DAFTAR PUSTAKA

[1] Kusuma H., dkk. 2021. Prototype Pendeteksi Kebocoran Pipa Berbasis IoT Menggunakan Nodemcu Esp8266 Melalui Dashboard Adafruit.Io. *Jurnal Teknologi dan Informasi Bisnis.* Vol 3, No 2.

[2] Sasmoko, D., Horman, R. 2020. Sistem Monitoring Aliran Air dan Penyiraman Otomatis pada Rumah Kaca Berbasis IoT dengan Esp8266 dan Blynk. *Jurnal Ilmiah Pendidikan Teknik Elektro*. Vol 4, No 1.

[3] Zulfikar, M. 2018. Perancangan Sistem Penyiraman Tanaman Otomatis Berbasis Mikrokontroler Atmega328. *Jurnal of Informatics and Computer Science.* Vol 4, No 1.

[4] Prasetyo, A.D., Haryanto., & Wibisono, K.A. 2020. Rancang Bangun Sistem Monitoring dan Pendeteksi Lokasi Kebocoran Pipa Berdasarkan Analisis Debit Air Berbasis IoT. *Jurnal Elektrika*. Vol 12, No 1, Hal 39-47.

[5] Marinus, Ferdinand., dkk. 2020. Rancang Bangun Sistem Penyiraman Tanaman Berdasarkan Waktu Menggunakan *RTC* Berbaisis Arduino Uno Pada Tanaman Tomat. *Jurnal Teknik Industri*.

[6] Darmawan, I.W.B., dkk. 2021. Smart Garden Sebagai Implementasi Sistem Kontrol dan Monitoring Tanaman Berbasis Teknologi Cerdas. *Jurnal SPEKTRUM.* Vol 8, NO 4.

[7] Ramadhani, F.E., Damayanti, T.N., Ramadan D.N. 2020. Rancang Bangun Penyiraman Tanaman Otomatis Berbasis Debit Air Menggunakan Mikrokontroler dengan *Realtime Database*. *e-Proceeding of Applied Science.* Vol 6, No 1.

[8] Hatopan, Abraham., Ayu, Nanda., & Indriani, Widya. 2019. Prototipe Pendeteksi Kebocoran pada Pipa Air. *Prosiding Seminar Nasional Teknik Elektro.* Vol 4.

[9] Febriani, O.M., dkk. 2020. Rancang Bangun Sistem Monitoring Sirkulasi Obat Pada Pedagang Besar Farmasi (PBF) Di Kota Bandar Lampung Berbasis Web. *Seminar Nasional Hasil Penelitian dan Pengabdian 2020.*

[10] Kadir, Abdul. (2017). *Pemograman Arduino & Android Menggunakan App Invertor*. Jakarta: PT Elex Media Komputindo.

[11] Antares. 2020. Pengenalan Mit App Invertor. https://antares.id/id/mitappinventor2.html. Diakses pada Tanggal 20 Juni 2022.
[12] Maulana, Ilham Firman. 2020. Penerapan *Firebase Realtime Database* pada aplikasi E-Tilang *Smartphone* Berbasis *Mobile* Android. *Jurnal RESTI*. Vol 4, No 5.
[13] Wahyuningsih, F.T., dkk. 2019. Pengembangan Alat Praga Pengukur Debit Air Menggunakan Sensor Flow Berbasis Arduino sebagai Media Pembelajaran Fluida. *RADIASI:Jurnal Berkala Pendidikan Fisika.* Vol 12, No 1.
[14] Efendi, M.Y., & Chandra, J.E. 2019. Implementasi *Internet of Things* pada Sistem Kendali Lampu Rumah Menggunakan Telegram *Messenger Bot* dan *Nodemcu* ESP 8266. *Global Journals.* Vol XIX.
[15] Dr. Junaidi, S.Si., M.Sc, Yuliyan Dwi Prabowo. 2018. Projek Sistem Kendali Elektronik Berbasis Arduino. Bandar Lampung : AURA, CV Anugrah Utama Raharja.
[16] Desnanjaya, I.G.M., Iswara I.B.A.I. 2018. *Trainer* Atmega32 Sebagai Media Pelatihan Mikrokontroler dan Arduino. Vol 1, No 1.
[17] Widianto, M.H. 2018. Pengaplikasian Sensor Hujan dan LDR untuk Lampu Mobil Otomatis Berbasis Arduino Uno. *RESISTOR*. Vol 1, No 2.
[18] Sutono., Nursoparisa, Asri. 2019. Perancangan Sistem Kendali Automatisasi *Control* Debit Air pada Pengisian Galon Menggunakan Modul Arduino. *Media Jurnal Informatika*. Vol 11, No 1.
[19] Martin, Kurnia., & Susandi, Dony. 2022. Perancangan dan Implementasi Sistem Irigasi Kabut Otomatis Tanaman Edelwis Menggunakan Mikrokontroler Arduino Uno. *Jurnal-IKRAITH-INFORMATIKA.* Vol 6, No 1.
[20] Hidayat, L.R., Sari, Zamah., & Nursandi, Fatimah. 2020. Implementasi Pemantauan Suhu Kelembaban serta Pengendali Penyiraman Tanaman secara Otomatis pada *Greenhouse* Berbasis Web. *Repositor.* Vol 2, No 4.
[21] Supriyanto, Hadi., Martawireja, A.R.H. 2021. Analisa Kontrol Aliran Fluida Bervikositas Tinggi dengan Sensor *Flow* YF-S201 pada Otomatisasi Dispenser Minyak Goreng. *Jurnal Teknologi Terapan.* Vol 7, No 1.

[22] Irsyam, Muhammad., Tanjung, Alamsyahzali. 2019. Sistem Otomasi Penyiraman Tanaman Berbasis Telegram. *Sigma Teknika*. Vol 2, No 1.
[23] Sulistyawan, Panca Muji. 2021. Perancangan Sistem Pemantauan Tekanan Darah Dengan Sensor Tekanan MPX5100GP Berbasis STM32f103. *Seminar Nasional Fortei 7-4.*

LAMPIRAN

LAMPIRAN A

Listing Program

1. *Listing* Program Arduino Mega 2560

```
// CONNECTIONS:
// DS3231 SDA --> SDA
// DS3231 SCL --> SCL
// DS3231 VCC --> 3.3v or 5v
// DS3231 GND --> GND

/* for software wire use below
#include <SoftwareWire.h>  // must be included here so that Arduino library object file references work
#include <RtcDS3231.h>
SoftwareWire myWire(SDA, SCL);
RtcDS3231<SoftwareWire> Rtc(myWire);
for software wire use above */

/* for normal hardware wire use below */
#include <Wire.h> // must be included here so that Arduino library object file references work
#include <RtcDS3231.h>
RtcDS3231<TwoWire> Rtc(Wire);
/* for normal hardware wire use above */

#define sensorhujan 7

volatile long flow_frequency1; // Measures flow sensor pulses
volatile long flow_frequency2; // Measures flow sensor pulses
volatile long flow_frequency3; // Measures flow sensor pulses
volatile long flow_frequency4; // Measures flow sensor pulses

// Calculated litres/hour
float vol1 = 0.0, l_minute1;
float vol2 = 0.0, l_minute2;
```

```
float vol3 = 0.0, l_minute3;
float vol4 = 0.0, l_minute4;

unsigned char flowsensor1 = 2; // Sensor Input
unsigned char flowsensor2 = 3; // Sensor Input
unsigned char flowsensor3 = 18; // Sensor Input
unsigned char flowsensor4 = 19; // Sensor Input

unsigned long currentTime;
unsigned long cloopTime;

void flow1 () // Interrupt function
{
  flow_frequency1++;
}
void flow2 () // Interrupt function
{
  flow_frequency2++;
}
void flow3 () // Interrupt function
{
  flow_frequency3++;
}
void flow4 () // Interrupt function
{
  flow_frequency4++;
}
void setup()
{
  Serial.begin(57600);
  Serial.print("compiled: ");
  Serial.print(__DATE__);
  Serial.println(__TIME__);

  //--------RTC SETUP ------------
 // if you are using ESP-01 then uncomment the line below to reset the
pins to
// the available pins for SDA, SCL
// Wire.begin(0, 2); // due to limited pins, use pin 0 and 2 for SDA, SCL
```

```
Rtc.Begin();
RtcDateTime compiled = RtcDateTime(__DATE__, __TIME__);
printDateTime(compiled);
Serial.println();
RtcDateTime now = Rtc.GetDateTime();
// never assume the Rtc was last configured by you, so
// just clear them to your needed state
Rtc.Enable32kHzPin(false);
Rtc.SetSquareWavePin(DS3231SquareWavePin_ModeNone);

pinMode(sensorhujan, INPUT);

pinMode(flowsensor1, INPUT_PULLUP);
pinMode(flowsensor2, INPUT_PULLUP);
pinMode(flowsensor3, INPUT_PULLUP);
pinMode(flowsensor4, INPUT_PULLUP);

attachInterrupt(digitalPinToInterrupt(flowsensor1), flow1, RISING); //
Setup Interrupt
attachInterrupt(digitalPinToInterrupt(flowsensor2), flow2, RISING); //
Setup Interrupt
attachInterrupt(digitalPinToInterrupt(flowsensor3), flow3, RISING); //
Setup Interrupt
attachInterrupt(digitalPinToInterrupt(flowsensor4), flow4, RISING); //
Setup Interrupt

currentTime = millis();
cloopTime = currentTime;
}
void loop () {
RtcDateTime now = Rtc.GetDateTime();
int jam   = now.Hour();
int menit = now.Minute();
int detik = now.Second();
int kondisisensor = digitalRead(sensorhujan);

currentTime = millis();
 // Every second, calculate and print litres/hour
 if (currentTime >= (cloopTime + 1000))
```

```
{
    cloopTime = currentTime; // Updates cloopTime
    if (flow_frequency1 != 0) {
       // Pulse frequency (Hz) = 7.5Q, Q is flow rate in L/min.
       l_minute1 = (flow_frequency1 / 7.5); // (Pulse frequency x 60 min)
       / 7.5Q = flowrate in L/hour
       l_minute1 = l_minute1 / 60;
       vol1 = vol1 + l_minute1;
       flow_frequency1 = 0; // Reset Counter
    }
    else {
       l_minute1 = 0;
       vol1      = 0;
    }
    if (flow_frequency2 != 0) {
       // Pulse frequency (Hz) = 7.5Q, Q is flow rate in L/min.
       l_minute2 = (flow_frequency2 / 7.5); // (Pulse frequency x 60 min)
       / 7.5Q = flowrate in L/hour
       l_minute2 = l_minute2 / 60;
       vol2 = vol2 + l_minute2;
       flow_frequency2 = 0; // Reset Counter
    }
    else {
       l_minute2 = 0;
       vol2      = 0;
    }
    if (flow_frequency3 != 0) {
       // Pulse frequency (Hz) = 7.5Q, Q is flow rate in L/min.
l_minute3 = (flow_frequency3 / 7.5); // (Pulse frequency x 60 min) / 7.5Q
= flowrate in L/hour
l_minute3 = l_minute3 / 60;
     vol3 = vol3 + l_minute3;
     flow_frequency3 = 0; // Reset Counter
   }
   else {
      l_minute3 = 0;
      vol3      = 0;
   }
   if (flow_frequency4 != 0) {
```

```
      // Pulse frequency (Hz) = 7.5Q, Q is flow rate in L/min.
l_minute4 = (flow_frequency4 / 7.5); // (Pulse frequency x  60 min) /
7.5Q = flowrate in L/hour
      l_minute4 = l_minute4 / 60;
      vol4 = vol4 + l_minute4;
      flow_frequency4 = 0; // Reset Counter
   }
   else {
      l_minute4 = 0;
      vol4     = 0;
   }
 }
 String minta = "";
 while (Serial.available() > 0) {
     minta += char (Serial.read());
 }
 minta.trim();
 if (minta == "Ya") {
    String datakirim = String (jam) + "#" + String(menit) + "#" +
    String(detik) + "#" + String(kondisisensor) + "#" + String(l_minute1)
    + "#" + String(vol1) + "#" + String(l_minute2) + "#" + String(vol2) +
    "#"  +  String(l_minute3)  +  "#"  +  String(vol3)  +  "#"  +
    String(l_minute4) + "#" + String(vol4);
    Serial.println(datakirim);
}
 minta = "";
 delay(1000);
}

#define countof(a) (sizeof(a) / sizeof(a[0]))

void printDateTime(const RtcDateTime& dt)
{
 char datestring[20];
 snprintf_P(datestring,
          countof(datestring),
          PSTR("%02u/%02u/%04u %02u:%02u:%02u"),
          dt.Month(),
          dt.Day(),
```

```
          dt.Year(),
          dt.Hour(),
          dt.Minute(),
          dt.Second() );
 Serial.print(datestring);
            }
```

2. *Listing* Program *NodeMCU* ESP8266

```
#include <ESP8266WiFi.h>
#define WIFI_SSID "Mi10TPro"
#define WIFI_PASSWORD "Akmal260701"

#include <SoftwareSerial.h>
SoftwareSerial DataSerial(12, 13);

#include <FirebaseArduino.h>
#define FIREBASE_HOST "tugas-akhir-sidang-default-
rtdb.firebaseio.com"
#define FIREBASE_AUTH
"qiFPg4QeuRYCjHXNfscz49cYxEPK9DsqFiG0cWpN"

unsigned long previousMillis = 0;
const long interval = 1000;
String arrData[12];

String jam, menit,
detik,rain,debit1,volume1,debit2,volume2,debit3,volume3,debit4,volum
e4;

#define relay1 0
#define relay2 2
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 20, 4);
void setup() {
Serial.begin(57600);
DataSerial.begin(57600);
lcd.begin();
 // connect to wifi.
WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
```

```
while (WiFi.status() != WL_CONNECTED) {
delay(500);
Serial.println("Koneksi Terputus");
lcd.setCursor(2, 1);
lcd.print("Koneksi Terputus");
}
Serial.println("Koneksi Terhubung");
lcd.setCursor(1, 1);
lcd.print("Koneksi  Terhubung");
Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);

pinMode(relay1, OUTPUT);
pinMode(relay2, OUTPUT);
digitalWrite(relay1, HIGH);
digitalWrite(relay2, HIGH);
}
void loop() {
unsigned long currentMillis = millis();

if (currentMillis - previousMillis >= interval) {
   previousMillis = currentMillis;

 String data = "";
 while (DataSerial.available() > 0) {
      data += char (DataSerial.read());
}
data.trim();
if (data != "") {
int index = 0;
     for (int i = 0; i <= data.length(); i++) {
          char delimiter = '#';
     if (data[i] != delimiter)
        arrData [index] += data[i];
     else
        index++;
     }
     if (index == 11) {
          Serial.print(arrData[0]);
          Serial.print(":");
```

```
    Serial.print(arrData[1]);
    Serial.print(":");
    Serial.println(arrData[2]);
    Serial.println(arrData[3]);
    Serial.println(arrData[4]);
    Serial.println(arrData[5]);
    Serial.println(arrData[6]);
    Serial.println(arrData[7]);
    Serial.println(arrData[8]);
    Serial.println(arrData[9]);
    Serial.println(arrData[10]);
    Serial.println(arrData[11]);
}
jam     = arrData[0];
menit   = arrData[1];
detik   = arrData[2];
rain    = arrData[3];
debit1  = arrData[4];
volume1 = arrData[5];
debit2  = arrData[6];
volume2 = arrData[7];
debit3  = arrData[8];
volume3 = arrData[9];
debit4  = arrData[10];
volume4 = arrData[11];

Firebase.setString("Jam", jam);
Firebase.setString("Menit", menit);
Firebase.setString("Detik", detik);
Firebase.setString("Kondisi hujan", rain);
Firebase.setString("Debit 1", debit1);
Firebase.setString("Volume 1", volume1);
Firebase.setString("Debit 2", debit2);
Firebase.setString("Volume 2", volume2);
Firebase.setString("Debit 3", debit3);
Firebase.setString("Volume 3", volume3);
Firebase.setString("Debit 4", debit4);
Firebase.setString("Volume 4", volume4);
```

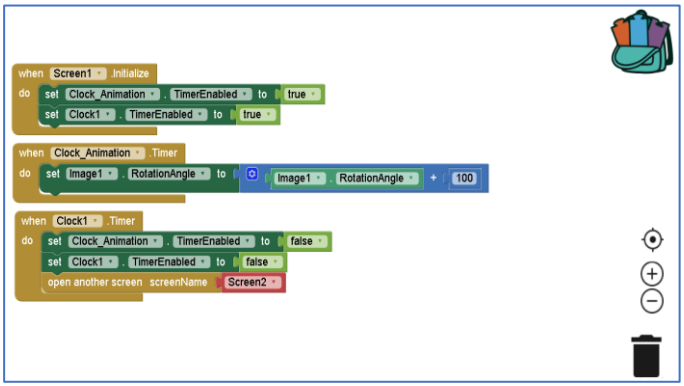
```
    if (Firebase.failed()) {
       Serial.println(Firebase.error());
       return;
    }
    arrData[0] = "";
    arrData[1] = "";
    arrData[2] = "";
    arrData[3] = "";
    arrData[4] = "";
    arrData[5] = "";
    arrData[6] = "";
    arrData[7] = "";
    arrData[8] = "";
    arrData[9] = "";
    arrData[10] = "";
    arrData[11] = "";
   }
  DataSerial.println("Ya");
}
String j1 = Firebase.getString("jam1");
String m1 = Firebase.getString("menit1");

String set1 = String(j1) + ":" + String(m1);
lcd.setCursor(3, 0);
lcd.print(set1);

String j2 = Firebase.getString("jam2");
String m2 = Firebase.getString("menit2");

String set2 = String(j2) + ":" + String(m2);
lcd.setCursor(12, 0);
lcd.print(set2);

if ((jam == j1 & menit == m1)||(jam == j2 & menit == m2)) {
    if (rain == "1") {
       digitalWrite(relay1, LOW);
       digitalWrite(relay2, LOW);
    }
 else if (rain == "0") {
```

```
      digitalWrite(relay1, HIGH);
      digitalWrite(relay2, HIGH);
    }
   }
   else if ((menit != m1)||(menit != m2)) {
      digitalWrite(relay1, HIGH);
      digitalWrite(relay2, HIGH);
  }
}
```

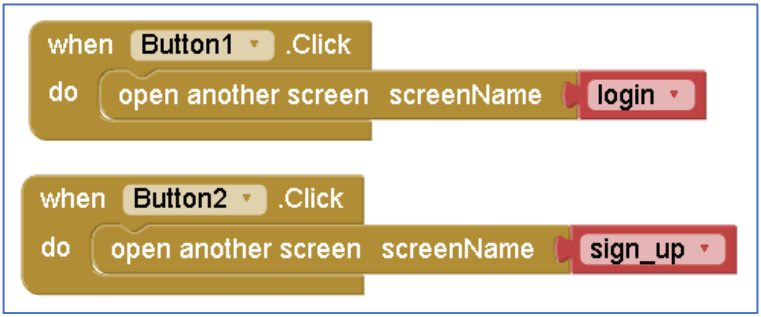
LAMPIRAN B

BLOK PUZZLE APLIKASI ANDROID

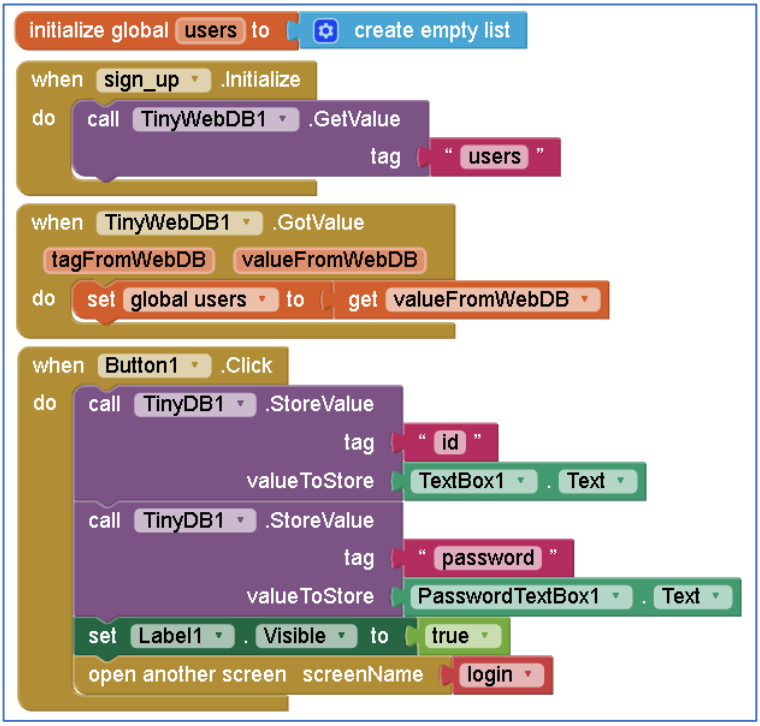
Screen 1 *loading* Aplikasi



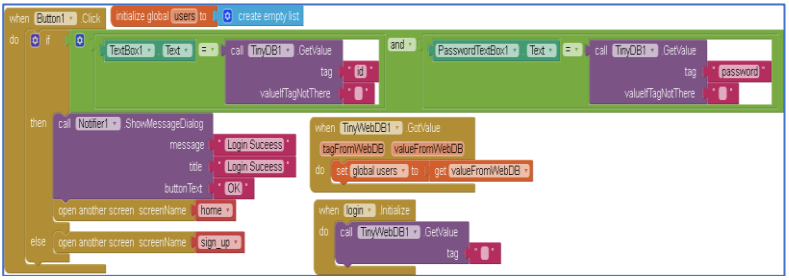
Screen 2 Halaman *Login* dan *Sign Up*



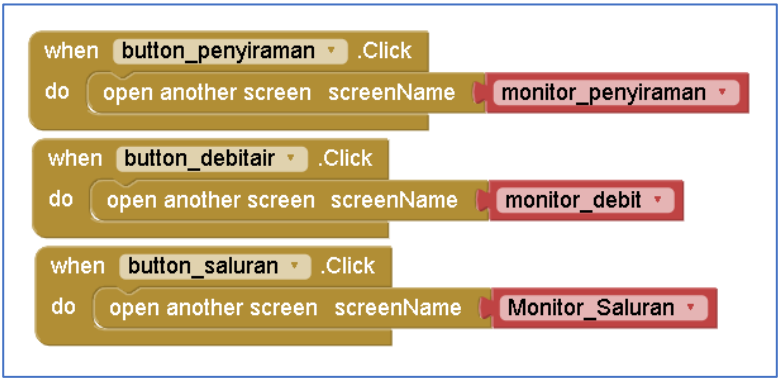
Screen 3 Halaman Pendaftaran Akun



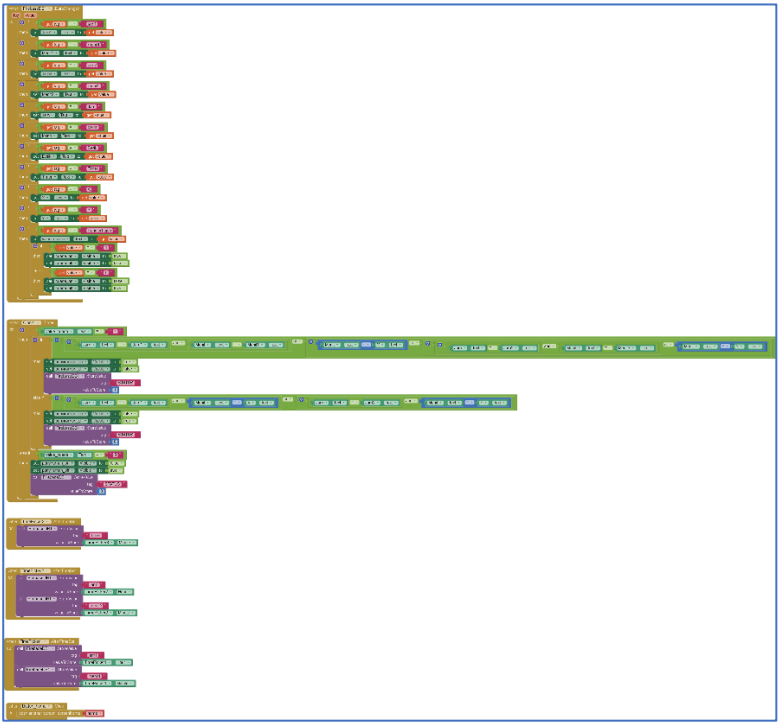
Screen 4 Halaman *Login*



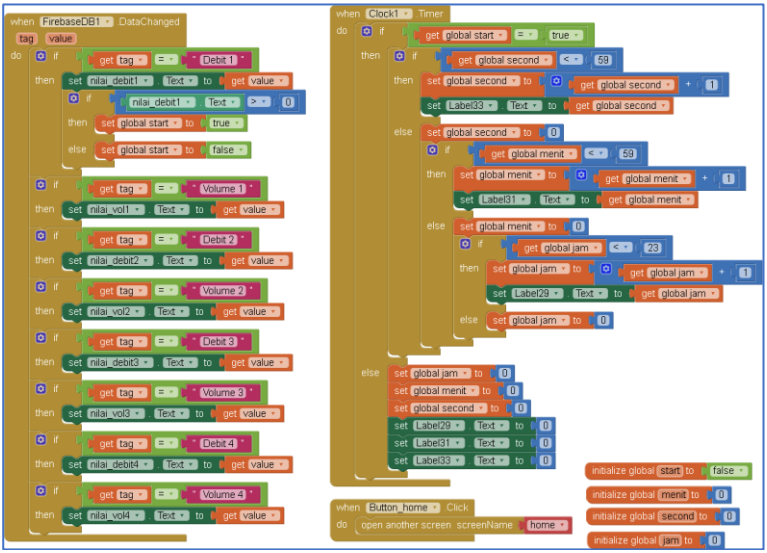
Screen 5 *Menu Home*



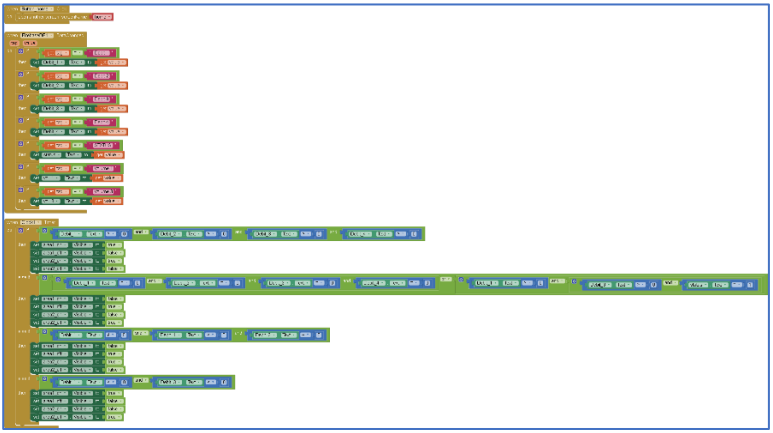
Screen 6 *Monitoring* Penyiraman



Screen 7 Monitoring Debit dan Volume



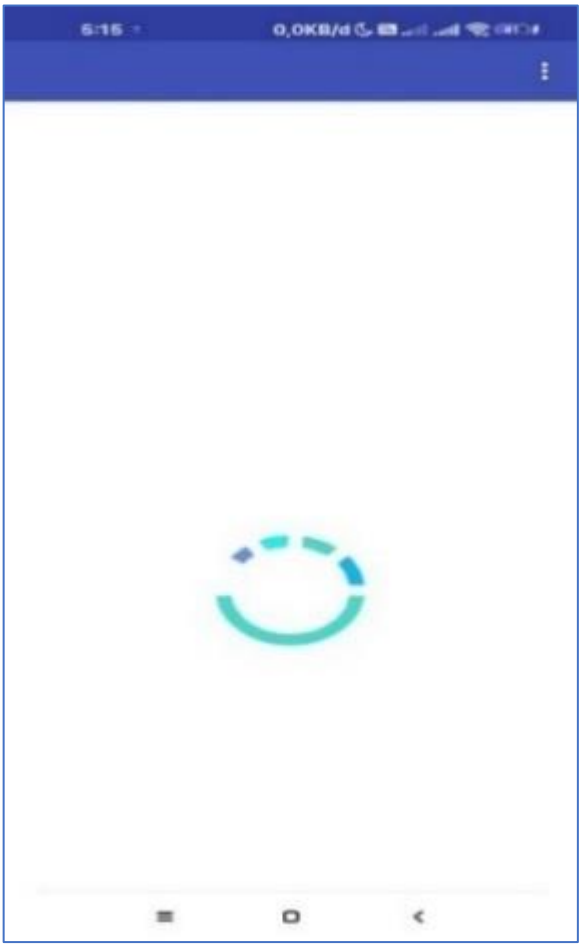
Screen 8 *Monitoring* Saluran Pipa



LAMPIRAN C

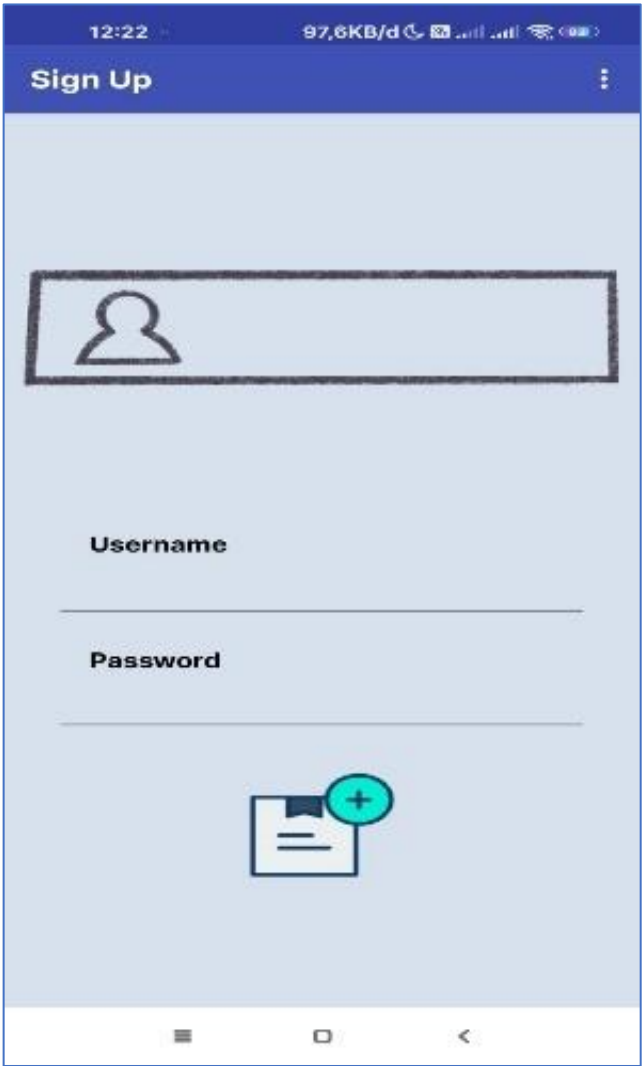
TAMPILAN APLIKASI ANDROID

Tampilan 1 *Loading* Aplikasi

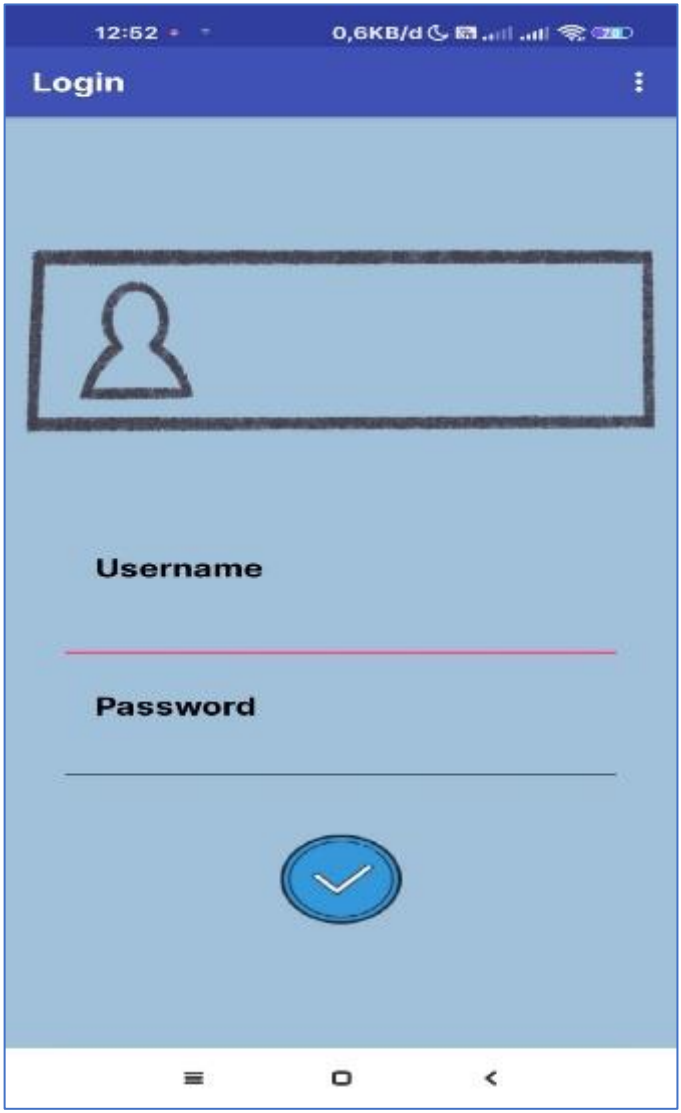


Tampilan 2 *Login* dan *Sign Up*

Tampilan 3 Pendaftaran Akun

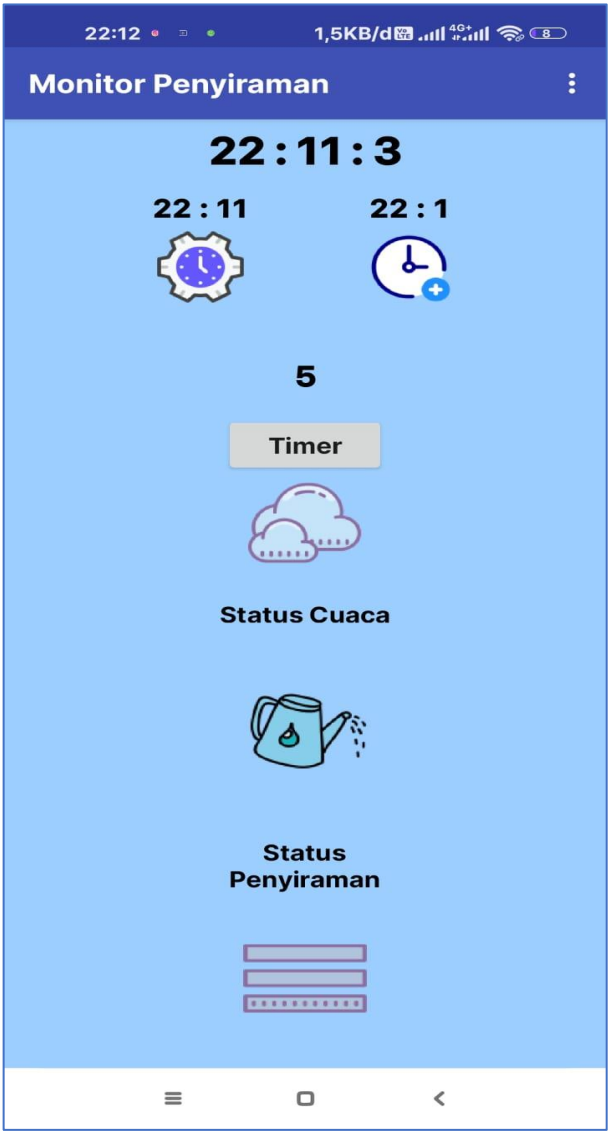


Tampilan 4 *Login*

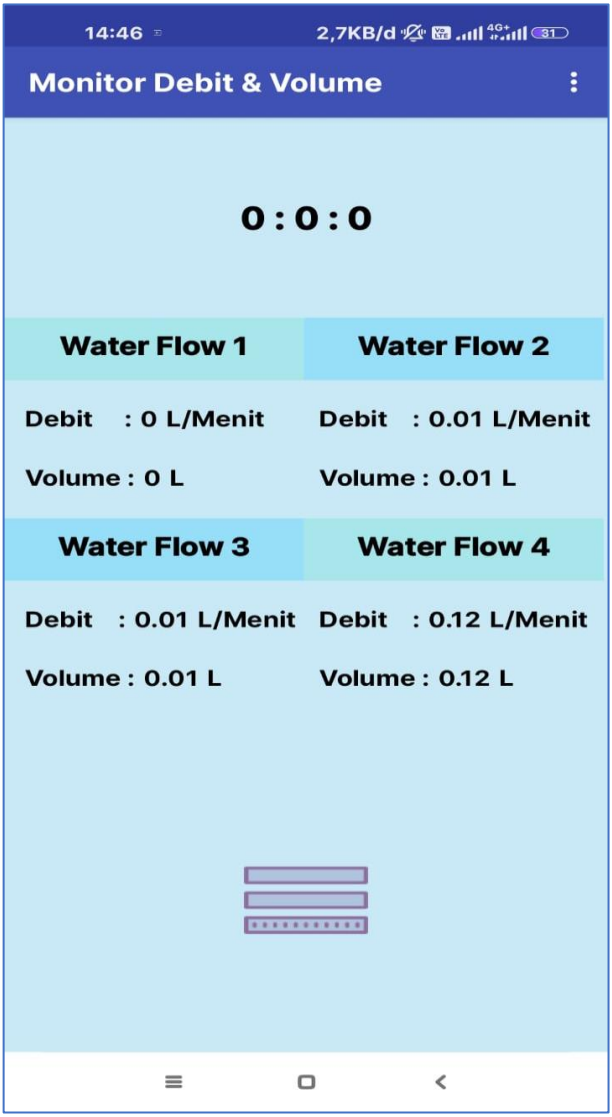


Tampilan 5 *Menu Home*

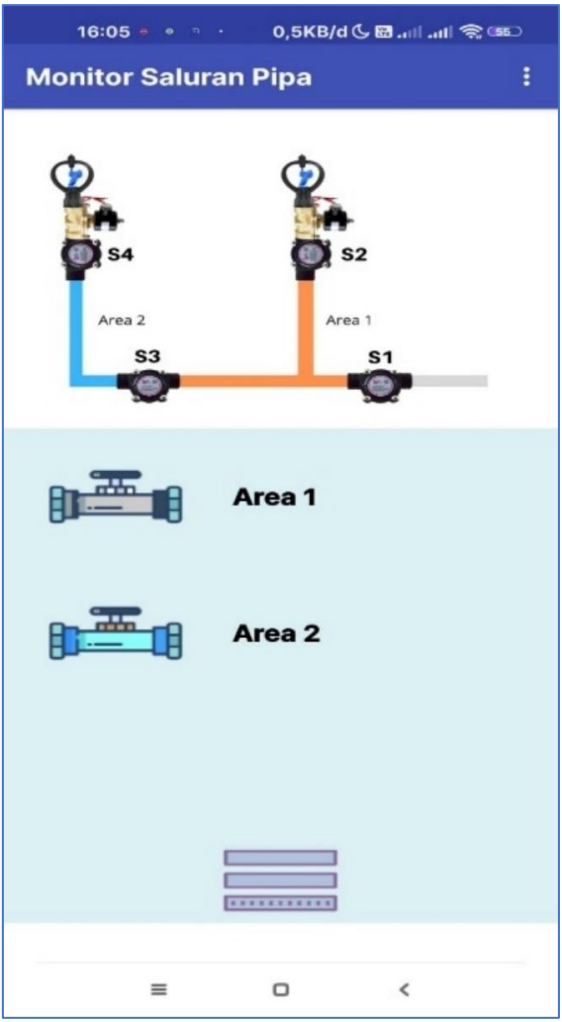
Tampilan 6 *Monitor* Penyiraman



Tampilan 7 *Monitor* Debit dan Volume



Tampilan *Monitor* Saluran Pipa



LAMPIRAN D

HASIL MEKANIK TUGAS AKHIR