pISSN: 2442-3386 eISSN: 2442-4293
Melek IT
Vol 9 No 2 Jul – Des 2023, 179 - 192
INFORMATION TECHNOLOGY JOURNAL

RANCANG BANGUN JEMURAN KERUPUK OTOMATIS BERBASIS IOT UNTUK PRODUSEN KERUPUK IKAN DI SIDAYU GRESIK

Tauhid Nur Irawan[1], Nonot Wisnu Karyanto[2]
[1,2]Program Studi Informatika, Fakultas Teknik, Universitas Wijaya Kusuma Surabaya,
tauhid.irwn@gmail.com*[1], nonotwk@uwks.ac.id[2]

Abstrak

Kabupaten Gresik merupakan salah satu kabupaten yang berada di provinsi Jawa Timur dengan letak geografis yang berdekatan langsung dengan pesisir Laut Jawa. Banyaknya budidaya ikan bandeng yang dapat dihasilkan dari tambak yang berada di pesisir utara kabupaten Gresik memiliki nilai potensial tinggi untuk dimanfaatkan sebagai produk olahan ikan. Salah satu produk hasil olahan ikan yang banyak diproduksi adalah kerupuk ikan bandeng. Ketersediaan bahan baku yang mudah diperoleh tidak menyulitkan dalam proses pengolahan kerupuk ikan. Pengeringan kerupuk dengan cara dijemur akan berjalan lancar pada cuaca sedang terik atau musim kemarau. Namun pada saat memasuki musim penghujan seringkali menemui banyak kendala yaitu panas matahari yang tidak sesuai dengan apa yang dibutuhkan seperti cuaca mendung bahkan jika tiba-tiba turun hujan yang terkadang menimbulkan jamur pada kerupuk karena tidak cukup waktu untuk mengambil kerupuk yang sedang dijemur. Proses penjemuran harus selalu diawasi langsung oleh manusia, selain itu ada beberapa jenis kerupuk dengan komposisi yang berbeda memiliki waktu penjemuran yang berbeda pula. Alat ini difungsikan untuk dapat mengenali cuaca termasuk intensitas cahaya, suhu udara, dan curah hujan sehingga dinilai lebih efisien digunakan pada saat musim penghujan. Pemakai cukup meletakkan kerupuk ikan di atas jemuran dan ketika alat mengenali kondisi hujan atau mendung maka jemuran kerupuk ikan akan ditarik oleh motor listrik untuk dimasukan ke dalam ruangan. Pada saat pemakai tidak sedang di tempat, jemuran juga dapat dimonitor dan dikontrol dari jarak jauh. Hasilnya Sistem Jemuran Kerupuk Otomatis Berbasis IoT dapat mengenali cuaca sedang cerah, panas, atau hujan. Sistem juga dapat mengulur jemuran secara otomatis ketika cuaca sedang cerah atau panas, sistem dapat menarik jemuran ketika terjadi hujan atau mendung secara otomatis. Sistem dapat mengirim notifikasi ke smartphone sesuai keadaan cuaca dan posisi jemuran sedang ditarik atau diulur secara *realtime*, sistem dapat menampilkan indikator tiap sensor ke dalam *smartphone*. *User* juga bisa memasukkan waktu penjemuran serta range suhu minimal kedalam sistem. Hasilnya sistem dapat berjalan dengan baik mengenali cuaca di sekitar jemuran kerupuk dan dapat memasukkan kerupuk ketika terjadi hujan, mendung atau saat waktu penjemuran telah habis dengan otomatis.
Kata Kunci: IoT, jemuran otomatis, *webserver*, mikrokontroller

Abstract

Gresik Regency is one of the regencies in East Java province with a geographical location directly adjacent to the coast of the Java Sea. The amount of milkfish cultivation that can be produced from ponds located on the north coast of Gresik Regency has a high potential value to be utilized as processed fish products. One of the most widely produced processed fish products is milkfish crackers. The availability of easily obtained raw materials does not make it difficult to process fish crackers. Drying crackers by drying in the sun will run smoothly in hot weather or dry season. However, when entering the rainy season, there are often many obstacles, namely the heat of the sun that does not match what is needed, such as cloudy weather and even if it suddenly rains, which sometimes causes mold on the crackers because there is not enough time to take the crackers that are being dried. The drying process must always be supervised directly by humans, besides that there are several types of crackers with different compositions that have different drying times. This tool is enabled to recognize the weather including light intensity, air temperature, and rainfall so that it is considered more efficient to use during the rainy season. The user simply puts the fish crackers on the clothesline and when the tool recognizes rainy or cloudy conditions, the fish cracker clothesline will be pulled by an electric motor to be put into the room. When the user is away, the clothesline can also be monitored and controlled remotely. As a result, the IoT-based Automatic Cracker Clothesline System can recognize whether the weather is sunny, hot, or rainy. The system can also extend the clothesline automatically when the weather is sunny or hot, the system can pull the clothesline when it rains or cloudy automatically. The system can send notifications to the smartphone according to the weather conditions and the position of the clothesline being pulled or extended in realtime, the system can display the indicators of each sensor to the smartphone. Users can also enter the drying time and minimum temperature range into the system. As a result, the system can run well to recognize the weather around the cracker *clothesline and can insert crackers when it is raining, cloudy or when the drying time has expired automatically.*

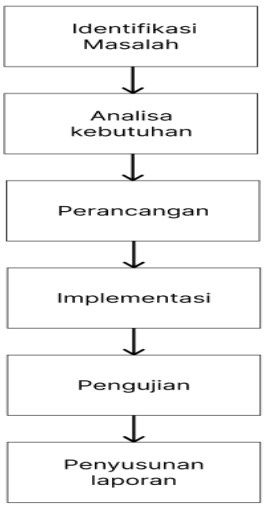
***Keywords**: IoT, automatic clothesline, webserver, microcontroller*

I. PENDAHULUAN

Di era teknologi saat ini, di mana aktivitas manusia semakin dimudahkan oleh kehadiran robot dan perangkat pintar, kehidupan sehari-hari telah mengalami transformasi. Salah satu konsep terkini yang mendukung kemajuan ini adalah *Internet of Things* (IoT) , di mana benda atau objek tertentu dilengkapi dengan sensor dan perangkat lunak untuk berkomunikasi, mengendalikan, menghubungkan, dan bertukar data melalui *internet*[1]. Kabupaten Gresik, sebagai salah satu wilayah potensial untuk pengembangan sektor perikanan, menonjol baik dalam perikanan tangkap maupun budidaya. Luas lahan di Kabupaten Gresik menjadi aset penting dalam mengembangkan potensi perikanan, terutama karena sekitar sepertiga wilayahnya adalah wilayah pesisir pantai sepanjang 140 km, meliputi Kecamatan Kebomas, Gresik, Manyar, Bungah, Sidayu, Ujungpangkah, Panceng, Sangkapura, Tambak, dan Pulau Bawean. Sementara Kabupaten Gresik dikenal sebagai salah satu kawasan industri utama di Jawa Timur, sektor perikanan di daerah ini juga memiliki kontribusi yang signifikan. Ketersediaan bahan baku yang mudah diperoleh mendukung pengolahan kerupuk ikan, tetapi proses penjemuran kerupuk masih menghadapi kendala, terutama selama musim penghujan. Kondisi cuaca yang tidak dapat diprediksi selama musim hujan seringkali menghambat proses penjemuran tradisional dengan sinar matahari. Oleh karena itu, diperlukan solusi *inovatif* berupa alat yang dapat mengatasi kendala tersebut. Alat ini dirancang untuk mengenali kondisi cuaca, termasuk intensitas cahaya, suhu udara, dan hujan, sehingga proses penjemuran kerupuk dapat berjalan lebih efisien, terutama saat musim hujan. Alat ini juga memungkinkan pemantauan dan pengendalian jarak jauh melalui *smartphone*. Permasalahan yang muncul dari latar belakang di atas melibatkan integrasi mekanisme penjemuran kerupuk dengan sistem pengenalan cuaca sekitar, otomatisasi penjemuran dan penarikan jemuran kerupuk, serta integrasi sistem dengan smartphone untuk pemantauan. Sejumlah batasan masalah ditetapkan agar penelitian dapat terfokus, dengan menggunakan mikrokontroller *Esp32*, sensor hujan, *sensor LDR*, *sensor suhu*, motor listrik, dan aplikasi *Blynk* sebagai bagian dari implementasi sistem yang diinginkan[3]. Tujuan dari penelitian ini adalah menghasilkan alat penjemuran kerupuk yang otomatis menyesuaikan dengan kondisi cuaca, terhubung dengan *smartphone*, dan memberikan kemudahan bagi pelaku industri kerupuk ikan dalam hal penjemuran kerupuk hasil produksi. Dengan demikian, penelitian ini diarahkan untuk mencapai solusi *inovatif* dan efisien dalam proses penjemuran kerupuk, menghadirkan manfaat bagi para produsen dan pemangku kepentingan terkait.

II. METODE

Agar penelitian dapat berjalan secara terstruktur dan selesai tepat pada waktunya, tahapan atau langkah-langkah yang akan dilakukan pada penelitian rancang bangun jemuran kerupuk otomatis berbasis *internet of things* ini dimodelkan peneliti dalam bentuk diagram alur penelitian yang bereferensi pada metode pengembangan sistem yaitu metode *waterfall*. Tahapan atau langkah-langkah yang dimaksud yaitu sebagai berikut :



Gambar 1. Alur penelitian

2.1 Identifikasi Masalah

Langkah pertama yang dilakukan adalah mengidentifikasi masalah, yaitu suatu kegiatan untuk mengetahui permasalahan yang menjadi alasan dibuatnya sistem. Tahap identifikasi masalah dilakukan di lokasi penelitian yaitu di kantor kelurahan Desa Sidayu, Kecamatan Sidayu, Kabupaten Gresik dengan metode pengumpulan data yang dipakai adalah pengamatan atau observasi dan wawancara dengan pihak terkait sebagai sumber data primer, serta studi literatur sebagai sumber data sekunder.

Adapun permasalahan yang didapat dari hasil identifikasi masalah yaitu :

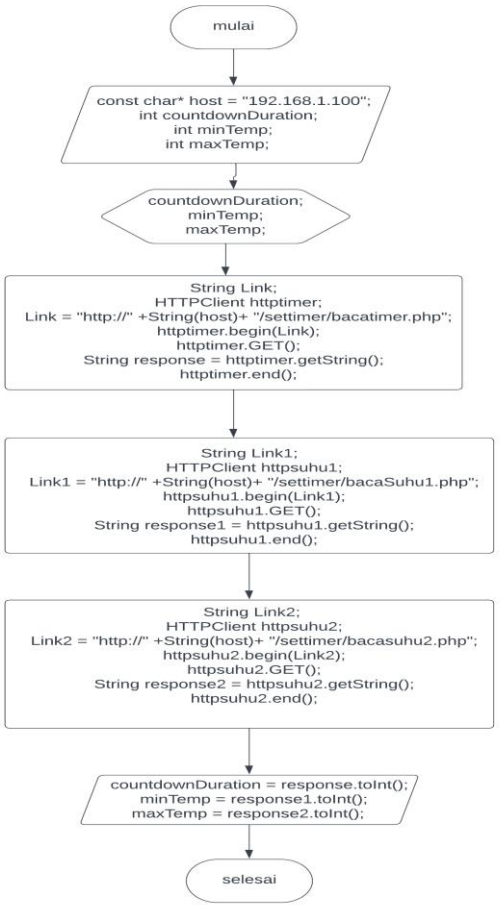
- Penjemuran dan pengangkatan jemuran kerupuk masih mengandalkan tenaga manusia.
- Penjemuran di musim penghujan tidak efektif disebabkan kurangnya tenaga manusia dengan banyaknya jemuran kerupuk yang ada tidak efisien.
- Produsen harus selalu memantau langsung jemuran kerupuk.

2.2 Analisa Kebutuhan

Setelah menyelesaikan tahap identifikasi masalah, langkah penulis selanjutnya adalah mengidentifikasi atau menganalisis kebutuhan. Analisis kebutuhan perangkat lunak dan perangkat keras adalah proses mendapatkan informasi, model dan spesifikasi sistem yang diinginkan pengguna. Analisis kebutuhan sistem

pISSN: 2442-3386 eISSN: 2442-4293
Vol 9 No 2 Jul – Des 2023, 181 - 192
Melek IT
INFORMATION TECHNOLOGY JOURNAL

dibagi menjadi dua, kebutuhan fungsional dan kebutuhan non- fungsional.



Gambar 2. Proses keluar masuk jemuran krupuk

A. Kebutuhan fungsional

Analisis kebutuhan fungsional adalah pernyataan layanan yang harus disediakan sistem untuk melakukan operasi dalam situasi tertentu dan menanggapi masukan tertentu. Persyaratan atau kebutuhan fungsional dapat diartikan sebagai persyaratan yang mencakup karakteristik atau fitur apa saja yang dimiliki sistem atau juga apa saja proses yang dilakukan sistem. Adapun proses yang ada pada sistem ini nanti yaitu :

- Sistem dapat menampilkan indikator tiap sensor.
- Sistem dapat menampilkan kondisi cuaca.
- Sistem dapat menampilkan notifikasi apabila mendeteksi hujan pada jemuran jemuran masuk kedalam rumah.
- Sistem mampu menampilkan notifikasi bahwa waktu penjemuran sudah selesai.
- User juga dapat memasukkan atau mengeluarkan jemuran secara manual lewat sistem

B. Kebutuhan non fungsional

Analisis kebutuhan non-fungsional meliputi pembatasan layanan atau fitur yang disediakan oleh sistem. Kebutuhan non fungsional juga berisi apa yang dibutuhkan pengguna untuk menjalankan sistem. Berikut kebutuhan non fungsional pada sistem jemuran otomatis berbasis *internet of things* yang akan dibangun :

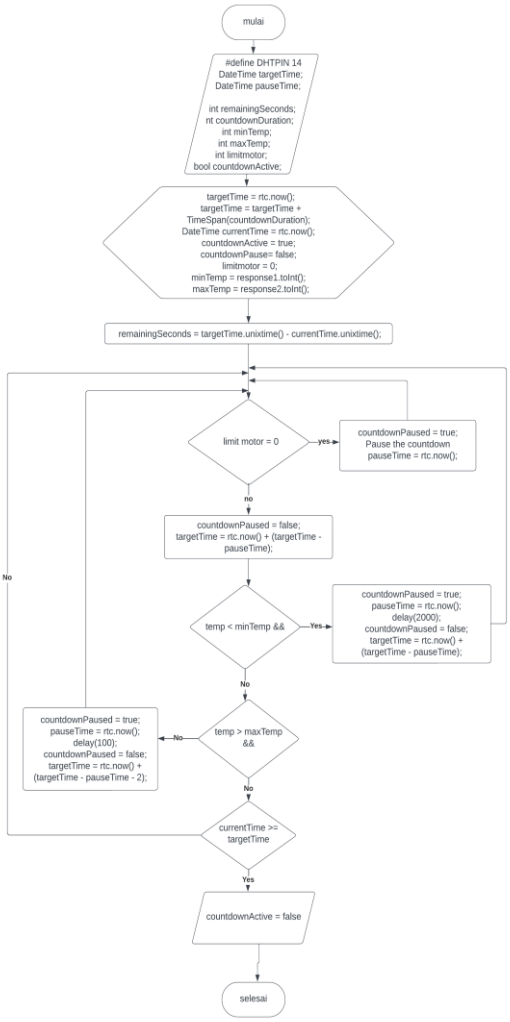
- Aplikasi dapat terinstal pada perangkat *Android*.
- Yang dapat mengendalikan jemuran hanya pengguna yang memiliki hak.
- Sistem hanya dapat diakses melalui file format apk yang telah terinstal di perangkat *Android*.
- *User interface* pada aplikasi dibuat dengan sederhana untuk memudahkan pengguna.

2.3 Perancangan

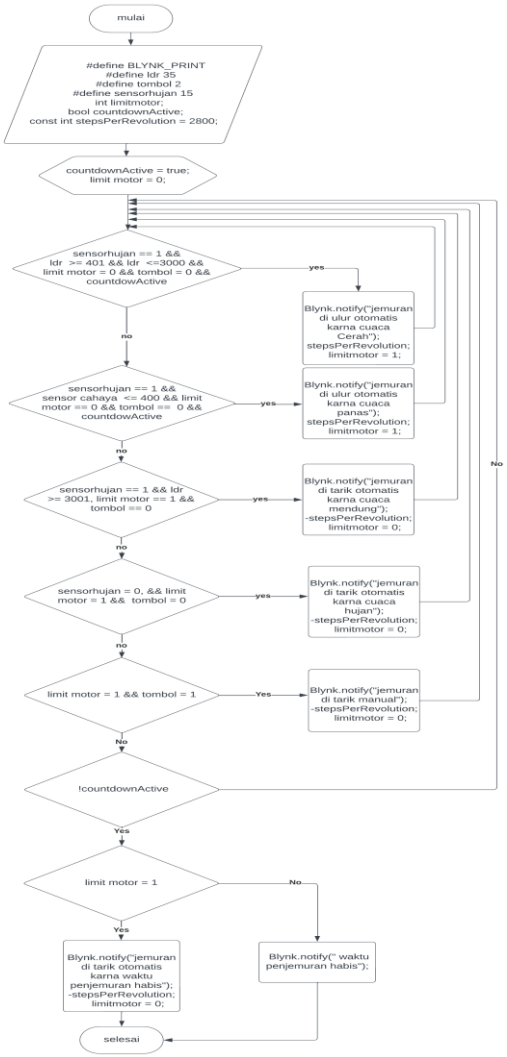
Perancangan adalah tahapan untuk menerjemahkan syarat kebutuhan ke sebuah perancangan perangkat lunak yang dapat diperkirakan sebelum melakukan implementasi (*coding*). Tahapan ini bertujuan untuk menghubungkan antara kebutuhan pengguna dengan proses implementasi (*coding*) yang akan dikembangkan agar sesuai dengan kebutuhan *user*. Desain yang akan dibuat pada sistem jemuran otomatis berbasis *internet of things* yaitu *flowchart* jalannya program, rangkaian modul sensor dan desain tampil *user interface* (UI) sistem yang akan penulis buat dengan menggunakan *software Adobe Photoshop*.

Flowchart Merupakan sebuah jenis diagram yang mewakili algoritma, alur kerja atau proses, yang menampilkan langkah-langkah dalam bentuk simbol-simbol grafis, dan urutannya dihubungkan dengan panah. Tujuan dari adanya diagram alur ini adalah untuk memudahkan membuat alur atau proses sistem keluar masuk pada jemuran kerupuk

Yang pertama adalah memasukkan alamat *server* untuk mengakses *database* yang berisi data yang akan diambil, kemudian membuat *Variabel Countdown Duration*, *minTemp*, dan *maxTemp* yang nantinya value dari variabel tersebut adalah data yang berasal dari *database*. Kemudian memberi kondisi awal yaitu dengan mengosongkan *value* dari *Variabel Countdown Duration*, *minTemp*, dan *maxTemp*. Kemudian mulai mengakses *webserver* dan mengunduh data lalu di deklarasikan sebagai *variabel response*, *response1*, dan *response2*. Kemudian *output* dari proses tersebut yang tersimpan pada *variabel response*, *response1*, dan *response2* digunakan untuk mengisi *value* dari *countdown duration*, *minTemp*, dan *maxTemp*.



Gambar 3, Proses *Timer* penjemuran



Gambar 4. Sensor waktu penjemuran.

Yang pertama dalam memasukkan variabel yang dibutuhkan untuk *timer* penjemuran, pin untuk *sensor* suhu, dan *variabel* yang mendeklarasikan jemuran berada di luar rumah atau di dalam rumah.kemudian memberi kondisi awal jemuran berada di dalam rumah, dan *timer* penjemuran aktif. Kemudian lanjut ke proses perhitungan mundur untuk timer penjemuran[7]. Lalu berlanjut untuk kondisi jemuran di dalam, jika kondisi benar maka timer penjemuran akan terjeda. *Timer* penjemuran akan berjalan kembali ketika jemuran berada di luar rumah. Kemudian berlanjut ke kondisi berikutnya yaitu suhu berada di bawah *value minTemp*, jika konsisi benar maka timer penjemuran akan terjeda selama 3 detik dan berlanjut Kembali, jika konsisi salah maka akan berlanjut pada konsisi berikutnya yaitu suhu berada di atas *value maxTemp*, jika kondisi benar maka timer penjemuran akan berlangsung lebih cepat 3 detik per perulangannya. Jika kondisi salah maka berlanjut pada kondisi terakhir dimana *timer* penjemuran sudah habis, jika benar maka *timer* penjemuran akan berubah status menjadi tidak aktif dan proses selesai, jika tidak maka perulangan akan terjadi dengan membaca jemuran sedang di dalam rumah atau tidak[4].

Memasukkan pin untuk sensor cahaya, sensor hujan, dan pin tombol manual, kemudian membuat variabel untuk mengetahui jemuran ada di dalam atau di luar, lalu variabel untuk mendeklarasikan bahwa timer penjemuran aktif, lalu membuat variabel yang berisi value untuk mengatur revolusi perputaran *motor dc*[4].

Masuk pada pengondisian awal yaitu *timer* penjemuran aktif dan kondisi jemuran ada di dalam rumah

Masuk ke pengondisian pertama yaitu cuaca tidak hujan, nilai sensor cahaya antara 401-3000, posisi jemuran ada di dalam, tombol manual tidak aktif, dan *timer* penjemuran aktif. Apa bila kondisi tersebut benar, Maka sistem akan mengirim notifikasi bahwa jemuran diulur otomatis karena cuaca cerah, kemudian *motor dc* akan berputar positif, dan jemuran akan berada di luar rumah

Apabila kondisi tadi salah, maka berlanjut ke kondisi kedua yaitu cuaca tidak hujan, nilai sensor cahaya di bawah 401, posisi jemuran ada di dalam, tombol manual tidak aktif, dan timer penjemuran aktif, jika kondisi tersebut benar maka sistem akan mengirim notifikasi bahwa jemuran diulur otomatis karena cuaca

pISSN: 2442-3386 eISSN: 2442-4293
Vol 9 No 2 Jul – Des 2023, 183 - 192
Melek IT
INFORMATION TECHNOLOGY JOURNAL

panas, kemudian *motor dc* akan berputar positif, dan jemuran akan berada di luar rumah

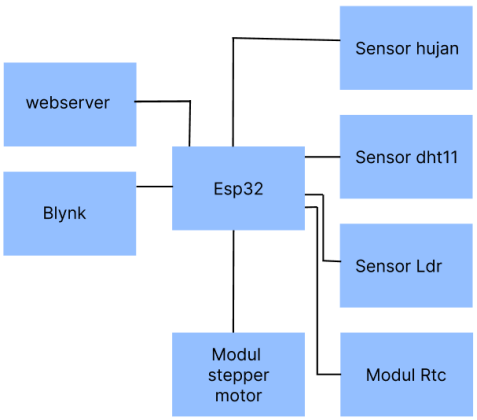
Apabila kondisi salah maka berlanjut ke kondisi ketiga yaitu, cuaca tidak hujan, nilai *sensor* cahaya lebih dari 3000, posisi jemuran ada di luar, tombol manual tidak aktif. Jika kondisi tersebut benar maka sisten akan mengirim notifikasi jemuran ditarik otomatis karena cuaca mendung, kemudian *motor dc* berbutar *negatif*, dan jemuran akan berada di dalam rumah

Apabila kondisi salah maka akan berlanjut ke kondisi ke 4 yaitu, cuaca hujan, posisi jemuran ada di luar, tombol manual tidak aktif. Jika kondisi tersebut benar maka sisten akan mengirim notifikasi jemuran di tarik otomatis karena cuaca hujan, kemudian *motor dc* berbutar *negatif*, dan jemuran akan berada di dalam rumah

Apabila kondisi salah maka akan berlanjut ke kondisi ke 5 yaitu, tombol manual aktif dan jemuran ada di luar. Jika kondisi tersebut benar maka sistem akan mengirim notifikasi jemuran ditarik manual, kemudian *motor dc* berputar *negatif*, dan jemuran akan berada di dalam rumah

Apabila kondisi salah maka akan berlanjut ke kondisi ke 6 yaitu, *timer* penjemuran tidak aktif dan jemuran di luar, maka sistem akan mengirim notifikasi jemuran ditarik otomatis karena waktu jemur habis, kemudian *motor dc* berputar *negatif* dan jemuran berada di dalam rumah, namun ketika waktu jemur habis dan jemuran berada di dalam rumah maka sistem akan mengirim notifikasi waktu jemur habis[9].

Pada tahap *modeling quick design* ini akan menggambarkan desain rancangan dari rangkaian sistem secara kasar atau sebuah perancangan blok fungsional sistem dalam bentuk blok diagram yang menggambarkan sistem kerja secara keseluruhan[2]. Adapun pemodelan desain cepat yang akan digunakan adalah sebagai berikut :

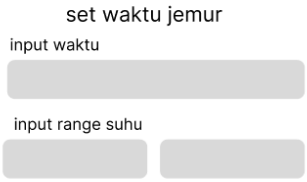


Gambar 7. Notifikasi penjemuran krupuk.

Rancangan desain UI ini akan menggambarkan desain rancangan yang akan dipakai untuk memuat tampilan ketika *user* mengakses *software Blynk* pada *Smartphone*[3] .

Gambar 5. Keterangan kondisi cuaca

Rancangan desain UI ini akan menggambarkan desain rancangan yang akan dipakai untuk memuat tampilan ketika *user* memasukkan waktu dan *range* suhu penjemuran .



Gambar 6. Seting waktu penjemuran

2.4 Implementasi

Tahapan selanjutnya setelah mendesain sistem adalah melakukan implementasi rancangan ke dalam bentuk aplikasi, tahapan ini disebut dengan istilah *coding*. *Coding* atau pengkodean adalah penerjemahan desain ke dalam bahasa yang dapat dikenali oleh komputer.

Pada sistem jemuran otomatis berbasis *internet of things* akan dibangun menggunakan *software Arduino ide* untuk pengkodean mikrokontroller *esp32* dan *software Blynk* untuk pengkodean aplikasi *smartphone* pada tahapan implementasi atau pengujian sistem berdasarkan hasil analisis dan perancangan yang dibuat melalui proses pengkodean[3].

2.5 Pengujian

Tahapan selanjutnya adalah pengujian atau proses eksekusi sistem untuk menentukan apakah sistem telah berjalan sesuai keinginan atau belum. Pada tahapan ini, pengujian akan dilakukan menggunakan metode *blackbox*, yaitu pengujian dari sisi fungsionalitas. Apabila dalam pengujian tersebut, fitur-fitur yang ada

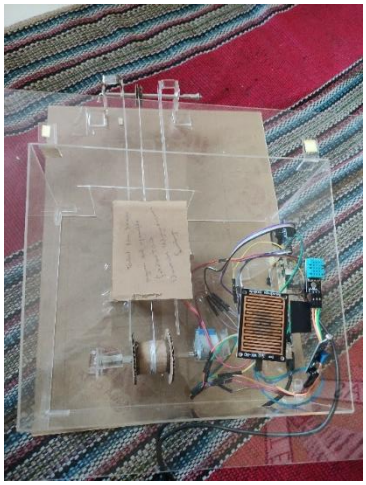
telah memenuhi kebutuhan fungsional maka sistem tersebut dianggap berhasil [2].

2.6 Penulisan Laporan

Pada tahapan ini penulis menyusun laporan hasil penelitian berdasarkan langkah-langkah penelitian yang telah dilakukan untuk menghasilkan laporan dalam bentuk teks.

III. HASIL DAN PEMBAHASAN

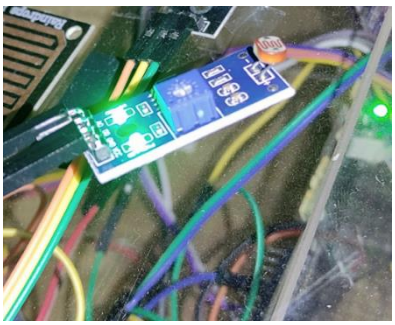
Untuk Tahap pengujian sistem ini berdasarkan rancangan yang telah dibuat dan untuk mengetahui apakah sistem yang telah dibuat dapat berfungsi dengan sesuai yang diharapkan atau tidak. Terdapat berbagai modul dan *sensor* yang telah di terapkan dalam sistem jemuran kerupuk otomatis di antaranya implementasi *sensor* hujan, *sensor ldr*, *sensor dht11*, *modul rtc*, *modul motor stepper*, *modul driver motor*. Berikut adalah uji coba sistem dari pengerjaan sistem berdasarkan rancangan desain yang telah dibuat. Adapun implementasinya jemuran kerupuk otomatis sebagai berikut.



Gambar 9. Miniatur Jemuran Kerupuk Otomatis

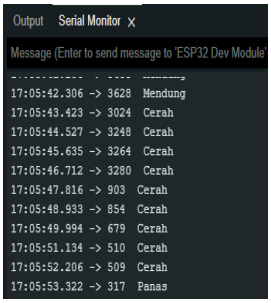
3.1. Hasil Uji Coba Sensor LDR

Pada sistem jemuran kerupuk otomatis ini, sensor LDR (*light Dependent Resistor*) digunakan untuk mengukur intensitas cahaya matahari yang adal di sekitar jemuran dan juga untuk pemicu bagi modul *driver motor* untuk mengeluarkan atau memasukan jemuran



Gambar 8. *Implementasi Sensor Ldr*

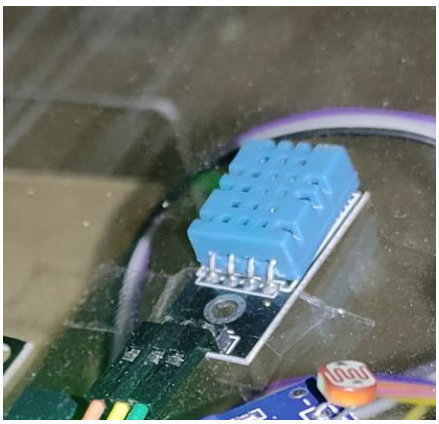
Penempatan sensor ini tepat di atas atap miniature sehingga dapat langsung menangkap sinar matahari ketika siang hari. *Implementasi sensor* ini menggunakan pin 13 sebagai *analog output* pada *esp32* dan menggunakan daya sebesar 3 *volt*. pengkondisian dibatasi ketika output analog yang di berikan oleh sensor menunjukan angka 401 sampai 3499 maka sistem akan membaca bahwa cuaca sedang cerah, kemudian bila *output sensor* menunjukkan angka lebih dari 3500 maka sistem akan membaca bahwa cuaca sedang mendung, lalu ketika sensor menunjukan angka output kurang dari 400 maka sistem akan membaca bahwa cuaca sedang panas. Dan apabila kode tersebut dijalankan pada esp32 yang telah tersambung dengan *sensor LDR* di pin yang sesuai maka *output* serial monitor pada *Arduino ide* adalah pada Gambar 10.



Gambar 10. Serial Monitor *Sensor Ldr*

3.2. Hasil Uji Coba Sensor DHT11

Pada sistem jemuran kerupuk otomatis ini, *sensor dht11* digunakan untuk mengukur suhu dan kelembapan yang adal di sekitar jemuran dan juga sebagai parameter untuk waktu penjemuran kerupuk.



Gambar 11. *Implementasi Sensor Dht11*

Penempatan sensor pada Gambar 11. berada di luar miniatur sehingga dapat langsung membaca suhu dan kelembapan sekitar. *Implementasi sensor* ini menggunakan tiga konektor yaitu yang pertama adalah konektor untuk output dari sensor yang tersampung pada *pin 14* pada *esp32* dan dua konektor sebagai supply daya untuk sensor tersebut yang terhubung pada *pin 3v3* yang berguna untuk arus positif dan pin gnd yang berfungsi untuk arus negatif

```
#include <DHT.h>
#define DHTPIN 14
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
```

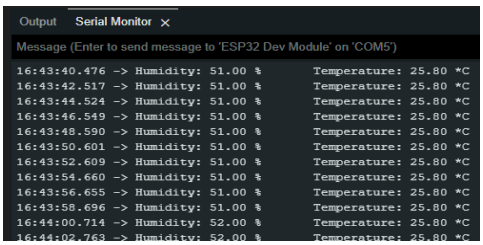
pISSN: 2442-3386 eISSN: 2442-4293
Vol 9 No 2 Jul – Des 2023, 185 - 192
Melek IT
INFORMATION TECHNOLOGY JOURNAL

```
void setup() {
  Serial.begin(115200);
  Serial.println("Deteksi Suhu dan kelembaban");
  dht.begin();
}

void loop() {
  dht11();
  delay(2000);
}
void dht11() {
  float temp = dht.readTemperature();
  float hum = dht.readHumidity();
  if (isnan(hum) || isnan(temp)) {
    Serial.println("Semsor tidak terbaca!");
    return;
  }
  Serial.print("Humidity: ");
  Serial.print(hum);
  Serial.print(" %\t");
  Serial.print("Temperature: ");
  Serial.print(temp);
  Serial.println(" *C ");
}
```

Gambar 12. Kode Dasar *Sensor Dht11*

Kode dalam Gambar 12 terdapat *library* atau perpustakaan yang di butuhkan yaitu DHT.h istilah "perpustakaan" atau "*library*" merujuk pada kumpulan kode yang telah ditulis sebelumnya untuk mempermudah pengembangan program *Arduino*. Perpustakaan berisi sekumpulan fungsi dan metode yang telah dirancang dan dikodekan untuk melakukan tugas tertentu, seperti mengendalikan perangkat keras atau melaksanakan fungsi-fungsi khusus. Kode tersebut dijalankan pada esp32 yang telah terhubung pada *sensor dht11* maka output *serial monitor* dari *Arduino ide*.

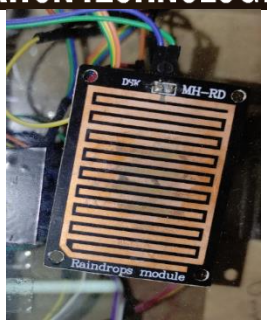
Output pada Gambar 13. menunjukan suhu serta kelembapan sekitar dan dicetak di serial monitor tiap 2 detik.



Gambar 13. *Serial Monitor Sensor Dht11*

3.3. Hasil Uji Coba Sensor Hujan

Pada sistem jemuran kerupuk otomatis ini, sensor hujan atau *rain drop sensor* digunakan untuk membaca keadaan sekitar sedang terjadi hujan atau tidak dan juga untuk pemicu bagi modul *driver motor* untuk mengeluarkan atau memasukan jemuran



Gambar 14. *Implementasi Pcb Rain Drop Sensor*

Penempatan sensor pada Gambar 14. tepat di atas atap *miniature* sehingga dapat langsung membaca ketika air hujan mengenai lempengan *sensor*. *Implementasi sensor* ini menggunakan dua konektor *positif* dan *negative* yang terhubung ke modul sensor pada Gambar 15.

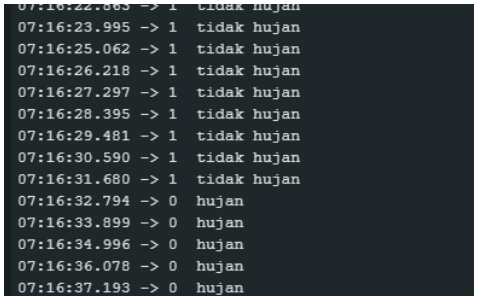
Gambar 15. *Implementasi Modul Rain Drop Sensor*

Kemudian ketiga konektor yang ada di modul pada Gambar 15. dihubungkan dengan *esp32* dengan konfigurasi *pin 15* sebagai *analog output* dan konektor *vcc* ke *pin 3v3* yaitu untuk arus positif kemudian konektor *gnd* ke *pin gnd* di *esp32* yaitu untuk arus *negative*[6].

```
#define sensorhujan 15
String cuaca;
int nilaisensorhujan;
void setup() {
  Serial.begin(115200);
}
void loop() {
  bacaHujan();
  Serial.print(nilaisensorhujan);
  Serial.print("  ");
  Serial.println(cuaca);
  delay(1000);
}
void bacaHujan(){
  nilaisensorhujan = digitalRead(sensorhujan);
  delay(100);
  if (nilaisensorhujan == 0)
  {cuaca = "hujan";}
  else if (nilaisensorhujan == 1)
  {cuaca = "tidak hujan";}
}
```

Gambar 16. Kode Dasar Sensor Hujan

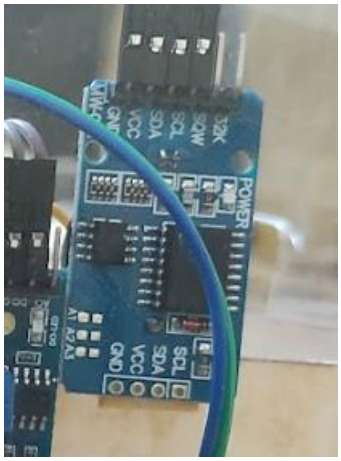
Dari kode pada gambar Gambar 16 sensor hujan memberikan *output* dengan nilai 1 atau 0, jika lempengan *sensor* atau pcb tidak terkena air maka nilai output yang diberikan adalah 1 atau bisa disebut tidak hujan, apabila lempengan *sensor* terkena air maka output yang diberikan adalah 0 yang bisa disebut mendeteksi adanya hujan yang dapat dilihat pada gambar 17



Gambar 17. Serial Monitor Rain Drop Sensor

3.4. Hasil Uji Coba Modul Rtc

Penggunaan *modul rtc* pada sistem jemuran kerupuk otomatis ini berfungsi untuk modul waktu timer penjemuran. Pin yang dipakai pada *modul rtc* kali ini adalah pin gnd sebagai pin untuk arus *power negative*, *pin vcc* untuk arus *power positif* serta pin sda digunakan sebagai jalur komunikasi untuk mengirim dan menerima data serial antara perangkat esp32 dan modul RTC.



Gambar 18. Implementasi Modul Rtc

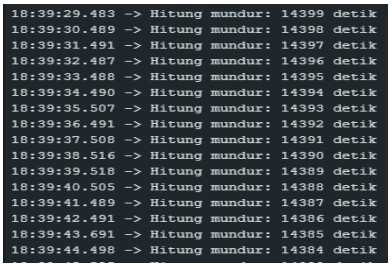
Data yang dikirim melalui pin sda pada Gambar 18. dikodekan menjadi *bit-bit biner* dan dikirim dalam bentuk seri. Pin scl digunakan sebagai jalur *clock* yang mengatur kecepatan *transfer data* antara perangkat *master* dan *slave* dalam protokol I2C. Kecepatan komunikasi ditentukan oleh *frekuensi clock* yang dikonfigurasi pada perangkat *master*. Sinyal *clock* ini digunakan untuk sinkronisasi transfer data pada *pin sda*. Dengan menggunakan protokol I2C dan *pin sda* serta *pin scl*, esp32 dapat berkomunikasi dengan modul RTC untuk membaca dan menulis waktu yang akurat pada modul RTC atau untuk mengatur parameter lainnya yang terkait dengan fungsi waktu pada sistem.

```
#include <Wire.h>
#include <RTClib.h>
RTC_DS1307 rtc;
void setup() {
  Serial.begin(9600);
  Wire.begin(21, 22);  // Inisialisasi I2C dengan pin SCL:
21 dan SDA: 22
  if (!rtc.begin()) {
    Serial.println("Modul RTC tidak terdeteksi!");
    while (1);
  }
  if (!rtc.isrunning()) {
    Serial.println("RTC tidak berjalan!");
    rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
  }
}
void loop() {
  DateTime now = rtc.now();
  // Hitung mundur dari 14400 detik
  DateTime countdown = now + TimeSpan(14400);
  while (now < countdown) {
    now = rtc.now();
    Serial.print("Hitung mundur: ");
    Serial.print(countdown - now);
    Serial.println(" detik");
        delay(1000);
  }
    Serial.println("Waktu hitung mundur selesai!");

  delay(5000);  // Tunggu 5 detik sebelum memulai hitung
mundur berikutnya

}
```

Gambar 19. Kode Dasar Modul Rtc

Dalam kode pada Gambar 19. adalah kode dasar untuk hitung mundur bisa berjalan menggunakan modul rtc, ada dua *library* yang diperlukan untuk dapat menghubungkan modul rtc dengan esp32 yaitu *library Wire* dan *library Rtclib pin sda* pada rtc terhubung pada pin 21 pada esp32, dan pin scl pada rtc terhubung pada pin 22 pada esp 32

Modul rtc diatur untuk melakukan hitung mundur selama 4 jam, sedangkan modul tersebut hanya menerima masukan dalam satuan detik, maka dari itu perintah yang di masukan adalah perhitungan selama 14400 detik atau setara dengan 4 jam seperti yang bisa dilihat pada gambar Gambar 20.



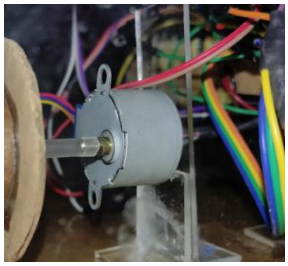
Gambar 20. *Output Serial Monitor Modul Rtc*

pISSN: 2442-3386 eISSN: 2442-4293
Vol 9 No 2 Jul – Des 2023, 187 - 192
Melek IT
INFORMATION TECHNOLOGY JOURNAL

3.5. Hasil Uji Coba Motor Dc

Hasil uji coba untuk modul *stepper motor* dilakukan untuk mengetahui mekanisme ulur dan tarik pada jemuran berjalan dengan baik atau tidak.

Motor stepper yang digunakan pada Gambar 21. di hubungkan pada modul ULN2003 sebagai *power* dan sistemasi untuk motor dc bergerak *negative* atau *positif.*



Gambar 21. *Implementasi Stepper Motor*

Gambar 22. Implementasi Modul ULN2003

Modul ULN2003 pada Gambar 22. dapat menggerakkan modul *motor dc* dengan tegangan hingga *12 volt*, namun pada implementasi jemuran kerupuk otomatis ini modul *motor dc* hanya membutuhkan daya sebesar 5 *volt* sehingga besaran *input power* yang diterima oleh modul ULN2003 hanya 5 *volt*, untuk *supply* arus listrik modul ini tidak bergantung pada *supply power* yang didapat dari *esp32*, melainkan modul harus mendapatkan *supply* listrik sendiri. Implementasi modul ini menggunakan *power* dengan *adaptor 8 volt*, atau bisa juga digantikan dengan *powerbank* dengan arus *output* yang serupa. Apabila modul menerima arus listrik yang teramat besar maka akan berimbas pada modul *motor dc* yang akan *error*, atau bahkan mengalami konsleting.

```
#include <Stepper.h>
const int stepsPerRevolution = 2048;  // change this to fit the number of steps per revolution
// ULN2003 Motor Driver Pins
#define IN1 19
#define IN2 18
#define IN3 5
#define IN4 4
Stepper myStepper(stepsPerRevolution, IN1, IN3, IN2, IN4);
void setup() {
  myStepper.setSpeed(5);  // set the speed at 5 rpm
  Serial.begin(115200);
  tarik();
  delay(3000);
  ulur();
}
void loop() {
}
void ulur() {
  myStepper.step(-stepsPerRevolution);
  delay(1000);
}
void tarik() {
  myStepper.step(stepsPerRevolution);
  delay(1000);
}
```

Gambar 23. Kode Dasar Modul ULN2003

Dalam kode pada Gambar 23. revolusi motor diatur dengan nilai 2048 dengan begitu ketika posisi motor berada pada titik revolusi 0 maka jemuran akan berada di dalam rumah dan ketika berada pada titik revolusi 2048 jemuran akan berada di luar rumah.

3.6. Hasil Uji Coba Tampilan *Blynk*

Hasil uji coba pada tampilan ini merupakan akses utama pada *software Blynk* yang terinstal di *smartphone*. Di tampilan ini terdapat beberapa informasi mulai dari situasi cuaca dan keterangan tiap sensornya yaitu sensor hujan, *sensor Ldr* dan *sensor Dht11*. Kemudian ada tombol untuk menarik jemuran secara manual.

Gambar 24. UI Blynk

Tampilan *Blynk* pada Gambar 24. dapat diakses dengan meng-*install library Blynk* untuk esp32 ke dalam *Arduino ide*.

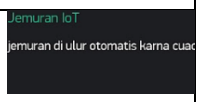
```
#define BLYNK_PRINT Serial
#include <BlynkSimpleEsp32.h>
```

Gambar 25. *Library Blynk* Untuk *Esp32*

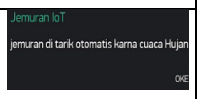
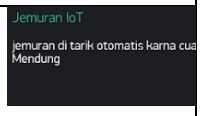
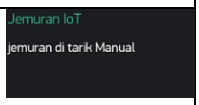
Kode pada Gambar 25. di bagian *#define BLYNK_PRINT* Serial: Baris ini mendefinisikan konstanta *BLYNK_PRINT* sebagai Serial. Konstanta ini digunakan untuk mengarahkan output debug dari *library Blynk* ke objek Serial. Kemudian kode *#include <BlynkSimpleEsp32.h>* digunakan untuk memasukkan *library BlynkSimpleEsp32.h* yang dipakai untuk menghubungkan *ESP32* dengan layanan *Blynk. Library* ini menyediakan fungsi-fungsi yang diperlukan untuk mengirim dan menerima data dari layanan *Blynk*.

3.7. Hasil Uji Coba Pengondisian

Tabel 1 Pengujian Pengondisian jemuran Terhadap cuacaa

Kondisi awal jemuran	**Kondisi sensor dan tombol penarikan manual**	**Kondisi akhir**	**Notifikasi Blynk**
Jemuran di dalam rumah	Sensor hujan tidak mendeteksi hujan sensor cahaya menangkap cahaya cerah tombol Tarik manual tidak aktif	Jemuran diulur keluar	
Jemuran di dalam rumah	Sensor hujan tidak mendeteksi hujan sensor cahaya menangkap cahaya panas tombol Tarik manual tidak aktif	Jemuran diulur keluar	
Jemuran di dalam rumah	Sensor hujan tidak mendeteksi hujan sensor cahaya menangkap cahaya mendung tombol Tarik manual tidak aktif	Jemuran tetap dalam rumah	
Jemuran di dalam rumah	Sensor hujan mendeteksi hujan sensor cahaya menangkap cahaya panas tombol Tarik manual tidak aktif	Jemuran tetap dalam rumah	
Jemuran di dalam rumah	Sensor hujan tidak mendeteksi hujan sensor cahaya menangkap cahaya cerah tombol Tarik manual aktif	Jemuran tetap dalam rumah	
Jemuran di dalam rumah	Sensor hujan mendeteksi hujan sensor cahaya menangkap	Jemuran tetap dalam rumah	

pISSN: 2442-3386 eISSN: 2442-4293
Vol 9 No 2 Jul – Des 2023, 189 - 192
Melek IT
INFORMATION TECHNOLOGY JOURNAL

	cahaya cerah tombol Tarik manual aktif		
Jemuran di dalam rumah	Sensor hujan mendeteksi hujan sensor cahaya menangkap cahaya mendung tombol Tarik manual aktif	Jemuran tetap dalam rumah	
Jemuran di luar rumah	Sensor hujan tidak mendeteksi hujan sensor cahaya menangkap cahaya cerah tombol Tarik manual tidak aktif	Jemuran tetap di luar rumah	
Jemuran di luar rumah	Sensor hujan mendeteksi hujan sensor cahaya menangkap cahaya cerah tombol Tarik manual tidak aktif	Jemuran ditarik kedalam rumah	
Jemuran di luar rumah	Sensor hujan tidak mendeteksi hujan sensor cahaya	Jemuran ditarik kedalam rumah	
	menangkap cahaya mendung tombol Tarik manual tidak aktif		
Jemuran di luar rumah	Sensor hujan tidak mendeteksi hujan sensor cahaya menangkap cahaya cerah tombol Tarik aktif	Jemuran ditarik kedalam rumah	

Pengondisian pada Tabel 1 diproses hanya ketika waktu penjemuran atau timer penjemuran masih berjalan.

3.8. Hasil Uji Coba Timer Penjemuran

Tabel 2 Timer Penjemuran

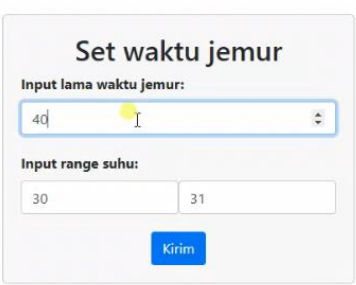
Status timer	Posisi jemuran	Hasil akhir timer
berjalan	Di luar	Timer tetap berjalan
berjalan	Di dalam	Timer di jeda
Di jeda	Di dalam	Timer tetap di jeda
Di jeda	Di luar	Timer Kembali berjalan
Timer habis	Di luar	Timer berhenti

Dari *table* pengujian *timer* penjemuran pada Tabel 2 dapat disimpulkan bahwa ketika *timer* penjemuran masih berjalan kemudian jemuran masuk kedalam rumah karena mendeteksi kondisi mendung atau hujan maka *timer* penjemuran akan dijeda, dan *timer* penjemuran akan berlanjut lagi ketika jemuran diulur keluar. Lalu ketika waktu penjemuran habis jemuran akan otomatis ditarik kedalam rumah yang menandakan penjemuran sudah selesai.

```
countdownDuration = response.toInt();
minTemp = response1.toInt();
```

```
maxTemp = response2.toInt();
```

Gambar 26. Variabel Waktu Dan suhu



Gambar 27. Halaman Input Waktu dan Suhu

Dari kode pada Gambar 26. adalah pengisian value dari *variabel countdown Duration*, *minTemp*, dan *maxTemp* dari hasil *import data* pada *database* dengan *variabel response*, *serponse1*, dan *response2*. *Countdown Duration* adalah variabel yang mewakilkan lama waktu penjemuran, minTemp adalah variabel untuk mewakilkan range suhu minimal, serta *maxtemp* adalah variabel yang mewakilkan range suhu minimal.

```
23:41:08.887 -> Connecting server....Connected server
23:41:10.209 -> timer penjemuran = 40
23:41:10.488 -> suhu minimal = 30
23:41:10.706 -> suhu maksimal = 31
```

Gambar 28. Hasil Input di Serial Monitor

Pada timer penjemuran ini juga berjalan dinamis menyesuaikan suhu sekitar, pada suhu konstan diantara *value minTemp* dan *maxTemp timer* penjemuran akan berjalan sesuai input *countdown duration* seperti dapat dilihat pada Gambar 29.

```
23:45:29.590 -> Hitung mundur:23        Temperature: 31.10 *C
23:45:30.601 -> Hitung mundur:22        Temperature: 31.10 *C
23:45:31.609 -> Hitung mundur:21        Temperature: 31.20 *C
23:45:32.660 -> Hitung mundur:20        Temperature: 31.20 *C
```

Gambar 29. timer penjemuran berjalan konstan

```
if (temp < minTemp && !countdownPaused) {
    countdownPaused = true;
    pauseTime = rtc.now();
    delay(2000);
    countdownPaused = false;
    targetTime = rtc.now() + (targetTime - pauseTime);
  }
```

Gambar 30. Kode Penyesuaian Timer Terhadap Suhu Minimum

Perintah yang dimasukan adalah ketika suhu sekitar di bawah *value minTemp* yang diperoleh dari *webserver* maka *timer* penjemuran akan berhenti selama 3 detik kemudian melanjutkan hitung mundur kembali, jika suhu terus menerus berada di bawah value minTemp, kondisi tersebut akan terus menerus terulang dan perhitungan mundur atau timer penjemuran akan berjalan normal kembali ketika suhu berada diantara *value minTemp* dan *maxTemp*.

```
23:43:40.476 -> Hitung mundur:39        Temperature: 25.80 *C
23:43:43.517 -> Hitung mundur:38        Temperature: 25.80 *C
23:43:46.524 -> Hitung mundur:37        Temperature: 25.80 *C
23:43:49.549 -> Hitung mundur:36        Temperature: 25.80 *C
```

Gambar 31. timer penjemuran berjalan lambat

Dapat dilihat pada Gambar 3.23 hitumg mundur berjalan lebih lambat tiga detik dibandingkan *realtime* waktu di sebelah kiri karena suhu sekitar kurang dari minTemp yang di masukkan.

```
if (temp > maxTemp && !countdownPaused) {
    countdownPaused = true;
    pauseTime = rtc.now();
    delay(100);
    countdownPaused = false;
    targetTime = rtc.now() + (targetTime - pauseTime - 3);
  }
```

Gambar 32. Kode Penyesuaian *Timer* Terhadap Suhu Maksimal

Perintah yang dimasukan adalah ketika suhu sekitar di atas value *maxTemp* yang diperoleh dari *webserver* maka timer penjemuran akan berhenti kemudian melanjutkan hitung mundur kembali dengan menambahkan pengurangan selama 3 detik, yang artinya perhitungan lebih cepat tiap pengulangannya, jika suhu terus menerus berada di atas *value maxTemp* kondisi tersebut akan terus menerus terulang dan perhitungan mundur atau timer penjemuran akan berjalan normal kembali ketika suhu berada di diantara *value minTemp* dan *maxTemp*

```
23:56:29.655 -> Hitung mundur:28        Temperature: 35.60 *C
23:56:30.696 -> Hitung mundur:25        Temperature: 35.60 *C
23:56:31.714 -> Hitung mundur:19        Temperature: 35.60 *C
23:56:32.763 -> Hitung mundur:16        Temperature: 35.60 *C
```

Gambar 33. timer penjemuran berjalan cepat

Dapat dilihat pada Gambar 33. hitumg mundur berjalan lebih cepat tiga detik dibandingkan *realtime* waktu di sebelah kiri karena suhu sekitar lebih dari *maxTemp* yang di masukkan.

IV. PENUTUP

4.1. Kesimpulan

Dari hasil pengujian Rancang Bangun Sistem Jemuran Kerupuk Otomatis Berbasis IoT dapat disimpulkan bahwa (1) sistem dapat mengenali cuaca cerah, hujan, mendung, panas. (2)Sistem dapat menarik jemuran secara otomatis ketika cuaca hujan, mendung, dan saat waktu penjemuran selesai. (3)Sistem dapat mengirim notifikasi ke *smartphone* sesuai keadaan cuaca cerah, panas, mendung, atau hujan dan posisi jemuran sedang ditarik atau diulur secara *realtime*. (4)Sistem dapat menampilkan indikator tiap sensor ke dalam *smartphone*. (5)User dapat memasukkan waktu jemur serta range suhu minimal dan maksimal kedalam sistem.

pISSN: 2442-3386 eISSN: 2442-4293
Vol 9 No 2 Jul – Des 2023, 191 - 192
Melek IT
INFORMATION TECHNOLOGY JOURNAL

4.2. Saran

Dalam laporan tugas akhir yang telah dibuat oleh penulis, maka penulis mencoba memberikan saran untuk pengembangan alat atau sistem ini kedepannya adalah (1) *Update* untuk penggunaan sensor yang lebih sensitif agar pengenalan cuaca lebih akurat. (2) Penggunaan mikrokontroler dengan tingkat *wifi* yang lebih tinggi untuk kinerja komunikasi dengan *smartphone* yang lebih baik. (3) Mengganti kabel *jumper* dengan *kabel fix* yang di *solder* karena rentan kendor apabila dipakai dalam jangka waktu yang lama.

DAFTAR PUSTAKA

[1] Acep Derby Yudha Anggara et al. 2019, "*Implementasi Iot Untuk Sistem Kendali Ac Otomatis Pada Ruang Kelas Di Universitas Serang Raya*" *Jurnal PROSISKO Vol. 6 No. 1*

[2] Andi Nurkholis et al. 2020, "*Sistem Pengontrol Irigasi Otomatis Menggunakan Mikrokontroler Arduino Uno*" JTST, Vol. 01, No. 01,

[3] Dahnial Syauqy et al. 2017 "*Sistem Monitoring Suhu, Kelembaban, Dan Pengendali Penyiraman Tanaman Hidroponik Menggunakan Blynk Android*" Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer

[4] Dringhuzen J. Mamahit et al. 2016 "*Rancang Bangun Rumah Pintar Otomatis Berbasis Sensor Suhu, Sensor Cahaya, Dan Sensor Hujan*" E-journal Teknik elektro dan computer

[5] Faizal Muchlis 2017 "*Perancangan Prototipe Jemuran Pakaian Otomatis Berbasis Arduino Mega 2560*"

[6] Hendra Kusumah dan Restu Adi Pradana 2019 "*Penerapan Trainer Interfacing Mikrokontroler Dan Internet of things Berbasis Esp32 Pada Mata Kuliah Interfacing*" Jurnal Cerita

[7] Ibnu Sholeh Herdidenanto et al. 2017, "*Rancang Bangun Miniatur Jemuran Pakaian Pintar Berbasis Internet of things*" Jurnal POROS TEKNIK Volume 9, No. 2,

[8] Kabul Setiya Budi dan Yudhiakto Pramudya 2017 "*Pengembangan Sistem Akuisisi Data Kelembaban Dan Suhu Dengan Menggunakan Sensor Dht11 Dan Arduino Berbasis Iot*" Prosiding Seminar Nasional Fisika

[9] M. Khoiron Ferdiansyah et al. 2020, "*Karakteristik Kerupuk Ikan Bandeng (Chanos Chanos) Dari Variasi Jenis Pengolahan Tepung Ikan Dan Pati*"

[10] Nasrun Marpaung 2017, "*Perancangan Prototype Jemuran Pintar Berbasis Arduino Uno R3 Menggunakan Sensor Ldr Dan Sensor Air*" Riau Journal Of Computer Science Vol.3 No.2

[11] Resmana Lim, Dkk 2016 "*Implementasi Internet of things Untuk Menjaga Kelembaban Udara Pada Budidaya Jamur*" https://idcloudhost.com/mari-mengenal-apa-itu-internet-thing-iot/

[12] Sri Supatmi 2011 "*Pengaruh Sensor Ldr Terhadap Pengontrolan Lampu*" Majalah Ilmiah UNIKOM

[13] Ummi Khanifah 2020, "*Rancang Bangun Alat Dengan Sistem Buka Tutup Pada Jemuran Kerupuk Putih Baraya Menggunakan Esp32 Dan Website*"

[Halaman ini sengaja dikosongkan]