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

SMART WATER LEVEL MONITORING SYSTEM USING FSM

GROUP PA26

Putri Kiara Salsabila Arief	**2306250743**
M. Ikhsan Kurniawan	**2306210784**
Siti Amalia Nurfaidah	**2306161851**
Wesley Frederick Oh	**2306202763**

PREFACE

Dalam proyek ini, kami mengembangkan *Smart Water Level Monitoring System*, sebuah sistem cerdas untuk memantau dan mengontrol ketinggian air dalam tangki secara otomatis. Sistem ini dirancang menggunakan konsep *Finite State Machine (FSM)*, yang memungkinkan pengelolaan kondisi tangki berdasarkan tiga klasifikasi utama yaitu *Low, Normal,* dan *High.* Masing-masing kondisi memiliki aksi spesifik, seperti mengaktifkan pompa untuk mengisi air saat kondisi *Low* dan mematikan pompa saat kondisi *Normal* dan *High*.

Sistem ini menggunakan beberapa input utama, seperti tingkat ketinggian air, suhu air, dan kecepatan aliran air, untuk menghitung *flow rate* dan efisiensi pompa (ηP). Selain itu, sistem dirancang untuk mendeteksi kondisi kritis apabila air berada terlalu lama di level rendah. Pada kondisi tersebut, pompa akan dimatikan, dan alarm diaktifkan untuk memberikan peringatan. Output dari sistem ini mencakup kontrol pompa (*ON/OFF*) dan aktivasi alarm berdasarkan situasi yang terdeteksi.

Dalam implementasinya, berbagai pendekatan desain berbasis VHDL digunakan untuk mewujudkan sistem ini. *Behavioral Style* digunakan untuk merancang logika FSM serta pengaturan *timer alarm*. Selain itu, *Structural Style* memungkinkan pemisahan sistem menjadi *component* terpisah seperti tank, pompa, dan alarm, sehingga meningkatkan modularitas. Sistem ini juga dilengkapi dengan *testbench* untuk memastikan bahwa setiap fungsi bekerja sesuai harapan.

Melalui laporan ini, kami akan menjelaskan secara rinci proses perancangan dan implementasi sistem ini, mulai dari definisi masalah, pendekatan solusi, hingga hasil pengujian. Diharapkan laporan ini dapat memberikan pemahaman yang mendalam tentang penerapan teknologi *Finite State Machine* dalam sistem pengendalian otomatis, sekaligus menjadi landasan bagi pengembangan lebih lanjut di bidang monitoring dan kontrol air berbasis VHDL.

Depok, December 5, 2024

Group PA26

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

Dalam era modern ini, teknologi otomatisasi menjadi salah satu elemen penting untuk mendukung efisiensi dan pengelolaan berbagai kebutuhan sehari-hari. Salah satu aplikasi yang memiliki peranan signifikan adalah sistem pemantauan dan pengendalian air dalam tangki. *Water Level Monitoring System* merupakan sistem cerdas yang dirancang untuk memantau dan mengontrol tingkat ketinggian air dalam tangki secara otomatis. Sistem ini menggunakan input berupa kondisi ketinggian air yang terbagi menjadi tiga kategori yaitu *Low, Normal,* dan *High.* Kategori tersebut merepresentasikan level air ketika berada di bawah batas minimum, di sekitar batas minimum, atau mencapai batas maksimum.

Kunci utama dari sistem ini terletak pada implementasi *Finite State Machine* (FSM), yang memungkinkan transisi antar kondisi berdasarkan keadaan air dalam tangki. Sistem ini memiliki empat *state* utama, yaitu:

- **Idle**, saat pompa tidak aktif.
- **Filling**, ketika pompa aktif untuk mengisi air.
- **Full**, saat tangki penuh dan pompa dimatikan.
- **Critical**, kondisi ketika air berada terlalu lama pada level *Low* tanpa diisi, sehingga memicu alarm sebagai peringatan.

Sebagai output, sistem ini menghasilkan dua sinyal utama yaitu kontrol untuk menyalakan atau mematikan pompa, dan aktivasi alarm saat kondisi kritis terdeteksi. Sistem juga memanfaatkan algoritma cerdas untuk menghitung efisiensi (*efficiency*) dan laju aliran air (*flow rate*), serta menyimpan data hasil pengolahan tersebut dalam *file output* untuk keperluan analisis dan debugging oleh pengguna.

Keunggulan *Water Level Monitoring System* terletak pada kemampuannya yang lebih cerdas dibandingkan pompa konvensional. Dengan memanfaatkan algoritma berbasis FSM dan logika digital, sistem ini tidak hanya mengelola ketinggian air tetapi juga memastikan efisiensi operasional yang optimal. Proyek ini diharapkan dapat menjadi solusi inovatif dalam pengelolaan sumber daya air yang lebih efektif dan otomatis.

1.2 PROJECT DESCRIPTION

Proyek *Water Level Monitoring System using FSM* adalah sistem berbasis logika digital yang dirancang untuk memantau dan mengontrol tingkat ketinggian air dalam tangki menggunakan bahasa pemrograman VHDL dengan prinsip Finite State Machine (FSM). Sistem ini menerima input berupa tingkat ketinggian air (Low, Normal, High), suhu tangki dan mengatur pompa serta alarm sesuai dengan kondisi tersebut. Pompa aktif saat tingkat air rendah, dan mati saat air mencapai level normal atau tinggi. Selain itu, sistem ini juga memonitor suhu dan flow rate untuk mendeteksi potensi kerusakan pada pompa, serta menghitung efisiensi pompa berdasarkan rumus yang melibatkan flow rate, delivery head, dan nilai konversi. Dengan pendekatan ini, sistem dapat memberikan kontrol otomatis serta deteksi dini terhadap potensi masalah pada pompa, meningkatkan efisiensi dan keandalan sistem pengisian air.

1.3 OBJECTIVES

The objectives of this project are as follows:

1. Membuat komponen pompa.
2. Membuat komponen tangki air.
3. Membuat sebuah komponen master/TopLevel untuk menghubungkan input dan output dari pompa ke tangki maupun sebaliknya.
4. Membuat output dari komponen agar dapat diimplementasikan pada file output berupa text file.

1.4 ROLES AND RESPONSIBILITIES

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

Roles	Responsibilities	Person
Membuat kode dan membuat laporan	● Membuat kode pompa ● Membuat kode top level	Wesley Frederick Oh

	• Membuat laporan	
Membuat kode dan membuat laporan	• Membuat kode pompa • Membuat kode top level • Membuat laporan	M. Ikhsan Kurniawan
Membuat kode dan membuat laporan	• Membuat kode tank • Membuat kode testbench • Membuat Readme • Membuat laporan	Siti Amalia Nurfaidah
Membuat kode dan membuat laporan	• Membuat kode tank • Membuat kode testbench • Membuat Readme • Membuat laporan	Putri Kiara Salsabila Arief

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:

- Visual Studio Code
- ModelSim - INTEL FPGA STARTER EDITION
- Quartus Prime Lite Edition
- Git and GitHub

2.2 IMPLEMENTATION

Ada 3 buah komponen VHDL utama yang digunakan dalam project ini, yakni komponen untuk Pompa, Tangki air, dan Top Level.

Pompa air adalah komponen yang bertujuan untuk mengindikasikan apakah pompa sedang mengisi tangki atau tidak, caranya dengan menggunakan FSM sebagai indikator apa yang sedang dilakukan oleh pompa tersebut, apakah sedang mengisi air atau tidak melakukan apa-apa. Alhasil, ada 2 state yang dimiliki oleh pompa, yaitu GABUT dan MENGISI. Hubungan antara kedua state tersebut dapat dilihat pada state diagram di bawah ini:

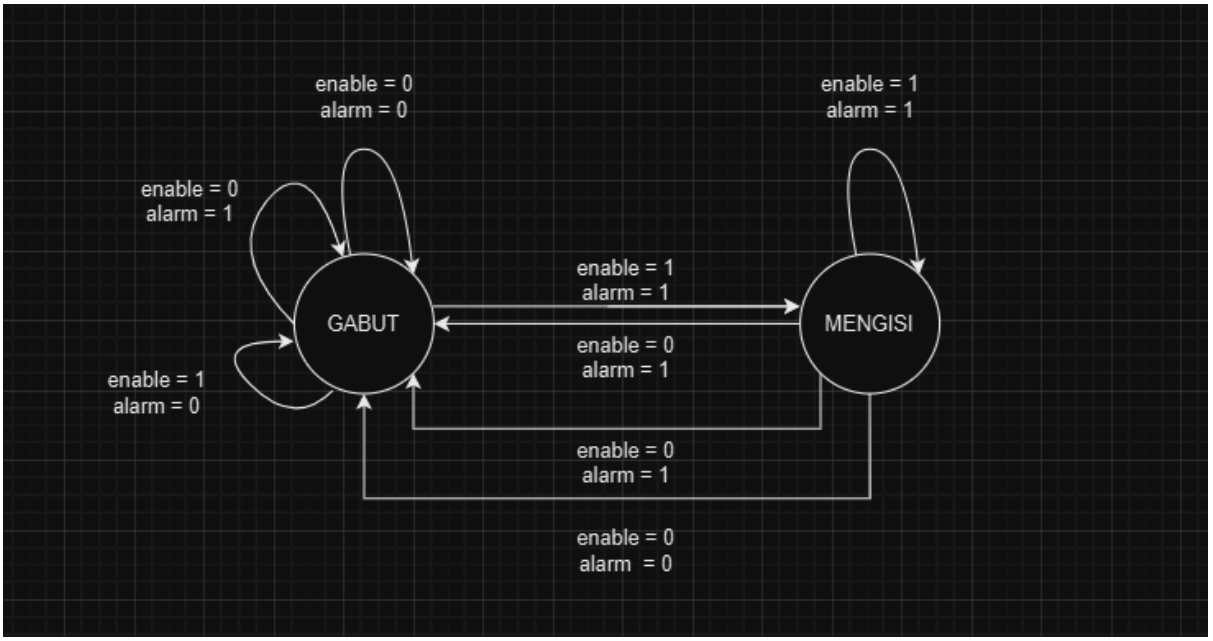


Fig 1. Finite State Machine Komponen Pompa

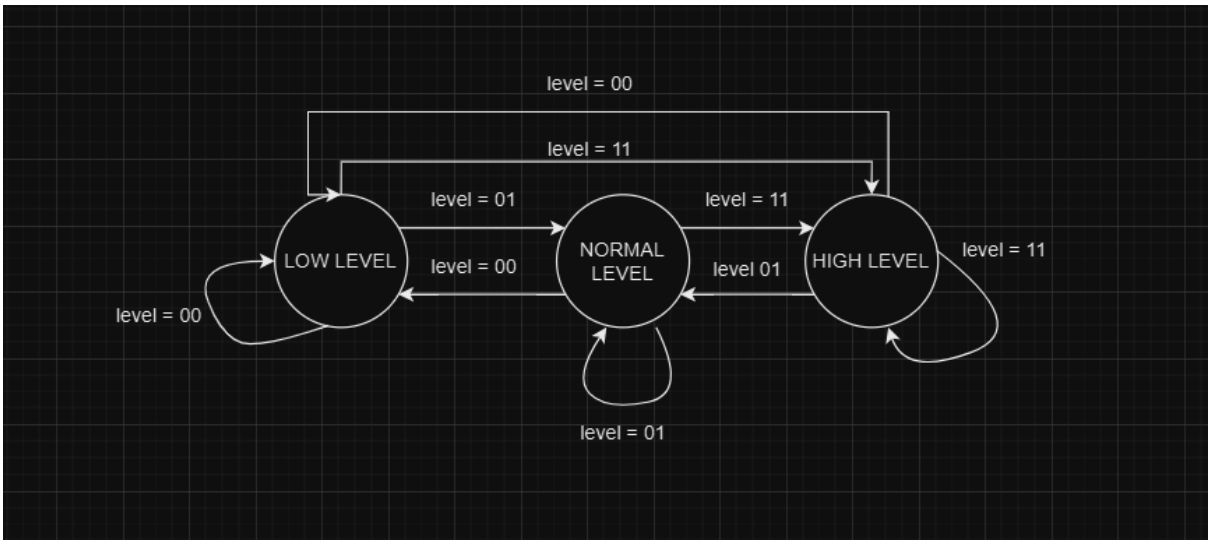


Fig 2. Finite State Machine Komponen Tank

Karena input state tidak mempengaruhi output dari masing-masing state, kita bisa mendeskripsikan FSM ini dalam bentuk Moore FSM yang di mana output state akan hanya bergantung pada state sekarang saja.

State awal dan juga state default dari pompa adalah GABUT, yaitu ketika pompa tidak melakukan apa apa. Pompa akan selalu berada dalam state ini ketika pompa dimatikan (enable_pompa = 0). Pompa baru akan melakukan transisi dari state GABUT ke MENGISI ketika pompa dihidupkan dan menerima input alarm dari Tangki yang menunjukkan bahwa kondisi air dalam tangki berada pada level rendah.

Selain mengisi air, pompa juga memiliki kegunaan lain, yakni dapat digunakan untuk menghitung efisiensi pompa secara *real-time* serta mengeluarkan outputnya di hasil simulasi dan juga di sebuah log file output berupa text file.

Di sisi lain, komponen Tank (atau tangki) adalah komponen yang berlaku sebagai tangki air di dunia nyata, yang akan di-*monitor* behaviornya tergantung tingkat atau ketinggian air di dalam tangki tersebut. Walaupun tidak disampaikan secara eksplisit state-nya dengan menggunakan StateType, Tank menggunakan FSM yang disimbolkan dengan logic vector. Ada 3 buah state yang digunakan, yaitu LOW atau “00,” NORMAL atau “01,” dan HIGH atau “11.” Tergantung ketinggian air yang dideteksi atau diberikan oleh user, Tank akan memasuki salah satu dari ketiga state ini.

Pada state LOW, Tank akan mengeluarkan output yang menunjukkan bahwa air sudah berada pada kondisi kritis, serta menunggu selama kurang lebih lima clock cycles lalu menyalakan alarm. Alarm inilah yang digunakan untuk men-*trigger* pompa untuk mengisi air. Kemudian pada state NORMAL dan HIGH, artinya air masing-masing dalam kondisi hampir penuh atau sudah sangat penuh, sehingga alarm akan dimatikan kembali.

TopLevel berlaku sebagai komponen Master yang bertujuan menggabungkan atau menghubungkan kedua komponen Tank dan Pompa dengan cara mengutilisasikan *port mapping* sebagai implementasi Structural Style dalam project ini. Dalam TopLevel, karena alarm berlaku sebagai output dari Tank, dan output tersebut dijadikan input oleh Pompa, jadi pada TopLevel, alarm didefinisikan sebagai INOUT sehingga tidak perlu mendeklarasikan 2 buah signal pompa yang berbeda.

CHAPTER 3

TESTING AND ANALYSIS

3.1 TESTING

Testing dilakukan pada software ModelSim untuk dapat melakukan simulasi VHDL dalam bentuk Wave. Untuk menyederhanakan proses ini, telah dibuatkan sebuah file testbench yang setelah disimulasikan di ModelSim.

Dalam project ini, *Testing* dilakukan guna untuk menguji apakah pompa dan tangki masing-masing telah bekerja dengan benar, yaitu tangki dapat memonitor tingkat air di dalamnya serta membunyikan alarm ketika tingkat air sudah terlalu rendah. Pompa juga harus diuji agar pompa dapat menerima peringatan alarm dari tangki, yang tandanya pompa harus mulai mengisi air hingga tingkat air kembali normal atau lebih dari normal. Dengan kata lain, testing di sini bertujuan untuk memastikan keberhasilan interaksi antar tangki dan juga pompa.

Dengan menggunakan testbench, hasil perhitungan dalam komponen pompa dapat dikeluarkan dalam bentuk sebuah output file berupa .txt file dengan memanfaatkan library TEXTIO dalam VHDL yang secara akurat menunjukkan hasil efisiensi dan debit air setiap periode waktu tertentu.

3.2 RESULT

Melalui testbench yang disimulasikan melalui ModelSim, didapatkan hasil simulasi sebagai berikut:

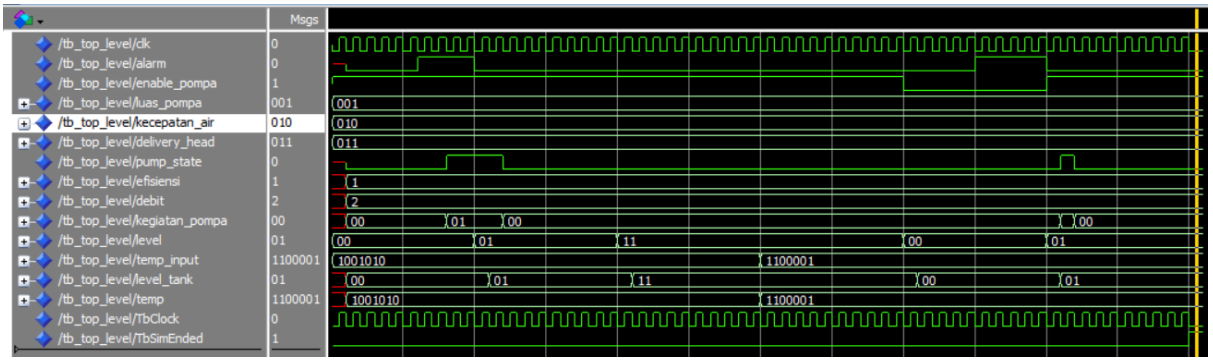
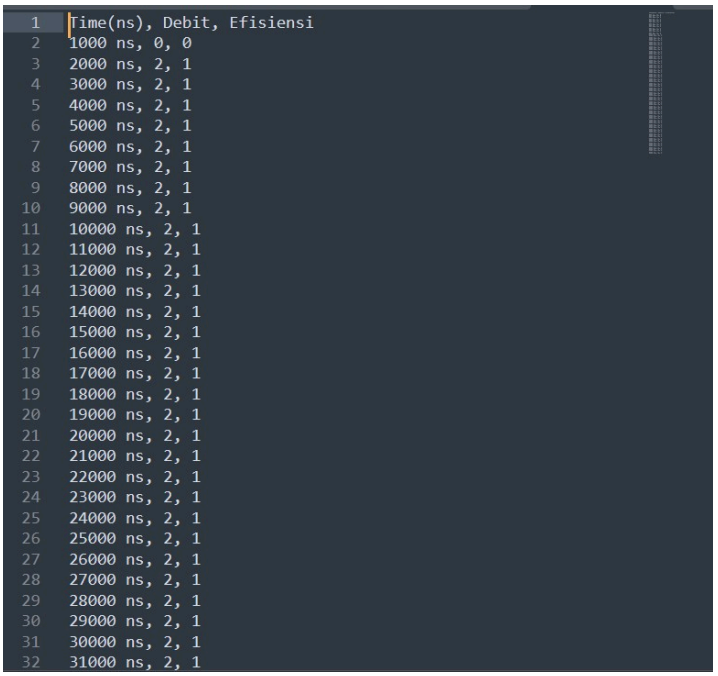


Fig 2. Testing Result

Pada hasil simulasi di atas, semua output dan input sudah terisi dengan benar, sehingga tidak ada kesalahan input maupun output pada project ini. Seluruh output sesuai dengan harapan, menunjukkan bahwa fungsi logika telah bekerja sesuai desain. Tidak ditemukan kesalahan pada sinyal input maupun output selama proses simulasi. Hal ini mengindikasikan bahwa desain telah terverifikasi dengan baik pada tahap simulasi.

Output yang muncul pada testbench terutama untuk rumus telah di-output secara benar di file text output seperti di bawah ini.



```
Time(ns), Debit, Efisiensi
1000 ns, 0, 0
2000 ns, 2, 1
3000 ns, 2, 1
4000 ns, 2, 1
5000 ns, 2, 1
6000 ns, 2, 1
7000 ns, 2, 1
8000 ns, 2, 1
9000 ns, 2, 1
10000 ns, 2, 1
11000 ns, 2, 1
12000 ns, 2, 1
13000 ns, 2, 1
14000 ns, 2, 1
15000 ns, 2, 1
16000 ns, 2, 1
17000 ns, 2, 1
18000 ns, 2, 1
19000 ns, 2, 1
20000 ns, 2, 1
21000 ns, 2, 1
22000 ns, 2, 1
23000 ns, 2, 1
24000 ns, 2, 1
25000 ns, 2, 1
26000 ns, 2, 1
27000 ns, 2, 1
28000 ns, 2, 1
29000 ns, 2, 1
30000 ns, 2, 1
31000 ns, 2, 1
```

Fig 3. Hasil Output File dalam Text File

3.3 ANALYSIS

Pada pengujian, terdapat sebuah clock yang berfungsi untuk mensinkronisasikan kedua komponen pompa dan tangki, yang kedua komponen tersebut di-*trigger* dengan signal falling edge.

Kemudian, tingkat air yang saat ini terdapat dalam tangki dapat dilihat berdasarkan signal level, di mana 00 menunjukkan air berada pada LOW level, 01 untuk NORMAL level, dan 11 untuk HIGH level. Dengan menggunakan konsep looping dalam VHDL, setiap kali

ketika tingkat air dalam tangki berada pada LOW level selama kurang lebih 5 clock cycle (falling edge), alarm akan dihidupkan yang diindikasikan oleh signal ‘alarm.’

Ketika alarm berbunyi, tangki mengirim peringatan pada pompa yang kemudian berfungsi untuk mengisi tangki yang ditandai dengan signal ‘pump_state’ bernilai 1. Ketika pompa mengisi, ini menyebabkan pompa berubah state menjadi MENGISI atau signal ‘kegiatan_pompa’ menjadi 01. Pompa baru akan berhenti mengisi jika tangki memberitahu bahwa tingkat air saat ini sudah tidak berada pada level kritis lagi, atau dengan kata lain pompa berhenti mengisi ketika tingkat air berada pada NORMAL atau HIGH. Saat pompa berhenti inilah pompa mengubah statenya kembali menjadi state default (atau dalam konteks ini, state GABUT).

Di saat yang sama, untuk setiap clock cyclenya, kita bisa memasukkan beberapa nilai yang menjadi bahan perhitungan rumus efisiensi pompa dan debit air, seperti luas pompa, kecepatan air, dan delivery head. Walaupun dalam hasil simulasi di Fig. 2 kedua nilai ini dinyatakan dalam bilangan unsigned, namun sebenarnya efisiensi pompa dinyatakan dalam persentase (%) sedangkan debit air dinyatakan dalam m^3/s. Selain itu, demi kenyamanan pengamat serta pembuat kode, kedua nilai ini tidak dinyatakan dalam floating point namun telah dibulatkan sesuai angka floating point yaitu jika terdapat floating point lebih dari .4 akan dibulatkan ke atas. Untuk rumus lengkapnya dapat dilihat sebagaimana di bawah ini:

$$\eta_P = \frac{Q \cdot H \cdot \rho}{367 \cdot P_2}$$

Fig 4. Rumus Efisiensi Pompa Air

Flow Rate (Q) = Luas Penampang Pipa (A) × Kecepatan Aliran (v)

Fig 5. Rumus Flow Rate Air

Terdapat 2 buah konstanta yaitu ρ air yang di-assign secara default yaitu 1000 kg/m^3 dan P2 yaitu daya pompa yaitu 2 kW. Dengan memasukkan nilai-nilai berkaitan yang ada di Fig 2., berhasil dibuktikan bahwa perhitungan di bagian testing sudah berjalan dengan benar.

Setelah semua perhitungan berhasil dilakukan, kedua rumus di atas dituliskan sebagai output pada sebuah output file berupa text file yang hasilnya dapat dilihat pada Fig. 3.

CHAPTER 4

CONCLUSION

Pompa air adalah suatu kebutuhan kritis yang diperlukan manusia untuk mengelola air demi kelengkapan hidup sehari-hari. Oleh karena itu, kelompok PA26 telah sukses membuat *Smart Water Level Monitoring System* yang secara akurat mengawasi tingkat ketinggian air dalam sebuah tangki, dan otomatis menghidupkan pompa air ketika tingkat air sudah terlalu rendah.

Secara keseluruhan, project ini berfokus pada pemanfaatan serta implementasi dari Finite State Machine (FSM) untuk memberitahu tingkat air saat ini serta state dari pompa itu sendiri apakah sedang mengisi air atau tidak, serta menuliskan output program ke output file sebagai fitur tambahan.

Setelah melalui proses simulasi, berhasil dibuktikan bahwa perilaku dari pompa, tangki, hubungan antara keduanya, serta rumus efisiensi dan debit sudah sesuai dengan kehendak, dan tidak ada kejanggalan yang terlihat pada hasil simulasi.

Karena alasan itulah, project ini bisa dikatakan sudah berhasil mengeksekusi *Smart Water Level Monitoring System* dengan sangat baik, meskipun memang masih terdapat banyak ruang untuk peningkatan dan penambahan fitur lainnya yang berpotensi dapat diimplementasikan juga.

REFERENCES

[1] A. Oukaira, A. Z. Benelhaouare, E. Kengne, and A. Lakhssassi, “FPGA-Embedded Smart Monitoring System for Irrigation Decisions Based on Soil Moisture and Temperature Sensors,” MDPI, https://www.mdpi.com/2073-4395/11/9/1881 (accessed Dec. 7, 2024).

[2] “What is the efficiency of a centrifugal pump?,” Wilo, https://wilo.com/ie/en/Solutions-Finder/FAQ/Centrifugal-pump/What-is-the-efficiency-of-a-centrifugal-pump/ (accessed Dec. 7, 2024).

[3] Vin, “Memahami Flow Rate: Definisi dan Cara Menghitung Lengkap dengan Tabel,” ACS, https://www.alvindocs.com/blog/flow-rate (accessed Dec. 7, 2024).

APPENDICES

Appendix A: Project Schematic

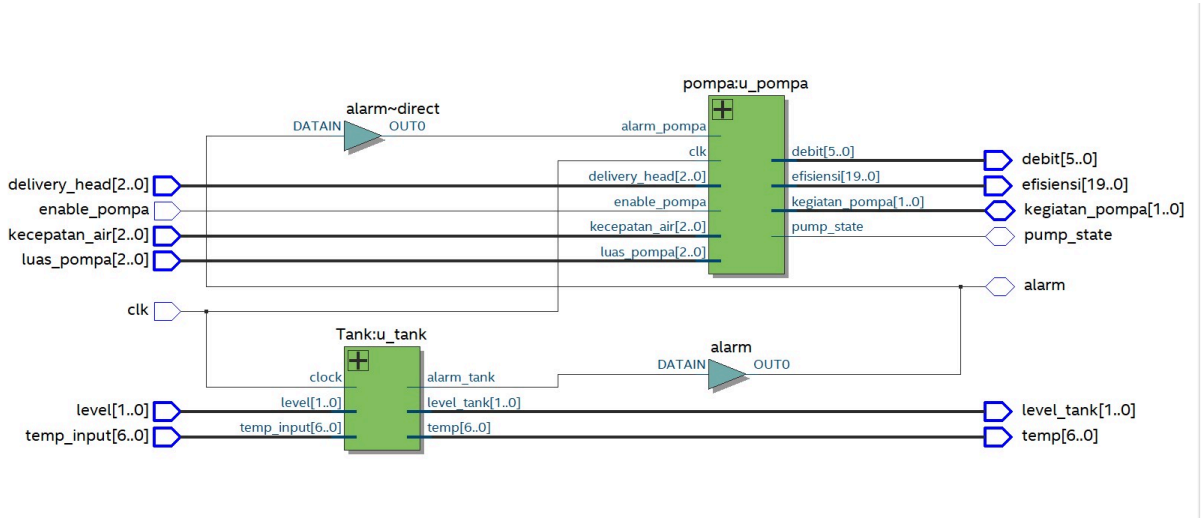


Fig 6. Hasil Sintesis TopLevel

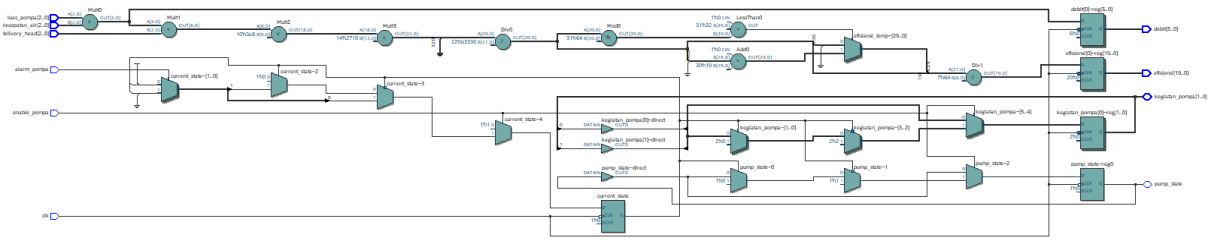


Fig 7. Hasil Sintesis Komponen Pompa

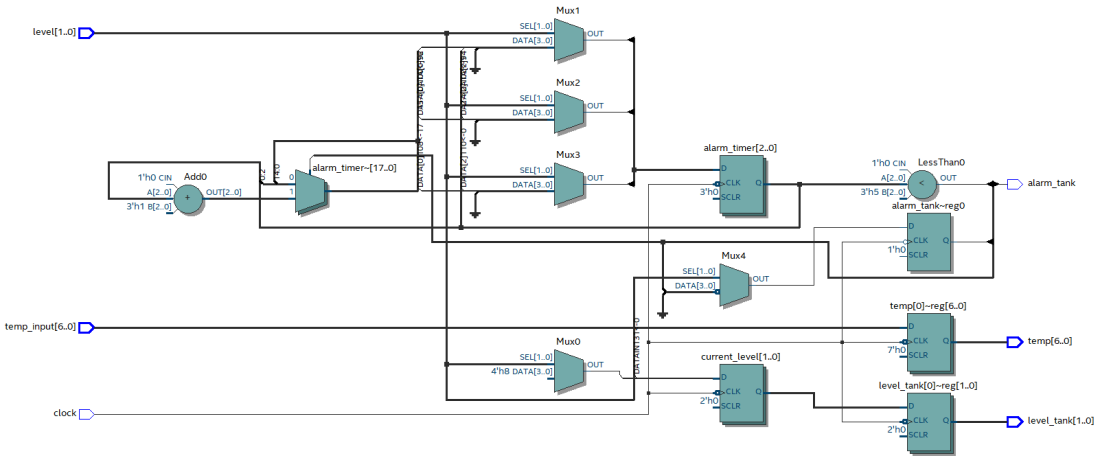


Fig 8. Hasil Sintesis Komponen Tank

Appendix B: Documentation