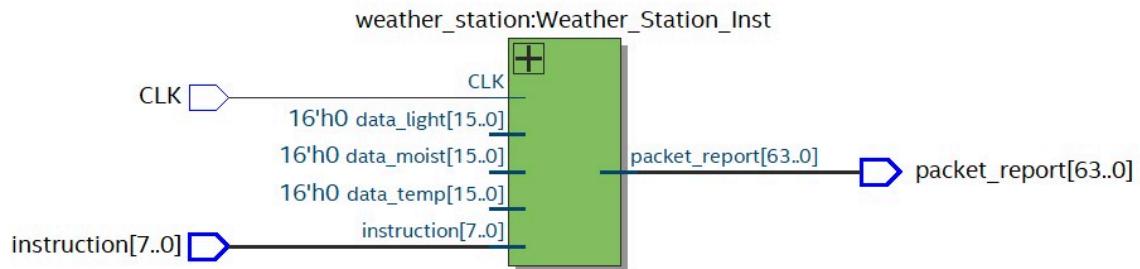


Fig. 3. Station Controller RTL Design

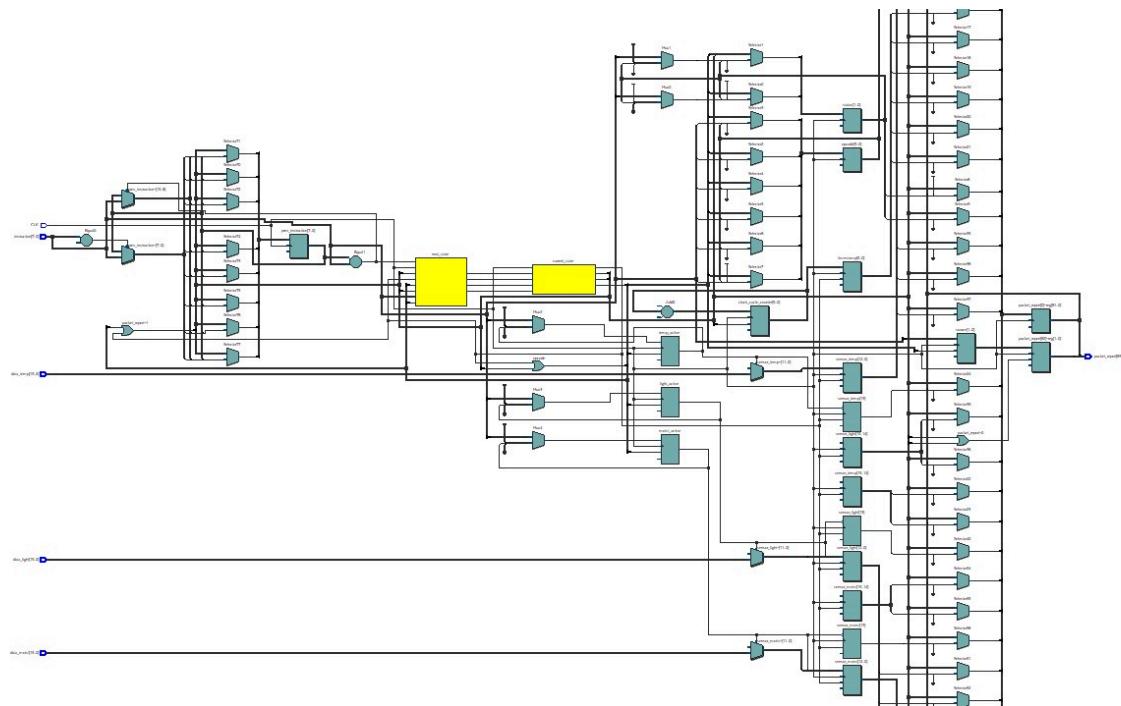


Fig. 4. Weather Station RTL Design

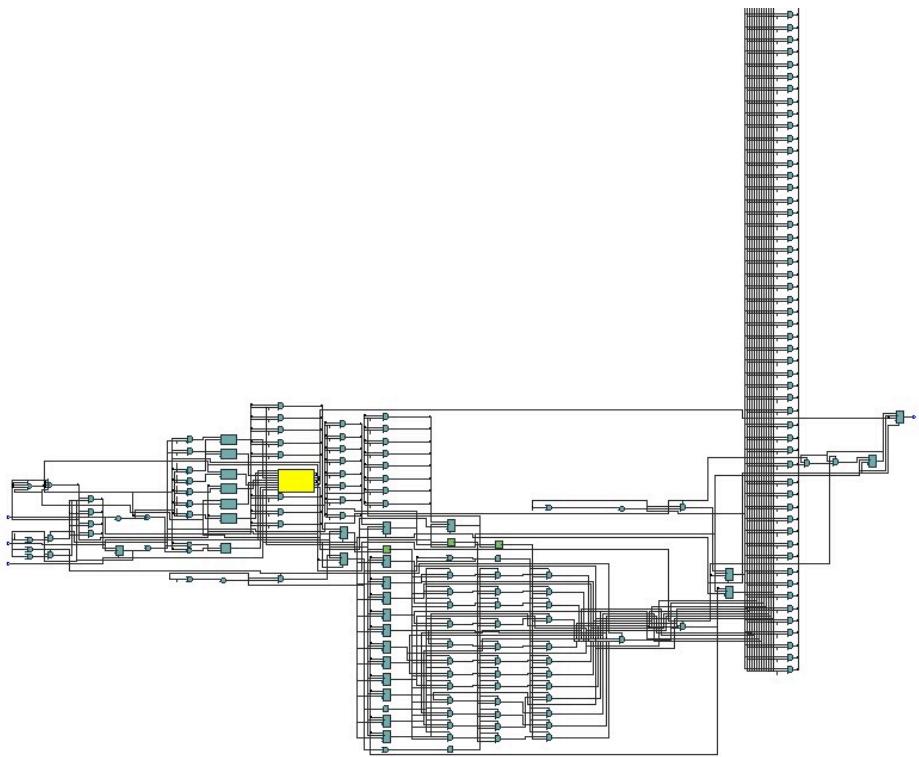


Fig. 5. Control Center RTL Design

CHAPTER 3

TESTING AND ANALYSIS

3.1 TESTING

Setelah mendesain dan mengimplementasikan selesai, selanjutnya adalah testing. Dalam proyek kali ini, Testing menggunakan 3 tipe, yaitu wave test, FSM, dan CSV. Wave test adalah metode pengujian yang umum digunakan dalam desain digital, FSM digunakan untuk memastikan status operasi komponen berjalan dengan benar, dan CSV adalah format data dalam basis data di mana setiap entri dipisahkan oleh koma (,) atau titik koma (;). Format ini mudah digunakan dan dapat dibuka dengan berbagai editor teks seperti Notepad, Wordpad, atau Microsoft Excel.

3.2 RESULT - WEATHER STATION

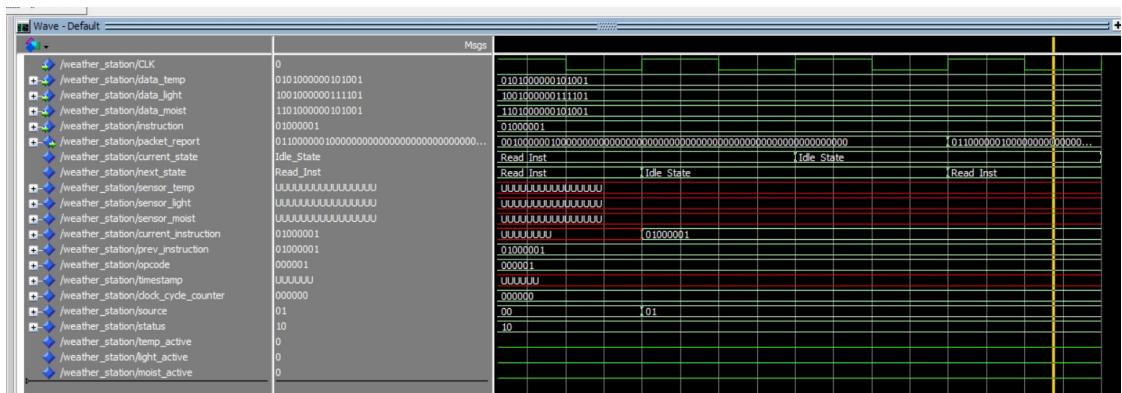


Fig. 6. Weather Station Startup & Instruction 01000001

Pada awal dijalankan, weather station akan masuk dalam mode Idle. Instruction yang diberikan adalah 01000001 (Source: 01, Opcode: 000001 Idle). Sehingga terjadi Read-Idle loop pada weather station, menunggu instruksi berikutnya

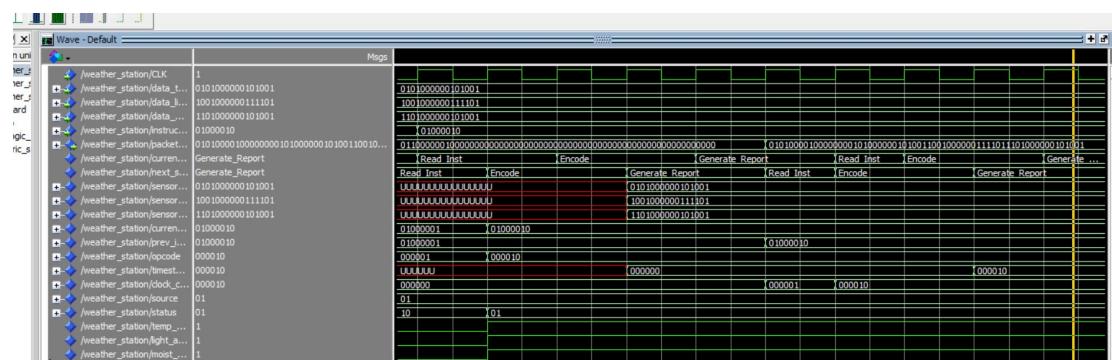


Fig. 7. Instruction 01000010

Setelah diberikan instruction 01000010 (Source: 01, Opcode: 000010 Run all sensor), weather station mulai membaca port semua sensor dan melakukan encoding serta delivery packet report 64-bit. Selain itu, Mengambil “Source” sebagai identitas station dan meng-increment refresh cycle count

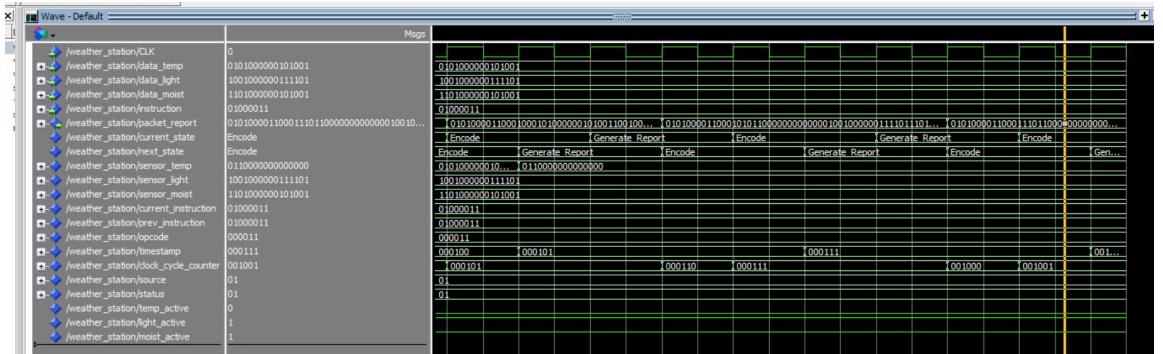


Fig. 8. Instruction 01000011

Saat diberikan instruction 01000011 (Source: 01, Opcode: 000011 Run without temp), Pelaporan package report berjalan seperti biasa, namun bisa dilihat data section 12-bit temp rata 0 dan statusnya 10 (stopped)

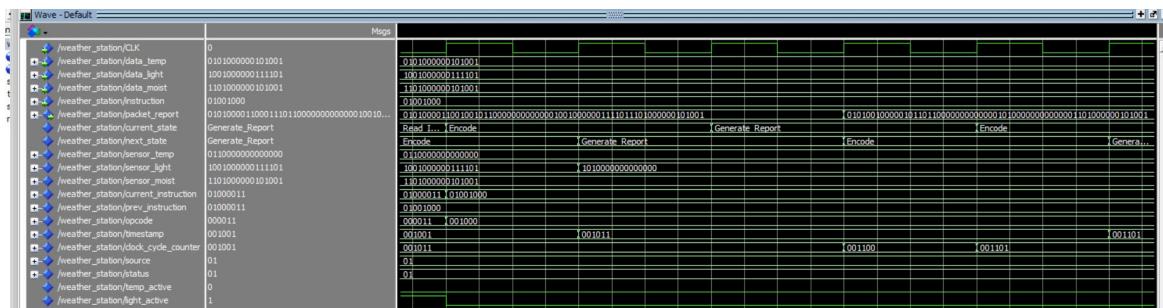


Fig. 9. Instruction 01001000

Saat diberikan instruction 01001000 (Source: 01, Opcode: 001000 Run only moist) bisa dilihat proses read inst, perubahan opcode, dan output report package tetap sama. Namun, sensor yang aktif dan memiliki data pada packet report hanya temp, sedangkan sensor lain nonaktif dan data sectionnya rata 0.

3.3 RESULT - STATION CONTROLLER

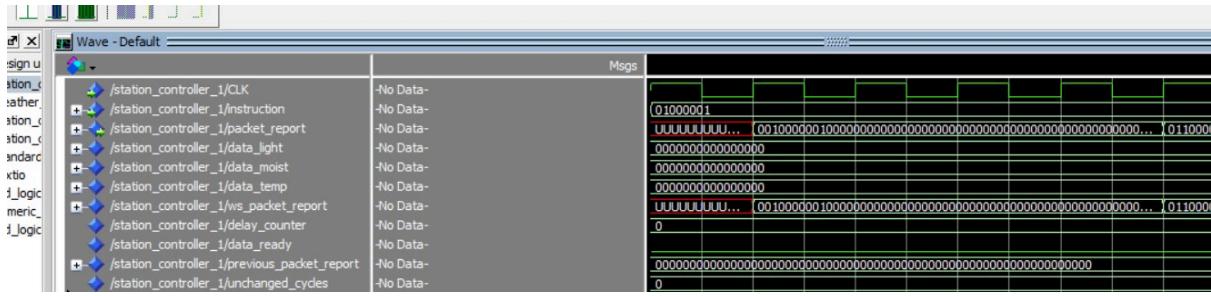


Fig. 10. Instruction 01000001

Pada awal dijalankan, weather station di instantiasi oleh station controller dan di port mapping. Ketika instruction yang diberikan 01000001 (Source: 01, Opcode: 000001 Idle), Instruksi akan di propagate menuju weather station dan dari controller sendiri akan menunda pembacaan CSV

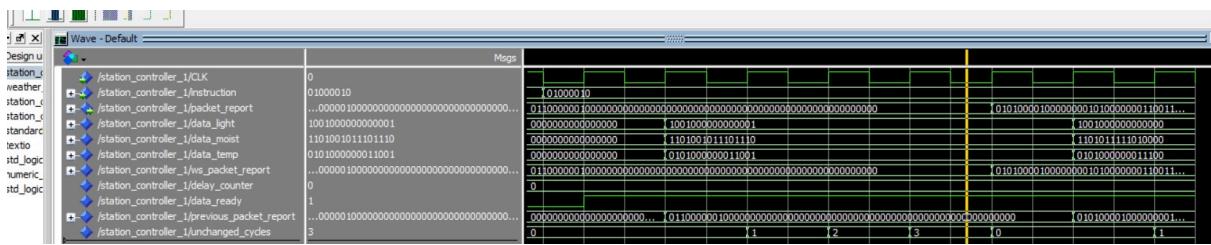


Fig. 11. Instruction 01000010

Saat controller diberikan instruction 01000010 (Source: 01, Opcode: 000010 Run all), Controller akan membaca file CSV dan diberikan kepada weather station. kemudian, weather station akan meng-encode sesuai opcode dan generate package report

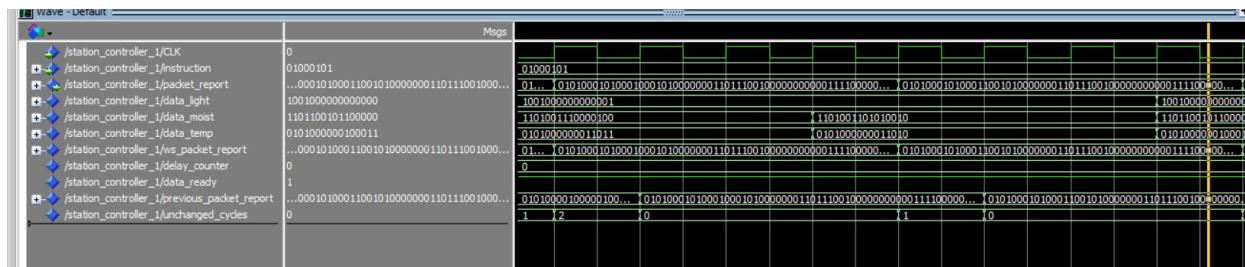


Fig. 12. Instruction 01000101

Saat controller diberikan instruction 01000101 (Source: 01, Opcode: 01000101 Run but stop moist), Controller akan membaca file CSV dan diberikan kepada weather station. Namun, sensor moist akan di stop dengan status 10 dan data section moist

menjadi rata 0. Terlihat pula sinkronisasi pembacaan sensor CSV dengan jeda 3 clock setiap report berubah

3.4 RESULT - CONTROL CENTER

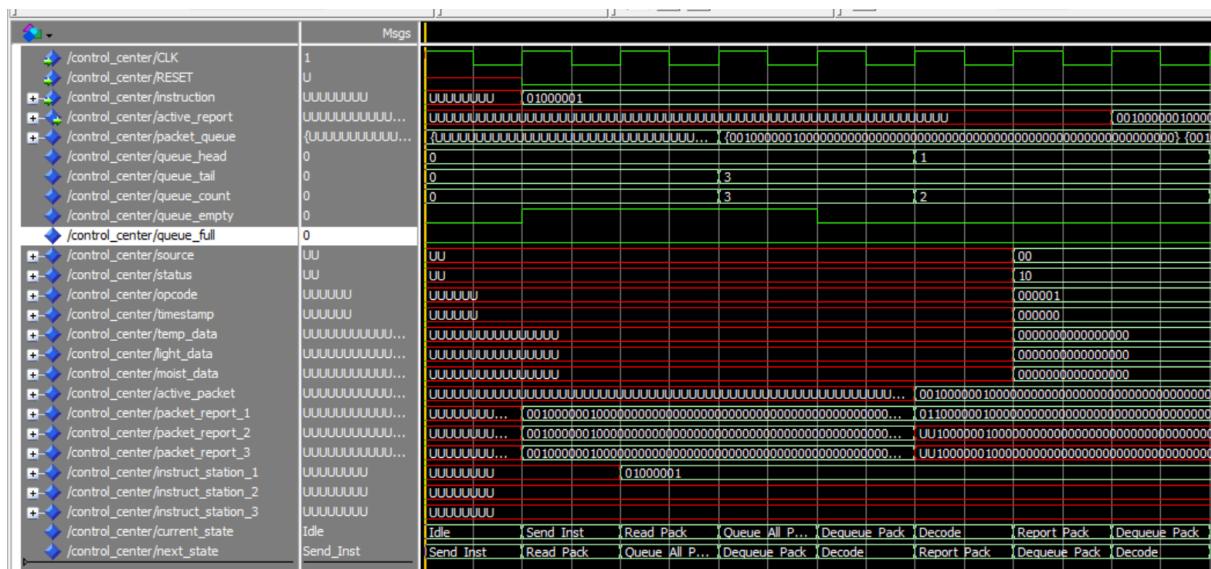


Fig. 13. Instruction 01000001

Saat Control Center dimulai, statenya adalah idle. Ketika instruction diterima dari reporter/testbench, maka state akan berpindah ke state send inst (memberikan instruksi ke weather station dan controller). Kemudian, setelah weather station memberikan paketnya, weather station akan mulai membaca packet dan menyimpannya dalam queue.

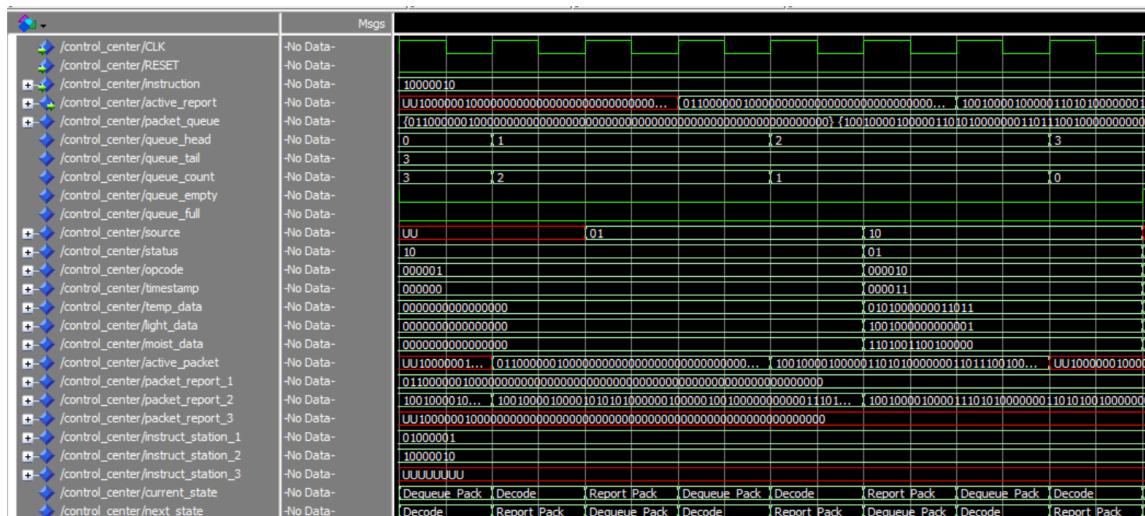


Fig. 14. Instruction 10000010

Control center memberikan instruksi 10000010, di mana station 2 akan melakukan run dan report semua sensornya. Disini setelah instruksi diberikan, maka paket akan dibaca dan disimpan dalam queue, sebelum didequeue, decode, dan siap untuk direport.

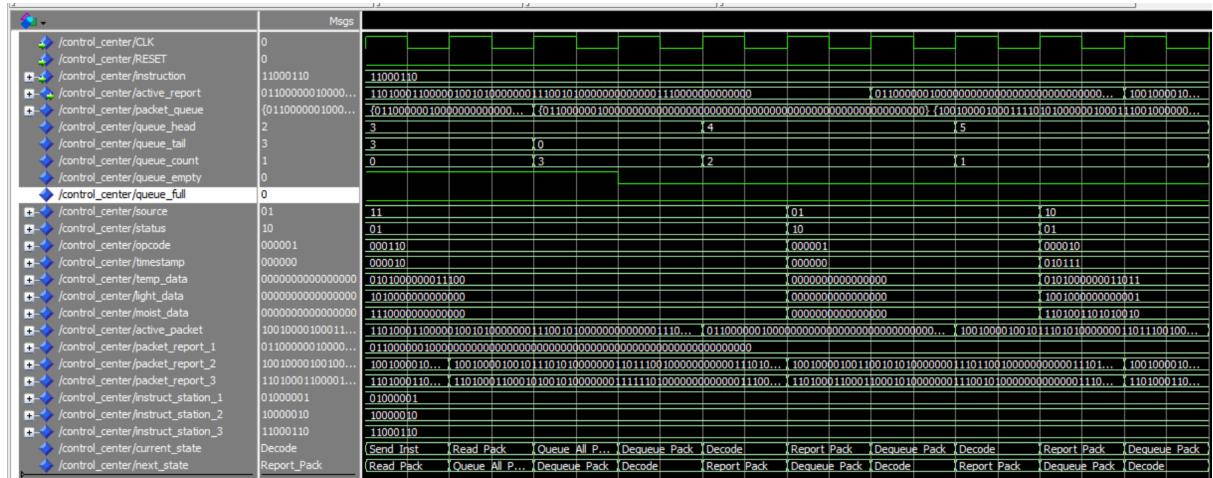


Fig. 15. Instruction 11000110

Control Center kembali memberikan instruksi. Kali ini untuk station 3, dengan opcode 11000110, dimana hanya sensor temp yang akan aktif. Setelah instruksi diberikan, weather station akan mengenerate paket yang akan diterima oleh control center. Paket itu akan dibaca dan dimasukkan dalam queue, sebelum didequeue, decode, dan siap untuk direport ke file CSV.

3.5 RESULT - CONTROL CENTER REPORTER (TB)

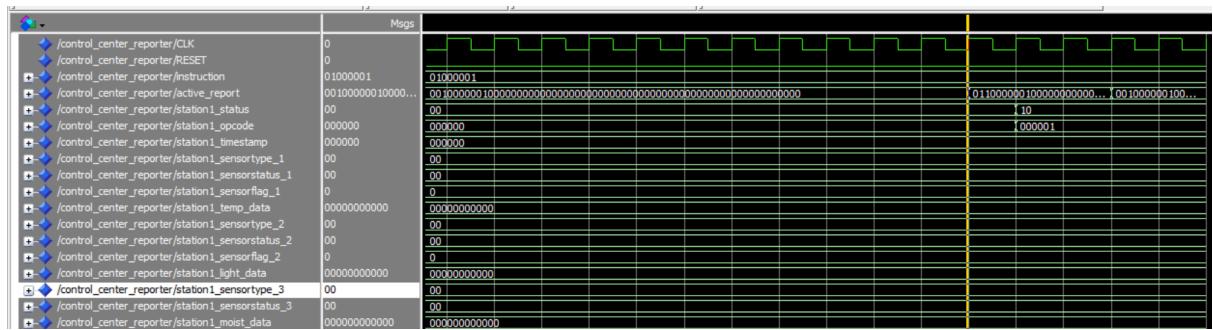


Fig. 16. Instruction 01000001

Saat memulai reporter (proses testbench), Sinyal pada control center reporter belum diupdate. Setelah diberikan instruction 01000001 (Source: Station 1, Opcode: 000001, Go Idle), maka station 1 akan diberikan state idle dan akan tetap idle selama belum ada instruksi dengan opcode run. Selain itu, sinyal pada reporter juga di update menjadi status 10 (Idle).

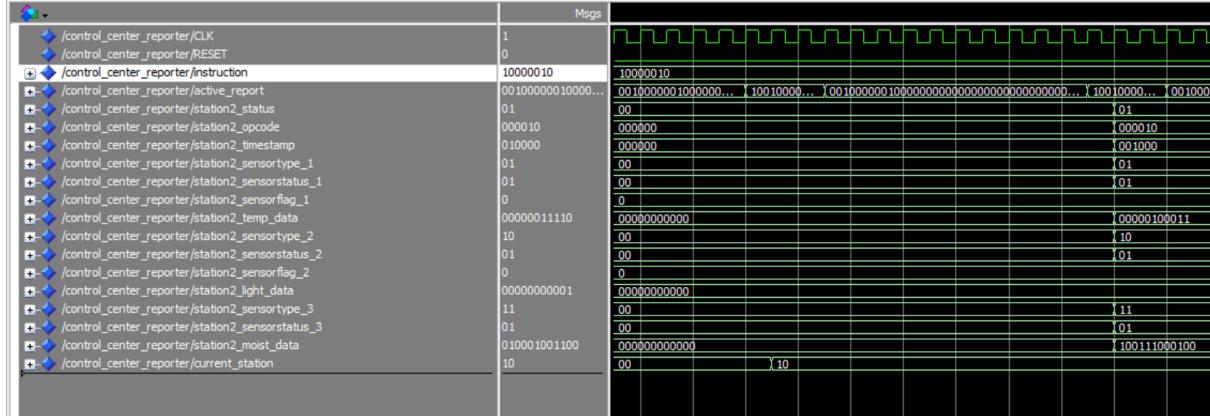


Fig. 17. Instruction 10000010

Setelah dimasukkan instruction 10000010 (Source: Station 2, Opcode: 000010, Run All Sensors), maka station 2 akan melakukan run all sensors. Sinyal pada reporter juga akan berubah. Terlihat bahwa active report tetap berganti-ganti. Ini karena control center akan secara bergiliran meletakkan report dari station 1-3 ke dalam queue, dan akan dilakukan dequeue di mana report yang didequeue akan menjadi active report.

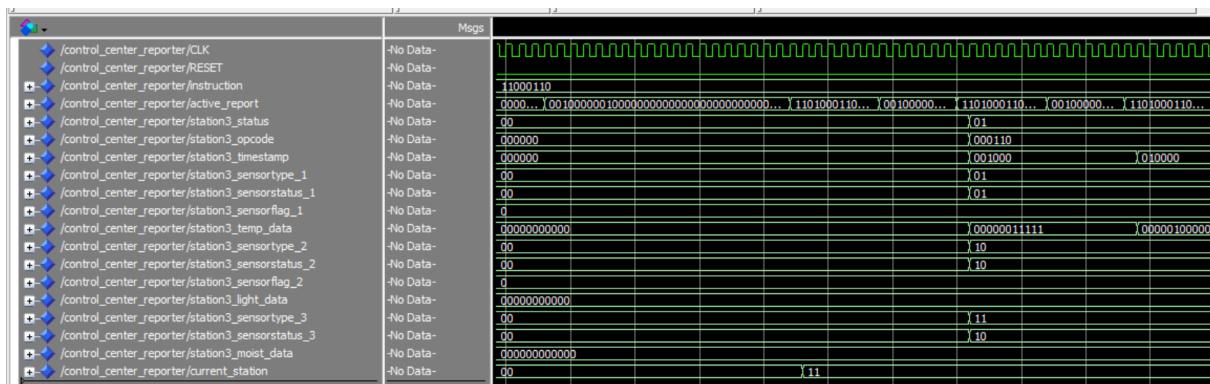


Fig. 18. Instruction 11000110

Setelah dimasukkan instruction 11000110 (Source: Station 3, Opcode: 000110, Run Temp Only), Station 2 akan melakukan run temp only dan generate report dengan

hanya sinyal temperatur yang aktif. Sama seperti sebelumnya, active report terus berganti bergiliran dari report station 1-3, dengan sistem queue dan dequeue.

	Source	Status	Opcode	Timestamp	SensorType1	SensorStatus1	SensorFlag1	Temp Data	SensorType2	SensorStatus2	SensorFlag2	Light Data	SensorType3	SensorStatus3	Moisture Data
1	1	Idle	1	0	Unknown	No Status	0	0	Unknown	No Status	0	Night	Unknown	No Status	0
2	2	Running	2	2	Temperature	Enabled	0	28	Light	Enabled	0	Night	Moisture	Enabled	850
4	1	Idle	1	0	Unknown	No Status	0	0	Unknown	No Status	0	Night	Unknown	No Status	0
5	2	Running	2	8	Temperature	Enabled	0	35	Light	Enabled	0	Night	Moisture	Enabled	2500
6	3	Running	6	2	Temperature	Enabled	0	28	Light	Disabled	0	Night	Moisture	Disabled	0
7	1	Idle	1	0	Unknown	No Status	0	0	Unknown	No Status	0	Night	Unknown	No Status	0
8	2	Running	2	16	Temperature	Enabled	0	30	Light	Enabled	0	Day	Moisture	Enabled	1100
9	3	Running	6	8	Temperature	Enabled	0	31	Light	Disabled	0	Night	Moisture	Disabled	0
10	1	Idle	1	0	Unknown	No Status	0	0	Unknown	No Status	0	Night	Unknown	No Status	0
11	2	Running	2	22	Temperature	Enabled	0	26	Light	Enabled	0	Night	Moisture	Enabled	1900
12	3	Running	6	16	Temperature	Enabled	0	32	Light	Disabled	0	Night	Moisture	Disabled	0
13	1	Idle	1	0	Unknown	No Status	0	0	Unknown	No Status	0	Night	Unknown	No Status	0
14	2	Running	2	28	Temperature	Enabled	0	35	Light	Enabled	0	Day	Moisture	Enabled	2200
15	3	Running	6	22	Temperature	Enabled	0	31	Light	Disabled	0	Night	Moisture	Disabled	0
16	1	Idle	1	0	Unknown	No Status	0	0	Unknown	No Status	0	Night	Unknown	No Status	0
17	2	Running	2	24	Temperature	Enabled	0	26	Light	Enabled	0	Day	Moisture	Enabled	900

Fig. 19. Output Hasil Report

Gambar diatas adalah hasil write oleh reporter ke dalam file control_center_report.csv. Bisa dilihat bahwa packet report dari ketiga station ditulis secara bergiliran. Terlihat juga station 1 melakukan instruction idle, station 2 melakukan instruction run all sensors, dan station 3 melakukan instruction run temp sensor only.

3.6 ANALYSIS

Weather Station bertugas untuk membaca data yang didapatkan dari Station Controller yang mengirimkan data berupa tiga sensor, yaitu Sensor Suhu (temperature), Sensor Kelembapan (Moisture), dan Sensor Intensitas Cahaya (daylight). Station Controller tersebut akan membaca dari file CSV yang memuat data berupa Suhu, Intensitas Cahaya, Massa Air, dan Volume Air. Massa Air dan Volume Air akan diproses melalui Station Controller akan dapat memberikan dan mendapatkan tingkat kelembaban udara dengan rumus yang sudah ditentukan. Pada masing-masing sensor, terdapat data 16-bit yang dikirimkan kepada Weather Station dengan format **AA BB C DDDDDDDDDDDDD**. Aliran data 16-bit ini terdiri dari Kode Jenis Sensor 2-bit (01 = temp, 10 = light, 11 = moist), 2-bit status (00 = no status, 01

= running, 10 = off), dan 11 bit signed integer value pembacaan. Ketiga komponen ini disusun menjadi packet Data dengan urutan sebagai berikut

Type	Status	Value
------	--------	-------

Weather Station sendiri akan memiliki beberapa state dan pembagian tugasnya masing-masing. Pada State IDLE, weather station akan terus menunggu sampai menerima instruksi dari pusat untuk dijalankan. Instruksi berupa 8-bit yang memiliki format berupa **AA BBBB BBB** yang merupakan selektor station tujuan ID (2-bit) dan 6-bit opcode untuk instruksi yang dijalankan. Instruksi memiliki struktur sebagai berikut

Destination	Opcode
-------------	--------

Ketika weather station berada di State READ_INST, weather station akan mengecek opcode yang diterima dari instruction tersebut, dan akan dibagi berdasarkan tabel opcode dibawah ini.

Opcode	Instruksi	Keterangan
000001	Idle	Idle
000010	Run	Semua sensor berjalan
000011	Run but stop temp	Semua sensor berjalan kecuali temperatur
000100	Run but stop daylight	Semua sensor berjalan kecuali light
000101	Run but stop moist	Semua sensor berjalan kecuali moisture
000110	Run temp only	Hanya temperatur aktif
000111	Run daylight only	Hanya light aktif
001000	Run moist only	Hanya moisture aktif

Nantinya, READ_INST akan ditransisi ke state berikutnya yaitu ENCODE. Pada State ENCODE, sensor data akan diencode sesuai dengan status sensor, yaitu Tipe (2-bit) yang berisi identifikasi sensor (01 = temp, 10 = light, 11 = moist), Status (2-bit) yang berisi menandai sensor berjalan atau berhenti (01 = running, 10 = stopped), dan Data (12-bit) yang berisi data sensor yang bergantung pada pemilihan

opcode sebelumnya. Setelah ter-encode, state akan bertransisi ke state GENERATE_REPORT yang akan membuat sebuah packet report berupa 64-bit dengan format:

Bit	Data
63 - 62 (2 bit)	Source Station ID
61 - 60 (2 bit)	Status
59 - 54 (6 bit)	Opcode
53 - 48 (6 bit)	Timestamp
47 - 32 (16 bit)	Data temp
31 - 16 (16 bit)	Data light
15 - 0 (16 bit)	Data moist

Packet report tersebutlah yang nantinya akan dikirim ke Control Center untuk dicatat. Packet report yang berupa 64-bit tersebut akan pertama-tama di decode menjadi beberapa bagian yang mencakup data pada tabel sebelumnya. Nantinya, Control Center tersebut akan membaca sekaligus memvalidasi data yang terkirim. Setelah selesai, nantinya data-data yang sudah ter-decode itu akan diproses dan disimpan ke dalam sebuah file CSV untuk mencatat setiap report pada setiap station controller yang ada secara real-time. Pencatatan report tersebut memiliki format sebagai berikut:

Source	Status	Opcode	Timestamp	Data temp	Data light	Data moist
--------	--------	--------	-----------	-----------	------------	------------

Data CSV report tersebut yang akan menjadi bahan pelaporan dan analisis cuaca yang ada, guna untuk quality control dari cuaca yang ada pada tiap jam.

CHAPTER 4

CONCLUSION

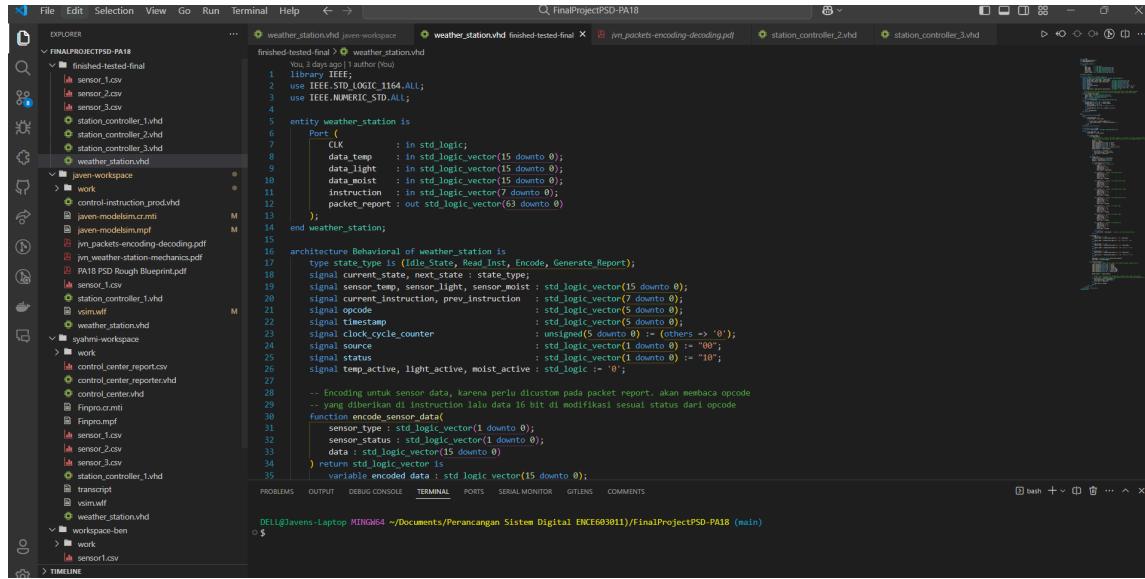
Proyek ini dengan judul IoT Remote Weather Station & Centralized Control Center yang telah kelompok kami rancang berhasil dijalankan dengan hasil yang memuaskan. Lalu IOT Weather Station & Center ini juga dapat mengimplementasikan modul serta teori perancangan sistem digital. Proses membaca sensor, decode, dan report juga dapat diimplementasikan dengan baik. Secara keseluruhan, proyek berjudul judul IoT Remote Weather Station & Centralized Control Center ini berjalan dengan baik dan kami harap dapat berkontribusi pengembangan sistem yang lebih canggih dan responsif untuk memantau serta mengelola bencana di Indonesia.

REFERENCES

- Ricardo-Jasinski, “Ricardo-Jasinski/VHDL-CSV-file-reader: VHDL package for reading formatted data from comma-separated-values (CSV) files,” GitHub, <https://github.com/ricardo-jasinski/vhdl-csv-file-reader>
- “Ada 1.300 Bencana Alam di ri Sampai September 2024, Ini Rinciannya: Databoks,” Pusat Data Ekonomi dan Bisnis Indonesia, <https://databoks.katadata.co.id/demografi/statistik/66d7d7a492e96/ada-1300-bencana-alam-di-ri-sampai-september-2024-ini-rinciannya>
- Digilab, “Digital System Design,” Digilab UI, <https://learn.digilabdte.com/books/digital-system-design>
- “VHDL structural modeling style,” Surf, <https://surf-vhdl.com/vhdl-syntax-web-course-surf-vhdl/vhdl-structural-modeling-style/>
- “Mastering loops in VHDL: Enhancing flexibility and performance in hardware design,” FPGA Insights, <https://fpgainsights.com/fpga/mastering-loops-in-vhdl-enhancing-flexibility-and-performance-in-hardware-design>
- IEEE, *Design of encoding and decoding of hamming code based on VHDL*. IEEE Xplore, <https://ieeexplore.ieee.org/document/9443744>
- “Implementing a finite state machine in VHDL” All About Circuits, <https://www.allaboutcircuits.com/technical-articles/implementing-a-finite-state-machine-in-vhdl/>

APPENDICES

Appendix A: Project Progress & Documentation



```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;

entity weather_station is
    Port (
        data_temp : in std_logic_vector(15 downto 0);
        data_light : in std_logic_vector(15 downto 0);
        data_moist : in std_logic_vector(15 downto 0);
        instruction : in std_logic_vector(7 downto 0);
        packet_report : out std_logic_vector(63 downto 0)
    );
end weather_station;

architecture Behavioral of weather_station is
begin
    type state_type is (Idle_State, Read_Inst, Encode, Generate_Report);
    signal current_state, next_state : state_type;
    signal sensor_temp, sensor_light, sensor_moist : std_logic_vector(15 downto 0);
    signal current_instruction, prev_instruction : std_logic_vector(7 downto 0);
    signal timestamp : std_logic_vector(5 downto 0);
    signal clock_cycle_counter : unsigned(5 downto 0) := (others => '0');
    signal source : std_logic_vector(1 downto 0) := "00";
    signal status : std_logic_vector(1 downto 0) := "10";
    signal temp_active, light_active, moist_active : std_logic := '0';

    -- Encoding untuk sensor data, karena perlu dicontum pada packet report, akan membutuhkan modifikasi sesuai status dari opcode
    function encode_sensor_data(
        sensor_type : std_logic_vector(1 downto 0);
        sensor_status : std_logic_vector(1 downto 0);
        data : std_logic_vector(15 downto 0)
    ) return std_logic_vector is
        variable encoded_data : std_logic_vector(15 downto 0);
    begin
        ... (implementation of the function)
    end function;

```

