Smart Water Level Monitoring System Using FSM

Final Project Praktikum Perancangan Sistem Digital Kelompok PA26

Anggota Kelompok

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

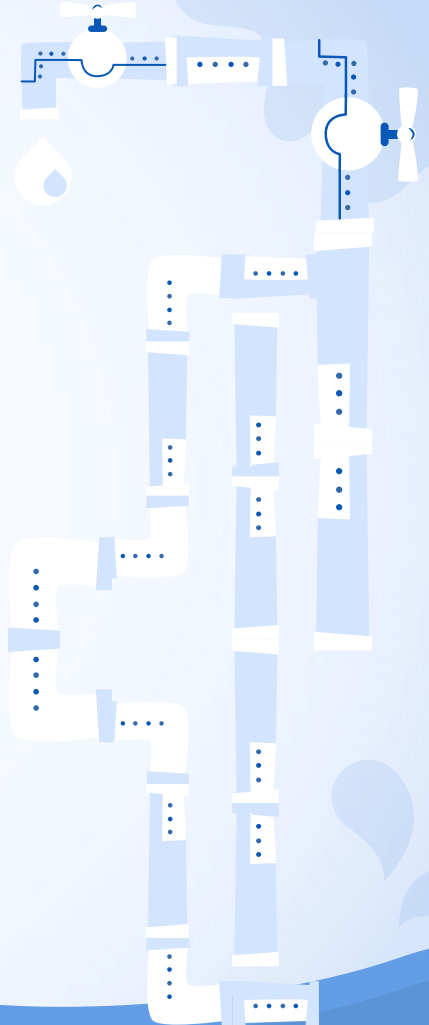


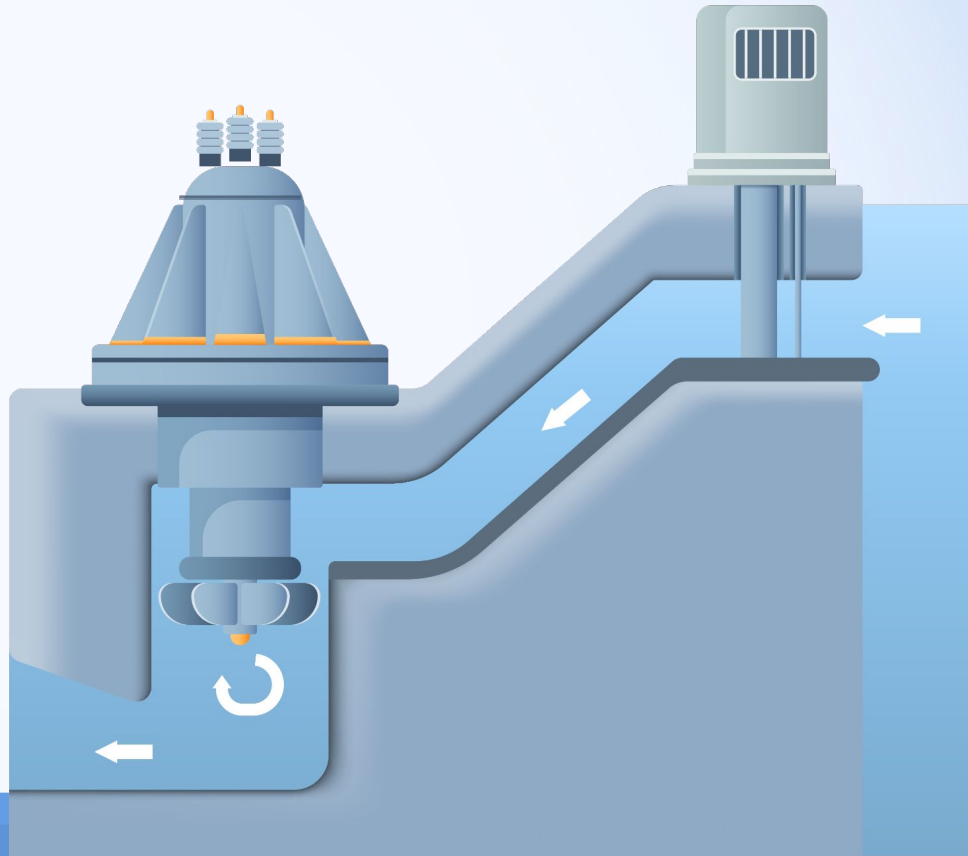
Table of contents

01

Latar Belakang

Latar Belakang

Dalam era modern, otomatisasi sangat penting untuk efisiensi pengelolaan berbagai kebutuhan sehari-hari, salah satunya dalam pengendalian tingkat air dalam tangki. Proyek Water Level Monitoring System menggunakan VHDL dan Finite State Machine (FSM) untuk memantau dan mengatur pompa berdasarkan level air. Sistem ini juga memonitor suhu dan flow rate, serta memberikan peringatan alarm saat level air mencapai batas kritis, meningkatkan efisiensi dan keandalan pengelolaan air secara otomatis.

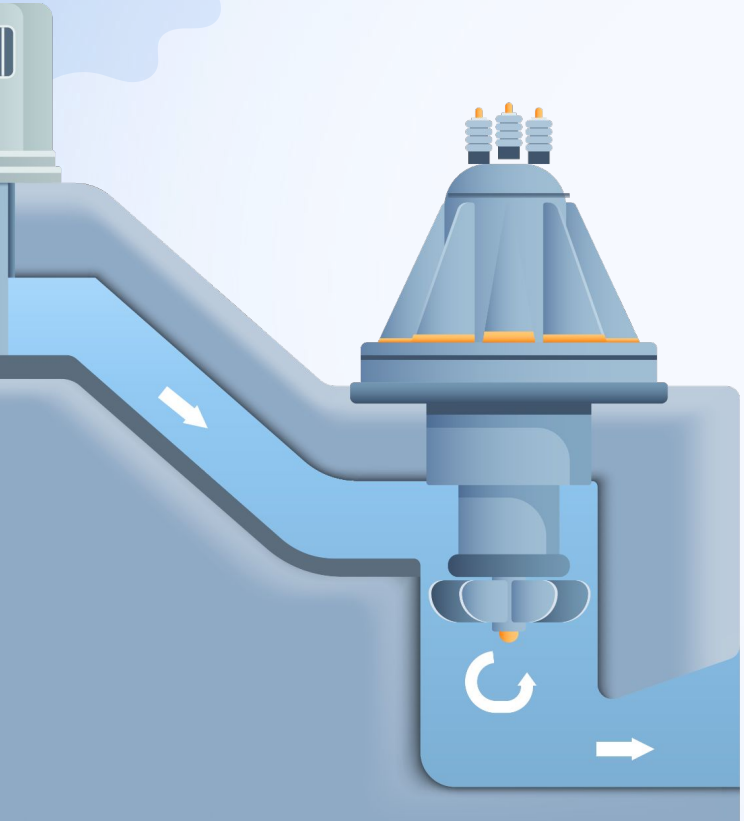


02

Deskripsi Proyek

Deskripsi Proyek

Proyek ini bertujuan untuk mengimplementasikan sistem pemantauan dan pengendalian tingkat air dalam tangki menggunakan VHDL dan Finite State Machine (FSM). Sistem ini akan memantau level air dalam tiga kategori yaitu Low, Normal, dan High, dan mengatur pompa serta alarm sesuai dengan kondisi tersebut. Fungsi utama yang akan diimplementasikan mencakup kontrol pompa otomatis berdasarkan level air, aktivasi alarm saat level air berada dalam kondisi kritis, serta pemantauan suhu dan flow rate untuk mendeteksi potensi kerusakan pada pompa. Selain itu, sistem ini juga akan menghitung efisiensi pompa berdasarkan flow rate dan delivery head, serta menyimpan data hasil pengolahan dalam file output untuk analisis lebih lanjut.



03

Objectives & Equipment

Objectives

- Membangun implementasi perangkat keras menggunakan VHDL untuk pemantauan dan pengendalian tingkat air dalam tangki.
- Mengontrol pompa secara otomatis berdasarkan level air (Low, Normal, High) untuk mengatur pengisian air pada tangki.
- Mengaktifkan alarm saat level air mencapai kondisi kritis, memberikan peringatan bagi pengguna.
- Memantau suhu dan flow rate untuk mendeteksi potensi kerusakan pada pompa dan memastikan kinerja optimal.
- Menghitung efisiensi pompa berdasarkan flow rate dan delivery head, serta menyimpan data hasil pengolahan dalam file output untuk analisis dan debugging.

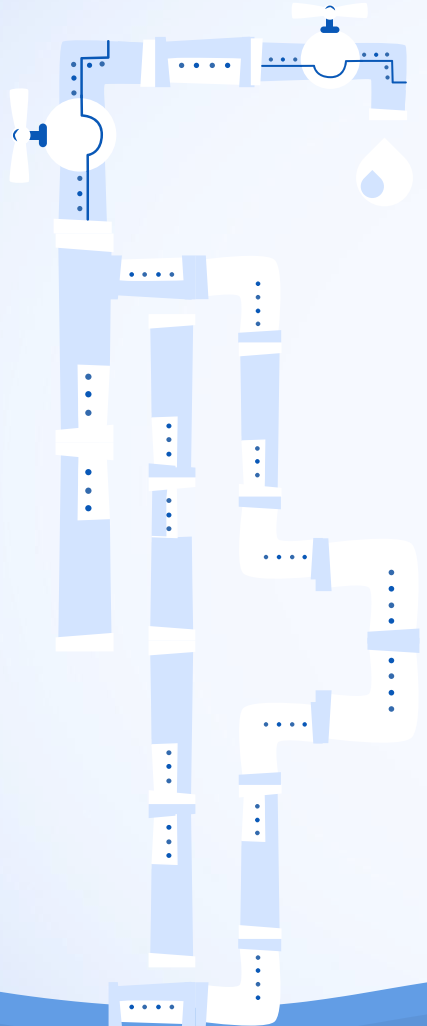
Equipment

Visual Studio Code

ModelSim - INTEL FPGA STARTER EDITION

Quartus Prime Lite Edition

Git and GitHub



04

Implementation

FSM Diagram

FSM Diagram untuk Pompa:

1. State GABUT (Idle):
   - Pompa berada dalam kondisi tidak aktif (idle) ketika sinyal enable = 0, tanpa memperdulikan status alarm (alarm = 0 atau alarm = 1).
   - Pompa tetap berada di state GABUT selama enable = 0. Jika enable = 1 dan alarm = 1, pompa bertransisi ke state MENGISI untuk mulai bekerja.

2. State MENGISI (Aktif):
   - Pompa aktif mengisi air ke tangki. Jika alarm = 0 (air sudah mencukupi) atau enable = 0 (pompa dimatikan secara manual), pompa kembali ke state GABUT.
   - Kombinasi enable = 1 dan alarm = 1 membuat pompa tetap berada di state MENGISI.

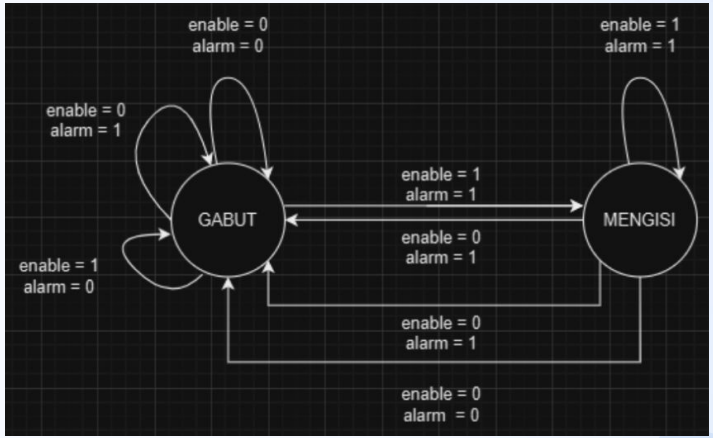


Fig 1. Finite State Machine Komponen Pompa

FSM Diagram

FSM Diagram untuk Tangki:

1. LOW LEVEL (00):
   Tangki memasuki state ini jika level air berada di bawah ambang batas minimum (level = 00). Pada kondisi ini, tangki menunggu hingga 5 siklus clock sebelum menyalakan alarm. Alarm yang menyala akan menjadi sinyal bagi pompa untuk mulai mengisi air.

2. NORMAL LEVEL (01):
   Tangki berada di state ini ketika level air berada di kondisi aman tetapi belum penuh (level = 01). Alarm dimatikan pada state ini, dan pompa berhenti mengisi air.

3. HIGH LEVEL (11):
   Tangki berada di state ini saat level air sudah penuh (level = 11). Alarm tetap mati di state ini, dan pompa tetap tidak bekerja.

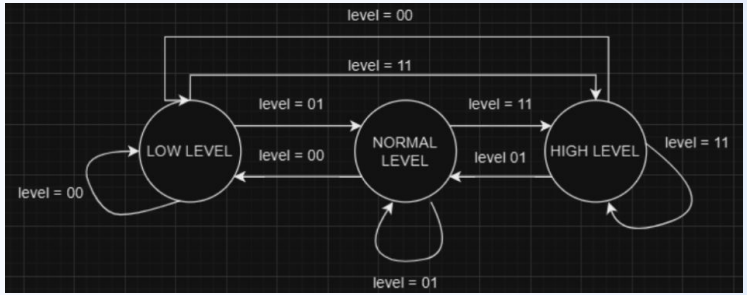


Fig 2. Finite State Machine Komponen Tangki

Pompa

Komponen Pompa akan aktif pada state MENGISI saat menerima sinyal alarm yang menunjukkan level air rendah dan kembali ke state GABUT ketika level air normal atau tinggi. Selain itu, pompa menghitung debit air (m^3/s) dan efisiensi operasional (persentase) dengan memperhitungkan luas pompa, kecepatan air, delivery head, daya, dan densitas air. Output utama adalah status pompa (Pump_state), debit air, dan efisiensi, yang memberikan informasi real-time tentang kinerja pompa.

```
54        PROCESS (clk, kecepatan_air, luas_pompa, delivery_head)
55            VARIABLE internal_debit : unsigned(5 DOWNTO 0);
56        BEGIN
57            IF falling_edge(clk) THEN
58                IF enable_pompa = '1' THEN
59                    IF current_state = MENGISI THEN
60                        pump_state <= '1';
61                        kegiatan_pompa <= "01";
62                        IF alarm_pompa = '0' THEN
63                            current_state <= GABUT;
64                        END IF;
65                    ELSIF current_state = GABUT THEN
66                        pump_state <= '0';
67                        kegiatan_pompa <= "00";
68                        IF alarm_pompa = '1' THEN
69                            current_state <= MENGISI;
70                        END IF;
71                    END IF;
72                ELSIF enable_pompa = '0' THEN
73                    REPORT "Pompa mati! Hidupkan dulu bos" SEVERITY warning;
74                    current_state <= GABUT;
75                END IF;
76
77                internal_debit := calculate_debit(luas_pompa, kecepatan_air);
78                efisiensi <= STD_LOGIC_VECTOR(calculate_efisiensi(internal_debit, delivery_head, densitas, daya));
79                debit <= STD_LOGIC_VECTOR(internal_debit);
80            END IF;
81        END PROCESS;
82    END ARCHITECTURE Behavioral;
```

Fig 3. Potongan Source Code Pompa

Tangki

Komponen Tangki memantau level air menggunakan tiga state utama: LOW, NORMAL, dan HIGH. Saat level air di LOW (00), tangki mengaktifkan alarm setelah 5 siklus clock untuk memicu pompa. Ketika level NORMAL (01) atau HIGH (11), alarm dimatikan. Output utama dari tangki adalah level air (Level_tank), status alarm (Alarm_tank), dan suhu air (Temp), yang memberi informasi penting tentang kondisi tangki.

```
28          if falling_edge(clock) then  -- Menggunakan falling edge clock
29              -- Update suhu (direct output dari temp_input)
30              temp <= temp_input;
31
32              -- Perbandingan level air dan kontrol alarm
33              case level is
34                  when "00" =>  -- LOW level
35                      current_level <= "00";  -- Set current level ke LOW
36                      -- Loop untuk mengaktifkan timer alarm
37                      for i in 0 to 5 loop
38                          if alarm_timer < 5 then
39                              alarm_timer <= alarm_timer + 1;
40                              alarm_tank <= '0';  -- Alarm mati selama timer belum mencapai 5 detik
41                          else
42                              alarm_tank <= '1';  -- Alarm menyala setelah 5 detik
43                          end if;
44                      end loop;
45                  when "01" =>  -- NORMAL level
46                      current_level <= "01";  -- Set current level ke NORMAL
47                      alarm_timer <= 0;  -- Reset timer
48                      alarm_tank <= '0';  -- Reset alarm
49                  when "11" =>  -- HIGH level
50                      current_level <= "11";  -- Set current level ke HIGH
51                      alarm_timer <= 0;  -- Reset timer
52                      alarm_tank <= '0';  -- Reset alarm
53                  when others =>
54                      -- Jika nilai level tidak valid, tetap di level rendah
55                      current_level <= "00";
56              end case;
```

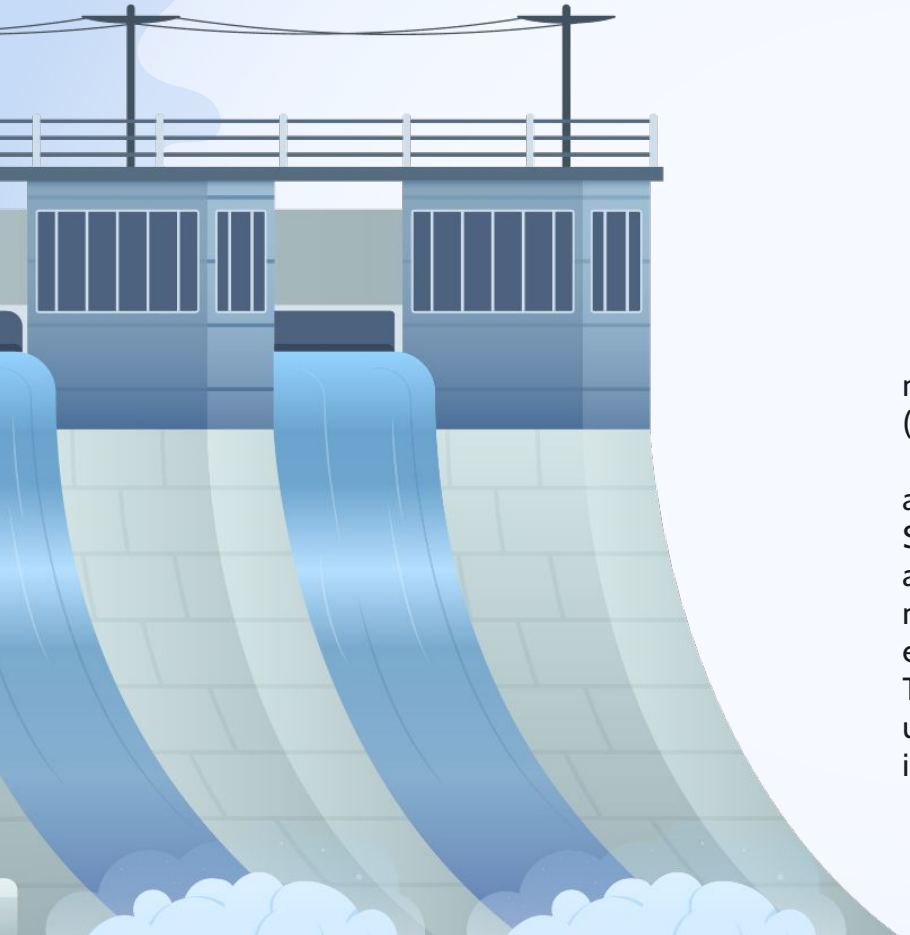
Fig 3. Potongan Source Code Tangki

Top-Level

Komponen Top-Level mengintegrasikan komponen Pompa dan Tangki untuk membangun sistem yang utuh. Alarm dari tangki diteruskan sebagai input ke pompa, sementara sinyal clock memastikan kedua komponen bekerja secara sinkron. Output dari Top-Level mencakup efisiensi dan debit air dari pompa, serta status alarm dan suhu dari tangki, yang memberikan informasi lengkap tentang sistem.

```
65        u_pompa : pompa
66        PORT MAP(
67            clk => clk,
68            alarm_pompa => alarm,
69            enable_pompa => enable_pompa,
70            pump_state => pump_state,
71            luas_pompa => luas_pompa,
72            kecepatan_air => kecepatan_air,
73            delivery_head => delivery_head,
74            efisiensi => efisiensi,
75            debit => debit,
76            kegiatan_pompa => kegiatan_pompa
77        );
78
79        u_tank : Tank
80        port map(
81            level => level,
82            temp_input => temp_input,
83            clock => clk,
84            level_tank => level_tank,
85            alarm_tank => alarm,
86            temp => temp
87        );
```

Fig 3. Potongan Source Code Top-level



05

Testing

Menggunakan software ModelSim untuk mensimulasikan desain VHDL dalam bentuk gelombang (wave).

Tangki diuji untuk memantau level air, membunyikan alarm pada kondisi kritis, dan memberi sinyal kepada pompa. Sementara itu, pompa diuji untuk memastikan respon terhadap alarm, yaitu memulai pengisian air hingga level air kembali normal atau lebih dari normal. Hasil perhitungan debit air dan efisiensi pompa dicatat dalam file .txt menggunakan library TEXTIO, memberikan data akurat setiap periode waktu tertentu untuk analisis lebih lanjut. Testing ini memastikan keberhasilan interaksi antara tangki dan pompa secara keseluruhan.

06

Result

Simulasi

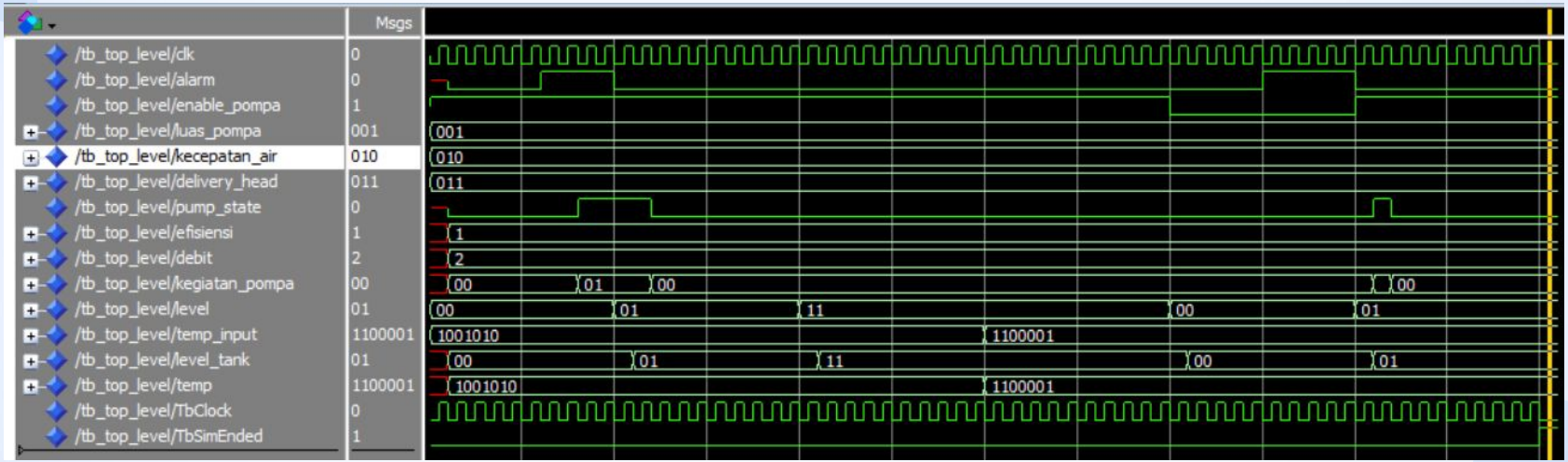
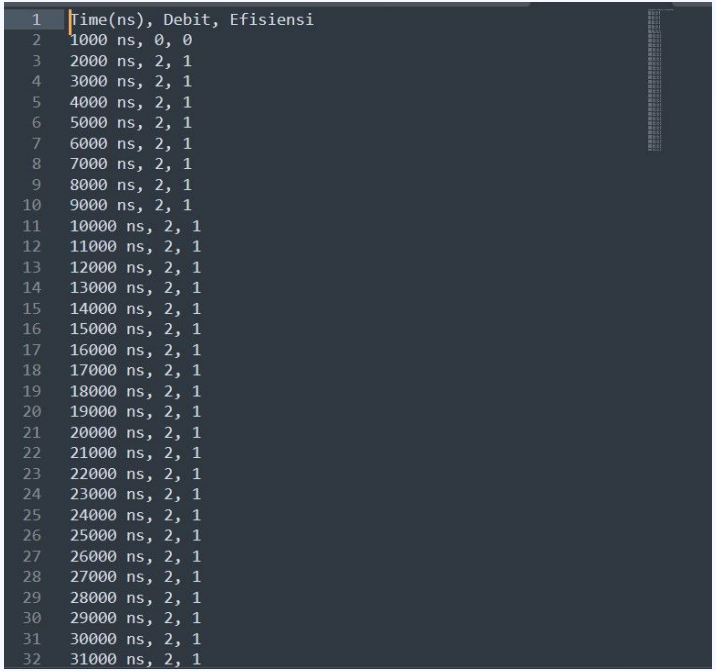


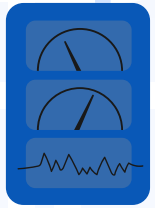
Fig 3. Testing Result

File .txt



```
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 .txt



Terima Kasih