

**PERANCANGAN MONITORING DAN CONTROLLING TERNAK AYAM
BERBASIS MICROCONTROLLER ARDUINO IDE**

TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat untuk Memperoleh Gelar Sarjana Strata 1 pada
Jurusan Teknik Industri Fakultas Teknologi Industri

Universitas Islam Indonesia



Disusun Oleh:

Nama : Akmal Maulana Ibrahim

NIM : 19522256

**PROGRAM STUDI TEKNIK INDUSTRI
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA**

2023

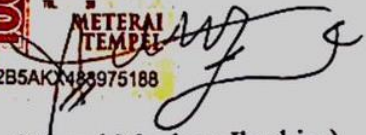
SURAT PERNYATAAN KEASLIAN TUGAS AKHIR

Saya mengakui bahwa tugas akhir yang saya teliti adalah hasil karya saya sendiri terkecuali kutipan dan ringkasan yang seluruhnya telah dijelaskan sumber-sumbernya. Jika dikemudian hari ternyata pengakuan saya terbukti tidak benar dan melanggar peraturan yang sah, maka saya sangat bersedia apabila ijazah yang saya terima ditarik kembali oleh Universitas Islam Indonesia.

Yogyakarta, 08 Mei 2023



2B5AKM483975188



(Akmal Maulana Ibrahim)

19522256

SURAT KETERANGAN PELAKSANAAN TUGAS AKHIR



FAKULTAS
TEKNOLOGI INDUSTRI

Gedung KH. Mas Mansur
Kampus Terpadu Universitas Islam Indonesia
Jl. Kaliurang km 14,5 Yogyakarta 55584
T. (0274) 898444 ext 4110, 4100
F. (0274) 895007
E. fti@uii.ac.id
W. fti.uii.ac.id

SURAT KETERANGAN PENELITIAN

Nomor : 95/Ka.lab SIMANTI/20/Lab.SIMANTI/V/2023

Assalamu'alaikum Warohmatullahi Wabarokaatuh

Dengan hormat,

Yang bertanda tangan dibawah ini, menerangkan bahwa:

Nama : Akmal Maulana Ibrahim

Nim : 19522256

Jurusan : Teknik Industri

Dosen Pembimbing : Ir. Abdullah 'Azzam S.T., M.T., IPM

Menyatakan bahwa mahasiswa tersebut diatas telah melaksanakan penelitian tugas akhir dengan judul **"PERANCANGAN PENGELOLAAN TERNAK AYAM BERBASIS MICROCONTROLLER ARDUINO IDE"** mulai pelaksanaan penelitian 25 November 2022 sampai 19 Mei 2023

Demikian surat keterangan penelitian ini kami buat. Atas perhatiannya dan kerja samanya kami mengucapkan terima kasih.

Wassalamu'alaikum Warohmatullahi Wabarokaatuh

Yogyakarta, 19 Mei 2023

Kepala Laboratorium
Sistem Manufaktur Terintegrasi

Ir. Abdullah 'Azzam., S.T., M.T., IPM

HALAMAN PENGESAHAN DOSEN PEMBIMBING

**PERANCANGAN *MONITORING* DAN *CONTROLLING* TERNAK AYAM
BERBASIS *MICROCONTROLLER* ARDUINO IDE**



(Ir. Abdullah 'Azzam, S.T., M.T., IPM.)

HALAMAN PENGESAHAN DOSEN PENGUJI

PERANCANGAN *MONITORING* DAN *CONTROLLING* TERNAK AYAM BERBASIS *MICROCONTROLLER* ARDUINO IDE

TUGAS AKHIR

Disusun Oleh:

Nama : Akmal Maulana Ibrahim

NIM : 19522256

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Strata S-1 Teknik Industri

Yogyakarta, 03 Januari 2023

Tim Penguji

Ir. Abdullah Azzam, S.T., M.T, IPM.

Ketua

Ir. Muchamad Sugarindra, ST., M.T.I., IPM

Anggota I

Atyanti Dyah Prabaswari, S.T., M.Sc

Anggota II

Mengetahui,

Ketua Program Studi Teknik Industri

Fakultas Teknologi Industri

Universitas Islam Indonesia



Ir. Muhammad Ridwan Andi Purnomo, S.T., M.Sc., Ph.D., IPM.

HALAMAN PERSEMBAHAN

Bismillahirrahmanirrahim

Puji syukur saya panjatkan kepada Allah SWT atas Rahmat-Nya yang Maha Pengasih Lagi Maha Penyayang saya diberikan kelancaran dalam menyelesaikan tugas akhir. Terima kasih sebesar-besarnya saya ucapkan kepada kedua orang tua saya, Bapak Mohammad Birendra Candra, dan Ibu Herawati yang selalu memanjatkan do'a, memberikan semangat, rasa sayang, dan motivasi untuk anaknya. Kepada Bapak Abdullah 'Azzam S.T., M.T. yang telah memberikan waktu, tenaga, dan pikiran selama membantu saya menyusun tugas akhir. Serta kepada Nabila Azzahra Yumanda dan teman-teman yang sudah banyak memberikan *support*, motivasi, dan do'a hingga akhir penelitian.

HALAMAN MOTTO

"Seseorang yang putus asa melihat kesulitan dalam setiap kesempatan, tetapi orang yang optimis melihat peluang dalam setiap kesulitan."

(Ali bin Abi Thalib)

KATA PENGANTAR

Assalamual'aikum Warahmatullahi Wabarakatuh

*Alhamdulillahil robbil 'Alamin Assholatu Wassalamu Ala Asyrofil Ambiya Wal Mursalin,
Wa 'Ala Wasohbihi Ajmain Amma Ba'du.*

Puji syukur senantiasa kita panjatkan kehadirat Allat SWT, lantaran kelimpahan kasih anugerah yang diberikan kepada kita semua agar menjadi manusia yang lebih baik akhirnya. Sholawat serta salam kami terus curahkan kepada baginda kami, Nabi besar Muhammad SAW yang sudah membawa kita dari Masa Jahiliah hingga dihormati oleh Allat SWT saat ini dan juga kepada keluarganya, sahabat, dan seluruh pengikutnya sampai yaumul akhir. Dengan demikian peneliti dapat menyelesaikan skripsi dengan judul “PERANCANGAN *MONITORING* DAN *CONTROLLING* TERNAK AYAM BERBASIS MICROCONTROLLER ARDUINO IDE” sebagai syarat untuk memperoleh gelar Strata-1 (S1) pada Fakultas Teknologi Industri, Universitas Islam Indonesia.

Dalam proses penelitian tugas akhir ini, peneliti menyadari bahwa semua tidak luput dari bimbingan, bantuan, serta dorongan dari berbagai pihak. Oleh karena itu, dalam kesempatan ini peneliti menyampaikan ucapan terimakasih yang tak terhingga kepada:

1. Bapak Prof. Dr. Ir. Hari Purnomo, M.T., IPU, ASEAN.Eng. selaku dekan Fakultas Teknologi Industri, Universitas Islam Indonesia.
2. Bapak Ir. Muhammad Ridwan Andi Purnomo, S.T., M.Sc., Ph.D., IPM. Selaku Kepala Program Studi Teknik Industri, Fakultas Teknologi Industri, Universitas Islam Indonesia.
3. Bapak Ir. Abdullah 'Azzam, S.T., M.T., IPM. selaku Dosen Pembimbing Tugas Akhir yang telah sabar dan memberikan arahan dalam menuntaskan penelitian saya.
4. Bapak Putra selaku pemilik peternakan ayam pandanaran yang telah bersedia meluangkan waktu serta peternakannya untuk pelaksanaan proses penelitian ini.
5. Bapak Heri Susilo, S.Pd.T. selaku Laboran Laboratorium Sistem Manufaktur yang senantiasa meluangkan waktunya dalam membimbing dan membantu saya mengoperasikan peralatan laboratorium.
6. Kedua orang tua saya, Ibu saya tersayang Herawati, dan Bapak saya tersayang Mohammad Birendra Candra yang memberi do'a, semangat, dan motivasi selama

pelaksanaan penelitian Tugas Akhir. Hal yang diberikan dan ajaran baik yang diajarkan ke saya sebagai modal untuk menjadi manusia lebih baik lagi.

7. Nabila Azzahra Yumanda dengan NIM 112011133197 orang terdekat saya setelah keluarga saya yang telah menemani selama ini. Kontribusinya yang banyak dalam proses penelitian ini dengan senantiasa meluangkan tenaga, pikiran, materi maupun moril kepada saya, serta kesabarannya menghadapi saya.
8. Bapak Duad selaku pengrajin kayu yang memberikan ilmu dalam pengolahan kayu.
9. Zainul Qalbi Arrifani yang telah menyumbangkan ide dan motivasi dalam mengembangkan penelitian ini.
10. Asisten laboratorium SIMAN, Mas Farhan, dan Mas Dwiki yang membimbing saya dalam penggunaan mesin *3D Print*.
11. Pihak lain yang tidak bisa disebutkan namanya satu-persatu yang telah ikut berperan dan mendoakan saya dalam penyelesaian tugas akhir ini, Terima kasih banyak, semoga Allah SWT memberikan balasan yang setimpal.

Peneliti menyadari bahwa penulisan dalam Tugas Akhir ini masih banyak kekurangan, karena berbagai keterbatasan peneliti. Oleh karena itu, kritik dan saran diharapkan untuk kemajuan peneliti. Semoga tugas akhir ini dapat berguna untuk pengetahuan serta bermanfaat bagi seluruh kalangan umum.

Wabillahi Taufik Wal Hidayah

Wassalamu'alaikum warahmatullahi wabarakatuh

Yogyakarta, 15 Mei 2023



Akmal Maulana Ibrahim

ABSTRAK

Penelitian ini merupakan penelitian dengan perancangan *microcontroller* Arduino IDE untuk meningkatkan produktivitas peternakan ayam serta dapat disesuaikan kebutuhannya oleh *user*. Metode konvensional yang masih diterapkan pada peternakan berdampak pada produktivitas pengelolaan peternakan dan ergonomi pekerja, salah satu contoh peternakan ayam tersebut berada di Jl. Pandanaran, Yogyakarta. Dilakukan *assessment* Lingkungan Kerja Fisik (LKF) untuk mengetahui produktivitas peternakan ayam. *Assessment* yang dilakukan pada 18 titik pengujian memiliki dua elemen fisik yaitu suhu, dan pencahayaan. Hasil menunjukkan kedua elemen masih belum sesuai standar peternakan menurut DITJENNAK. Ergonomi yang merupakan penyesuaian tugas oleh pekerja berfokus pada *interface* ruang kerja untuk menghindari postur yang memiliki risiko pemicu cedera *Musculoskeletal Disorder* (MSD). Perhitungan risiko MSD dihitung menggunakan *Rapid Entire Body Assessment* (REBA) dengan *software ErgoFellow* dengan data yang diperoleh berdasarkan studi pustaka, observasi lapangan, serta wawancara dengan pemilik serta pekerja di peternakan. Langkah pengelolaan peternakan ayam dengan memanfaatkan sistem automasi untuk mengurangi dampak MSD yang dirancang berbasis IoT dengan *Blynk Cloud* ini terdapat 5 tahapan pembangunan *prototyping* di dalamnya, yaitu *Requirements Gathering*, *Quick Design*, *Build Prototype*, *Evaluation and Refining*, dan *Engineer the Product*. Perhitungan REBA pertama dilakukan dengan hasil skor 9 tergolong pada kategori *High Risk*. Maka dari itu, peneliti ingin mengurangi risiko pekerja. Pembangunan serta pengujian *prototype* dilakukan untuk memastikan sistem dapat berfungsi semestinya serta dapat menyesuaikan kebutuhan *user* seperti *monitoring* elemen fisik dan *controlling feeder* yang dapat diakses melalui *Blynk Cloud*. Setelah diuji, penerapan automasi diterapkan kembali serta dilakukan perhitungan risiko cedera kedua menggunakan REBA, skor hasil menunjukkan 3 yang tergolong pada kategori rendah risiko. Manfaat dari diterapkannya sistem automasi ini dapat mengurangi repetisi postur pekerja yang memiliki risiko cedera, serta memudahkan *user* mengelola kegiatan ternak seperti pemberian pakan dan minum, pencahayaan, hingga pengaturan suhu ternak.

Kata Kunci: Peternakan ayam, *Microcontroller* Arduino IDE, *Prototyping*, LKF, *Rapid Entire Body Assessment* (REBA)

DAFTAR ISI

SURAT PERNYATAAN KEASLIAN TUGAS AKHIR.....	i
SURAT KETERANGAN PELAKSANAAN TUGAS AKHIR	ii
HALAMAN PENGESAHAN DOSEN PEMBIMBING.....	iii
HALAMAN PENGESAHAN DOSEN PENGUJI	iv
HALAMAN PERSEMBAHAN	v
HALAMAN MOTTO	vi
KATA PENGANTAR.....	vii
ABSTRAK	ix
DAFTAR ISI	x
DAFTAR TABEL	xiii
DAFTAR GAMBAR.....	xv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	7
1.3 Batasan Penelitian	7
1.4 Tujuan Penelitian	8
1.5 Manfaat Penelitian	8
1.6 Sistematika Penelitian	8
BAB II TINJAUAN PUSTAKA	10
2.1 Kajian Literatur.....	10
2.2 Landasan Teori	17
2.2.1 <i>Musculoskeletal Disorder (MSD)</i>	17
2.2.2 <i>Rapid Entire Body Assessment (REBA)</i>	18
2.2.3 Lingkungan Kerja Fisik	22
2.2.4 <i>Microcontroller</i>	24
2.2.5 Arduino IDE	24
2.2.6 Automasi	26
2.2.7 <i>Prototyping</i>	28
BAB III METODE PENELITIAN.....	30
3.1 Subjek dan Objek Penelitian.....	30
3.2 Metode Pengambilan Data.....	30
3.3 Alur Penelitian	31
3.4 Analisis Hasil.....	34

3.5 Penutup	34
BAB IV PEMBANGUNAN SISTEM.....	35
4.1 <i>Requirements Gathering</i>	35
4.1.1 Kebutuhan Pengguna	35
4.1.2 Kebutuhan Dasar Sistem.....	37
4.1.3 Pengambilan Data LKF Peternakan.....	38
4.2 <i>Quick Design</i>	40
4.2.1 Peralatan Penelitian.....	40
4.2.2 Diagram Blok Perancangan Sistem	47
4.3 <i>Build Prototype</i>	49
4.3.1 Penyetelan Awal Blynk Cloud.....	49
4.3.2 Pembangunan <i>Feeder</i> Pakan.....	52
4.3.3 Pembangunan Kontrol Temperatur.....	60
4.3.4 Pembangunan Sistem Pencahayaan	67
4.3.5 Pembangunan Sensor Penyimpanan Pakan dan Minum.....	73
4.4 <i>Evaluation and Refining</i>	79
4.4.1 Evaluasi <i>Prototype</i>	79
4.4.2 Perbaikan <i>Prototype</i>	80
4.5 <i>Engineer the Product</i>	87
4.5.1 Desain Skematik Keseluruhan	87
4.5.2 <i>Script</i> Sistem Keseluruhan.....	89
4.5.3 Penetapan Pemberian Pakan	98
4.5.4 Pembangunan Tata Letak Sistem.....	99
BAB V PENGUJIAN DAN PEMBAHASAN SISTEM.....	103
5.1 Pengujian Sistem.....	103
5.1.1 <i>User Acceptance Test</i>	103
5.1.2 <i>Performance Test</i>	105
5.2 Pembahasan Proyeksi Peletakan Alat	110
5.2.1 Peletakan Sensor DHT11	110
5.2.2 Peletakan Sensor LDR	111
5.3 Pembahasan REBA (<i>Rapid Entire Body Assessment</i>)	113
5.3.1 Sudut Postur Kerja	113
5.3.2 Perhitungan Skor REBA Sebelum Perancangan Pengelolaan Otomatis	115
5.3.3 Perhitungan Skor REBA Setelah Perancangan Pengelolaan Otomatis	116

5.3.4 Perbandingan Beban Kerja Sebelum dan Sesudah Perancangan Otomatis .	117
BAB VI PENUTUP	119
6.1 Kesimpulan	119
6.2 Saran	120
DAFTAR PUSTAKA.....	121
LAMPIRAN	A-1
Lampiran 1. Hasil Perancangan Skematik Sistem	A-1
Lampiran 2. Hasil <i>Dashboard Blynk Cloud IoT</i>	A-2
Lampiran 3. Desain Model 3D <i>Prototype</i>	A-3
Lampiran 4. Perancangan <i>Prototype</i>	A-4
Lampiran 5. Kuesioner <i>User Acceptance Test</i>	A-6
Lampiran 6. Lembar <i>Performance Test</i>	A-7
Lampiran 7. Foto Pengangkatan Beban Pakan	A-9
Lampiran 8. Pengkodean Arduino IDE	A-10

DAFTAR TABEL

Tabel 1. 1 Perbandingan Kondisi Peternakan dengan Kondisi Optimal.....	3
Tabel 2. 1 Tabel posisi penelitian	14
Tabel 2. 2 Skor Pergerakan Punggung (Batang Tubuh)	19
Tabel 2. 3 Skor Beban yang Diangkat	19
Tabel 2. 4 Skor <i>Coupling</i>	20
Tabel 2. 5 <i>Activity Score</i>	20
Tabel 2. 6 Level Risiko dan Tindakan	21
Tabel 2. 7 Perbandingan antara ESP 32 dan Arduino Uno.....	26
Tabel 4. 1 Kebutuhan Pengguna	36
Tabel 4. 2 Suhu Lingkungan Optimal.....	37
Tabel 4. 3 Kebutuhan Pakan Ayam	38
Tabel 4. 4 Perangkat Keras Penelitian	40
Tabel 4. 5 Perangkat Lunak Penelitian	46
Tabel 4. 6 Peralatan <i>Feeder</i> Pakan	52
Tabel 4. 7 Keterangan Skematik <i>Feeder</i> Pakan.....	53
Tabel 4. 8 <i>Pseudo Code</i> <i>Feeder</i> Pakan	56
Tabel 4. 9 Peralatan Kontrol Temperatur	60
Tabel 4. 10 Keterangan Skematik Kontrol Temperatur.....	61
Tabel 4. 11 <i>Pseudo Code</i> Kontrol Temperatur	64
Tabel 4. 12 Peralatan Sistem Pencahayaan.....	67
Tabel 4. 13 Keterangan Skematik Sistem Pencahayaan	68
Tabel 4. 14 <i>Pseudo Code</i> Sistem Pencahayaan	70
Tabel 4. 15 Peralatan Sensor Penyimpanan.....	73
Tabel 4. 16 Keterangan Skematik Sensor Penyimpanan	73
Tabel 4. 17 <i>Pseudo Code</i> Sensor Penyimpanan	76
Tabel 4. 18 Keterangan Skematik Keseluruhan.....	88
Tabel 4. 19 <i>Pseudo Code</i> Sistem Keseluruhan	89
Tabel 4. 20 Acuan <i>Timer</i> Pemberian Pakan	98
Tabel 4. 21 <i>Timer</i> Pemberian Pakan 1	98
Tabel 4. 22 <i>Timer</i> Pemberian Pakan 2	98

Tabel 4. 23 <i>Timer</i> Pemberian Pakan 3	99
Tabel 4. 24 Keterangan Proyeksi Pelatakan Sistem	102
Tabel 5. 1 UAT-01	103
Tabel 5. 2 UAT-02	104
Tabel 5. 3 UAT-03	104
Tabel 5. 4 PT-01	105
Tabel 5. 5 PT-02	106
Tabel 5. 6 PT-03	106
Tabel 5. 7 PT-04	107
Tabel 5. 8 PT-05	108
Tabel 5. 9 PT-06	109
Tabel 5. 10 Rangkuman Sudut Yang Terbentuk	113
Tabel 5. 11 Beban, <i>Coupling</i> , dan <i>Activity Score</i>	114

DAFTAR GAMBAR

Gambar 1. 1 Produksi Ayam Ras Pedaging di Indonesia (Juta Ton)	2
Gambar 1. 2 Ilustrasi Pengangkatan Beban Pakan Ternak 50 Kg	5
Gambar 1. 3 Hasil Perhitungan Risiko Menggunakan <i>Software</i>	5
Gambar 2. 1 <i>Range</i> Pergerakan Punggung	19
Gambar 2. 2 Penentuan Skor REBA.....	21
Gambar 2. 3 <i>PINOUT</i> ESP 32	25
Gambar 2. 4 Perbandingan proses manual dengan automasi.....	27
Gambar 2. 5 Tahapan dalam proses <i>prototyping</i>	29
Gambar 3. 1 <i>Flowchart</i> Alur Penelitian.....	31
Gambar 4. 1 Titik Pengujian LKF	38
Gambar 4. 2 LKF Suhu.....	39
Gambar 4. 3 LKF Intensitas Cahaya 1.....	39
Gambar 4. 4 LKF Intensitas Cahaya 2.....	39
Gambar 4. 5 Perangkat ESP32.....	40
Gambar 4. 6 Perangkat Laptop	40
Gambar 4. 7 Kabel USB	41
Gambar 4. 8 Kabel Jumper	41
Gambar 4. 9 Kabel Jumper Buaya	41
Gambar 4. 10 Adaptor 9V.....	42
Gambar 4. 11 Adaptor 12V.....	42
Gambar 4. 12 Modul Sensor DHT11.....	43
Gambar 4. 13 Sensor HC-SR04.....	43
Gambar 4. 14 Modul Sensor LDR.....	43
Gambar 4. 15 Modul L298N.....	44
Gambar 4. 16 DC Fan 12V	44
Gambar 4. 17 DC Motor	44
Gambar 4. 18 LED WS2812B	45
Gambar 4. 19 Arduino IDE.....	46
Gambar 4. 20 Blynk Cloud	46
Gambar 4. 21 Diagram Blok Desain Sistem.....	47
Gambar 4. 22 Laman Login Blynk Cloud	49

Gambar 4. 23 Membuat Template Blynk	50
Gambar 4. 24 Menu <i>New Device</i> Blynk	50
Gambar 4. 25 <i>Firmware Configuration</i> untuk Pengkodean	51
Gambar 4. 26 Skematik <i>Feeder</i> Pakan Proteus	52
Gambar 4. 27 Menyiapkan <i>Virtual Pin Feeder</i> Pakan.....	54
Gambar 4. 28 Menyiapkan <i>Virtual Pin</i> Penjadwalan Pakan	54
Gambar 4. 29 Menyiapkan <i>Widget Switch Feeder</i> Pakan	55
Gambar 4. 30 Menyiapkan <i>Widget Time Input</i> Pakan	55
Gambar 4. 31 Penetapan Penjadwalan Jam Pemberian Pakan	56
Gambar 4. 32 Skematik Kontrol Temperatur Proteus	61
Gambar 4. 33 Menyiapkan <i>Virtual Pin</i> Temperatur	62
Gambar 4. 34 Menyiapkan <i>Virtual Pin</i> Kelembaban	63
Gambar 4. 35 Menyiapkan <i>Virtual Pin</i> Kelembaban	63
Gambar 4. 36 Menyiapkan <i>Widget Gauge</i> Kontrol Temperatur	64
Gambar 4. 37 Skematik Sistem Pencahayaan Proteus.....	68
Gambar 4. 38 Menyiapkan <i>Virtual Pin</i> Pencahayaan.....	69
Gambar 4. 39 Menyiapkan <i>Widget Switch</i> Pencahayaan.....	69
Gambar 4. 40 Menyiapkan <i>Widget Display</i> LED	70
Gambar 4. 41 Skematik Sensor Penyimpanan Proteus.....	73
Gambar 4. 42 Menyiapkan <i>Virtual Pin</i> Pakan.....	74
Gambar 4. 43 Menyiapkan <i>Virtual Pin</i> Minum.....	75
Gambar 4. 44 Menyiapkan <i>Widget Gauge</i> Penyimpanan.....	75
Gambar 4. 45 Diagram Blok yang Diperbarui.....	79
Gambar 4. 46 Unduh Blynk IoT	80
Gambar 4. 47 <i>Login Akun Blynk Smartphone</i>	81
Gambar 4. 48 Halaman Utama Aplikasi Blynk	81
Gambar 4. 49 Penambahan <i>Widget</i>	82
Gambar 4. 50 Tata Letak <i>Widget</i> Blynk IoT	82
Gambar 4. 51 Penyetelan <i>Automations</i>	83
Gambar 4. 52 <i>Create Automation</i>	83
Gambar 4. 53 Pengaturan Kondisi Automasi Penyimpanan.....	84
Gambar 4. 54 Penyetelan Pesan Notifikasi Penyimpanan	84
Gambar 4. 55 Pengaturan Kondisi Notifikasi <i>Feeder</i> Menyala.....	85

Gambar 4. 56 Penyetelan Pesan Notifikasi <i>Feeder</i> Menyala	85
Gambar 4. 57 Pengaturan Kondisi Notifikasi <i>Feeder</i> Berhenti.....	86
Gambar 4. 58 Penyetelan Pesan Notifikasi <i>Feeder</i> Berhenti.....	86
Gambar 4. 59 Desain Skematik Keseluruhan	87
Gambar 4. 60 Model 3D Bangun Ruang <i>Prototype</i> Peternakan Ayam	98
Gambar 4. 61 Gambar Teknik <i>Prototype</i> Peternakan Ayam	98
Gambar 4. 62 Hasil Instalasi Sistem.....	99
Gambar 4. 63 Proyeksi Peletakan Sistem	100
Gambar 5. 1 Jangkauan Sensor DHT111.....	109
Gambar 5. 2 Data Hasil Penerapan LKF	109
Gambar 5. 3 Pengujian Proyeksi Sistem Pencahayaan.....	110
Gambar 5. 4 Perbandingan Sistem Pencahayaan.....	111
Gambar 5. 5 Sudut-sudut Tubuh Pekerja.....	112
Gambar 5. 6 Skor REBA Sebelum Perbaikan	114
Gambar 5. 7 Skor REBA Setelah Perbaikan.....	115
Gambar 5. 8 Ilustrasi Sebelum Dilakukan Perbaikan	116
Gambar 5. 9 Ilustrasi Setelah Dilakukan Perbaikan	117

BAB I

PENDAHULUAN

1.1 Latar Belakang

Indonesia dikenal dengan beragam sumber daya alamnya yang menjadi sumber kehidupan negara, salah satu sektor yang menunjang perekonomian nasional tersebut adalah sektor pertanian. Hal ini dikarenakan sektor pertanian memiliki cakupan ragam yang luas sehingga memberi kontribusi dalam pertumbuhan ekonomi sebagai sektor penghasil bahan pangan, penyedia bahan pangan, absorpsi tenaga kerja, dan penghasil devisa negara (Nadzirah, 2020). Pada kontribusinya, sektor pertanian memiliki dampak paling tinggi terhadap perekonomian dan penyerapan tenaga kerja, dilihat dari pendapatan rumah tangga subsektor perkebunan serta peternakan memiliki peningkatan pendapatan paling besar (Setiawan Prabowo, 2020). Menurut (Parmadi et al., 2018) dengan menjaga stabilitas perekonomian Indonesia dapat menjadi modal bagi sektor lain (sektor industri tekstil, industri makanan dan minuman), menjaga ketahanan pangan, dan mendorong pengembangan industri yang berbasis pertanian. Berdasarkan hal tersebut, kinerja sektor pertanian dapat dijadikan salah satu acuan untuk mengukur kesejahteraan masyarakat. Sektor pertanian yang memiliki peran penting bagi berbagai pihak ini memiliki lima sub sektor didalamnya yaitu hortikultura, perkebunan, peternakan, kehutanan, dan perikanan (Lestari, 2021).

Sub sektor peternakan memiliki peran tak kalah penting dalam pembangunan pertanian pada masyarakat. Produksi ternak yang meningkat pada beberapa tahun terakhir memiliki perubahan dalam strategi intervensi sehingga tujuan otonomi Pemerintah Indonesia belum tercapai (Basyar, 2021 dalam Valerio et al., 2022). Berdasarkan (KEMPERRI, 2020) data realisasi rekomendasi ekspor Direktorat Jenderal Peternakan dan Kesehatan Hewan (PKH), capaian ekspor sub sektor peternakan pada tahun 2015 hingga 2018 semester I mencapai Rp 30,15 T. Rekap data Badan Pusat Statistik (BPS), jumlah produksi ayam ras pedaging atau broiler di Indonesia naik 6,43% yaitu mencapai 3,43 juta ton pada tahun 2021 (Karnadi, 2022).



Gambar 1. 1 Produksi Ayam Ras Pedaging di Indonesia (Juta Ton)

(Sumber: dataindonesia.id/sektor-riil/detail/produksi-ayam-pedaging-capai-343-juta-ton-pada-2021)

Hal ini membuktikan laporan BPS bahwa hasil ternak yang banyak dikonsumsi masyarakat Indonesia adalah daging ayam dengan rata-rata konsumsi pada kelompok rumah tangga mencapai 6,048 Kg per kapita per tahun pada 2021 (Annur, 2022). Pernyataan tersebut memberikan titik terang bahwa permintaan produk peternakan menjadi lebih banyak, karena konsumsi masyarakat akan makanan kaya protein hewani meningkat serta mampu mendorong perekonomian regional maupun nasional (KARIMUNKAB, 2019). Berdasarkan laporan data tersebut, peternakan menjadi salah satu isu yang harus diperhatikan karena dampaknya dapat menentukan kelangsungan hidup bangsa.

Produksi pengelolaan ternak ayam menurut (BPSKM, 2022) yang terus meningkat tiap tahunnya membuktikan bahwa komoditas unggas ini memiliki prospek yang baik karena permintaannya yang tinggi. Daerah Istimewa Yogyakarta menjadi salah satu kota dengan populasi ayam sebanyak 55.667.224 ekor dan terus mengalami kenaikan (BPS, 2021). Salah satu alasan mengapa ternak ayam menjadi jenis ternak yang banyak dikembangkan karena biaya produksi yang cenderung rendah dengan perputaran keuntungan yang relatif cepat.

Sayangnya dengan biaya produksi rendah, mayoritas peternakan ayam di Indonesia masih menggunakan metode konvensional yang belum memperhatikan efisiensi pekerja beserta lingkungan fisik ternak ayam. Maraknya peternakan ayam konvensional belum

memerhatikan faktor-faktor yang berpengaruh pada produktivitas serta efisiensi pada pengelolaan peternakan ayam yang menjadi dasar pertimbangan penting. Hal ini dikarenakan faktor-faktor tersebut memiliki hubungan timbal balik antara pekerja dengan hasil produksi.

Lingkungan Kerja Fisik (LKF) diterapkan untuk menggeneralisasikan setiap reaksi pekerja atau individu terhadap lingkungan kerjanya (Sander et al., 2019). Salah satunya seperti elemen fisik pada peternakan yang berpengaruh pada pengelolaan. Menurut (Hidayat, 2021) elemen fisik pada peternakan ayam meliputi sirkulasi udara dan ventilasi, pencahayaan, suhu lingkungan, kebisingan, hingga keamanan. Dengan mengetahui kebutuhan intensitas elemen pada pengelolaan dapat membantu memaksimalkan produksi bahkan meningkatkan produktivitas pekerja (Sander et al., 2019). Berdasarkan penilaian tersebut, menerapkan Lingkungan Kerja Fisik (LKF) dapat memudahkan peneliti mengetahui efisiensi lingkungan kerja.

Salah satu peternakan ayam broiler di Jalan Pandanaran masih menggunakan metode konvensional. Peternakan ini belum memiliki alat bantu untuk proses *monitoring* dan *controlling* berbagai elemen fisik yang sifatnya krusial dalam proses pengelolaan. Setelah pengamatan dilakukan, didapat hasil *assesment* LKF yang telah dibandingkan dengan kondisi optimal menurut (DITJENNAK, 2018) sebagai berikut:

Tabel 1. 1 Perbandingan Kondisi Peternakan dengan Kondisi Optimal

Elemen Fisik	Rekapitulasi pada Peternakan	Kondisi Optimal
Suhu	~ 30°C	25°C-28°C
Intensitas cahaya tanpa lampu	~ 5 Lux	20 Lux minimal
Intensitas cahaya dengan lampu	~100 Lux	50 Lux maksimal

Dapat dilihat pada tabel diatas, hasil rekapitulasi kondisi peternakan pada elemen pertama, yaitu suhu lebih tinggi dibanding kondisi optimal. Suhu yang terlalu panas pada peternakan dapat menimbulkan stres termal pada ayam, penurunan produktivitas, bahkan kematian hewan ternak (Doe, 2022). Kurangnya *controlling* pada kondisi suhu peternakan dapat disebabkan oleh sistem sirkulasi udara yang kurang baik. Elemen kedua dan ketiga merupakan intensitas cahaya, berdasarkan data elemen ini berada di luar

kondisi optimal, hal tersebut dapat mempengaruhi kesejahteraan serta pola makan ayam (Doe, 2022). Selain perhitungan terhadap faktor penunjang kesejahteraan ayam, juga dilakukan perhitungan tingkat kebisingan pada peternakan. Kegiatan ini dilakukan untuk memastikan keamanan dari para pekerja. Berdasarkan perhitungan, didapat hasil sebesar 72 dB pada peternakan yang dimana kondisi ini masih aman karena berada di bawah nilai ambang batas menurut Kementerian Tenaga Kerja dan Transmigrasi RI No. PER.13/MEN/X/2011 yang seharusnya Nilai Ambang Batas (NAB) kebisingan sebesar 85 dB.

Assesment serta observasi yang telah dilakukan pada peternakan ayam di Jalan pandanaran ini membuktikan bahwa proses *monitoring* dan *controlling* masih belum dilakukan sesuai dengan kebutuhan peternakan ayam. Didukung juga dari hasil proses wawancara, pekerja mendapat keluhan terkait proses pemberian pakan. Kegiatan pemberian pakan pada peternakan ayam ini manual dengan pengangkatan beban yang dilakukan berulang oleh pekerja. Kegiatan ini memiliki risiko tinggi pada pekerja yaitu *Musculoskeletal Disorder* (MSD). (Saputra et al., 2020) menekankan bahwa pekerjaan yang masih didominasi oleh tenaga manusia yang salah satunya pengangkatan berulang kali dapat meningkatkan risiko MSD. *Musculoskeletal Disorder* (MSD) merupakan kondisi yang terjadi karena adanya kerusakan pada sistem muskuloskeletal seperti nyeri pada otot, tendon, tulang, sendi, dan jaringan ikat (Megawati et al., 2021). Kegiatan pengangkatan beban masih dilakukan berulang kali oleh para pekerja saat memberi pakan dengan beban yang diangkat pada karung pakan mencapai 50 Kg. Kondisi ini menjadi salah satu kekurangan yang memiliki dampak pada kurangnya akurasi saat *monitoring* kondisi ternak. Metode konvensional yang masih diterapkan ini berdampak pada produktivitas ergonomi pekerja. Menurut (Fatallah & Victor, 2017) ergonomi merupakan penyesuaian tugas dengan pekerja yang berfokus pada *interface* ruang kerja untuk membantu menghindari postur serta aktivitas yang memiliki risiko pemicu cedera *Musculoskeletal Disorder* (MSD).



Gambar 1. 2 Ilustrasi Pengangkatan Beban Pakan Ternak 50 Kg

Foto di atas merupakan ilustrasi pengangkatan beban yang dapat menyebabkan MSD apabila kegiatan tersebut terus dilakukan berulang. Dapat dibuktikan berdasarkan perhitungan menggunakan metode *Rapid Entire Body Assessment* (REBA) dengan *software ErgoFellow* sebagai berikut:

REBA

CHOOSE AN OPTION BELOW

☐ Neck, trunk and legs
 ☐ Load
 ☐ Upper arm, lower arm and wrist
 ☐ Coupling
 ☐ Activity

RESULT

SCORE: **9**

SCORE	RISK
1	Negligible risk
2 or 3	Low risk, change may be needed
4 to 7	Medium risk, further investigation, change soon
8 to 10	High risk, investigate and implement change
11 or more	Very high risk, implement change

RESULT

SAVE

DATABASE

CONTROL

INFORMATION

Gambar 1. 3 Hasil Perhitungan Risiko Menggunakan *Software*

Berdasarkan hasil perhitungan tersebut, didapatkan tingkat risiko dengan skor 9 yaitu masuk ke dalam *High Risk* sehingga perlu ada titik balik untuk menghindari risiko MSD pada pekerja peternakan ayam. Menurut (Dwiseptianto & Wahyuningsih, 2021) MSD dapat menyebabkan absensi kerja cukup tinggi yang berdampak pada ketersediaan jumlah pekerja berkurang untuk melakukan kegiatan peternakan. Salah satu upaya yang dapat dilakukan untuk mengurangi risiko MSD pada pekerja peternakan ayam adalah dengan

mengimplementasikan teknologi sistem automasi yang dapat membantu pengelolaan peternakan agar pekerja dapat memonitor serta mengelola peternakan dengan mudah.

Pesatnya pertumbuhan pada bidang robotika, teknologi visualisasi, *machine learning*, dan automasi yang terintegrasi telah membantu industri manufaktur berkembang. Hal tersebut memungkinkan automasi tidak hanya pada proses produksi dengan skala besar, tetapi juga dapat dilakukan pada lingkungan produksi dengan lingkup kecil. (Khoirul Anaam et al., 2022) Menjelaskan bahwa potensi yang didapat dalam penggunaan automasi adalah berbagai macam bisnis dapat menjalankan beragam proses dengan atau bahkan tanpa bantuan manusia. Selain itu, automasi merupakan kunci dalam melacak perkembangan industri 5.0 karena dapat meningkatkan kualitas, efisiensi, dan efektivitas saat menjalankan proses produksi. Pengelolaan peternakan ayam seperti yang telah diketahui memiliki beberapa aktivitas yang dapat mempengaruhi produktivitas ergonomi pekerja, salah satunya pada proses pemeliharaan. Hadirnya sistem automasi dapat dimanfaatkan pada peternakan ayam seperti pemberian pakan dan minum ternak yang dapat diatur sesuai jadwal. Selain itu, dengan memanfaatkan sistem automasi juga dapat dilakukan pengawasan teratur pada suhu, sirkulasi udara, pencahayaan, dan berbagai faktor penting lainnya yang dapat diatur hingga dimonitor dengan meminimalisir campur tangan manusia.

Pemanfaatan sistem automasi berbasis *microcontroller* dapat mengurangi *motion* dari pekerja, hal ini dikarenakan kegiatan yang seharusnya dikerjakan oleh manusia dapat dilakukan oleh sistem dengan automasi menggunakan sensor dan alat pendukung lainnya serta dapat memaksimalkan waktu dan tenaga agar lebih efisien (Adha et al., 2020). *Microcontroller* Arduino IDE merupakan sistem komputer yang sebagian besar elemennya terbentuk dalam *chip IC (Integrated Circuit)*. Sistem komputer ini dapat dihubungkan dengan berbagai macam jenis sensor sesuai tujuan perakitannya. Pada peternakan ayam di Indonesia yang masih banyak menggunakan metode konvensional memungkinkan untuk beralih memanfaatkan sistem automasi dengan *Microcontroller* Arduino IDE. Hal ini dikarenakan Arduino memiliki berbagai macam kelebihan seperti harga terjangkau, sederhana, pemrograman yang mudah, serta perangkat lunak (*software*) yang digunakan merupakan *Open Source* sehingga tidak perlu *chip programmer* dan sudah memiliki sarana komunikasi USB (Fani et al., 2020).

Berdasarkan uraian permasalahan di atas, maka penelitian ini akan dirancang sebuah automasi berbasis *Microcontroller* Arduino IDE pada skala *prototype* untuk menunjang

proses pengelolaan peternakan ayam menggunakan metode *prototyping*. Penerapan sistem automasi menggunakan metode tersebut diharapkan dapat memperbesar tingkat efektivitas, efisiensi, serta dapat meminimalisir risiko kerja hingga pengeluaran biaya operasional peternakan ayam. Alasan penggunaan metode *prototyping* pada penelitian ini dikarenakan penentuan kebutuhan sistem dapat disesuaikan dengan kebutuhan konsumen atau *user* pada berbagai situasi maupun kondisi tertentu. Hal ini dapat mempersingkat waktu dalam pengembangan produk, sementara itu metode pembandingan yang merupakan metode *waterfall* lebih cocok untuk sistem yang bersifat generik. Sebagaimana seluruh kebutuhan metode *waterfall* telah diidentifikasi dari awal dengan pengerjaan yang runtut, maka dari itu apabila terdapat permintaan perubahan dari *user* di tengah pekerjaan konsep perlu mengulang kembali dari tahap awal. Mekanisme ini akan memakan waktu cukup panjang. Berdasarkan hal tersebut, dengan menerapkan sistem automasi pada peternakan ayam dapat menjadi acuan transisi bagi para peternak lain dari peternakan ayam tradisional yang masih mengandalkan tenaga manusia menjadi peternakan automasi. Penerapan peternakan automasi ini dapat meminimalisir campur tangan manusia serta meningkatkan efisiensi *monitoring* hingga *controlling* yang dapat berdampak positif pada produktivitas pengelolaan peternakan ayam.

1.2 Rumusan Masalah

Berdasarkan latar belakang permasalahan tersebut, maka dapat dirumuskan yaitu: Bagaimana rancangan *microcontroller* Arduino IDE untuk meningkatkan produktivitas peternakan ayam?

1.3 Batasan Penelitian

Agar penelitian senantiasa berfokus pada objek yang akan di teliti, maka dapat ditentukan Batasan masalah sebagai berikut:

1. Penelitian dilakukan sebatas *prototype* berbasis *microcontroller*, bukan merupakan produk akhir.
2. Rancangan *Prototype* yang dibuat ditujukan untuk peternakan ayam broiler.

1.4 Tujuan Penelitian

Berikut merupakan tujuan dari penelitian ini yaitu sebagai berikut:

1. Mengidentifikasi berbagai elemen fisik yang berpengaruh pada produktivitas peternakan ayam.
2. Merancang sistem automasi untuk menunjang pengelolaan peternakan ayam dengan *microcontroller arduino*.
3. Mengidentifikasi tingkat risiko cedera pada pekerja sebelum dan setelah dibuatnya sistem automasi.

1.5 Manfaat Penelitian

Berikut merupakan manfaat dari penelitian ini yaitu sebagai berikut:

1. Bagi peneliti penelitian ini diharapkan dapat acuan yang bermanfaat dalam mengimplementasikan pengetahuan dalam bidang teknik industri terkait sistem automasi (*Advanced Automation*).
2. Bagi pelaku peternakan, penelitian ini diharapkan dapat menjadi sarana yang tepat dalam membantu pengembangan peternakan yang ada di Indonesia.
3. Bagi Universitas, penelitian ini diharapkan dijadikan landasan untuk penelitian lanjutan pada masa mendatang.

1.6 Sistematika Penelitian

Penelitian ini disusun secara sistematis menjadi beberapa bagian yang dibatasi oleh tiap bab yang memiliki fokus masing-masing, berikut merupakan uraian dari masing-masing bab tersebut:

BAB I PENDAHULUAN

Bab ini menyajikan tentang latar belakang masalah, perumusan masalah, Batasan masalah, tujuan penelitian, manfaat penelitian dan sistematika penulisan.

BAB II KAJIAN LITERATUR

Bab ini memuat tentang teori-teori dan studi literatur yang merupakan hasil penelitian terdahulu yang telah dilakukan. Teori-teori tersebut diperoleh melalui buku, jurnal dan sumber-sumber lain. Maka dari itu,

landasan teori ini akan menjadi dasar teori mengenai penelitian yang akan dilakukan.

BAB III METODE PENELITIAN

Bab ini memuat tentang objek penelitian, pengumpulan data, kerangka penelitian, tahap pengolahan, analisis data serta diagram alir untuk menjabarkan alur dari sistem penelitian yang dilakukan.

BAB IV PEMBANGUNAN SISTEM

Bab ini menyajikan dan menjelaskan langkah susunan elemen-elemen yang dibutuhkan dalam membangun *prototype* berdasarkan indikator yang dibuat beserta pembuatan pemrograman pada *microcontroller*. Selain itu, juga disajikan desain instalasi pada ruang peternakan.

BAB V PENGUJIAN SISTEM DAN PEMBAHASAN

Bab ini memuat hasil pengujian sistem berdasarkan *prototype* yang telah dibangun serta dilakukan analisis dan membahas hasil dari pengujian sistem hingga dapat diimplementasikan dengan baik.

BAB VI PENUTUP

Bab ini menyajikan kesimpulan dari hasil penelitian dan diberikan saran untuk penelitian selanjutnya.

BAB II

TINJAUAN PUSTAKA

2.1 Kajian Literatur

Pada kajian literatur perbandingan dari literatur atau karya-karya tulis ilmiah relevan dengan tujuan memperoleh pemahaman yang cukup mendalam terkait permasalahan yang diteliti untuk mengidentifikasi kesenjangan atau “gap” yang belum terpecahkan. Dalam penelitian ini disertakan sebanyak 5 jurnal internasional dan 5 jurnal nasional. Berikut merupakan hasil pengkajian yang dilakukan:

Penelitian ini dilakukan oleh Irma Yulia Basri, Dony Novaliandry, dan Intan Maisa Tania pada tahun 2022 yang bertujuan untuk merancang alat pembelajaran pendeteksi bahan metal menggunakan sensor induktif dan arduino sebagai sarana pembelajaran pada laboratorium teknik automotif. Metode yang digunakan pada penelitian ini adalah *Research and Development* (RND). Hasil dari penelitian ini adalah didapatkannya tingkat akurasi dari pembangunan alat bantu pembelajaran pendeteksi bahan metal sebesar 96,68% yang menjadikan rancangan berbasis arduino sangat bermanfaat untuk digunakan sebagai media pembelajaran. Jurnal ini dapat dijadikan referensi dalam pembangunan rancangan berbasis arduino, akan tetapi metode RND kurang dapat digunakan dalam perancangan pengelolaan ternak ayam sebab perlu ada komunikasi dua arah dengan pengguna. Selebihnya, belum ada penggunaan IOT pada perancangan alat di jurnal ini.

Penelitian ini dilakukan oleh Nurul Yaqin, Tholib Hariono, Risfi Ubaidur Rohman pada tahun 2021 yang bertujuan untuk membuat sistem pemompa air otomatis berbasis arduino berdasarkan tingkat air (*water level*) pada reservoir. Metode yang digunakan pada penelitian ini adalah *waterfall*. Hasil dari penelitian ini merupakan pembangunan sistem pemompa otomatis ini dapat menjaga ekosistem pengairan pada tanaman hidroponik dengan acuan pembacaan sensor tingkat air pada *reservoir* dengan ketentuan apabila tingkat air berada di bawah 60% maka pompa akan menyala sehingga kondisi air pada media hidroponik terjaga dan stabil. Jurnal ini dapat dijadikan referensi dalam pembangunan rancangan berbasis arduino, akan tetapi metode *waterfall* kurang tepat dalam perancangan pengelolaan ternak ayam. Hal ini disebabkan kurangnya fleksibilitas apabila terdapat *adjusting* karena proses pembangunan harus mengikuti rancangan yang

ditetapkan di awal. Selain itu, perancangan alat pada jurnal ini belum dilakukan pemanfaatan IOT.

Penelitian ini dilakukan oleh Ifamalinda, Khandra Fahmi, Siska Dwi Putri pada tahun 2022 yang bertujuan untuk merancang alat pengering tanaman asam menggunakan oven berbasis arduino. Metode yang digunakan pada penelitian ini adalah metode eksperimen. Hasil dari penelitian ini adalah terdapat perbedaan hasil antara penggunaan sistem oven menggunakan arduino dengan metode pengeringan dengan sinar matahari, hasil pengeringan menggunakan sistem oven dengan ketentuan temperatur yang diatur pada 60°C merupakan kelembaban tanaman asam tersisa (9,6%) dengan waktu pengeringan 9,3 jam dan tidak ada tumbuh jamur selama proses pengeringan. Jurnal ini dapat dijadikan referensi pembangunan rancangan berbasis arduino, tetapi metode eksperimen kurang maksimal apabila diaplikasikan pada perancangan peternakan ayam karena kurangnya gambaran sistem secara jelas sebelum dilakukan pembangunan sistem. Selain itu, pada penelitian ini belum dilakukan pemanfaatan IOT.

Penelitian ini dilakukan oleh M Jusoh, N Zuhariman, dan M Muttalib pada tahun 2020 yang bertujuan untuk merancang alat pengevaluasi tingkat kelembaban tanah dari beberapa tipe tanah berbeda. Metode yang digunakan pada penelitian ini adalah eksperimen. Hasil dari penelitian ini merupakan penggunaan alat ini dapat membantu dalam memudahkan pekerjaan untuk mengukur kelembaban tanah dalam skala besar, pemanfaatan dilakukan untuk menentukan jaringan irigasi serta memonitor dan pengendalian. Akan tetapi, performa alat ini kurang baik dalam mendeteksi material tanah berpasir dan tanah liat sehingga perlu dilakukan pengembangan lagi pada penelitian selanjutnya. Jurnal ini dapat dijadikan referensi pembangunan rancangan berbasis arduino, akan tetapi metode eksperimen kurang maksimal apabila diaplikasikan pada perancangan peternakan ayam karena kurangnya gambaran sistem secara jelas sebelum dilakukan pembangunan sistem. Selain itu, pada penelitian ini belum dilakukan pemanfaatan IOT.

Penelitian ini dilakukan oleh Martinus, Metta Saridewi Wahab, Yudi, dan Hanry Ham pada tahun 2020 yang bertujuan untuk merancang sistem troli belanja berbasis arduino untuk memudahkan dalam transmisi data belanja. Metode yang digunakan pada penelitian ini adalah *waterfall*. Hasil dari penelitian ini adalah sistem yang digunakan meningkatkan pertumbuhan pelanggan dengan pesat karena efisiensi yang didapat dengan pemasangan RFID pada troli belanja sangat menghemat waktu karena tidak ada

antrian yang panjang lagi pada proses *checkout* pada kasir, seluruh belanjaan yang dimasukkan ke troli terhubung pada aplikasi di *smartphone* dan hanya perlu melakukan pembayaran pada kasir. Jurnal ini dapat dijadikan referensi dalam pembangunan rancangan berbasis arduino, akan tetapi metode *waterfall* kurang tepat dalam perancangan pengelolaan ternak ayam karena kurangnya fleksibilitas apabila terdapat *adjusting* karena proses pembangunan harus mengikuti rancangan yang ditetapkan di awal.

Penelitian ini dilakukan oleh Irawati, Sultan Achmed, dan Titus Y. Agung pada tahun 2022 yang bertujuan untuk merancang mobil remot kontrol yang dikontrol melalui *smartphone* android berbasis arduino. Metode yang digunakan pada penelitian ini adalah metode eksperimen. Hasil dari penelitian ini adalah mobil remot dapat dimanfaatkan sebagai alat untuk menjangkau lokasi yang sulit atau cukup berbahaya untuk dijangkau oleh manusia, dengan adanya robot ini manusia dapat menjangkau suatu lokasi dengan hanya mengendalikan dari jarak jauh. Pada tahap pengujian dilakukan percobaan jarak efektif dalam pengendalian robot ini, didapatkan setelah lebih dari 22 meter robot mulai terputus dari koneksi dan tidak dapat dikendalikan. Jurnal ini dapat dijadikan referensi pembangunan rancangan berbasis arduino, akan tetapi metode eksperimen kurang maksimal apabila diaplikasikan pada perancangan peternakan ayam karena kurangnya gambaran sistem secara jelas sebelum dilakukan pembangunan sistem.

Penelitian ini dilakukan oleh Suleman dkk., pada tahun 2020 yang bertujuan untuk merancang alat pendeteksi kelembaban tanah serta penyiram otomatis berbasis arduino. Metode yang digunakan pada penelitian ini adalah metode observasi. Hasil dari penelitian ini merupakan perancangan alat otomatis memberikan solusi untuk menjaga kualitas kelembaban tanah agar tetap stabil. Dengan adanya alat tersebut, persentase kadar air dengan terjaga dari 50%-70% serta dengan mudah dimonitor. Jurnal ini dapat dijadikan referensi pembangunan rancangan berbasis arduino, akan tetapi metode observasi tidak ada komunikasi dua arah antara pengguna dan pengembang sehingga memakan waktu lebih lama dalam menentukan sistem yang ingin dibangun dengan tepat. Pada penelitian ini belum diterapkan pemanfaatan IOT.

Penelitian ini dilakukan oleh Fajar Mahardika dkk pada tahun 2021 yang bertujuan untuk merancang sistem pengaman sepeda motor berbasis arduino. Metode yang digunakan dalam penelitian ini adalah metode *waterfall*. Hasil dari penelitian ini adalah dengan adanya sistem penggabungan sistem pengamanan berbasis arduino dengan saklar

kunci pada motor diharapkan dapat meminimalisir tindak kriminalitas. Jurnal ini dapat dijadikan referensi dalam pembangunan rancangan berbasis arduino, akan tetapi metode *waterfall* kurang tepat dalam perancangan pengelolaan ternak ayam karena kurangnya fleksibilitas apabila terdapat *adjusting* karena proses pembangunan harus mengikuti rancangan yang ditetapkan di awal. Selebihnya, belum ada penggunaan IOT pada perancangan alat di jurnal ini.

Penelitian ini dilakukan oleh Sandi Youda dan Juli Sardi pada tahun 2022 yang bertujuan untuk merancang alat pengontrol kecepatan cairan infus berbasis arduino. Metode yang digunakan pada penelitian ini adalah metode *prototyping*. Hasil dari penelitian ini adalah sistem ini dapat mempermudah pekerjaan perawat di rumah sakit karena memanfaatkan sistem IOT mulai dalam memonitor sehingga perawat tidak perlu mengecek satu persatu ruangan untuk pergantian cairan infus, akan tetapi terdapat kekurangan pada mekanisme kontrol kecepatan tetesan infus karena alat mekanik kurang memadai. Jurnal ini dapat dijadikan referensi dalam pembangunan rancangan berbasis arduino, metode ini memiliki tingkat fleksibilitas yang tinggi serta kesalahan dapat dideteksi lebih cepat karena umpan balik dari pengguna dapat langsung diperbaiki dan mendapatkan efisiensi waktu yang tinggi.

Penelitian ini dilakukan oleh Khairul Imtihan, Ernawati, dan Lalu Mutawalli pada tahun 2022 yang bertujuan untuk merancang alat penyiram tanaman otomatis berbasis arduino. Metode yang digunakan pada penelitian ini adalah metode *Research and Development* (RND). Hasil dari penelitian ini adalah alat ini dirancang untuk melakukan penyiraman ke sekali banyak media tanam sekaligus, dengan bantuan sensor *waterflow* dapat dihitung jumlah debit air yang tersedia. Dengan perintah rumus “jumlah penyiraman : jumlah tanaman” maka sistem arduino dengan otomatis membagi jumlah debit air kepada 16 media tanaman semangka dengan jumlah yang sama. Jurnal ini dapat dijadikan referensi dalam pembangunan rancangan berbasis arduino, akan tetapi metode RND kurang dapat digunakan dalam perancangan pengelolaan ternak ayam karena perlu ada komunikasi dua arah dengan pengguna. Selebihnya, belum ada penggunaan IOT pada perancangan alat di jurnal ini.

Tabel 2. 1 Tabel posisi penelitian

No	Penulis, Tahun	Objek Penelitian	Metode					Penerapan IoT
			<i>Research & Development</i>	Eksperimen	Observasi	<i>Prototyping</i>	<i>Waterfall</i>	
1.	(Basri, IY; Novaliandry, Dony; Tania, 2020)	Sensor induktif untuk mendeteksi material <i>metal</i> (internasional)	✓					
2.	(Yaqin et al., 2021)	Tingkat air pada tanaman hidroponik (internasional)					✓	
3.	(Ifmalinda, Fahmi, & Putri, 2022)	Tempratur kontrol untuk pengeringan tanaman asam (Internasional)		✓				
4.	(Jusoh et al., 2020)	Kelembaban tanah dari tipe tanah yang berbeda (Internasional)		✓				
5.	(Martinus et al., 2021)	Transmisi data menggunakan RFID pada troli belanja (Internasional)					✓	✓

No	Penulis, Tahun	Objek Penelitian	Metode					Penerapan IoT
			<i>Research & Development</i>	<i>Eksperimen</i>	<i>Observasi</i>	<i>Prototyping</i>	<i>Waterfall</i>	
6.	(Irawati et al., 2022)	Pengontrol mobil remot kontrol via android		✓				✓
7.	(Suleman et al., 2020)	Mendeteksi kelembaban tanah dan penyiram otomatis			✓			
8.	(Muhammad Zaky Nur Fuadi, Indra Darmawan, 2021)	Sistem pengaman sepeda motor					✓	
9.	(Youda & Sardi, 2022)	Kecepatan cairan infus				✓		✓
10.	(Imtihan, Ernawati, & Mutawalli, 2022)	Alat penyiram tanaman otomatis	✓					
11.	Penelitian saat ini	Automasi pengelolaan ternak ayam				✓		✓

Berdasarkan kumpulan ulasan penelitian yang telah dikumpulkan, dapat disimpulkan bahwa terdapat berbagai metode yang digunakan dalam perancangan bangun mikrokontroler Arduino dengan objek yang bervariasi. Adapun penggunaan metode dari penelitian sebelumnya disesuaikan dengan kebutuhan masing-masing objek dalam menyelesaikan permasalahan yang ada.

Dalam hasil penelitian tersebut, terlihat bahwa beberapa metode yang digunakan antara lain *Research and Development* (RND), Eksperimen, Observasi, *Prototyping*, serta *Waterfall*. Temuan literatur dapat menjadi dasar dalam memilih metode penelitian perancangan *prototype* pengelolaan ternak ayam berbasis *microcontroller Arduino*, yakni metode *Prototyping*. Metode ini digunakan karena memiliki struktur dan tahapan yang runtut serta dapat dilakukan perbaikan secara langsung apabila terjadi kesalahan. Kelebihan lain dalam penelitian perancangan ini adalah digunakannya penerapan *Internet of Things* (IoT) menggunakan sistem kontrol *cloud-based* melalui *Blynk* sehingga lebih mudah dalam memonitor serta mengontrol *microcontroller* tanpa batasan jarak.

2.2 Landasan Teori

Pada landasan teori dipaparkan uraian lengkap mengenai teori-teori pendukung berdasarkan penelitian sebelumnya yang dijadikan acuan dalam pengembangan penelitian ini. Dengan adanya landasan teori yang relevan dan kuat, penelitian menjadi lebih terarah, valid, dan signifikan dalam menghasilkan kontribusi ilmiah.

2.2.1 *Musculoskeletal Disorder (MSD)*

Menurut (Hossain et al., 2018) Musculoskeletal disorder (MSD) adalah kondisi yang melibatkan gangguan atau cedera pada sistem muskuloskeletal yang meliputi otot, tulang, sendi, ligamen, tendon, saraf, serta jaringan lunak lainnya. MSD dapat terjadi karena berbagai faktor seperti ketidakseimbangan postur, gerakan yang berulang-ulang, posisi yang tidak nyaman, kelelahan fisik, dan faktor lingkungan lainnya.

Bahaya MSD dalam dunia kerja perlu diperhatikan karena menyangkut kesejahteraan pekerja yang dapat berdampak pada kelancaran serta efisiensi produksi. Berikut merupakan beberapa jenis *Musculoskeletal Disorder (MSD)* serta penyebabnya:

1. *Low Back Pain (LBP)*

LBP adalah salah satu MSD yang umum terjadi pada pekerjaan yang memerlukan pengangkatan beban yang berat, postur tubuh yang buruk, atau gerakan berulang. Penyebab lainnya adalah faktor psikologis seperti stres dan depresi (Dzukou T, 2021).

2. *Carpal Tunnel Syndrome (CTS)*

CTS adalah MSD yang disebabkan oleh tekanan pada saraf median di pergelangan tangan. Hal ini terjadi akibat posisi tangan dan pergelangan yang berulang yang sering dilakukan dalam pekerjaan. Penyebab lainnya adalah faktor genetik, kelebihan berat badan, dan cedera pada pergelangan tangan (Goo YK, 2019).

3. Nyeri Leher

Nyeri leher adalah MSD yang sering terjadi pada pekerjaan yang memerlukan postur tubuh yang buruk, gerakan berulang, dan penggunaan komputer dalam waktu lama. Penyebab lainnya adalah stres, kelelahan otot, dan cedera pada leher (Kim J, 2018).

4. *Epicondylitis Lateral (Tennis Elbow)*

Epicondylitis lateral adalah MSD yang terjadi pada sisi luar siku. Penyebabnya adalah gerakan repetitif yang melibatkan otot-otot lengan dan tangan, seperti pada olahraga tenis atau pekerjaan yang memerlukan penggunaan alat berat (Ahmad I, 2020).

5. *Osteoarthritis*

Osteoarthritis adalah MSD yang disebabkan oleh kerusakan pada tulang rawan di sendi, biasanya terjadi pada usia tua. Faktor risiko lainnya adalah obesitas, cedera pada sendi, dan kelainan tulang (Zhang Y, 2018).

Pada dunia kerja, risiko MSD tidak jarang ditemukan pada hampir seluruh aspek aktivitas yang menggunakan tenaga manusia, hal tersebut diakibatkan oleh postur kerja yang kurang baik serta beban kerja (*workload*) yang berlebih. Maka dari itu, perlu adanya evaluasi rutin terkait postur tubuh pekerja menggunakan acuan ergonomi.

2.2.2 *Rapid Entire Body Assessment (REBA)*

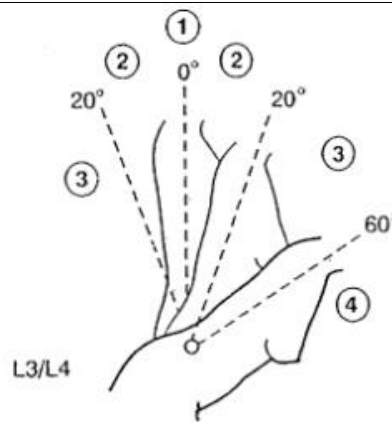
Menurut (Abdullah et al., 2020) *Rapid Entire Body Assessment (REBA)* adalah metode yang menyeluruh dan cepat untuk mengevaluasi postur kerja atau postur leher, badan, kaki, lengan, dan pergelangan tangan pekerja. Selain itu, *coupling* atau beban eksternal yang ditopang oleh tubuh serta aktivitas pekerja merupakan faktor lain yang dapat mempengaruhi skor REBA. REBA merupakan cabang studi dari *ergonomic* yang berfokus pada analisis postur tubuh manusia dalam melaksanakan aktivitas pekerjaan. Adapun *input* dalam pelaksanaan metode REBA yaitu:

1. Pengambilan data postur postur pekerja menggunakan kamera.
2. Penentuan sudut lekuk tubuh dalam melakukan pekerjaan mulai dari batang tubuh, leher, kaki, lengan atas, lengan bawah, dan pergelangan tangan.

Dalam perhitungan REBA dapat menggunakan cara manual ataupun menggunakan *software*. Dalam perhitungan manual dibagi menjadi dua segmen, yaitu grup A dan Grup B. Grup A meliputi punggung (batang tubuh), leher, dan kaki. Sementara grup B meliputi lengan atas, lengan bawah, dan pergelangan tangan. Pada penentuan skor dari tiap bagian tubuh mengikuti acuan yang sudah ditetapkan berdasarkan sudut yang terbentuk seperti contoh penentuan skor pada punggung sebagai berikut:

Tabel 2. 2 Skor Pergerakan Punggung (Batang Tubuh)

Pergerakan	Skor	Penambahan Skor
Tegak/alamiah	1	
0°-20° <i>flexion</i>	2	+ 1 jika memutar atau miring ke samping
0°-20° <i>extension</i>		
20°-60° <i>flexion</i>	3	
>20° <i>extension</i>		
>60° <i>flexion</i>	4	



Gambar 2. 1 Range Pergerakan Punggung

(Sumber: Hignett & McAtamney, 2002)

Setelah didapatkan skor dari tiap bagian tubuh, dilanjutkan dengan penentuan berat benda yang diangkat, *coupling*, dan *activity score*. Ketiga elemen tersebut menggunakan acuan skor yang telah ditetapkan sebagai berikut:

Tabel 2. 3 Skor Beban yang Diangkat

Berat Beban	Skor	Penambahan Skor
< 5 Kg	0	+1 Jika terjadi penambahan berat beban secara tiba-tiba.
5 – 10 Kg	1	
> 10 Kg	2	

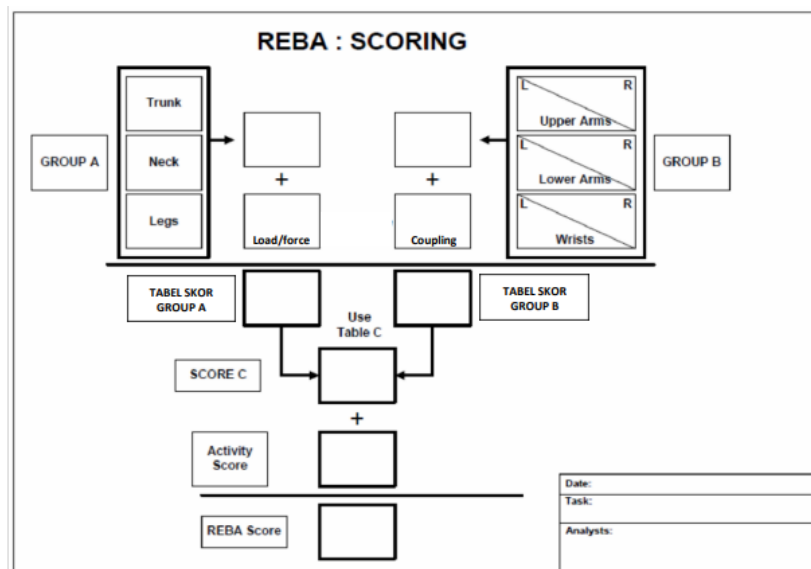
Tabel 2. 4 Skor *Coupling*

Skor	Kategori	Keterangan
0	<i>Good</i>	Jenis pegangan kuat dan tepat berada di tengah bagian sisi beban.
1	<i>Fair</i>	Pegangan tangan bisa diterima tapi belum ideal atau <i>coupling</i> lebih sesuai digunakan oleh bagian lain.
2	<i>Poor</i>	Pegangan tangan tidak bisa diterima walaupun memungkinkan.
3	<i>Unacceptable</i>	Pegangan tangan tidak ideal karena dipaksakan dan tidak aman. Tidak sesuai jika digunakan oleh bagian tubuh yang lain (tanpa <i>coupling</i>).

Tabel 2. 5 *Activity Score*

Skor	Keterangan
+1	Satu atau lebih bagian tubuh statis, ditahan lebih dari satu menit.
+1	Pengulangan gerakan dalam rentang waktu singkat, diulang lebih dari empat kali per menit (tidak termasuk berjalan).
+1	Gerakan menyebabkan perubahan atau pergeseran postur yang cepat dari postur awal.

Dilanjutkan dengan penjumlahan skor menyeluruh pada metode REBA dengan runtutan seperti gambar dibawah ini:



Gambar 2. 2 Penentuan Skor REBA

(Sumber: Hignett & McAtamney, 2000)

Level risiko dapat ditentukan berdasarkan hasil nilai REBA yang didapat dari hasil perhitungan sebelumnya. Perlu atau tidak adanya tindakan untuk melakukan perbaikan disesuaikan berdasarkan tabel risiko dan tindakan seperti pada tabel berikut:

Tabel 2. 6 Level Risiko dan Tindakan

Action Level	Skor REBA	Level Risiko	Tindakan Perbaikan
0	1	Bisa diabaikan	Tidak perlu
1	2-3	Rendah	Mungkin perlu
2	4-7	Sedang	Perlu
3	8-10	Tinggi	Perlu segera
4	11-15	Sangat Tinggi	Perlu saat ini juga

Dalam mempersingkat waktu perhitungan risiko cedera dengan metode REBA dapat menggunakan *software* yaitu *ErgoFellow*. *ErgoFellow* adalah perangkat lunak yang dapat digunakan untuk mengukur tingkat risiko cedera berdasarkan postur pengerjaan suatu aktivitas secara otomatis. *Software* ini didesain untuk membantu para ergonom dalam melakukan analisis risiko postur tubuh dengan lebih efisien dan akurat (Varmazyar, 2017). Hanya dengan memasukkan foto saat melakukan aktivitas pekerjaan dan *input mouse* dalam menentukan skor berdasarkan sudut yang dibentuk oleh tubuh pekerja, secara instan *software* ini akan menganalisis risiko cedera yang dialami oleh pekerja.

Singkatnya metode ini digunakan untuk mengetahui tindakan apa yang harus dilakukan dengan tujuan memperbaiki postur kerja yang statis, dinamis, berubah-ubah dengan cepat, serta beban kerja yang ditangani memiliki risiko menimbulkan *Musculoskeletal Disorders* (MSDs) agar dapat meningkatkan kesehatan dan keselamatan kerja sehingga tercipta produktivitas lingkungan kerja yang terjamin. (Haslindah et al., 2021).

2.2.3 Lingkungan Kerja Fisik

Menurut (Tarwaka, 2011) Lingkungan Kerja Fisik (LKF) merujuk elemen fisik pada lingkungan di lokasi kerja. Elemen tersebut merupakan segala sesuatu yang berpengaruh ketika menjalankan tugas, seperti elemen fisik yang terdapat di sekitar tempat kerja. Hal ini termasuk struktur bangunan, ruang kerja, suhu, pencahayaan, kebisingan, fasilitas fisik, serta faktor lain yang berpotensi mempengaruhi kesehatan, keselamatan, dan kenyamanan para karyawan. Terdapatnya fasilitas yang efisien, pengaturan ruang yang terorganisir, dan aksesibilitas yang baik terhadap sumber daya dan peralatan kerja dapat membantu karyawan bekerja dengan lebih efektif hingga efisien.

Dalam peternakan ayam, terdapat beberapa hal penting berkaitan dengan lingkungan kerja fisik yang perlu diperhatikan. Berikut beberapa faktor yang dapat mempengaruhi kesehatan dan kesejahteraan ayam serta keamanan dan kenyamanan pekerja di peternakan ayam (Hidayat, 2021):

1. Sirkulasi udara dan ventilasi

Sirkulasi udara yang baik dan sistem ventilasi yang memadai sangat penting di peternakan ayam. Sistem ventilasi yang efektif dapat membantu mengatur suhu, kelembaban, dan kualitas udara di dalam kandang. Udara segar yang cukup dan sirkulasi

yang baik dapat menjaga kualitas udara yang bermanfaat pada kesehatan ayam bahkan pekerja serta membantu peternakan ayam tercegah dari penyebaran penyakit.

2. Pencahayaan

Pencahayaan yang tepat sangat penting dalam peternakan ayam. Cahaya yang cukup dan teratur mempengaruhi siklus tidur dan bangun ayam serta produksi telur. Pencahayaan yang baik juga dapat membantu mengurangi stres pada ayam dan memungkinkan pekerja untuk bekerja dengan efisien.

3. Kebersihan dan sanitasi

Kebersihan dan sanitasi yang baik sangat penting pada peternakan ayam. Kandang harus rutin dibersihkan serta disterilkan untuk mengurangi risiko penyebaran penyakit. Area pekerjaan juga harus dijaga kebersihannya untuk memastikan keamanan dan kesehatan pekerja.

4. Kualitas air

Ketersediaan air yang bersih dan segar sangat penting bagi ayam. Pastikan pasokan air yang memadai dan berkualitas baik untuk minum dan sanitasi.

5. Suhu dan suhu lingkungan

Suhu yang tepat harus dipertahankan di kandang ayam. Ayam membutuhkan suhu yang nyaman agar tetap sehat dan produktif. Sistem pemanasan dan pendinginan harus disesuaikan untuk menjaga suhu yang tepat sesuai dengan kebutuhan ayam.

6. Jadwal pemberian pakan

Jumlah pemberian pakan harus teratur dengan waktu pemberian pakan pada pagi dan sore hari. Pemberian pakan harus diperhatikan, karena jika tidak akan mengakibatkan pembengkakan pengeluaran untuk pakan yang dikonsumsi oleh ternak serta perkembangan hewan ternak yang kurang maksimal.

7. Keamanan

Keamanan adalah faktor penting dalam lingkungan kerja di peternakan ayam. Keamanan yang memadai dapat mencegah kecelakaan, seperti pengaman mesin dan peralatan, penanganan bahan kimia yang aman, serta tanda peringatan dan pelatihan keselamatan bagi pekerja.

2.2.4 Microcontroller

Menurut (Ali Mazidi, 2017) *microcontroller* adalah sebuah komponen elektronik yang terdiri dari CPU (*Central Processing Unit*), RAM (*Random Access Memory*), ROM (*Read Only Memory*), serta berbagai macam modul *input-output* yang terintegrasi dalam satu *chip*. *Microcontroller* sering digunakan dalam berbagai aplikasi elektronik, dari peralatan rumah tangga, robotika hingga berbagai macam industri.

Microcontroller merupakan sistem komputer yang sebagian elemennya terbentuk dalam chip IC (*integrated circuit*) *open-source* yang didesain untuk memudahkan pengembangan berbagai macam proyek elektronik yang menggunakan mikrokontroler (Bobsis, 2019). Mikrokontroler terprogram dengan *firmware* khusus sehingga dapat memproses input dari berbagai macam sensor dan melakukan *output* ke berbagai macam aktuator. *Microcontroller* memungkinkan pengguna untuk mengembangkan proyek elektronik dengan mudah tanpa harus menguasai bahasa pemrograman yang kompleks atau memiliki pengetahuan teknis yang mendalam tentang elektronika.

Pada *microcontroller* program yang dimasukkan sebagai *input* merupakan instruksi-instruksi yang dikemas dalam kode yang dibuat dengan Bahasa pemrograman tertentu sesuai dengan *library* yang digunakan. Program ini dibuat menggunakan komputer dan kemudian dimasukkan ke dalam mikrokontroler dengan menggunakan bantuan perangkat keras (*hardware*) sesuai dengan mikrokontroler yang digunakan.

Penggunaan *microcontroller* untuk kegiatan peternakan dapat mengurangi *motion* dari pekerja, karena kegiatan yang seharusnya dikerjakan oleh manusia dapat dilakukan dengan automasi menggunakan sensor dan alat pendukung lainnya serta dapat memaksimalkan waktu dan tenaga agar lebih efisien (Adha & Asyhadie, 2020).

2.2.5 Arduino IDE

Arduino IDE (*Integrated Development Environment*) adalah perangkat lunak/*software* yang digunakan untuk memprogram mikrokontroler dari papan Arduino. Arduino IDE menyediakan lingkungan yang mudah digunakan bagi pemula untuk mempelajari dan memprogram mikrokontroler, namun juga dapat digunakan oleh pengguna yang lebih berpengalaman untuk membuat proyek yang lebih kompleks (Arduino, 2020). Arduino IDE menggunakan bahasa pemrograman yang disebut *Arduino Sketch*, yang berbasis pada bahasa pemrograman C++. Pengguna dapat mengakses fungsi-fungsi perangkat keras melalui pustaka Arduino yang telah

(Sumber: <https://www.mischianti.org/2021/02/17/doit-esp32-dev-kit-v1-high-resolution-pinout-and-specs/>)

Dalam pengembangan proyek IoT, ESP 32 sangat berada didepan dibandingkan dengan Arduino Uno yang sering dijumpai karena dengan adanya pertukaran informasi melalui *cloud*, dapat dengan mudah memonitor serta mengontrol data sesuai dengan kebutuhan. Berikut merupakan perbandingan antara Arduino UNO dengan ESP 32:

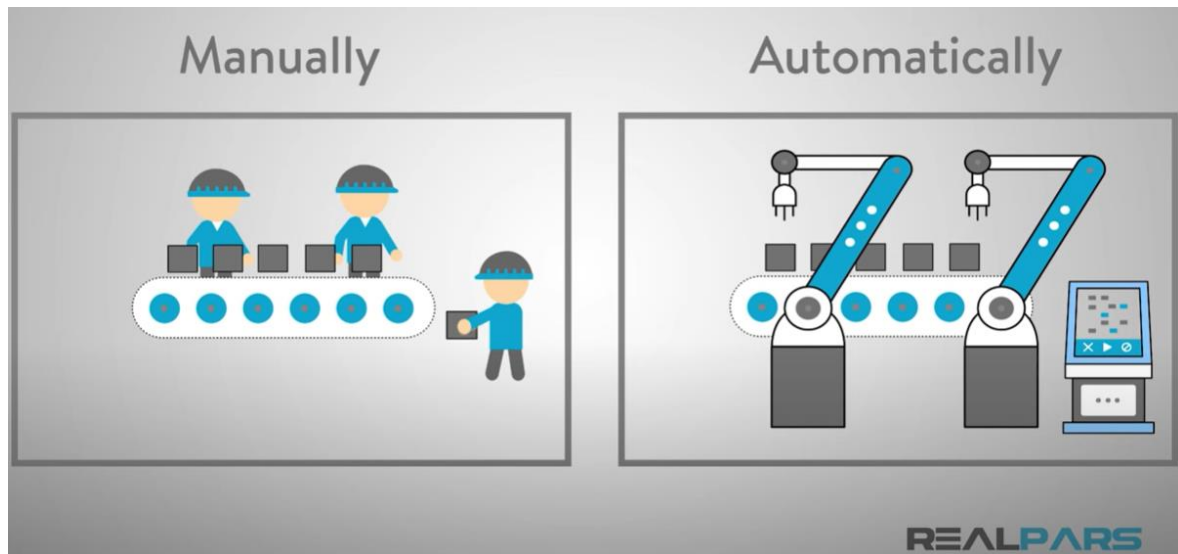
Tabel 2. 7 Perbandingan antara ESP 32 dan Arduino Uno

Karakteristik	ESP 32	Arduino Uno
Pin digital I/O	36	14
Arus DC tiap pin	40mA	20mA
Pin Analog	Hingga 18	6, 10-bit ADC
Prosesor	Xtensa <i>Dual Core</i> 32-bit	ATmega328P
Flash Memory	4 MB	32 KB
Clock Speed	80 MHz hingga 240 MHz	16 MHz
Kebutuhan Tegangan	3.3 V	5 V
WiFi	802.11 b/g/n	Tidak ada
Bluetooth	v4.2 BR/EDR dan BLE	Tidak ada

2.2.6 Automasi

Menurut artikel di Forbes (Johnson, 2022) automasi merujuk pada penggunaan teknologi untuk mengotomatiskan tugas-tugas atau proses tertentu agar meminimalisir intervensi manusia secara langsung. Bahkan automasi dapat membantu mulai dari produksi industri hingga bisnis dan layanan publik. Tujuan utama automasi merupakan peningkatan efisiensi, memperbaiki kualitas produk atau layanan.

Menggunakan automasi untuk melakukan tugas berulang bisa menjadi penghemat waktu yang sangat besar. Kemampuan untuk “*set and forget*” pada aspek-aspek tertentu dari bisnis memberi kebebasan yang lebih besar untuk mengejar bidang-bidang yang membutuhkan perhatian lebih besar, seperti menginisiasi proyek inovasi kedepannya. Selain itu, penggunaan automasi terbukti sangat berguna dalam meminimalisir error yang disebabkan oleh manusia. Dengan begitu, automasi dapat membawa lebih banyak keuntungan dalam proses bisnis dikarenakan tidak diperlukannya begitu banyak tenaga kerja manusia yang perlu digaji, dan minimalnya tingkat error sangat berdampak pada konsumsi waktu dan biaya operasional.



Gambar 2. 4 Perbandingan proses manual dengan automasi

(Sumber: <https://youtu.be/xowQkxFXTNg/>)

Pada ilustrasi di atas, dapat dilihat bahwa proses manual membutuhkan banyak tenaga manusia dengan *jobdesc* dan skill dalam bidang mereka masing-masing. Pada proses produksi tenaga kerja manusia perlu dilatih terlebih dahulu dan membutuhkan waktu untuk sampai pada titik siap memproduksi, terlebih lagi apabila ada *human error* atau hambatan yang berakibat tenaga kerja manusia tidak dapat melaksanakan pekerjaannya, maka akan berimbas pada proses produksi. Lain halnya dengan proses otomatis, hanya dengan melakukan *set up* atau pengaturan sesuai dengan kebutuhan produksi, maka mesin-mesin tersebut akan terus melakukan tugasnya secara berulang hingga target produksi tercapai.

Menurut (Rakhmanto, 2021) proses automasi memiliki keuntungan dan manfaat saat diterapkan pada bisnis, antara lain:

1. *Cost reduction*

Manfaat yang jelas dapat dirasakan dalam penerapan sistem automasi adalah pengurangan biaya, dikarenakan kecilnya tingkat kebutuhan tenaga kerja dan tingkat kesalahan produksi yang disebabkan oleh manusia.

2. *Speed increase*

Automasi dapat menghemat waktu produksi yang dibutuhkan pada suatu proses yang kompleks. Selain itu, kecepatan pemrosesan sistem automasi juga jauh lebih efisien dan lebih cepat.

3. *Quality improvement*

Salah satu manfaat terbaik dari sistem automasi adalah dapat menciptakan dan merakit produk dengan konsistensi serta efisien. Beberapa kesalahan yang sering disebabkan oleh kesalahan manusia dapat dihindari.

4. *Free up time to perform higher value task*

Automasi dapat memberikan kebebasan pada karyawan dari tugas-tugas yang berulang, monoton, melelahkan, dan kurang inspiratif. Hal ini memungkinkan karyawan untuk fokus pada pekerjaan yang lebih strategis dan bernilai tambah.

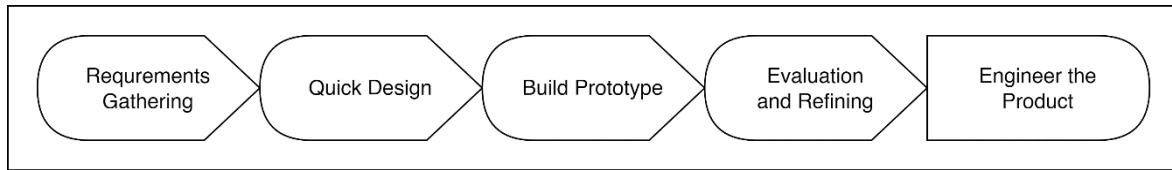
2.2.7 Prototyping

Menurut (Singh, 2018) *prototyping* merupakan suatu langkah awal dalam pembuatan model produk atau sistem yang bertujuan untuk memvalidasi ide, desain, fitur, dan fungsionalitasnya sebelum produk akhir dibuat. Proses ini dilakukan dengan cara membuat model atau percobaan yang dapat dilakukan secara fisik maupun digital dengan cepat dan bertahap. Hal ini dilakukan untuk memastikan bahwa produk yang dihasilkan sesuai dengan yang diinginkan dan dapat berfungsi dengan baik sesuai dengan kebutuhan *customer* atau pihak kedua.

Prototyping memungkinkan pengembang untuk mengidentifikasi dan memperbaiki kesalahan dan kekurangan sejak dini sebelum produk diimplementasikan secara penuh (Huynh, 2020). Menurut (Sun, 2018) dalam penggunaan metode *prototyping* terdapat beberapa keunggulan yang menjadikan metode ini masih layak digunakan dalam dunia pengembangan produk, antara lain:

1. Mempercepat proses pengembangan produk.
2. Kebutuhan pengguna dapat segera didapatkan dan diperbaiki dengan cepat.
3. Mengurangi risiko kesalahan dan kegagalan produk.
4. Menyediakan umpan balik awal dari pengguna dan pihak terkait.
5. Menghemat biaya dan waktu dalam jangka panjang.

Menurut (Alazzam, 2021) metode *prototyping* memiliki proses tahapan yang berurutan sehingga dapat lebih mudah dalam mengidentifikasi kecacatan dalam pengembangan serta pengujian performa yang lebih baik. Berikut merupakan urutan dalam proses *prototyping*:



Gambar 2. 5 Tahapan dalam proses *prototyping*

1. *Requirements Gathering*

Metode *prototype* dimulai dari penentuan kebutuhan, dilakukan analisis kebutuhan sistem yang didefinisikan secara detail. Tahap ini melibatkan diskusi antara klien dan tim *developer* atau pengembang untuk memahami keinginan dan kebutuhan pengguna sistem yang akan dikembangkan.

2. *Quick Design*

Tahap kedua dalam proses pembuatan model *prototype* adalah membuat desain awal yang sederhana serta memberikan gambaran ringkas tentang sistem yang akan dikembangkan, berdasarkan diskusi pada tahap awal analisis kebutuhan.

3. *Build Prototype*

Setelah desain awal (*quick design*) disetujui, langkah selanjutnya adalah membangun prototipe sebenarnya yang akan digunakan sebagai acuan oleh tim *programmer* dalam pembuatan program atau produk.

4. *Evaluation and Refining*

Pada tahap ini, sistem yang telah dibuat dalam bentuk prototipe akan dipresentasikan kepada pengguna untuk dievaluasi. Kemudian, pengguna akan memberikan komentar dan saran mengenai hasil prototipe yang telah dibuat. Jika pengguna tidak memiliki catatan revisi pada prototipe, maka tim dapat melanjutkan ke tahap selanjutnya. Namun, jika pengguna memberikan catatan perbaikan sistem, maka tahap ini akan terus diulang hingga klien setuju dengan sistem yang akan dikembangkan.

5. *Engineer the Product*

Pada tahap akhir ini, produk sebenarnya akan segera dibuat oleh pihak *programmer* dengan menggunakan prototipe akhir sebagai acuan. Kemudian, sistem tersebut akan diuji dan diserahkan kepada klien. Selanjutnya, akan dilakukan fase pemeliharaan untuk memastikan sistem dapat berjalan lancar tanpa kendala sesuai dengan kebutuhan pengguna.

BAB III

METODE PENELITIAN

3.1 Subjek dan Objek Penelitian

Pelaksanaan penelitian ini berlangsung mulai dari bulan November 2022 hingga bulan Mei 2023 dengan keterangan subjek dan objek sebagai berikut:

1. Subjek Penelitian

Subjek yang digunakan pada penelitian ini adalah proses pengelolaan peternakan ayam *broiler*.

2. Objek Penelitian

Objek yang digunakan pada penelitian ini adalah perancangan automasi pengelolaan peternakan ayam.

3.2 Metode Pengambilan Data

Dalam melaksanakan penelitian ini, terdapat beberapa jenis data yang dibutuhkan dalam membantu kelancaran penelitian ini. Berikut merupakan data-data yang dibutuhkan:

1. Data Primer

Data Primer adalah jenis data yang didapatkan secara langsung dari lokasi penelitian melalui metode wawancara, dan observasi. Peneliti berkonsultasi langsung dengan pemilik peternakan ayam pandanaran untuk mendiskusikan perancangan sistem yang akan dirancang. Selain itu, untuk data tambahan juga dilakukan metode yang sama kepada pemilik peternakan lainnya melalui media sosial.

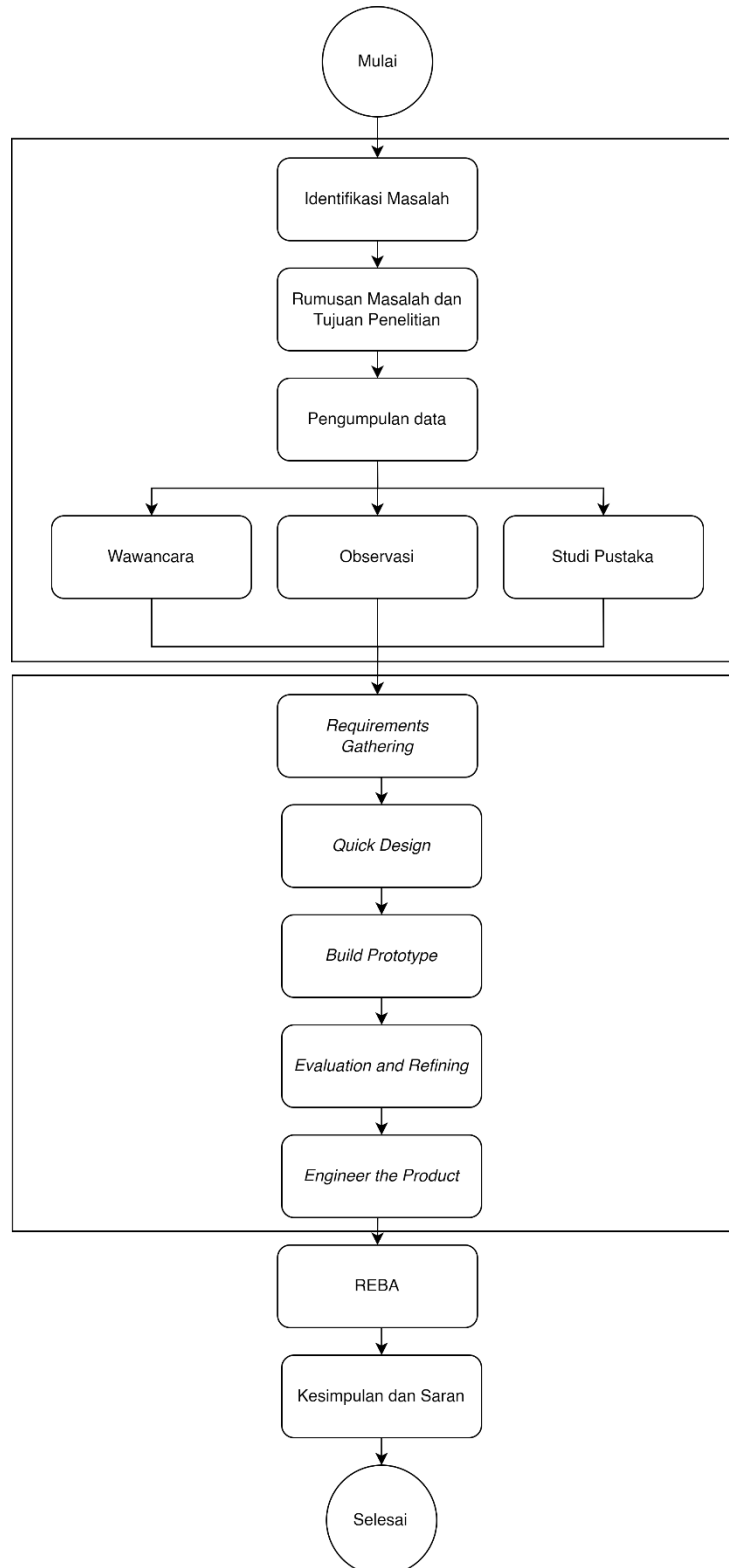
2. Data Sekunder

Data sekunder adalah jenis data yang tidak diperoleh secara langsung melainkan berasal dari sumber lain. Beberapa contoh data sekunder yang dibutuhkan adalah artikel jurnal, buku, *website*, dan sumber lainnya yang relevan. Data sekunder ini akan digunakan untuk mendukung dan melengkapi data primer yang telah diperoleh.

3.3 Alur Penelitian

Alur penelitian merupakan runtutan pengerjaan penelitian dari awal hingga penelitian selesai.

Berikut merupakan *flowchart* dari alur penelitian yang dirancang:



Gambar 3. 1 *Flowchart* Alur Penelitian

Berikut merupakan keterangan terkait *flowchart* alur penelitian yang telah ditampilkan:

1. Identifikasi Masalah

Menganalisis akar penyebab dari permasalahan yang berkaitan dengan pengelolaan peternakan ayam *broiler*.

2. Rumusan Masalah

Bagaimana rancangan *microcontroller* Arduino IDE untuk menunjang pengelolaan ternak ayam agar lebih efisien?

3. Tujuan Penelitian

Merancang *microcontroller* Arduino IDE untuk menunjang pengelolaan ayam agar lebih efisien.

4. Pengumpulan Data

Pada pengumpulan data dibagi menjadi tiga metode yaitu:

a. Wawancara

Wawancara dilakukan secara langsung kepada pemilik peternakan ayam di Jalan Pandanaran, pertanyaan seputar kegiatan pengelolaan ternak beserta dengan kesulitan yang dialami selama proses pengelolaan. Wawancara juga dilakukan secara daring kepada pemilik peternakan ayam di kota lampung yang digunakan sebagai data tambahan.

b. Observasi

Observasi dilakukan secara langsung pada peternakan yang bertempat di Jalan Pandanaran. Kegiatan seperti pemberian makan, minum, penerangan cahaya, dan pengaturan ventilasi menjadi acuan dalam pengembangan *prototype*.

c. Studi Pustaka

Tahapan ini dilakukan dengan mengkaji berbagai jurnal nasional maupun internasional berkaitan dengan pengembangan sistem menggunakan Arduino dengan tujuan memperkaya ilmu dalam mendukung penelitian.

5. *Requirements Gathering*

Berdasarkan hasil wawancara dan observasi, maka dapat dilakukan analisis kebutuhan terkait pengembangan sistem kedepannya. Pada tahap ini juga ditentukan *output* yang ditargetkan secara garis besar yakni berfokus pada pengelolaan, *monitoring*, dan *controlling*.

6. *Quick Design*

Berdasarkan hasil analisis kebutuhan pada tahap sebelumnya, dilakukan pembuatan desain sederhana berupa gambaran dasar terkait perancangan sistem. Selain itu, terdapat penentuan kebutuhan *hardware* dan *software* dalam proses perancangan berdasarkan kebutuhan yang diharapkan oleh *user*.

7. *Build Prototype*

Pada tahap ini dilakukan perancangan skematik pada sistem menggunakan perangkat lunak proteus, dilanjutkan dengan pengkodean pada perangkat lunak Arduino IDE yang akan diinputkan ke dalam *microcontroller*. Selain itu, terdapat proses penyetelan pada *Blynk Cloud* yang menjadi wadah utama *monitoring* dan *controlling* sistem berbasis IoT.

8. *Evaluation and Refining*

Setelah *prototype* awal berhasil dirancang, maka perlu dilakukan *testing* kepada calon pengguna yaitu pemilik peternakan ayam pandanaran, dengan begitu didapatkan umpan balik (*feedback*) terkait produk yang dirancang. Apabila terdapat revisi perbaikan yang perlu dilakukan, maka dapat langsung dilakukan pembenahan maupun penambahan pada sistem.

9. *Engineer the Product*

Setelah sudah tidak terdapat revisi perbaikan, maka produk final akan segera dibuat berdasarkan *prototype* akhir dan dapat diartikan sebagai produk matang. Sistem yang telah dirancang, dapat langsung diintegrasikan menjadi satu kesatuan yang diinput ke dalam *microcontroller* beserta *Blynk Cloud* yang telah dihubungkan dalam proses pengkodean.

10. REBA

Berdasarkan perancangan produk final, dilakukan evaluasi manfaat terhadap pekerja peternakan yang dijadikan sebagai pokok permasalahan dibuatnya sistem automasi ini dengan bantuan metode REBA. Terdapat perbandingan beban kerja yang dialami oleh pada saat sebelum ada penerapan pengelolaan peternakan otomatis dan setelah diimplementasikan perancangan agar dapat memvalidkan manfaat dari perancangan sistem pada penelitian ini.

11. Kesimpulan dan Saran

Membuat kesimpulan berdasarkan penelitian yang diteliti serta memberikan rekomendasi pengembangan pada peneliti selanjutnya.

3.4 Analisis Hasil

Analisis hasil dilakukan berdasarkan hasil pengujian pada *prototype* sistem dalam melakukan setiap tugasnya. Dari setiap *output* yang diatur di dalam sistem dilakukan pengecekan apakah sudah sesuai berdasarkan pengkodean yang di input-kan ke dalam *microcontroller*.

3.5 Penutup

Pada bagian penutup, terdapat kesimpulan dan saran. Kesimpulan ditarik berdasarkan tujuan dari penelitian ini, sedangkan saran ditarik berdasarkan kekurangan yang didapati pada penelitian ini, sehingga dapat diteruskan pada penelitian yang akan mendatang.

BAB IV

PEMBANGUNAN SISTEM

4.1 *Requirements Gathering*

Requirements Gathering atau analisis kebutuhan merupakan langkah pertama dalam perancangan *prototype*. Pada tahap ini kebutuhan sistem didefinisikan dengan rinci berdasarkan hasil diskusi dua arah, yaitu dari kebutuhan pihak pengguna dan pihak pengembang. *Prototype* ini dirancang untuk membantu pengelola peternak ayam yang sebelumnya masih menggunakan metode konvensional atau semi modern.

Prototype yang dikembangkan ini merupakan sistem automasi yang meminimalisir kegiatan campur tangan manusia dalam proses pengelolaan ternak ayam. Kelebihan dari rancang bangun ini terdapat pada pengembangan berbasis *Internet of Things* (IOT) yang dapat sangat membantu pengguna dalam memonitor hingga mengontrol peternakan dari jarak jauh atau secara daring (*online*).

4.1.1 Kebutuhan Pengguna

Pada tahap ini dilakukan analisis terhadap kebutuhan pengguna yaitu pihak peternak ayam. Dalam proses pengelolaannya untuk mengetahui apa saja yang perlu dikembangkan peneliti menggunakan metode wawancara langsung kepada pihak pengguna. Selain itu, dilakukan juga observasi terkait sistem dan mekanisme pada struktur pengelolaan ternak ayam secara langsung sebagai acuan dalam proses pengembangan sistem automasi.

Setelah mendapat berbagai informasi terkait kebutuhan sistem yang dibutuhkan oleh pihak pengguna, maka dapat memfokuskan analisis yang dibutuhkan sistem agar dapat dikonversi menjadi sistem automasi. Hasil diskusi bersama dengan pihak pengguna menjadi acuan dalam perencanaan rancang bangun sistem automasi terkait proses pengelolaan ternak ayam yang tepat. Hal ini tetap memperhatikan saran serta masukan yang diberikan sesuai dengan kebutuhan pengguna.

Tabel 4. 1 Kebutuhan Pengguna

No	Kegiatan	Kondisi Saat Ini	Solusi
1.	Pengelolaan	Kegiatan pengelolaan peternakan masih secara tradisional dengan kebutuhan tenaga manusia yang banyak.	Memudahkan proses kegiatan pengelolaan peternakan tanpa harus menggunakan banyak tenaga manusia. Kegiatan pengelolaan yang perlu dilakukan automasi yaitu: pemberian pakan, pemberian minum, pencahayaan, dan pengaturan suhu ternak.
2.	<i>Monitoring and controlling</i>	Kegiatan pemantauan serta pengontrolan peternakan masih dilakukan dengan rutin secara tradisional oleh pekerja secara langsung.	Memudahkan proses pemantauan dan kontrol sistem peternakan ayam sehingga peternak tidak perlu bolak balik ke peternakan untuk melakukan pengecekan dan pengelolaan melalui sistem <i>cloud</i> dengan bantuan IoT.

4.1.2 Kebutuhan Dasar Sistem

4.1.2.1 Dasar perancangan kontrol temperatur

Pada kondisi saat ini, peternakan ayam di Jalan Pandanaran belum memenuhi standar dikarenakan belum ada alat yang digunakan untuk mengukur temperatur lingkungan peternakan. Suhu lingkungan dapat berimbas pada pertumbuhan ayam seperti yang dijelaskan oleh (DITJENNAK, 2018) bahwa untuk masa pertumbuhan dan perkembangan ayam pedaging di Indonesia, suhu lingkungan yang optimal adalah sebagai berikut:

Tabel 4. 2 Suhu Lingkungan Optimal

Periode	Suhu
Masa pertumbuhan awal (0-3 minggu)	32°C hingga 35°C
Masa pertumbuhan menengah (3-6 minggu)	28°C hingga 30°C
Masa pertumbuhan akhir (6-8 minggu)	25°C hingga 28°C

4.1.2.2 Dasar perancangan penjadwalan pemberian pakan

Penjadwalan pemberian pakan yang teratur akan berdampak pada hewan ternak, dengan adanya penjadwalan ayam akan memperoleh nutrisi yang seimbang dan mencukupi untuk memenuhi kebutuhan metabolisme dan kesehatan mereka. Ini membantu mencegah masalah kesehatan dan defisiensi nutrisi yang dapat mempengaruhi pertumbuhan, kekebalan tubuh, dan kualitas daging ayam. Menurut (DITJENNAK, 2018) pemberian pakan yang optimal dapat dilakukan pada waktu-waktu berikut:

1. Pagi hari: Pemberian pakan pertama dilakukan pada pagi hari sekitar pukul 6:00 hingga 8:00. Hal ini memungkinkan ayam untuk mendapatkan energi yang cukup untuk memulai aktivitas harian mereka.
2. Siang hari: Pemberian pakan kedua dilakukan pada siang hari sekitar pukul 12:00 hingga 14:00. Pemberian pakan pada waktu ini membantu menjaga energi dan metabolisme ayam selama periode aktif mereka.
3. Sore hari: Pemberian pakan ketiga dilakukan pada sore hari sekitar pukul 16:00 hingga 18:00. Ini memberikan asupan nutrisi tambahan sebelum ayam beristirahat untuk malam hari.

Selain penjadwalan pakan, jumlah pemberian pakan juga perlu disesuaikan dengan umur ayam. Berikut merupakan data kebutuhan pakan ayam menurut (BPP, 2019):

Tabel 4. 3 Kebutuhan Pakan Ayam

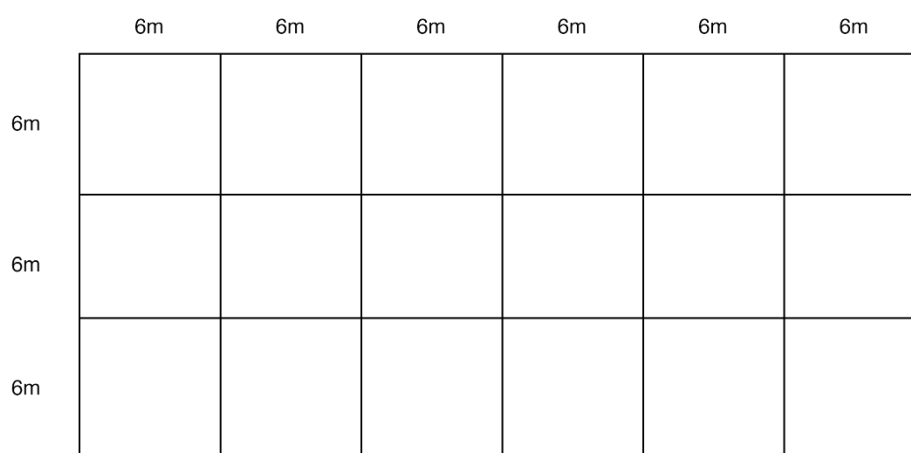
Periode	Kebutuhan per 100 ekor
Masa pertumbuhan awal (0-3 minggu)	2 Kg
Masa pertumbuhan menengah (3-6 minggu)	6 Kg
Masa pertumbuhan akhir (6-8 minggu)	9 Kg

4.1.2.3 Dasar perancangan pencahayaan

Pencahayaan pada peternakan ayam perlu diperhatikan karena memiliki dampak yang signifikan terhadap kesehatan, pertumbuhan, dan produktivitas ayam. Pencahayaan yang tepat dapat membantu mengatur siklus fisiologi ayam, seperti siklus tidur dan bangun serta membantu mempertahankan keseimbangan hormon dan mempengaruhi ritme internal mereka. Untuk berternak ayam pedaging di Indonesia intensitas cahaya yang cocok adalah minimal 20 lux hingga 50 lux maksimal (DITJENNAK, 2018).

4.1.3 Pengambilan Data LKF Peternakan

Pengambilan data LKF pada peternakan ayam di Jalan Pandanaran dengan luas bangunan 648m² dilakukan pada beberapa titik pengujian. Data yang diambil meliputi suhu lingkungan dan intensitas cahaya pada lingkungan. Berikut merupakan titik pengujian lingkungan kerja pada peternakan ayam di Jalan Pandanaran:



Gambar 4. 1 Titik Pengujian LKF

4.1.3.1 Pengambilan data suhu

Pengambilan data suhu pada peternakan ayam dilakukan pada jam 14:00 WIB dan didapatkan hasil rekapitulasi sebagai berikut:

	6m	6m	6m	6m	6m	6m
6m	30	30	30	30	30	30
6m	31	30	30	30	29	30
6m	30	30	30	30	30	30

Gambar 4. 2 LKF Suhu

4.1.3.2 Pengambilan data pencahayaan

Pengambilan data pencahayaan pada peternakan ayam dilakukan pada jam 18:00 dengan kondisi tanpa bantuan pencahayaan dan didapatkan hasil rekapitulasi sebagai berikut:

	6m	6m	6m	6m	6m	6m
6m	5	5	5	5	5	5
6m	5	5	5	5	5	5
6m	5	4	5	5	5	5

Gambar 4. 3 LKF Intensitas Cahaya 1

Dan setelah diberikan pencahayaan didapatkan hasil rekapitulasi sebagai berikut:

	6m	6m	6m	6m	6m	6m
6m	88	88	88	88	88	88
6m	124	124	124	124	124	124
6m	88	88	88	88	88	88

Gambar 4. 4 LKF Intensitas Cahaya 2

4.2 Quick Design

4.2.1 Peralatan Penelitian

Dalam membangun *prototypie* pada penelitian ini dibutuhkan peralatan yang mendukung perancangan sistem. Terdapat dua macam peralatan yang digunakan, yaitu perangkat keras (*hardware*) dan perangkat lunak (*software*). Berikut merupakan keterangan dari masing-masing peralatan yang digunakan:

1. Perangkat keras (*hardware*)

Perangkat keras atau yang disebut *hardware* merupakan komponen yang memiliki wujud fisik yang dapat dipegang dan juga dapat dilihat. Dalam penggunaannya biasanya dibantu oleh perangkat lunak.

Tabel 4. 4 Perangkat Keras Penelitian

No	Nama dan Gambar Alat	Keterangan
1.	Mikrokontroler ESP-32 	ESP-32 merupakan jenis mikrokontroler penerus dari ESP-8266. Mikrokontroler ini telah dilengkapi dengan modul WiFi didalamnya sehingga penggunaan ESP-32 ini dijadikan pilihan peneliti dalam penelitian ini sesuai dengan tujuan pengelolaan dengan pemanfaatan IoT.
2.	Laptop 	Perangkat laptop merupakan alat yang digunakan dalam melakukan pengkodean (<i>coding</i>) serta melakukan <i>monitoring</i> dan <i>controlling</i> sistem yang akan dibuat.

Gambar 4. 5 Perangkat ESP32

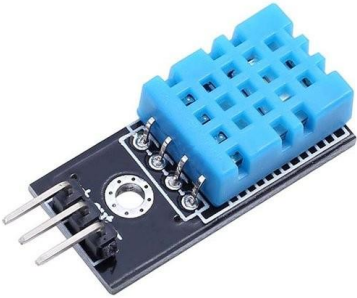
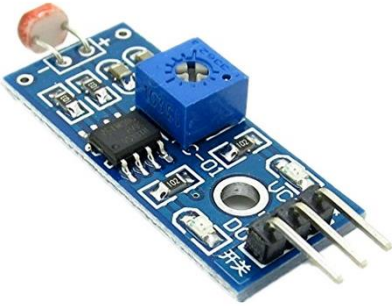
(Sumber: <https://raharja.ac.id/mikrokontroler-esp32-3/>)

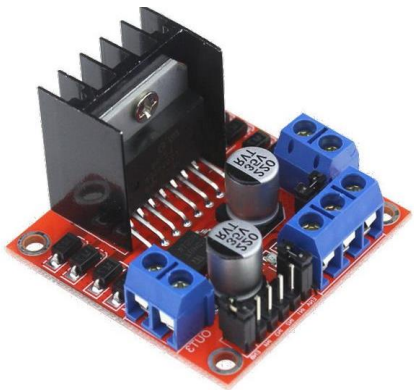
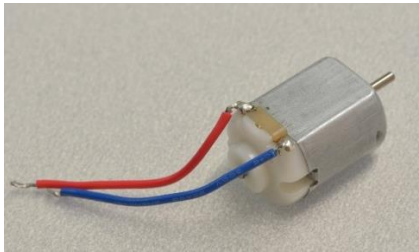
Gambar 4. 6 Perangkat Laptop

(Sumber: <https://laptopnesia.com/>)

No	Nama dan Gambar Alat	Keterangan
3.	<p>Kabel USB</p>  <p>Gambar 4. 7 Kabel USB</p> <p>(Sumber: https://uzone.id/-ini-beda-kabel-usb-type-c-lightning-dan-micro-usb)</p>	<p>Kabel USB merupakan alat penghubung antara laptop dengan mikrokontroler ESP-32, kabel ini berfungsi sebagai alat bertukar informasi dari kedua perangkat seperti pengunggahan kode ke dalam mikrokontroler. Selain itu, kabel USB juga dapat digunakan sebagai sumber daya (<i>power supply</i>) oleh mikrokontroler ESP-32.</p>
4.	<p>Kabel <i>Jumper</i></p>  <p>Gambar 4. 8 Kabel <i>Jumper</i></p> <p>(Sumber: https:// jualarduinojogja.com/kabel-jumper-female-to-female/)</p>	<p>Kabel <i>jumper</i> merupakan alat penghubung antar komponen pada proyek arduino, kabel ini memungkinkan konduksi listrik yang aman tanpa harus melakukan proses solder karena memiliki pin di setiap ujungnya.</p>
5.	<p>Kabel <i>Jumper</i> Buaya</p>  <p>Gambar 4. 9 Kabel <i>Jumper</i> Buaya</p> <p>(Sumber: https://tokopedia.com/electr/kabel-jumper-jepit-buaya)</p>	<p>Kabel <i>jumper</i> buaya tidak jauh beda fungsinya dengan kabel <i>jumper</i> pada umumnya, hanya saja pada ujung kabel tidak berbentuk pin, melainkan berbentuk jepitan yang dapat menjadi konduktor apabila ada rangkaian yang memiliki serabut tembaga.</p>

No	Nama dan Gambar Alat	Keterangan
6.	<p>Adaptor 9V</p>  <p>Gambar 4. 10 Adaptor 9V (Sumber: https://ecadio.com/jual-adaptor-9v-uno)</p>	<p>Adaptor merupakan sumber daya yang digunakan dalam rangkaian perangkat, penggunaan adaptor dengan kapasitas 9V ini digunakan sebagai tambahan <i>power supply</i> pada mikrokontroler ESP-32 agar hantaran listrik tidak kekurangan daya yang disebabkan oleh banyaknya sensor yang digunakan.</p>
7.	<p>Adaptor 12V 12V/3A</p>  <p>Gambar 4. 11 Adaptor 12V (Sumber: https://lazada.co.id/products/power-adaptor-led-strip-monitor-dc-12v-3a)</p>	<p>Penggunaan adaptor dengan kapasitas 12V ini digunakan sebagai sumber daya untuk modul L298N yang memerlukan daya sebesar 12V.</p>
8.	<p>LCD I2C 16x2</p> 	<p>LCD atau <i>liquid crystal display</i> merupakan media penampil <i>interface</i> pada mikrokontroler. Pada proyek ini LCD yang digunakan berukuran 16x2 yang berfungsi sebagai alat pemantau jalannya program.</p>

No	Nama dan Gambar Alat	Keterangan
9.	<p>Sensor DHT11</p>  <p>Gambar 4. 12 Modul Sensor DHT11</p> <p>(Sumber: https://digiwarestore.com/temperature-humidity-sensor-module/)</p>	<p>Sensor DHT11 merupakan modul yang berfungsi sebagai alat penangkap suhu dan kelembaban suatu ruangan, <i>output</i> dari sensor ini berupa tegangan analog yang nantinya akan diolah lebih lanjut menggunakan mikrokontroler ESP-32.</p>
10.	<p>Sensor Ultrasonic HC-SR04</p>  <p>Gambar 4. 13 Sensor HC-SR04</p> <p>(Sumber: https://tokopedia.com/arduino-robot/sensor-ultrasonik-hc-sr04)</p>	<p>Sensor ultrasonic HC-SR04 merupakan modul yang berfungsi sebagai alat pengukur jarak dari suatu objek. Modul ini memancarkan <i>pulse</i> ultrasonik yang nantinya dipantulkan oleh objek didepannya dan ditangkap kembali. Kisaran jarak yang dapat diukur oleh modul ini sekitar 2 cm hingga 450cm.</p>
11.	<p>Sensor LDR</p>  <p>Gambar 4. 14 Modul Sensor LDR</p> <p>(Sumber: https://aliexpress.com/item/32965201554)</p>	<p>Sensor <i>light dependent resistor</i> (LDR) merupakan modul yang berfungsi sebagai alat pengukur cahaya. Resistor dapat mengalami perubahan resistensi apabila terdapat perbedaan penerimaan cahaya. <i>Output</i> dari modul ini berupa digital yaitu 0 atau 1 tergantung dengan pengaturan yang dibuat melalui kotak pin digital menggunakan obeng.</p>

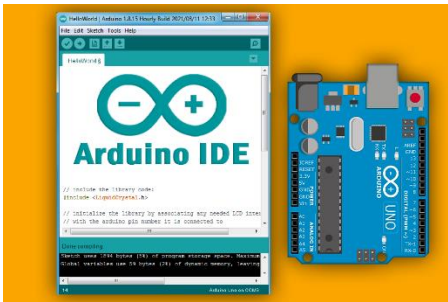
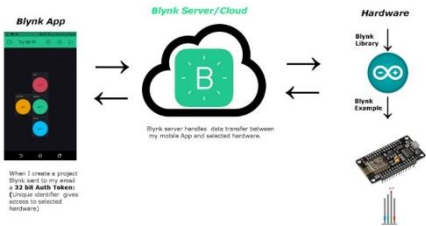
No	Nama dan Gambar Alat	Keterangan
12.	<p>Modul L298N</p>  <p>Gambar 4. 15 Modul L298N</p> <p>(Sumber: https://edukasi elektronik.com/2020/12/modul-driver-motor-l298n)</p>	<p>Modul L298N merupakan modul yang berfungsi sebagai <i>driver</i> atau alat pengontrol kecepatan dan arah perputaran alat berbasis DC. Pada proyek ini modul L298N digunakan sebagai alat pengatur kecepatan pada DC Fan dan DC Motor. Input yang digunakan oleh modul L298N sebesar 12 volt, maka dari itu sumber daya modul ini menggunakan adaptor berkapasitas 12 volt.</p>
13.	<p>DC Fan 12V</p>  <p>Gambar 4. 16 DC Fan 12V</p> <p>(Sumber: https://shopee.co.id/Cooling-Fan-KIPAS-12V-DC-Brushless-pendingin)</p>	<p>DC fan merupakan kipas penggerak satu arah yang berfungsi sebagai alat pengatur sirkulasi udara. DC fan yang digunakan pada penelitian ini berkapasitas 12 volt yang kecepatan perputarannya diatur berdasarkan suhu ruang yang ditangkap oleh sensor DHT11 serta diolah oleh mikrokontroler ESP-32.</p>
14.	<p>DC Motor</p>  <p>Gambar 4. 17 DC Motor</p> <p>(Sumber: https://bc-robotics.com/tutorials/controlling-dc-motor-arduino/)</p>	<p>DC motor merupakan perangkat pengubah energi listrik menjadi energi kinetik. Pada DC motor terdapat kumparan yang apabila dialiri listrik dapat berputar menjadi alat penggerak. Pada penelitian ini, DC motor digunakan sebagai alat penarik pakan ternak dari <i>storage</i> atau penyimpanan secara otomatis.</p>

No	Nama dan Gambar Alat	Keterangan
15.	<p>LED WS2812B</p>  <p>Gambar 4. 18 LED WS2812B</p> <p>(Sumber: https:// amazon.com/BTF-LIGHTING-Flexible-Individually-Addressable)</p>	<p>LED WS2812B merupakan lampu yang sangat kompatibel untuk proyek arduino karena dapat dilakukan pengkodean pada masing-masing lampu yang tersedia. Pada penelitian ini LED ini digunakan sebagai sumber pencahayaan yang mendapati sinyal dari modul LDR apabila pencahayaan ruangan berkurang maka LED ini akan menyala secara otomatis.</p>

2. Perangkat lunak (*software*)

Perangkat lunak atau yang disebut *software* merupakan sebuah sistem program pada komputer yang menjembatani antara pengguna dengan perangkat keras yang akan digunakan.

Tabel 4. 5 Perangkat Lunak Penelitian

No	Nama dan Gambar Alat	Keterangan
1.	Arduino IDE 	Arduino IDE atau <i>integrated development environment</i> merupakan <i>software</i> atau perangkat lunak yang digunakan untuk membuat <i>sketch</i> atau sketsa pemrograman. Perangkat lunak ini digunakan sebagai media untuk pembuatan program, mengedit, serta mengunggah program ke dalam <i>board</i> ESP-32 menggunakan bahasa Java yang dilengkapi oleh <i>library</i> C++ sehingga dapat membuat pengaturan operasi input dan <i>output</i> pada proyek menjadi lebih mudah.
2.	Blynk Cloud 	Blynk merupakan perangkat lunak <i>open-source</i> yang mendukung penggunaan IoT atau <i>internet of things</i> . Perangkat lunak ini memungkinkan pengguna untuk memonitor serta mengontrol perangkat keras yang terhubung pada mikrokontroler yang terhubung dengan WiFi dari jarak jauh menggunakan <i>widget</i> yang tersedia. Penggunaan perangkat lunak ini dapat digunakan melalui laman web maupun <i>smartphone</i> secara <i>real time</i> .

Gambar 4. 19 Arduino IDE

(Sumber: <https://andprof.com/tools/what-is-arduino-software-ide-and-how-use-it/>)

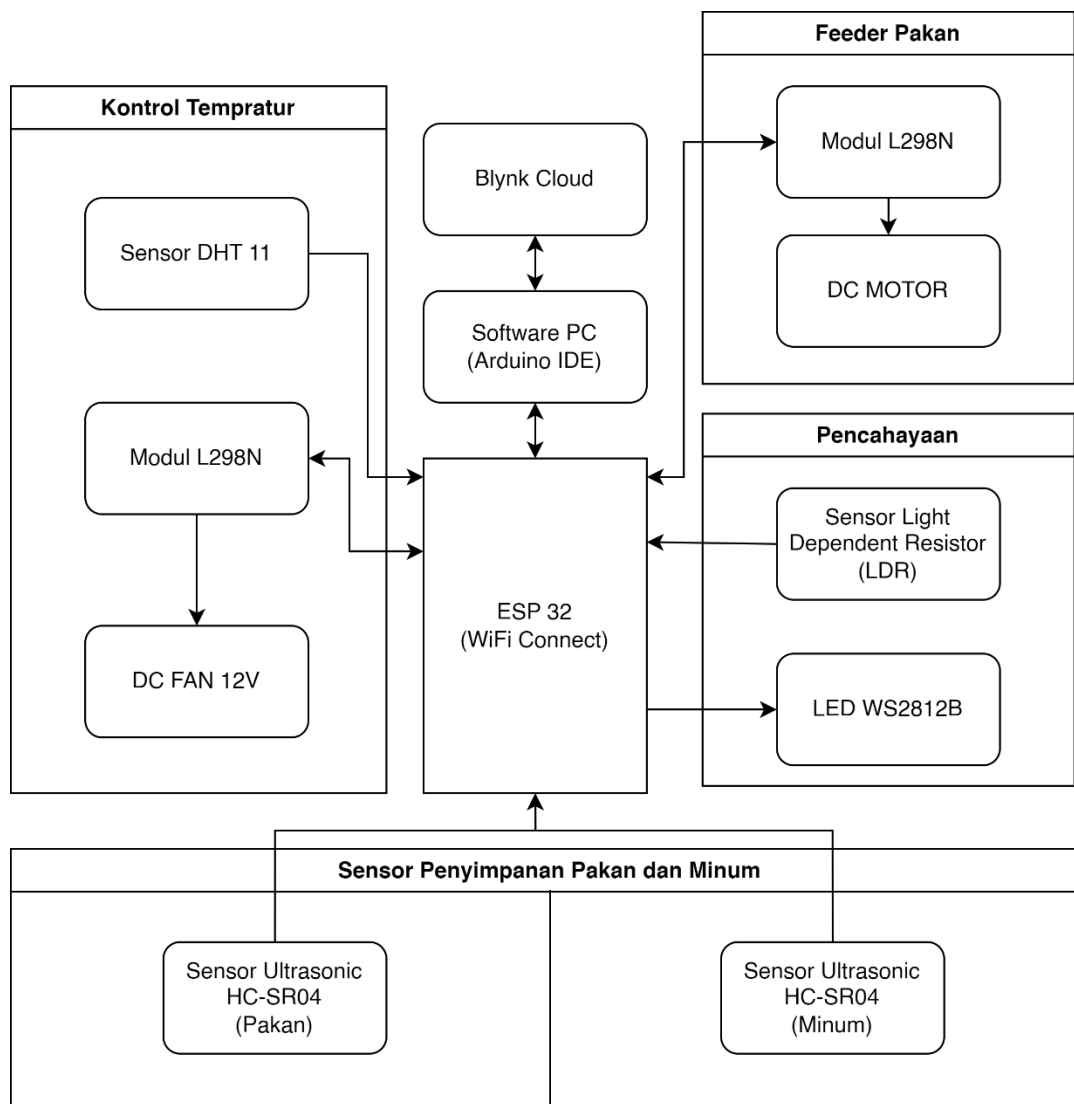
Gambar 4. 20 Blynk Cloud

(Sumber:

<https://lummoshop.com/hsmprojects/products/INT-001/platform-iot-blynk-app?position=5>)

4.2.2 Diagram Blok Perancangan Sistem

Pada tahap ini dilakukan pembuatan desain sederhana berupa gambaran dasar perancangan sistem yang akan dijadikan acuan dalam proses *prototyping*. Desain sederhana yang diterapkan meliputi rangkaian keterhubungan pada setiap alat yang digunakan beserta alur komunikasi yang mempengaruhi kinerja satu sama lainnya. Hal ini sangat penting untuk memudahkan proses perakitan dan gambaran cara kerja masing-masing rangkaian yang akan dibuat. Maka dari itu, berikut bentuk desain sederhana yang berupa diagram blok beserta arah komunikasi dari tiap alat yang digunakan:



Gambar 4. 21 Diagram Blok Desain Sistem

Pada diagram blok yang telah dipaparkan di atas, terdapat beberapa kotak *container* yang memiliki fungsi berbeda satu sama lain. Berikut merupakan keterangan dari masing-masing alat pada tiap *container*:

1. Kontrol Temperatur

Sensor DHT11 mengirimkan data suhu ruangan ke mikrokontroler ESP-32, kemudian data tersebut diolah berdasarkan pengkodean, selanjutnya mikrokontroler ESP-32 mengirim sinyal kepada modul L298N untuk menggerakkan putaran kipas DC FAN 12V. Seluruh pertukaran informasi dalam sistem di-*upload* ke server Blynk Cloud.

2. *Feeder* Pakan

Sistem kerja *feeder* ini menggunakan Blynk Cloud sebagai alat kontrol. Berdasarkan pengkodean yang dilakukan, *switch* untuk menyalakan dan mematikan DC Motor terdapat pada server Blynk Cloud yang mengirimkan informasi ke mikrokontroler ESP-32 untuk memberi sinyal modul L298N untuk menggerakkan perputaran DC Motor.

3. Pencahayaan

Sensor *light dependent resistor* (LDR) mengirimkan data yang ditangkap dari pencahayaan di ruangan, kemudian data tersebut diolah berdasarkan pengkodean pada *software*. Apabila pencahayaan ruangan kurang dari ketentuan pada pengkodean sistem, maka mikrokontroler ESP-32 akan mengirimkan sinyal kepada LED WS2812B untuk menghidupkan lampu ruangan secara otomatis. Seluruh pertukaran informasi dalam sistem di-*upload* ke server Blynk Cloud.

4. Sensor Penyimpanan Pakan dan Minum

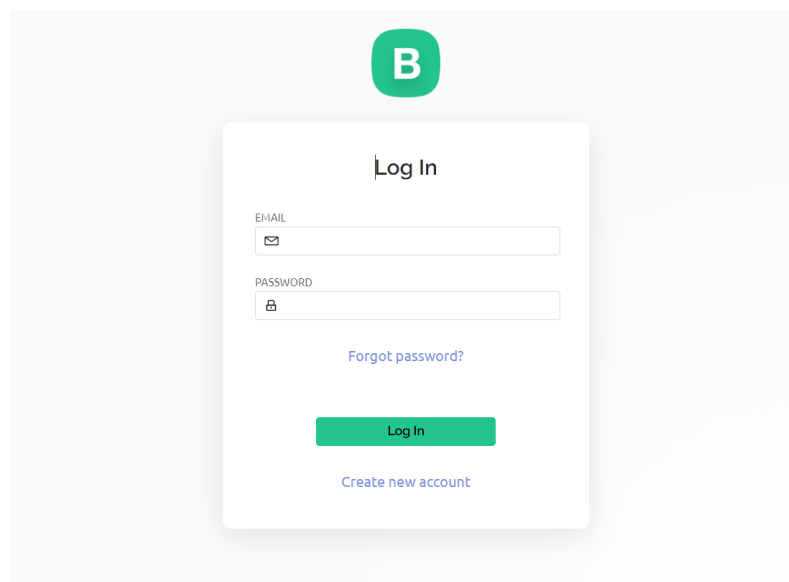
Sistem ini menggunakan sebanyak dua sensor *ultrasonic* HC-SR04 untuk masing-masing kebutuhan, satu untuk pengukuran tempat penyimpanan pakan dan satu sensor lainnya untuk pengukuran tempat penyimpanan air minum. Sensor menangkap jarak pada tempat penyimpanan dan mengirimkan data informasi tersebut ke mikrokontroler ESP-32 yang juga di-*upload* pada server Blynk Cloud untuk dilakukan *monitoring*.

4.3 Build Prototype

4.3.1 Penyetelan Awal Blynk Cloud

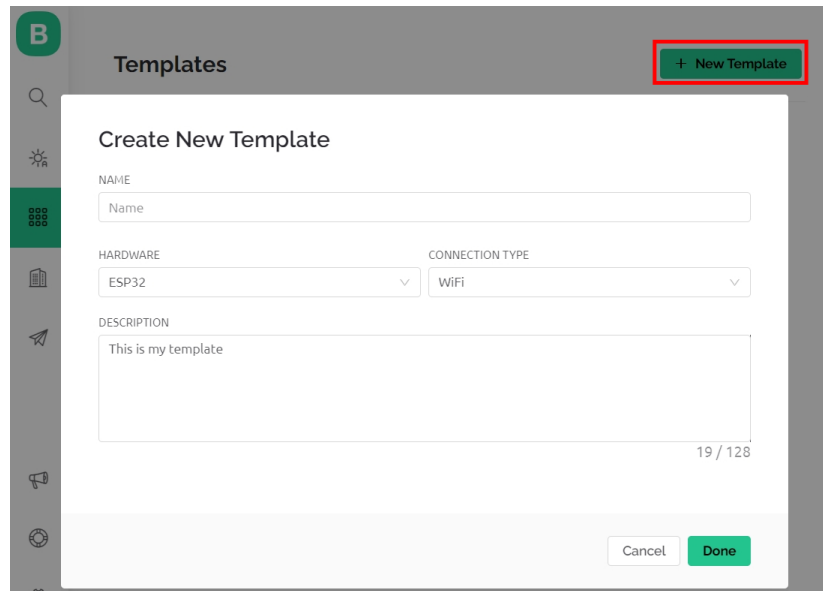
Blynk merupakan *platform* IoT yang dapat digunakan melalui laman web maupun *smartphone* untuk mengendalikan proyek mikrokontroler arduino yang terhubung dengan internet. *Platform* ini memiliki banyak fitur yang tersedia sehingga memudahkan pengguna mengaplikasikan ke dalam proyek. Berikut merupakan tahapan penyetelan awal dalam penggunaan Blynk Cloud:

1. Tahap pertama adalah masuk ke laman web <https://sgp1.blynk.cloud/dashboard/login> dan melakukan pendaftaran akun atau apabila telah memiliki akun dapat langsung melakukan *login*.



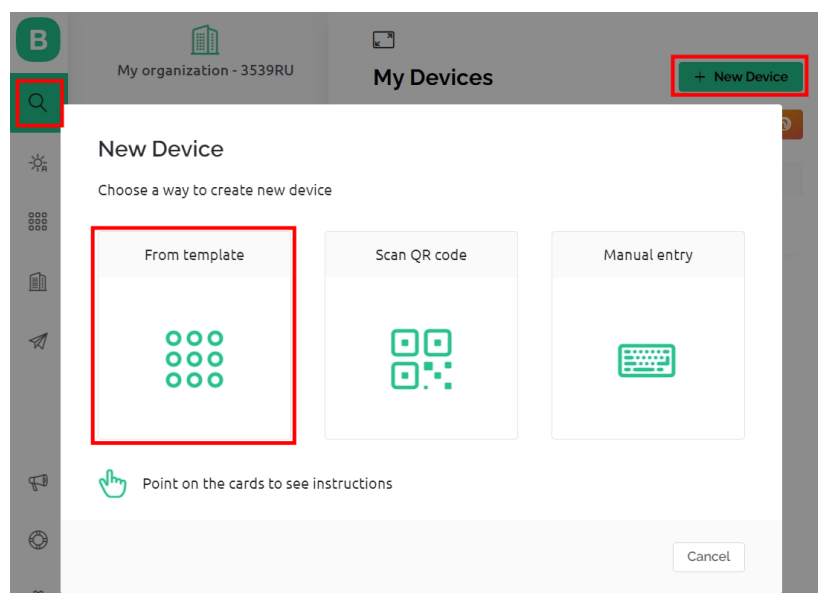
Gambar 4. 22 Laman Login Blynk Cloud

- Setelah membuat akun Blynk Cloud, klik pada “Templates” dan “+ New Template” untuk membuat sistem baru. Selanjutnya mengisi nama proyek, *hardware* ESP32, dan *connection type* yaitu WiFi. Klik *done* apabila data telah dilengkapi.



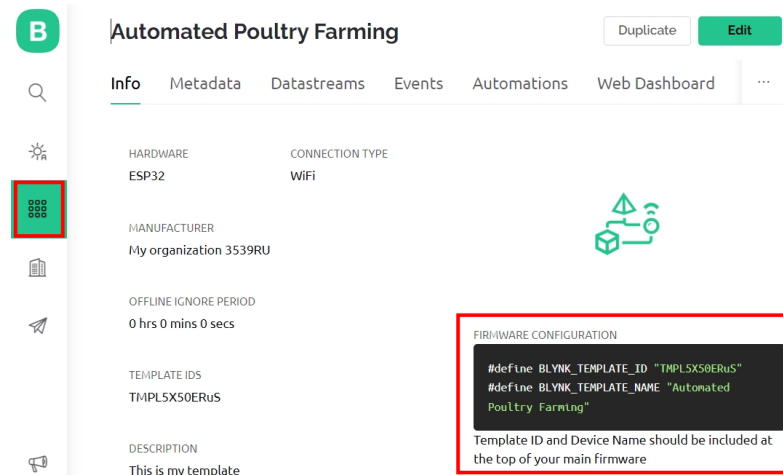
Gambar 4. 23 Membuat *Template* Blynk

- Tahapan berikutnya adalah klik laman *search* dan “+ New Device”. Dalam pembuatan perangkat baru, pilih sistem yang telah dibuat pada tahap sebelumnya dengan mengklik *from template* lalu pilih sesuai nama *template* yang telah dibuat.



Gambar 4. 24 Menu *New Device* Blynk

4. Setelah itu kembali pada menu “Templates” untuk mendapatkan *template_id* dan *template_name* yang akan digunakan untuk proses pengkodean pada perangkat lunak Arduino IDE.



Gambar 4. 25 *Firmware Configuration* untuk Pengkodean

4.3.2 Pembangunan *Feeder* Pakan

4.3.2.1 Peralatan *feeder* pakan

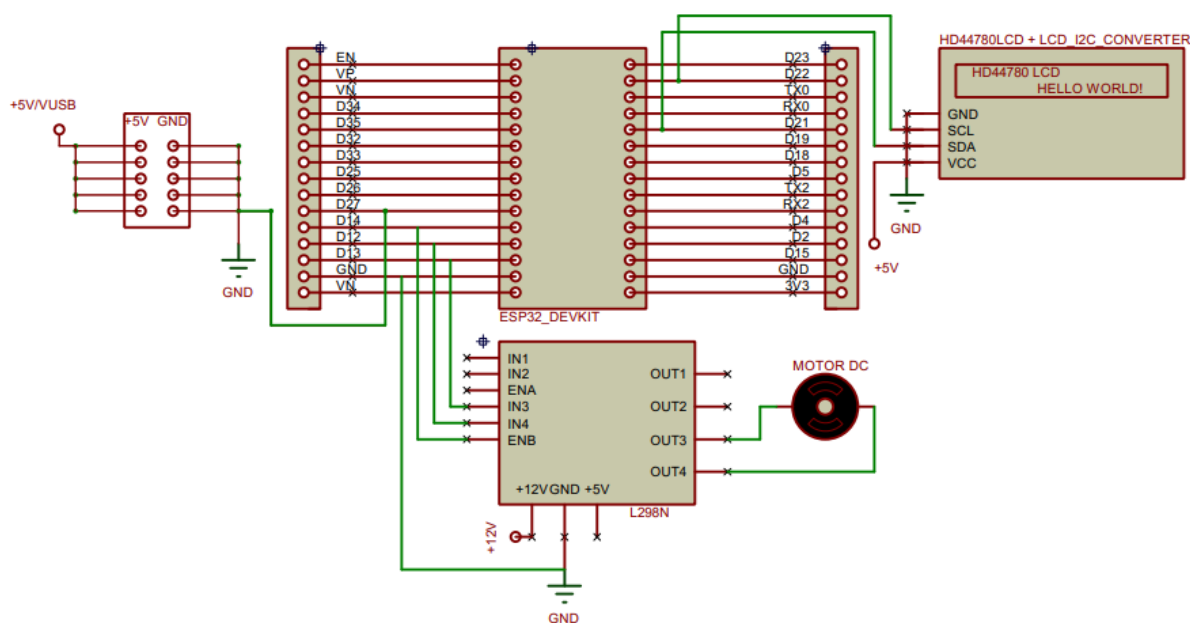
Dalam pembangunan sistem *feeder* pakan otomatis, diperlukan peralatan sebagai berikut:

Tabel 4. 6 Peralatan Feeder Pakan

No	Nama Alat	Banyaknya
1.	Mikrokontroler ESP-32	1
2.	DC Motor	1
3.	Modul L298N	1
4.	LCD I2C 16x2	1
5.	Kabel <i>Jumper</i>	8
6.	Adaptor 12V	1

4.3.2.2 Skematik *feeder* pakan

Skematik *feeder* pakan dirancang menggunakan *software* Proteus. *Software* ini sangat berguna dalam mendesain rangkaian elektronik sebelum proses perangkaian sebenarnya dilakukan. Berikut merupakan skematik pada *feeder* pakan:



Gambar 4. 26 Skematik *Feeder* Pakan Proteus

Keterangan:

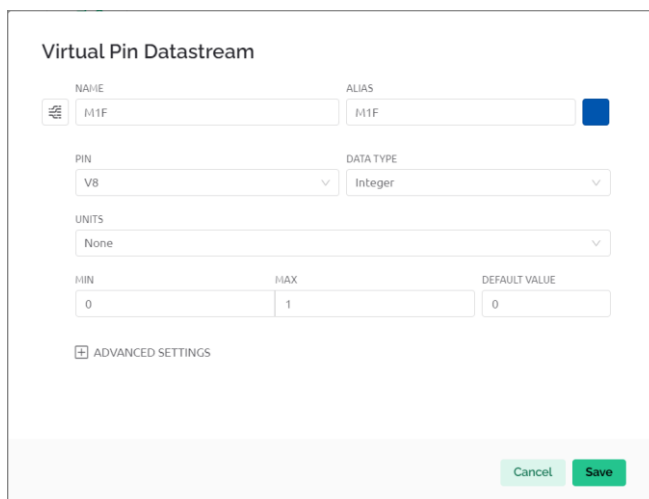
Tabel 4. 7 Keterangan Skematik *Feeder* Pakan

Awalan	Sambungan
Pin D22 pada ESP-32	Pin SCL pada LCD
Pin D21 pada ESP-32	Pin SDA pada LCD
Pin D13 pada ESP-32	Pin IN3 pada modul L298N
Pin D12 pada ESP-32	Pin IN4 pada modul L298N
Pin D14 pada ESP-32	Pin ENB pada modul L298N
Pin GND pada ESP-32	Pin GND pada modul L298N
Pin OUT3 pada modul L298N	Pin + pada DC Motor
Pin OUT4 pada modul L298N	Pin – pada DC Motor
Kabel + pada Adaptor 12V	Pin +12V pada L298N
Kabel – pada Adaptor 12V	Pin GND pada modul L298N

4.3.2.3 Setting Blynk Cloud feeder pakan

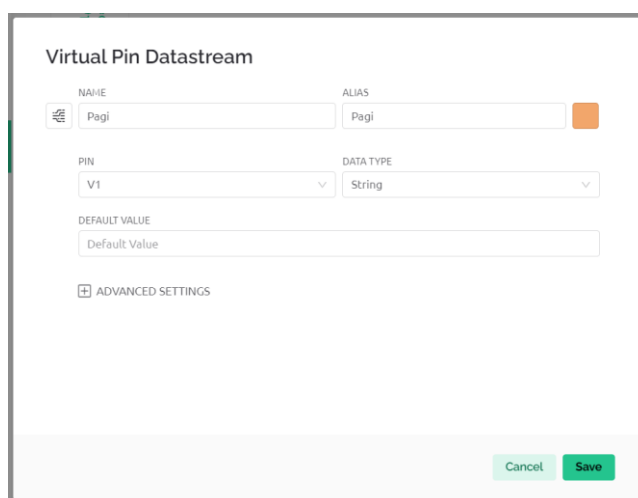
Sebelum masuk proses penerapan kode pada sistem, perlu dilakukan penyetelan input/output pada Blynk Cloud. Berikut merupakan tahapan penyetelannya:

1. Masuk pada menu “Templates”, kemudian *datastreams*. Pada *datastreams* klik “+ New Datastream” berupa *Virtual Pin*. Pada tampilan *Virtual Pin Datastream*, isi nama, Pin yaitu V8, *Data Type* yaitu *Integer*, *Units* yaitu *None*, dan Min,Max output berupa 0,1. Kemudian *Save and Apply*.



Gambar 4. 27 Menyiapkan *Virtual Pin Feeder Pakan*

2. Masih pada *datastreams* klik kembali “+New Datastream” berupa *Virtual Pin*. Pada tahap ini dibuat aliran data untuk penjadwalan pakan ayam. Isi nama “Pagi”, Pin yaitu V1, *Data Type* yaitu *String*. Lakukan Hal yang sama untuk penjadwalan siang dengan Pin yaitu V9, dan penjadwalan makan sore dengan Pin V6. Kemudian *Save and Apply*.



Gambar 4. 28 Menyiapkan *Virtual Pin Penjadwalan Pakan*

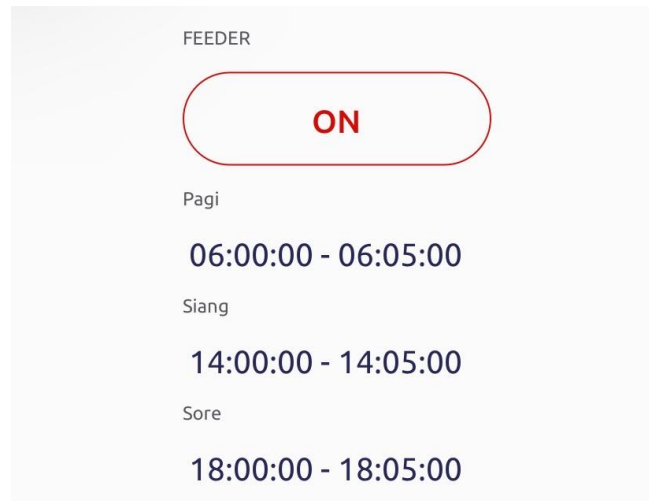
- Selanjutnya masuk pada *web dashboard* dan klik *edit*. Pilih “Switch” pada *widget* yang disediakan pada menu di sebelah kiri. Beri nama, isikan *datastream* V8, *On Value* dan *Off Value* sebesar 1,0 dan klik *save*.

Gambar 4. 29 Menyiapkan *Widget Switch Feeder* Pakan

- Selanjutnya pilih kembali *widget* “Time Input” pada menu di sebelah kiri. Isikan *datastream* yang digunakan yaitu V1, *Time Format* berupa HH:mm:ss, kemudian nyalakan *Allot start/stop input* dan klik *save*. Lakukan hal yang sama untuk penjadwalan makan siang dan sore hari.

Gambar 4. 30 Menyiapkan *Widget Time Input* Pakan

5. Selanjutnya klik pada *widget* untuk mengatur waktu pemberian pakan. Pada *widget* makan pagi, atur *start time* pada jam 06:00:00 dan *end time* 06:05:00 untuk menyalakan feeder pada jam 6 pagi selama 5 menit. Lakukan hal yang sama untuk penjadwalan makan siang dengan ketentuan 14:00:00-14:05:00 dan penjadwalan makan sore dengan ketentuan 18:00:00-18:05:00.



Gambar 4. 31 Penetapan Penjadwalan Jam Pemberian Pakan

4.3.2.4 Script feeder pakan

Berikut merupakan tabel *script* yang berfungsi sebagai penjelasan dari pengkodean sistem yang dilakukan pada *software* Arduino IDE serta diinput ke dalam mikrokontroler ESP-32:

Tabel 4. 8 *Pseudo Code* Feeder Pakan

<i>Pseudo Code</i>	Fungsi
#define BLYNK_TEMPLATE_ID "TMPL5X50ERuS"	Mendefinisikan <i>template id</i> Blynk yang telah didapat saat penyetelan awal.
#define BLYNK_TEMPLATE_NAME "Automated Poultry Farming"	Mendefinisikan <i>name template</i> Blynk yang telah didapat saat penyetelan awal.
#define BLYNK_AUTH_TOKEN "T10irugKyILVfUBSZ5zoq16CBYBHxIFx"	Mendefinisikan token autentikasi yang telah didapat saat penyetelan awal.
#define BLYNK_PRINT Serial	Mendefinisikan <i>serial print</i> Blynk agar dapat dilakukan pemantauan melalui <i>serial monitor</i> .
#include <WiFi.h>	Menginput <i>library</i> WiFi agar dapat menyambungkan koneksi WiFi.
#include <BlynkSimpleEsp32.h>	Menginput <i>library</i> Blynk ESP32.
char auth[] = BLYNK_AUTH_TOKEN;	Memasukkan token autentikasi yang telah didefinisikan sebelumnya.

<i>Pseudo Code</i>	<i>Fungsi</i>
<pre>char ssid[] = "amang"; char pass[] = "123123123"; BlynkTimer timer;</pre>	<p>Memasukkan informasi SSID dan Password koneksi WiFi yang digunakan. Berfungsi sebagai perhitungan waktu <i>boot-up</i> selama mikrokontroler terhubung pada server secara <i>default</i>.</p>
<pre>#include <LiquidCrystal_I2C.h> LiquidCrystal_I2C lcd(0x27, 16, 2);</pre>	<p>Menginput <i>library</i> LCD. Berfungsi sebagai pendefinisian LCD I2C pada <i>library</i>.</p>
<pre>const int motorfreq = 5000; const int motorpwmChannel = 2; const int motorresolution = 8; const int M1PWM = 14; const int M1F = 13; const int M1R = 12;</pre>	<p>Berfungsi sebagai penyetelan sinyal PWM untuk membuat <i>digital pulse</i> agar dapat mengontrol rangkaian secara analog. Memasukkan ketentuan pin yang terhubung antar mikrokontroler ESP32 dengan modul L298N</p>
<pre>#define APP_DEBUG #include <NTPClient.h> #include <WiFiUdp.h> WiFiUDP ntpUDP; const long utcOffsetInSeconds = 25200;</pre>	<p>Mendefinisikan dan menginput <i>library time client</i> untuk menarik waktu <i>real</i>.</p>
<pre>NTPClient timeClient(ntpUDP, "asia.pool.ntp.org", utcOffsetInSeconds); #define APP_DEBUG #include <NTPClient.h> #include <WiFiUdp.h> WiFiUDP ntpUDP;</pre>	<p>Menambahkan <i>offset</i> waktu +7 dari <i>universal time coordinated</i> (UTC) agar menjadi waktu WIB. Mengintegrasikan <i>timeclient</i> UTC menjadi WIB. Mendefinisikan dan menginput <i>library time client</i> untuk menarik waktu <i>real</i>.</p>
<pre>int startTime1; int endTime1; int startTime2; int endTime2; int startTime3; int endTime3;</pre>	<p>Mendeklarasikan waktu mulai dan berhenti pemberian pakan yang dibagi menjadi tiga kali pemberian pakan.</p>
<pre>void setup() {</pre>	<p>Void setup berfungsi sebagai atap penyetelan dari sistem pengkodean yang berisi kode-kode yang hanya satu kali dijalankan saat program baru dimulai.</p>
<pre> lcd.init(); lcd.backlight(); Blynk.begin(auth, ssid, pass);</pre>	<p>Menyalakan sistem LCD. Menjalankan fungsi Blynk agar dapat tersambung pada server.</p>

<i>Pseudo Code</i>	<i>Fungsi</i>
<pre>pinMode(M1PWM, OUTPUT); pinMode(M1F, OUTPUT); pinMode(M1R, OUTPUT); ledcWrite(motorpwmChannel,120); digitalWrite(M1F,LOW); digitalWrite(M1R,LOW); lcd.setCursor(4,0); lcd.print("Starting"); lcd.setCursor(2,1); lcd.print("Components.."); delay(2500); lcd.clear(); timeClient.begin(); Serial.begin(115200); }</pre>	<p>Mendeklarasikan pin-pin yang telah dideklarasikan sebelumnya menjadi tipe <i>output</i> pada sistem.</p> <p>Menyetel kecepatan perputaran DC motor pada 120 PWM atau 6V.</p> <p>Menyatakan kondisi awalan dari DC motor nonaktif.</p>
<pre>lcd.setCursor(4,0); lcd.print("Starting"); lcd.setCursor(2,1); lcd.print("Components.."); delay(2500); lcd.clear(); timeClient.begin(); Serial.begin(115200); }</pre>	<p>Melakukan penyetelan awal tampilan LCD I2C saat melakukan <i>boot-up</i> sistem.</p>
<pre>timeClient.begin(); Serial.begin(115200); }</pre>	<p>Menjalankan <i>timeclient</i>.</p> <p>Menjalankan fungsi <i>serial monitor</i>.</p>
<pre>BLYNK_WRITE(V8) { int value = param.asInt(); if(value == 1){ digitalWrite(M1F, HIGH); lcd.setCursor(9,0); lcd.print("Feed:*"); } else{ digitalWrite(M1F, LOW); lcd.setCursor(9,0); lcd.print("Feed:o");} }</pre>	<p>Menjalankan sistem <i>write</i> dari <i>widget feeder</i> yang dimana apabila <i>switch</i> di-on kan akan menyebabkan <i>value</i> menjadi 1 dan <i>feeder</i> akan menyala. Sebaliknya apabila <i>switch</i> mati, maka <i>value</i> menjadi 0 dan <i>feeder</i> akan mati.</p>
<pre>BLYNK_WRITE(V1){ startTime1 = param[0].asInt(); endTime1 = param[1].asInt(); if(startTime1 == 0 && endTime1 == 0) { startTime1 = 999999; endTime1 = 999999; } Serial.println(startTime1); Serial.println(endTime1); }</pre>	<p>Menjalankan sistem <i>write</i> dari <i>time input</i> penjadwalan pakan pagi hari.</p>
<pre>BLYNK_WRITE(V9){ startTime2 = param[0].asInt(); endTime2 = param[1].asInt(); if(startTime2 == 0 && endTime2 == 0)</pre>	<p>Menjalankan sistem <i>write</i> dari <i>time input</i> penjadwalan pakan siang hari.</p>

<i>Pseudo Code</i>	<i>Fungsi</i>
<pre> { startTime2 = 999999; endTime2 = 999999; } Serial.println(startTime2); Serial.println(endTime2); } BLYNK_WRITE(V6){ startTime3 = param[0].asInt(); endTime3 = param[1].asInt(); if(startTime3 == 0 && endTime3 == 0) { startTime3 = 999999; endTime3 = 999999; } Serial.println(startTime3); Serial.println(endTime3); } void loop() { timeClient.update(); int HH = timeClient.getHours(); int MM = timeClient.getMinutes(); int SS = timeClient.getSeconds(); int server_time = 3600*HH + 60*MM + SS; if((startTime1 == server_time) (startTime2 == server_time) (startTime3 == server_time)) { digitalWrite(M1F, HIGH); Blynk.virtualWrite(V8, 1); lcd.setCursor(9,0); lcd.print("Feed:*"); else if ((endTime1 == server_time) (endTime2 == server_time) (endTime3 == server_time)) { digitalWrite(M1F, LOW); Blynk.virtualWrite(V8, 0); lcd.setCursor(9,0); lcd.print("Feed:o") </pre>	<p>Menjalankan sistem <i>write</i> dari <i>time input</i> penjadwalan pakan sore hari.</p> <p>Menjalankan <i>library</i> Blynk serta <i>timer</i> Blynk agar dapat berkomunikasi dengan server.</p> <p>Menarik data <i>real time</i> dari <i>time client</i> agar sistem dapat membaca situasi berdasarkan waktu yang telah ditetapkan.</p> <p>Memberi kondisi apabila waktu penjadwalan pemberian pakan sesuai dengan waktu yang terbaca oleh <i>server time client</i> maka <i>feeder</i> akan menyala secara otomatis.</p> <p>Dan apabila waktu penjadwalan pemberhentian pakan sesuai dengan waktu yang terbaca oleh <i>server time client</i> maka <i>feeder</i> akan berhenti secara otomatis.</p>

<i>Pseudo Code</i>	Fungsi
Blynk.run(); timer.run();	Menjalankan <i>library</i> Blynk serta <i>timer</i> Blynk agar dapat berkomunikasi dengan <i>server</i> .

4.3.3 Pembangunan Kontrol Temperatur

4.3.3.1 Peralatan kontrol temperatur

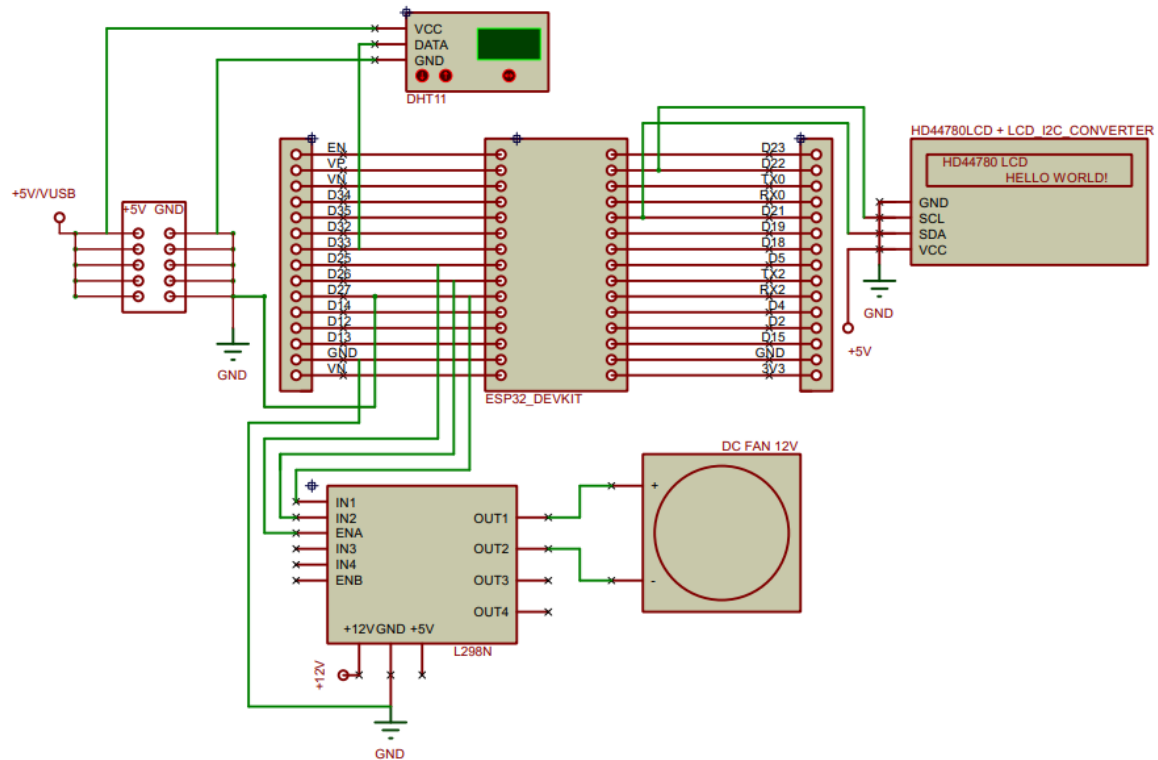
Dalam pembangunan sistem kontrol temperatur, diperlukan peralatan sebagai berikut:

Tabel 4. 9 Peralatan Kontrol Temperatur

No	Nama Alat	Banyaknya
1.	Mikrokontroler ESP-32	1
2.	DC Fan 12V	1
3.	Modul L298N	1
4.	LCD I2C 16x2	1
5.	Sensor DHT11	1
6.	Kabel <i>Jumper</i>	11
7.	Adaptor 12V	1

4.3.3.2 Skematik kontrol temperatur

Skematik kontrol temperatur dirancang menggunakan *software* Proteus. *Software* ini berguna untuk mendesain rangkaian elektronik sebelum proses perangkaian sebenarnya dilakukan. Berikut merupakan skematik pada kontrol temperatur:



Gambar 4. 32 Skematik Kontrol Temperatur Proteus

Keterangan:

Tabel 4. 10 Keterangan Skematik Kontrol Temperatur

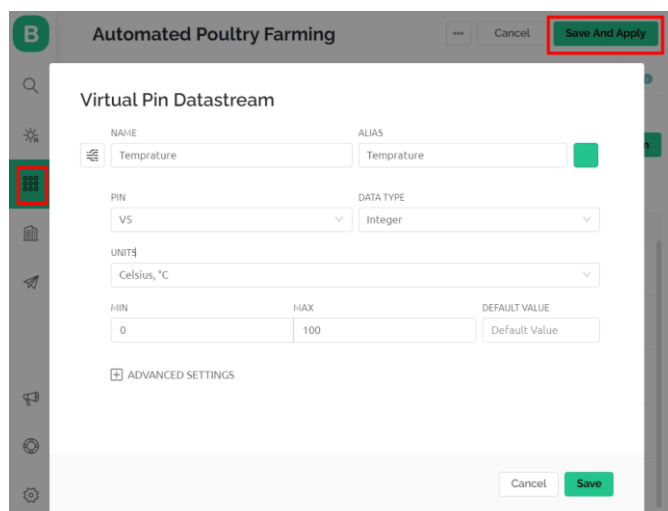
Awalan	Sambungan
Pin D22 pada ESP-32	Pin SCL pada LCD
Pin D21 pada ESP-32	Pin SDA pada LCD
Pin D27 pada ESP-32	Pin IN1 pada modul L298N
Pin D26 pada ESP-32	Pin IN2 pada modul L298N
Pin D25 pada ESP-32	Pin ENA pada modul L298N
Pin GND pada ESP-32	Pin GND pada modul L298N
Pin OUT1 pada modul L298N	Pin + pada DC Fan 12V
Pin OUT2 pada modul L298N	Pin – pada DC Fan 12V
Kabel + pada Adaptor 12V	Pin +12V pada L298N

Awalan	Sambungan
Kabel – pada Adaptor 12V	Pin GND pada modul L298N
Pin D33 pada ESP-32	Pin DATA pada sensor DHT11
Port +5V	Pin VCC pada sensor DHT11
Port GND	Pin GND pada sensor DHT11

4.3.3.3 Setting Blynk Cloud kontrol temperatur

Penyetelan input/output pada Blynk Cloud perlu dilakukan sebelum memasuki proses pengkodean sitem. Berikut merupakan tahapan penyetelan pada Blynk Cloud:

1. Masuk pada menu “Templates”, kemudian *datastreams*. Pada *datastreams* klik “+ New Datastream” berupa Virtual Pin. Pada tampilan *Virtual Pin Datastream*, isi “Temperature” pada kolom nama, Pin yaitu V5, *Data Type* yaitu *Integer*, *Units* yaitu Celsius, dan Min,Max output berupa 0,100. Kemudian *Save*.



Gambar 4. 33 Menyiapkan *Virtual Pin* Temperatur

2. Klik kembali “+ New Datastream” berupa *Virtual Pin*. Pada tampilan *Virtual Pin Datastream*, isi “Humidity” pada kolom nama, Pin yaitu V6, *Data Type* yaitu *Integer*, *Units* yaitu *Percentage*, dan Min,Max output berupa 0,100. Kemudian *Save*.

The screenshot shows the 'Virtual Pin Datastream' configuration window. The 'NAME' field is set to 'Humidity', 'ALIAS' is 'Humidity', 'PIN' is 'V6', 'DATA TYPE' is 'Integer', 'UNITS' is 'Percentage, %', 'MIN' is '0', 'MAX' is '100', and 'DEFAULT VALUE' is 'Default Value'. The 'Save' button at the bottom right is highlighted with a red box.

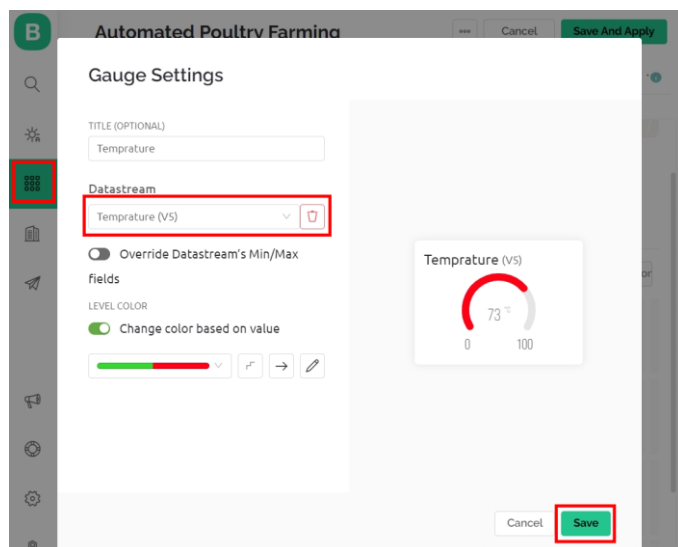
Gambar 4. 34 Menyiapkan *Virtual Pin* Kelembaban

3. Klik kembali “+ New Datastream” berupa *Virtual Pin*. Pada tampilan *Virtual Pin Datastream*, isi “Fan Speed” pada kolom nama, Pin yaitu V7, *Data Type* yaitu *Integer*, *Units* yaitu *Percentage*, dan Min,Max output berupa 0,100. Kemudian *Save*.

The screenshot shows the 'Virtual Pin Datastream' configuration window. The 'NAME' field is set to 'Fan Speed', 'ALIAS' is 'Fan Speed', 'PIN' is 'V7', 'DATA TYPE' is 'Integer', 'UNITS' is 'Percentage, %', 'MIN' is '0', 'MAX' is '100', and 'DEFAULT VALUE' is '0'. The 'Save' button at the bottom right is highlighted with a red box.

Gambar 4. 35 Menyiapkan *Virtual Pin* Kelembaban

4. Selanjutnya masuk pada *web dashboard* dan klik *edit*. Pilih “Gauge” pada *widget* yang disediakan pada menu di sebelah kiri. Beri nama “Temperature” dan isikan *datastream* sesuai dengan yang telah dibuat lalu klik *save*, lakukan kembali langkah ini untuk kelembaban dan kecepatan kipas.



Gambar 4. 36 Menyiapkan *Widget Gauge* Kontrol Temperatur

4.3.3.4 Script kontrol temperatur

Berikut merupakan tabel *script* yang berfungsi sebagai penjelasan dari pengkodean sistem yang dilakukan pada *software* Arduino IDE dan diinput ke mikrokontroler ESP-32:

Tabel 4. 11 *Pseudo Code* Kontrol Temperatur

<i>Pseudo Code</i>	<i>Fungsi</i>
<code>#define BLYNK_TEMPLATE_ID</code> <code>"TMPL5X50ERuS"</code>	Mendefinisikan <i>template id</i> Blynk yang telah didapat saat penyetelan awal.
<code>#define BLYNK_TEMPLATE_NAME</code> <code>"Automated Poultry Farming"</code>	Mendefinisikan <i>name template</i> Blynk yang telah didapat saat penyetelan awal.
<code>#define BLYNK_AUTH_TOKEN</code> <code>"T10irugKyILVfUBSZ5zoq16CBYBHyIFx"</code>	Mendefinisikan token autentikasi yang telah didapat saat penyetelan awal.
<code>#define BLYNK_PRINT Serial</code>	Mendefinisikan <i>serial print</i> Blynk agar dapat dilakukan pemantauan melalui <i>serial monitor</i> .
<code>#include <WiFi.h></code>	Menginput <i>library</i> WiFi agar dapat menyambungkan koneksi WiFi.
<code>#include <BlynkSimpleEsp32.h></code>	Menginput <i>library</i> Blynk ESP32.
<code>char auth[] = BLYNK_AUTH_TOKEN;</code>	Memasukkan token autentikasi yang telah didefinisikan sebelumnya.
<code>char ssid[] = "amang";</code> <code>char pass[] = "123123123";</code>	Memasukkan informasi SSID dan Password koneksi WiFi yang digunakan.

<i>Pseudo Code</i>	<i>Fungsi</i>
BlynkTimer timer;	Berfungsi sebagai perhitungan waktu <i>boot-up</i> selama mikrokontroler terhubung pada server secara <i>default</i> .
#include <LiquidCrystal_I2C.h> LiquidCrystal_I2C lcd(0x27, 16, 2);	Menginput <i>library</i> LCD. Berfungsi sebagai pendefinisian LCD I2C pada <i>library</i> .
#include "DHT.h" #define DHTPIN 33 #define DHTTYPE DHT11 DHT dht(DHTPIN, DHTTYPE);	Menginput <i>library</i> sensor DHT Mendefinisikan Pin sensor DHT Mendefinisikan tipe sensor DHT Memasukkan Pin dan tipe sensor DHT sesuai <i>library</i> .
int temp; int humi; int tempMin = 25; int tempMax = 35;	Membuat variabel untuk pembacaan sensor temperatur dan kelembaban. Menetapkan variabel minimal dan maksimal pada kontrol temperatur.
const int fan = 25; const int IN1_PIN = 27; const int IN2_PIN = 26; int fanSpeed;	Memasukkan ketentuan pin yang terhubung antar mikrokontroler ESP32 dengan modul L298N Mendeklarasikan variabel kontrol data kecepatan kipas.
int fanLCD;	Mendeklarasikan variabel pembacaan kecepatan kipas yang akan ditampilkan pada LCD.
const int fanfreq = 5000; const int fanpwmChannel = 0; const int fanresolution = 8;	Berfungsi sebagai penyetelan sinyal PWM untuk membuat <i>digital pulse</i> agar dapat mengontrol rangkaian secara analog.
void setup() {	Void setup berfungsi sebagai atap penyetelan dari sistem pengkodean yang berisi kode-kode yang hanya satu kali dijalankan saat program baru dimulai.
lcd.init(); lcd.backlight();	Menyalakan sistem LCD.
dht.begin(); pinMode(fan, OUTPUT); pinMode(IN1_PIN, OUTPUT); pinMode(IN2_PIN, OUTPUT);	Menjalankan sensor DHT11 Mendeklarasikan pin-pin yang telah didefinisikan sebelumnya menjadi tipe <i>output</i> pada sistem.

Pseudo Code	Fungsi
<pre> ledcSetup(fanpwmChannel, fanfreq, fanresolution); ledcAttachPin(fan, fanpwmChannel); lcd.setCursor(4,0); lcd.print("Starting"); lcd.setCursor(2,1); lcd.print("Components.."); delay(2500); lcd.clear(); Blynk.begin(auth, ssid, pass); Serial.begin(115200); timer.setInterval(100L, DHT11sensor); timer.setInterval(100L, FanControl); void DHT11sensor() { temp = dht.readTemperature(); humi = dht.readHumidity(); Blynk.virtualWrite(V5, temp); Blynk.virtualWrite(V6, humi); lcd.print("Hum:"); lcd.print(humi); lcd.print("%"); lcd.setCursor(8, 0); lcd.print("Temp:"); lcd.print(temp); lcd.print("C"); void FanControl(){ digitalWrite(IN1_PIN, HIGH); digitalWrite(IN2_PIN, LOW); temp = dht.readTemperature(); if(temp < tempMin){ fanSpeed = 180; ledcWrite(fanpwmChannel, fanSpeed); fanLCD=0; </pre>	<p>Melakukan penyetelan awal ledc berdasarkan ketentuan sinyal PWM.</p> <p>Melakukan penyetelan awal tampilan LCD I2C saat melakukan <i>boot-up</i> sistem.</p> <p>Menjalankan fungsi Blynk agar dapat tersambung pada server. Menjalankan fungsi <i>serial monitor</i>.</p> <p>Menjalankan <i>timer</i> pada sistem DHT dan kipas agar dapat terhubung pada server Blynk.</p> <p>Berfungsi sebagai lingkungan untuk pengkodean sistem DHT sensor. Berfungsi sebagai perintah pembacaan suhu dan kelembaban pada sensor DHT11. Berfungsi sebagai perintah untuk mengirimkan hasil pembacaan data pada sensor DHT11 ke server melalui <i>Virtual Pin</i> yang telah ditentukan. Berfungsi sebagai perintah penulisan hasil pembacaan dari kedua sensor agar dapat ditampilkan pada monitor LCD.</p> <p>Berfungsi sebagai lingkungan untuk pengkodean sistem kontrol kipas. Berfungsi untuk menetapkan aliran listrik pada kedua pin input modul L298N yang digunakan dalam sistem kontrol kipas. Berfungsi sebagai perintah pembacaan suhu pada sensor DHT11. Berfungsi sebagai logika pemrograman pada sistem kontrol kipas, apabila temperatur yang dibaca oleh sensor</p>

<i>Pseudo Code</i>	Fungsi
<pre> } else if((temp >= tempMin) && (temp <= tempMax)){ fanSpeed = map(temp, tempMin, tempMax, 170, 255); fanLCD = map(temp, tempMin, tempMax, 0, 100); ledcWrite(fanpwmChannel, fanSpeed); } lcd.setCursor(0,1); lcd.print("FanSpeed:"); lcd.print(fanLCD); lcd.print("%"); Blynk.virtualWrite(V7, fanLCD); } void loop() { Blynk.run(); timer.run(); </pre>	<p>dibawah ketetapan temperatur minimal, maka kipas tidak akan berputar.</p> <p>Berfungsi apabila fungsi “if” sebelumnya tidak terpenuhi, apabila pembacaan temperatur pada sensor berada di atas ketetapan minimal dan apabila dibawah ketetapan maksimal.</p> <p>Berfungsi sebagai logika pemrograman agar kecepatan kipas dapat diatur berdasarkan suhu yang didapat menggunakan PWM.</p> <p>Berfungsi sebagai logika pemrograman agar kecepatan dapat diukur dalam bentuk persentasi dari 0-100%.</p> <p>Menjalankan seluruh perintah yang telah diolah pada logika pemrograman sebelumnya agar dapat memutar kipas.</p> <p>Berfungsi sebagai perintah penulisan hasil <i>output</i> kecepatan kipas agar dapat ditampilkan pada monitor LCD.</p> <p>Berfungsi sebagai perintah untuk mengirimkan hasil kecepatan pada kipas ke server melalui <i>Virtual Pin</i> yang telah ditentukan.</p> <p>Menjalankan <i>library</i> Blynk serta <i>timer</i> Blynk agar dapat berkomunikasi dengan server.</p>

4.3.4 Pembangunan Sistem Pencahayaan

4.3.4.1 Peralatan sistem pencahayaan

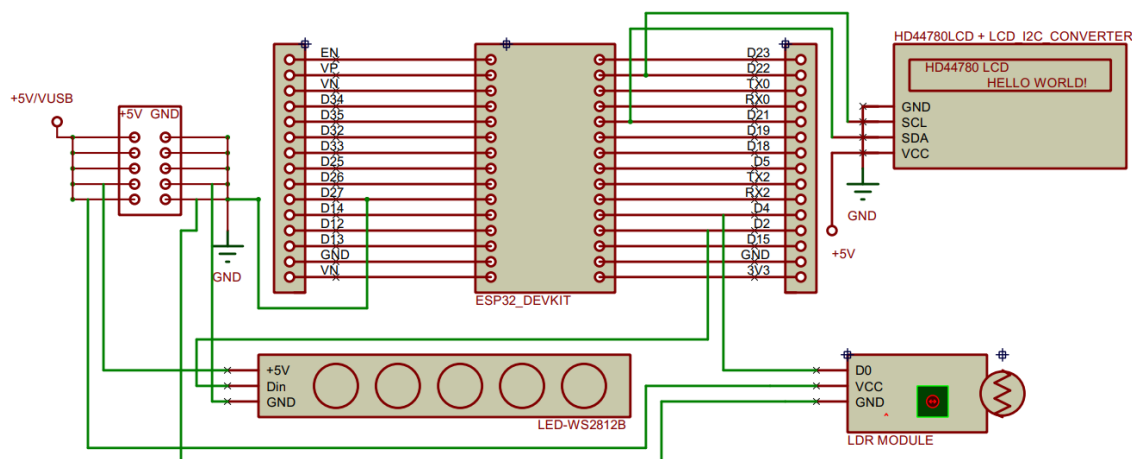
Dalam pembangunan sistem pencahayaan, diperlukan peralatan sebagai berikut:

Tabel 4. 12 Peralatan Sistem Pencahayaan

No	Nama Alat	Banyaknya
1.	Mikrokontroler ESP-32	1
2.	LCD I2C 16x2	1
3.	Modul LDR	1
4.	LED WS2812B	1
5.	Kabel <i>Jumper</i>	8

4.3.4.2 Skematik sistem pencahayaan

Skematik sistem pencahayaan dirancang menggunakan *software* Proteus. *Software* ini berguna dalam mendesain rangkaian elektronik sebelum proses perangkaian sebenarnya dilakukan. Berikut merupakan skematik pada sistem pencahayaan:



Gambar 4. 37 Skematik Sistem Pencahayaan Proteus

Keterangan:

Tabel 4. 13 Keterangan Skematik Sistem Pencahayaan

Awalan	Sambungan
Pin D22 pada ESP-32	Pin SCL pada LCD
Pin D21 pada ESP-32	Pin SDA pada LCD
Pin D4 pada ESP-32	Pin D0 pada modul LDR
Pin D2 pada ESP-32	Pin Din pada LED WS2812B
Port +5V	Pin VCC pada modul LDR
Port +5V	Pin VCC pada LED WS2812B
Port GND	Pin GND pada Modul LDR
Port GND	Pin GND pada LED WS2812B

4.3.4.3 Setting Blynk Cloud sistem pencahayaan

Sebelum masuk ke proses pengkodean sistem, perlu dilakukan penyetelan input/output pada Blynk Cloud. Berikut merupakan tahapan penyetelannya:

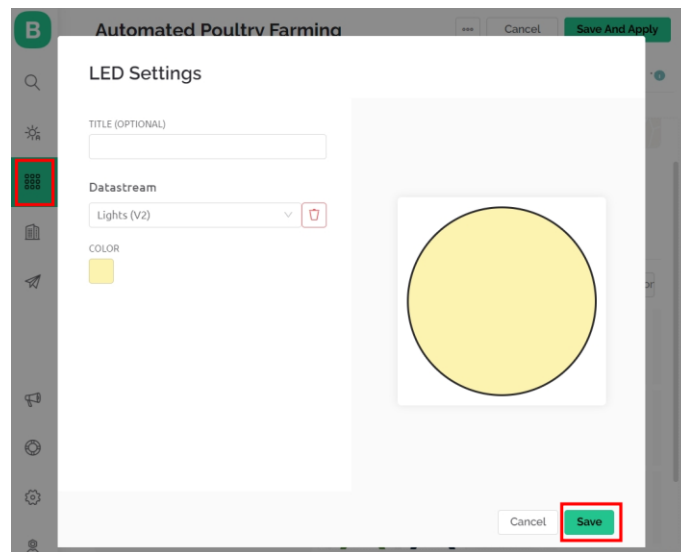
1. Masuk pada menu “Templates”, kemudian *datastreams*. Pada *datastreams* klik “+ New Datastream” berupa Virtual Pin. Pada tampilan *Virtual Pin Datastream*, isi “Lights” pada kolom nama, Pin yaitu V2, *Data Type* yaitu *Integer*, *Units* yaitu *None*, dan Min, Max output berupa 0,1. Kemudian *Save*.

Gambar 4. 38 Menyiapkan *Virtual Pin* Pencahayaan

2. Selanjutnya masuk pada *web dashboard* dan klik *edit*. Pilih “Switch” pada *widget* yang disediakan pada menu di sebelah kiri. Beri nama “Lights”, isikan V2 pada *Datastream*, pada *On Value* isikan 1, pada *Off Value* isikan 0 lalu klik *save*.

Gambar 4. 39 Menyiapkan Widget Switch Pencahayaan

3. Pada menu yang sama , tambahkan *widget* yaitu “Display LED”. Masukkan V2 sebagai *Datastream* yang digunakan lalu klik *save*.



Gambar 4. 40 Menyiapkan Widget Display LED

4.3.4.4 Script sistem pencahayaan

Berikut merupakan tabel *script* yang berfungsi sebagai penjelasan dari pengkodean sistem yang dilakukan pada *software* Arduino IDE dan diinput ke mikrokontroler ESP-32:

Tabel 4. 14 *Pseudo Code* Sistem Pencahayaan

<i>Pseudo Code</i>	Fungsi
#define BLYNK_TEMPLATE_ID "TMPL5X50ERuS"	Mendefinisikan <i>template id</i> Blynk yang telah didapat saat penyetelan awal.
#define BLYNK_TEMPLATE_NAME "Automated Poultry Farming"	Mendefinisikan <i>name template</i> Blynk yang telah didapat saat penyetelan awal.
#define BLYNK_AUTH_TOKEN "T10irugKyILVfUBSZ5zoq16CBYBHxIFx"	Mendefinisikan token autentikasi yang telah didapat saat penyetelan awal.
#define BLYNK_PRINT Serial	Mendefinisikan <i>serial print</i> Blynk agar dapat dilakukan pemantauan melalui <i>serial monitor</i> .
#include <WiFi.h>	Menginput <i>library</i> WiFi agar dapat menyambungkan koneksi WiFi.
#include <BlynkSimpleEsp32.h>	Menginput <i>library</i> Blynk ESP32.
char auth[] = BLYNK_AUTH_TOKEN;	Memasukkan token autentikasi yang telah didefinisikan sebelumnya.
char ssid[] = "amang"; char pass[] = "123123123";	Memasukkan informasi SSID dan Password koneksi WiFi yang digunakan.

<i>Pseudo Code</i>	<i>Fungsi</i>
BlynkTimer timer;	Berfungsi sebagai perhitungan waktu <i>boot-up</i> selama mikrokontroler terhubung pada server secara <i>default</i> .
#include <LiquidCrystal_I2C.h> LiquidCrystal_I2C lcd(0x27, 16, 2);	Menginput <i>library</i> LCD. Berfungsi sebagai pendefinisian LCD I2C pada <i>library</i> .
#include <FastLED.h> #define LED_TYPE WS2812B #define COLOR_ORDER GRB #define NUM_LEDS 15 #define DATA_PIN 2 #define BRIGHTNESS 200 CRGB leds[NUM_LEDS]; int LDRpin = 4; WidgetLED led1(V2); void setup() { lcd.init(); lcd.backlight(); pinMode (LDRpin, INPUT); FastLED.addLeds<LED_TYPE, DATA_PIN, COLOR_ORDER>(leds, NUM_LEDS); lcd.setCursor(4,0); lcd.print("Starting"); lcd.setCursor(2,1); lcd.print("Components.."); delay(2500); lcd.clear();	Menginput <i>library</i> lampu LED. Mendefinisikan tipe LED. Mendefinisikan mode warna yang digunakan yaitu <i>Green</i> , <i>Red</i> , dan <i>Blue</i> . Mendefinisikan total lampu LED yang digunakan, yaitu 15. Mendefinisikan pin data yang terhubung dengan mikrokontroler ESP32. Mendefinisikan aliran listrik yang digunakan dalam mengatur tingkat keterangan pencahayaan yang digunakan. Menginput banyaknya lampu ke dalam <i>library</i> . Berfungsi sebagai variabel yang terhubung dengan <i>Digital Output</i> modul LDR. Berfungsi sebagai <i>virtual pin</i> yang menjadi indikator pada server Blynk apabila lampu menyala. Void setup berfungsi sebagai atap penyetelan dari sistem pengkodean yang berisi kode-kode yang hanya satu kali dijalankan saat program baru dimulai. Menyalakan Sistem LCD. Mendeklarasikan Pin LDR sebagai input. Berfungsi sebagai penyetelan lampu LED berdasarkan <i>library</i> FastLED yang digunakan. Melakukan penyetelan awal tampilan LCD I2C saat melakukan <i>boot-up</i> sistem.

<i>Pseudo Code</i>	Fungsi
Blynk.begin(auth, ssid, pass);	Menjalankan fungsi Blynk agar dapat tersambung pada server.
Serial.begin(115200);	Menjalankan fungsi <i>serial monitor</i> .
timer.setInterval(100L, LightControl); }	Menjalankan <i>timer</i> pada sistem pencahayaan.
void LightControl(){	Berfungsi sebagai lingkungan untuk pengkodean sistem pencahayaan.
Serial.println(digitalRead(LDRpin));	Berfungsi sebagai alat monitor <i>digital output</i> yang dihasilkan modul LDR.
if(digitalRead(LDRpin)==0){	Apabila <i>digital output</i> menghasilkan angka 0 maka LED tidak menyala.
fill_solid(&(leds[0]), 15, CRGB(0, 0, 0));	Tampilan pada monitor LCD, sebagai indikator apabila lampu LED menyala atau mati.
lcd.setCursor(13,1);	
lcd.print("L:o");	
FastLED.show();	Berfungsi sebagai <i>trigger</i> untuk mematikan dan menyalakan lampu.
led1.off(); }	Memiliki fungsi dalam menyalakan dan mematikan <i>widget LED</i> pada server Blynk.
else{	Berfungsi apabila fungsi “if” sebelumnya tidak terpenuhi, apabila <i>digital output</i> menghasilkan angka 1, maka lampu akan menyala terang dengan warna putih.
fill_solid(&(leds[0]), 15, CRGB(255, 255, 255));	Tampilan pada monitor LCD, sebagai indikator apabila lampu LED menyala atau mati.
lcd.setCursor(13,1);	
lcd.print("L:*");	
FastLED.show();	Berfungsi sebagai <i>trigger</i> untuk mematikan dan menyalakan lampu.
led1.on(); }}	Memiliki fungsi dalam menyalakan dan mematikan <i>widget LED</i> pada server Blynk.
void loop() {	Menjalankan <i>library</i> Blynk serta <i>timer</i>
Blynk.run();	Blynk agar dapat berkomunikasi dengan server.
timer.run();	

4.3.5 Pembangunan Sensor Penyimpanan Pakan dan Minum

4.3.5.1 Peralatan sensor penyimpanan

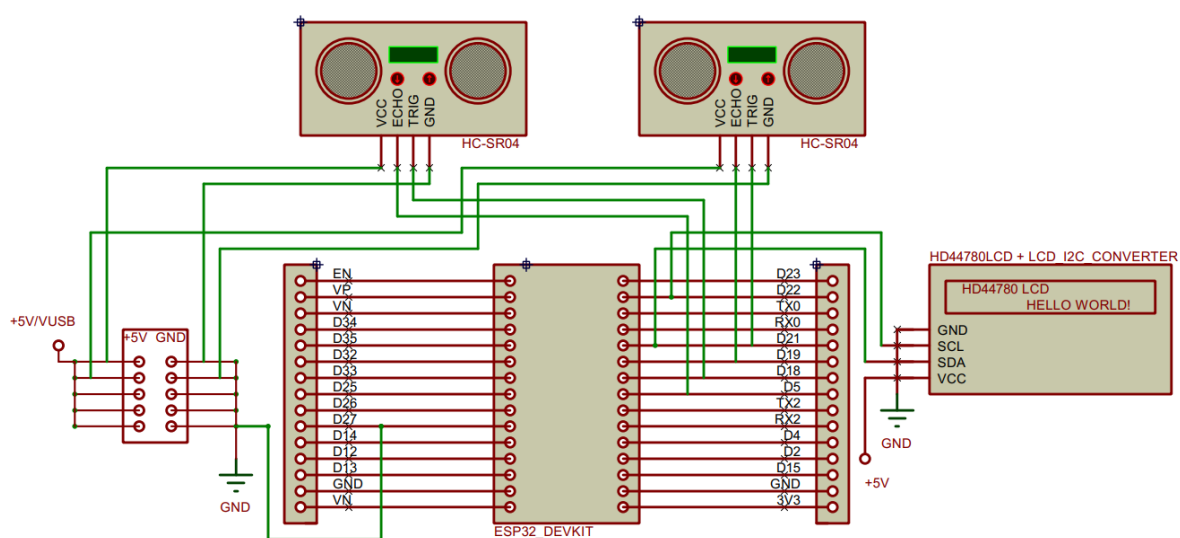
Pada pembangunan sistem pencahayaan, diperlukan peralatan sebagai berikut:

Tabel 4. 15 Peralatan Sensor Penyimpanan

No	Nama Alat	Banyaknya
1.	Mikrokontroler ESP-32	1
2.	LCD I2C 16x2	1
3.	Sensor HC-SR04	1
5.	Kabel <i>Jumper</i>	10

4.3.5.2 Skematik sensor penyimpanan

Skematik sensor penyimpanan dirancang menggunakan *software* Proteus. *Software* ini berguna dalam mendesain rangkaian elektronik sebelum proses perangkaian sebenarnya dilakukan. Berikut merupakan skematik pada sensor penyimpanan:



Gambar 4. 41 Skematik Sensor Penyimpanan Proteus

Keterangan:

Tabel 4. 16 Keterangan Skematik Sensor Penyimpanan

Awalan	Sambungan
Pin D22 pada ESP-32	Pin SCL pada LCD

Awalan	Sambungan
Pin D21 pada ESP-32	Pin SDA pada LCD
Pin D4 pada ESP-32	Pin D0 pada modul LDR
Pin D2 pada ESP-32	Pin Din pada LED WS2812B
Pin D5 pada ESP-32	Pin ECHO pada HC-SR04 pertama
Pin D19 pada ESP-32	Pin ECHO pada HC-SR04 kedua
Pin D18 pada ESP-32	Pin TRIG pada HC-SR04 pertama
Pin D21 pada ESP-32	Pin TRIG pada HC-SR04 pertama
Port +5V	Pin VCC pada sensor HC-SR04 pertama
Port +5V	Pin VCC pada sensor HC-SR04 kedua
Port GND	Pin GND pada sensor HC-SR04 pertama
Port GND	Pin GND pada sensor HC-SR04 kedua

4.3.5.3 Setting Blynk Cloud sensor penyimpanan

Sebelum memasuki proses penerapan kode pada sistem, perlu dilakukan penyetelan input/output pada Blynk Cloud. Berikut merupakan tahapan penyetelannya:

1. Masuk pada menu “Templates”, kemudian *datastreams*. Pada *datastreams* klik “+ New Datastream” berupa Virtual Pin. Pada tampilan *Virtual Pin Datastream*, isi “Food Storage” pada kolom nama, Pin yaitu V3, *Data Type* yaitu *Integer*, *Units* yaitu *Percentage*, dan Min,Max output berupa 0,100. Kemudian *save*.

Gambar 4. 42 Menyiapkan *Virtual Pin* Pakan

2. Klik kembali “+ New Datastream” berupa *Virtual Pin*. Pada tampilan *Virtual Pin Datastream*, isi “Water Storage” pada kolom nama, Pin yaitu V4, *Data Type* yaitu *Integer*, *Units* yaitu *Percentage*, dan Min,Max output berupa 0,100. Kemudian *save*.

Gambar 4. 43 Menyiapkan Virtual Pin Minum

3. Selanjutnya masuk pada *web dashboard* dan klik *edit*. Pilih “Gauge” pada *widget* yang disediakan pada menu di sebelah kiri. Beri nama “Food Storage” dan isikan *datastream* sesuai dengan yang telah dibuat lalu klik *save*, lakukan kembali langkah ini untuk “Water Storage”.

Gambar 4. 44 Menyiapkan Widget Gauge Penyimpanan

4.3.5.4 Script sensor penyimpanan

Berikut merupakan tabel *script* yang berfungsi sebagai penjelasan dari pengkodean sistem yang dilakukan pada *software* Arduino IDE dan diinput ke dalam mikrokontroler ESP-32:

Tabel 4. 17 *Pseudo Code* Sensor Penyimpanan

<i>Pseudo Code</i>	<i>Fungsi</i>
<pre>#define BLYNK_TEMPLATE_ID "TMPL5X50ERuS" #define BLYNK_TEMPLATE_NAME "Automated Poultry Farming" #define BLYNK_AUTH_TOKEN "T10irugKyILVfUBSZ5zoq16CBYBHyIFx" #define BLYNK_PRINT Serial #include <WiFi.h> #include <BlynkSimpleEsp32.h> char auth[] = BLYNK_AUTH_TOKEN; char ssid[] = "amang"; char pass[] = "123123123"; BlynkTimer timer; #include <LiquidCrystal_I2C.h> LiquidCrystal_I2C lcd(0x27, 16, 2); #define wecho 5 #define wtrig 18 int wdepth = 10; #define fecho 19</pre>	<p>Mendefinisikan <i>template id</i> Blynk yang telah didapat saat penyetelan awal.</p> <p>Mendefinisikan <i>name template</i> Blynk yang telah didapat saat penyetelan awal.</p> <p>Mendefinisikan token autentikasi yang telah didapat saat penyetelan awal.</p> <p>Mendefinisikan <i>serial print</i> Blynk agar dapat dilakukan pemantauan melalui <i>serial monitor</i>.</p> <p>Menginput <i>library</i> WiFi agar dapat menyambungkan koneksi WiFi.</p> <p>Menginput <i>library</i> Blynk ESP32.</p> <p>Memasukkan token autentikasi yang telah didefinisikan sebelumnya.</p> <p>Memasukkan informasi SSID dan <i>Password</i> koneksi WiFi yang digunakan.</p> <p>Berfungsi sebagai perhitungan waktu <i>boot-up</i> selama mikrokontroler terhubung pada server secara <i>default</i>.</p> <p>Menginput <i>library</i> LCD.</p> <p>Berfungsi sebagai pendefinisian LCD I2C pada <i>library</i>.</p> <p>Mendefinisikan pin echo penyimpanan minum yang terhubung dengan mikrokontroler ESP32.</p> <p>Mendefinisikan pin trig penyimpanan minum yang terhubung dengan mikrokontroler ESP32.</p> <p>Mendefinisikan variabel kedalaman tangki penyimpanan minum sebesar 10 cm.</p> <p>Mendefinisikan pin echo penyimpanan pakan yang terhubung dengan mikrokontroler ESP32.</p>

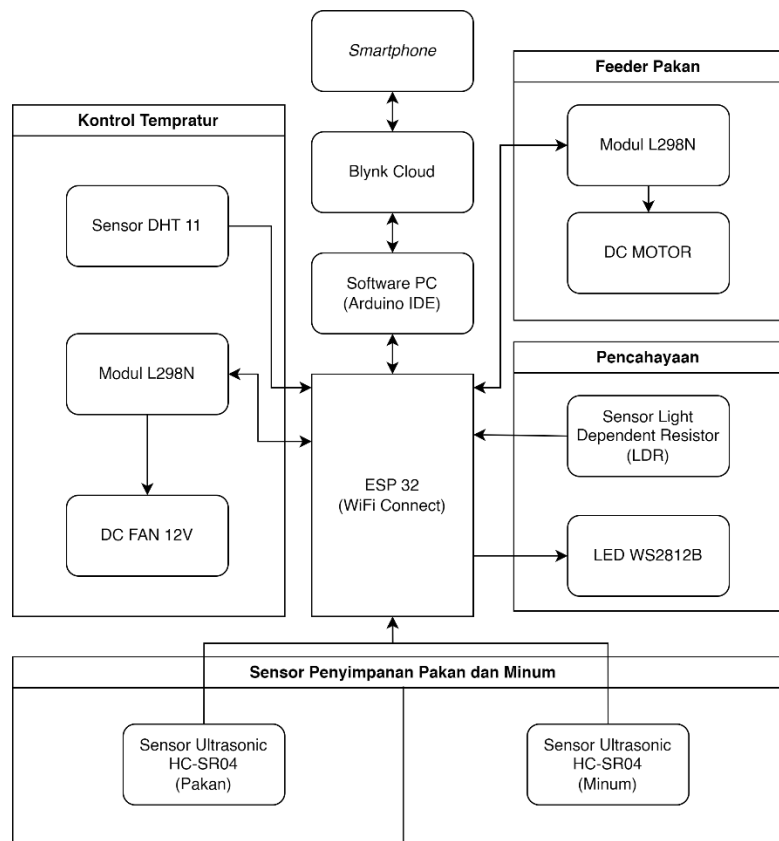
Pseudo Code	Fungsi
#define ftrig 21	Mendefinisikan pin trig penyimpanan pakan yang terhubung dengan mikrokontroler ESP32.
int fdepth = 10;	Mendefinisikan variabel kedalaman tangki penyimpanan pakan sebesar 10 cm.
void setup() {	Void setup berfungsi sebagai atap penyetelan dari sistem pengkodean yang berisi kode-kode yang hanya satu kali dijalankan saat program baru dimulai.
lcd.init();	Menyalakan sistem LCD.
lcd.backlight();	
pinMode(wtrig, OUTPUT);	
pinMode(wecho, INPUT);	Mendeklarasikan pin-pin yang telah didefinisikan sebelumnya menjadi input/output pada sistem penyimpanan.
pinMode(ftrig, OUTPUT);	
pinMode(fecho, INPUT);	
Blynk.begin(auth, ssid, pass);	Menjalankan fungsi Blynk agar dapat tersambung pada server.
Serial.begin(115200);	Menjalankan fungsi <i>serial monitor</i> .
timer.setInterval(100L, WaterLevel);	
timer.setInterval(100L, FoodLevel);	Menjalankan <i>timer</i> pada sistem penyimpanan agar dapat terhubung pada server Blynk.
void WaterLevel(){	Berfungsi sebagai lingkungan untuk pengkodean sistem penyimpanan minum.
digitalWrite(wtrig, LOW);	Berfungsi sebagai sistem yang mengaktifkan dan mematikan pancaran agar dapat dipantulkan pada pin echo.
delayMicroseconds(2);	
digitalWrite(wtrig, HIGH);	
delayMicroseconds(10);	
digitalWrite(wtrig, LOW);	
long t = pulseIn(wecho, HIGH)	Berfungsi sebagai penerima pancaran yang telah dipantulkan.
long cm = t / 29 / 2;	Berfungsi sebagai konversi angka hasil penerima pancaran menjadi satuan sentimeter.
Serial.println(cm);	Berfungsi sebagai tampilan untuk memonitor pada <i>serial monitor</i> .
long wlevel= wdepth-cm;	Berfungsi sebagai rumus penghitung sistem penyimpanan berdasarkan kedalaman.
if (wlevel<0)	Berfungsi sebagai logika pemrograman yang menjadi <i>output</i> pembacaan sensor dari satuan sentimeter menjadi persentase.
wlevel=0;	
wlevel = map(wlevel,0,wdepth,0,100);	

Pseudo Code	Fungsi
Blynk.virtualWrite(V4, wlevel); }	Berfungsi sebagai perintah untuk mengirimkan hasil perhitungan persentase ke server melalui <i>Virtual Pin</i> yang telah ditentukan
<pre> void FoodLevel(){ digitalWrite(ftrig, LOW); delayMicroseconds(2); digitalWrite(ftrig, HIGH); delayMicroseconds(10); digitalWrite(ftrig, LOW); long t = pulseIn(fecho, HIGH) long cm = t / 29 / 2; Serial.println(cm); long flevel= fdepth-cm; if (flevel<0) flevel=0; flevel = map(flevel,0,fdepth,0,100); Blynk.virtualWrite(V3, flevel); } </pre>	<p>Berfungsi sebagai lingkungan untuk pengkodean sistem penyimpanan pakan.</p> <p>Berfungsi sebagai sistem yang mengaktifkan dan mematikan pancaran agar dapat dipantulkan pada pin echo.</p> <p>Berfungsi sebagai penerima pancaran yang telah dipantulkan.</p> <p>Berfungsi sebagai konversi angka hasil penerima pancaran menjadi satuan sentimeter.</p> <p>Berfungsi sebagai tampilan untuk memonitor pada <i>serial monitor</i>.</p> <p>Berfungsi sebagai rumus penghitung sistem penyimpanan berdasarkan kedalaman.</p> <p>Berfungsi sebagai logika pemrograman yang menjadi <i>output</i> pembacaan sensor dari satuan sentimeter menjadi persentase.</p> <p>Berfungsi sebagai perintah untuk mengirimkan hasil perhitungan persentase ke server melalui <i>Virtual Pin</i> yang telah ditentukan</p>
<pre> void loop() { Blynk.run(); timer.run(); </pre>	Menjalankan <i>library</i> Blynk serta <i>timer</i> Blynk agar dapat berkomunikasi dengan server.

4.4 Evaluation and Refining

4.4.1 Evaluasi *Prototype*

Sistem yang telah dibuat dalam bentuk *prototype* dipresentasikan pada hari Sabtu, 15 April 2023 kepada *user* penelitian, yaitu pemilik peternakan ayam di Jalan Pandanaran. Berdasarkan presentasi yang dilakukan peneliti, terdapat komentar dari sudut pandang *user* bahwa diperlukannya kontrol penuh untuk memonitor serta mengontrol sistem automasi pengelolaan ternak ayam yang menggunakan *smartphone* serta tambahan notifikasi sebagai pengingat apabila stok pakan dan air minum mulai menipis. Berdasarkan hal tersebut, berikut merupakan gambaran diagram blok yang telah diperbarui:



Gambar 4. 45 Diagram Blok yang Diperbarui

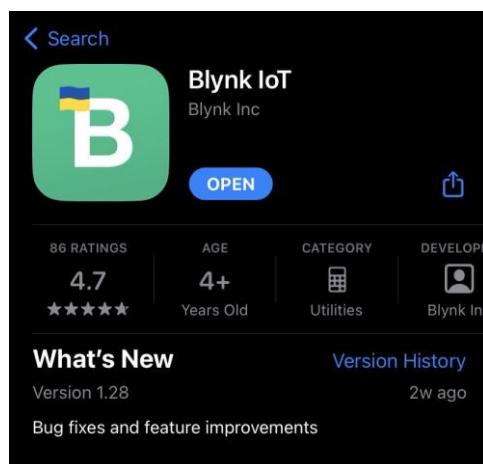
4.4.2 Perbaikan *Prototype*

Berdasarkan evaluasi yang telah dilaksanakan, terdapat beberapa hasil diskusi antara *user* dengan *developer* yang memerlukan perbaikan. Berikut beberapa hal yang membutuhkan perbaikan berdasarkan hasil evaluasi yaitu, pemanfaatan *smartphone* untuk memonitor dan mengontrol sistem pengelolaan serta tambahan notifikasi sebagai pengingat. Maka dari itu, di bawah ini tertera tahapan perbaikan *prototype*:

4.4.2.1 *Monitoring dan controlling menggunakan smartphone*

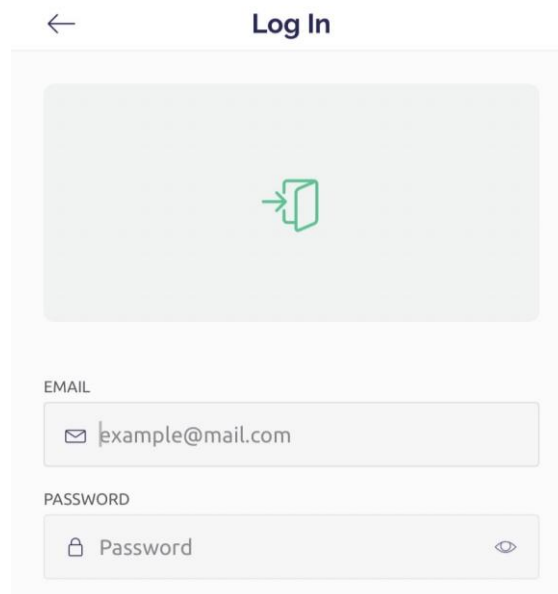
Laman web dapat memonitor serta mengontrol sistem, namun demikian tetap memerlukan alternatif lain dalam melakukan *controlling* serta *monitoring*, salah satunya dengan menggunakan *smartphone* atau telepon genggam. Berikut merupakan tahapan dalam melakukan hal tersebut:

1. Unduh atau *download* aplikasi Blynk IoT pada *appstore* atau *playstore* pada *smartphone*.



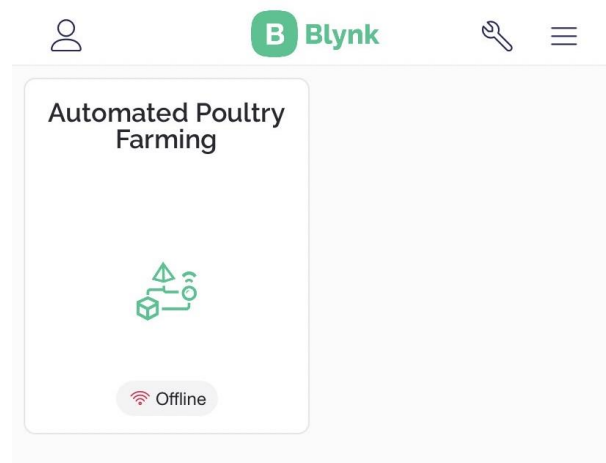
Gambar 4. 46 Unduh Blynk IoT

2. Lakukan *login* dengan akun yang sama dengan Blynk Web.



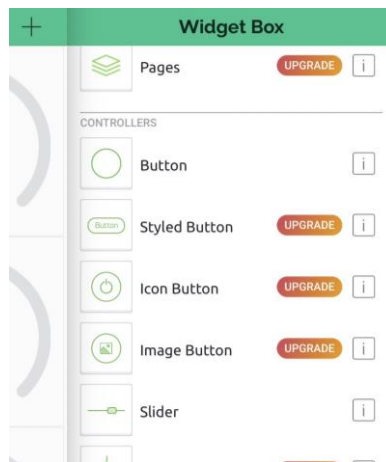
Gambar 4. 47 Login Akun Blynk *Smartphone*

3. Pada halaman utama akan muncul perangkat yang sama dengan yang telah dibuat pada Blynk web. Klik *device* tersebut.



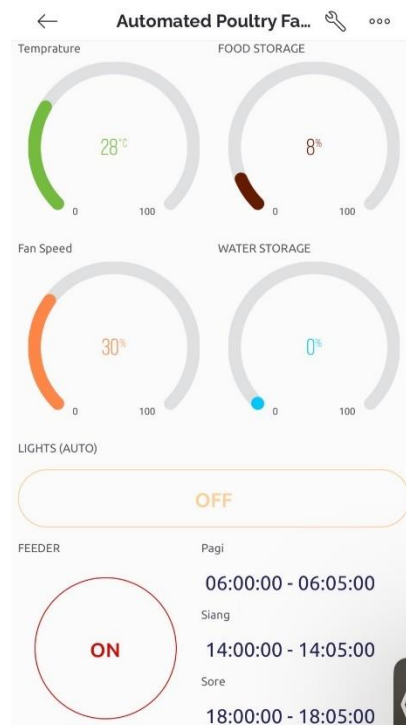
Gambar 4. 48 Halaman Utama Aplikasi Blynk

4. Buka *widget box* menu pada kanan atas dan masukkan *widget* yang sama dengan yang telah dilakukan pada Blynk web.



Gambar 4. 49 Penambahan Widget

5. Atur *layout* tata letak *widget* agar dapat melakukan *monitoring* dan *controlling* dengan mudah.

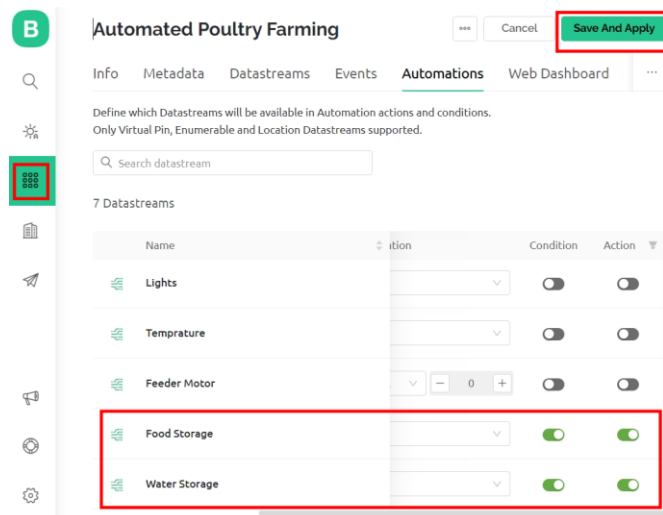


Gambar 4. 50 Tata Letak Widget Blynk IoT

4.4.2.2 Penyetelan notifikasi

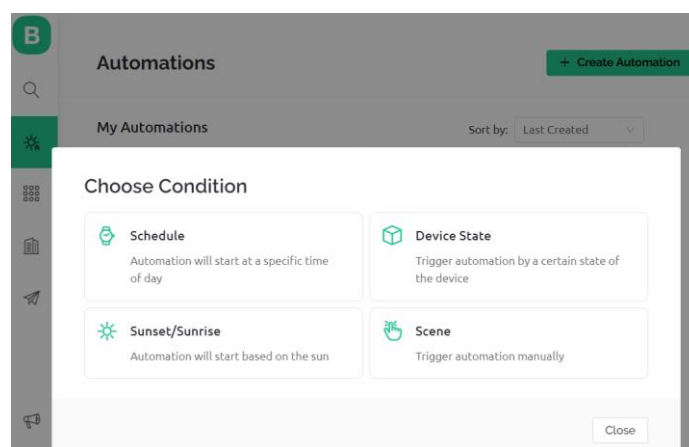
Dalam pembangunan sistem automasi pengelolaan peternakan ayam pada penelitian ini terdapat sensor HC-SR04 yang dimanfaatkan sebagai pengukur penyimpanan pakan dan minum ayam. *Output* yang dihasilkan berupa persentase dari tempat penyimpanan. Berdasarkan hasil evaluasi dengan *user*, diperlukan tambahan notifikasi apabila laporan sensor dari penyimpanan berada di bawah persentase yang ditentukan. Berikut merupakan tahapan perbaikan yang dapat dilakukan:

1. Pada Blynk Web, buka menu “Templates” dilanjutkan dengan membuka *tab* “automations”. Klik *edit*, setelah itu nyalakan *condition* dan *action* pada *datastream* “Food Storage” serta “Water Storage”. Lalu klik *save and apply*.



Gambar 4. 51 Penyetelan *Automations*

2. Selanjutnya buka menu “Automations”, klik “+ Create Automation”, dan klik kembali pada “Device State”.



Gambar 4. 52 *Create Automation*

3. Beri nama sistem automasi “Water Automation”. Selanjutnya pilih *water storage* sebagai *datastream* yang dituju, lalu pilih kondisi “is less than 15%” sebagai batasan indikator yang diinginkan. Kemudian untuk *action* pilih “Send In-App Notifications”.

The screenshot shows the 'Water Automation' configuration page. Under the 'When' section, the datastream is set to 'Automated Poultry Farming' and the condition is 'Is less than 15 %'. The 'Water Storage' datastream is highlighted with a red box. In the 'Do this' section, the 'Send In-App Notifications' action is highlighted with a red box.

Gambar 4. 53 Pengaturan Kondisi Automasi Penyimpanan

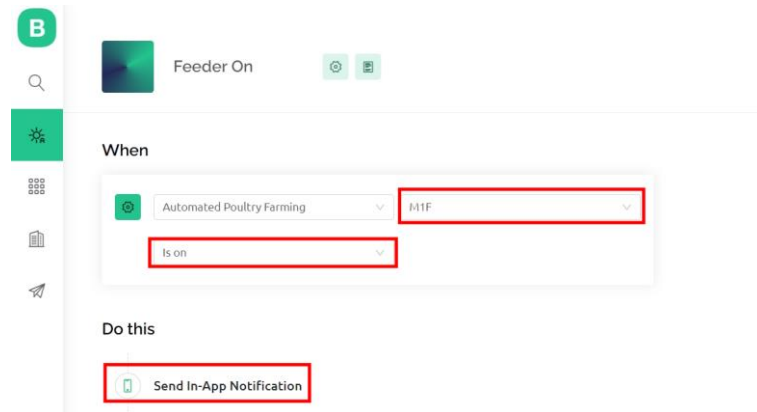
4. Isi subjek dan pesan notifikasi apabila kondisi yang telah diatur terpenuhi, kemudian klik *save*.

The screenshot shows the 'Send In-App Notification' configuration page. The 'RECIPIENTS' field is filled with 'Akmal'. The 'SUBJECT' field is filled with 'Warning!'. The 'MESSAGE' field is filled with 'Water storage is running out.'. The 'Save' button is highlighted in green.

Gambar 4. 54 Penyetelan Pesan Notifikasi Penyimpanan

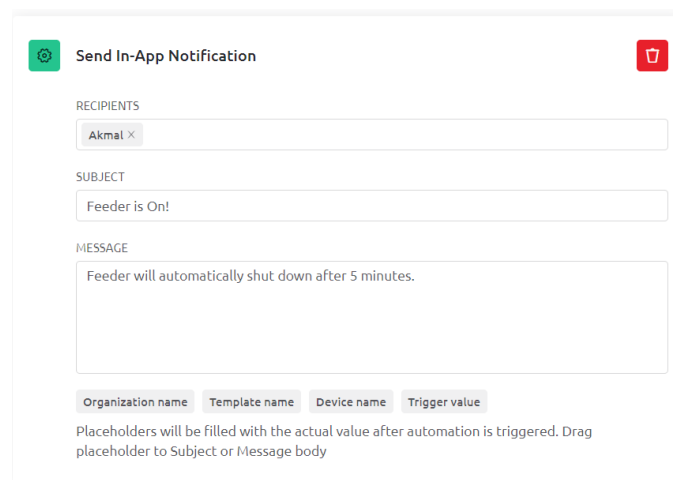
5. Ulangi langkah-langkah di atas untuk penyetelan penyimpanan pakan.

6. Selanjutnya untuk notifikasi pemberian pakan. Beri nama sistem automasi “Feeder On”. Selanjutnya pilih M1F sebagai *datastream* yang dituju, lalu pilih kondisi “is On” sebagai acuan indikator saat *feeder* menyala. Kemudian untuk *action* pilih “Send In-App Notifications”.



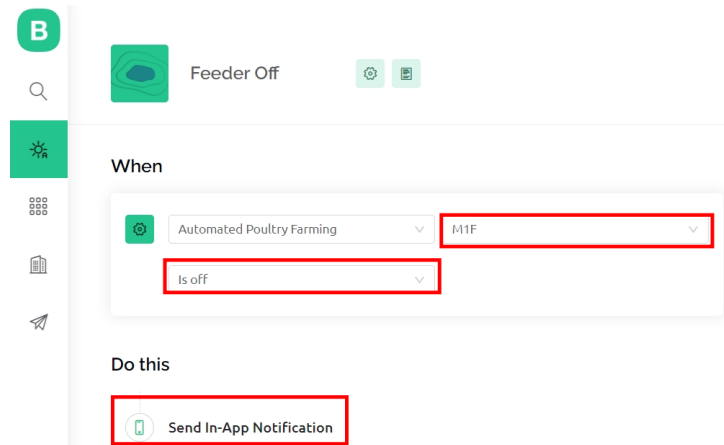
Gambar 4. 55 Pengaturan Kondisi Notifikasi *Feeder* Menyala

7. Isi subjek dan pesan notifikasi apabila kondisi yang telah diatur terpenuhi, kemudian klik *save*.



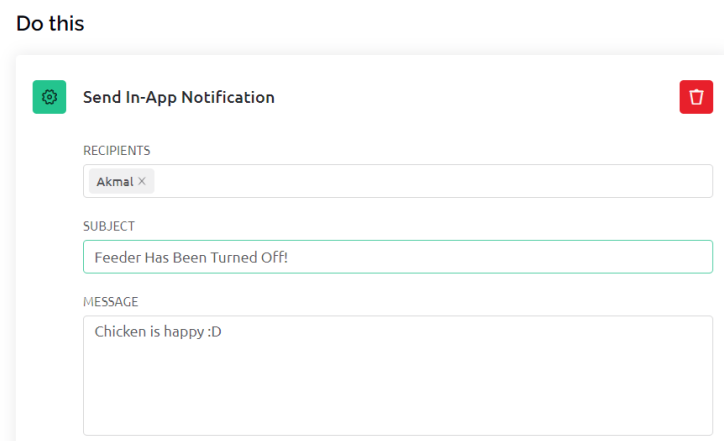
Gambar 4. 56 Penyetelan Pesan Notifikasi *Feeder* Menyala

8. Kemudian untuk notifikasi *feeder off*. Beri nama sistem automasi “Feeder Off”. Selanjutnya pilih M1F sebagai *datastream* yang dituju, lalu pilih kondisi “is Off” sebagai acuan indikator saat *feeder* menyala. Kemudian untuk *action* pilih “Send In-App Notifications”.



Gambar 4. 57 Pengaturan Kondisi Notifikasi *Feeder* Berhenti

9. Isi subjek dan pesan notifikasi apabila kondisi yang telah diatur terpenuhi, kemudian klik *save*.



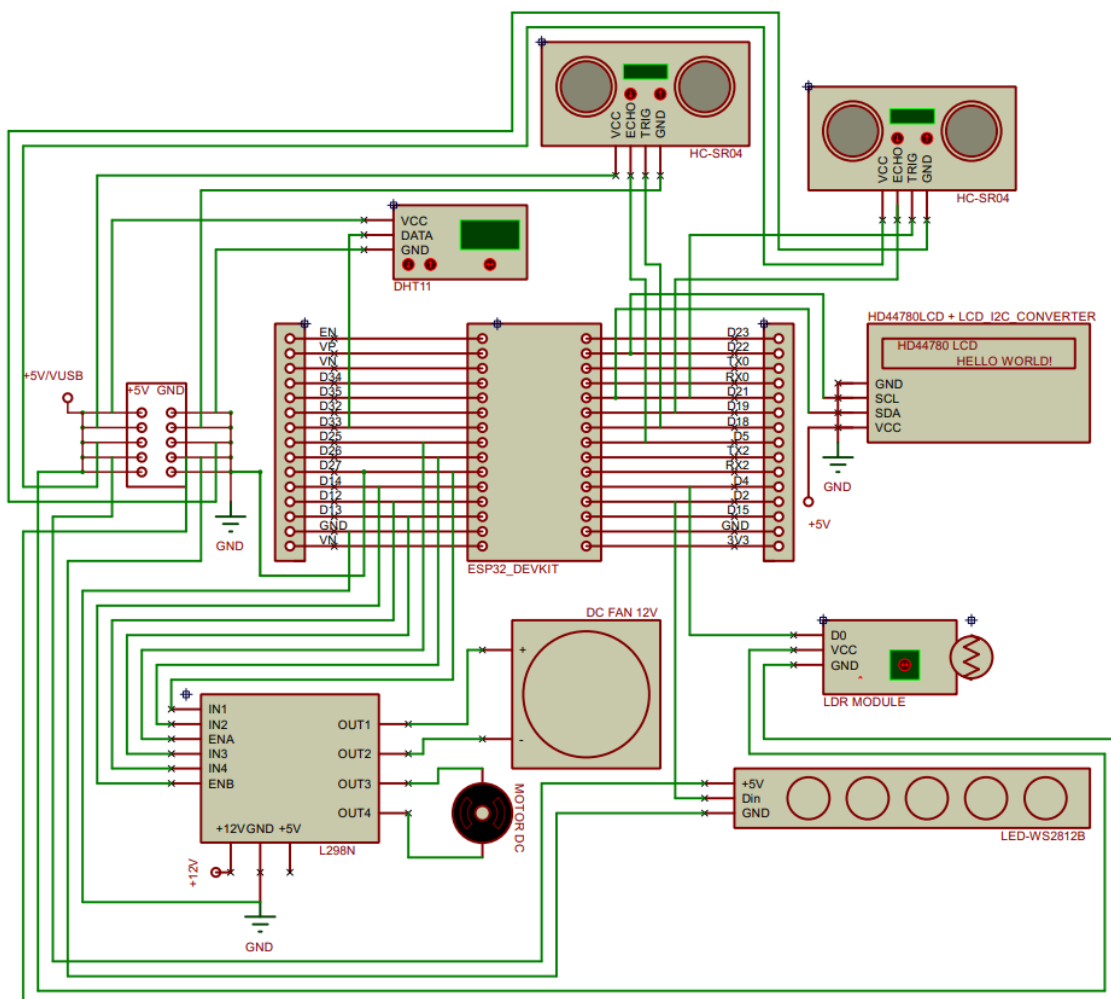
Gambar 4. 58 Penyetelan Pesan Notifikasi *Feeder* Berhenti

4.5 Engineer the Product

Berdasarkan hasil evaluasi pada *prototype*, terdapat beberapa produk yang perlu dilakukan perbaikan. Setelah *prototype* hasil perbaikan telah memenuhi kriteria yang diinginkan oleh pihak *user*, maka *developer* dapat membentuk produk yang sesuai serta memenuhi kebutuhan *user*.

4.5.1 Desain Skematik Keseluruhan

Skematik sistem dirancang secara menyeluruh menggunakan *software* Proteus, desain skematik ini memudahkan pihak pengembang dalam membangun rangkaian elektronik. Berikut merupakan skematik pada sistem automasi pengelolaan peternakan ayam:



Gambar 4. 59 Desain Skematik Keseluruhan

Keterangan skematik:

Tabel 4. 18 Keterangan Skematik Keseluruhan

Awalan	Sambungan
Pin D22 pada ESP-32	Pin SCL pada LCD
Pin D21 pada ESP-32	Pin SDA pada LCD
Pin D13 pada ESP-32	Pin IN3 pada modul L298N
Pin D12 pada ESP-32	Pin IN4 pada modul L298N
Pin D14 pada ESP-32	Pin ENB pada modul L298N
Pin GND pada ESP-32	Pin GND pada modul L298N
Pin OUT3 pada modul L298N	Pin + pada DC Motor
Pin OUT4 pada modul L298N	Pin – pada DC Motor
Kabel + pada Adaptor 12V	Pin +12V pada L298N
Kabel – pada Adaptor 12V	Pin GND pada modul L298N
Pin D27 pada ESP-32	Pin IN1 pada modul L298N
Pin D26 pada ESP-32	Pin IN2 pada modul L298N
Pin D25 pada ESP-32	Pin ENA pada modul L298N
Pin GND pada ESP-32	Pin GND pada modul L298N
Pin OUT1 pada modul L298N	Pin + pada DC Fan 12V
Pin OUT2 pada modul L298N	Pin – pada DC Fan 12V
Kabel + pada Adaptor 12V	Pin +12V pada L298N
Kabel – pada Adaptor 12V	Pin GND pada modul L298N
Pin D33 pada ESP-32	Pin DATA pada sensor DHT11
Port +5V	Pin VCC pada sensor DHT11
Port GND	Pin GND pada sensor DHT11
Pin D4 pada ESP-32	Pin D0 pada modul LDR
Pin D2 pada ESP-32	Pin Din pada LED WS2812B
Port +5V	Pin VCC pada modul LDR
Port +5V	Pin VCC pada LED WS2812B
Port GND	Pin GND pada Modul LDR
Port GND	Pin GND pada LED WS2812B
Pin D2 pada ESP-32	Pin Din pada LED WS2812B

Awalan	Sambungan
Pin D5 pada ESP-32	Pin ECHO pada HC-SR04 pertama
Pin D19 pada ESP-32	Pin ECHO pada HC-SR04 kedua
Pin D18 pada ESP-32	Pin TRIG pada HC-SR04 pertama
Pin D21 pada ESP-32	Pin TRIG pada HC-SR04 pertama
Port +5V	Pin VCC pada sensor HC-SR04 pertama
Port +5V	Pin VCC pada sensor HC-SR04 kedua
Port GND	Pin GND pada sensor HC-SR04 pertama
Port GND	Pin GND pada sensor HC-SR04 kedua

4.5.2 Script Sistem Keseluruhan

Berikut tercantum tabel *script pseudo code* yang merupakan penjelasan dari pengkodean sistem yang dilakukan pada *software* Arduino IDE dan diinput ke mikrokontroler ESP32:

Tabel 4. 19 *Pseudo Code* Sistem Keseluruhan

<i>Pseudo Code</i>	Fungsi
<pre>#define BLYNK_TEMPLATE_ID "TMPL5X50ERuS" #define BLYNK_TEMPLATE_NAME "Automated Poultry Farming" #define BLYNK_AUTH_TOKEN "T10irugKyILVfUBSZ5zoq16CBYBHyIFx" #define BLYNK_PRINT Serial</pre>	<p>Mendefinisikan <i>template id</i> Blynk yang telah didapat saat penyetelan awal.</p> <p>Mendefinisikan <i>name template</i> Blynk yang telah didapat saat penyetelan awal.</p> <p>Mendefinisikan token autentikasi yang telah didapat saat penyetelan awal.</p> <p>Mendefinisikan <i>serial print</i> Blynk agar dapat dilakukan pemantauan melalui <i>serial monitor</i>.</p>
<pre>#include <WiFi.h></pre>	<p>Menginput <i>library</i> WiFi agar dapat menyambungkan koneksi WiFi.</p>
<pre>#include <BlynkSimpleEsp32.h> char auth[] = BLYNK_AUTH_TOKEN;</pre>	<p>Menginput <i>library</i> Blynk ESP32.</p> <p>Memasukkan token autentikasi yang telah didefinisikan sebelumnya.</p>
<pre>char ssid[] = "amang"; char pass[] = "123123123"; BlynkTimer timer;</pre>	<p>Memasukkan informasi SSID dan Password koneksi WiFi yang digunakan.</p> <p>Berfungsi sebagai perhitungan waktu <i>boot-up</i> selama mikrokontroler terhubung pada server secara <i>default</i>.</p>
<pre>#include <LiquidCrystal_I2C.h> LiquidCrystal_I2C lcd(0x27, 16, 2);</pre>	<p>Menginput <i>library</i> LCD.</p> <p>Berfungsi sebagai pendefinisian LCD I2C pada <i>library</i>.</p>

Pseudo Code	Fungsi
#include "DHT.h"	Menginput <i>library</i> sensor DHT
#define DHTPIN 33	Mendefinisikan Pin sensor DHT
#define DHTTYPE DHT11	Mendefinisikan tipe sensor DHT
DHT dht(DHTPIN, DHTTYPE);	Memasukkan Pin dan tipe sensor DHT sesuai <i>library</i> .
int temp;	Membuat variabel untuk pembacaan sensor temperatur dan kelembaban.
int humi;	Menetapkan variabel minimal dan maksimal pada kontrol temperatur.
int tempMin = 25;	
int tempMax = 35;	
const int fanfreq = 5000;	Berfungsi sebagai penyetelan sinyal PWM untuk membuat <i>digital pulse</i> agar dapat mengontrol rangkaian secara analog.
const int fanpwmChannel = 0;	Memasukkan ketentuan pin yang terhubung antar mikrokontroler ESP32 dengan modul L298N
const int fanresolution = 8;	Mendeklarasikan variabel kontrol data kecepatan kipas.
const int fan = 25;	Mendeklarasikan variabel pembacaan kecepatan kipas yang akan ditampilkan pada LCD.
const int IN1_PIN = 27;	
const int IN2_PIN = 26;	
int fanSpeed;	
int fanLCD;	
const int motorfreq = 5000;	Berfungsi sebagai penyetelan sinyal PWM untuk membuat <i>digital pulse</i> agar dapat mengontrol rangkaian secara analog.
const int motorpwmChannel = 2;	Memasukkan ketentuan pin yang terhubung antar mikrokontroler ESP32 dengan modul L298N
const int motorresolution = 8;	
const int M1PWM = 14;	
const int M1F = 13;	
const int M1R = 12;	
#define APP_DEBUG	Mendefinisikan dan menginput <i>library time client</i> untuk menarik waktu <i>real</i> .
#include <NTPClient.h>	
#include <WiFiUdp.h>	
WiFiUDP ntpUDP;	
const long utcOffsetInSeconds = 25200;	Menambahkan <i>offset</i> waktu +7 dari <i>universal time coordinated</i> (UTC) agar menjadi waktu WIB.
NTPClient timeClient(ntpUDP, "asia.pool.ntp.org", utcOffsetInSeconds);	Mengintegrasikan <i>timeclient</i> UTC menjadi WIB.
#define APP_DEBUG	Mendefinisikan dan menginput <i>library time client</i> untuk menarik waktu <i>real</i> .
#include <NTPClient.h>	
#include <WiFiUdp.h>	
WiFiUDP ntpUDP;	

<i>Pseudo Code</i>	<i>Fungsi</i>
<pre> int startTime1; int endTime1; int startTime2; int endTime2; int startTime3; int endTime3; #include <FastLED.h> #define LED_TYPE WS2812B #define COLOR_ORDER GRB #define NUM_LEDS 15 #define DATA_PIN 2 #define BRIGHTNESS 200 CRGB leds[NUM_LEDS]; int LDRpin = 4; WidgetLED led1(V2); #define wecho 5 #define wtrig 18 int wdepth = 7.5; #define fecho 19 #define ftrig 21 int fdepth = 12; </pre>	<p>Mendeklarasikan waktu mulai dan berhenti pemberian pakan yang dibagi menjadi tiga kali pemberian pakan.</p> <p>Menginput <i>library</i> lampu LED.</p> <p>Mendefinisikan tipe LED.</p> <p>Mendefinisikan mode warna yang digunakan yaitu <i>Green</i>, <i>Red</i>, dan <i>Blue</i>.</p> <p>Mendefinisikan total lampu LED yang digunakan, yaitu 15.</p> <p>Mendefinisikan pin data yang terhubung dengan mikrokontroler ESP32.</p> <p>Mendefinisikan aliran listrik yang digunakan dalam mengatur tingkat keterangan pencahayaan yang digunakan.</p> <p>Menginput banyaknya lampu ke dalam <i>library</i>.</p> <p>Berfungsi sebagai variabel yang terhubung dengan <i>Digital Output</i> modul LDR.</p> <p>Berfungsi sebagai <i>virtual pin</i> yang menjadi indikator pada server Blynk apabila lampu menyala.</p> <p>Mendefinisikan pin echo penyimpanan minum yang terhubung dengan mikrokontroler ESP32.</p> <p>Mendefinisikan pin trig penyimpanan minum yang terhubung dengan mikrokontroler ESP32.</p> <p>Mendefinisikan variabel kedalaman tangki penyimpanan minum sebesar 7.5 cm.</p> <p>Mendefinisikan pin echo penyimpanan pakan yang terhubung dengan mikrokontroler ESP32.</p> <p>Mendefinisikan pin trig penyimpanan pakan yang terhubung dengan mikrokontroler ESP32.</p> <p>Mendefinisikan variabel kedalaman tangki penyimpanan pakan sebesar 12 cm.</p>

<i>Pseudo Code</i>	Fungsi
void setup() {	Void setup berfungsi sebagai atap penyetelan dari sistem pengkodean yang berisi kode-kode yang hanya satu kali dijalankan saat program baru dimulai.
lcd.init();	Menyalakan sistem LCD.
lcd.backlight();	
Blynk.begin(auth, ssid, pass);	Menjalankan fungsi Blynk agar dapat tersambung pada server.
dht.begin();	Menjalankan sensor DHT11
pinMode(fan, OUTPUT);	Mendeklarasikan pin-pin yang telah didefinisikan sebelumnya menjadi tipe <i>output</i> pada sistem.
pinMode(IN1_PIN, OUTPUT);	Mendeklarasikan pin-pin yang telah dideklarasikan sebelumnya menjadi tipe <i>output</i> pada sistem.
pinMode(IN2_PIN, OUTPUT);	
pinMode(M1PWM, OUTPUT);	Menyetel kecepatan perputaran DC motor pada 120 PWM atau 6V.
pinMode(M1F, OUTPUT);	Menyatakan kondisi awalan dari DC motor nonaktif.
pinMode(M1R, OUTPUT);	
ledcWrite(motorpwmChannel,120);	
digitalWrite(M1F,LOW);	
digitalWrite(M1R,LOW);	
pinMode (LDRpin, INPUT);	Mendeklarasikan Pin LDR sebagai input.
pinMode(wtrig, OUTPUT);	Mendeklarasikan pin-pin yang telah didefinisikan sebelumnya menjadi input/output pada sistem penyimpanan.
pinMode(wecho, INPUT);	
pinMode(ftrig, OUTPUT);	
pinMode(fecho, INPUT);	
FastLED.addLeds<LED_TYPE, DATA_PIN, COLOR_ORDER>(leds, NUM_LEDS);	Berfungsi sebagai penyetelan lampu LED berdasarkan <i>library</i> FastLED yang digunakan.
ledcSetup(motorpwmChannel, motorfreq, motorresolution);	Melakukan penyetelan awal ledc berdasarkan ketentuan sinyal PWM.
ledcAttachPin(M1PWM, motorpwmChannel);	
ledcSetup(fanpwmChannel, fanfreq, fanresolution);	
ledcAttachPin(fan, fanpwmChannel);	
lcd.setCursor(4,0);	Melakukan penyetelan awal tampilan LCD I2C saat melakukan <i>boot-up</i> sistem.
lcd.print("Starting");	
lcd.setCursor(2,1);	
lcd.print("Components..");	
delay(2500);	
lcd.clear();	
timeClient.begin();	Menjalankan <i>timeclient</i> .
Serial.begin(115200); }	Menjalankan fungsi <i>serial monitor</i> .

<i>Pseudo Code</i>	<i>Fungsi</i>
<pre> timer.setInterval(100L, DHT11sensor); timer.setInterval(100L, FanControl); timer.setInterval(100L, LightControl); } timer.setInterval(100L, WaterLevel); timer.setInterval(100L, FoodLevel); void DHT11sensor() { temp = dht.readTemperature(); humi = dht.readHumidity(); Blynk.virtualWrite(V5, temp); Blynk.virtualWrite(V6, humi); lcd.print("Hum:"); lcd.print(humi); lcd.print("%"); lcd.setCursor(8, 0); lcd.print("Temp:"); lcd.print(temp); lcd.print("C"); void FanControl(){ digitalWrite(IN1_PIN, HIGH); digitalWrite(IN2_PIN, LOW); temp = dht.readTemperature(); if(temp < tempMin){ fanSpeed = 180; ledcWrite(fanpwmChannel, fanSpeed); fanLCD=0; } else if((temp >= tempMin) && (temp <= tempMax)){ fanSpeed = map(temp, tempMin, tempMax, 170, 255); </pre>	<p>Menjalankan <i>timer</i> pada sistem DHT dan kipas agar dapat terhubung pada server Blynk.</p> <p>Menjalankan <i>timer</i> pada sistem pencahayaan.</p> <p>Menjalankan <i>timer</i> pada sistem penyimpanan agar dapat terhubung pada server Blynk.</p> <p>Berfungsi sebagai lingkungan untuk pengkodean sistem DHT sensor.</p> <p>Berfungsi sebagai perintah pembacaan suhu dan kelembaban pada sensor DHT11.</p> <p>Berfungsi sebagai perintah untuk mengirimkan hasil pembacaan data pada sensor DHT11 ke server melalui <i>Virtual Pin</i> yang telah ditentukan.</p> <p>Berfungsi sebagai perintah penulisan hasil pembacaan dari kedua sensor agar dapat ditampilkan pada monitor LCD.</p> <p>Berfungsi sebagai lingkungan untuk pengkodean sistem kontrol kipas.</p> <p>Berfungsi untuk menetapkan aliran listrik pada kedua pin input modul L298N yang digunakan dalam sistem kontrol kipas.</p> <p>Berfungsi sebagai perintah pembacaan suhu pada sensor DHT11.</p> <p>Berfungsi sebagai logika pemrograman pada sistem kontrol kipas, apabila temperatur yang dibaca oleh sensor dibawah ketetapan temperatur minimal, maka kipas tidak akan berputar.</p> <p>Berfungsi apabila fungsi “if” sebelumnya tidak terpenuhi, apabila pembacaan temperatur pada sensor berada di atas ketetapan minimal dan apabila dibawah ketetapan maksimal.</p> <p>Berfungsi sebagai logika pemrograman agar kecepatan kipas dapat diatur berdasarkan suhu yang didapat menggunakan PWM.</p>

<i>Pseudo Code</i>	Fungsi
<pre> fanLCD = map(temp, tempMin, tempMax, 0, 100); ledcWrite(fanpwmChannel, fanSpeed); } lcd.setCursor(0,1); lcd.print("FanSpeed:"); lcd.print(fanLCD); lcd.print("%"); Blynk.virtualWrite(V7, fanLCD); } </pre>	<p>Berfungsi sebagai logika pemrograman agar kecepatan dapat diukur dalam bentuk persentasi dari 0-100%.</p> <p>Menjalankan seluruh perintah yang telah diolah pada logika pemrograman sebelumnya agar dapat memutar kipas.</p> <p>Berfungsi sebagai perintah penulisan hasil <i>output</i> kecepatan kipas agar dapat ditampilkan pada monitor LCD.</p>
<pre> BLYNK_WRITE(V8) { int value = param.asInt(); if(value == 1){ digitalWrite(M1F, HIGH); lcd.setCursor(9,0); lcd.print("Feed:*"); } else{ digitalWrite(M1F, LOW); lcd.setCursor(9,0); lcd.print("Feed:o");} } </pre>	<p>Menjalankan sistem <i>write</i> dari <i>widget feeder</i> yang dimana apabila <i>switch</i> di-on kan akan menyebabkan <i>value</i> menjadi 1 dan <i>feeder</i> akan menyala. Sebaliknya apabila <i>switch</i> mati, maka <i>value</i> menjadi 0 dan <i>feeder</i> akan mati.</p>
<pre> BLYNK_WRITE(V1){ startTime1 = param[0].asInt(); endTime1 = param[1].asInt(); if(startTime1 == 0 && endTime1 == 0) { startTime1 = 999999; endTime1 = 999999; } Serial.println(startTime1); Serial.println(endTime1); } </pre>	<p>Menjalankan sistem <i>write</i> dari <i>time input</i> penjadwalan pakan pagi hari.</p>
<pre> BLYNK_WRITE(V9){ startTime2 = param[0].asInt(); endTime2 = param[1].asInt(); if(startTime2 == 0 && endTime2 == 0) { startTime2 = 999999; endTime2 = 999999; } } </pre>	<p>Menjalankan sistem <i>write</i> dari <i>time input</i> penjadwalan pakan siang hari.</p>

<i>Pseudo Code</i>	<i>Fungsi</i>
<pre> Serial.println(startTime2); Serial.println(endTime2); } BLYNK_WRITE(V6){ startTime3 = param[0].asInt(); endTime3 = param[1].asInt(); if(startTime3 == 0 && endTime3 == 0) { startTime3 = 999999; endTime3 = 999999; } Serial.println(startTime3); Serial.println(endTime3); } </pre>	<p>Menjalankan sistem <i>write</i> dari <i>time input</i> penjadwalan pakan sore hari.</p>
<pre> void LightControl(){ Serial.println(digitalRead(LDRpin)); if(digitalRead(LDRpin)==0){ fill_solid(&(leds[0]), 15, CRGB(0, 0, 0)); lcd.setCursor(13,1); lcd.print("L:o"); FastLED.show(); led1.off(); } else{ fill_solid(&(leds[0]), 15, CRGB(255, 255, 255)); lcd.setCursor(13,1); lcd.print("L:*"); FastLED.show(); led1.on(); }} </pre>	<p>Berfungsi sebagai lingkungan untuk pengkodean sistem pencahayaan.</p> <p>Berfungsi sebagai alat monitor <i>digital output</i> yang dihasilkan modul LDR. Apabila <i>digital output</i> menghasilkan angka 0 maka LED tidak menyala. Tampilan pada monitor LCD, sebagai indikator apabila lampu LED menyala atau mati.</p> <p>Berfungsi sebagai <i>trigger</i> untuk mematikan dan menyalakan lampu. Memiliki fungsi dalam menyalakan dan mematikan <i>widget LED</i> pada server Blynk.</p> <p>Berfungsi apabila fungsi “if” sebelumnya tidak terpenuhi, apabila <i>digital output</i> menghasilkan angka 1, maka lampu akan menyala terang dengan warna putih. Tampilan pada monitor LCD, sebagai indikator apabila lampu LED menyala atau mati.</p> <p>Berfungsi sebagai <i>trigger</i> untuk mematikan dan menyalakan lampu. Memiliki fungsi dalam menyalakan dan mematikan <i>widget LED</i> pada server Blynk.</p>
<pre> void WaterLevel(){ digitalWrite(wtrig, LOW); delayMicroseconds(2); digitalWrite(wtrig, HIGH); </pre>	<p>Berfungsi sebagai lingkungan untuk pengkodean sistem penyimpanan minum. Berfungsi sebagai sistem yang mengaktifkan dan mematikan pancaran agar dapat dipantulkan pada pin echo.</p>

<i>Pseudo Code</i>	<i>Fungsi</i>
<pre> delayMicroseconds(10); digitalWrite(wtrig, LOW); long t = pulseIn(wecho, HIGH) long cm = t / 29 / 2; Serial.println(cm); long wlevel= wdepth-cm; if (wlevel<0) wlevel=0; wlevel = map(wlevel,0,wdepth,0,100); Blynk.virtualWrite(V4, wlevel); } void FoodLevel(){ digitalWrite(ftrig, LOW); delayMicroseconds(2); digitalWrite(ftrig, HIGH); delayMicroseconds(10); digitalWrite(ftrig, LOW); long t = pulseIn(fecho, HIGH) long cm = t / 29 / 2; Serial.println(cm); long flevel= fdepth-cm; if (flevel<0) flevel=0; flevel = map(flevel,0,fdepth,0,100); Blynk.virtualWrite(V3, flevel); }</pre>	<p>Berfungsi sebagai penerima pancaran yang telah dipantulkan.</p> <p>Berfungsi sebagai konversi angka hasil penerima pancaran menjadi satuan sentimeter.</p> <p>Berfungsi sebagai tampilan untuk memonitor pada <i>serial monitor</i>.</p> <p>Berfungsi sebagai rumus penghitung sistem penyimpanan berdasarkan kedalaman.</p> <p>Berfungsi sebagai logika pemrograman yang menjadi <i>output</i> pembacaan sensor dari satuan sentimeter menjadi persentase.</p> <p>Berfungsi sebagai perintah untuk mengirimkan hasil perhitungan persentase ke server melalui <i>Virtual Pin</i> yang telah ditentukan</p> <p>Berfungsi sebagai lingkungan untuk pengkodean sistem penyimpanan pakan.</p> <p>Berfungsi sebagai sistem yang mengaktifkan dan mematikan pancaran agar dapat dipantulkan pada pin echo.</p> <p>Berfungsi sebagai penerima pancaran yang telah dipantulkan.</p> <p>Berfungsi sebagai konversi angka hasil penerima pancaran menjadi satuan sentimeter.</p> <p>Berfungsi sebagai tampilan untuk memonitor pada <i>serial monitor</i>.</p> <p>Berfungsi sebagai rumus penghitung sistem penyimpanan berdasarkan kedalaman.</p> <p>Berfungsi sebagai logika pemrograman yang menjadi <i>output</i> pembacaan sensor dari satuan sentimeter menjadi persentase.</p> <p>Berfungsi sebagai perintah untuk mengirimkan hasil perhitungan persentase ke server melalui <i>Virtual Pin</i> yang telah ditentukan</p>

Pseudo Code	Fungsi
<pre> void loop() { timeClient.update(); int HH = timeClient.getHours(); int MM = timeClient.getMinutes(); int SS = timeClient.getSeconds(); int server_time = 3600*HH + 60*MM + SS; if((startTime1 == server_time) (startTime2 == server_time) (startTime3 == server_time)) { digitalWrite(M1F, HIGH); Blynk.virtualWrite(V8, 1); lcd.setCursor(9,0); lcd.print("Feed:*"); else if ((endTime1 == server_time) (endTime2 == server_time) (endTime3 == server_time)) { digitalWrite(M1F, LOW); Blynk.virtualWrite(V8, 0); lcd.setCursor(9,0); lcd.print("Feed:o") Blynk.run(); timer.run(); </pre>	<p>Menjalankan <i>library</i> Blynk serta <i>timer</i> Blynk agar dapat berkomunikasi dengan server.</p> <p>Menarik data <i>real time</i> dari <i>time client</i> agar sistem dapat membaca situasi berdasarkan waktu yang telah ditetapkan.</p> <p>Memberi kondisi apabila waktu penjadwalan pemberian pakan sesuai dengan waktu yang terbaca oleh <i>server time client</i> maka <i>feeder</i> akan menyala secara otomatis.</p> <p>Dan apabila waktu penjadwalan pemberhentian pakan sesuai dengan waktu yang terbaca oleh <i>server time client</i> maka <i>feeder</i> akan berhenti secara otomatis.</p> <p>Menjalankan <i>library</i> Blynk serta <i>timer</i> Blynk agar dapat berkomunikasi dengan server.</p>

4.5.3 Penetapan Pemberian Pakan

Kebutuhan pemberian pakan perlu disesuaikan dengan usia ayam mengacu pada tabel 4.3 menurut Balai Pengembangan Peternakan (BPP, 2019). Berdasarkan pengujian pada sistem untuk 1 Kg pakan memerlukan waktu aktivasi DC Motor selama 45 detik. Berikut merupakan hasil pengujian sistem untuk dijadikan acuan penetapan *timer*:

Tabel 4. 20 Acuan Timer Pemberian Pakan

Periode	Kebutuhan/100 Ekor	Waktu
Masa pertumbuhan awal (0-3 minggu)	2 Kg	1,5 Menit
Masa pertumbuhan menengah (3-6 minggu)	6 Kg	4,5 Menit
Masa pertumbuhan akhir (6-8 minggu)	12 Kg	9 Menit

Hasil pengujian tersebut dapat diaplikasikan pada sistem *timer* sebagai berikut:

1. Usia ternak 0-3 minggu

Tabel 4. 21 *Timer* Pemberian Pakan 1

Jadwal Pemberian Pakan	Waktu Mulai	Waktu Berhenti
Pagi	06:00:00	06:01:30
Siang	12:00:00	12:01:30
Sore	17:00:00	17:01:30

2. Usia ternak 3-6 minggu

Tabel 4. 22 *Timer* Pemberian Pakan 2

Jadwal Pemberian Pakan	Waktu Mulai	Waktu Berhenti
Pagi	06:00:00	06:04:30
Siang	12:00:00	12:04:30
Sore	17:00:00	17:04:30

3. Usia ternak 6-8 minggu

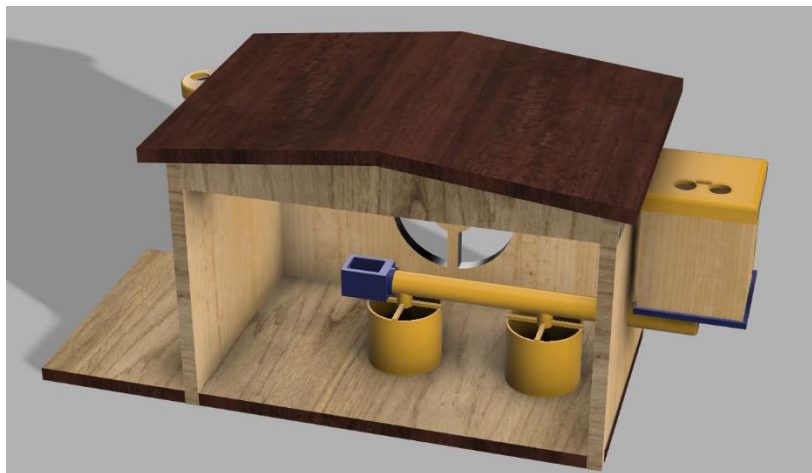
Tabel 4. 23 *Timer* Pemberian Pakan 3

Jadwal Pemberian Pakan	Waktu Mulai	Waktu Berhenti
Pagi	06:00:00	06:09:00
Siang	12:00:00	12:09:00
Sore	17:00:00	17:09:00

4.5.4 Pembangunan Tata Letak Sistem

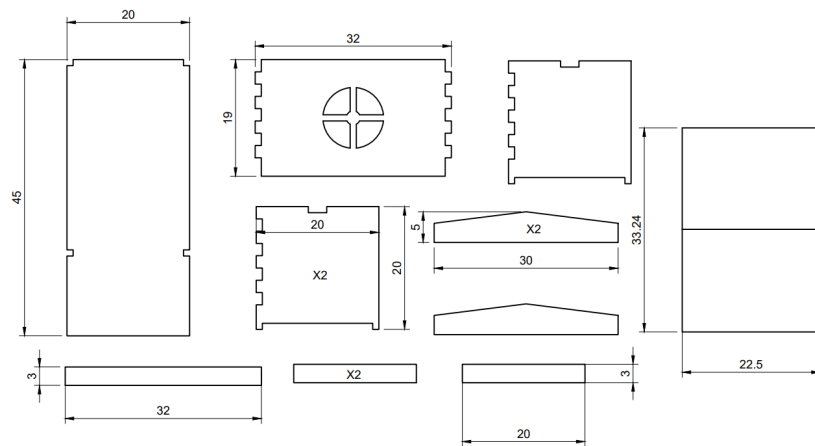
4.5.4.1 Desain bangun ruang prototype

Desain bangun ruang adalah proses merencanakan serta mengatur ruangan pada sebuah lingkungan dengan tujuan terciptanya ruangan yang efisien dan berfungsi dengan baik. Pada penelitian ini, *prototype* automasi pengelolaan ternak ayam didesain menggunakan *software Autodesk Fusion 360*. Berikut ini merupakan hasil desain model 3 dimensi bangun ruang *prototype* peternakan ayam pada penelitian ini:



Gambar 4. 60 Model 3D Bangun Ruang *Prototype* Peternakan Ayam

Setelah dilakukannya pemodelan 3 dimensi, dapat dibuat gambar teknik yang dijadikan acuan dalam proses eksekusi pembangunan secara nyata. Berikut ini merupakan gambar teknik dari pemodelan 3 dimensi *prototype* peternakan ayam pada penelitian ini:



Gambar 4. 61 Gambar Teknik *Prototype* Peternakan Ayam

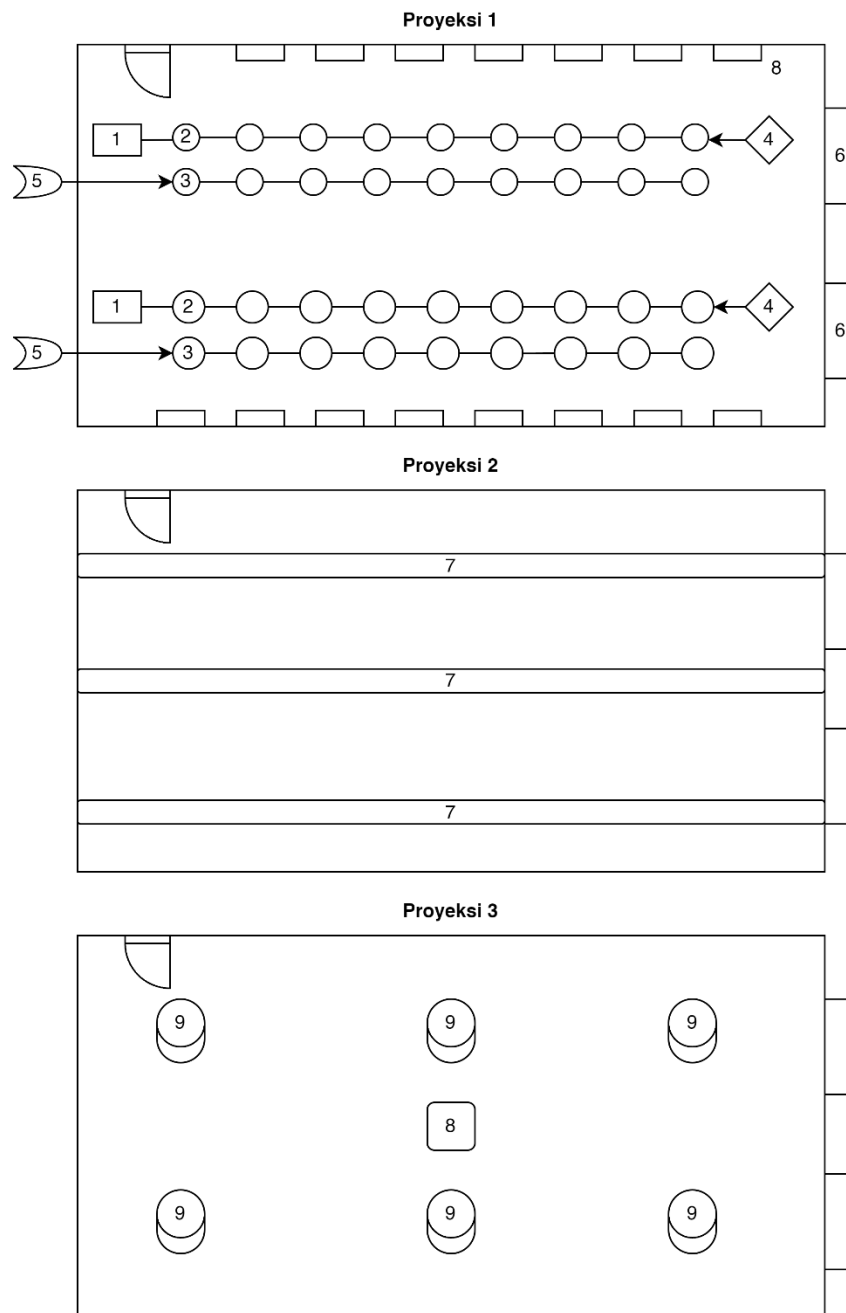
4.5.4.2 Hasil pemasangan *prototype* tata letak sistem pada bangun ruang prototype

Setelah bangun ruang selesai dibangun, maka dapat dilakukan instalasi sistem. Pemasangan instalasi dilakukan secara menyeluruh agar dapat dilakukan pengujian sesuai dengan harapan *user*. Berikut ini merupakan hasil pemasangan *prototype* tata letak sistem pengelolaan peternakan ayam otomatis berbasis mikrokontroler:



Gambar 4. 62 Hasil Instalasi Sistem

4.5.4.3 Proyeksi peletakan sistem pada peternakan



Gambar 4. 63 Proyeksi Peletakan Sistem

Berikut merupakan keterangan dari proyeksi peletakan sistem:

Tabel 4. 24 Keterangan Proyeksi Pelatakan Sistem

Nomor	Keterangan
1	DC Motor
2	<i>Feeder</i>
3	<i>Water Nozzle</i>
4	Penyimpanan Pakan
5	Penyimpanan Air
6	Kipas Sirkulasi
7	Lampu
8	Sensor DHT11
9	Sensor LDR

BAB V

PENGUJIAN DAN PEMBAHASAN SISTEM

5.1 Pengujian Sistem

Pengujian sistem bertujuan untuk mengevaluasi sistem *prototype* secara keseluruhan. Selain itu, pengujian dilaksanakan demi memastikan bahwa sistem dapat berfungsi dengan benar dalam lingkup produksi yang sebenarnya karena melibatkan pengujian fungsionalitas dan non-fungsionalitas pada seluruh sistem. Pelaksanaan pengujian ini dapat mengobservasi jalannya sistem apabila masih terdapat *bug* atau masalah lain yang mungkin belum terdeteksi.

5.1.1 User Acceptance Test

Pada penelitian ini dilaksanakan uji fungsionalitas yang dilakukan dengan *user acceptance testing*. Uji fungsionalitas ini menggunakan parameter yang disusun berdasarkan fungsi pokok yang disesuaikan kebutuhannya oleh *user*. Pengujian ini dilakukan langsung kepada *user* yaitu pemilik peternakan ayam pandanaran.

Tabel 5. 1 UAT-01

<i>User Acceptance Test</i>			
Nama Sistem	Sistem <i>monitoring-controlling feeder</i> pakan serta minum		
Nomor Pengujian	UAT-01		
Tanggal Pengujian	17 Mei 2023		
No.	Fungsi Pokok	Sesuai	
		Ya	Tidak
1.	Memantau status sisa penyimpanan pakan	✓	
2.	Menerima notifikasi apabila penyimpanan di bawah 15%	✓	
3.	Mematikan dan menyalakan kontrol pemberian pakan	✓	
4.	Penjadwalan pemberian pakan	✓	
5.	Mengakses sistem melalui <i>smartphone</i>	✓	
	Jumlah	5	0

Tabel di atas merupakan hasil dari pengujian sistem *monitoring* serta *controlling feeder* pakan dan penyimpanan air minum yang sepenuhnya dapat diakses melalui *Blynk Cloud*. *User*

acceptance test yang telah dilaksanakan berdasarkan kasus pengujian memiliki hasil 100% sesuai dengan harapan *user*.

Tabel 5. 2 UAT-02

<i>User Acceptance Test</i>			
Nama Sistem	Sistem <i>monitoring</i> kontrol temperatur		
Nomor Pengujian	UAT-02		
Tanggal Pengujian	17 Mei 2023		
No.	Fungsi Pokok	Sesuai	
		Ya	Tidak
1.	Memantau suhu dan kelembaban lingkungan	✓	
2.	Memantau kecepatan kipas sebagai sistem sirkulasi udara	✓	
3.	Mengakses sistem melalui <i>smartphone</i>	✓	
	Jumlah	3	0

Tabel di atas merupakan hasil dari pengujian sistem *monitoring* kontrol temperatur yang sepenuhnya dapat diakses melalui *Blynk Cloud*. *User acceptance test* yang telah dilaksanakan berdasarkan kasus pengujian memiliki hasil 100% sesuai dengan harapan *user*.

Tabel 5. 3 UAT-03

<i>User Acceptance Test</i>			
Nama Sistem	Sistem <i>monitoring</i> sistem pencahayaan		
Nomor Pengujian	UAT-03		
Tanggal Pengujian	17 Mei 2023		
No.	Fungsi Pokok	Sesuai	
		Ya	Tidak
1.	Memantau status nyala lampu	✓	
2.	Mematikan dan menyalakan lampu	✓	
3.	Mengakses sistem melalui <i>smartphone</i>	✓	
	Jumlah	3	0

Tabel di atas merupakan hasil dari pengujian sistem *monitoring* pencahayaan yang sepenuhnya dapat diakses melalui *Blynk Cloud*. *User acceptance test* yang telah dilaksanakan berdasarkan kasus pengujian memiliki hasil 100% sesuai dengan harapan *user*.

5.1.2 Performance Test

Performance testing dilakukan untuk memperoleh tingkat performansi pada sistem yang telah dibuat. Pada *testing* ini, peneliti membentuk parameter yang digunakan untuk memvalidasi serta menjadi tolak ukur dari sistem yang telah dibuat.

Tabel 5. 4 PT-01

<i>Performance test</i>			
Nama Sistem	Sistem <i>monitoring</i> kontrol temperatur		
Nomor Pengujian	PT-01		
No.	Suhu (°C)	Kecepatan Kipas (%)	Respon <i>cloud</i>
1.	25	10%	0,2 detik
2.	26	20%	0,2 detik
3.	27	30%	0,4 detik
4.	28	40%	0,2 detik
5.	29	50%	0,3 detik
6.	30	60%	0,2 detik
7.	31	70%	0,2 detik
8.	32	80%	0,2 detik
9.	33	90%	0,3 detik
10.	34	100%	0,2 detik

Pada sistem *monitoring* kontrol temperatur, dilakukan *performance testing* berupa tingkat kecepatan putaran kipas berdasarkan suhu yang ditangkap oleh sensor. Dapat dilihat pada tabel di atas bahwa semakin tingginya suhu lingkungan, maka semakin cepat juga perputaran yang dihasilkan oleh kipas sebagai alat pertukaran sirkulasi serta pengatur suhu. Berdasarkan hasil tes didapatkan hasil respon yang ditangkap oleh sistem dan diteruskan ke *cloud* dengan rata-rata 0,24 detik dengan tingkat keberhasilan 100%.

Tabel 5. 5 PT-02

<i>Performance test</i>			
Nama Sistem	Sistem <i>controlling</i> (menyalakan dan mematikan) <i>feeder</i> pakan		
Nomor Pengujian	PT-02		
No.	<i>Switch feeder</i>	Hasil	Respon <i>cloud</i>
1.	Nyala	Benar	0,4 detik
2.	Nyala	Benar	0,5 detik
3.	Nyala	Benar	0,4 detik
4.	Nyala	Benar	0,3 detik
5.	Nyala	Benar	0,4 detik
6.	Mati	Benar	0,4 detik
7.	Mati	Benar	0,2 detik
8.	Mati	Benar	0,2 detik
9.	Mati	Benar	0,2 detik
10.	Mati	Benar	0,2 detik

Pada sistem *controlling* (menyalakan dan mematikan) *feeder* pakan, dilakukan *performance testing* berupa mengaktifkan serta mematikan *switch* pada *Blynk Cloud* yang diteruskan ke DC Motor agar dapat menarik pakan ayam dari tempat penyimpanan. Berdasarkan hasil tes, didapatkan hasil respon yang diteruskan dari *cloud* ke sistem dengan rata-rata 0,32 detik dengan tingkat keberhasilan 100%.

Tabel 5. 6 PT-03

<i>Performance test</i>				
Nama Sistem	Sistem penjadwalan <i>feeder</i> pakan			
Nomor Pengujian	PT-03			
No.	<i>Feeder</i>	Jam	Hasil	Respon <i>cloud</i>
1.	Nyala	06.00	Benar	0,2 detik
2.	Nyala	14.00	Benar	0,2 detik
3.	Nyala	18.00	Benar	0,2 detik
4.	Mati	06.05	Benar	0,2 detik

5.	Mati	14.05	Benar	0,2 detik
6.	Mati	18.05	Benar	0,2 detik

Pada sistem pakan juga terdapat penjadwalan otomatis yang berada pada pukul 06.00-06.05, 14.00-14.05, dan 18.00-18.05. Dilakukan *performance testing* berupa pengamatan fungsionalitas pemberian dan pemberhentian pakan sesuai jadwal yang telah ditentukan. Berdasarkan hasil tes, didapatkan hasil respon yang diteruskan dari *cloud* ke sistem dengan rata-rata 0,2 detik dengan tingkat keberhasilan 100%.

Tabel 5. 7 PT-04

<i>Performance test</i>			
Nama Sistem		Sistem <i>monitoring</i> pencahayaan	
Nomor Pengujian		PT-04	
No.	<i>Intensitas cahaya</i>	Nyala lampu	Respon <i>cloud</i>
1.	100	Mati	0,1 detik
2.	90	Mati	0,2 detik
3.	80	Mati	0,2 detik
4.	70	Mati	0,2 detik
5.	60	Mati	0,1 detik
6.	50	Mati	0,2 detik
7.	40	Mati	0,2 detik
8.	30	Mati	0,1 detik
9.	20	Mati	0,1 detik
10.	10	Nyala	0,3 detik

Pada sistem *monitoring* pencahayaan, dilakukan *performance testing* berupa respon *Light Dependant Sensor* (LDR) terhadap intensitas cahaya yang ditangkap. Dapat dilihat pada tabel bahwa semakin gelap tingkat pencahayaan pada ruangan, maka dapat memicu sistem pencahayaan secara otomatis. Pada sistem pencahayaan ini, penyetelan pemicu pencahayaan ditetapkan pada 20 Lux yang artinya apabila suatu ruangan berada pada tingkat pencahayaan ≤ 20 Lux akan secara otomatis menyalakan lampu LED. Berdasarkan hasil tes didapatkan hasil

respon yang ditangkap oleh sistem dan diteruskan ke *cloud* dengan rata-rata 0,17 detik dengan tingkat keberhasilan 100%.

Tabel 5. 8 PT-05

<i>Performance test</i>			
Nama Sistem		Sistem peringatan notifikasi penyimpanan	
Nomor Pengujian		PT-05	
No.	Status penyimpanan	Notifikasi	Respon <i>cloud</i>
1.	100%	Tidak ada	0,1 detik
2.	90%	Tidak ada	0,2 detik
3.	80%	Tidak ada	0,1 detik
4.	70%	Tidak ada	0,2 detik
5.	60%	Tidak ada	0,2 detik
6.	50%	Tidak ada	0,2 detik
7.	40%	Tidak ada	0,5 detik
8.	30%	Tidak ada	0,1 detik
9.	20%	Tidak ada	0,2 detik
10.	10%	ada	0,2 detik

Pada sistem peringatan penyimpanan, dilakukan *performance testing* berupa tingkat ketersediaan pakan dan air minum pada masing-masing tempat penyimpanan. Dapat dilihat pada tabel bahwa semakin berkurangnya pakan dan air pada penyimpanan maka akan terus melaporkan pada sistem *cloud* agar dapat dimonitor. Berdasarkan penyetelan pada sistem, pemicu munculnya notifikasi adalah saat penyimpanan tersisa $\leq 15\%$, maka dari itu pada repetisi ke-10 muncul peringatan agar tempat penyimpanan dapat diisi kembali. Berdasarkan hasil tes, didapatkan hasil respon yang ditangkap oleh sistem dan diteruskan ke *cloud* dengan rata-rata 0,2 detik dengan tingkat keberhasilan 100%.

Tabel 5. 9 PT-06

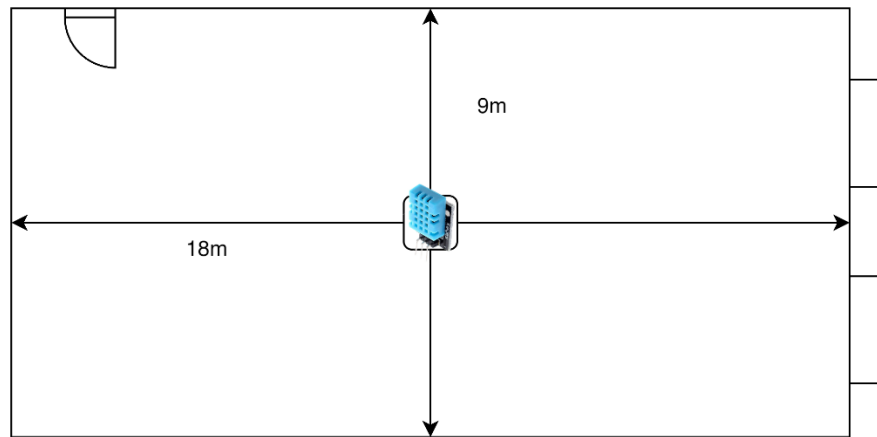
<i>Performance test</i>				
Nama Sistem	Sistem notifikasi <i>feeder</i> pakan			
Nomor Pengujian	PT-06			
No.	<i>Feeder</i>	Jam	Notifikasi	Respon <i>cloud</i>
1.	Nyala	06.00	Ada	0,2 detik
2.	Nyala	14.00	Ada	0,2 detik
3.	Nyala	18.00	Ada	0,2 detik
4.	Mati	06.05	Ada	0,2 detik
5.	Mati	14.05	Ada	0,2 detik
6.	Mati	18.05	Ada	0,2 detik

Pada sistem notifikasi *feeder*, dilakukan *performance testing* berupa pemberian notifikasi saat kondisi *feeder* nyala maupun berhenti. Dapat dilihat pada tabel bahwa setiap terjadi perubahan kondisi menyala dan mati maka akan diteruskan pelaporan pada sistem *cloud* agar dapat dimonitor. Berdasarkan hasil tes, didapatkan hasil respon yang dikirimkan oleh sistem *cloud* memiliki rata-rata 0,2 detik dengan tingkat keberhasilan 100%.

5.2 Pembahasan Proyeksi Peletakan Alat

5.2.1 Peletakan Sensor DHT11

Berdasarkan hasil pengujian pada 18 titik uji menggunakan *assessment* lingkungan kerja fisik (LKF) didapati titik paling optimal dalam penerapan *prototype* sistem automasi ini seperti pada gambar berikut:



Gambar 5. 1 Jangkauan Sensor DHT111

Hal tersebut didasari oleh data menurut (LME, 2023) yang dimana sebuah sensor DHT11 dapat menangkap data suhu lingkungan dengan proyeksi hingga 20 meter apabila diberi aliran daya sebesar 5V. Suhu yang ditangkap pada peletakan sensor di tengah bangunan mendapat nilai 32°C.

Selain itu, juga dilakukan pengukuran ulang pada 18 titik pengujian LKF untuk memastikan kebenaran hal tersebut. Berikut ini merupakan hasil pengujian pada titik yang ditentukan pada jam 15:00 sore hari:

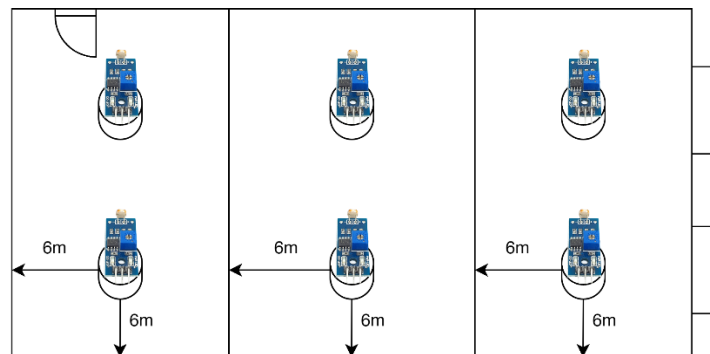
	6m	6m	6m	6m	6m	6m
6m	32	32	32	32	32	32
6m	32	32	32	32	32	32
6m	32	32	32	32	32	32

Gambar 5. 2 Data Hasil Penerapan LKF

Seluruh hasil yang didapat sesuai dengan data yang diambil dengan peletakan sebuah sensor DHT11 yang berada ditengah bangunan.

5.2.2 Peletakan Sensor LDR

Berdasarkan hasil pengujian pada 18 titik uji menggunakan *assessment* lingkungan kerja fisik (LKF) didapati titik paling optimal dalam penerapan *prototype* sistem automasi ini seperti pada gambar berikut:



Gambar 5. 3 Pengujian Proyeksi Sistem Pencahayaan

Hal tersebut didasari oleh data menurut (Bayusari, 2014) yang dimana sebuah sensor LDR dapat menangkap data pencahayaan dengan proyeksi hingga 6,5 meter apabila diberi aliran daya sebesar 5V. Intensitas cahaya yang ditangkap dengan penempatan pada enam titik seperti pada gambar stabil menghasilkan data dengan kondisi tanpa bantuan lampu sebesar 5 Lux dan setelah diberikan penerangan sebesar 34 Lux.

Selain itu, juga dilakukan pengukuran ulang pada 18 titik pengujian LKF untuk memastikan kebenaran hal tersebut. Berikut ini merupakan hasil pengujian pada titik yang ditentukan pada jam 19:00 malam hari:

Sebelum						
	6m	6m	6m	6m	6m	6m
6m	5	5	5	5	5	5
6m	5	5	5	5	5	5
6m	5	5	5	5	5	5
Sesudah						
	6m	6m	6m	6m	6m	6m
6m	34	34	34	34	34	34
6m	34	34	34	34	34	34
6m	34	34	34	34	34	34

Gambar 5. 4 Perbandingan Sistem Pencahayaan

Dapat dilihat pada hasil pengujian, penerangan sistem sudah merata pada intensitas cahaya yang ditetapkan untuk pengujian yaitu 34 Lux yang masih berada dijangkauan 20-50 Lux.

5.3 Pembahasan REBA (*Rapid Entire Body Assessment*)

Pada penelitian ini, terdapat pokok permasalahan yang dijadikan acuan untuk dilakukan perbaikan yaitu keluhan pekerja saat melakukan pemberian pakan kepada hewan ternak. Maka dari itu, peneliti membuat perancangan pengelolaan ternak ayam berbasis mikrokontroler. *Rapid Entire Body Assessment* merupakan suatu metode untuk melakukan pengukuran risiko MSD (*musculoskeletal disorder*) yang dialami para pekerja.

5.3.1 Sudut Postur Kerja

Berikut ini merupakan gambar pekerja dengan sudut-sudut yang terbentuk dalam melakukan pengangkatan karung pakan seberat 50 Kg. Sudut-sudut yang terbentuk diukur menggunakan *software Corel Draw*.



Gambar 5. 5 Sudut-sudut Tubuh Pekerja

Dari hasil pengukuran tersebut, sudut-sudut yang terbentuk menjadi dasar dalam pengukuran REBA, berikut merupakan rangkuman data sudut dari postur pekerja:

Tabel 5. 10 Rangkuman Sudut Yang Terbentuk

No.	Posisi Bagian Tubuh	Keterangan	Sudut (°)
1.	Leher	<i>Neck</i>	34,85
2.	Batang Tubuh	<i>Trunk</i>	55,3

No.	Posisi Bagian Tubuh	Keterangan	Sudut (°)
3.	Kaki	<i>Legs</i>	31,6
4.	Lengan Atas	<i>Upper Arm</i>	15,5
5.	Lengan Bawah	<i>Lower Arm</i>	59,51
6.	Pergelangan Tangan	<i>Wrist</i>	18,2

Keterangan beban, *coupling*, dan *activity score*:

Tabel 5. 11 Beban, *Coupling*, dan *Activity Score*

Beban yang diangkat	50 Kg
<i>Coupling</i>	<i>Poor</i> (Pegangan tangan tidak bisa diterima walaupun memungkinkan.)
<i>Activity Score</i>	<ul style="list-style-type: none"> - Satu atau lebih bagian tubuh statis, ditahan lebih dari satu menit - Gerakan menyebabkan perubahan atau pergeseran postur yang cepat dari postur awal

5.3.2 Perhitungan Skor REBA Sebelum Perancangan Pengelolaan Otomatis

Perhitungan skor REBA dilakukan menggunakan bantuan *ErgoFellow* yang merupakan perangkat lunak yang dikembangkan dalam ranah ergonomi. Berikut merupakan hasil perhitungan skor REBA sebelum perancangan pengelolaan peternakan ayam otomatis diimplementasikan:

REBA - DATABASE

Export

Name of the worker: Pekerja

Company: Peternakan

Department: Ayam

Function: -

Description of the task: Sebelum Perbaikan

Neck: More than 20 degrees Additional: No

Trunk: 20 to 60 degrees Additional: No

Legs: Support in the two legs, walking or seated Additional: 30 to 60 degrees

Load: > 10 kg (> 22 lb) Additional: No

Wrist: More than 15 degrees up or down Additional: No

Upper Arm: -20 to 20 degrees Additional: No No No

Lower Arm: 0 to 60 degrees or more than 100 degrees Coupling: Poor

Activity 1: One or more body parts are held for longer than 1 minute (static)

Activity 2: No

Activity 3: No

Result: 9

3 of 3

Buttons: PRINT, DELETE, SEARCH, COMPLETE LIST, BACK

Gambar 5. 6 Skor REBA Sebelum Perbaikan

Berdasarkan perhitungan skor REBA di atas, hasil skor menunjukkan angka 9 yang berarti memiliki risiko tinggi dan perlu segera dilakukan perbaikan. Perbaikan kerja yang dapat dilakukan adalah perancangan ulang peralatan kerja agar dapat membantu mengurangi risiko dari pekerjaan tersebut.

5.3.3 Perhitungan Skor REBA Setelah Perancangan Pengelolaan Otomatis

Hasil perhitungan skor REBA sebelum adanya perancangan pengelolaan peternakan otomatis, didapatkan skor 9 yaitu memiliki risiko tinggi. Maka dari itu, peneliti membuat perancangan pengelolaan otomatis agar dapat memudahkan pekerjaan pengelola peternakan. Di bawah ini merupakan hasil perhitungan skor REBA setelah dilakukan perancangan pengelolaan otomatis:

REBA - DATABASE

Export

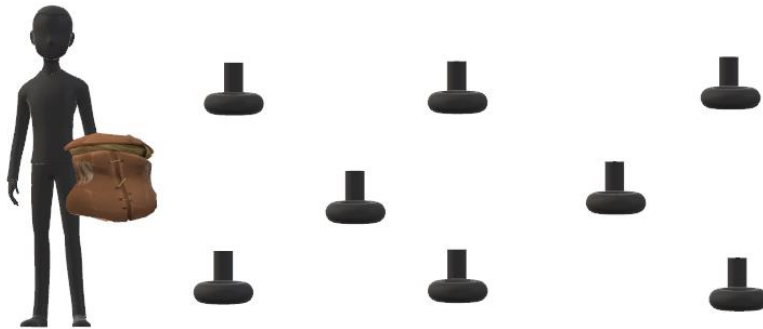
Name of the worker	Setelah Perbaikan			 PRINT DELETE SEARCH COMPLETE LIST BACK
Company	peternakan			
Department	ayam			
Function	karung			
Description of the task	angkat karung ke perbaikan			
Neck	0 to 20 degrees	Additional	Neck is twisted or side bending	
Trunk	Straight	Additional	No	
Legs	Support in the two legs, walking or seated	Additional	No	
Load	> 10 kg (> 22 lb)	Additional	No	
Wrist	Between 15 degrees up and 15 degrees down	Additional	No	
Upper Arm	20 to 45 degrees	Additional	No	No
Lower Arm	60 to 100 degrees	Coupling	Fair	
Activity 1	No			
Activity 2	No			
Activity 3	No			
Result:	3			

Navigation: 2 of 3

Gambar 5. 7 Skor REBA Setelah Perbaikan

Berdasarkan hasil perhitungan skor REBA di atas, didapatkan skor sebesar 3 yang termasuk risiko rendah.

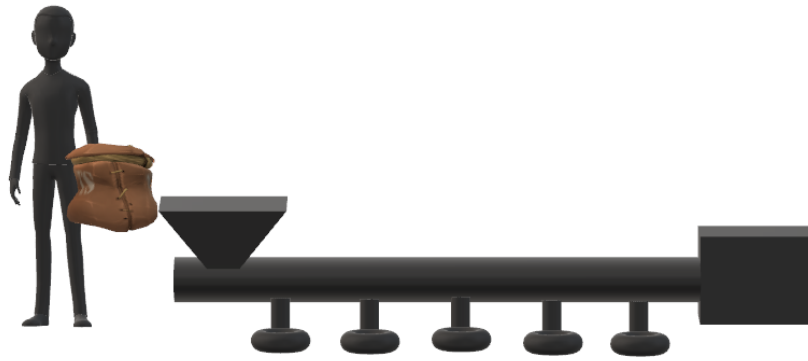
5.3.4 Perbandingan Beban Kerja Sebelum dan Sesudah Perancangan Otomatis



Gambar 5. 8 Ilustrasi Sebelum Dilakukan Perbaikan

Dapat dilihat pada ilustrasi di atas dan hasil perhitungan skor REBA, bahwa sebelum adanya perancangan otomatis pekerja melakukan pengangkatan beban sebesar 50 Kg dengan repetisi penuangan lebih 25 kali dalam satu sesi sehingga menyebabkan skor REBA 9 yang termasuk dalam kategori risiko tinggi.

Maka dari itu perlu dilakukan perbaikan dengan cara yang lebih efektif dan efisien dengan perancangan pengelolaan otomatis. Perancangan pengelolaan peternakan otomatis ini menggantikan proses kerja yang memiliki repetisi tinggi menjadi hanya dengan 1 kali pengangkatan. Berkurangnya intensitas pengangkatan dapat meningkatkan efisiensi, meringankan, dan membantu pekerja dalam meminimalisir MSD. Hal tersebut dibuktikan dengan perhitungan REBA yang dilakukan setelah perancangan pengelolaan otomatis berbasis mikrokontroler diterapkan, hasil skor menunjukkan angka 3 yang termasuk dalam kategori risiko rendah. Berikut ini merupakan ilustrasi pengangkatan beban setelah adanya perancangan pengelolaan otomatis:



Gambar 5. 9 Ilustrasi Setelah Dilakukan Perbaikan

Intensitas pengangkatan pakan yang telah berkurang hanya menjadi satu kali dan ketersediaan pakan dalam kontainer dapat dipantau melalui aplikasi atau *website* yang menggunakan internet. Dalam proses pemberian pakan juga dapat dilakukan kontrol dengan basis *cloud*. Penerapan pengelolaan otomatis ini dapat memudahkan pekerja peternakan ayam sehingga dapat meningkatkan efisiensi pengelolaan.

BAB VI

PENUTUP

6.1 Kesimpulan

Setelah dilakukan pembangunan serta pengujian pada sistem, diperoleh kesimpulan dari penelitian ini sebagai berikut:

1. Pengidentifikasian elemen dilakukan melalui *assessment* Lingkungan Kerja Fisik (LKF). *Assessment* dilakukan untuk mengetahui dua elemen, yaitu suhu dan pencahayaan yang diuji pada 18 titik. Pada elemen suhu, peternakan belum memenuhi standar. Suhu pada peternakan terlalu tinggi yaitu sebesar 32°C sehingga dapat menyebabkan stres termal pada ayam ternak yang dapat berakibat kematian. Elemen kedua yaitu pencahayaan, pada pencahayaan tanpa penerangan lampu sebesar 5 Lux dan dengan bantuan penerangan lampu sebesar 100 Lux yang masih belum memenuhi standar. Pencahayaan ini dapat berpengaruh pada pola makan ayam ternak. Standar yang ditetapkan pada pengelolaan peternakan ayam ini mengikuti DITJENNAK yaitu dengan *range* 25°C-30°C untuk elemen suhu dan 20 Lux - 50 Lux untuk elemen pencahayaan.
2. Perancangan sistem automasi dilakukan dengan metode *prototyping* untuk menunjang pengelolaan peternakan ayam berupa *monitoring* dan *controlling* dengan media *microcontroller* arduino IDE yang terintegrasi pada *Blynk Cloud*. Hasil perancangan tersebut berupa sistem temperatur, sistem pemberian pakan, sistem pencahayaan, dan sistem penyimpanan. Data untuk sistem temperatur didapat dari *assessment* LKF yang telah dilakukan dan hasil menunjukkan titik optimal penempatan sensor DHT11 pada tengah bangunan yang mencakup radius 20 meter serta automasi DC Fan yang telah disesuaikan dengan suhu yang ditangkap. Pada sistem pemberian pakan, telah dijadwalkan beserta banyaknya jumlah pakan yang diberikan sesuai dengan acuan DITJENNAK. Jumlah pemberian pakan terbagi menjadi tiga sesi pemberian. Data untuk sistem pencahayaan didapat dari *assessment* LKF dan hasil menunjukkan terdapat 6 titik optimal penempatan sensor LDR dengan cakupan radius 6,5 meter per sensor, sistem ini disesuaikan dengan standar optimal pencahayaan apabila intensitas cahaya dibawah 20 Lux akan otomatis menyalakan lampu LED. Pada sistem penyimpanan menggunakan sensor *ultrasonic*, untuk mengukur status sisa kapasitas penyimpanan apabila sensor melaporkan status penyimpanan

dibawah 15% akan otomatis mengirimkan notifikasi pada *smartphone user*. Maka dari itu perancangan sistem telah disesuaikan dengan standar optimal dan kebutuhan *user*.

3. Pekerjaan pada peternakan ayam ini belum ergonomis karena terdapat proses pengangkatan sebanyak 25 kali dengan beban 50Kg per sesinya yang mempengaruhi risiko cedera *musculoskeletal disorder* atau MSD pada pekerja. Tingkat risiko cedera pekerja diuji menggunakan *assessment* REBA. Berdasarkan uji pertama, pekerja memiliki tingkat risiko sangat tinggi dengan nilai skor 9 dikarenakan pengangkatan beban pakan berulang kali setiap harinya. Sistem konvensional yang masih diterapkan sebelumnya belum ergonomis sehingga peneliti mengimplementasikan sistem automasi yang telah disesuaikan dengan kebutuhan *user* pada sistem pemberian pakan. Setelah diterapkannya sistem automasi, dilakukan kembali uji REBA. Hasil menunjukkan tingkat risiko cedera pekerja mendapat skor 3 yang merupakan risiko rendah dikarenakan proses pengangkatan karung pakan dapat diminimalisir hingga tersisa 1 kali pengangkatan.

6.2 Saran

Berikut merupakan saran peneliti agar menjadi pertimbangan yang dapat dikembangkan pada penelitian selanjutnya:

1. Pengembangan proses pengangkatan yang masih tersisa 1 kali pengangkatan diharapkan kedepannya dapat dieliminasi.
2. Pada sistem pemberian pakan diharapkan ada sensor yang mendeteksi berat pakan pada tiap *feeder*, sehingga pendistribusian pakan lebih akurat.
3. Pada tempat penampungan air diharapkan ada alat pendeteksi kebersihan air. Mengingat ayam sangat bergantung pada kualitas air minum terlebih saat suhu lingkungan sedang tinggi.

DAFTAR PUSTAKA

- Abdullah, S., Khamis, N. K., Ghani, J. A., & Kurniawan, R. (2020). Posture Evaluation of the Automotive Maintenance Workers: A Case Study. *Jurnal Kejuruteraan*, 3(1), 65–70.
- Adha, L. H., Asyhadie, Z., & Kusuma, R. (2020). Digitalisasi Industri Dan Pengaruhnya Terhadap Ketenagakerjaan dan Hubungan Kerja Di Indonesia Industrial. *Jurnal Kompilasi Hukum*, V(2), 268–298.
- Adha, & Asyhadie. (2020). DIGITALISASI INDUSTRI DAN PENGARUHNYA TERHADAP KETENAGAKERJAAN DAN HUBUNGAN KERJA DI INDONESIA. *Jurnal Kompilasi Hukum*.
- Ahmad I, K. M. (2020). Prevalence of lateral epicondylitis among workers engaged in repetitive hand and wrist movements in a Pakistani engineering industry. *Int J Occup Saf Ergon*.
- Alazzam, M. &. (2021). A systematic review of prototyping methods for internet of things. *Sensors*.
- Ali Mazidi, S. N. (2017). *The AVR microcontroller and embedded systems: using assembly and C for AVR microcontrollers*. Pearson.
- Annur, C. M. (2022, Oktober 25). *Konsumsi Daging Ayam Ras di Rumah Tangga Naik 8,62% pada 2021*. Retrieved from Katadata Media Network (online): <https://databoks.katadata.co.id/datapublish/2022/10/25/konsumsi-daging-ayam-ras-di-rumah-tangga-naik-862-pada-2021> (27 Januari 2023)
- Arduino. (2020). Retrieved from arduino: <https://www.arduino.cc/> (09 Maret 2023)
- Banzi, M. (2022). *Getting Started with Arduino*. arduino.cc. Retrieved from Arduino.
- Basri, IY; Novaliandry, Dony; Tania, I. (2020). Design and Development of Inductive Sensor Trainer Based on Arduino and IDE 1.8.19 Software. *Jurnal Teknologi Informasi Dan Pendidikan (JTIP)*, 15(1). <https://doi.org/10.24036/tip.v15i1>
- Bayusari, I. (2014). *PENGARUH CAHAYA YANG DITERIMA SENSOR LDR (LIGHT DEPENDANT RESISTOR) PADA. 1(1)*.
- Bobsis. (2019). *Mengenal Mikrokontroler*. Retrieved from Binus University: <https://binus.ac.id/bandung/2019/11/mengenal-mikrokontroler/> (14 Maret 2023)
- BPS. (2021). *Populasi Ayam Ras Pedaging menurut Provinsi*. Retrieved from Badan Pusat Statistik (online): <https://www.bps.go.id/indicator/24/478/1/populasi-ayam-ras-pedaging-menurut-provinsi.html> (25 Januari 2023)

- BPSKM. (2022, Juli 8). *Produksi Peternakan Menurut Satuannya, 2017 - 2021*. Retrieved from Badan Pusat Statistik Kabupaten Majalengka (online): <https://majalengkakab.bps.go.id/statictable/2022/07/08/562/produksi-peternakan-menurut-satuannya-2017-2021.html> (28 Januari 2023)
- DITJENNAK. (2018, 14 3). *Pedoman Pemeliharaan Ayam Pedaging di Kandang dengan Sistem Litter*. Retrieved from Direktorat Jenderal Peternakan dan Kesehatan Hewan (online): <https://ditjennak.pertanian.go.id/website/petunjuk-umum-ternak-ayam-pedaging-di-kandang-litter> (28 Juni 2023)
- Doe, J. (2022). The Impact of Poor Physical Working Conditions on Poultry Farming. *Journal of Agricultural Science*.
- Dwiseptianto, R. W., & Wahyuningsih, A. S. (2021). Keluhan Muskuloskeletal pada Pekerja Sektor Informal. *Indonesian Journal of Public Health and Nutrition*, 1(1), 472–478. <http://journal.unnes.ac.id/sju/index.php/IJPHN>
- Dzukou T, S. I. (2021). Prevalence and risk factors of low back pain among office workers in a tertiary hospital in Nigeria. *J Back Musculoskelet Rehabil*.
- Fani, H. Al, Sumarno, S., Jalaluddin, J., Hartama, D., & Gunawan, I. (2020). Perancangan Alat Monitoring Pendeteksi Suara di Ruangan Bayi RS Vita Insani Berbasis Arduino Menggunakan Buzzer. *Jurnal Media Informatika Budidarma*, 4(1), 144. <https://doi.org/10.30865/mib.v4i1.1750>
- Fatallah, & Victor, D. (2017). *Small Changes Make Big Differences: The role of Ergonomics in Agriculture*. Resour. Mag.
- Fauzan. (2020). KOTAK PENERIMA PAKET BERBASIS IoT MENGGUNAKAN MODUL ESP32-CAM. *Institutional Repository UIN Syarif Hidayatullah Jakarta*, 1–66. <https://repository.uinjkt.ac.id/dspace/handle/123456789/56069>
- Goo YK, S. J. (2019). Prevalence and risk factors of carpal tunnel syndrome among radiologic technologists in South Korea. *Saf Health Work*.
- Haslindah, A., Andrie, A., Afrayana, R., & Endayani Guntur, S. Y. (2021). Analisa Postur Kerja Operator Mesin Pembuatan Adonan Mie Menggunakan Metode REBA (Rapid Entire Body Assesment). *Journal Industrial Engineering and Management (JUST-ME)*, 2(01), 39–45. <https://doi.org/10.47398/justme.v2i01.16>
- Hidayat, F. S. (2021). *FAKTOR – FAKTOR YANG MEMPENGARUHI PRODUKSI AYAM BROILER PADA CV MYTRA PRATAMA FARM BATANG*.

- Hossain, M. D., Aftab, A., Al Imam, M. H., Mahmud, I., Chowdhury, I. A., Kabir, R. I., & Sarker, M. (2018). Prevalence of work related musculoskeletal disorders (WMSDs) and ergonomic risk assessment among readymade garment workers of Bangladesh: A cross sectional study. *PLoS ONE*, 13(7), 1–18. <https://doi.org/10.1371/journal.pone.0200122>
- Huynh, M. Q. (2020). The role of prototyping in the design process: a case study. *International Journal of Human-Computer Interaction*, 333-342.
- Ifmalinda, Fahmi, K., & Putri, S. D. (2022). Implementation of Arduino Uno-Based Temperature Control for Drying Tamarind (*Garcinia xanthochymus*). *JURNAL KETEKNIKAN PERTANIAN*.
- Imtihan, K., Ernawati, & Mutawalli, L. (2022). PENERAPAN RESEARCH AND DEVELOPMENT (RND) DALAM MEMBANGUN ALAT PENYIRAM TANAMANA OTOMATIS BERBASIS ARDUINO. *Jurnal Manajemen Informatika & Sistem Informasi (MISI)*.
- Irawati, Achmed, S., & Agung, T. Y. (2022). PERANCANGAN MOBIL REMOT KONTROL ARDUINO DENGAN BLUETOOTH VIA ANDROID. *JE-UNISLA*, 7(1), 21–25.
- Jember, U. M. (2021). Retrieved from Repository UnMuhJember (online): <http://repository.unmuhjember.ac.id/7172/2/10.%20BAB%201%20Pendahuluan.pdf> (27 Januari 2023)
- Johnson, M. (2022, June 6). *How Automation Can Help Your Business*. Retrieved from Forbes Business Council: <https://www.forbes.com/sites/forbesbusinesscouncil/2022/06/06/how-automation-can-help-your-business/?sh=14cf4ac11642> (24 Maret 2023)
- Jusoh, M., Zuhariman, N., & Muttalib, M. (2020). Efficacy of Arduino based low-cost Resistive Sensor in evaluating Soil Moisture from different Soil types collected in the Kelantan-Terengganu plain of Malaysia. *IOP Conference Series: Earth and Environmental Science*, 596(1). <https://doi.org/10.1088/1755-1315/596/1/012097>
- KARIMUNKAB. (2019, Maret 13). *Sektor Peternakan*. Retrieved from Dinas Penanaman Modal dan Pelayanan Terpadu Satu Pintu Kabupaten Karimun (online).

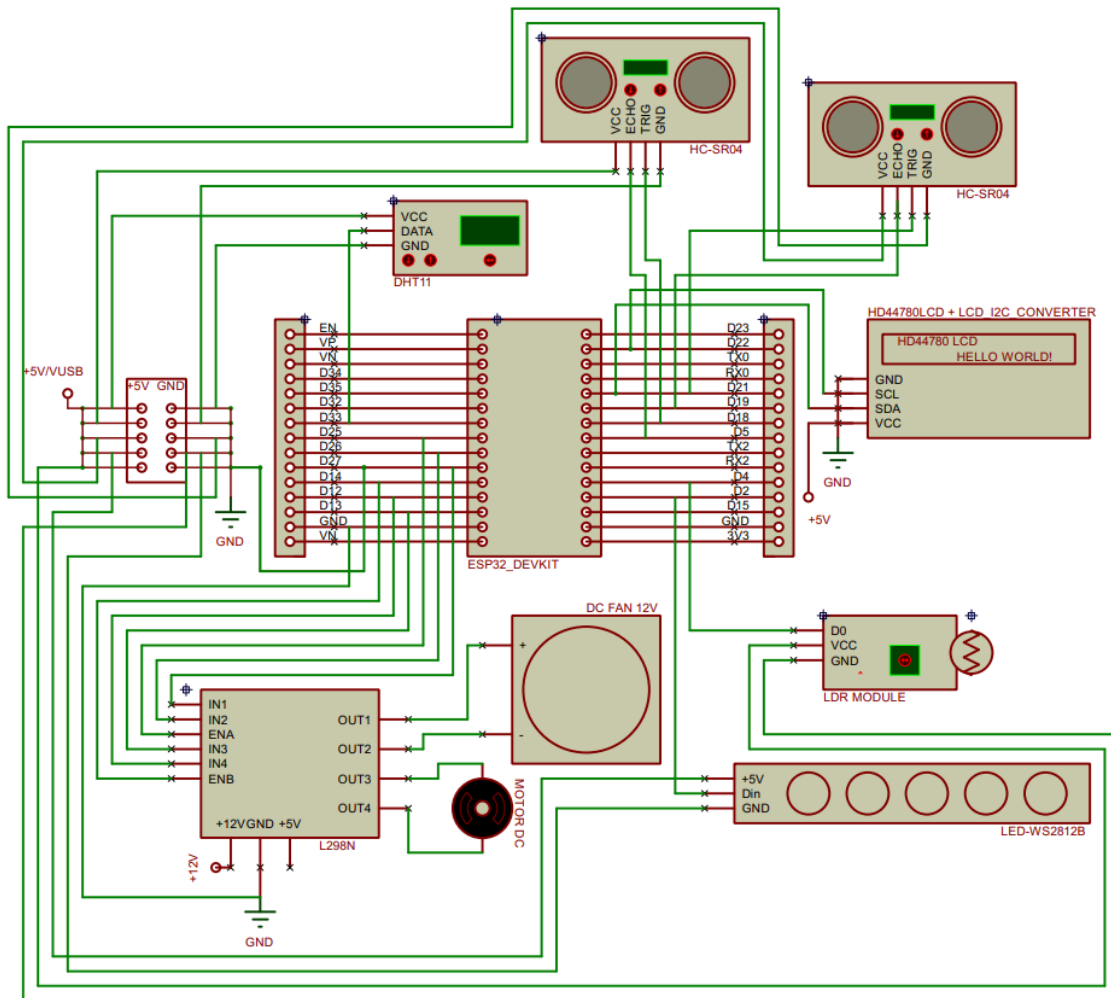
- Karnadi, A. (2022, April 15). *Agribisnis & Kehutanan*. Retrieved from DataIndonesia.id (online): <https://dataindonesia.id/sektor-riil/detail/produksi-ayam-pedaging-capai-343-juta-ton-pada-2021> (31 Januari 2023)
- KEMPERRI. (2020, September 16). Retrieved from KEMENTERIAN PERTANIAN REPUBLIK INDONESIA (online): <https://www.pertanian.go.id/home/?show=news&act=view&id=3472> (27 Januari 2023)
- Khoirul Anaam, I., Hidayat, T., Yuga Pranata, R., Abdillah, H., & Yhuto Wibisono Putra, A. (2022). Pengaruh trend otomasi dalam dunia manufaktur dan industri. *Vocational Education National Seminar*, 1(1), 46–50.
- Kim J, K. J. (2018). The association between work-related musculoskeletal disorders and fatigue: a cross-sectional study. *Int J Environ Res Public Health*.
- Lestari, E. D. (2021). Peran sektor pertanian, kehutanan dan perikanan dalam pembangunan daerah di kawasan gerbangkertosusila, Jawa Timur. *Repository.Uinjkt.Ac.Id*. <https://repository.uinjkt.ac.id/dspace/handle/123456789/56328>
- LME. (2023). *Interfacing DHT11 and DHT22 Sensors with Arduino (Online)*. Retrieved from Last Minute Engineer: <https://lastminuteengineers.com/dht11-dht22-arduino-tutorial/#:~:text=Wiring%20DHT11%20and%20DHT22%20Sensors%20to%20an%20Arduino,-Now%20it's%20time&text=Connecting%20DHT%20sensors%20to%20Arduino,pin%20to%20digital%20pin%20%238>. (10 Juli 2023)

- Martinus, Wahab, M. S., Yudi, & Ham, H. (2021). Data Transmission Using RFID System on Smart Shopping Carts for Checkout Process Efficiency in Supermarket at Indonesia. *Procedia Computer Science*, 179(2020), 902–912.
<https://doi.org/10.1016/j.procs.2021.01.080>
- Megawati, E., Saputra, W. S., Attaqwa, Y., & Fauzi, S. (2021). EDUKASI PENGURANGAN RESIKO TERJADINYA MUSCULOSKELETAL DISORDERS (MSDs) DINI, PADA PENJAHIT KELILING DI NGALIYAN SEMARANG. *Jurnal BUDIMAS*, 03(02), 450–456.
- Muhammad Zaky Nur Fuadi, Indra Darmawan, F. M. A. S. (2021). Penerapan Metode Waterfall pada Skema Sistem Pengaman Sepeda Motor dengan Arduino Nano. *Respati*, 16(2), 63. <https://doi.org/10.35842/jtir.v16i2.402>
- Nadziroh. (2020). Peran Sektor Pertanian Dalam Pertumbuhan Ekonomi Di Kabupaten Magetan. *Jurnal Agristan*.
- Parmadi, P., Emilia, E., & Zulgani, Z. (2018). Daya saing produk unggulan sektor pertanian Indonesia dalam hubungannya dengan pertumbuhan Ekonomi. *Jurnal Paradigma Ekonomika*, 13(2), 77–86. <https://doi.org/10.22437/paradigma.v13i2.6677>
- Rakhmanto, A. (2021, May 10). *Keuntungan dan Manfaat Menerapkan Automasi*. Retrieved from Cias Corporate Innovation Consulting: <https://www.cias.co/post/keuntungan-dan-manfaat-menerapkan-automasi> (24 Maret 2023)
- Sander, E. (Libby) J., Caza, A., & Jordan, P. J. (2019). Psychological perceptions matter: Developing the reactions to the physical work environment scale. *Building and Environment*, 148(August 2018), 338–347.
<https://doi.org/10.1016/j.buildenv.2018.11.020>
- Saputra, H. M., Sari, M., & Husna, M. (2020). FAKTOR PENYEBAB PRIMER DAN KOMBINASI DENGAN MUSCULOSKELETAL DISORDERS PADA PEKERJA PETERNAK AYAM DI NAGARI MUNGKA. *Journal of Industrial Hygiene and Occupational Health*, 5(1), 15. <https://doi.org/10.21111/jihoh.v5i1.4210>
- Setiawan Prabowo, P. (2020). Ekonomi dan Kebijakan Publik Indonesia Keywords. *Jurnal Ekonomi Dan Kebijakan Publik Indonesia*, 7(2), 156–169.
- Singh, J. &. (2018). Prototyping and its types in product design and development: A review. *International Journal of Engineering and Technology*, 107-114.
- Suleman, S., Hidayat, A. S., Ferdiansyah, D., Akhirianto, P. M., & Nuryadi, N. (2020).

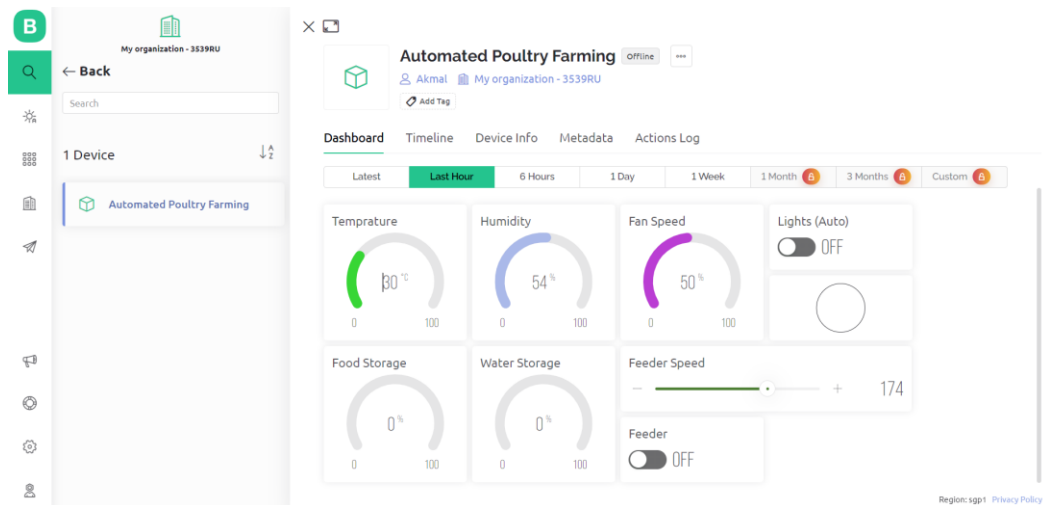
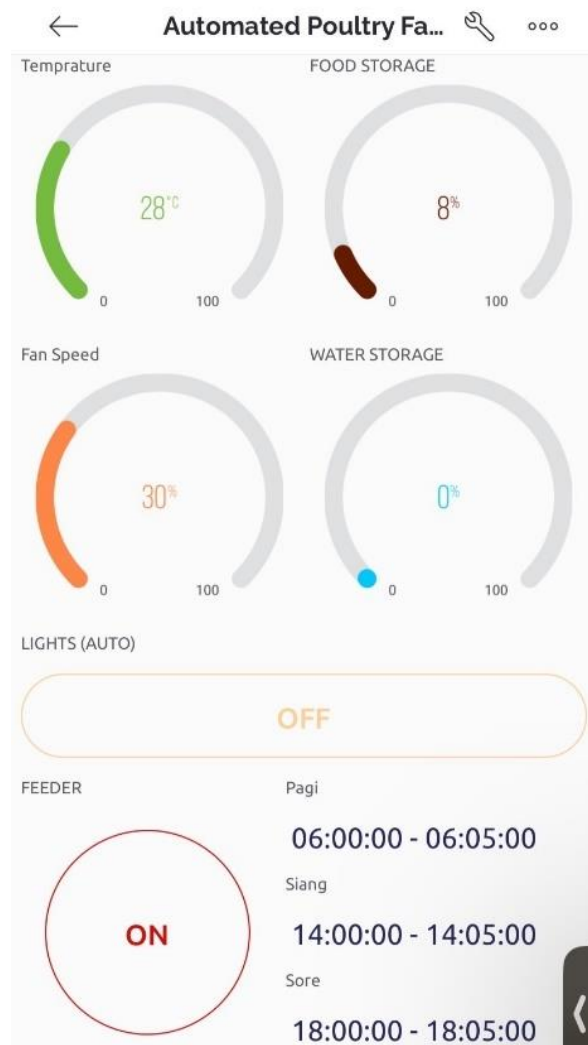
- Rancang Bangun Alat Pendeteksi kelembaban tanah dan Penyiram Otomatis Berbasis Arduino Uno. *Indonesian Journal on Software Engineering (IJSE)*, 6(2), 240–249. <https://doi.org/10.31294/ijse.v6i2.9120>
- Sun, Q. &. (2018). The function and realization method of prototyping in product design. *IOP Conference Series: Materials Science and Engineering*.
- Valerio, E., Hilmiati, N., Prior, J., & Dahlanuddin, D. (2022). Analysis of the agricultural innovation system in Indonesia: A case study of the beef sector in Nusa Tenggara Barat. *Agricultural Systems*, 203(October), 103529. <https://doi.org/10.1016/j.agry.2022.103529>
- Varmazyar, S. F. (2017). Evaluation of the REBA Method with Computer Software and by Expert Judgment. *International Journal of Occupational Hygiene*.
- Yaqin, N., Hariono, T., & Rohman, R. U. (2021). *Automatic Water Level Control Tem On Hydroponic Plants Based On Arduino*.
- Zhang Y, J. J. (2018). Epidemiology of osteoarthritis. *Clin Geriatr Med*.

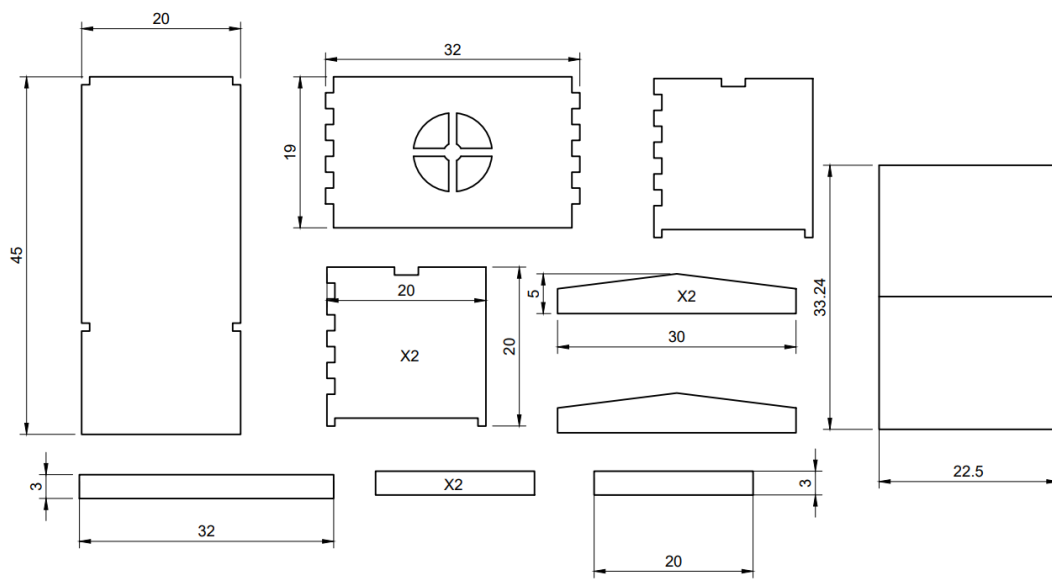
LAMPIRAN

Lampiran 1. Hasil Perancangan Skematik Sistem



Lampiran 2. Hasil Dashboard Blynk Cloud IoT



Lampiran 3. Desain Model 3D *Prototype*

Lampiran 4. Perancangan *Prototype*





Lampiran 5. Kuesioner *User Acceptance Test*

<i>User Acceptance Test</i>			
Nama Sistem	Sistem <i>monitoring-controlling feeder</i> pakan serta minum		
Nomor Pengujian	UAT-01		
Tanggal Pengujian			
No.	Fungsi Pokok	Sesuai	
		Ya	Tidak
1.	Memantau status sisa penyimpanan pakan		
2.	Menerima notifikasi apabila penyimpanan di bawah 15%		
3.	Mematikan dan menyalakan kontrol pemberian pakan		
4.	Mengakses sistem melalui <i>smartphone</i>		
Jumlah			
<i>User Acceptance Test</i>			
Nama Sistem	Sistem <i>monitoring</i> kontrol temperatur		
Nomor Pengujian	UAT-02		
Tanggal Pengujian			
No.	Fungsi Pokok	Sesuai	
		Ya	Tidak
1.	Memantau suhu dan kelembaban lingkungan		
2.	Memantau kecepatan kipas sebagai sistem sirkulasi udara		
3.	Mengakses sistem melalui <i>smartphone</i>		
Jumlah			
<i>User Acceptance Test</i>			
Nama Sistem	Sistem <i>monitoring</i> sistem pencahayaan		
Nomor Pengujian	UAT-03		
Tanggal Pengujian			
No.	Fungsi Pokok	Sesuai	
		Ya	Tidak
1.	Memantau status nyala lampu		
2.	Mematikan dan menyalakan lampu		
3.	Mengakses sistem melalui <i>smartphone</i>		
Jumlah			

Lampiran 6. Lembar *Performance Test*

<i>Performance test</i>			
Nama Sistem	Sistem <i>monitoring</i> kontrol temperatur		
Nomor Pengujian	PT-01		
No.	Suhu (°C)	Kecepatan Kipas (%)	Respon <i>cloud</i>
1.	25		
2.	26		
3.	27		
4.	28		
5.	29		
6.	30		
7.	31		
8.	32		
9.	33		
10.	34		
<i>Performance test</i>			
Nama Sistem	Sistem <i>controlling</i> (menyalakan dan mematikan) <i>feeder</i> pakan		
Nomor Pengujian	PT-02		
No.	<i>Switch feeder</i>	Hasil	Respon <i>cloud</i>
1.	Nyala		
2.	Nyala		
3.	Nyala		
4.	Nyala		
5.	Nyala		
6.	Mati		
7.	Mati		
8.	Mati		
9.	Mati		
10.	Mati		

<i>Performance test</i>			
Nama Sistem		Sistem <i>monitoring</i> pencahayaan	
Nomor Pengujian		PT-03	
No.	<i>Intensitas cahaya</i>	Nyala lampu	Respon <i>cloud</i>
1.	100		
2.	90		
3.	80		
4.	70		
5.	60		
6.	50		
7.	40		
8.	30		
9.	20		
10.	10		
<i>Performance test</i>			
Nama Sistem		Sistem peringatan notifikasi penyimpanan	
Nomor Pengujian		PT-04	
No.	Status penyimpanan	Notifikasi	Respon <i>cloud</i>
1.	100%		
2.	90%		
3.	80%		
4.	70%		
5.	60%		
6.	50%		
7.	40%		
8.	30%		
9.	20%		
10.	10%		

Lampiran 7. Foto Pengangkatan Beban Pakan



Lampiran 8. Pengkodean Arduino IDE

```
//BLYNK
#define BLYNK_TEMPLATE_ID "TMPL5X50ERuS"
#define BLYNK_TEMPLATE_NAME "Automated Poultry Farming"
#define BLYNK_AUTH_TOKEN "T10irugKyILVfUBSZ5zoq16CBYBHyIFx"
#define BLYNK_PRINT Serial
#include <WiFi.h>
#include <BlynkSimpleEsp32.h>
char auth[] = BLYNK_AUTH_TOKEN;
char ssid[] = "Rumah Semut";
char pass[] = "balikpapan123";
BlynkTimer timer;

//LCD
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);

//TEMPERATURE CONTROL
#include "DHT.h"
#define DHTPIN 33
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
int temp;
int tempMin = 25;
int tempMax = 35;

//FAN CONTROL
const int fanfreq = 5000;
const int fanpwmChannel = 0;
const int fanresolution = 8;
const int fan = 25;
const int IN1_PIN = 27;
const int IN2_PIN = 26;
int fanSpeed;
int fanLCD;

//FEEDER CONTROL
const int motorfreq = 5000;
const int motorpwmChannel = 2;
const int motorresolution = 8;
const int M1PWM = 14;
const int M1F = 13;
const int M1R = 12;
```

```

//Schedule
#define APP_DEBUG
#include <NTPClient.h>
#include <WiFiUdp.h>
WiFiUDP ntpUDP;
const long utcOffsetInSeconds = 25200;
NTPClient timeClient(ntpUDP, "asia.pool.ntp.org", utcOffsetInSeconds);
int startTime1;
int endTime1;
int startTime2;
int endTime2;
int startTime3;
int endTime3;

//LED AND LDR
#include <FastLED.h>
#define LED_TYPE WS2812B
#define COLOR_ORDER GRB
#define NUM_LEDS 15
#define DATA_PIN 2
#define BRIGHTNESS 200
CRGB leds[NUM_LEDS];
int LDRpin = 4;
WidgetLED led1(V2);

//WaterSensor
#define wecho 5
#define wtrig 18
int wdepth = 7.5;

//FoodSensor
#define fecho 19
#define ftrig 23
int fdepth = 12;

void setup() {
  lcd.init();
  lcd.backlight();
  Blynk.begin(auth, ssid, pass);
  dht.begin();

  pinMode(fan, OUTPUT);
  pinMode(IN1_PIN, OUTPUT);
  pinMode(IN2_PIN, OUTPUT);
  pinMode(M1PWM, OUTPUT);
}

```

```

pinMode(M1F, OUTPUT);
pinMode(M1R, OUTPUT);
digitalWrite(M1F, LOW);
digitalWrite(M1R, LOW);
pinMode (LDRpin, INPUT);
pinMode(wtrig, OUTPUT);
pinMode(wecho, INPUT);
pinMode(ftrig, OUTPUT);
pinMode(fecho, INPUT);
FastLED.addLeds<LED_TYPE, DATA_PIN, COLOR_ORDER>(leds, NUM_LEDS);

// Set the frequency of the PWM signal
ledcSetup(fanpwmChannel, fanfreq, fanresolution);
ledcAttachPin(fan, fanpwmChannel);
ledcSetup(motorpwmChannel, motorfreq, motorresolution);
ledcAttachPin(M1PWM, motorpwmChannel);
ledcWrite(motorpwmChannel, 120);

//LCD SHOW
lcd.setCursor(4,0);
lcd.print("Starting");
lcd.setCursor(2,1);
lcd.print("Components..");
delay(2500);
lcd.clear();

timeClient.begin();
Serial.begin(115200);

timer.setInterval(100L, DHT11sensor);
timer.setInterval(100L, FanControl);
timer.setInterval(100L, LightControl);
timer.setInterval(100L, FoodStor);
timer.setInterval(100L, WaterLevel);
}

//=====DHT11=====
void DHT11sensor() {
  temp = dht.readTemperature();
  Blynk.virtualWrite(V5, temp);
  lcd.setCursor(0, 0);
  lcd.print("Temp:");
  lcd.print(temp);
  lcd.print("C");
}

//=====FANCONTROL=====

```

```

void FanControl(){
  digitalWrite(IN1_PIN, HIGH);
  digitalWrite(IN2_PIN, LOW);
  temp = dht.readTemperature();
  if(temp < tempMin)
  {
    fanSpeed = 170;
    ledcWrite(fanpwmChannel, fanSpeed);
    fanLCD=0;
  }
  else if((temp >= tempMin) && (temp <= tempMax))
  {
    fanSpeed = map(temp, tempMin, tempMax, 170, 255);

    fanLCD = map(temp, tempMin, tempMax, 0, 100);
    ledcWrite(fanpwmChannel, fanSpeed);
  }
  lcd.setCursor(0,1);
  lcd.print("FanSpeed:");
  lcd.print(fanLCD);
  lcd.print("%");
  Blynk.virtualWrite(V7, fanLCD);
}

//=====FEEDER=====
BLYNK_WRITE(V8)
{
  int value = param.asInt();
  if(value == 1){
    digitalWrite(M1F, HIGH);
    lcd.setCursor(9,0);
    lcd.print("Feed:*");
  }
  else{
    digitalWrite(M1F, LOW);
    lcd.setCursor(9,0);
    lcd.print("Feed:o");}
}

BLYNK_WRITE(V1){
  startTime1 = param[0].asInt();
  endTime1 = param[1].asInt();
  if(startTime1 == 0 && endTime1 == 0)
  {
    startTime1 = 999999;
    endTime1 = 999999;
  }
}

```

```

    Serial.println(startTime1);
    Serial.println(endTime1);
}

BLYNK_WRITE(V9){
    startTime2 = param[0].asInt();
    endTime2 = param[1].asInt();
    if(startTime2 == 0 && endTime2 == 0)
    {
        startTime2 = 999999;
        endTime2 = 999999;
    }
    Serial.println(startTime2);
    Serial.println(endTime2);
}

BLYNK_WRITE(V6){
    startTime3 = param[0].asInt();
    endTime3 = param[1].asInt();
    if(startTime3 == 0 && endTime3 == 0)
    {
        startTime3 = 999999;
        endTime3 = 999999;
    }
    Serial.println(startTime3);
    Serial.println(endTime3);
}

//=====LIGHTS=====
void LightControl(){
    Serial.println(digitalRead(LDRpin));
    if(digitalRead(LDRpin)==0){
        fill_solid( &(leds[0]), 15, CRGB(0, 0, 0));
        lcd.setCursor(13,1);
        lcd.print("L:o");
        FastLED.show();
        led1.off();
    }
    else{
        fill_solid( &(leds[0]), 15, CRGB(255, 255, 255));
        lcd.setCursor(13,1);
        lcd.print("L:*");
        FastLED.show();
        led1.on();
    }
}
}

```

```
//=====STORAGE=====
void WaterLevel(){
    digitalWrite(wtrig, LOW);
    delayMicroseconds(2);
    digitalWrite(wtrig, HIGH);
    delayMicroseconds(10);
    digitalWrite(wtrig, LOW);
    long t = pulseIn(wecho, HIGH);
    long cm = t / 29 / 2;
    Serial.println(cm);
    long wlevel= wdepth-cm;
    if (wlevel<0)
        wlevel=0;
    wlevel = map(wlevel,0,wdepth,0,100);
    Blynk.virtualWrite(V4, wlevel);
}

void FoodStor(){
    digitalWrite(ftrig, LOW);
    delayMicroseconds(2);
    digitalWrite(ftrig, HIGH);
    delayMicroseconds(10);
    digitalWrite(ftrig, LOW);
    long t = pulseIn(fecho, HIGH);
    long cm = t / 29 / 2;
    Serial.println(cm);
    long flevel= fdepth-cm;
    if (flevel<0)
        flevel=0;
    flevel = map(flevel,0,fdepth,0,100);
    Blynk.virtualWrite(V3, flevel);
}

//=====
void loop() {
    timeClient.update();
    int HH = timeClient.getHours();
    int MM = timeClient.getMinutes();
    int SS = timeClient.getSeconds();
    int server_time = 3600*HH + 60*MM + SS;

    if((startTime1 == server_time) ||
        (startTime2 == server_time) ||
        (startTime3 == server_time))
    {
        digitalWrite(M1F, HIGH);
        Blynk.virtualWrite(V8, 1);
        lcd.setCursor(9,0);
    }
}
```

```
    lcd.print("Feed:*");
}
else if ((endTime1 == server_time)||
         (endTime2 == server_time)||
         (endTime3 == server_time))
{
    digitalWrite(M1F, LOW);
    Blynk.virtualWrite(V8, 0);
    lcd.setCursor(9,0);
    lcd.print("Feed:o");
}

Blynk.run();//Run the Blynk library
timer.run();//Run the Blynk timer
}
```