



***RUNNING LED DISPLAY* UNTUK MENAMPILKAN INFORMASI  
KEDATANGAN DOSEN**

**LAPORAN TUGAS AKHIR**

**Program Studi  
S1 TEKNIK KOMPUTER**

**Oleh:**

**REZA ALDEV MAHENDRA**

**19410200042**

---

**FAKULTAS TEKNOLOGI DAN INFORMATIKA**

**UNIVERSITAS DINAMIKA**

**2023**

***RUNNING LED DISPLAY* UNTUK MENAMPILKAN  
INFORMASI KEDATANGAN DOSEN**

**TUGAS AKHIR**

**Diajukan sebagai salah satu syarat untuk menyelesaikan  
Program Sarjana Teknik**



**Disusun Oleh:**

**Nama : Reza Aldev Mahendra**

**NIM : 19410200042**

**Program : S1 (Strata Satu)**

**Jurusan : Teknik Komputer**

**FAKULTAS TEKNOLOGI DAN INFORMATIKA**

**UNIVERSITAS DINAMIKA**

**2023**

## TUGAS AKHIR

### ***RUNNING LED DISPLAY* UNTUK MENAMPILKAN INFORMASI KEDATANGAN DOSEN**

Dipersiapkan dan disusun oleh:

**Reza Aldev Mahendra**

**NIM: 19410200042**

Telah diperiksa, dibahas, dan disetujui oleh Dewan Pembahas

Pada: 21 Juli 2023

#### Susunan Dewan Pembahas

##### **Pembimbing:**

**I. Harianto, S.Kom., M.Eng.**

NIDN 0722087701


**II. Pauladie Susanto, S.Kom., M.T.**


NIDN 0729047501


##### **Pembahas:**

**Heri Pratikno, M.T., MTCNA., MTCRE.**

NIDN 0716117302

  
cn=Harianto Harianto,  
o=Universitas Dinamika,  
ou=Prodi S1 Teknik Komputer,  
email=hari@dinamika.ac.id, c=ID  
2023.07.27 14:07:23 +07'00'

  
cn=Pauladie Susanto, o=FTI Undika,  
ou=Prodi S1 TK,  
email=pauladie@dinamika.ac.id, c=ID  
2023.07.27 12:30:39 +07'00'

  
Digitally signed by  
Heri Pratikno, M.T.  
Date: 2023.07.28  
11:01:07 +07'00'

Tugas Akhir ini telah diterima sebagai salah satu persyaratan  
untuk memperoleh gelar sarjana



Digitally signed by  
Universitas Dinamika  
Date: 2023.08.02  
11:06:30 +07'00'

**Tri Sagirani, S.Kom., M.MT.**

NIDN: 0731017601

Dekan Fakultas Teknologi dan Informatika

UNIVERSITAS DINAMIKA



*“Seorang terpelajar harus sudah berbuat adil sejak dalam pikiran apalagi dalam perbuatan”*

*-Pramoedya Ananta Toer-*

UNIVERSITAS  
Dinamika



*Atas berkat Tuhan yang Maha Esa,*

*Ku persembahkan kepada keluarga saya untuk dukungannya selama ini dan doa tiada hentinya. Bersama teman-teman Teknik Komputer Angkatan 2019. Terimakasih atas bantuan doa, dukungan yang membuat saya lebih semangat dan terus belajar menjadi pribadi yang lebih baik lagi.*

**PERNYATAAN**  
**PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH**

Sebagai mahasiswa **Universitas Dinamika**, Saya :

Nama : Reza Aldev Mahendra  
NIM : 19410200042  
Program Studi : S1 Teknik Komputer  
Fakultas : Fakultas Teknologi dan Informatika  
Jenis Karya : Laporan Tugas Akhir  
Judul Karya : **RUNNING LED DISPLAY UNTUK MENAMPILKAN  
INFORMASI KEDATANGAN DOSEN**

Menyatakan dengan sesungguhnya bahwa :

1. Demi pengembangan Ilmu Pengetahuan, Teknologi dan Seni, Saya menyetujui memberikan kepada **Universitas Dinamika** Hak Bebas Royalti Non-Eksklusif (*Non-Exclusive Royalty Free Right*) atas seluruh isi/sebagian karya ilmiah Saya tersebut diatas untuk disimpan, dialihmediakan, dan dikelola dalam bentuk pangkalan data (*database*) untuk selanjutnya didistribusikan atau dipublikasikan demi kepentingan akademis dengan tetap mencantumkan nama Saya sebagai penulis atau pencipta dan sebagai pemilik Hak Cipta.
2. Karya tersebut diatas adalah hasil karya asli Saya, bukan plagiat baik sebagian maupun keseluruhan. Kutipan, karya, atau pendapat orang lain yang ada dalam karya ilmiah ini semata-mata hanya sebagai rujukan yang dicantumkan dalam Daftar Pustaka Saya.
3. Apabila dikemudian hari ditemukan dan terbukti terdapat tindakan plagiasi pada karya ilmiah ini, maka Saya bersedia untuk menerima pencabutan terhadap gelar kesarjanaan yang telah diberikan kepada Saya.

Demikian surat pernyataan ini Saya buat dengan sebenar-benarnya.

Surabaya, 21 Juli 2023



METERAI  
TEMPEL  
5AAKX522098459

Reza Aldev Mahendra  
NIM : 19410200042

## ABSTRAK

*Running LED Display* adalah salah satu metode yang paling banyak digunakan untuk memberikan informasi terperinci dan meningkatkan pengalaman pengguna. Dalam penelitian ini akan membuat *Running LED Display* yang dapat digunakan untuk menampilkan status informasi kehadiran dosen. Fungsi kontrol lampu LED Matrix P10 menggunakan mikrokontroler ESP8266, mikrokontroler ini dapat terhubung ke jaringan internet melalui koneksi Wi-Fi. Untuk mendapatkan data dari URL ([https://e-sicyca.dinamika.ac.id/api/today\\_attendance?prodi=41020](https://e-sicyca.dinamika.ac.id/api/today_attendance?prodi=41020)) diperlukan library yang kompatibel dengan ESP8266. Pengujian yang dilakukan atas kemampuan ESP8266 dalam mengontrol tampilan LED pada saat pengambilan data dari URL, hasil pengujian menunjukkan bahwa *Running LED Display* dapat menampilkan data dengan baik sesuai dengan yang diinginkan, proses pengambilan data nama dan kehadiran dosen menggunakan *get* HTTPS ke URL. Proses pengambilan data dari PPTI menggunakan 2 *hotspot* yang berbeda mempunyai selisih 0,002 detik saja, pada saat proses pengambilan data tersebut terjadi *flicker* pada tampilan LED yang mana lamanya *flicker* tersebut dipengaruhi oleh kualitas jaringan, tampilan LED Matrix P10 dapat menampilkan karakter hingga ribuan dalam satu waktu, sehingga dapat menampilkan informasi yang panjang.

**Kata kunci** : LED P10, URL, HTTPS, ESP8266

## KATA PENGANTAR

Puji syukur saya panjatkan pada Tuhan Yang Maha Esa atas segala rahmat yang telah diberikan-Nya, sehingga penulis dapat menyelesaikan Laporan Tugas Akhir yang berjudul “*Running LED Display* untuk Menampilkan Informasi Kedatangan Dosen”. Dalam perjalanan menyelesaikan pengerjaan Laporan Tugas Akhir ini penulis banyak mendapatkan bantuan dari berbagai pihak. Oleh karena itu penulis mengucapkan terima kasih kepada:

1. Allah SWT, karena dengan rahmat-Nya penulis dapat menyelesaikan Laporan Tugas Akhir ini.
2. Orang tua dan seluruh keluarga yang telah memberikan dorongan dan dukungan baik secara moril maupun materiil, sehingga penulis dapat menempuh dan menyelesaikan Laporan Tugas Akhir ini.
3. Ibu Tri Sagirani, S.Kom., M.MT., selaku Dekan Fakultas Teknologi dan Informatika (FTI) Universitas Dinamika.
4. Bapak Pauladie Susanto, S.Kom., M.T., selaku Ketua Program Studi S1 Teknik Komputer Universitas Dinamika.
5. Bapak Heri Pratikno, M.T., MTCNA., MTCRE., selaku dosen pembahas. Penulis mengucapkan terimakasih atas bimbingan yang diberikan dan kesempatan serta tuntunan baik secara lisan maupun tertulis, sehingga penulis dapat menyelesaikan Tugas Akhir.
6. Bapak Harianto, S.Kom., M.Eng., selaku dosen pembimbing yang telah memberikan dukungan berupa motivasi, wawasan, dan saran bagi penulis selama pelaksanaan pengerjaan Tugas Akhir dan dalam pembuatan laporan Tugas Akhir.
7. Bapak Pauladie Susanto, S.Kom., M.T., selaku dosen pembimbing yang banyak memberikan masukan dan solusi agar Tugas Akhir ini menjadi lebih baik dan penulis dapat menyelesaikan Tugas Akhir ini.
8. Seluruh rekan-rekan S1 Teknik Komputer angkatan 2019 yang telah memberikan dukungan dan semangatnya untuk membantu penulis menyelesaikan laporan Tugas Akhir ini.



9. Seluruh pihak yang tidak dapat disebutkan satu per satu yang telah memberikan dukungan serta bantuan dalam segala bentuk yang akhirnya terselesaikannya laporan Tugas Akhir ini.

Penulis berharap semoga laporan ini dapat berguna dan bermanfaat untuk menambah wawasan bagi pembacanya. Penulis juga menyadari dalam penulisan laporan ini banyak terdapat kekurangan. Oleh karena itu, penulis sangat mengharapkan saran dan kritik untuk memperbaiki kekurangan dan berusaha untuk lebih baik lagi.

Surabaya, 21 Juli 2023

Penulis



UNIVERSITAS  
Dinamika

## DAFTAR ISI

	Halaman
ABSTRAK .....	vii
KATA PENGANTAR .....	viii
DAFTAR ISI.....	x
DAFTAR GAMBAR .....	xiii
DAFTAR TABEL.....	xiv
DAFTAR LAMPIRAN.....	xv
BAB I PENDAHULUAN.....	10
1.1 Latar Belakang.....	10
1.2 Rumusan Masalah.....	12
1.3 Batasan Masalah .....	12
1.4 Tujuan .....	13
1.5 Manfaat .....	13
BAB II LANDASAN TEORI.....	14
2.1 PPTI Universitas Dinamika .....	14
2.2 NodeMCU ESP8266.....	14
2.3 <i>Display</i> LED Matrix P10 .....	15
2.4 Arduino IDE .....	16
2.5 PCB Controller JWS 2.4.....	17
2.6 Power Supply 5V .....	17
2.7 HTTPS .....	18
BAB III METODOLOGI PENELITIAN.....	19
3.1 Blok Diagram.....	19
3.2 Perancangan Perangkat Keras.....	19
3.3 <i>Flowchart</i> .....	20
3.4 Subproses WiFi.....	21
3.5 Subproses Waktu .....	22
3.6 Subproses Inisialisasi Variabel .....	23
3.7 Subproses <i>Update</i> Waktu .....	24
3.8 Subproses <i>Update</i> Data PPTI .....	24

3.9	Subproses <i>Running</i> LED.....	25
3.10	Mengambil Data Dari <i>Link</i> URL yang Telah Diberikan oleh PPTI .....	26
3.10.1	Mengakses <i>Link</i> URL dengan <i>Get</i> HTTP .....	26
3.10.2	Mengambil data Nama dan Kehadiran saja pada URL.....	27
3.11	Desain Mekanik .....	28
BAB IV HASIL DAN PEMBAHASAN.....		37
4.1	Pengujian Ketepatan Waktu .....	37
4.1.1	Tujuan.....	37
4.1.2	Alat dan Bahan .....	37
4.1.3	Prosedur.....	37
4.1.4	Analisis Pengujian Ketepatan Waktu .....	39
4.2	Pengujian Ketepatan Data PPTI .....	39
4.2.1	Tujuan.....	39
4.2.2	Alat dan Bahan .....	39
4.2.3	Prosedur.....	40
4.2.4	Analisis Ketepatan Data PPTI.....	41
4.3	Pengujian Waktu Proses (Pengambilan Waktu).....	42
4.3.1	Tujuan.....	42
4.3.2	Alat dan Bahan .....	42
4.3.3	Prosedur.....	42
4.3.4	Analisis Pengujian Waktu Proses (Pengambilan Waktu) .....	45
4.4	Pengujian Waktu Proses (Pengambilan Data PPTI).....	46
4.4.1	Tujuan.....	46
4.4.2	Alat dan Bahan .....	46
4.4.3	Prosedur.....	46
4.4.4	Analisis Pengujian Waktu Proses (Pengambilan Data PPTI) .....	49
4.5	Pengujian Seluruh Alat .....	49
4.5.1	Tujuan.....	49
4.5.2	Alat dan Bahan .....	49

4.5.3	Prosedur.....	50
4.5.4	Analisis Pengujian Seluruh Alat .....	52
4.6	Pengujian Panjang Karakter Pada LED Matrix P10.....	53
4.6.1	Tujuan.....	53
4.6.2	Alat dan Bahan .....	53
4.6.3	Prosedur.....	53
4.6.4	Analisis Pengujian Panjang Karakter Pada LED Matrix P10 .....	54
4.7	Penjelasan Konektor yang digunakan.....	54
BAB V PENUTUP.....		57
5.1	Kesimpulan.....	57
5.2	Saran.....	57
DAFTAR PUSTAKA .....		39
LAMPIRAN .....		41



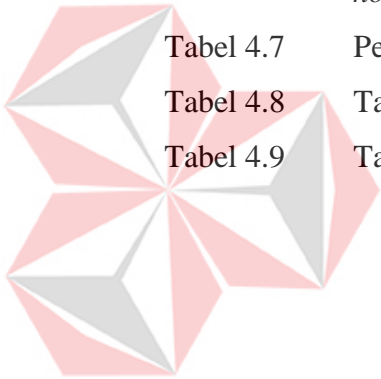
UNIVERSITAS  
**Dinamika**

## DAFTAR GAMBAR

	Halaman
Gambar 2.1 NodeMCU ESP8266 .....	15
Gambar 2.2 LED Matrix P10 .....	16
Gambar 2.3 PCB controller JWS 2.4 .....	17
Gambar 2.4 Power supply 5V .....	18
Gambar 3.1 Blok diagram .....	19
Gambar 3.2 Perancangan <i>hardware</i> .....	20
Gambar 3.3 <i>Flowchart</i> .....	21
Gambar 3.4 Subproses WiFi .....	22
Gambar 3.5 Subproses waktu .....	22
Gambar 3.6 Subproses inisialisasi variabel .....	23
Gambar 3.7 Subproses <i>update</i> waktu .....	24
Gambar 3.8 Subproses <i>update</i> data PPTI .....	25
Gambar 3.9 Subproses <i>running</i> LED .....	26
Gambar 3.10 Desain mekanik (tampak depan) .....	28
Gambar 3.11 Desain mekanik (tampak belakang) .....	29
Gambar 4.1 Data pada URL .....	40

## DAFTAR TABEL

	Halaman
Tabel 4.1	Pengujian ketepatan waktu .....38
Tabel 4.2	Pengujian ketepatan data PPTI.....40
Tabel 4.3	Pengujian waktu proses (pengambilan waktu) menggunakan WiFi kos .....43
Tabel 4.4	Pengujian waktu proses (pengambilan waktu) menggunakan <i>hotspot</i> pribadi.....44
Tabel 4.5	Pengujian waktu proses (pengambilan data PPTI) menggunakan WiFi kos.....46
Tabel 4.6	Pengujian waktu proses (pengambilan data PPTI) menggunakan <i>hotspot</i> pribadi .....48
Tabel 4.7	Pengujian seluruh alat.....50
Tabel 4.8	Tabel pengujian panjang karakter pada LED Matrix P10.....53
Tabel 4.9	Tabel nama beserta koneksinya.....54



UNIVERSITAS  
Dinamika

## DAFTAR LAMPIRAN

	Halaman
Lampiran 1 Hasil Tampilan LED P10 Dosen .....	41
Lampiran 2 Program Keseluruhan Mikrokontroller .....	54
Lampiran 3 Program Pengujian Panjang Karakter .....	59
Lampiran 4 Program Pengujian Waktu Proses (Pengambilan Waktu).....	61
Lampiran 5 Program Pengujian Waktu Proses (Pengambilan Data PPTI).....	63
Lampiran 6 Hasil Turnitin.....	66



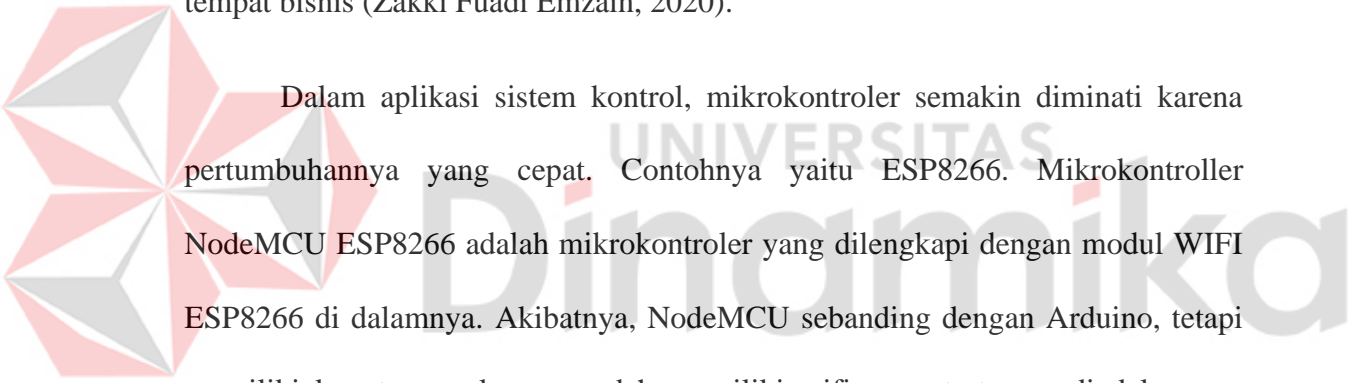
UNIVERSITAS  
**Dinamika**

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Selama dekade terakhir, perkembangan teknologi sangat pesat. Hampir dalam kehidupan sehari-hari, penggunaan teknologi sudah tidak bisa dipisahkan lagi. Kemajuan ini berdampak besar pada semua bidang pendidikan, ekonomi dan masyarakat. Contoh teknologi yang dihasilkan dan sangat berguna diantaranya adalah *running text* LED yang dapat dimanfaatkan sebagai media penyampaian informasi elektronik, bisa juga sebagai media iklan, promosi di depan toko dan tempat bisnis (Zakki Fuadi Emzain, 2020).

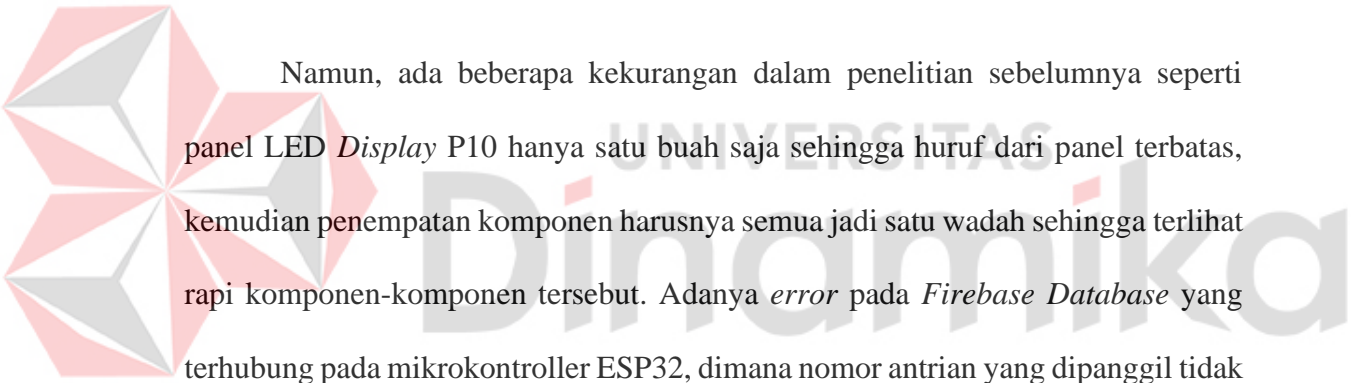


Dalam aplikasi sistem kontrol, mikrokontroler semakin diminati karena pertumbuhannya yang cepat. Contohnya yaitu ESP8266. Mikrokontroler NodeMCU ESP8266 adalah mikrokontroler yang dilengkapi dengan modul WIFI ESP8266 di dalamnya. Akibatnya, NodeMCU sebanding dengan Arduino, tetapi memiliki keuntungan karena sudah memiliki wifi yang tertanam di dalamnya, meskipun portnya lebih sedikit daripada Arduino. NodeMCU atau mikrokontroler ini bisa mendukung untuk pemrograman pada Arduino IDE (*Integrated Development Environment*). (Dendy Ramdani, 2020).

Adapun penelitian instrument antrian menggunakan mikrokontroler ESP32 yang ditampilkan pada *Display* P10 LED. Pertama-tama, klien aplikasi memilih jenis antrian yang diambil akan menentukan jenis layanan yang diperoleh. Data nomor antrian akan diambil oleh aplikasi sistem antrian dari database Firebase yang baru dibuat. Data yang diambil akan dikirim kembali dan ditambahkan sehingga



pengguna berikutnya menerima antrian yang sesuai. Kemudian, saat perhitungan sistem yang melayani data untuk nomor antrian yang dipanggil akan ditampilkan pada P10 LED *Display*. Sementara itu, panggilan akan terdengar melalui *text-to-speech* dari smartphone pengguna, dan data nomor antrean panggilan serta loket pelayanan akan ditampilkan di smartphone pengguna. Jika nomor antrean terlewatkan, pengguna juga akan mendapat opsi untuk mengambil nomor antrian baru melalui notifikasi. (I Made Agus Chandra Wijaya, 2022). Dalam penelitian ini menggunakan perhitungan sistem pemanggilan dan akan dikirimkan oleh mikrokontroler ESP32 sebagai respon dari tombol yang ditekan ke *Firestore Database*.



Namun, ada beberapa kekurangan dalam penelitian sebelumnya seperti panel LED *Display* P10 hanya satu buah saja sehingga huruf dari panel terbatas, kemudian penempatan komponen harusnya semua jadi satu wadah sehingga terlihat rapi komponen-komponen tersebut. Adanya *error* pada *Firestore Database* yang terhubung pada mikrokontroler ESP32, dimana nomor antrian yang dipanggil tidak berubah. Hal ini disebabkan oleh koneksi internet yang tidak bagus sehingga proses pengambilan data dari *Firestore Database* tidak berubah. Fitur *text-to-speech* juga dapat mengalami pengulangan beberapa kali jika terjadi kendala koneksi internet yang lambat.

Oleh karena itu penelitian ini membuat, *RUNNING LED DISPLAY UNTUK MENAMPILKAN INFORMASI KEDATANGAN DOSEN* yang dapat memanfaatkan mikrokontroler dan sensor yang ada, yang tidak menggunakan mikrokontroler ESP32 tetapi menggunakan NodeMCU ESP8266 V3 yang dimana tidak jauh beda dengan ESP32, kemudian 2 buah panel *Display* LED P10 yang lebih

dari cukup membuat karakter atau huruf bisa lebih luas, dan yang paling penting menggunakan adanya bantuan URL yang diberikan dari kampus sendiri yang disebut pihak PPTI (Maulana, 2022). Dari pihak PPTI akan menjadi akses pengambilan data penelitian ini, dimana pihak PPTI sebagai input sebuah data yang nantinya akan ditampilkan pada *Display* LED P10.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang diatas, maka dapat dirumuskan masalah pada Tugas Akhir ini sebagai berikut :

1. Bagaimana cara membuat alat *Display* LED dengan menggunakan mikrokontroller?
2. Bagaimana mengimplementasikan koneksi HTTPS pada ESP8266 untuk mengambil data mengenai kedatangan dosen?
3. Bagaimana cara memperbarui informasi kedatangan dosen secara *real-time* pada *Running* LED *Display* dengan menggunakan ESP8266?

## 1.3 Batasan Masalah

Dalam pembuatan Tugas Akhir ini, pembahasan masalah dibatasi pada beberapa hal berikut :

1. Pengujian dilakukan berdasarkan URL kampus yang dihubungkan ke ESP8266.
2. Alat ini dibuat 2 buah *display* LED yang dijadikan satu.
3. Kehadiran Dosen hanya Prodi S1 Teknik Komputer.

## 1.4 Tujuan

Berdasarkan latar belakang dan rumusan masalah diatas, mendapatkan tujuan pada tugas akhir ini sebagai berikut :

- 1 Merancang dan mengimplementasikan sistem *Running LED Display* yang dapat secara *real-time* menampilkan informasi kedatangan dosen.
- 2 Menghubungkan *Running LED Display* dengan ESP8266 untuk memungkinkan pengambilan data mengenai kedatangan dosen melalui koneksi HTTPS.
- 3 Mengolah data yang diperoleh melalui koneksi HTTPS dan mengatur tampilan informasi dengan efektif pada *Running LED Display*.

## 1.5 Manfaat

Adapun dari Tugas Akhir ini dapat diperoleh manfaat sebagai berikut :

- 1 Memudahkan informasi kehadiran dosen yang berada di kampus dan di ruangan atau tidak.
- 2 Layaknya alat ini untuk diimplementasikan dan efisien sebagai media penyampaian informasi.

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 PPTI Universitas Dinamika**

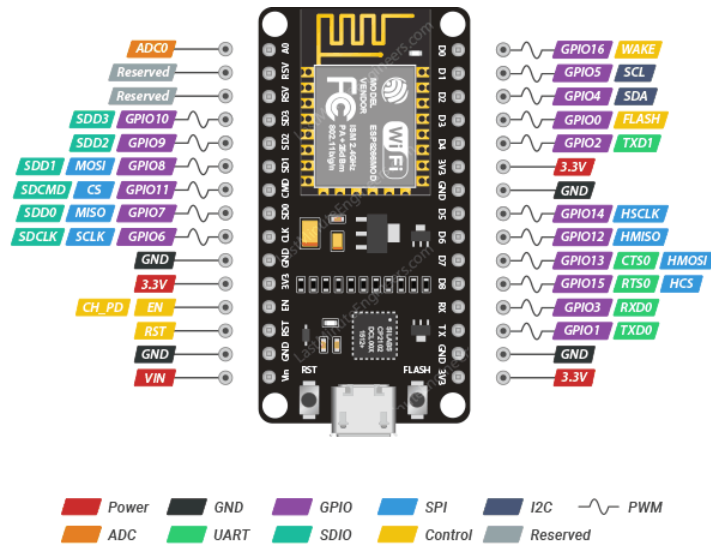
Unit Kerja Pengembangan dan Penerapan Teknologi Informasi (PPTI) adalah salah satu unit kerja di Universitas Dinamika Surabaya yang menyediakan fasilitas dan layanan berbasis teknologi informasi untuk dosen, karyawan, dan mahasiswa. Layanan ini ditujukan untuk dapat menunjang aktivitas kegiatan belajar mengajar dari segenap komponen Universitas Dinamika Surabaya dengan memanfaatkan perangkat teknologi informasi serta melakukan penerapan teknologi baru untuk meningkatkan efektifitas pekerjaan.

#### **2.2 NodeMCU ESP8266**

NodeMCU merupakan sebuah Platform Internet of Things (IoT) bersifat opensource. Terdiri dari system on Chip (SoC) ESP8266 buatan Espressif System, menggunakan bahasa pemrograman scripting Lua, bersifat open source (Septian Yogi, 2021). NodeMCU harus terintegrasi pada internet agar data yang di dapat agar menampilkan inputan yang dapat di ubah menjadi data dan informasi.

NodeMCU juga memiliki beberapa port, antara lain :

1. 10 Port GPIO dari D0-D10
2. Pin PWM
3. Antarmuka I2C dan SPI
4. Antarmuka 1 Wire
5. Pin ADC.



ESP8266 NodeMCU Pinout



Gambar 2.1 NodeMCU ESP8266

(lastminuteengineers/getting-started-with-esp8266, n.d.)

### 2.3 Display LED Matrix P10

LED Matrix P10 yaitu deretan LED yang membentuk kolom dan baris dengan jumlah tertentu. LED Matrix P10 adalah sebuah susunan LED yang dirancang dengan ukuran 16x32cm yang dapat digunakan untuk menampilkan suatu teks. Sehingga membentuk titik-titik LED yang menyala dapat membentuk karakter berupa huruf, angka dan tanda baca dengan efek animasi tertentu (Nasution & Daud, 2023). Pada LED Matrix P10 ini dapat disambungkan dengan LED Matrix P10 lainnya dengan rangkain- paralel dan seri. Pada LED Matrix P10 dapat menggunakan tegangan masukkan dari power supply maupun dari Arduino langsung yang terhubung dengan PC (*personal computer*). Penggunaan LED matrix P10 ini banyak digunakan pada running text. (Malik Yuhanas, 2021) . Jenis *display* led matrix antara lain : P10, P4.75, P76.2, P13.33 dll. Adapun perbedaan antara jarak dot/led misal P10 : 10cm dan P4.75 : 4,75 cm.



Gambar 2.2 LED Matrix P10  
(Ledcontrolcard, n.d.).

## 2.4 Arduino IDE

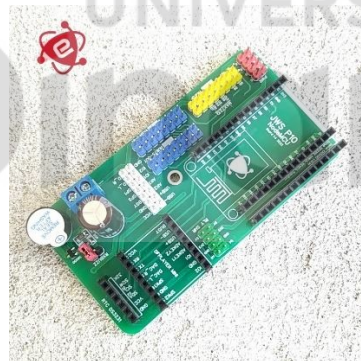
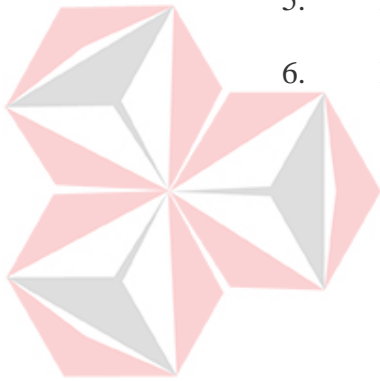
Arduino IDE itu merupakan kependekan dari *Integrated Development Environment*, atau secara bahasa mudahnya merupakan lingkungan terintegrasi yang digunakan untuk melakukan pengembangan. Disebut sebagai lingkungan karena melalui *software* inilah Arduino dilakukan pemrograman untuk menggunakan fungsi-fungsi yang dinamakan melalui sintaks pemrograman (IDE, n.d.).

Arduino IDE ini dikembangkan dari *software* Processing yang dirombak menjadi Arduino IDE khusus untuk pemrograman dengan Arduino IDE (*Integrated Development Environment*) berarti bentuk alat pengembangan program yang terintegrasi sehingga berbagai keperluan disediakan dan dinyatakan dalam bentuk antarmuka berbasis menu (Agus Setiawan, 2021).

## 2.5 PCB Controller JWS 2.4

PCB *controller* JWS merupakan controller yang menghubungkan antara NodeMCU ESP8266 dengan mudah pada *display* LED Matrix P10 sehingga tidak harus memakai kabel female to male, male to male, dsb. Adapun spesifikasi (Fungsional Port and Konektor) sesuai gambar komponen:

1. Ukuran: 93mm x 49mm
2. Ketebalan PCB: 1.6
3. Layers : 2
4. Warna PCB: Green
5. Berat Tembaga: 1
6. Detail Bahan: FR4-Standard Tg 130-140C



Gambar 2.3 PCB controller JWS 2.4  
(Tokopedia/elektronmartcom, n.d.)

## 2.6 Power Supply 5V

Power Supply atau dalam bahasa Indonesia disebut dengan Catu Daya adalah suatu alat listrik yang dapat menyediakan energi listrik untuk perangkat listrik ataupun elektronika lainnya. Suatu jenis catu daya yang dikenal sebagai *Switch-Mode Power Supply* (SMPS) mengubah tegangan input AC menjadi

tegangan DC dengan memperbaiki dan memfilternya. Untuk menghasilkan arus AC yang dapat melewati Trafo Frekuensi Tinggi, tegangan DC kemudian dinyalakan dan dimatikan pada frekuensi tinggi menggunakan rangkaian frekuensi tinggi. (Trisetiyanto, 2020).



Gambar 2.4 Power supply 5V

(su.eager-led, n.d.).

## 2.7 HTTPS

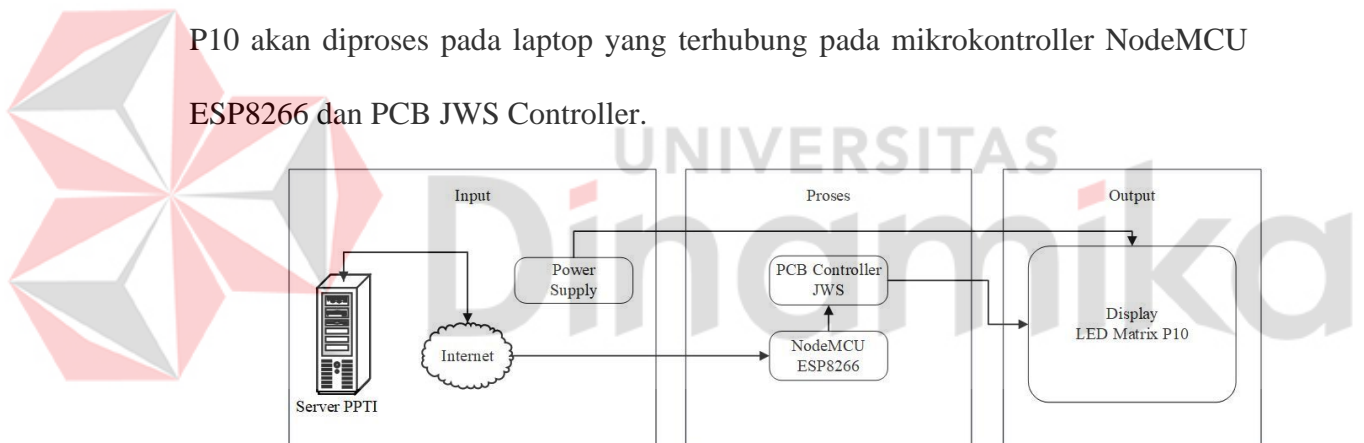
HTTP dan HTTPS memiliki kepentingan yang hampir sama, yang penting adalah kemampuan keamanan (*secure*). Pemanfaatan kemampuan SSL (*Secure Attachments Layer*) atau TLS (*Transport Layer Security*) pada sublapisan di bawah lapisan aplikasi HTTP standar. Selama proses komunikasi antara pengguna dan server web atau sebaliknya, HTTPS memiliki teknologi untuk mencegah informasi penting "dicuri". Pada umumnya website yang menggunakan HTTPS menggunakan enkripsi SSL untuk mengenkripsi data (informasi). Oleh karena itu, meskipun seseorang telah berhasil "mengambil" informasi saat dalam perjalanan, klien server web tidak dapat membacanya karena telah diubah oleh strategi enkripsi SSL.



## BAB III METODOLOGI PENELITIAN

### 3.1 Blok Diagram

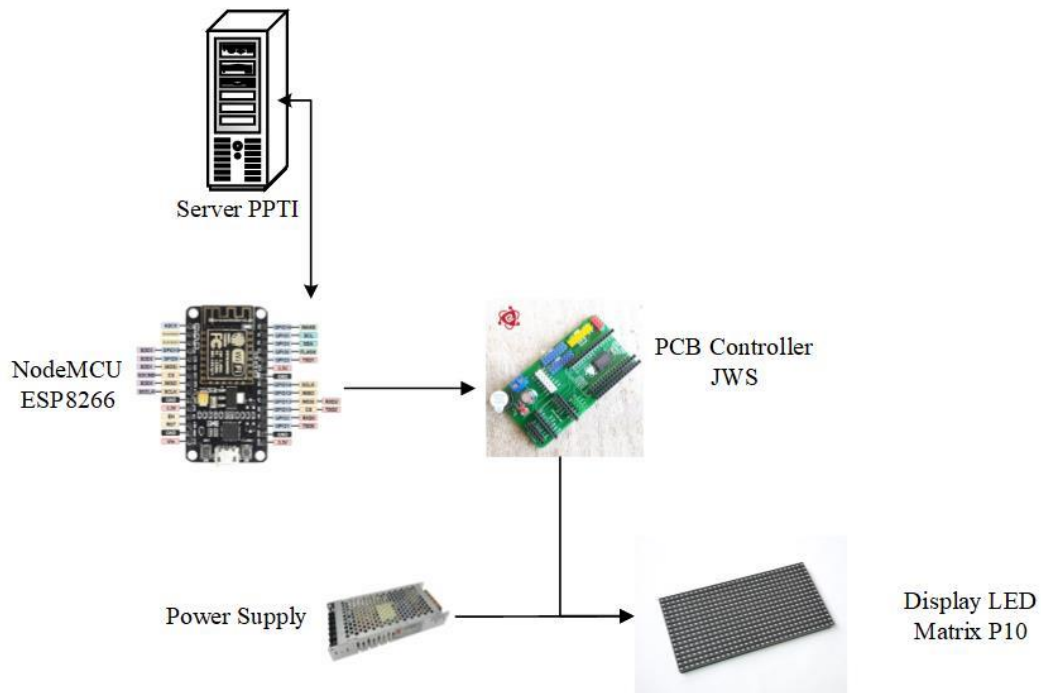
Pada Gambar 3.1 dibawah dapat dilihat bahwa alat ini menggunakan sumber tegangan 220V sebagai input power supply 5V DC (Seftiana, Najeri, Anggono, & Thyo Priandika, 2021), selain itu tegangan 220V juga digunakan input *Display* LED Matrix P10. Pada sistem ini mikrokontroler menggunakan NodeMCU ESP8266 berfungsi sebagai pusat pengolah data dan mengkomunikasikan data kepada *Server* PPTI yang terhubung ke internet. Kemudian tampilan pada *display* LED Matrix P10 akan diproses pada laptop yang terhubung pada mikrokontroler NodeMCU ESP8266 dan PCB JWS Controller.



Gambar 3.1 Blok diagram

### 3.2 Perancangan Perangkat Keras

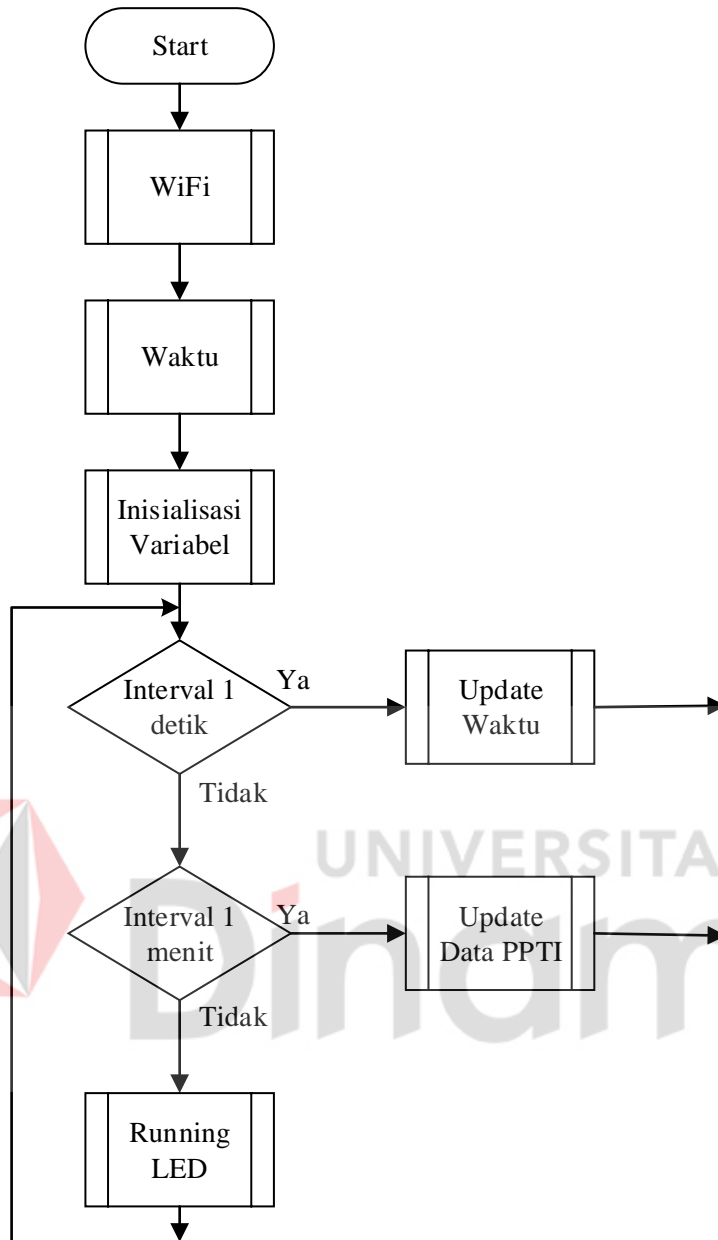
Pada Gambar 3.2 di bawah dapat dilihat pada bagian input yaitu *server* PPTI, yang nantinya data pada server PPTI akan menjadi input bagi mikrokontroler. Kemudian NodeMCU ESP8266 terhubung pada PCB Controller JWS dan diteruskan ke *Display* LED Matrix P10 (Bahtiar, 2022), sebagai outputnya dan mendapat input atau tegangan dari power supply.



Gambar 3.2 Perancangan *hardware*

### 3.3 Flowchart

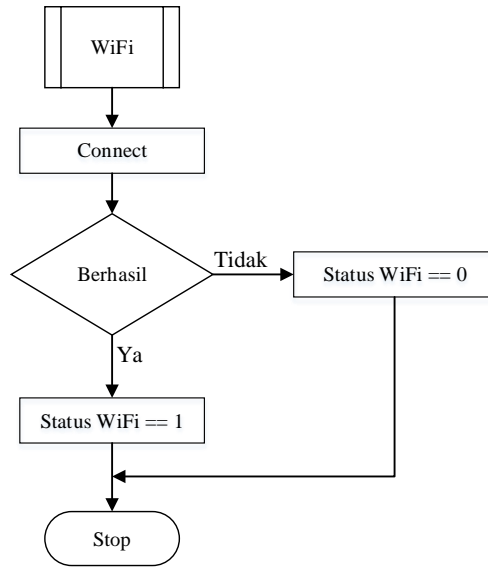
*Flowchart* sebagaimana tampak pada Gambar 3.3 di bawah merupakan proses seluruh alat yang telah dibuat. Pertama void setup, jika status wifi terhubung lanjut akses ke pengambilan waktu dari internet (Ismail, 2020) menggunakan mikrokontroller yang sudah terhubung ke wifi, tidak lupa inisialisasi variabel yang telah dibuat. Terakhir void loop yang berisi 2 kondisi, yang pertama interval 1 detik jika “Ya” maka update waktu akan dilakukan selama 1 detik dan jika “Tidak” lanjut yang kedua, yaitu interval 3 menit jika “Ya” maka update data dari PPTI selama 1 menit dan jika “Tidak” maka lanjut Running LED yang berisi tampilan, pertama Waktu selama 10 detik, kedua Identitas (Nama Prodi, Fakultas, Nama Kampus) selama 20 detik, terakhir nama Dosen dan status kehadiran selama 40 detik.



Gambar 3.3 Flowchart

### 3.4 Subproses WiFi

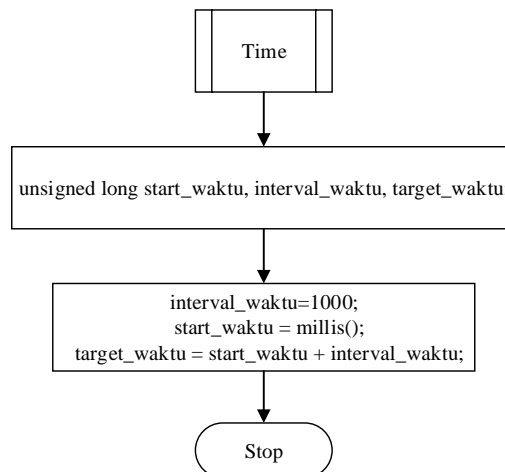
Flowchart Subproses Wifi terlihat pada Gambar 3.4 di bawah jika *Connect* kemudian sudah di atur pada SSID dan Password yang mau disambungkan maka *Connect*, ketika berhasil ada dua kondisi jika “Ya” status wifi ==1 dan jika “Tidak” status wifi == 0.



Gambar 3.4 Subproses WiFi

### 3.5 Subproses Waktu

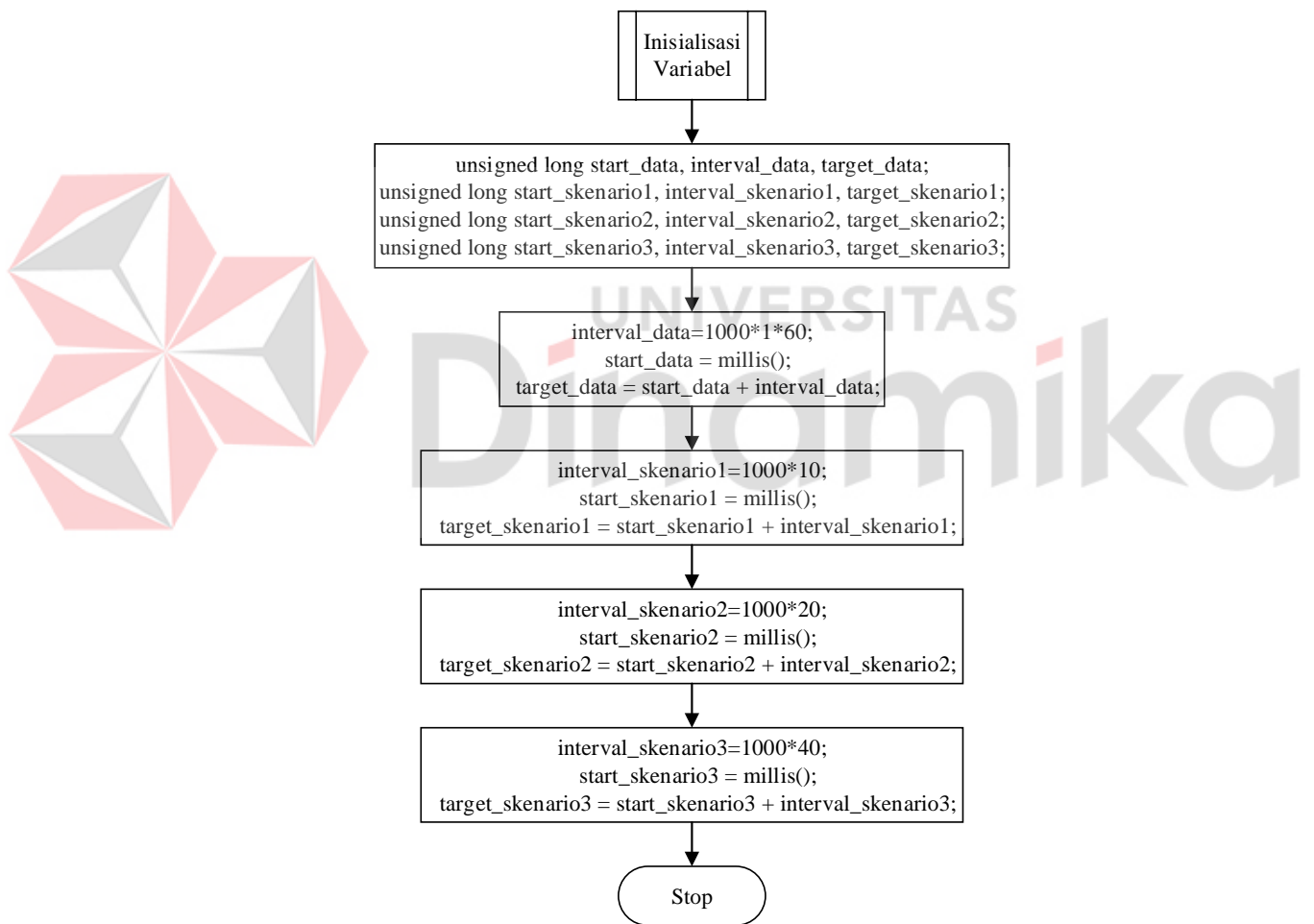
Terdapat tipe data pada *flowchart* subproses waktu terlihat pada Gambar 3.5 di bawah yaitu *unsigned long* digunakan untuk penghitungan jangka waktu yang panjang, kemudian proses selanjutnya variabel `interval_waktu` dengan nilai 1000, memakai fungsi `millis()`, dan bertujuan untuk mendapatkan `target_waktu` lebih maju ketika `interval_waktu` yang ditentukan telah berlalu.



Gambar 3.5 Subproses waktu

### 3.6 Subproses Inisialisasi Variabel

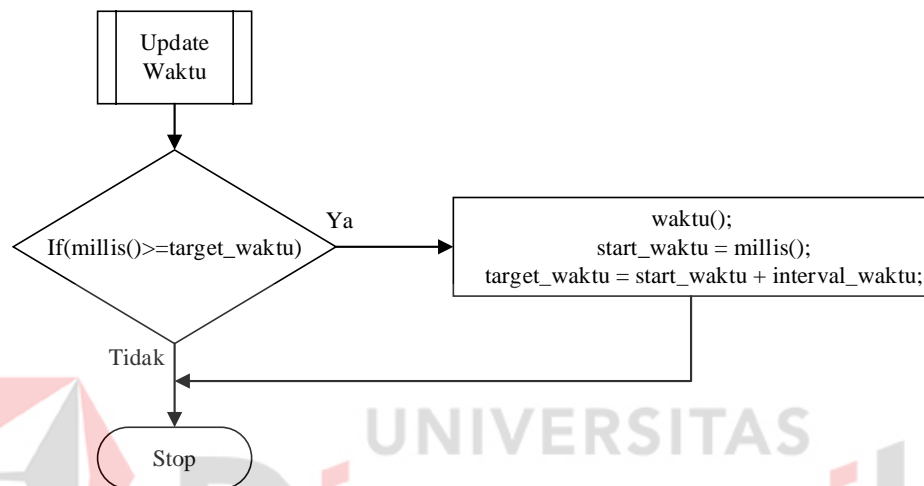
Flowchart pada Gambar 3.6 di bawah merupakan subproses dari inisialisasi variabel, terdapat variabel global menggunakan *unsigned long* pengambilan data dan skenario 1,2, dan 3. Kemudian untuk melakukan setup terdapat 4 proses yang berbeda, pertama `interval_data` bernilai  $1000*1*60$ , kedua `interval_skenario1` bernilai  $1000*10$ , ketiga `interval_skenario2` bernilai  $1000*20$ , dan keempat `interval_skenario2` bernilai  $1000*40$ . Masing-masing memakai fungsi `millis()` dan bertujuan mendapatkan target yang sama yaitu lebih maju, ketika interval dari 4 proses tersebut yang telah ditentukan telah berlalu.



Gambar 3.6 Subproses inisialisasi variabel

### 3.7 Subproses Update Waktu

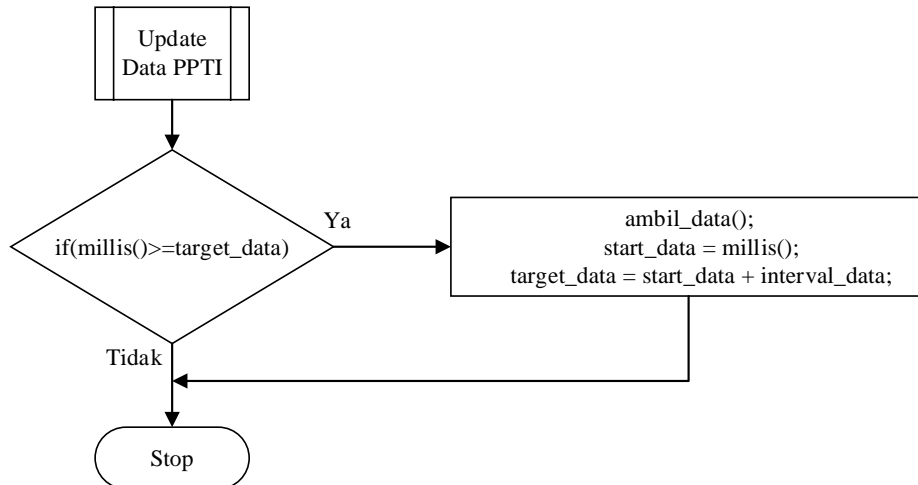
Flowchart pada Gambar 3.7 di bawah merupakan subproses update waktu yang mempunyai *actions* jika “Ya” maka langkah – langkah yang dijalankan adalah memanggil fungsi waktu() untuk melakukan tindakan tertentu, kemudian mengupdate nilai pada start\_waktu dengan waktu saat ini menggunakan millis(), lalu mengupdate target\_waktu dengan menambahkan start\_waktu dan interval\_waktu. Jika “Tidak” maka langsung berhenti atau stop.



Gambar 3.7 Subproses update waktu

### 3.8 Subproses Update Data PPTI

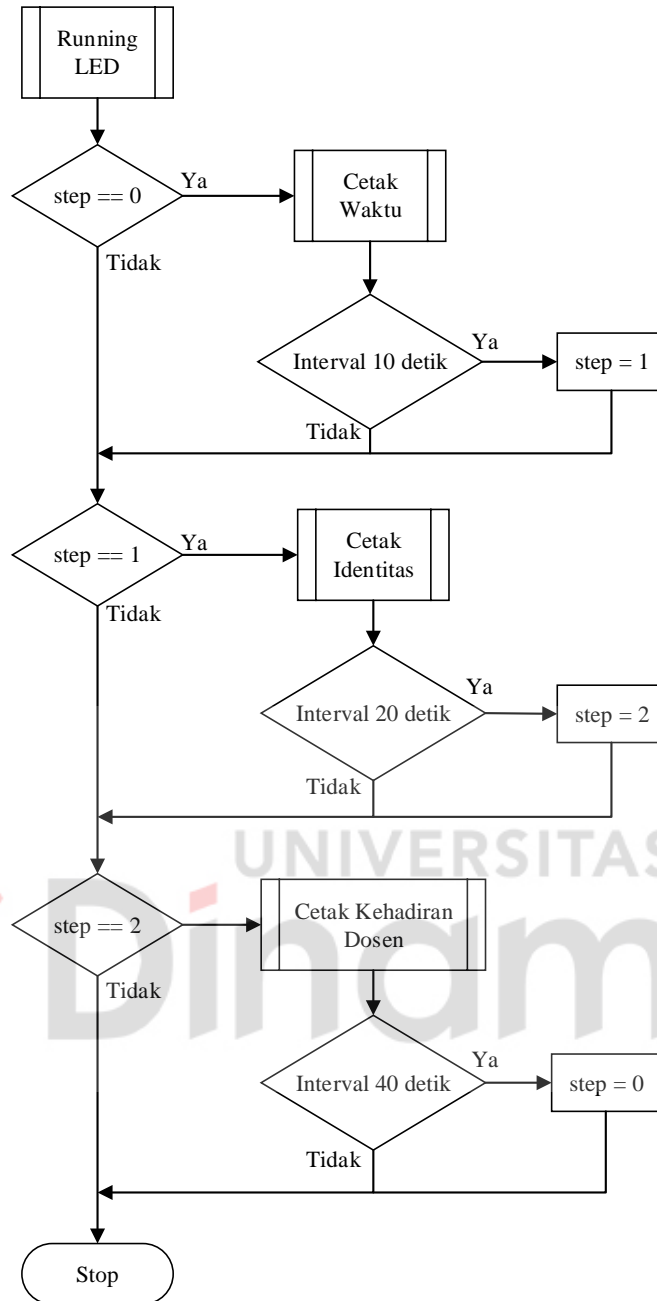
Flowchart pada Gambar 3.8 di bawah merupakan subproses update data PPTI hampir sama dengan subproses update waktu yang membedakan hanyalah pemanggilan fungsi ambil\_data() untuk melakukan tindakan tertentu, kemudian mengupdate nilai pada start\_data dengan waktu saat ini menggunakan millis(), lalu mengupdate target\_data dengan menambahkan start\_data dan interval\_data. Kemudian jika “Tidak” maka langsung berhenti atau stop.



Gambar 3.8 Subproses *update* data PPTI

### 3.9 Subproses Running LED

*Flowchart* pada Gambar 3.9 di bawah merupakan proses dicetaknya skenario 1 yang berisi cetak waktu selama 10 detik, skenario 2 cetak identitas (nama prodi, fakultas, dan kampus) selama 20 detik, dan terakhir mencetak kehadiran dosen selama 40 detik. Proses ini bisa berulang dengan adanya tambahan kondisi step.



Gambar 3.9 Subproses *running* LED

### 3.10 Mengambil Data Dari *Link* URL yang Telah Diberikan oleh PPTI

#### 3.10.1 Mengakses *Link* URL dengan Get HTTP

Pertama mengabaikan proses verifikasi keaslian dan keabsahan sertifikat SSL saat terjadi koneksi aman antara pengguna dan server :

```
client->setInsecure();
```



Setelah itu membuat permintaan HTTP ke server melalui sambungan aman menggunakan objek klien HTTP (Ramadhanu & Priandika, 2021), yang telah diinisialisasi dengan protokol HTTPS:

```
HTTPClient https;
```

Menginisialisasi koneksi HTTP dengan menggunakan objek `https` yang mewakili klien HTTP yang menggunakan protokol HTTPS (Oktaviani & Ayu, 2021), URL yang diberikan sebagai berikut :

```
if (https.begin(*client, "https://e-sicyca.dinamika.ac.id/api/today_attendance?prodi=41020")) { // HTTPS
  // Mulai koneksi dan kirim HTTP header
  int httpCode = https.GET();
```

tujuannya adalah untuk membuat koneksi HTTPS ke endpoint API "https://e-sicyca.dinamika.ac.id/api/today\_attendance?prodi=41020" dan melakukan permintaan GET untuk mengambil data terkait hari ini (Hendrastuty & Ihza, 2021).

### 3.10.2 Mengambil data Nama dan Kehadiran saja pada URL

Proses pertama yaitu dengan mengambil nilai yang terkait dengan kunci "NAMA" dan "IS\_HADIR" dengan mengetik :

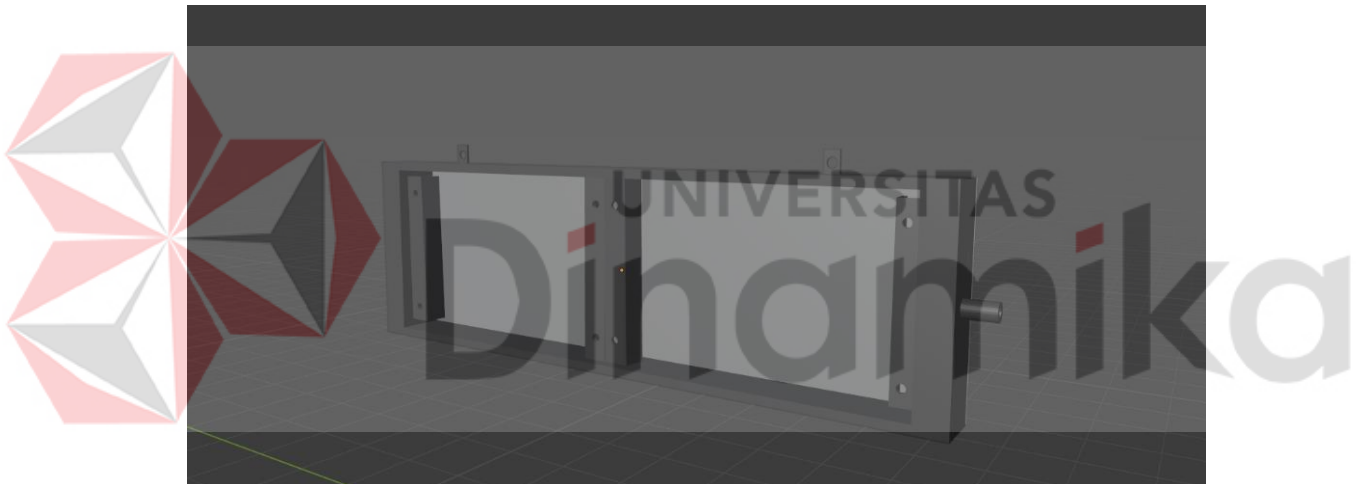
```
const char* nama = attendance["NAMA"];
const char* status_kehadiran = attendance["IS_HADIR"];
```

Pengambilan nilai dengan kunci "NAMA" dari objek `attendance` dan menyimpannya dalam variabel `const char* nama`, menunjukkan bahwa variabel `nama` adalah pointer ke sebuah string yang tidak dapat diubah (`const`). Kemudian dengan kunci "IS\_HADIR" dari objek `attendance` dan menyimpannya dalam variabel `const char* status_kehadiran`, seperti sebelumnya, `const char*` menunjukkan bahwa variabel `status_kehadiran` adalah pointer ke sebuah string yang tidak dapat diubah.

### 3.11 Desain Mekanik

Pada Gambar 3.10 di bawah merupakan contoh gambar desain tempat atau wadah bagi *Display* LED Matrix P10, berikut ukurannya :

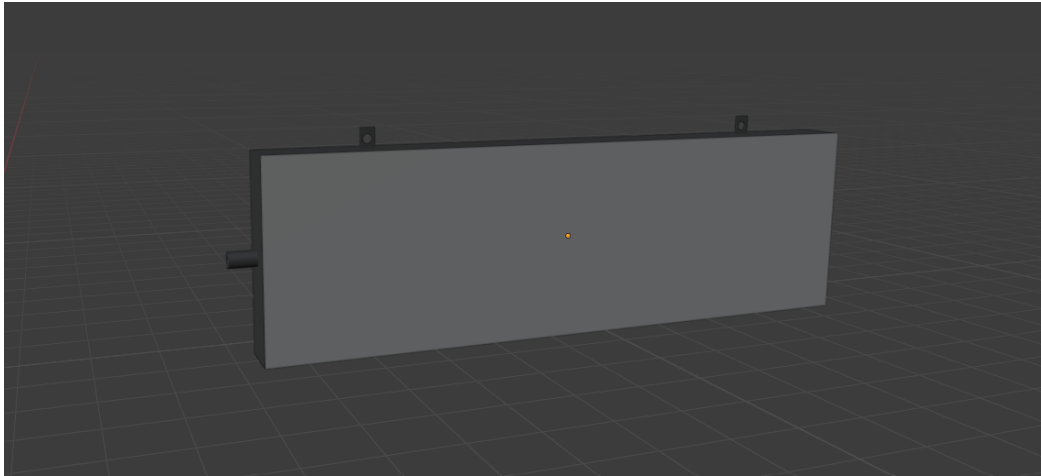
1. Frame running text 2 Panel *Display* LED Matrix P10 berukuran 16x64 cm sehingga diameter ukuran dalam yaitu 16x64 cm.
2. Diameter luar ukurannya panjang 69cm, lebar 5 cm, dan tingginya 16cm.
3. Terdapat 4 pangkon penyangga, berfungsi sebagai penyangga 2 panel *Display* LED Matrix P10.



Gambar 3.10 Desain mekanik (tampak depan)

Pada Gambar 3.11 di bawah terdapat penampakan bagian belakang desain dan penjelasan sebagai berikut :

1. Lubang pada kiri desain(tampak belakang) merupakan tempat kabel yang nantinya sebagai tempat masuk dan keluarnya kabel pada komponen.
2. Kemudian tempat power supply berada dibelakang dengan komponen komponen lainnya.



Gambar 3.11 Desain mekanik (tampak belakang)



UNIVERSITAS  
**Dinamika**

## **BAB IV**

### **HASIL DAN PEMBAHASAN**

Pada Bab 4 ini membahas pengujian dan analisis, dimulai dari waktu yang diambil pada internet, data dari url yang telah diberikan, dan tampilan pada LED P10. Pada tahap pengujian pengambilan data dan waktu ini memastikan bahwa proses berjalan dengan baik. Lalu pada tahap pengujian tampilan pada LED P10 memastikan kemampuan komponen alat yang digunakan berjalan sesuai dengan yang telah dibuat.

#### **4.1 Pengujian Ketepatan Waktu**

##### **4.1.1 Tujuan**

Menguji ketepatan pengambilan waktu pada tampilan LED Matrix P10 dengan membandingkan smartphone sesuai atau belum sesuai.

##### **4.1.2 Alat dan Bahan**

1. Laptop
2. ESP8266
3. LED Matrix P10
4. Smartphone
5. Power Supply 5V

##### **4.1.3 Prosedur**

Menyalakan laptop kemudian menggunakan program (lampiran 4) yang telah di upload pada Arduino IDE dengan menghubungkan wifi agar dapat tersambung pada internet, kemudian hasil bisa terlihat pada serial monitor dengan pengujian sebanyak 30 kali.

Pada pengujian ini hal-hal yang perlu diamati adalah kemunculan informasi waktu pada tampilan LED Matrix dan waktu pada smartphone lalu mencatat selisih antara tampilan pada LED Matrix P10 dengan waktu *Smartphone*, kemudian dihitung rata-rata selisihnya.

Tabel 4.1 Pengujian ketepatan waktu

No.	Waktu LED Matrix P10	Waktu <i>Smartphone</i>	Selisih (detik)
1.	08:10:11	08:10:11	0 detik
2.	08:10:12	08:10:12	0 detik
3.	08:10:13	08:10:13	0 detik
4.	08:10:14	08:10:14	0 detik
5.	08:10:15	08:10:15	0 detik
6.	08:10:16	08:10:16	0 detik
7.	08:10:17	08:10:17	0 detik
8.	08:10:18	08:10:18	0 detik
9.	08:10:19	08:10:19	0 detik
10.	08:10:20	08:10:20	0 detik
11.	09:10:11	09:10:11	0 detik
12.	09:10:12	09:10:12	0 detik
13.	09:10:13	09:10:13	0 detik
14.	09:10:14	09:10:14	0 detik
15.	09:10:15	09:10:15	0 detik
16.	09:10:16	09:10:16	0 detik
17.	09:10:17	09:10:17	0 detik
18.	09:10:18	09:10:18	0 detik
19.	09:10:19	09:10:19	0 detik
20.	09:10:20	09:10:20	0 detik
21.	10:10:11	10:10:11	0 detik
22.	10:10:12	10:10:12	0 detik
23.	10:10:13	10:10:13	0 detik

24.	10:10:14	10:10:14	0 detik
25.	10:10:15	10:10:15	0 detik
26.	10:10:16	10:10:16	0 detik
27.	10:10:17	10:10:17	0 detik
28.	10:10:18	10:10:18	0 detik
29.	10:10:19	10:10:19	0 detik
30.	10:10:20	10:10:20	0 detik
Rata – Rata Selisih			0 detik

#### 4.1.4 Analisis Pengujian Ketepatan Waktu

Pada Tabel 4.1 adanya pengujian percobaan selama 30 kali yang diamati dengan benar mulai dari jam 8 sampai dengan jam 10 sesuai dengan rencana. Terdapat kemiripan pada waktu yang ditampilkan LED Matrix P10 dengan waktu Smartphone. Sehingga selisih pada kedua informasi tersebut yaitu waktu pada tampilan LED Matrix P10 dengan waktu Smartphone bernilai 0. Pengujian ketepatan waktu ini menggunakan jaringan seadanya pada saat itu. Jika jaringan kurang bagus maka informasi waktu pada tampilan LED Matrix P10 hasilnya berubah menjadi acak. Jadi, kesimpulan dari pengujian ini adalah selisih pada kedua informasi tersebut bernilai 0, sehingga rata-rata selisihnya adalah 0 detik.

## 4.2 Pengujian Ketepatan Data PPTI

### 4.2.1 Tujuan

Menguji ketepatan pengambilan data PPTI pada URL yang telah diberikan dan ditampilkan sesuai atau tidaknya.

### 4.2.2 Alat dan Bahan

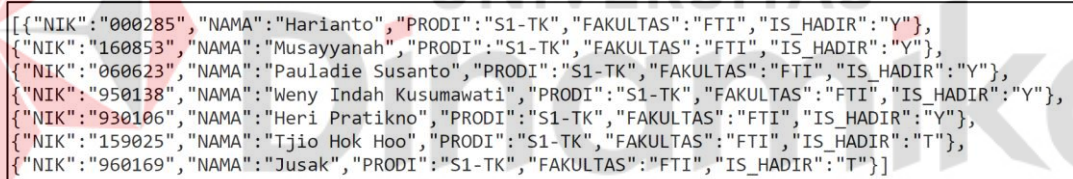
1. Laptop

2. ESP8266
3. LED Matrix P10
4. Power Supply 5V

### 4.2.3 Prosedur

Menyalakan laptop kemudian menggunakan program (lampiran 2) yang telah di upload dan tambahan menggunakan URL yang telah diberikan pada Arduino IDE dengan menghubungkan wifi agar dapat tersambung pada internet, kemudian hasil bisa terlihat pada serial monitor.

Pada Gambar 4.1 di bawah ini merupakan data JSON yang kemudian nantinya akan diambil setiap 1 menit sekali dan ditampilkan pada LED Matrix P10, lalu untuk pengujian ketepatan data PPTI ini akan ada data pengamatan (serial monitor) dan keterangan sebanyak 30 kali.



```
[{"NIK": "000285", "NAMA": "Harianto", "PRODI": "S1-TK", "FAKULTAS": "FTI", "IS_HADIR": "Y"},
{"NIK": "160853", "NAMA": "Musayyanah", "PRODI": "S1-TK", "FAKULTAS": "FTI", "IS_HADIR": "Y"},
{"NIK": "060623", "NAMA": "Pauladie Susanto", "PRODI": "S1-TK", "FAKULTAS": "FTI", "IS_HADIR": "Y"},
{"NIK": "950138", "NAMA": "Weny Indah Kusumawati", "PRODI": "S1-TK", "FAKULTAS": "FTI", "IS_HADIR": "Y"},
{"NIK": "930106", "NAMA": "Heri Pratikno", "PRODI": "S1-TK", "FAKULTAS": "FTI", "IS_HADIR": "Y"},
{"NIK": "159025", "NAMA": "Tjio Hok Hoo", "PRODI": "S1-TK", "FAKULTAS": "FTI", "IS_HADIR": "T"},
{"NIK": "960169", "NAMA": "Jusak", "PRODI": "S1-TK", "FAKULTAS": "FTI", "IS_HADIR": "T"}]
```

Gambar 4.1 Data pada URL

Tabel 4.2 Pengujian ketepatan data PPTI

No.	Data Pengamatan (Serial Monitor)	Keterangan
1.	Sama	Sesuai
2.	Sama	Sesuai
3.	Sama	Sesuai
4.	Sama	Sesuai
5.	Sama	Sesuai
6.	Sama	Sesuai
7.	Sama	Sesuai



8.	Sama	Sesuai
9.	Sama	Sesuai
10.	Sama	Sesuai
11.	Sama	Sesuai
12.	Sama	Sesuai
13.	Sama	Sesuai
14.	Sama	Sesuai
15.	Sama	Sesuai
16.	Sama	Sesuai
17.	Sama	Sesuai
18.	Sama	Sesuai
19.	Sama	Sesuai
20.	Sama	Sesuai
21.	Sama	Sesuai
22.	Sama	Sesuai
23.	Sama	Sesuai
24.	Sama	Sesuai
25.	Sama	Sesuai
26.	Sama	Sesuai
27.	Sama	Sesuai
28.	Sama	Sesuai
29.	Sama	Sesuai
30.	Sama	Sesuai
Rata – Rata Keberhasilan		100%

#### 4.2.4 Analisis Ketepatan Data PPTI

Pada Tabel 4.2 30 kali percobaan telah dilakukan, ketepatan data PPTI dosen selalu muncul dengan tingkat kesesuaian yang sama. Pada pengamatan (serial monitor) telah diprogram sedemikian rupa agar sesuai dengan rencana. Pengujian pernah dilakukan menggunakan jaringan yang kualitasnya rendah, hasilnya



terdapat kedipan yang terus menerus sehingga mengganggu tampilan LED Matrix P10.

$$\text{Tingkat kesesuaian} = \frac{\sum \text{Sesuai}}{\sum \text{Uji}} \times 100\% \quad (1)$$

Jadi, kesimpulan dari pengujian ini adalah seluruh informasi sesuai dengan rencana dari pengamatan (serial monitor) dan keterangan yang telah diperoleh. Namun ada beberapa masalah jika jaringan tersebut kurang bagus, akan ada kedipan yang terjadi setiap melakukan aktifitas pengambilan data PPTI. Kemudian, tingkat kesesuaian pada ketepatan data PPTI memperoleh rata-rata 100%.

### **4.3 Pengujian Waktu Proses (Pengambilan Waktu)**

#### **4.3.1 Tujuan**

Menguji waktu proses (pengambilan waktu) yang diambil pada internet dengan dua jaringan yang berbeda.

#### **4.3.2 Alat dan Bahan**

1. Laptop
2. ESP8266
3. LED Matrix P10
4. Power Supply 5V
5. *Hotspot*

#### **4.3.3 Prosedur**

Menyalakan laptop kemudian menggunakan program (lampiran 4) yang telah di upload pada Arduino IDE dengan menghubungkan wifi agar dapat tersambung pada internet, kemudian hasil bisa terlihat pada serial monitor dengan menyalakan timestampnya.

Pada program Arduino IDE nantinya akan ada start dan stop dibagian ambil waktu, tentunya dikasih delay bernilai 1000, jadi 2 baris pertama akan dihitung sesuai prosedur yaitu waktu akhir dikurangi waktu awal selama 30 kali percobaan, dan percobaan kali ini menggunakan 2 jaringan yang berbeda.

Tabel 4.3 Pengujian waktu proses (pengambilan waktu) menggunakan WiFi kos

No.	Waktu Awal	Waktu Akhir	Waktu Proses
1.	10:39:35.500	10:39:35.500	0 detik
2.	10:39:36.488	10:39:36.488	0 detik
3.	10:39:37.475	10:39:37.475	0 detik
4.	10:39:38.468	10:39:38.468	0 detik
5.	10:39:39.495	10:39:39.495	0 detik
6.	10:39:40.495	10:39:40.495	0 detik
7.	10:39:41.473	10:39:41.473	0 detik
8.	10:39:42.499	10:39:42.499	0 detik
9.	10:39:43.482	10:39:43.482	0 detik
10.	10:39:44.501	10:39:44.501	0 detik
11.	10:39:45.481	10:39:45.481	0 detik
12.	10:39:46.463	10:39:46.463	0 detik
13.	10:39:47.465	10:39:47.465	0 detik
14.	10:39:48.483	10:39:48.483	0 detik
15.	10:39:49.472	10:39:49.472	0 detik
16.	10:39:50.494	10:39:50.494	0 detik
17.	10:39:50.494	10:39:50.494	0 detik
18.	10:39:51.493	10:39:51.493	0 detik
19.	10:39:52.472	10:39:52.472	0 detik
20.	10:39:53.495	10:39:53.495	0 detik
21.	10:39:54.483	10:39:54.483	0 detik
22.	10:39:55.503	10:39:55.503	0 detik
23.	10:39:56.481	10:39:56.481	0 detik
24.	10:39:57.503	10:39:57.503	0 detik

25.	10:39:58.491	10:39:58.491	0 detik
26.	10:39:59.513	10:39:59.513	0 detik
27.	10:40:00.487	10:40:00.487	0 detik
28.	10:40:01.514	10:40:01.514	0 detik
29.	10:40:02.497	10:40:02.497	0 detik
30.	10:40:03.481	10:40:03.481	0 detik
Rata – Rata Waktu Proses			0 detik

Tabel 4.4 Pengujian waktu proses (pengambilan waktu) menggunakan *hotspot* pribadi

No.	Waktu Awal	Waktu Akhir	Waktu Proses
1.	11:13:50.308	11:13:50.308	0 detik
2.	11:13:51.282	11:13:51.282	0 detik
3.	11:13:52.293	11:13:52.293	0 detik
4.	11:13:53.292	11:13:53.292	0 detik
5.	11:13:54.300	11:13:54.300	0 detik
6.	11:13:55.305	11:13:55.305	0 detik
7.	11:13:56.287	11:13:56.287	0 detik
8.	11:13:57.304	11:13:57.304	0 detik
9.	11:13:58.306	11:13:58.306	0 detik
10.	11:13:59.306	11:13:59.306	0 detik
11.	11:14:00.304	11:14:00.304	0 detik
12.	11:14:01.301	11:14:01.301	0 detik
13.	11:14:02.303	11:14:02.303	0 detik
14.	11:14:03.298	11:14:03.298	0 detik
15.	11:14:04.293	11:14:04.293	0 detik
16.	11:14:05.311	11:14:05.311	0 detik
17.	11:14:06.296	11:14:06.297	0,001 detik
18.	11:14:07.299	11:14:07.299	0 detik
19.	11:14:08.319	11:14:08.319	0 detik

20.	11:14:09.310	11:14:09.310	0 detik
21.	11:14:10.307	11:14:10.307	0 detik
22.	11:14:11.304	11:14:11.304	0 detik
23.	11:14:12.283	11:14:12.284	0,001 detik
24.	11:14:13.311	11:14:13.311	0 detik
25.	11:14:14.306	11:14:14.306	0 detik
26.	11:14:15.319	11:14:15.319	0 detik
27.	11:14:16.315	11:14:16.315	0 detik
28.	11:14:17.292	11:14:17.292	0 detik
29.	11:14:18.313	11:14:18.313	0 detik
30.	11:14:19.313	11:14:19.313	0 detik
Rata – Rata Waktu Proses			0,00006 detik

#### 4.3.4 Analisis Pengujian Waktu Proses (Pengambilan Waktu)

Pada Tabel 4.3 dan Tabel 4.4 setelah melakukan 30 kali percobaan dengan menggunakan jaringan yang berbeda, untuk informasi waktu proses di jaringan pertama yaitu menggunakan WiFi kos sesuai dengan rencana sampai waktu proses bernilai 0 detik. Pada kasus dimana menggunakan jaringan *hotspot* pribadi mendapatkan waktu proses yang belum sama tetapi hanya kemungkinan kecil yaitu 0,00006 detik saja. Pengambilan waktu proses ini memerlukan waktu tunggu yang tidak dapat dikendalikan, karena bergantung kepada kualitas mikrokontroler juga.

Jadi, kesimpulan dari pengujian ini adalah didapatkan total percobaan 30 kali dari jaringan yang berbeda, rata-rata waktu proses menggunakan WiFi kos 0 detik dan rata-rata waktu proses dari *hotspot* pribadi 0,00006 detik.

## 4.4 Pengujian Waktu Proses (Pengambilan Data PPTI)

### 4.4.1 Tujuan

Menguji waktu proses (pengambilan data PPTI) yang diambil pada internet dengan dua jaringan yang berbeda.

### 4.4.2 Alat dan Bahan

1. Laptop
2. ESP8266
3. LED Matrix P10
4. Power Supply 5V
5. Hotspot

### 4.4.3 Prosedur

Menyalakan laptop kemudian menggunakan program (lampiran 5) yang telah di upload pada Arduino IDE dengan menghubungkan wifi agar dapat tersambung pada internet, kemudian hasil bisa terlihat pada serial monitor dengan menyalakan timestampnya.

Pada program Arduino IDE nantinya akan ada start dan stop dibagian ambil data, tentunya dikasih delay bernilai 1000, jadi 2 baris pertama akan dihitung sesuai prosedur yaitu waktu akhir dikurangi waktu awal selama 30 kali percobaan, dan percobaan kali ini menggunakan 2 jaringan yang berbeda.

Tabel 4.5 Pengujian waktu proses (pengambilan data PPTI) menggunakan WiFi kos

No.	Waktu Awal	Waktu Akhir	Waktu Proses
1.	12:38:21.681	12:38:21.681	0 detik
2.	12:38:23.890	12:38:23.890	0 detik

3.	12:38:26.375	12:38:26.415	0.04 detik
4.	12:38:28.717	12:38:28.717	0 detik
5.	12:38:30.980	12:38:31.022	0.042 detik
6.	12:38:33.326	12:38:33.359	0.033 detik
7.	12:38:35.595	12:38:35.595	0 detik
8.	12:38:37.852	12:38:37.893	0.041 detik
9.	12:38:40.098	12:38:40.098	0 detik
10.	12:38:45.151	12:38:45.151	0 detik
11.	12:38:47.380	12:38:47.411	0.031 detik
12.	12:38:49.722	12:38:49.722	0 detik
13.	12:38:51.974	12:38:52.017	0.043 detik
14.	12:38:54.305	12:38:54.307	0.002 detik
15.	12:38:56.691	12:38:56.737	0.046 detik
16.	12:38:58.852	12:38:58.885	0.033 detik
17.	12:39:01.679	12:39:01.679	0 detik
18.	12:39:03.884	12:39:03.884	0 detik
19.	12:39:06.161	12:39:06.202	0.041 detik
20.	12:39:08.515	12:39:08.515	0 detik
21.	12:39:11.109	12:39:11.109	0 detik
22.	12:39:13.733	12:39:13.733	0 detik
23.	12:39:16.616	12:39:16.616	0 detik
24.	12:39:18.874	12:39:18.874	0 detik
25.	12:39:23.105	12:39:23.138	0.033 detik
26.	12:39:25.651	12:39:25.651	0 detik
27.	12:39:28.013	12:39:28.013	0 detik
28.	12:39:30.311	12:39:30.345	0.034 detik
29.	12:39:32.680	12:39:32.680	0 detik
30.	12:39:35.092	12:39:35.092	0 detik
Rata – Rata Waktu Proses			0,014 detik

Tabel 4.6 Pengujian waktu proses (pengambilan data PPTI) menggunakan *hotspot* pribadi

No.	Waktu Awal	Waktu Akhir	Waktu Proses
1.	13:11:03.630	13:11:03.630	0 detik
2.	13:11:06.317	13:11:06.363	0.046 detik
3.	13:11:09.149	13:11:09.149	0 detik
4.	13:11:12.013	13:11:12.013	0 detik
5.	13:11:14.894	13:11:14.894	0 detik
6.	13:11:17.618	13:11:17.618	0 detik
7.	13:11:20.436	13:11:20.478	0.042 detik
8.	13:11:24.030	13:11:24.066	0.036 detik
9.	13:11:26.750	13:11:26.750	0 detik
10.	13:11:29.424	13:11:29.424	0 detik
11.	13:11:32.196	13:11:32.196	0 detik
12.	13:11:34.829	13:11:34.861	0.032 detik
13.	13:11:37.558	13:11:37.602	0.044 detik
14.	13:11:40.256	13:11:40.256	0 detik
15.	13:11:42.976	13:11:42.976	0 detik
16.	13:11:45.649	13:11:45.690	0.041 detik
17.	13:11:48.361	13:11:48.361	0 detik
18.	13:11:50.966	13:11:50.966	0 detik
19.	13:11:53.877	13:11:53.877	0 detik
20.	13:11:57.314	13:11:57.350	0.036 detik
21.	13:12:00.066	13:12:00.066	0 detik
22.	13:12:03.095	13:12:03.095	0 detik
23.	13:12:05.730	13:12:05.764	0.034 detik
24.	13:12:08.536	13:12:08.536	0 detik
25.	13:12:11.510	13:12:11.510	0 detik
26.	13:12:14.313	13:12:14.313	0 detik
27.	13:12:17.094	13:12:17.128	0.034 detik
28.	13:12:20.335	13:12:20.335	0 detik

29.	13:12:23.107	13:12:23.107	0 detik
30.	13:12:25.916	13:12:25.957	0 detik
Rata – Rata Waktu Proses			0,012 detik

#### 4.4.4 Analisis Pengujian Waktu Proses (Pengambilan Data PPTI)

Pada Tabel 4.5 dan Tabel 4.6 setelah melakukan 30 kali percobaan dengan menggunakan jaringan yang berbeda, untuk informasi waktu proses pengambilan data PPTI di jaringan pertama yaitu menggunakan WiFi kos rencana cukup berhasil dikarenakan mendapatkan 12 waktu proses yang berbeda. Pada kasus dimana menggunakan jaringan hotspot pribadi mendapatkan 9 waktu proses pengambilan data PPTI yang belum sama.

Jadi, kesimpulan dari pengujian ini adalah seluruh pengujian total dilakukan 30 kali dari jaringan yang berbeda, rata-rata keberhasilan dari WiFi kos 0,014 detik dan *hotspot* pribadi 0,012 perbedaannya hanya 0,002 detik saja.

### 4.5 Pengujian Seluruh Alat

#### 4.5.1 Tujuan

Pengujian ini dilakukan untuk mengetahui keseluruhan alat bekerja dengan tepat dan pengamatan kedipan (*flicker*) saat pengambilan data setiap 1 menit sekali.

#### 4.5.2 Alat dan Bahan

1. Laptop
2. ESP8266
3. LED Matrix P10
4. Power Supply 5V
5. PCB JWS Kontroller



## 6. Stopwatch di HP

### 4.5.3 Prosedur

Menyalakan laptop kemudian menggunakan program (lampiran 2) yang telah di upload pada Arduino IDE dengan menghubungkan wifi agar dapat tersambung pada internet, kemudian hasil bisa dilihat pada tampilan LED Matrix P10 pengamatan kedipan setiap 1 menit sekali dan dengan pengulangan sebanyak 30 kali.

Hal-hal yang diamati pada pengujian ini adalah kemunculan informasi jam, identitas, dan kehadiran pada tampilan LED Matrix P10. Selain itu diamati pula munculnya *flicker* yang terlihat pada tampilan LED Matrix P10 serta mencatat waktu kemunculannya.

Tabel 4.7 Pengujian seluruh alat

No.	Waktu Proses (MM:DD)	Jam	Identitas	Kehadiran	Keterangan
1.	01:01	✓	✓	✓	Ada kedipan pada kehadiran
2.	02:02	✓	✓	✓	Ada kedipan pada kehadiran
3.	03:03	✓	✓	✓	Ada kedipan pada kehadiran
4.	04:04	✓	✓	✓	Ada kedipan pada kehadiran
5.	05:05	✓	✓	✓	Ada kedipan pada identitas
6.	06:06	✓	✓	✓	Ada kedipan pada identitas
7.	07:07	✓	✓	✓	Ada kedipan pada jam

8.	08:08	✓	✓	✓	Ada kedipan pada kehadiran
9.	09:09	✓	✓	✓	Ada kedipan pada kehadiran
10.	10:10	✓	✓	✓	Ada kedipan pada kehadiran
11.	11:11	✓	✓	✓	Ada kedipan pada kehadiran
12.	12:12	✓	✓	✓	Ada kedipan pada kehadiran
13.	13:13	✓	✓	✓	Ada kedipan pada identitas
14.	14:14	✓	✓	✓	Ada kedipan pada identitas
15.	15:15	✓	✓	✓	Ada kedipan pada jam
16.	16:16	✓	✓	✓	Ada kedipan pada kehadiran
17.	17:17	✓	✓	✓	Ada kedipan pada kehadiran
18.	18:18	✓	✓	✓	Ada kedipan pada kehadiran
19.	19:19	✓	✓	✓	Ada kedipan pada kehadiran
20.	20:20	✓	✓	✓	Ada kedipan pada kehadiran
21.	21:21	✓	✓	✓	Ada kedipan pada identitas
22.	22:22	✓	✓	✓	Ada kedipan pada jam
23.	23:23	✓	✓	✓	Ada kedipan pada kehadiran
24.	24:24	✓	✓	✓	Ada kedipan pada kehadiran

25.	25:25	✓	✓	✓	Ada kedipan pada kehadiran
26.	26:26	✓	✓	✓	Ada kedipan pada kehadiran
27.	27:27	✓	✓	✓	Ada kedipan pada kehadiran
28.	28:28	✓	✓	✓	Ada kedipan pada identitas
29.	29:29	✓	✓	✓	Ada kedipan pada jam
30.	30:30	✓	✓	✓	Ada kedipan pada identitas

#### 4.5.4 Analisis Pengujian Seluruh Alat

Pada Tabel 4.7 setelah melakukan 30 kali percobaan, informasi waktu, identitas, dan kehadiran dosen selalu muncul sesuai dengan rencana. Muncul juga kedipan yang terjadi secara berkala setiap 1 menit 1 detik (61 detik). Kedipan (*flicker*) ini terjadi karena mikrokontroller melakukan pengambilan data PPTI yang dilakukan setiap 1 menit sekali. Pengambilan data ini memerlukan waktu tunggu yang tidak dapat dikendalikan, karena bergantung kepada kualitas jaringan yang ada pada saat tersebut. Pengujian pernah dilakukan menggunakan jaringan yang kualitasnya rendah, hasilnya kedipan terlihat sangat menyolok sehingga mengganggu tampilan LED Matrix P10.

Jadi, kesimpulan dari pengujian ini adalah seluruh informasi berhasil ditampilkan, namun ada *flicker* yang terjadi setiap melakukan aktifitas pengambilan data PPTI. Kemudian, kualitas jaringan yang baik akan mempercepat pengambilan data, sehingga dapat mengurangi *flicker* pada tampilan LED Matrix P10.

## 4.6 Pengujian Panjang Karakter Pada LED Matrix P10

### 4.6.1 Tujuan

Mengetahui seberapa panjang karakter yang akan ditampilkan pada LED Matrix P10

### 4.6.2 Alat dan Bahan

1. Laptop
2. ESP8266
3. LED Matrix P10
4. Power Supply 5V
5. PCB JWS Kontroller

### 4.6.3 Prosedur

Menyalakan laptop kemudian menggunakan program (lampiran 3) yang telah di upload pada Arduino IDE dengan menghubungkan wifi agar dapat tersambung pada internet, kemudian hasil bisa dilihat pada tampilan LED Matrix P10.

Pertama menguji dari nilai array 1000 sampai dengan bernilai 3000 jadi untuk format datanya 4 digit angka, 1 spasi, total 5 karakter, dan dimulai 0000 sampai dengan 0599. Hal yang perlu diperhatikan adalah pengujian dilakukan secara bertahap.

Tabel 4.8 Tabel pengujian panjang karakter pada LED Matrix P10

No.	Jumlah Karakter	Keterangan
1.	500	Sesuai
2.	1000	Sesuai
3.	1500	Sesuai

4.	2000	Sesuai
5.	2500	Sesuai
6.	3000	Sesuai

#### 4.6.4 Analisis Pengujian Panjang Karakter Pada LED Matrix P10

Pada Tabel 4.8 setelah melakukan percobaan pada pengujian kali ini, telah ditemukan yaitu panjang karakter bisa mencapai 3000 sesuai dengan rencana. Pengujian ini dilakukan secara bertahap dengan dimulai dari 500 karakter sampai 3000 karakter. Kemudian untuk pengujian panjang karakter ini harus melihat tampilan LED Matrix P10 secara terus menerus, agar mendapatkan data yang diperoleh bisa sesuai.

Jadi, kesimpulan dari pengujian ini adalah pada tampilan LED Matrix P10 memiliki kemampuan untuk menampilkan karakter hingga ribuan dalam satu waktu, memungkinkan untuk menampilkan informasi yang cukup detail dan panjang dengan jelas dan terbaca.

#### 4.7 Penjelasan Konektor yang digunakan

Tabel 4.9 Tabel nama beserta koneksinya

No.	Nama	Koneksi
1.	Mikrokontroler ESP8266	PCB JWS Kontroller
2.	Kabel Data Modul Running Text	Panel Modul Running Text dan PCB JWS Kontroller.
3.	Panel Modul Running Text P10	Kabel Power Modul Running Text dan Kabel Data Modul Running Text.

4.	PCB JWS Kontroller	Mikrokontroller ESP8266, Kabel Data Modul Running Text dan Kabel Jumper Male to Female/Female to Male.
5.	Power Supply 5V	Kabel Power Supply 220V AC, Kabel Power Modul Running Text, Kabel Jumper Male to Female/Female to Male
6.	Kabel Jumper Male to Female/Female to Male	PCB JWS Kontroller dan Power Supply

Berikut Penjelasannya:

1. Semua pin yang ada pada Mikrokontroller ESP8266 terhubung ke PCB JWS Kontroller sehingga bisa diakses dengan mudah dan efektif.
2. Kabel data ini membawa sinyal data dari mikrokontroller ke modul running text, yang kemudian diinterpretasikan oleh modul untuk menampilkan karakter dan efek animasi yang sesuai.
3. Panel modul P10 dilengkapi dengan antarmuka komunikasi data yang memungkinkan pengiriman informasi teks dan instruksi kontrol dari Mikrokontroller ke panel. Karena terhubung dengan kabel data modul running text.
4. Dengan menggunakan PCB ini, kita dapat mengontrol pola pencahayaan, kecerahan, kecepatan perubahan warna, dan efek animasi pada sistem pencahayaan LED yang terhubung. Dengan koneksi mikrokontroller ESP8266 akan menjadi lebih luas lagi untuk mengaturnya.
5. Sebuah perangkat yang berfungsi untuk menyediakan tegangan listrik sebesar 5 Volt yang terhubung dengan kabel power supply 220V AC, kabel power modul running text, dan kabel jumper Male to Female/Female to Male.
6. Kabel jumper ini memudahkan penghubungan dan pemrograman komponen elektronik dengan cepat dan mudah, karena tidak memerlukan soldering atau

penyambungan permanen. Sehingga Pin vcc dan ground pada mikrokontroler ESP8266 yang terletak pada PCB JWS Kontroller dapat terhubung dengan power supply.



UNIVERSITAS  
**Dinamika**

## BAB V PENUTUP

### 5.1 Kesimpulan

Selama pengujian, sistem *Running LED Display* juga berhasil dalam menjaga konsistensi penampilan informasi kedatangan dosen, serta menunjukkan kemampuan untuk meng-*update* secara otomatis saat ada perubahan data, maka ada beberapa kesimpulan:

1. Pengujian waktu proses (pengambilan data PPTI) didapatkan total percobaan 30 kali dari jaringan yang berbeda, rata-rata keberhasilan dari WiFi kos 0,014 detik dan *hotspot* pribadi 0,012 detik perbedaannya hanya 0,002 detik saja.
2. Pengujian seluruh alat mendapatkan kesimpulan yaitu seluruh informasi berhasil ditampilkan, namun ada kedipan (*flicker*) yang terjadi setiap melakukan aktifitas pengambilan data PPTI. Kualitas jaringan yang baik akan mempercepat proses pengambilan data, sehingga dapat mengurangi *flicker* pada tampilan LED Matrix P10.
3. Pada pengujian panjang karakter dapat disimpulkan bahwa pada tampilan LED Matrix P10 memiliki kemampuan untuk menampilkan karakter hingga ribuan dalam satu waktu, memungkinkan untuk menampilkan informasi yang cukup detail dan panjang dengan jelas dan terbaca.

### 5.2 Saran

Adapun saran untuk pengembangan pada penelitian berikutnya, agar mendapatkan hasil yang lebih baik terdapat beberapa saran sebagai berikut:

1. Pengembangan dapat dilakukan menampilkan data secara dinamis, diantaranya: tampilan data dari arah kanan, kiri, atas, bawah atau tampilan data dari tengah.
2. Penelitian ini dapat dijadikan referensi untuk penelitian selanjutnya.
3. Memperluas tampilan informasi yang dapat ditampilkan di *Running LED Display* sebagai contoh menyertakan nama dosen, mata kuliah, jam mata kuliah dan ruang kuliah.



## DAFTAR PUSTAKA

- Agus Setiawan, D. A. (2021). IMPLEMENTASI INTERNET OF THINGS PADA ALAT HAND SANITIZER OTOMATIS MENGGUNAKAN TELEGRAM MESSENGER BOT BERBASIS ESP8266. *Rekursif: Jurnal Informatika*, 137-143.
- Bahtiar. (2022). KENDALI INFORMASI DIGITAL PADA RUANG PERTEMUAN UNIT PELAKSANA TRANSMISI MAKASSAR. *JURNAL MOSFET*, 1-5.
- Dendy Ramdani, F. M. (2020). Rancang Bangun Sistem Otomatisasi Suhu Dan Monitoring pH Air Aquascape Berbasis IoT (Internet Of Thing) Menggunakan NodeMCU Esp8266 Pada Aplikasi Telegram. *Journal of Informatics Information System Software Engineering and Applications (INISTA)*, 59-68.
- Hendrastuty, N., & Ihza, Y. (2021). Rancang Bangun Aplikasi Monitoring Santri Berbasis Android. *Jurnal Data Mining dan Sistem Informasi (JDMSI)*, 21-34.
- I Made Agus Chandra Wijaya, S. W. (2022). RANCANG BANGUN SISTEM NOMOR ANTRIAN BERBASIS INTERNET OF THINGS (IOT). *Jurnal SAINTEKOM*, 176-188.
- IDE, A. (n.d.). *About/Arduino IDE*. Retrieved from <https://docs.arduino.cc/software/ide-v1/tutorials/arduino-ide-v1-basics>
- Ismail. (2020). AKURASI WAKTU JAM MASJID DI KOTA LHKSEUMAWE. *Jurnal Al-Ijtimaayah*, 75-90.
- lastminuteengineers/getting-started-with-esp8266*. (n.d.). Retrieved from <https://lastminuteengineers.b-cdn.net/wp-content/uploads/iot/ESP8266-Pinout-NodeMCU.png>
- Ledcontrolcard*. (n.d.). Retrieved from <https://www.ledcontrolcard.com/wp-content/uploads/2021/01/p10-smd-outdoor-fullcolor-led-panel-1.jpg>
- Malik Yuhanas, C. F. (2021). RANCANG BANGUN RUNNING TEXT MENGGUNAKAN MODUL LED MATRIX P10 BERBASIS ARDUINO UNO DI FAKULTAS TEKNIK UNIVERSITAS PGRI BANYUWANGI. *JURNAL ZETROEM*, 16-22.
- Maulana, Y. M. (2022). Model SOP IT Service Desk Berdasarkan Framework ITIL V3. *Jurnal Teknologi Informasi Komunikasi (TEMATIK)*, 100-107.
- Nasution, Z. M., & Daud, M. (2023). Desain dan Realisasi Papan Informasi Jadwal Shalat Berbasis Aplikasi Telegram. *Jurnal Janitra Informatika dan Sistem Informasi*, 30-39.

- Oktaviani, L., & Ayu, M. (2021). Pengembangan Sistem Informasi Sekolah Berbasis Web Dua Bahasa SMA Muhammadiyah Gading Rejo. *Jurnal Pengabdian Pada Masyarakat*, 437-444.
- Ramadhanu, P. B., & Priandika, A. T. (2021). RANCANG BANGUN WEB SERVICE API APLIKASI SENTRALISASI PRODUK UMKM PADA UPTD PLUT KUMKM PROVINSI LAMPUNG. *Jurnal Teknologi dan Sistem Informasi (JTSI)*, 59-64.
- Seftiana, M., Najeri, A., Anggono, H., & Thyo Priandika, A. (2021). SISTEM PENGELOLAAN KEBERSIHAN BERBASIS MIKROKONTROLER ARDUINO PADA PETERNAKAN UNGGAS. *Jurnal Teknik dan Sistem Komputer (JTIKOM)*, 29-39.
- Septian Yogi, A. P. (2021). INTERNET OF THINGS (IOT) PENGONTROL HARGA BBM PADA TOTEM SPBU 14.212.272 KISARAN MENGGUNAKAN PANEL 10. *Journal of Science and Social Research (JSSR)*, 74-79.
- su.eager-led*. (n.d.). Retrieved from [https://cdn.shopify.com/s/files/1/0578/7323/6159/products/A-200AF-51\\_1024x1024.jpg?v=1630394532](https://cdn.shopify.com/s/files/1/0578/7323/6159/products/A-200AF-51_1024x1024.jpg?v=1630394532)
- Tokopedia/elektronmartcom*. (n.d.). Retrieved from <https://images.tokopedia.net/img/cache/900/VqbcmM/2021/12/22/6e4f7111-96a5-4e1d-98c4-d4c2778241ba.jpg>
- Trisetiyanto, A. N. (2020). RANCANG BANGUN ALAT PENYEMPROT DISINFECTAN OTOMATIS UNTUK MENCEGAH PENYEBARAN VIRUS CORONA. *Joined Journal (Journal of Informatics Education)*, 45-51.
- Zakki Fuadi Emzain, U. S. (2020). Pembuatan dan Pelatihan Mengoperasikan Display LED Dot Matrix berbasis NodeMCU ESP8266 sebagai Alarm Peningkat Sholat di Mushola Nurul Huda Poncokusumo-Malang. *JURPIKAT (Jurnal Pengabdian Kepada Masyarakat)*, 94-104.