

BAB V

PENUTUP

V.1 Kesimpulan

Berdasarkan hasil implementasi dan pengujian yang telah dilakukan terhadap sistem *Monitoring* dan kendali berbasis *Human Machine Interface (HMI)* pada mesin *Roasting rotary drum* kopi kapasitas 3 kg, dapat disimpulkan bahwa sistem yang dirancang telah mampu berfungsi sesuai dengan tuntutan dan spesifikasi yang ditetapkan. Seluruh komponen penting, mulai dari sensor suhu *thermocouple* tipe K yang dikonversi oleh *Thermocontrol* Azbil SDC-15, mikrokontroler Arduino Mega dan *ESP32*, hingga antarmuka HMI Nextion, berhasil diintegrasikan dan diuji secara fungsional.

Pengujian awal menunjukkan bahwa sebelum dilakukan kalibrasi, pembacaan suhu memiliki deviasi rata-rata hingga lebih dari 22°C dari alat referensi, mengindikasikan perlunya perbaikan akurasi. Setelah diterapkan metode kalibrasi berbasis koreksi linier dan non-linier, serta dilengkapi filter digital seperti averaging dan Exponential Moving Average (EMA), pembacaan suhu pada *HMI* menunjukkan peningkatan signifikan dengan error kurang dari $\pm 0.1^\circ\text{C}$ terhadap suhu referensi. Hal ini menunjukkan bahwa proses kalibrasi berhasil meningkatkan ketepatan dan kestabilan sistem *Monitoring* secara signifikan.

Dari sisi fungsi antarmuka, halaman *Monitoring* pada *HMI* telah menampilkan suhu drum, suhu bean, serta waktu proses *Roasting* secara *Real-time* dengan akurasi *visual* yang baik. Tombol kontrol seperti *Start*, *Pause*, *Delete*, dan *Send* juga berfungsi dengan baik selama pengujian, mendukung kebutuhan interaksi operator selama proses *Roasting* manual. Selain itu, sistem ini juga mampu mencatat dan mengirimkan data ke *Firestore Realtime Database* dengan interval stabil dan sinkron dengan waktu aktual, serta dapat diekspor ke *Microsoft Excel* untuk dokumentasi *Roasting*. Keberhasilan sistem ini juga terlihat dari kecocokan pola data grafik *Roasting* yang dihasilkan dengan data grafik dari perangkat lunak profesional *Artisan*, yang digunakan sebagai alat pembanding dan validasi terhadap

akurasi sistem. Hal ini membuktikan bahwa sistem yang dibangun mampu merepresentasikan dinamika proses *Roasting* kopi secara nyata dan responsif.

Selain berfungsi sebagai alat bantu *Monitoring*, sistem ini juga telah mengimplementasikan pengiriman parameter PID (K_p , K_i , K_d) secara manual dari *HMI* ke mikrokontroler dan Firebase, meskipun belum diintegrasikan dengan algoritma kendali aktif. Efisiensi waktu proses *Roasting* juga menunjukkan peningkatan yang nyata, di mana sistem mampu mencatat dan menyajikan durasi proses secara tepat dan stabil, sehingga mendukung percepatan dan konsistensi proses produksi dari satu batch ke batch berikutnya. Hal ini membuktikan bahwa sistem mampu meningkatkan efektivitas kerja operator sekaligus mengurangi potensi variasi yang tidak diinginkan dalam waktu penyangraian.

Dengan seluruh keberhasilan tersebut, sistem ini membuktikan bahwa pendekatan desain terintegrasi lintas domain mekanik, elektronik, informatika, dan sistem pengendalian dapat melahirkan solusi teknologi yang fungsional, presisi, dan siap diimplementasikan dalam proses produksi nyata. Sistem ini sangat potensial untuk digunakan oleh pelaku industri kecil dan menengah sebagai alat bantu produksi yang mampu meningkatkan mutu, konsistensi, dan efisiensi produk kopi secara digital dan terukur.

V.2 Saran

Pengembangan sistem ini masih memiliki ruang yang luas untuk ditingkatkan di masa depan. Salah satu upaya yang dapat dilakukan adalah menambahkan fitur pengendalian aktuator secara otomatis berbasis PID, sehingga sistem tidak hanya terbatas pada fungsi *Monitoring*, namun mampu melakukan kendali suhu secara mandiri. Penggunaan sensor tambahan seperti sensor kelembapan, tekanan, atau bahkan deteksi gas hasil pembakaran juga dapat memperkaya data proses dan memberikan insight yang lebih komprehensif terhadap profil *Roasting*. Selain itu, pengembangan aplikasi *Monitoring* berbasis web atau mobile akan sangat bermanfaat untuk memperluas akses dan kemudahan pemantauan dari jarak jauh. Implementasi sistem ini juga dapat diperluas pada mesin *Roasting* dengan kapasitas yang lebih besar, tentunya dengan penyesuaian pada sistem kelistrikan, dimensi panel, dan suplai daya. Dengan berbagai

pengembangan tersebut, sistem ini berpotensi menjadi pondasi bagi otomasi industri kopi yang lebih modern dan terstandarisasi.

DAFTAR PUSTAKA

- [1] G. A. Arumsari, R. Surya, S. Irmasuryani, and W. Sapitri, "Analisis Proses Roasting pada Kopi," *J. Beta Kim.*, vol. 1, no. 2, pp. 98–101, 2021, [Online]. Available: <http://ejurnal.undana.ac.id/index.php/jbkHalaman%7C98>
- [2] H. Yulindoko, I. W. Suardinata, R. A. Julianto, and R. C. Abiandsa, "Analisis Suhu Optimum Roasting Kopi Lokal Banyuwangi Dengan Monitoring Suhu Berbasis Arduino," *Semin. Nas. Terap. Ris. Inov. Ke-6 ISAS Publ. Ser. Eng. Sci.*, vol. 6, no. 1, pp. 1086–1094, 2020, [Online]. Available: <https://proceeding.isas.or.id/index.php/sentrinov/article/view/580>
- [3] A. Rahmandhika, E. D. Defantyan, and V. T. Lutfi, "J-Proteksion : Jurnal Kajian Ilmiah dan Teknologi Teknik Mesin," *J-Proteksion*, vol. 4, no. 13, pp. 1–6, 2024, doi: 10.32528/jp.v9i1.1040.
- [4] B. Fatkhurrozi, H. T. Setiawan, and M. N. Abdillah, "Pengembangan Alat Ukur Kadar Air Biji Kopi Berbasis Sensor SHT11 untuk Meningkatkan Kualitas Hasil Olahan Kopi Desa Ketep," *Kontribusi J. Penelit. dan Pengabd. Kpd. Masy.*, vol. 4, no. 2, pp. 225–236, 2024, doi: 10.53624/kontribusi.v4i2.358.
- [5] Y. Sarvina, T. June, E. Surmaini, R. Nurmalina, and S. S. Hadi, "Strategi Peningkatan Produktivitas Kopi serta Adaptasi terhadap Variabilitas dan Perubahan Iklim melalui Kalender Budidaya," *J. Sumberd. Lahan*, vol. 14, no. 2, p. 65, 2020, doi: 10.21082/jsdl.v14n2.2020.65-78.
- [6] M. D. Fahmi, "Sistem Monitoring Dan Kendali Proses Sampel," pp. 1–57, 2021.
- [7] Solikhin, P. A. Wicaksono, and A. W. B. Santoso, "Penerapan Teknologi Tepat Guna Mesin Roasting Kopi Pada Ukm Kopi Pinanggih," *J. Pasopati*, vol. 5, no. 3, pp. 138–143, 2023, [Online]. Available: <http://ejournal2.undip.ac.id/index.php/pasopati>
- [8] N. P. Lestari, D. Purbasari, I. Taruna, and M. Hasan, "Peningkatan Produktivitas dan Kualitas dengan Modernisasi Mesin Roasting di UKM

- ‘Bromo Coffee’-Probolinggo,” *KIAT J. Community Dev.*, vol. 1, no. 2, pp. 76–81, 2022, [Online]. Available: <https://kiatjcd.com/ojs/index.php/kjcd/article/view/39%0Ahttps://kiatjcd.com/ojs/index.php/kjcd/article/download/39/18>
- [9] Alief Muhammad, Dani Hari Tunggal Prasetyo, Hartawan Abdillah, Mas Ahmad Baihaqi, Hermanto, and Ahmad Iskandar Rahmansyah, “Teknologi Artisan Coffee Roasting Pada Pengusaha Café Probolinggo Dalam Upaya Menghasilkan Kopi Lokal Berkualitas,” *TEKIBA J. Teknol. dan Pengabd. Masy.*, vol. 3, no. 2, pp. 1–9, 2023, doi: 10.36526/tekiba.v3i2.3069.
- [10] A. Lubis, S. Syafriandi, M. Idkham, and A. Maulana, “Design and construction of coffee roasting machine with rounding cylinder tube using electric heat source,” *Res. Agric. Eng.*, vol. 69, no. 3, pp. 118–123, 2023, doi: 10.17221/69/2022-RAE.
- [11] H. Mukhtar *et al.*, “Implementasi Mesin Roasting Kopi Untuk Peningkatan Kualitas Produksi Kopi UMKM Darma Coffee,” *SWAGATI J. Community Serv.*, vol. 1, no. 3, pp. 127–132, 2024, doi: 10.24076/swagati.2023v1i3.1082.
- [12] R. Arifuddin, I. Mujahidin, S. Subairi, and R. Wikantiyoso, “Sistem Kontrol Suhu dan Waktu Otomatis Mesin Roasting Kopi Portabel,” *Cyclotron*, vol. 4, no. 2, pp. 4–7, 2021, doi: 10.30651/cl.v4i2.6517.
- [13] H. M. K. K. M. B. Herath, S. V. A. S. H. Ariyathunge, and H. D. N. S. Priyankara, “Development of a Data Acquisition and Monitoring System Based on MODBUS RTU Communication Protocol,” *Int. J. Innov. Sci. Res. Technol.*, vol. 5, no. 6, pp. 433–440, 2020, doi: 10.38124/ijisrt20jun479.
- [14] R. Manurung, O. I. A. Nugroho, and E. Apriliyanto, “Pelatihan Penggunaan Mesin Roasting Modern dalam Pengelolaan Kopi Arabika pada Kelompok Swadaya Masyarakat Galuh Lestari,” *J. Abdidas*, vol. 1, no. 5, pp. 471–477, 2020, doi: 10.31004/abdidas.v1i5.102.
- [15] S. Aminah, N. Faizin, and A. Mahardiyanto, “Implementasi *Rotary* Dryer dan Website Guna Meningkatkan Pendapatan pada Kelompok Tani Kopi Sumber Kembang,” *J. Community Dev.*, vol. 3, no. 2, pp. 97–105, 2022,

- doi: 10.47134/comdev.v3i2.74.
- [16] A. B. Saptomo *et al.*, “Alamat : Jl. Ir. Sutami 36 A Ketingan Surakarta,” pp. 65–72.
- [17] I. Artikel, “Pengadaan Mesin Pengering Biji Kopi Tipe Drum *Rotary* Di Desa Buttu Bartong Lama Kecamatan Raya Kabupaten Simalungun,” vol. 5, no. 3, pp. 3224–3232, 2024.
- [18] J. R. Material and M. Energi, “Perancangan Mesin Penggiling Kopi Dan Penakar Bubuk Kopi Untuk Usaha Mikro Kecil Menengah,” *J. Rekayasa Mater. Manufaktur dan Energi*, vol. 6, no. 2, pp. 267–279, 2023, doi: 10.30596/rmme.v6i2.16316.
- [19] D. Farida, “Penerapan Konsep Monitoring Dan Evaluasi Dalam Sistem Informasi Kegiatan Mahasiswa Di Perbanas Institute Jakarta,” *J. Tek. Inform. dan Sist. Inf.*, vol. 4, pp. 2443–2229, 2018, [Online]. Available: <http://dx.doi.org/10.28932/jutisi.v4i3.876>
- [20] A. Sari and F. Jusmi, “Perancangan sistem kontrol pid dengan aplikasi scilab,” *APCP (Applied Phys. Cokroaminoto Palopo)*, vol. 1, pp. 31–41, 2020.
- [21] “Apa Itu Human Machine Interface, Fungsi, dan Cara Kerja - Delta Mitra Solusindo.”
- [22] A. Iskandar, M. Muhajirin, and L. Lisah, “Sistem Keamanan Pintu Berbasis Arduino Mega,” *J. Inform. Upgris*, vol. 3, no. 2, pp. 99–104, 2017, doi: 10.26877/jiu.v3i2.1803.
- [23] B. Operation, “Single Loop Controller User ’ s Manual”.
- [24] A. ArjunPratikto, “Simulasi Kendali Dan Monitoring Daya Listrik Peralatan Rumah Tangga Berbasis ESP32,” *ALINIER J. Artif. Intell. Appl.*, vol. 3, no. 1, pp. 38–48, 2022, doi: 10.36040/aliner.v3i1.4855.
- [25] G. W. Bluetooth and B. Le, “ESP32 Series,” 2023.
- [26] M. Ben Aissa, S. Ferjani, M. S. Abassi, N. Al-Suwailem, and I. Boutiba, “Characterization of *Escherichia coli* Cefotaxime-Resistance in Al-Ahsa, KSA: Predominance of CTX-15 and First Report of blaCMY-42 Gene,” *Appl. Sci.*, vol. 12, no. 19, 2022, doi: 10.3390/app12199964.
- [27] K. T. Ngiam, “Constructing an air quality detection system using

- embedded system,” 2023, [Online]. Available:
http://eprints.utar.edu.my/6076/%0Ahttp://eprints.utar.edu.my/6076/1/Ngiam_Kee_Tiong_2101924.pdf
- [28] M. Fournelle *et al.*, “Portable ultrasound research system for use in automated bladder monitoring with machine-learning-based segmentation,” *Sensors*, vol. 21, no. 19, 2021, doi: 10.3390/s21196481.
- [29] V. C. Andrés-Valeri, L. A. Sañudo-Fontaneda, C. Rey-Mahía, S. J. Coupe, and F. P. Alvarez-Rabanal, “Thermal Performance of Wet Swales Designed as Multifunctional Green Infrastructure Systems for Water Management and Energy Saving,” p. 1433, 2018, doi: 10.3390/proceedings2231433.
- [30] A. I. Ferdiansyah, “Coding Jadi Lebih Mudah dan Asik dengan Visual Studio Code,” 2021. [Online]. Available:
<https://www.gamelab.id/news/559-coding-jadi-lebih-mudah-dan-asyik-dengan-visual-studio-code>
- [31] M. H. Ramadhan and M. Fathurahman, “Perancangan Sistem Monitoring Performansi Suhu Mesin Roasting Kopi Berbasis Internet Of Things Dengan Notifikasi Telegram,” vol. 3, no. 1, pp. 66–74, 2024.
- [32] Taufik Rahman, Retnaningtyas Susanti, Nidia Wulansari, and Hendri Azwar, “Analisis Cita Rasa Kopi di Tilik Kopi Kota Padang,” *J. Kaji. Pariwisata Dan Perhotelan*, vol. 1, no. 3, pp. 82–87, 2024, doi: 10.62379/jkph.v1i3.728.
- [33] M. Allmond Yani, E. Efrina, and R. Ridawati, “Pengaruh Perbedaan Ukuran Gilingan Terhadap Seduhan Kopi Arabika Batumirah Dengan Teknik Vietnam Drip,” *J. Pangan dan Agroindustri*, vol. 10, no. 2, pp. 93–101, 2022, doi: 10.21776/ub.jpa.2022.010.02.4.

LAMPIRAN

Lampiran 1. Program Arduino IDE

```
// === Pin Konfigurasi ===
#define SUHU1_PIN A0          // SDC15 ET (atas)
#define SUHU2_PIN A1          // SDC15 BT (bawah)

// === PID String input dari HMI ===
String inputString = "";
bool stringComplete = false;
float kp = 0.0;
float ki = 0.0;
float kd = 0.0;

// === Variabel Global untuk Filter Suhu (EMA) ===
float filteredSuhuET = 0.0;
float filteredSuhuBT = 0.0;

// === Konfigurasi Averaging ===
#define NUM_TEMP_SAMPLES 64 // Jumlah sampel untuk rata-rata,
seperti teman Anda

// === KONVERSI SUHU ===
const float minVoltage = 1.2;
const float maxVoltage = 4.8;
const float minTemp = 0.0;
const float maxTemp = 400.0;

// === Variabel Timer untuk HMI dan ESP32/Serial Monitor ===
unsigned long lastHMIUpdateTime = 0;
const unsigned long HMI_UPDATE_INTERVAL = 1000; // Perbarui HMI
setiap 1 detik (1000 ms)

unsigned long lastESP32SendTime = 0;
const unsigned long ESP32_SEND_INTERVAL = 5000; // Kirim ke ESP32
dan Serial Monitor setiap 5 detik (5000 ms)

void setup() {
  Serial.begin(9600);          // Debug ke Serial Monitor
  Serial1.begin(9600);        // Serial1 untuk Nextion HMI
  Serial2.begin(9600);        // UART2 ke ESP32
  inputString.reserve(100);
  // Tidak perlu delay(500) di setup jika menggunakan millis() di
loop
}
```

```

void loop() {
  // === Baca Suhu dan Terapkan Kalibrasi & EMA (berjalan terus-
  menerus) ===
  // Ini memastikan nilai filteredSuhuET dan filteredSuhuBT selalu
  terbaru
  float tempFromSensorET = bacaSuhu(SUHU1_PIN);
  float tempFromSensorBT = bacaSuhu(SUHU2_PIN);

  tempFromSensorET = 0.62960 * tempFromSensorET;
  tempFromSensorBT = 0.62887 * tempFromSensorBT;

  if (tempFromSensorET > 0 && tempFromSensorET < 140)
tempFromSensorET -= mapFloat(tempFromSensorET, 0, 140, 1.35, 0);
  if (tempFromSensorBT > 0 && tempFromSensorBT < 140)
tempFromSensorBT -= mapFloat(tempFromSensorBT, 0, 140, 0.85, 0);
  if (tempFromSensorET > 180 && tempFromSensorET < 250)
tempFromSensorET += mapFloat(tempFromSensorET, 0, 260, 2.9, 0);
  if (tempFromSensorBT > 180 && tempFromSensorBT < 250)
tempFromSensorBT += mapFloat(tempFromSensorBT, 0, 250, 2.45, 0);

  if (tempFromSensorET >= 220.0) tempFromSensorET += 2.0;
  if (tempFromSensorBT >= 220.0) tempFromSensorBT += 3.0;

  if (tempFromSensorET >= 0.0) tempFromSensorET += 5.4;
  if (tempFromSensorBT >= 0.0) tempFromSensorBT += 5.3;

  filteredSuhuET = (0.2 * tempFromSensorET) + (0.8 *
filteredSuhuET);
  filteredSuhuBT = (0.2 * tempFromSensorBT) + (0.8 *
filteredSuhuBT);

  // === Terima string PID dari HMI (berjalan terus-menerus) ===
  // Ini memastikan input HMI segera diproses
  while (Serial1.available()) {
    char inChar = (char)Serial1.read();
    inputString += inChar;
    if (inChar == '\n') {
      stringComplete = true;
    }
  }

  // Jika ada input PID baru dari HMI, parse dan update nilai KP,
  KI, KD
  if (stringComplete) {
    parsePID(inputString);

    if (kp < 0) kp = 0;
  }
}

```

```

    if (ki < 0) ki = 0;
    if (kd < 0) kd = 0;

    inputString = "";
    stringComplete = false;
}

// === Kirim ke HMI Nextion (Setiap 1 Detik) ===
if (millis() - lastHMIUpdateTime >= HMI_UPDATE_INTERVAL) {
    lastHMIUpdateTime = millis(); // Perbarui waktu terakhir update
HMI

    String commandET = "txt1.txt=\" + String(filteredSuhuET, 1) +
"\\";
    Serial1.print(commandET); Serial1.write(0xFF);
Serial1.write(0xFF); Serial1.write(0xFF);

    String commandBT = "txt2.txt=\" + String(filteredSuhuBT, 1) +
"\\";
    Serial1.print(commandBT); Serial1.write(0xFF);
Serial1.write(0xFF); Serial1.write(0xFF);
}

// === Kirim ke ESP32 dan Debug ke Serial Monitor (Setiap 5 Detik)
===
if (millis() - lastESP32SendTime >= ESP32_SEND_INTERVAL) {
    lastESP32SendTime = millis(); // Perbarui waktu terakhir kirim
ke ESP32

    String outData = "ET:" + String(filteredSuhuET, 1) +
";BT:" + String(filteredSuhuBT, 1) +
";KP:" + String(kp, 1) +
";KI:" + String(ki, 1) +
";KD:" + String(kd, 1) + ";\n";
    Serial2.print(outData);
    Serial.flush(); // Pastikan buffer kosong

    Serial.println("Data ke ESP32: " + outData);
}

// Penting: Hapus delay() di sini karena kita menggunakan timer
berbasis millis()
// delay(1000); // HAPUS BARIS INI
}

// === Fungsi bacaSuhu ===
float bacaSuhu(int pin) {

```

```

long totalAdcValue = 0;
for (int i = 0; i < NUM_TEMP_SAMPLES; i++) {
    totalAdcValue += analogRead(pin);
}
float averagedAdcValue = (float)totalAdcValue / NUM_TEMP_SAMPLES;
float voltage = (averagedAdcValue * 5.0) / 1023.0;
return constrain(mapFloat(voltage, minVoltage, maxVoltage,
minTemp, maxTemp), minTemp, maxTemp);
}

// === Fungsi mapFloat ===
float mapFloat(float x, float in_min, float in_max, float out_min,
float out_max) {
    return (x - in_min) * (out_max - out_min) / (in_max - in_min) +
out_min;
}

void parsePID(String data) {
    int kpStart = data.indexOf("KP:") + 3;
    int kiStart = data.indexOf("KI:") + 3;
    int kdStart = data.indexOf("KD:") + 3;

    int kpEnd = data.indexOf(';', kpStart);
    int kiEnd = data.indexOf(';', kiStart);
    int kdEnd = data.indexOf(';', kdStart);

    if (kpStart >= 3 && kpEnd > kpStart)
        kp = data.substring(kpStart, kpEnd).toFloat();
    if (kiStart >= 3 && kiEnd > kiStart)
        ki = data.substring(kiStart, kiEnd).toFloat();
    if (kdStart >= 3 && kdEnd > kdStart)
        kd = data.substring(kdStart, kdEnd).toFloat();
}

```

Lampiran 2. Program ESP32

```

#include <Arduino.h>
#include <WiFi.h>
#include <WiFiClientSecure.h>
#include <FirebaseClient.h>

// === Koneksi WiFi dan Firebase ===
#define WIFI_SSID "WillyWonka 🍫"
#define WIFI_PASSWORD "coklatenak2"
#define Web_API_KEY "AIzaSyAHMmxBQyNAjBxBx2QmMIu9vk3DvSVuXwJ4"
#define DATABASE_URL "https://HMI-roaster-default-rtdb.asia-southeast1.firebaseio.com/"
#define USER_EMAIL "zaky.abdurrozaq@mhs.polman-bandung.ac.id"

```

```

#define USER_PASS "085erfan085"

// Firebase Setup
UserAuth user_auth(Web_API_KEY, USER_EMAIL, USER_PASS);
FirebaseApp app;
WiFiClientSecure ssl_client;
using AsyncClient = AsyncClientClass;
AsyncClient aClient(ssl_client);
RealtimeDatabase Database;

void processData(AsyncResult &aResult) {
    if (!aResult.isResult()) return;
    if (aResult.isError())
        Serial.printf("[ERROR] %s: %s\n", aResult.uid().c_str(),
aResult.error().message().c_str());
    if (aResult.available())
        Serial.printf("[OK] %s: %s\n", aResult.uid().c_str(),
aResult.c_str());
}

// Variabel data dari Arduino
float et = 0.0, bt = 0.0;
float kp = 0.0, ki = 0.0, kd = 0.0;
String incoming = "";

void setup() {
    Serial.begin(115200);
    Serial2.begin(9600, SERIAL_8N1, 16, 17); // RX=16, TX=17

    // WiFi
    WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
    Serial.print("Connecting to Wi-Fi");
    while (WiFi.status() != WL_CONNECTED) {
        Serial.print(".");
        delay(300);
    }
    Serial.println(" Connected!");

    // Firebase init
    ssl_client.setInsecure();
    ssl_client.setConnectionTimeout(1000);
    ssl_client.setHandshakeTimeout(5);

    initializeApp(aClient, app, getAuth(user_auth), processData,
"authTask");
    app.getApp<RealtimeDatabase>(Database);
    Database.url(DATABASE_URL);
}

```

```

}

void loop() {
  app.loop(); // Firebase internal handler

  if (Serial2.available()) {
    incoming = Serial2.readStringUntil('\n');
    // 🖱️ Output ini akan muncul setiap kali data diterima dari
    Arduino (setiap 5 detik)
    Serial.println("Data from Arduino: " + incoming);

    et = parseFloat(incoming, "ET:");
    bt = parseFloat(incoming, "BT:");
    kp = parseFloat(incoming, "KP:");
    ki = parseFloat(incoming, "KI:");
    kd = parseFloat(incoming, "KD:");

    // Kirim ke Firebase
    if (app.ready()) {
      Database.set<float>(aClient, "/roaster/et", et, processData,
"sendET");
      Database.set<float>(aClient, "/roaster/bt", bt, processData,
"sendBT");
      Database.set<float>(aClient, "/roaster/kp", kp, processData,
"sendKp");
      Database.set<float>(aClient, "/roaster/ki", ki, processData,
"sendKi");
      Database.set<float>(aClient, "/roaster/kd", kd, processData,
"sendKd");
    }
  }
}

// === Parser Utility ===
float parseFloat(String data, String key) {
  int index = data.indexOf(key);
  if (index >= 0) {
    int end = data.indexOf(';', index);
    return data.substring(index + key.length(), end).toFloat();
  }
  return 0.0;
}
}

```

Lampiran 3. Program Python pada *Visual Studio Code*

```

import firebase_admin
from firebase_admin import credentials, db
import csv
from datetime import datetime
import os

# === Inisialisasi Firebase ===
cred = credentials.Certificate(
    r"C:\Users\MyBook Hype AMD\OneDrive\Desktop\DRAFT TUGAS
AKHIR\[Temp]\python\HMI-roaster-firebase-adminsdk-fbsvc-
5d84bfedb1.json"
)
firebase_admin.initialize_app(cred, {
    'databaseURL': 'https://HMI-roaster-default-rtdb.asia-
southeast1.firebaseio.com/'
})

# === Generate nama file berdasarkan waktu saat logging dimulai ===
timestamp_info = datetime.now().strftime('%Y-%m-%d_%H-%M-%S')
log_filename = f"roaster_log_{timestamp_info}.csv"

# --- FIX START ---
# Use os.path.dirname(__file__) to get the directory of the current
script.
# This ensures the log file is created in the same folder as
firebase_main.py,
# where you typically have write permissions.
script_dir = os.path.dirname(__file__)
log_path = os.path.join(script_dir, log_filename)
# --- FIX END ---

# === Lokasi Firebase node ===
ref = db.reference('/roaster')

# === Penyimpanan nilai sebelumnya untuk dibandingkan ===
last_data = {}

# === Buat file CSV baru dengan header ===
try:
    with open(log_path, mode='w', newline='') as file:
        writer = csv.writer(file)
        writer.writerow(['Timestamp', 'ET', 'BT', 'Kp', 'Ki',
'Kd']) # Header kolom
    print(f"📁 File baru dibuat: {log_filename} di:
{os.path.abspath(script_dir)}")

```

```

except PermissionError:
    print(f"✘ ERROR: Permission denied to create file at
'{log_path}'.")
    print("Please ensure the script has write permissions in its
directory or run it from a user-writable location.")
    exit() # Exit if the initial file creation fails

# === Fungsi logging ke CSV ===
def log_to_csv(data):
    timestamp = datetime.now().strftime('%Y-%m-%d %H:%M:%S')
    row = [timestamp, data.get('et'), data.get('bt'),
data.get('kp'), data.get('ki'), data.get('kd')]

    try:
        with open(log_path, mode='a', newline='') as file:
            writer = csv.writer(file)
            writer.writerow(row)
            print("☑ Logged:", row)
    except PermissionError:
        print("✘ ERROR: File sedang dibuka atau terkunci. Tutup
file dan coba lagi.")
    except Exception as e:
        print(f"✘ An unexpected error occurred during logging:
{e}")

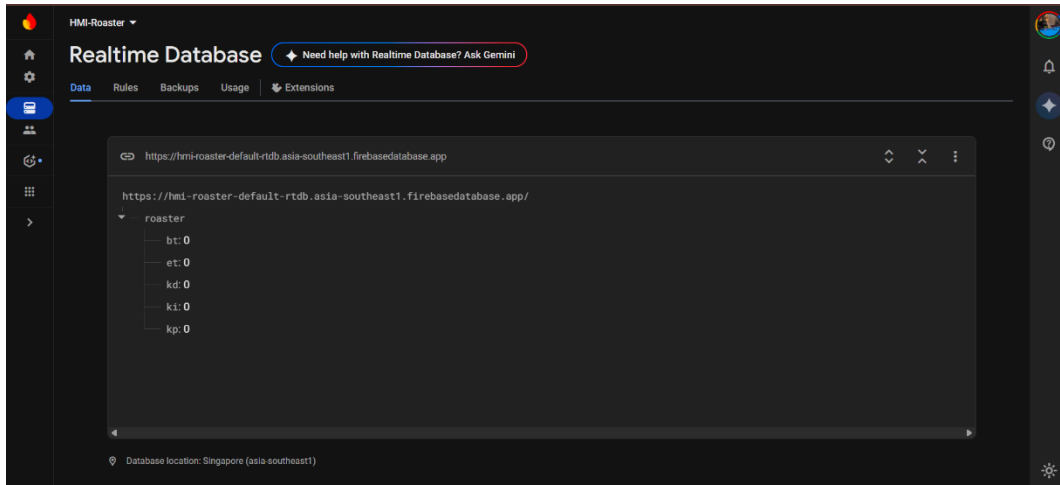
# === Listener ===
def listener(event):
    global last_data
    current_data = ref.get()

    if current_data and current_data != last_data:
        print("📄 Data changed!")
        log_to_csv(current_data)
        last_data = current_data.copy()
    else:
        print("🔇 No significant change.")

# === Start Listening ===
print("👂 Listening to Firebase... Tekan Ctrl+C untuk berhenti.")
ref.listen(listener)

```

Lampiran 4. Halaman Console Firebase



AEA-1

221341019_muhammad zaky abdurrozaq_rancang bangun sistem monitoring dan kendali proses mesin roasting

 Politeknik Manufaktur Bandung

Document Details

Submission ID

trn:oid::3618:108294943

Submission Date

Aug 14, 2025, 1:51 PM GMT+7

Download Date

Aug 14, 2025, 1:57 PM GMT+7

File Name

221341019_muhammad zaky abdurrozaq_rancang bangun sistem monitoring dan kendali proses....pdf

File Size

2.6 MB

69 Pages

13,644 Words

82,122 Characters




3% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Abstract

Top Sources

- 3%  Internet sources
- 1%  Publications
- 0%  Submitted works (Student Papers)

Integrity Flags

1 Integrity Flag for Review

-  **Hidden Text**
272 suspect characters on 4 pages
Text is altered to blend into the white background of the document.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.