

**PEMANTAUAN TEKANAN DARAH  
BERBASIS INTERNET OF THINGS (IoT)**



Disusun Oleh:

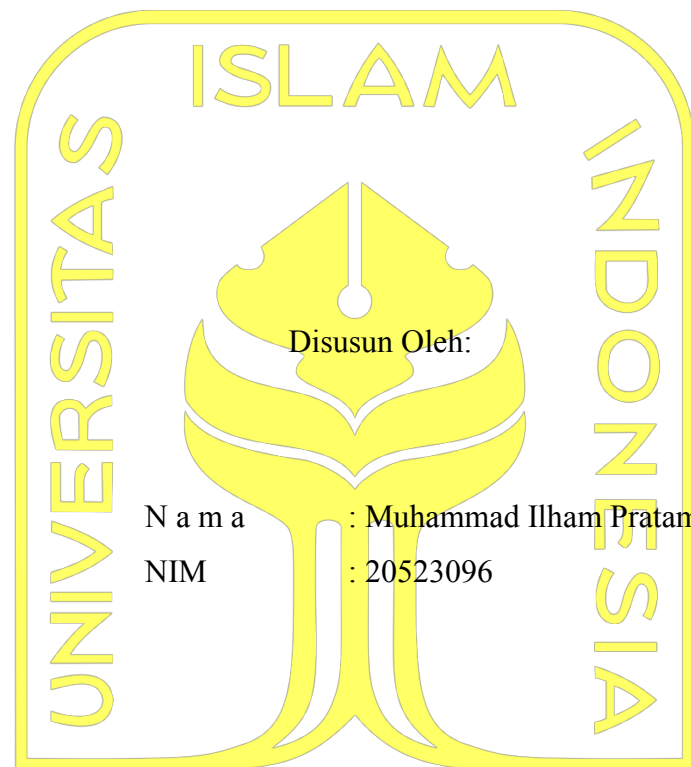
N a m a : Muhammad Ilham Pratama  
NIM : 20523096

**PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA  
FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS ISLAM INDONESIA  
2024**

**HALAMAN PENGESAHAN DOSEN PEMBIMBING**

**PEMANTAUAN TEKANAN DARAH**  
**BERBASIS INTERNET OF THINGS (IoT)**

**TUGAS AKHIR**



N a m a : Muhammad Ilham Pratama  
NIM : 20523096

الجمهورية الإسلامية الإندونيسية

Yogyakarta, 25 Oktober 2024

Pembimbing,

( Kurniawan Dwi Irianto, S.T., M.Sc. )

**HALAMAN PENGESAHAN DOSEN PENGUJI**

**PEMANTAUAN TEKANAN DARAH**

**BERBASIS INTERNET OF THINGS (IoT)**

**TUGAS AKHIR**

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika – Program Sarjana di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, Nopember 2024

Tim Penguji

Kurniawan Dwi Irianto, S.T., M.Sc.

**Anggota 1**

Ari Sujarwo, S.Kom., M.I.T.

**Anggota 2**

Dr. Novi Setiani, S.T., M.T.

Mengetahui,

Ketua Program Studi Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia



(Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D)

## HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Muhammad Ilham Pratama

NIM : 20523096

Tugas akhir dengan judul:

### PEMANTAUAN TEKANAN DARAH BERBASIS INTERNET OF THINGS (IoT)

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung risiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 25 Oktober 2024



( Muhammad Ilham Pratama )

## **HALAMAN PERSEMBAHAN**

Tugas akhir ini dipersembahkan kepada kedua orangtua saya yang telah memberikan dukungan penuh disaat susah maupun senang. Tanpa doa dari kedua orangtua, saya tidak akan pernah sampai dititik ini. Terimakasih atas perhatian yang diberikan setiap saat sehingga saya dapat menjalani semuanya dengan baik.

**HALAMAN MOTO**

*“it doesn't matter how slowly yo go as long as you don't stop” .*

-Confucius

## KATA PENGANTAR

*Assalamu'alaikum Warrahmatullahi Wabarakatuh*

Puji syukur saya panjatkan ke hadirat Allah SWT atas limpahan rahmat, karunia, serta petunjuk-Nya, sehingga saya dapat menyelesaikan tugas akhir yang berjudul "PEMANTAUAN TEKANAN DARAH BERBASIS INTERNET OF THINGS(IOT)".

Tugas akhir ini disusun sebagai salah satu syarat untuk menyelesaikan pendidikan Strata Satu (S1) di Universitas Islam Indonesia, Fakultas Teknologi Industri, Program Studi Informatika. Pada proses penyusunan tugas akhir ini, saya menyadari tanpa dukungan, bimbingan, dan arahan yang diberikan dari berbagai pihak, tugas akhir ini tidak akan terselesaikan dengan baik. Oleh karena itu, saya ingin mengucapkan terimakasih kepada :

1. Allah SWT yang telah membantu disetiap langkahnya.
2. Ayah (Novagus Viraputra) Ibu (Fitri) Adik (Najwa Avrilya) dan nenek(Yohanis), yang selalu memberikan dukungan serta semangat disaat susah maupun senang.
3. Bapak Kurniawan Dwi Irianto, S.T., M.Sc. Selaku dosen pembimbing tugas akhir atas arahan,bantuan, serta bimbingannya dalam proses pengerjaan tugas akhir hingga selesai.
4. Bapak Dr. Raden Teduh Dirgahayu, S.T., M.Sc. Ketua Jurusan Informatika Universitas Islam Indonesia.
5. Bapak Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D Ketua Program Studi Informatika Universitas Islam Indonesia.
6. Teman-teman dari Pekanbaru yang telah memberikan motivasi serta semangat saat saya kesulitan. Secara khusus saya ucapkan terimakasih kepada Akan, Pawang, Ijal, Sila, Olip, Uwik, Sani, dan yang lainnya yang tidak dapat saya tuliskan satu persatu.
7. Rekan-rekan diperkuliahan yang telah menemani dalam pengerjaan tugas akhir.

Saya menyadari akan banyaknya kekurangan dalam tugas akhir yang saya kerjakan. Oleh karena itu, saya menerima kritik dan saran yang membangun untuk penyempurnaan penelitian kedepannya. Semoga Allah SWT memberikan taufik dan hidayah-Nya kepada kita semua. Amin.

*Wassalamu'alaikum Warrahmatullahi Wabarakatuh*

Yogyakarta, 25 Oktober 2024

A handwritten signature in black ink, appearing to be 'Muhammad Ilham Pratama', written in a cursive style.

( Muhammad Ilham Pratama )

## SARI

Dengan populasi manusia yang semakin bertambah, diperlukan perhatian khusus, termasuk pemantauan kesehatan. Untuk mencegah komplikasi, tekanan darah harus diperiksa secara rutin karena merupakan indikator kesehatan penting bagi manusia. Organisasi Kesehatan Dunia (WHO) menyatakan bahwa pada tahun 2015, sekitar 1,13 miliar orang di seluruh dunia didiagnosis menderita hipertensi, yang setara dengan satu dari tiga orang. Kebanyakan orang tidak secara teratur memantau tekanan darah mereka karena sulitnya akses untuk ke rumah sakit. Banyak orang yang sering lupa mencatat hasil pengukuran tekanan darahnya setelah melakukan pengukuran di rumah. Penelitian ini membuat Solusi berupa sebuah sistem pemantauan tekanan darah dengan menggunakan sensor MPX5700AP berbasis IoT yang dapat diakses melalui Blynk dan Telegram. Dalam penelitian ini terdapat proses identifikasi kebutuhan, desain sistem, perakitan alat, implementasi, pengujian sistem, dan evaluasi yang digunakan sebagai metodologi. Hasil pengujian menunjukkan bahwa sistem dapat menampilkan hasil pengukuran tekanan darah pada aplikasi. Diharapkan dengan adanya teknologi ini dapat membantu penggunaanya dalam mengukur tekanan darah dan mencatat hasilnya secara otomatis tanpa perlu mencatatnya secara manual.

Kata kunci: IoT, hipertensi, Blynk, Telegram, tekanan darah.

## GLOSARIUM

Mikrokontroler	Komputer mini yang dapat diprogram untuk menjalankan tugas spesifik seperti mengontrol sensor dan mengirimkan data pengukuran ke aplikasi.
ESP32	Mikrokontroler dengan kemampuan Wi-Fi dan Bluetooth yang banyak digunakan dalam aplikasi <i>Internet of Things</i> (IoT). Berfungsi untuk mengontrol perangkat keras dan mengirimkan data.
Telegram	Aplikasi yang digunakan untuk bertukar pesan.
MPX5700AP	Sensor yang digunakan untuk mengukur tekanan darah.
<i>Internet of Things</i>	Teknologi yang menghubungkan berbagai perangkat elektronik ke internet.
<i>Driver</i> L298N	Digunakan untuk mengendalikan pompa motor dc dan solenoid.
Pompa Motor DC	Pompa listrik yang digunakan untuk memompa udara.
Solenoid	Komponen elektromekanis yang digunakan untuk membuka atau menutup katup udara.
ADC	Perangkat yang mengubah sinyal analog dari sensor menjadi sinyal digital yang dapat dibaca oleh mikrokontroler.
<i>systole</i>	Tekanan darah tertinggi yang terjadi saat jantung berkontraksi untuk memompa darah ke seluruh tubuh.
<i>diastole</i>	Tekanan darah terendah yang terjadi saat jantung beristirahat di antara detak jantung, tepat setelah kontraksi.
Arduino IDE	Perangkat lunak yang digunakan untuk menulis program.

## DAFTAR ISI

HALAMAN JUDUL .....	i
HALAMAN PENGESAHAN DOSEN PEMBIMBING .....	ii
HALAMAN PENGESAHAN DOSEN PENGUJI .....	iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR .....	iv
HALAMAN PERSEMBAHAN .....	v
HALAMAN MOTO .....	vi
KATA PENGANTAR .....	vii
SARI .....	ix
GLOSARIUM .....	x
DAFTAR ISI .....	xi
DAFTAR TABEL .....	xiii
DAFTAR GAMBAR .....	xiv
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Tujuan Penelitian .....	2
1.4 Batasan Masalah .....	2
1.5 Sistematika Penulisan .....	3
BAB II LANDASAN TEORI .....	4
2.1 Kajian Teori .....	4
2.1.1 Tekanan Darah .....	4
2.1.2 Mikrokontroler .....	4
2.1.3 MPX5700AP .....	6
2.1.4 Pompa Motor DC .....	8
2.1.5 Solenoid .....	8
2.1.6 Manset Cuff .....	9
2.1.7 Motor Driver L298N .....	10
2.1.8 Blynk .....	11
2.1.9 Telegram .....	11
2.1.10 Kalibrasi .....	12
2.2 Kajian Literatur .....	12
BAB III METODOLOGI PENELITIAN .....	17
3.1 Tahapan Pengembangan Sistem .....	17
3.2 Identifikasi Kebutuhan .....	18
3.2.1 Kebutuhan Pengguna .....	18
3.2.2 Kebutuhan <i>Input</i> .....	19
3.2.3 Kebutuhan <i>Output</i> .....	19
3.2.4 Kebutuhan <i>Hardware</i> .....	19
3.2.5 Kebutuhan <i>Software</i> .....	20
3.3 Perancangan Sistem .....	20
3.3.1 Perancangan Umum Sistem .....	20
3.3.2 Perancangan Alur Kerja Sistem .....	21
3.3.3 Perancangan Komponen <i>Hardware</i> .....	23
3.4 Perakitan Alat .....	24
3.4.1 Konfigurasi Sensor Tekanan .....	24
3.4.2 Konfigurasi Motor Driver L298N .....	25
3.4.3 Konfigurasi Pompa Motor DC dan Solenoid .....	25

3.4.4	Konfigurasi LCD .....	26
3.5	Implementasi Serta Pengujian Alat dan Sistem .....	27
3.5.1	Skenario Pengujian .....	28
3.6	Evaluasi .....	30
	<b>BAB IV HASIL DAN PEMBAHASAN</b> .....	<b>31</b>
4.1	Implementasi Sistem dan Sintaks Program .....	31
4.1.1	Implementasi Perangkat Keras ( <i>Hardware</i> ) .....	31
4.1.2	Implementasi Perangkat Lunak ( <i>Software</i> ) .....	32
4.2	Pembahasan Sintaks Program .....	34
4.3	Pengujian Perangkat .....	41
4.3.1	Pengujian Kalibrasi Sensor MPX5700AP .....	41
4.3.2	Pengujian Sensor MPX5700AP dan Tensimeter Digital .....	43
4.3.3	Pengujian pada Blynk dan Telegram .....	44
4.3.4	Pengujian pada Pengguna .....	45
4.4	Evaluasi .....	46
	<b>BAB V KESIMPULAN</b> .....	<b>48</b>
5.1	Kesimpulan .....	48
5.2	Saran .....	48
	<b>DAFTAR PUSTAKA</b> .....	<b>49</b>

**DAFTAR TABEL**

Tabel 2. 1 Kajian Literatur.....	13
Tabel 3. 1 Skenario Pengujian kalibrasi dari sensor MPX5700AP .....	28
Tabel 3. 2 Skenario Pengujian pada Blynk dan Telegram.....	29
Tabel 3. 3 Daftar Pertanyaan Pengujian pada Pengguna .....	30
Tabel 4. 1 kalibrasi sensor .....	41
Tabel 4. 2 Pengujian Sensor.....	43
Tabel 4. 3 Pengujian pada Blynk dan Telegram .....	45
Tabel 4. 4 Pengujian Pada Pengguna.....	45

## DAFTAR GAMBAR

Gambar 2. 1 ESP32.....	5
Gambar 2. 2 MPX5700AP.....	7
Gambar 2. 3 Pompa Motor DC.....	8
Gambar 2. 4 Solenoid.....	9
Gambar 2. 5 Manset cuff.....	10
Gambar 2. 6 Motor Driver L298N.....	10
Gambar 3. 1 Tahapan pengerjaan.....	17
Gambar 3. 2 Gambaran Umum Sistem.....	21
Gambar 3. 3 Alur Kerja Sistem.....	23
Gambar 3. 4 Rancangan Rangkaian Alat.....	23
Gambar 3. 5 konfigurasi sensor MPX5700AP dan ESP32.....	24
Gambar 3. 6 konfigurasi motor driver L298N.....	25
Gambar 3. 7 konfigurasi pompa motor DC dan <i>solenoid</i> .....	26
Gambar 3. 8 konfigurasi LCD.....	27
Gambar 4. 1 Tampilan luar alat.....	32
Gambar 4. 2 Tampilan Blynk.....	33
Gambar 4. 3 Tampilan Telegram.....	34
Gambar 4. 4 Sintaks <i>library</i> dan integrasi Blynk serta Telegram.....	34
Gambar 4. 5 Sintaks konfigurasi lcd,definisi telegram,dan variabel penyimpanan nilai tekanan .....	35
Gambar 4. 6 Sintaks inisialisasi dari komunikasi serial,pin,wifi,blynk, dan lcd.....	37
Gambar 4. 7 Sintaks memulai pengukuran.....	37
Gambar 4. 8 Sintaks pembacaan nilai.....	38
Gambar 4. 9 Sintaks pengiriman hasil ke Blynk dan Telegram.....	39
Gambar 4. 10 Sintaks fungsi <i>inflateCuff</i> dan <i>deflateCuff</i> .....	39
Gambar 4. 11 Sintaks fungsi <i>detectSystoleDiastole</i> .....	41
Gambar 4. 12 grafik kalibrasi linear sensor.....	42
Gambar 4. 13 Pengujian sensor.....	43
Gambar 4. 14 Pengujian pada Blynk dan Telegram.....	45
Gambar 4. 15 Pengujian pada Pengguna.....	46

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Seiring bertambahnya usia, manusia semakin rentan terhadap risiko penyakit. Oleh karena itu, sangat penting untuk lebih memperhatikan berbagai aspek kesejahteraan, seperti menjaga pola makan yang teratur dan seimbang, kesehatan fisik dan mental, serta menjamin kenyamanan dan keamanan di rumah (Rihhadatulaisy & Irianto, 2024). Salah satu cara untuk mengetahui apakah ada masalah dengan fungsi dasar tubuh adalah dengan melakukan pemeriksaan tanda vital (Cahyana et al., 2023). Salah satu pemeriksaan tanda vital yang harus dicek adalah tekanan darah.

Tekanan darah adalah tekanan yang dialami darah saat jantung memompa darah ke seluruh tubuh. Terdapat dua metode pengukuran tekanan darah yaitu langsung (invasif) dan tidak langsung (non-invasif). Metode invasif adalah dengan cara memasukkan jarum ke dalam arteri, sedangkan metode non-invasif yaitu menggunakan tensimeter untuk mengukur tekanan darah (Sulista et al., 2021). Tidak semua tekanan darah berada dalam batas normal sehingga menyebabkan munculnya gangguan pada tekanan darah yakni dikenal dengan hipertensi atau tekanan darah tinggi dan hipotensi atau tekanan darah rendah (Fitriani & Nilamsari, 2017). Gagal jantung kronis, stroke, aneurisma arteri, dan Serangan jantung merupakan faktor risiko utama yang disebabkan oleh tekanan darah tinggi (Nur Fariz & jamaaluddin, 2021). Data World Health Organization (WHO) tahun 2015 menunjukkan sekitar 1,13 Miliar orang di dunia menyandang hipertensi, artinya 1 dari 3 orang di dunia terdiagnosis hipertensi.

Jumlah penyandang hipertensi terus meningkat setiap tahunnya, diperkirakan pada tahun 2025 akan ada 1,5 Miliar orang yang terkena hipertensi, dan diperkirakan setiap tahunnya 9,4 juta orang meninggal akibat hipertensi dan komplikasinya. Prevalensi hipertensi berdasarkan hasil pengukuran pada penduduk usia  $\geq 18$  tahun sebesar 34,1%, tertinggi di Kalimantan Selatan (44,1%), sedangkan terendah di Papua sebesar (22,2%). Hipertensi terjadi pada kelompok umur 31-44 tahun (31,6%), umur 45-54 tahun (45,3%), umur 55-64 tahun (55,2%) (K. K. RI, 2019).

Mengonsumsi makanan yang sehat, melakukan pemeriksaan medis teratur, dan berolahraga secara teratur adalah beberapa cara untuk tetap sehat, terutama bagi orang tua (Syani et al., 2023). Keterbatasan sumber daya dan tenaga serta meningkatnya kebutuhan

pelayanan kesehatan khususnya untuk pengukuran dan pemantauan tekanan darah secara berkala membuat rumah sakit kewalahan dalam mengatasi kebutuhan pelayanan yang semakin meningkat. Belum lagi jarak antara rumah dan rumah sakit yang jauh sehingga membuat orang-orang kesulitan untuk harus pergi mengecek kesehatannya secara rutin. Maka dibutuhkan solusi dengan membuat sebuah sistem yang dapat membantu pelayanan kesehatan khususnya dalam pemantauan tekanan darah.

Pada era industry 4.0, penggunaan internet merambah di segala aspek kehidupan manusia (Prayoga, 2020). Salah satu perkembangan teknologi yang sedang banyak dibicarakan yaitu *Internet of Things* (IoT). *Internet of Things* adalah jaringan komunikasi di mana alat-alat dan sensor saling berhubungan satu sama lain atau dengan sistem yang lebih besar (Ratna et al., 2020). *Internet of things* bisa menjadi solusi atas permasalahan di atas. Oleh karena itu, peneliti ingin membuat sebuah sistem berbasis *Internet of Things* dengan tujuan untuk mempermudah penggunaannya dalam melakukan pemantauan tekanan darah sehingga pengguna dapat mengecek dan memantau tekanan darahnya secara berkala. Sistem dibuat menggunakan sensor tekanan yang kemudian datanya akan dibaca oleh mikrokontroler yang akan mengolah data tersebut untuk dikirim ke Blynk dan Telegram yang merupakan tempat untuk monitoring tekanan darah pengguna.

## **1.2 Rumusan Masalah**

Berdasarkan latar belakang di atas, didapat rumusan masalah pada penelitian ini yaitu bagaimana membangun sistem pemantauan tekanan darah berbasis Internet of Things (IoT)?

## **1.3 Tujuan Penelitian**

Tujuan dari penelitian ini adalah untuk membangun sebuah sistem yang dapat mempermudah pengguna dalam melakukan pemantauan tekanan darah sehingga pengguna dapat mengecek dan memantau tekanan darahnya secara berkala.

## **1.4 Batasan Masalah**

Untuk memberikan serta menetapkan fokus pada penelitian ini, maka dibutuhkan batasan agar penelitian ini tetap fokus ke arah permasalahan yang dituju, adapun batasan masalah adalah sebagai berikut :

- a. Untuk mengakses hasil pengukuran dibutuhkan koneksi internet.

- b. Sensor yang digunakan untuk mengukur tekanan darah hanya sensor tekanan.
- c. Pengguna sistem nantinya adalah pengguna yang pandai menggunakan teknologi.

### **1.5 Sistematika Penulisan**

Sistematika penulisan berguna untuk mengetahui isi dan mempermudah pembaca dalam memahami penelitian ini, adapun sistematika penulisan adalah sebagai berikut:

#### **BAB I PENDAHULUAN**

Bagian ini terdiri dari latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, dan sistematika penulisan.

#### **BAB II LANDASAN TEORI**

Pada landasan teori akan membahas tentang review dari penelitian terkait, Internet of Things, Tekanan darah, dan perangkat serta komponen yang akan digunakan dalam penelitian.

#### **BAB III METODOLOGI**

Pada bagian ini akan mengulas gambaran umum sistem, kebutuhan proses sistem, tahap perancangan sistem, proses perakitan alat serta pengujian pada sistem.

#### **BAB IV HASIL DAN PEMBAHASAN**

Pada bab ini akan membahas tentang hasil dari perangkat pemantauan tekanan dari berbasis Internet of Things.

#### **BAB V KESIMPULAN DAN SARAN**

Pada bab ini akan membahas tentang kesimpulan hasil dari penelitian sistem pemantauan tekanan darah berbasis Internet of Things serta saran untuk mengembangkan sistem tersebut agar sistem menjadi lebih baik kedepannya.

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Kajian Teori**

Kajian teori dari pengembangan sistem dalam penelitian ini adalah sebagai berikut.

##### **2.1.1 Tekanan Darah**

Tekanan darah adalah ukuran seberapa kuat aliran darah pada dinding arteri yang juga menunjukkan seberapa kuat jantung dalam memompa darah keseluruh tubuh. Karena manometer air raksa telah digunakan sebagai rujukan baku untuk pengukuran tekanan darah dalam sejarah fisiologi, tekanan darah hampir selalu dinyatakan dalam millimeter air raksa (mmHg) (Isyanto et al., 2022). Tekanan darah dapat diukur dan hasil pengukurannya terdiri dari dua angka yang disebut sistolik dan diastolik. Tekanan darah secara umum ditulis dengan tekanan sistolik terlebih dahulu, diikuti dengan tekanan diastolik, seperti contohnya 120/80 mmHg (Ashshiddiq & Rahmadya, 2023).

Tekanan darah normal pada manusia ada pada angka <120mmHg untuk sistolik dan <80 mmHg untuk diastolik. Sementara itu, untuk tekanan di atas 130/85 mmHg sudah termasuk tinggi.

Semakin bertambah usia maka semakin tinggi tekanan darah normalnya. Ini disebabkan pembuluh darah manusia cenderung mengeras atau kaku seiring bertambahnya usia, yang membuat jantung bekerja lebih keras. Kurangnya aktifitas fisik meningkatkan risiko menderita hipertensi karena meningkatkan risiko kelebihan berat badan. Orang yang kurang melakukan aktivitas fisik juga mempunyai frekuensi denyut jantung yang lebih tinggi sehingga otot jantungnya harus bekerja lebih keras untuk memompa lebih keras pada saat kontraksi (Stikes, 2019).

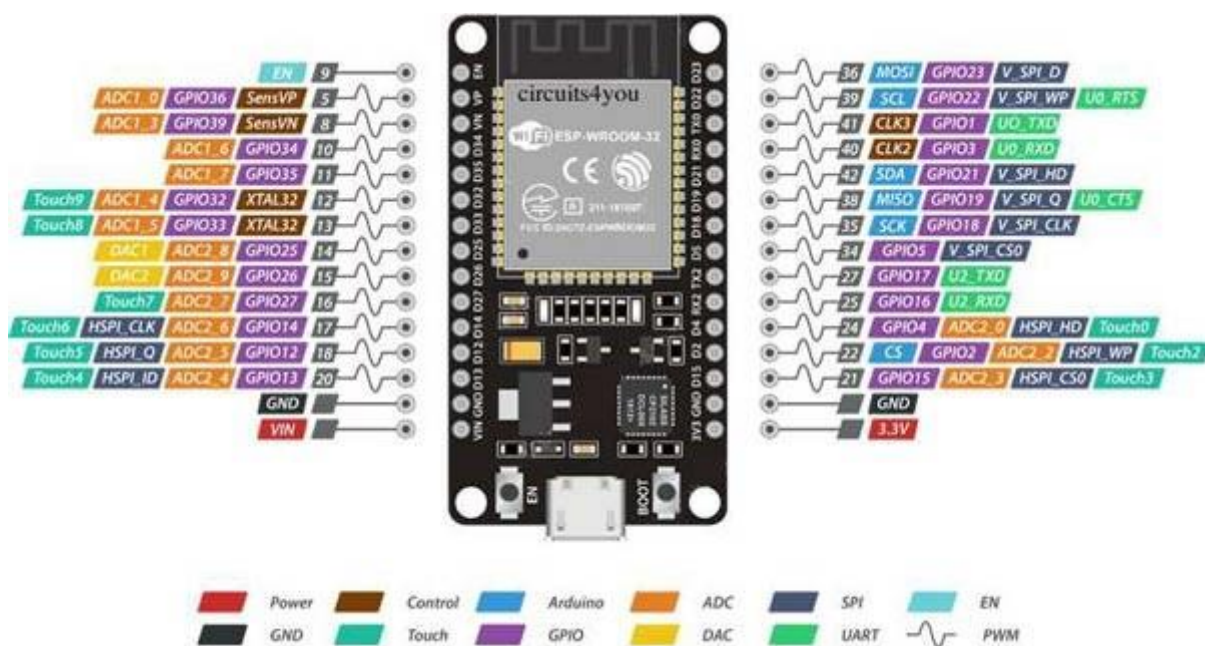
##### **2.1.2 Mikrokontroler**

Mikrokontroler adalah sebuah perangkat elektronik berupa sistem komputer kecil yang dirancang untuk mengendalikan fungsi atau proses tertentu pada perangkat lain. Mikrokontroler umumnya terdiri atas unit pemroses (CPU), memori (RAM dan ROM), serta antarmuka input/output (I/O) yang terintegrasi dalam satu chip.

###### **a. ESP 32**

ESP32 merupakan sebuah mikrokontroler yang dikembangkan oleh perusahaan Espressif Systems, yang merupakan produk lanjutan atau penerus dari ESP8266 yang

memiliki berbagai peningkatan fitur dan kinerja sistem. ESP32 didasarkan dengan Xtensa LX6 dengan prosesor *dual-core* yang berjalan pada *clock speed* hingga 240Mhz yang membuat ESP32 dapat memproses data lebih cepat daripada mikrokontroler pendahulunya. ESP32 juga menawarkan konektivitas nirkabel yang luas yang memungkinkannya terhubung dengan Wi-Fi dan Bluetooth. Ini membuat ESP32 menjadi salah satu mikrokontroler yang populer dalam mengerjakan proyek *internet of things*. ESP32 memiliki total 30 pin yang diantaranya adalah pin tegangan dan pin GPIO (*General Purpose Input Output*) dengan spesifikasi 15 pin ADC (*Analog to Digital Converter*), 3 UART Interface, 3 SPI Interface, 2 I2C Interface, 16 pin PWM (*Pulse Width Modulation*), 2 pin DAC (*Digital to Analog Converter*) (Ardutech, 2020). Bentuk dari ESP32 dapat dilihat pada gambar 2. 1.



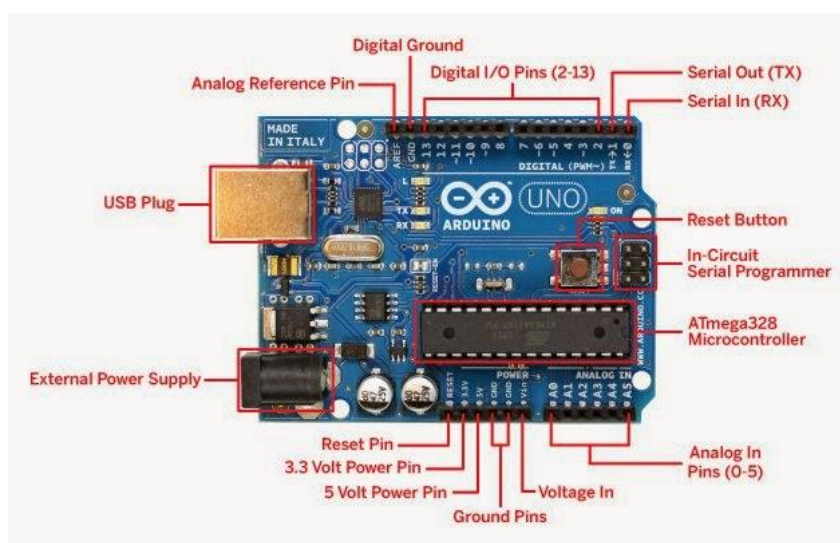
Gambar 2. 1. ESP32

Sumber (Ardutech, 2020)

#### b. Arduino Uno

Arduino Uno merupakan salah satu papan mikrokontroler yang populer dan banyak digunakan dalam pengembangan proyek elektronik dan otomasi. Papan ini berbasis pada mikrokontroler ATmega328P, yang dilengkapi dengan sejumlah pin input dan output yang memungkinkan interaksi dengan berbagai komponen eksternal seperti

sensor, motor, LED, dan perangkat lainnya. Arduino Uno memiliki beberapa spesifikasi penting, antara lain tegangan operasional 5V, memori flash sebesar 32 KB, serta kecepatan clock 16 MHz. Papan ini juga menyediakan 14 pin digital I/O, 6 di antaranya dapat digunakan sebagai output PWM, serta 6 pin analog input. Arduino Uno berfungsi sebagai platform pengembangan yang mudah diakses dengan menggunakan IDE Arduino, yang mendukung bahasa pemrograman C/C++ untuk menulis dan mengunggah program ke mikrokontroler. Bentuk dari Arduino Uno dapat dilihat pada gambar 2. 2.



Gambar 2. 2 Arduino Uno

Berdasarkan perbandingan antara ESP32 dan Arduino Uno, ESP32 dipilih sebagai mikrokontroler yang digunakan dalam penelitian ini karena memiliki berbagai keunggulan dibandingkan Arduino Uno. ESP32 dilengkapi dengan fitur Wi-Fi dan Bluetooth bawaan, sehingga lebih mendukung aplikasi IoT yang memerlukan pengiriman data secara nirkabel tanpa memerlukan modul tambahan. Selain itu, ESP32 memiliki prosesor dual-core dengan kecepatan hingga 240 MHz dan memori yang lebih besar, memungkinkan pengolahan data yang lebih cepat dan mendukung algoritma yang lebih kompleks dibandingkan Arduino Uno, yang hanya memiliki prosesor single-core 16 MHz dan memori terbatas.

### 2.1.3 MPX5700AP

MPX5700AP adalah sensor tekanan terpadu yang mengadopsi teknologi *piezoresistive*. Sensor ini dirancang khusus untuk memberikan output yang akurat dan sesuai dengan tekanan

yang diberikan. Sensor ini cocok untuk berbagai aplikasi, terutama ketika digunakan dengan mikrokontroler atau mikroprosesor yang memiliki kemampuan input analog ke digital.



Gambar 2. 3 MPX5700AP

Gambar 2. 3 adalah gambar dari sensor MPX5700AP yang memiliki spesifikasi sebagai berikut :

- a. Jenis: sensor tekanan terintegrasi
- b. Teknologi: piezoresistive
- c. Rentang tekanan: 0 hingga 700 kPa (0 hingga 101,5 psi)
- d. Output: 0,2 hingga 4,7 V
- e. Akurasi:  $\pm 2,5\%$
- f. Tegangan pasokan: 4,75 hingga 5,25 V
- g. Ukuran port: 0,19 inci (4,93 mm) tabung
- h. Gaya port: berduri
- i. Jenis paket: 6 pin SIP (Single In-line Package)
- j. Dimensi paket: 17,78 mm x 29,47 mm x 8,01 mm
- k. Jumlah terminal: 6
- l. Pitch: 2,54 mm
- m. Temperatur operasi: -40 hingga 125 derajat Celsius
- n. Kemasan: reel, potongan, atau tabung

### 2.1.4 Pompa Motor DC

Sebuah motor arus searah, juga dikenal sebagai motor DC, adalah perangkat yang dapat menghasilkan energi mekanik dari energi listrik. Salah satu dari dua jenis motor dasar adalah motor arus bolak-balik (AC). Motor DC terdiri dari beberapa komponen, seperti komutator dengan sikat, armature, rotor, dan stator. Motor DC adalah jenis motor yang paling umum dan sering digunakan dalam peralatan seperti pompa oksigen pada aquarium. Prinsip kerja motor DC terletak pada aliran arus melalui armature atau kumparan yang ditempatkan antara kutub utara dan selatan magnet. Medan magnet yang dihasilkan oleh armature berinteraksi dengan medan magnet lainnya dan menghasilkan torsi pada motor. Dalam motor DC, magnet biasanya terletak pada stator, sedangkan armature ditempatkan pada rotor, dan komutator berperan dalam mengalihkan arus dari satu kumparan ke kumparan lainnya. Komutator berfungsi untuk menghubungkan sumber daya listrik dari stator ke armature melalui penggunaan sikat atau batang konduktif. Adapun gambar dari Pompa Motor DC bisa dilihat seperti Gambar 2. 4.



Gambar 2. 4 Pompa Motor DC

### 2.1.5 Solenoid

*Solenoid* adalah jenis elektromagnet yang panjangnya jauh lebih besar daripada diameternya. *Solenoid* dapat membuat medan magnet yang dapat disesuaikan. Medan magnet yang dihasilkan oleh *solenoid* dapat digunakan dalam berbagai aplikasi. Ini termasuk dalam katup *solenoid* dan perangkat yang mendeteksi dan memantau tekanan darah, seperti *sphygmomanometer*. Ketika arus listrik mengalir melalui kumparan kawat, *solenoid* dapat menciptakan medan magnet yang hampir seragam di suatu area tertentu. Seringkali, inti besi ditambahkan ke dalam *solenoid* untuk meningkatkan daya magnetnya. *Solenoid* dapat

terhubung ke sumber daya listrik dan dikontrol melalui mikrokontroler untuk mengatur aliran cairan atau gas untuk tujuan tertentu. *Solenoid* juga digunakan dalam berbagai aplikasi lain, seperti perangkat bantu pernapasan dan pendengaran. Adapun gambar dari *solenoid* bisa dilihat seperti Gambar 2. 5.



Gambar 2. 5 Solenoid

### 2.1.6 Manset Cuff

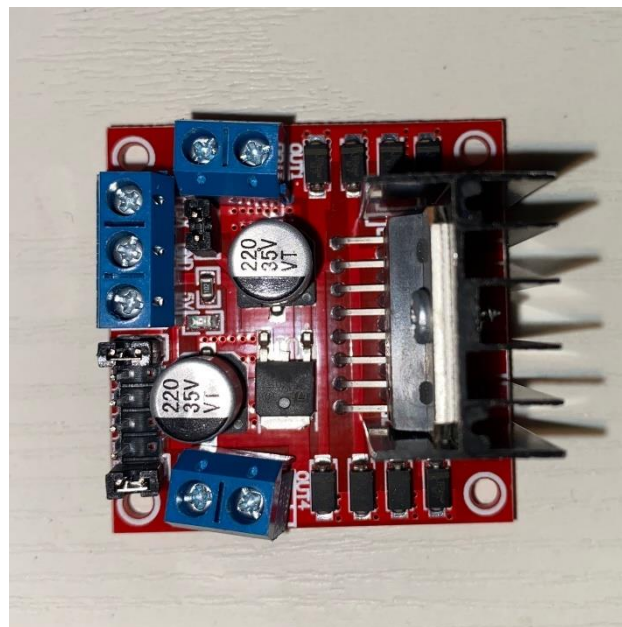
Manset *cuff* adalah bagian dari alat pengukur tekanan darah yang digunakan untuk mengukur tekanan darah seseorang. Ketika tekanan darah diukur, manset *cuff* biasanya terdiri dari kantong udara yang dapat diisi dengan udara yang terbuat dari karet atau nilon, serta lapisan kain yang melilit lengan atas orang yang sedang diukur. Manset *cuff* digunakan untuk menekan arteri brakialis lengan atas, menghentikan aliran darah ke lengan atas dan mengukur tekanan darah. Untuk memastikan hasil pengukuran yang akurat, sangat penting bahwa manset *cuff* dipasang dengan benar dan sesuai dengan ukuran lengan atas orang yang diukur. Hasil pengukuran tekanan darah dapat salah jika manset *cuff* tidak sesuai dengan ukuran lengan. Selain itu, manset *cuff* harus dikenakan dengan ketat, tetapi tidak terlalu ketat sehingga memungkinkan pengukuran tekanan darah yang tepat. Perangkat tensi darah digital dan manual keduanya dapat menggunakan manset *cuff*. Perangkat tensi darah digital menggunakan pompa otomatis untuk memberi udara ke manset *cuff*. Adapun gambar dari Manset *cuff* bisa dilihat seperti Gambar 2. 6.



Gambar 2. 6 Manset *cuff*

### 2.1.7 Motor Driver L298N

Motor *Driver* L298N adalah modul *driver* motor dual H-Bridge yang sering digunakan dalam aplikasi pengendalian motor DC dan *stepper* motor. Ini adalah salah satu IC (*Integrated Circuit*) yang populer digunakan dalam berbagai proyek elektronik, terutama dalam robotika dan kendaraan otomatisasi. Motor *Driver* L298N merupakan dual H-Bridge yang berarti L298N dapat mengontrol dua motor DC sekaligus, pada project ini Motor *Driver* L298N digunakan untuk mengontrol satu motor DC dan satu *solenoid*.



Gambar 2. 7 Motor Driver L298N

Gambar 2. 7 adalah gambar dari sensor Motor *Driver* L298N yang memiliki spesifikasi sebagai berikut :

- Tegangan Supply Logika ( $V_{ss}$ ): 4.5V hingga 7V.
- Tegangan Supply Motor ( $V_{cc}$ ): Hingga 46V.
- Arus Maksimum (Per Channel): Hingga 2A kontinu, 3A puncak (tanpa pendingin tambahan).
- Tegangan Drop (Voltage Drop): Sekitar 2.5V saat arus maksimum.
- Tegangan High-Level Input ( $V_{in\ High}$ ): Minimal 2.3V.
- Tegangan Low-Level Input ( $V_{in\ Low}$ ): Maksimal 1.5V.
- Tegangan Keluaran High-Level ( $V_{oh}$ ): Minimal  $V_{cc} - 2.5V$ .
- Tegangan Keluaran Low-Level ( $V_{ol}$ ): Maksimal 1.5V.

### 2.1.8 Blynk

Blynk adalah platform *Internet of Things* (IoT) yang sangat memudahkan dalam pengembangan proyek IoT dengan menyediakan kemampuan untuk mengendalikan perangkat keras dari jarak jauh melalui aplikasi *smartphone*. Platform ini mendukung berbagai jenis mikrokontroler, termasuk ESP32, dan menawarkan antarmuka yang intuitif untuk menghubungkan perangkat keras dengan aplikasi *mobile*. Dengan menggunakan protokol Blynk memungkinkan sensor atau aktuator apa pun yang terhubung ke ESP32 untuk mengirimkan dan memproses data ke *database cloud* Blynk (Irianto, 2023).

Dengan Blynk, pengguna dapat dengan mudah membuat antarmuka pengguna (UI) yang kustom, menarik, dan mudah digunakan, bahkan oleh orang yang tidak memiliki pengalaman teknis. Beragam *widget* seperti tombol, *slider*, *display*, grafik, dan banyak lagi, dapat ditambahkan oleh pengguna untuk memperkaya fungsionalitas dan interaktivitas proyek IoT mereka.

### 2.1.9 Telegram

Telegram merupakan aplikasi pesan yang memungkinkan pengguna untuk berkomunikasi kepada pengguna lainnya berupa teks, gambar, video, suara, dan *file*. Telegram sendiri adalah aplikasi berbasis *cloud* yang berarti semua pesan yang dikirim dan diterima akan tersimpan pada server Telegram.

Telegram juga menyediakan penggunaan *bot* yang merupakan sebuah program yang dapat memiliki berbagai fungsi yang salah satunya adalah menampilkan hasil pengukuran tekanan darah. Pembuatan *bot* dilakukan menggunakan *API* yang disediakan oleh Telegram. Telegram

memiliki beberapa keunggulan yaitu keamanan yang kuat serta dapat digunakan *multi platform* atau dapat digunakan di berbagai *device*.

### 2.1.10 Kalibrasi

Kalibrasi adalah proses pengaturan dan pengujian alat ukur untuk memastikan bahwa hasil pengukuran yang dihasilkan sesuai dengan standar yang telah ditetapkan. Tujuan utama kalibrasi adalah meningkatkan keakuratan dan konsistensi alat ukur dengan membandingkan hasil pengukurannya terhadap nilai referensi atau standar yang diketahui. Proses ini sangat penting untuk memastikan bahwa data yang dihasilkan dapat diandalkan, terutama dalam penelitian atau aplikasi yang memerlukan presisi tinggi.

Pada perangkat elektronik seperti sensor tekanan darah, kalibrasi dilakukan dengan menguji sensor terhadap tekanan yang diketahui dan kemudian menyesuaikan parameter pengukuran agar sesuai dengan nilai referensi tersebut. Salah satu model kalibrasi adalah kalibrasi linier. Kalibrasi linier digunakan untuk mengukur tekanan darah, kolesterol, suhu, dan lain-lain serta percobaan kalibrasi biasanya ditandai dengan mengamati dua variabel, X dan Y, melalui fungsi yang telah diketahui (Widyaiswara Hairil & Islamiyati, 2015). Model ini digunakan untuk menyesuaikan nilai yang dikeluarkan sensor dan nilai mmHg yang merupakan satuan untuk tekanan darah serta hanya sebatas menyesuaikan nilai keluaran sensor. Untuk evaluasi keakuratan dibutuhkan kalibrasi lanjutan.

Kalibrasi lanjutan dapat dilakukan melalui perangkat simulator. Perangkat simulator dikenal sebagai Simulator Tekanan Darah Non-Invasif (*NIBP Simulator*) dan harga perangkat ini cukup mahal (mencapai 141 juta rupiah untuk simulator merek FLUKE Biomedical, Model BPPM2L / ECG-SHK pada November 2017) (DARWONGSO et al., 2019).

## 2.2 Kajian Literatur

Sistem perawatan kesehatan di seluruh dunia menghadapi tantangan besar karena harapan hidup yang tinggi. Untuk memberikan perawatan yang efektif kepada masyarakat, diperlukan pemantauan kesehatan yang efektif untuk mengidentifikasi potensi masalah kesehatan, termasuk penyakit hipertensi. Teknologi *Internet of Things* (IoT) telah mengubah cara kita melihat dan mengelola kesehatan dalam beberapa waktu belakangan. IoT memungkinkan pengembangan sistem pemantauan yang memberikan data historis dan *real-time* yang membantu dalam diagnosis dini serta mengurangi risiko terjadinya komplikasi. Pada

Tabel 2.1 merupakan beberapa penelitian terdahulu yang peneliti jadikan referensi dalam melakukan penelitian ini.

Tabel 2. 1 Kajian Literatur

No	Judul	Tahun	Mikrokontroler	Penulis	sensor	Evaluasi
1	Rancang Alat Pengukur Tekanan Darah Otomatis Berbasis Internet Of Things	2023	Arduino uno dan ESP 8266	Ridho Ashshiddiq, Budi Rahmadya	MPX10DP	Pada penelitian ini, sistem memompa tekanan udara secara otomatis pada manset cuff yang dijalankan oleh motor DC, kemudian memberikan tekanan udara pada lengan pengguna yang merupakan proses pada pengukuran nilai tekanan darah. Sistem ini menggunakan rumus perhitungan <i>Analog Digital Converter</i> (ADC) untuk mendapatkan hasil pengukuran dengan tingkat akurasi 95,4%.
2	Sistem Pendeteksi Tekanan Darah dan Suhu Tubuh Portabel Menggunakan Protokol MQTT	2021	ESP8266	Ginjar Suwasono Adi, Ferry Satria, Krisna Gumilar	MPX5700 AP, AMG8833	Pada sistem ini terdapat dua sensor yang digunakan untuk mengukur tekanan darah dan suhu tubuh. Sistem ini juga menggunakan protokol MQTT yang berfungsi sebagai penghubung komunikasi antara sensor dan juga perangkat seluler.
3	Rancang Bangun Monitoring Tekanan Darah, Gula Darah Non-Invasif, Berat, Dan Suhu Badan	2021	ESP8266	Muhammad Nur Fariz, Jamaaludin	MAX30100, DS18B20, sensor Load Cell	Sistem ini dapat mengukur tekanan darah, gula darah non-Invasif, berat, dan suhu tubuh. Tingkat akurasi setiap sensor berfungsi sangat baik dengan

	Berbasis Internet Of Things					tingkat akurasi sensor MAX30100 dalam mengukur gula darah mencapai 93,21%, mengukur tekanan darah(Systole) mencapai 94,78%, mengukur tekanan darah(diastole) mencapai 93,37 %, Sensor Load Cell atau Sensor Berat Badan mencapai 98,27% dan Sensor Ds18b20 (suhu) mencapai 96,55 %. Sistem ini menggunakan NodeMCU sebagai mikrokontroler yang berfungsi untuk mengolah data yang berasal dari sensor untuk ditampilkan pada LCD dan <i>smartphone</i> .
4	Penerapan Internet of Things Pada Aplikasi Alat Deteksi dan Monitoring Tekanan Darah	2023	ESP8266	Winarno Fadjar Bastari , Akbar Sujiwa , Rizky Setyobudi	MPX5050 GP	Sistem ini menggunakan NodeMCU ESP8266 sebagai mikrokontroler dan dilengkapi dengan sensor MPX5050GP yang berfungsi untuk mengukur tekanan darah yang hasilnya akan ditampilkan melalui aplikasi android serta FireBase sebagai web server. <i>Error</i> tertinggi pada pengujian nilai <i>systole</i> adalah 3% dan <i>diastole</i> sebesar 4%.
5	Desain Alat Monitoring Real Time Suhu Tubuh, Detak Jantung dan Tekanan Darah secara Jarak Jauh	2022	Raspberry Pie	Haris Isyanto , Andri Syahrul Wahid , Wahyu Ibrahim	DS18B20, pulse sensor, MPX5050 GP	Sistem ini menggunakan sensor DS18B20 sebagai sensor untuk mengukur suhu tubuh, pulse sensor sebagai sensor untuk mengukur detak jantung, serta MPX5050GP sebagai

	melalui Smartphone berbasis Internet of Things Smart Healthcare					sensor tekanan darah. Hasil dari pengukuran kemudian diolah oleh raspberry pi sebagai mikrokontroler yang dikonversi sesuai pengukuran masing-masing sensor. Untuk tekanan darah, nilai yang didapat dari sensor berupa sinyal analog yang diteruskan ke ADS1115 yang berfungsi sebagai pengubah sinyal analog menjadi digital dan dikonversi menjadi satuan mmHg. Setelah itu data ditampilkan pada LCD dan dikirimkan ke server ubidots serta data disimpan menjadi datalog.
--	---	--	--	--	--	--

Berdasarkan kajian terhadap berbagai penelitian terkait sistem pemantauan tekanan darah berbasis Internet of Things (IoT), terdapat beragam komponen teknologi yang digunakan, baik dalam jenis sensor maupun perangkat mikrokontroler. Sensor MPX5700AP, MPX5050GP, dan MPX10DP yang digunakan pada penelitian terdahulu menunjukkan akurasi pengukuran tekanan darah yang baik. Penelitian terdahulu juga menampilkan hasil yang dapat diintegrasikan dengan platform Firebase dan Ubidots untuk penyimpanan dan analisis data. Pada penelitian terdahulu juga banyak komponen berbeda yang digunakan dan juga beberapa penelitian menggabungkan beberapa sensor yang dapat mengukur beberapa macam hal seperti suhu tubuh dan denyut jantung. Dari sisi mikrokontroler, ESP8266 menjadi pilihan utama pada sebagian besar penelitian karena mendukung konektivitas nirkabel bawaan yang andal.

Pada penelitian ini, peneliti menggunakan mikrokontroler ESP32 karena performanya lebih baik dibandingkan ESP8266. ESP32 merupakan penerus dari ESP8266 atau merupakan *upgrade*. Mikrokontroler ini dapat menjadi alternatif yang unggul, terutama jika sistem membutuhkan pengolahan data lebih cepat atau integrasi ke perangkat tambahan.

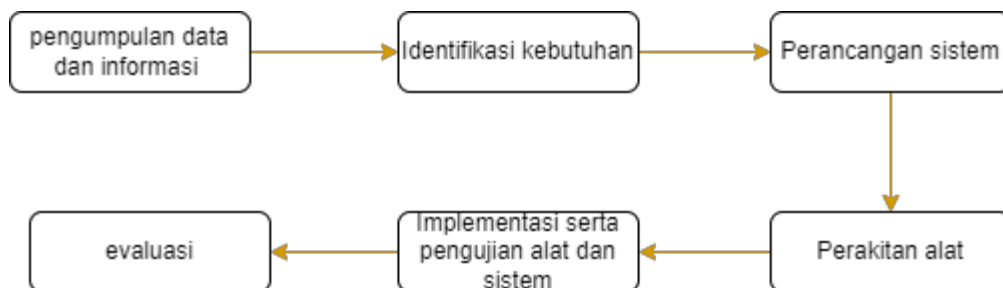
Selain itu, peliti juga memilih sensor MPX5700AP karena sensor ini yang paling mudah ditemukan dipasaran dan juga harga yang paling terjangkau dengan tingkat akurasi yang tidak jauh berbeda dari sensor lainnya.

Pada penelitian ini digunakan Blynk dan Telegram sebagai tempat monitoring tekanan darah karena proses integrasinya yang tidak terlalu rumit serta tampilannya yang memanjakan mata sehingga mudah untuk dilihat. Pada penelitian terdahulu ada yang menggunakan Firebase sebagai tempat penyimpanan data, tetapi harus membuat aplikasi tersendiri untuk menampilkan datanya sehingga mempersulit prosesnya. Penggunaan dua aplikasi monitoring ditujukan untuk pengguna utama dan juga wali dari pengguna untuk melihat riwayat hasil pengukurannya. Di sini peneliti mencoba memfokuskan hanya pada pengukuran tekanan darah tanpa menambah sensor lain sehingga alat dapat digunakan dengan lebih mudah.

## BAB III METODOLOGI PENELITIAN

### 3.1 Tahapan Pengembangan Sistem

Untuk menyelesaikan penelitian mengenai pemantauan tekanan darah berbasis *internet of things* dibutuhkan beberapa proses dan tahapan pengerjaan. Proses dan tahapan pengerjaan dapat dilihat pada Gambar 3. 1 di bawah.



Gambar 3. 1 Tahapan pengerjaan

a. Proses pengumpulan data dan informasi

Pada proses ini peneliti melakukan pengumpulan data dan informasi yang berguna untuk dijadikan referensi dalam mengerjakan sistem pemantauan tekanan darah berbasis *internet of things*. Proses yang terjadi di sini adalah studi literatur yang bertujuan untuk menyelidiki aspek-aspek yang terkait dengan teori-teori yang relevan. Hal ini bertujuan untuk mendukung dalam proses perancangan sistem.

b. Proses identifikasi kebutuhan

Pada proses ini peneliti melakukan identifikasi terhadap kebutuhan apa saja yang diperlukan dalam membangun sistem seperti alat-alat yang akan digunakan serta skema rangkaian sistem yang menjadi panduan dalam perancangan sistem.

c. Proses perancangan sistem

Pada proses ini peneliti membuat Gambaran umum sistem yang berguna untuk memahami cara kerja alat secara umum, dilanjutkan membuat alur kerja sistem berupa

*flowchart* yang merupakan tahapan dari pengukuran tekanan darah, kemudian membuat skematik sebagai petunjuk dalam merakit alat.

d. Proses perakitan alat

Pada proses ini peneliti menghubungkan berbagai komponen yang digunakan untuk alat yang dirancang sehingga alat dapat di program dan berjalan sesuai yang diharapkan.

e. Proses implementasi serta pengujian alat dan sistem

Pada proses ini peneliti menjelaskan mengenai hasil dari sistem yang dibangun dan melakukan pengujian terhadap sistem yang telah dibangun untuk mengetahui apakah alat sudah bekerja sesuai dengan apa yang diharapkan.

f. Proses evaluasi

Pada proses ini tujuannya adalah melakukan evaluasi menyeluruh terhadap sistem yang dibangun dengan melihat kelebihan dan kekurangannya.

### **3.2 Identifikasi Kebutuhan**

Berdasarkan dari analisis kebutuhan, sistem pemantauan tekanan darah berbasis *internet of things* memiliki beberapa kebutuhan dalam proses perancangan sistem.

#### **3.2.1 Kebutuhan Pengguna**

Kebutuhan pada pengguna dibuat untuk mengenali kebutuhan dari calon pengguna. Dalam melakukan identifikasi kebutuhan pengguna, dilakukan observasi terhadap penggunaan tensimeter yang dijual di pasaran. Ditemukan bahwa alat tensimeter yang saat ini banyak dijual di pasaran umumnya hanya berfungsi untuk mengukur tekanan darah secara manual atau digital tanpa menyediakan fitur pencatatan riwayat hasil pengukuran. Berdasarkan hasil observasi, rata-rata pengguna tidak mencatat hasil pengukuran tekanan darah mereka setelah melakukan pengukuran. Hal ini menyebabkan pengguna sering kali tidak memiliki data riwayat pengukuran tekanan darah yang terdokumentasi dengan baik ketika mereka berkonsultasi atau melakukan pemeriksaan lebih lanjut ke tenaga medis.

Situasi ini dapat menjadi kendala, baik itu pada pengguna dan juga tenaga medis yang membutuhkan data pengukuran tekanan darah dari waktu ke waktu untuk memberikan

diagnosis yang lebih akurat dan memantau kondisi kesehatan pengguna secara berkelanjutan. Ketidakadaan catatan tersebut juga mengakibatkan pengguna tidak mengetahui pola fluktuasi tekanan darah mereka, sehingga dapat meningkatkan risiko terlewatnya tanda-tanda awal dari masalah kesehatan yang lebih serius.

Oleh karena itu, sistem yang dirancang dalam penelitian ini bertujuan untuk mengatasi permasalahan tersebut dengan menyediakan fitur pencatatan otomatis hasil pengukuran tekanan darah. Sistem ini akan mencatat setiap hasil pengukuran tekanan darah, sehingga pengguna dapat memiliki akses mudah terhadap riwayat pengukuran mereka. Dengan adanya fitur ini, pengguna tidak perlu lagi mencatat hasil secara manual, dan mereka dapat dengan mudah memberikan data historis tekanan darah kepada tenaga medis untuk keperluan konsultasi atau analisis lebih lanjut.

Fitur pencatatan otomatis ini diharapkan dapat membantu pengguna dari berbagai kalangan usia yang mahir menggunakan teknologi untuk mengelola data kesehatan mereka dengan lebih baik.

### **3.2.2 Kebutuhan *Input***

Kebutuhan *input* dari sistem pemantauan tekanan darah berbasis *internet of things* adalah data yang berasal dari pengukuran sensor tekanan darah berupa nilai *systole* dan *diastole*.

### **3.2.3 Kebutuhan *Output***

Kebutuhan *output* dari sistem pemantauan tekanan darah berbasis *internet of things* adalah informasi nilai *systole* dan *diastole* yang didapat setelah melakukan pengukuran tekanan darah.

### **3.2.4 Kebutuhan *Hardware***

Kebutuhan *Hardware* dari sistem pemantauan tekanan darah berbasis *internet of things* yaitu:

- a. ESP32
- b. Sensor tekanan MPX5700AP
- c. Solenoid
- d. Motor Driver L298N
- e. Pompa Motor DC

- f. Manset cuff
- g. BreadBoard
- h. Power Supply
- i. Kabel jumper male to male, female to female, male to female
- j. LCD

### 3.2.5 Kebutuhan *Software*

Kebutuhan *software* dari sistem pemantauan tekanan darah berbasis *internet of things* yaitu:

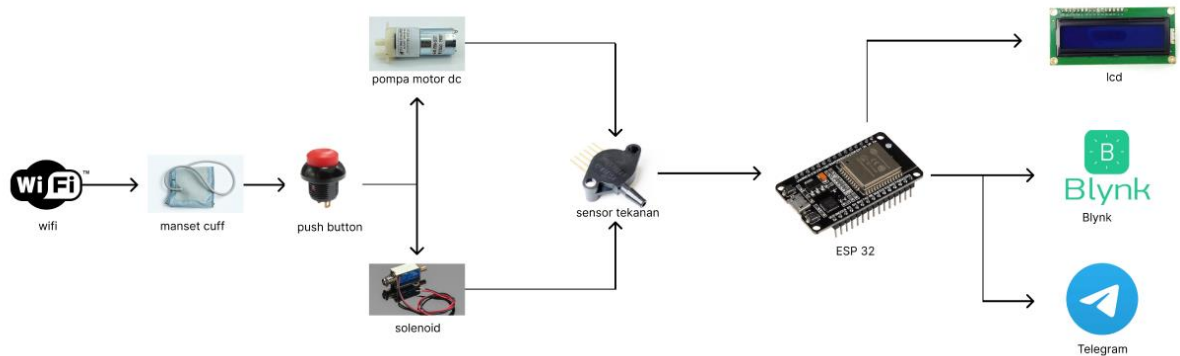
- a. Arduino IDE
- b. Blynk
- c. Fritzing
- d. Telegram

## 3.3 Perancangan Sistem

Alat yang dirancang pada sistem pemantauan tekanan darah berbasis *internet of things* nantinya akan dipasangkan kepada pengguna. Di dalam alat tersebut akan ada sensor yang dapat mengukur nilai *systole* dan *diastole* yang diperlukan untuk mengetahui tekanan darah. Hasil dari pengukuran dapat dilihat pada LCD, Blynk, dan Telegram.

### 3.3.1 Perancangan Umum Sistem

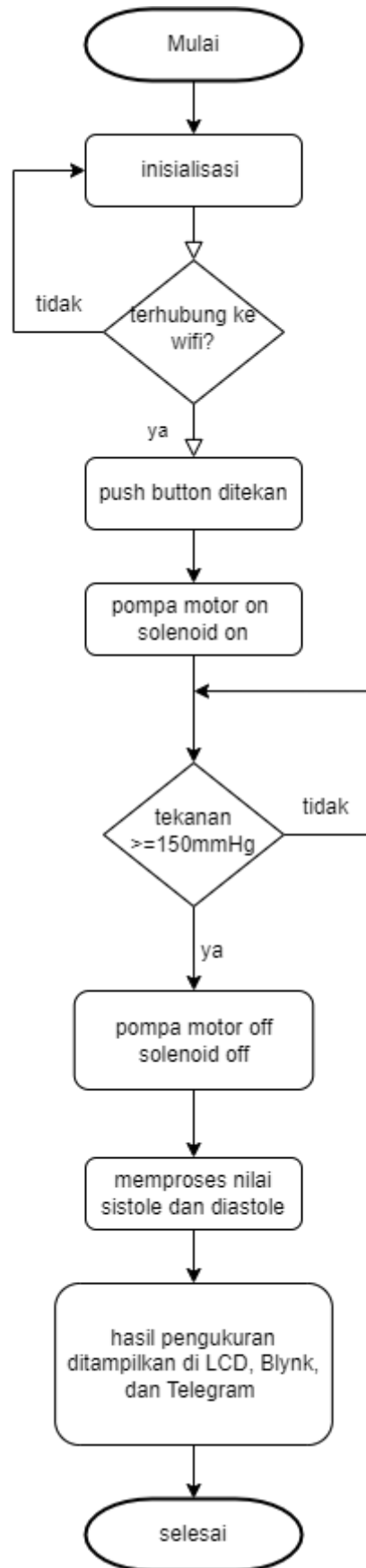
Sistem pemantauan tekanan darah berbasis *internet of things* dibuat untuk memudahkan pengguna dalam melakukan kontrol terhadap tekanan darahnya secara berkala serta melakukan pencatatan terhadap hasil pengukuran tekanan darah. Ini dapat memudahkan dalam pencegahan dini terhadap perubahan tekanan darah yang signifikan atau perubahan yang memerlukan perhatian medis. Sistem ini menggunakan manset *cuff* sebagai media tempat pengambilan data tekanan darah, kemudian setelah mulai menekan *push button* maka pompa motor DC akan berfungsi untuk memompa angin ke manset *cuff* sehingga manset *cuff* dapat mengetat seiring pengukuran tekanan darah oleh alat dengan sensor tekanan darah dan setelah pengisian manset dengan udara mencapai titik tertinggi maka *solenoid* akan mengeluarkan udara tersebut untuk mengempiskan manset. Keluaran dari sensor tekanan darah akan diolah oleh mikrokontroler dan hasilnya ditampilkan melalui LCD, Blynk, dan Telegram. Untuk gambaran umum sistem bisa dilihat pada gambar 3. 2 di bawah.



Gambar 3. 2 Gambaran Umum Sistem

### 3.3.2 Perancangan Alur Kerja Sistem

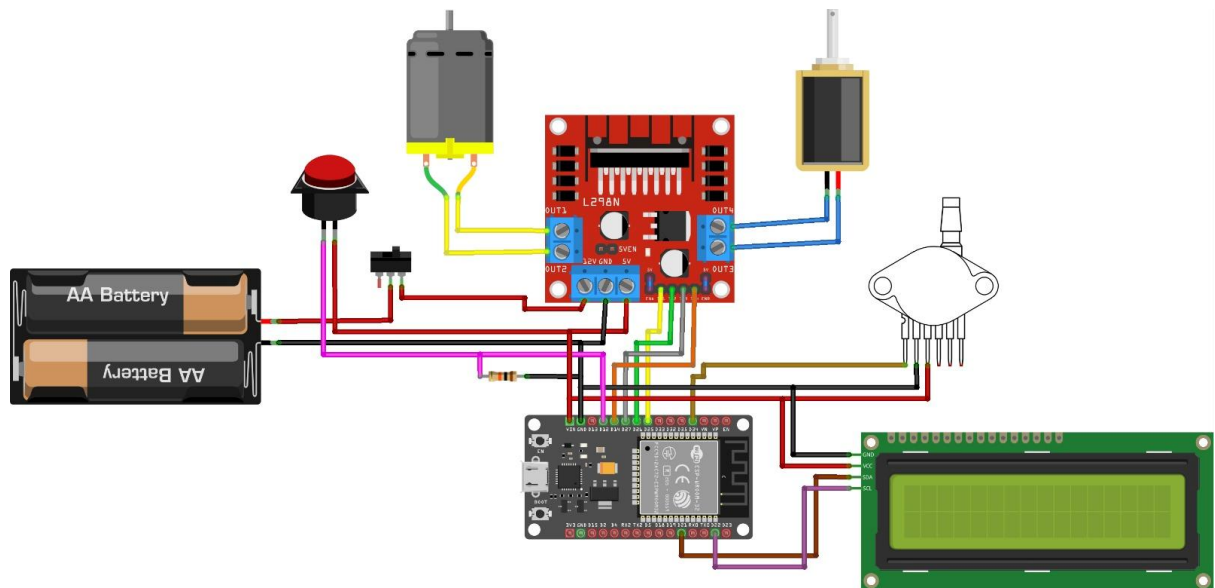
Alur kerja sistem atau *flowchart* dari sistem pemantauan tekanan darah berbasis *internet of things* dimulai dengan inisialisasi ESP32 dan apakah sudah terhubung dengan jaringan Wi-Fi. Jika sudah maka alat dapat dimulai dengan menekan tombol atau *push button* setelah itu pompa motor DC akan menyala dan selanjutnya pengukuran tekanan darah dilakukan. Apabila tekanan lebih besar sama dengan 150 mmHg, maka pompa motor DC akan otomatis mati dan *solenoid* akan membuka untuk mengeluarkan udara pada manset *cuff* dan sensor akan mendeteksi nilai *systole* dan *diastole*. Setelah itu data dari hasil pengukuran *systole* dan *diastole* dapat dilihat pada LCD, Blynk, dan Telegram. Untuk gambar alur kerja sistem atau *flowchart* dari sistem pemantauan tekanan darah berbasis *internet of things* dapat dilihat pada gambar 3.3 di bawah.



Gambar 3. 3 Alur Kerja Sistem

### 3.3.3 Perancangan Komponen *Hardware*

Perancangan komponen perangkat keras pada sistem Pemantauan Tekanan Darah Berbasis *Internet of Things* (IoT) terdiri dari ESP32, motor driver l298n, sensor mpx5700ap, *breadboard*, lcd 16x2 i2c, pompa motor dc, solenoid valve, baterai 7.4v , dan *push button*. Komponen yang ada akan terhubung ke ESP32 sebagai pusat kontrol sistem. Pompa motor dc dan solenoid terhubung ke ESP32 melalui motor driver l298n yang berfungsi untuk menggerakkan kedua komponen tersebut. Motor driver l298n membutuhkan tambahan daya dari luar untuk bekerja, maka digunakan baterai 7.4v. Sensor mpx5700ap terhubung langsung ke ESP32 sebagai sensor yang akan mendeteksi nilai tekanan yang diukur. *Push button* berfungsi untuk memulai proses kerja alat dan lcd 16x2 i2c berfungsi untuk menampilkan hasil pengukuran dari alat. Rancangan rangkaian alat dapat dilihat pada Gambar 3. 4.



Gambar 3. 4 Rancangan Rangkaian Alat

Sumber : Fritzing

Berikut penjelasan dari rancangan rangkaian alat pada gambar 3.4 :

- ESP32 berfungsi otak dari sistem yang menerima dan mengirim data ke database serta memberi perintah untuk menggerakkan komponen lainnya.
- Motor driver l298n berfungsi sebagai penggerak pompa motor dc dan solenoid valve.

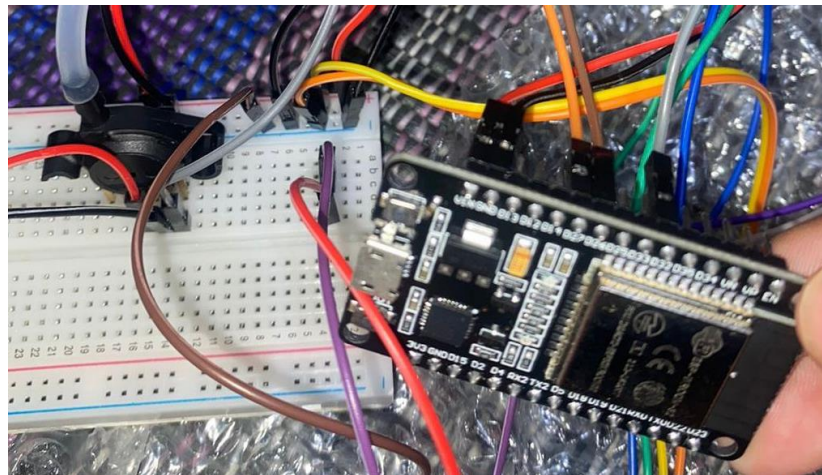
- c. Pompa motor dc berfungsi untuk memompa udara ke manset.
- d. Solenoid berfungsi sebagai tempat keluarnya udara yang telah dipompa ke manset.
- e. Lcd 16x2 i2c berfungsi untuk menampilkan hasil pengukuran *systole* dan *diastole*.
- f. *Push button* bekerja untuk mengaktifkan alat.
- g. Baterai 7.4v bekerja untuk membantu menggerakkan motor driver l298n.

### 3.4 Perakitan Alat

Perakitan alat dari pengembangan sistem dalam penelitian ini adalah sebagai berikut.

#### 3.4.1 Konfigurasi Sensor Tekanan

Sensor tekanan memiliki fungsi untuk menghitung tekanan udara pada manset *cuff* yang telah dipompa yang berupa data analog dan akan dikonversi ke data digital dengan menggunakan mikrokontroler ESP32. Data yang telah di dapat dari manset *cuff* akan diterjemah menjadi tekanan *systole* dan *diastole*. Gambar 3. 5 menunjukkan konfigurasi sensor MPX5700AP dan ESP32.

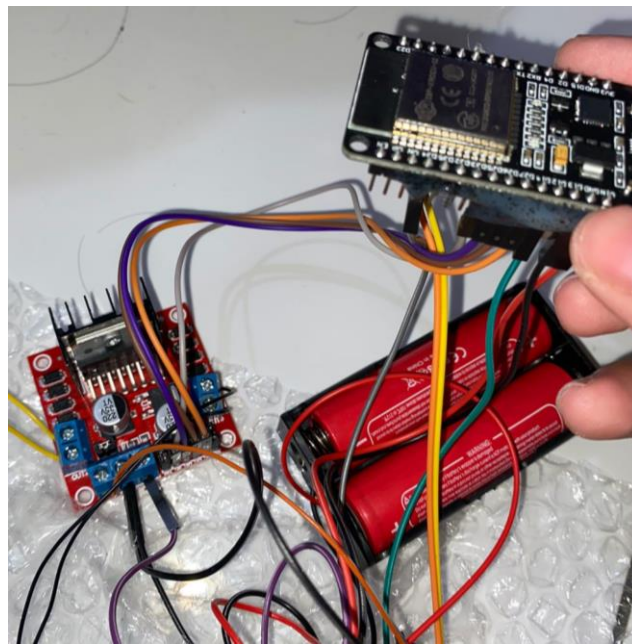


Gambar 3. 5 konfigurasi sensor MPX5700AP dan ESP32

Sensor tekanan MPX5700AP memiliki 3 pin yaitu VCC (suplai daya), GND (ground), dan Vout (analog output). Pin VCC berfungsi sebagai jalur daya/power positif dari sumber energi dan terhubung langsung ke pin VIN pada ESP32, sedangkan pin GND berfungsi sebagai jalur daya/power negatif dari sumber energi juga terhubung langsung ke pin GND pada ESP32, sementara pin Vout berfungsi untuk menghasilkan sinyal analog yang nanti akan di konversi menjadi sinyal digital oleh mikrokontroler, pin ini terhubung ke pin D34 pada ESP32 yang merupakan pin *input* analog yang terhubung langsung ke ADC (*Analog to Digital Converter*).

### 3.4.2 Konfigurasi Motor Driver L298N

Motor *driver* L298N berfungsi untuk menggerakkan pompa motor DC dan *solenoid* agar kedua komponen tersebut dapat berjalan sesuai keinginan. Motor *driver* L298N menggunakan *power* dari baterai dan ESP32 agar bekerja secara optimal. Motor *driver* L298N terhubung langsung dengan ESP32 yang dapat dilihat pada Gambar 3. 6.



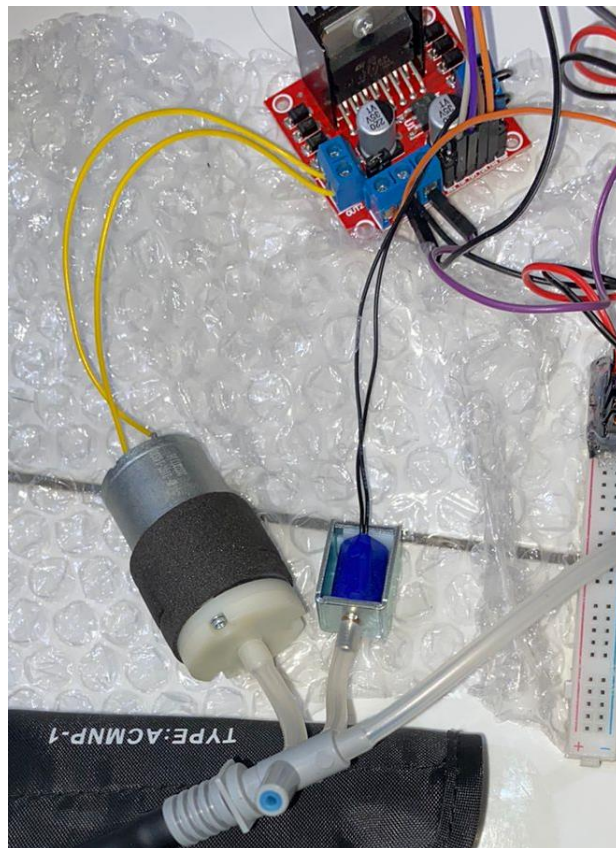
Gambar 3. 6 konfigurasi motor driver L298N

Motor *driver* L298N memiliki beberapa pin yaitu pin IN1, IN2, IN3, IN4, OUT1, OUT2, OUT3, OUT4, 12V, GND, dan 5V. Pin OUT1 sampai OUT4 nantinya akan terhubung pada pompa motor DC dan *solenoid*. Pin IN1 sampai IN4 berfungsi untuk mengontrol komponen yang terhubung ke pin OUT1 sampai OUT4. Pin IN1 terhubung ke pin D25 pada ESP32, pin IN2 terhubung ke pin D26 pada ESP32, pin IN3 terhubung ke pin D27 pada ESP32, dan pin IN4 terhubung ke pin D14 pada ESP32. Pin 5V dan GND terhubung langsung dengan pin VIN dan GND pada ESP32, sementara pin 12V terhubung ke baterai.

### 3.4.3 Konfigurasi Pompa Motor DC dan Solenoid

Pompa motor DC berfungsi untuk memompa udara ke manset *cuff* agar manset *cuff* mengembang dan mengunci lengan, sedangkan *solenoid* berfungsi sebagai tempat keluarnya

udara yang ada di manset *cuff*. Kedua komponen di hubungkan ke motor *driver* L298N yang dapat dilihat pada Gambar 3. 7.

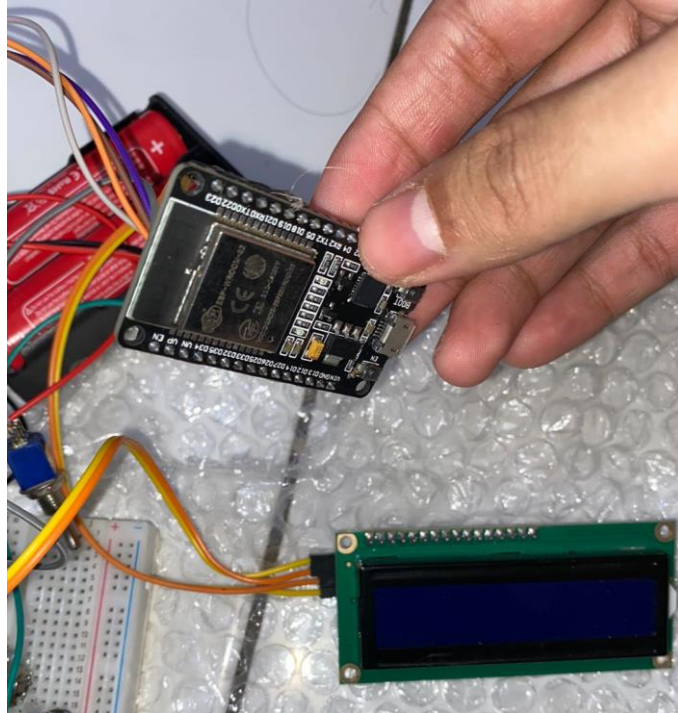


Gambar 3. 7 konfigurasi pompa motor DC dan *solenoid*

Pompa motor DC mempunyai dua pin yang keduanya dihubungkan ke pin Out1 dan Out2 pada motor *driver* l298n. *Solenoid* juga memiliki dua pin yang keduanya dihubungkan ke pin Out3 dan Out4. Untuk menjalankan motor *driver* l298n dibutuhkan *power supply* dari luar berupa baterai karena jika hanya memakai *supply* dari ESP32 akan membuat mikrokontroler tersebut panas.

#### 3.4.4 Konfigurasi LCD

LCD berfungsi untuk menampilkan hasil pengukuran tekanan darah berupa nilai *systole* dan *diastole*. Konfigurasi LCD ke mikrokontroler dapat dilihat pada Gambar 3. 8 .



Gambar 3. 8 konfigurasi LCD

LCD memiliki empat buah pin yaitu SDA (*Serial Data Line*) yang berfungsi untuk mengirim dan menerima data, SCL (*Serial Clock Line*) yang berfungsi mengontrol *timing* dari komunikasi, GND, dan VCC. Untuk pin GND dan VCC pada LCD akan terhubung langsung dengan pin GND dan VIN pada ESP32. Pin SDA akan terhubung dengan pin D21 pada ESP32 dan pin SCL akan terhubung dengan pin D22 pada ESP32.

### 3.5 Implementasi Serta Pengujian Alat dan Sistem

Implementasi serta pengujian alat dan sistem pada sistem Pemantauan Tekanan Darah Berbasis *Internet of Things* (IoT) dilakukan untuk melihat apakah sistem yang dibuat sudah bekerja sesuai yang diinginkan dan dapat mengirimkan data dengan baik. Pengujian dilakukan dengan mengukur langsung tekanan darah untuk melihat hasil *systole* dan *diastole* yang dibaca oleh sensor dan diolah oleh mikrokontroler. Hasil pengujian ditambahkan ke dalam tabel sebagai bukti dari pengujian yang dilakukan. Berikut pengujian yang akan dilakukan:

- a. Kalibrasi dari sensor MPX5700AP untuk menyesuaikan nilai ADC dan mmHg.
- b. Sensor MPX5700AP mengukur tekanan darah berupa nilai *systole* dan *diastole*. Kemudian nilai akan dibandingkan dengan hasil pengukuran menggunakan tensimeter digital.

- c. Pengujian untuk melihat apakah nilai tekanan darah berhasil dikirim ke Blynk dan Telegram.
- d. Pengujian langsung ke pengguna untuk melihat seberapa mudah dan paham pengguna dalam menggunakan sistem.

### 3.5.1 Skenario Pengujian

Pengujian pada sistem Pemantauan Tekanan Darah Berbasis Internet of Things (IoT) bertujuan untuk memastikan bahwa baik perangkat keras maupun perangkat lunak yang digunakan berfungsi sesuai dengan tujuan yang telah dibuat dan menghasilkan keluaran yang sesuai. Pengujian yang dilakukan yaitu kalibrasi dari sensor MPX5700AP untuk menyesuaikan nilai ADC dan mmHg, perbandingan nilai tekanan darah pada prototipe dan alat yang ada di pasaran, pengujian hasil pengukuran pada Blynk dan Telegram, pengujian langsung pada pengguna untuk melihat seberapa mudah dan paham pengguna dalam menggunakan sistem. Dalam melakukan pengujian, diperlukan sebuah skenario agar proses pengujian dapat berjalan dengan lancar. Berikut skenario dari pengujian yang akan dilakukan:

- a. Kalibrasi dari Sensor MPX5700AP

Kalibrasi dilakukan untuk menyesuaikan nilai mmHg dan ADC yang merupakan nilai yang dikeluarkan oleh sensor. Kalibrasi akan dilakukan dengan memasang *sphygmomanometer* pada prototipe untuk melihat nilai mmHg dan menyesuaikan nilainya. Alat akan memompa sampai tekanan yang nilainya ingin diketahui kemudian dicatat untuk melihat peningkatan. Pada pengujian ini nilai diukur setiap peningkatan 20 mmHg dan hasilnya kemudian dicatat pada kolom ADC.

Tabel 3. 1 Skenario Pengujian kalibrasi dari sensor MPX5700AP

<b>mmHg</b>	<b>ADC</b>
0	
20	
40	
60	
80	
100	
120	

140	
160	

b. Perbandingan Nilai pada Prototipe dan Alat yang Ada Di Pasaran

Pengujian perbandingan dilakukan untuk mengetahui hasil pengukuran tekanan darah dari prototipe dan hasil pengukuran alat yang dijual di pasaran. Pengujian dilakukan dengan mengukur tekanan darah dengan prototipe terlebih dahulu kemudian dilanjutkan dengan alat yang dijual di pasaran. Berdasarkan hasil pengukuran tersebut akan dicari berapa persen eror dari tiap pengukuran kemudian akan diakumulasi untuk mengetahui rata-rata eror.

c. Pengujian pada Blynk dan Telegram

Pengujian ini dilakukan untuk mengetahui apakah nilai tekanan darah berhasil masuk ke aplikasi Blynk dan Telegram. Dimulai dengan menjalankan alat sampai mendapatkan hasil pengukuran, kemudian hasil pengukuran akan dicek pada Blynk dan Telegram apakah hasil berhasil masuk atau tidak.

Tabel 3. 2 Skenario Pengujian pada Blynk dan Telegram

<b>Parameter</b>	<b>Hasil Diharapkan</b>
Blynk	Hasil berhasil masuk ke Blynk
Telegram	Hasil berhasil masuk ke Telegram

d. Pengujian pada Pengguna

Pengujian dimulai dengan memberikan demo dari cara kerja alat dan sistem terlebih dahulu kepada pengguna, kemudian pengguna akan menggunakan alat dan sistem secara langsung, setelah selesai pengguna akan diwawancara dengan memberikan beberapa pertanyaan dengan tujuan mengetahui pemahaman pengguna terhadap sistem yang dibangun serta seberapa mudah sistem dapat digunakan oleh pengguna. Berikut daftar pertanyaan yang diajukan:

Tabel 3. 3 Daftar Pertanyaan Pengujian pada Pengguna

<b>Pertanyaan</b>
Apakah sistem memudahkan Anda melakukan pengukuran mandiri ?
Apakah sistem ini dibutuhkan?
Apakah sistem ini sulit digunakan?

### **3.6 Evaluasi**

Pada bagian ini akan dibahas apa saja yang didapat dari pengujian serta apa yang menjadi kelebihan dari sistem serta kekurangan dari sistem.

## BAB IV HASIL DAN PEMBAHASAN

### 4.1 Implementasi Sistem dan Sintaks Program

Pada bagian ini membahas hasil dari alat Pemantauan Tekanan Darah Berbasis *Internet of Things* (IoT) . Dimulai dengan pembahasan mengenai implementasi perangkat keras atau *hardware*, kemudian dilanjutkan dengan pembahasan mengenai implementasi perangkat lunak atau *software*, dan diakhiri dengan pengujian alat dari sistem pemantauan tekanan darah berbasis *internet of things* (IoT) serta membahas kekurangan dan kelebihan dari sistem pemantauan tekanan darah berbasis *internet of things* (IoT).

#### 4.1.1 Implementasi Perangkat Keras (*Hardware*)

Dalam penelitian ini, peneliti membuat sistem pemantauan tekanan darah berbasis *internet of things* (IoT) memanfaatkan komponen *internet of things* (IoT) yaitu mikrokontroler ESP32, sensor MPX5700AP, motor *driver* l298n, LCD 16x2 I2C, *Breadboard*, *solenoid*, pompa motor DC, dua baterai 3.7V , dan *push button*. Pada alat ini sensor MPX5700AP akan terhubung dengan ESP32, kemudian data yang diperoleh dari sensor selama pengukuran akan di olah oleh ESP32 dan hasilnya akan ditampilkan.

Data yang didapat dari sensor merupakan data analog yang berupa tegangan yang akan diubah ke digital oleh mikrokontroler sehingga hasil pengukuran dapat terbaca. *Systole* dan *diastole* merupakan nilai yang diukur pada alat ini. Untuk tampilan luar dari alat dapat dilihat pada Gambar 4. 1.



Gambar 4. 1 Tampilan luar alat

#### 4.1.2 Implementasi Perangkat Lunak (*Software*)

Pada penelitian ini, peneliti menggunakan perangkat lunak Blynk dan Telegram sebagai tempat monitoring tekanan darah pada sistem pemantauan tekanan darah berbasis *internet of things* (IoT).

- a. Blynk digunakan pada penelitian ini karena kemudahan dalam mengakses serta tampilan yang simpel dan memanjakan mata. Blynk nantinya akan digunakan oleh pengguna untuk melihat hasil tekanan darahnya selain di LCD. Untuk tampilan pada aplikasi Blynk dapat dilihat pada Gambar 4. 2 di bawah.



Gambar 4. 2 Tampilan Blynk

- b. Telegram pada penelitian ini digunakan sebagai tempat monitoring berikutnya dari pengguna sistem. Alat ini akan mengirimkan hasil dari pengukuran ke telegram, sehingga pengguna dapat memantau tekanan darah secara berkala. Pada Telegram terdapat tanggal dan jam saat pengukuran dilakukan sehingga pengguna dapat menentukan waktu pengukurannya secara berkala setiap harinya. Kemudian juga ditampilkan peringatan tentang tekanan darah dibawah hasil pengukuran. Tampilan hasil pada Telegram dapat dilihat pada Gambar 4. 3.



### Gambar 4. 3 Tampilan Telegram

## 4.2 Pembahasan Sintaks Program

Pada penelitian ini, peneliti menggunakan Arduino IDE sebagai *tools* untuk melakukan pemrograman. Sintaks adalah program yang berupa rangkaian kode-kode yang berisi perintah agar sistem berjalan sesuai dengan semestinya. Langkah awal yang dilakukan adalah melakukan *install* dari *tools* yang akan digunakan yaitu Arduino IDE. Setelah berhasil, lakukan *install* untuk kebutuhan menulis kode seperti *library* dan *board* untuk ESP32. Setelah itu tuliskan *library* yang telah di unduh tadi, kemudian lakukan integrasi terhadap Blynk dan Telegram yang akan digunakan untuk melihat pengukuran tekanan darah. Sintaks untuk memasukkan *library* dan integrasi Blynk dan Telegram dapat dilihat pada Gambar 4. 4.

```
#define BLYNK_TEMPLATE_ID "TMPL66ek5B5Cq"
#define BLYNK_TEMPLATE_NAME "Tensimeter"
#define BLYNK_AUTH_TOKEN "Rs_J97MOxaJ_01R0wpqL0PY_pQfMphDn"

#define BOT_TOKEN "7171183688:AAFA0z2B_J9b-z7RD5bT37B5r3JAtsfNv6Q"
#define CHAT_ID "959916772"
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <WiFi.h>
#include <BlynkSimpleEsp32.h>
#include <UniversalTelegramBot.h>
#include <WiFiClientSecure.h>
```

Gambar 4. 4 Sintaks *library* dan integrasi Blynk serta Telegram

Setelah itu adalah melakukan konfigurasi terhadap LCD yang nantinya akan digunakan untuk menampilkan informasi tekanan darah yang langsung pada alat. Kemudian membuat kredensial yang berfungsi agar ESP32 bisa terhubung ke internet dan dapat bekerja sesuai fungsinya, apabila ESP32 tidak terhubung dengan jaringan ini, maka alat tidak akan dapat bekerja. Kemudian lakukan definisi terhadap *bot* Telegram agar ESP32 dapat berkomunikasi dengan baik dengan telegram. Selanjutnya yaitu melakukan definisi terhadap pin dari perangkat yang terhubung ke ESP32, pin 34 merupakan pin yang menghubungkan sensor MPX5700AP ke pin ADC milik ESP32, pin 12 merupakan pin yang menghubungkan *push button* dan pin 12 pada ESP32 yang nantinya berfungsi untuk memulai pengukuran tekanan darah pada alat. Pin 25 dan 26 dari ESP32 akan terhubung ke motor driver L298N yang digunakan untuk mengontrol inflasi dan deflasi pada alat pengukur tekanan darah. Untuk pin 27 dan 14 juga terhubung ke motor driver L298N yang fungsinya mengontrol aliran udara pada

alat pengukur tekanan darah. Selanjutnya adalah menginisialisasi variabel untuk menyimpan serta membaca nilai tekanan darah, di sini `numReadings` disetel menjadi 10 yang artinya akan ada 10 pembacaan dari sensor untuk mendukung stabilitas data agar data yang dihasilkan akurat. Sintaks program dapat dilihat pada Gambar 4. 5.

```
// Konfigurasi LCD
LiquidCrystal_I2C lcd(0x27, 16, 2);

// Blynk and Telegram credentials
char ssid[] = "ilhampratama";
char pass[] = "888888888";

// telegram definisi
WiFiClientSecure secured_client;
UniversalTelegramBot bot(BOT_TOKEN, secured_client);

// Pin definisi
#define PRESSURE_PIN 34 // ADC pin untuk sensor MPX5700AP
#define BUTTON_PIN 12 // Pin untuk push button
#define MOTOR_PIN1 25 // L298N IN1
#define MOTOR_PIN2 26 // L298N IN2
#define SOLENOID_PIN1 27 // L298N IN3
#define SOLENOID_PIN2 14 // L298N IN4

// Variabel untuk menyimpan nilai tekanan
float pressure_kPa = 0;
float systolePressure = 0;
float diastolePressure = 0;
bool isMeasuring = false;
bool buttonPressed = false;

const int numReadings = 10;

int readings[numReadings]; // the readings from the analog input
int readIndex = 0; // the index of the current reading
int total = 0; // the running total
int average = 0; // the average
```

Gambar 4. 5 Sintaks konfigurasi lcd,definisi telegram,dan variabel penyimpan nilai tekanan

Kemudian masuk ke program untuk fungsi *setup* yang isinya merupakan inisialisasi dari komunikasi serial, pin, wifi, blynk, dan LCD. Komunikasi serial di sini berfungsi untuk melakukan *debug* dan melihat hasil pengukuran pada *serial* monitor. Kemudian ada proses untuk memeriksa apakah ESP32 sudah terhubung ke wifi, apabila perangkat belum terhubung maka akan ada tanda titik di serial monitor yang muncul setiap 500ms, jika sudah terhubung maka akan menampilkan kalimat yang menandakan wifi sudah terhubung dan akan terlihat alamat *ip* pada serial monitor. Setelah itu masuk ke proses inisialisasi pin, *Pressure\_pin* diatur menjadi *input* untuk membaca data dari sensor MPX5700AP, *button pin* diatur menjadi *input* karena berfungsi untuk mendeteksi apakah tombol sudah ditekan, *motor\_pin1* dan *motor\_pin2*

diatur sebagai *output* untuk mengontrol pompa motor yang digunakan, sementara *solenoid\_pin1* dan *solenoid\_pin2* diatur sebagai *output* untuk mengontrol *solenoid*. Selanjutnya pompa motor dan *solenoid* dibuat *low* untuk mengonfirmasi apakah pompa motor dan *solenoid* dalam kondisi mati saat alat dinyalakan. Proses selanjutnya yaitu menginisialisasi elemen pada *array* dengan nilai 0 sebelum digunakan untuk menyimpan nilai pembacaan tekanan darah. Kemudian adalah inisialisasi LCD untuk menampilkan tampilan awal sebelum alat dijalankan. Sintaks program dapat dilihat pada Gambar 4. 6.

```

void setup() {
  // Inisialisasi komunikasi serial
  Serial.begin(115200);

  // Inisialisasi WiFi dan Blynk
  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
  // attempt to connect to Wifi network:
  Serial.print("Connecting to Wifi SSID ");
  Serial.print(ssid);
  WiFi.begin(ssid, pass);
  secured_client.setCACert(TELEGRAM_CERTIFICATE_ROOT); // Add root
certificate for api.telegram.org
  while (WiFi.status() != WL_CONNECTED)
  {
    Serial.print(".");
    delay(500);
  }
  Serial.print("\nWiFi connected. IP address: ");
  Serial.println(WiFi.localIP());

  // Inisialisasi pin
  pinMode(PRESSURE_PIN, INPUT);
  pinMode(BUTTON_PIN, INPUT);
  pinMode(MOTOR_PIN1, OUTPUT);
  pinMode(MOTOR_PIN2, OUTPUT);
  pinMode(SOLENOID_PIN1, OUTPUT);
  pinMode(SOLENOID_PIN2, OUTPUT);

  digitalWrite(MOTOR_PIN1, LOW);
  digitalWrite(MOTOR_PIN2, LOW);
  digitalWrite(SOLENOID_PIN1, LOW);
  digitalWrite(SOLENOID_PIN2, LOW);

  for (int thisReading = 0; thisReading < numReadings; thisReading++) {
    readings[thisReading] = 0;
  }

  // Inisialisasi LCD
  lcd.begin();
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("Tensimeter");
  delay(2000);
  lcd.clear();
  lcd.clear();
}

```

```
lcd.setCursor(0, 0);
lcd.print("Tek: 00");
```

Gambar 4. 6 Sintaks inisialisasi dari komunikasi serial,pin,wifi,blynk, dan lcd

Selanjutnya masuk ke program utama untuk menjalankan alat, dimulai dengan memanggil fungsi run untuk Blynk yang diperlukan untuk menjaga koneksi dan komunikasi dengan server Blynk tetap berjalan. Fungsi ini harus dipanggil secara berulang di dalam *loop* agar perangkat tetap terhubung dengan Blynk dan dapat menerima serta mengirim data. Kemudian dilanjutkan dengan membaca status *push button*, apabila kondisi terpenuhi maka pompa motor akan mulai mengisi udara ke manset. Kode program dapat dilihat pada Gambar 4. 7.

```
void loop() {
  Blynk.run();

  // Membaca status tombol
  if (digitalRead(BUTTON_PIN) == HIGH && !buttonPressed) {
    buttonPressed = true;
    isMeasuring = true;
    systolePressure = 0;
    diastolePressure = 0;
    inflateCuff(); // Mulai inflasi
    Serial.println("Button pressed, starting measurement.");
  }
}
```

Gambar 4. 7 Sintaks memulai pengukuran

Apabila alat sudah mulai melakukan pengukuran, maka alat akan mulai membaca nilai ADC dengan *total* adalah penjumlahan dari semua pembacaan dalam *array* yang digunakan untuk menghitung rata-rata. Pertama, *total* akan dikurang dengan nilai lama pada indeks agar nilai lama tidak ikut dalam perhitungan rata-rata. Nilai baru akan dibaca oleh sensor dan disimpan di *array*,kemudian ditambahkan ke total. *readIndex* ditambah 1 untuk berpindah ke indeks berikutnya, dan proses ini akan dilakukan berulang sampai batas pembacaan. Apabila *readIndex* telah mencapai batas maksimal pembacaan yang ditentukan oleh *numReadings*, maka indeks diatur kembali ke 0. Ini berguna agar sistem selalu melakukan pembacaan terbaru secara terus-menerus, dengan mempertahankan data pembacaan sebelumnya. Setelah selesai membaca data dari sensor, nilai rata-rata akan dihitung dengan membagikan *total* dan *numReadings* yang berguna untuk meminimalisir fluktuasi dari sensor. Setelah itu rata-rata yang masih berbentuk nilai ADC dikonversi menjadi nilai tekanan dengan satuan mmHg dengan menggunakan persamaan yang didapat dari kalibrasi linier yang telah di uji coba sebelumnya dengan menyesuaikan nilai tekanan ADC dengan tekanan mmHg yang merupakan

satuan standar dalam melakukan pengukuran tekanan darah. Untuk nilai 877.6 merupakan *offset* atau titik awal untuk tekanan nol, dan 1.106 adalah faktor skala yang mengubah perubahan nilai ADC menjadi perubahan tekanan dalam mmHg. Sintaks program dapat dilihat pada Gambar 4. 8.

```

if (isMeasuring) {
  // Membaca nilai ADC dari sensor
  total = total - readings[readIndex];
  readings[readIndex] = analogRead(PRESSURE_PIN);
  total = total + readings[readIndex];
  readIndex = readIndex + 1;

  if (readIndex >= numReadings) {
    readIndex = 0;
  }

  average = total / numReadings;
  Serial.println(average);
  delay(1);

  // Kalibrasi linier data ADC ke mmHg
  float pressure_mmHg = (average - 877.6) / 1.106;

```

Gambar 4. 8 Sintaks pembacaan nilai

Jika nilai tekanan sudah terbaca, maka selanjutnya adalah mendeteksi nilai *systole* dan *diastole* dengan memanggil fungsi *detectSystoleDiastole* berdasarkan nilai *pressure\_mmHg*. Setelah nilai *systole* dan *diastole* didapatkan, maka nilai akan ditampilkan di LCD dan juga alat akan mengirim nilai tekanan ke *dashboard* Blynk dan juga Telegram. Sintaks program dapat dilihat pada Gambar 4. 9.

```

// Deteksi sistole dan diastole
detectSystoleDiastole(pressure_mmHg);

// Menampilkan nilai tekanan pada LCD
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Tek: ");
lcd.print(pressure_mmHg);
lcd.print(" mmHg");

Blynk.virtualWrite(V0, pressure_mmHg);

if (systolePressure > 0 && diastolePressure > 0) {
  lcd.setCursor(0, 1);
  lcd.print(systolePressure);
  lcd.print("/");
  lcd.print(diastolePressure);
  Serial.print("Systole: ");
  Serial.print(systolePressure);
  Serial.print(" Diastole: ");
  Serial.println(diastolePressure);

```

```

// Kirim hasil ke Blynk
Blynk.virtualWrite(V1, systolePressure);
Blynk.virtualWrite(V2, diastolePressure);

// Kirim hasil ke Telegram
String message = "Systole: " + String(systolePressure) + "
mmHg\nDiastole: " + String(diastolePressure) + " mmHg\n\n*Jika Tekanan darah
anda di atas 130mmHg maka termasuk tinggi.\n Segera istirahat atau kunjungi
tenaga medis";
bot.sendMessage(CHAT_ID, message, "");

// Reset ESP32 setelah mengirim data
delay(2000); // Optional: delay to ensure data is sent before restart
ESP.restart();
}

delay(500);
}
}

```

Gambar 4. 9 Sintaks pengiriman hasil ke Blynk dan Telegram

Selanjutnya adalah kode untuk fungsi *inflateCuff* dan *deflateCuff* yang berfungsi untuk menyalakan pompa dan mematikan pompa. Fungsi ini digunakan pada program utama saat alat memulai pengukuran dan setelah pengukuran. Sintaks program dapat dilihat pada Gambar 4.9.

```

void inflateCuff() {
// pompa on dan solenoid on
digitalWrite(MOTOR_PIN1, HIGH);
digitalWrite(MOTOR_PIN2, LOW);
digitalWrite(SOLENÖID_PIN1, HIGH);
digitalWrite(SOLENÖID_PIN2, LOW);
Serial.println("Inflating cuff.");
}

void deflateCuff() {
// pompa off dan solenoid off
digitalWrite(MOTOR_PIN1, LOW);
digitalWrite(MOTOR_PIN2, LOW);
digitalWrite(SOLENÖID_PIN1, LOW);
digitalWrite(SOLENÖID_PIN2, LOW);
Serial.println("Deflating cuff.");
}
}

```

Gambar 4. 10 Sintaks fungsi *inflateCuff* dan *deflateCuff*

Selanjutnya ada kode untuk fungsi *detectSystoleDiastole* yang berfungsi untuk mendeteksi nilai dari tekanan *systole* dan *diastole*. Fungsi ini dipanggil pada program utama yang nantinya akan bekerja sesuai dengan perubahan pada nilai tekanan *pressure\_mmHg* dengan logika yang dirancang untuk mengenali saat terjadinya *sytole* dan *diastole*. Variabel *previousPressure* berguna untuk menyimpan nilai tekanan dari hasil pembacaan sebelumnya. Nilai ini digunakan untuk membandingkan perubahan tekanan saat ini dengan tekanan sebelumnya. Kemudian ada

variabel *systoleDetected* dengan fungsi untuk menyimpan informasi apakah tekanan *systole* sudah terdeteksi atau belum. Selanjutnya ada *diastoleDetected* untuk menyimpan informasi apakah tekanan *diastole* sudah terdeteksi atau belum. Sementara *stableCount* merupakan sebuah variabel penghitung yang digunakan untuk memastikan bahwa pengukuran tidak mengalami fluktuasi, sehingga tekanan *systole* dan *diastole* dapat ditentukan dengan akurat. Kemudian dalam prosesnya apabila tekanan sudah mencapai nilai 150 mmHg atau lebih, maka akan memanggil fungsi *deflateCuff* untuk mematikan pompa agar mansett mulai mengempis. Disini nilai 150 mmHg ditetapkan karena tekanan *systole* normal pada manusia adalah 120 mmHg yang mana 150 mmHg sudah sangat jauh di atas tekanan darah normal.

Selanjutnya adalah untuk mendeteksi *systole* yang terjadi saat tekanan dalam manset mulai turun setelah mencapai titik tertinggi, karena pembuluh darah kembali terbuka. Kondisi ini dicek ketika tekanan mencapai puncak (tekanan sebelumnya lebih tinggi dari tekanan saat ini) dan sebelum mencapai 150 mmHg. Setiap kali tekanan turun dalam kisaran yang diharapkan, *stableCount* akan bertambah. Hal ini untuk memastikan bahwa penurunan tekanan benar-benar terjadi dan bukan karena fluktuasi acak. Jika pengukuran stabil selama tiga kali berturut-turut, nilai *previousPressure* disimpan sebagai *systolePressure*, yang berarti tekanan sistolik telah terdeteksi. Status *systoleDetected* kemudian diubah menjadi true.

Setelah tekanan *systole* terdeteksi (*systoleDetected* = true), fungsi beralih untuk mendeteksi tekanan *diastole*. Tekanan *diastole* diidentifikasi dalam rentang 50-100 mmHg, yang merupakan kisaran normal untuk tekanan diastolik. Sama seperti pada proses deteksi *systole*, jika tekanan berada dalam kisaran yang tepat dan stabil (dengan *stableCount* > 5, yang berarti harus lebih stabil dibandingkan dengan *systole*), maka nilai *diastolePressure* disimpan dari *previousPressure*. Setelah tekanan *diastole* terdeteksi, proses pengukuran dianggap telah selesai. Untuk sintaks program dapat dilihat pada Gambar 4. 11.

```
void detectSystoleDiastole(float pressure_mmHg) {
    static float previousPressure = 0;
    static bool systoleDetected = false;
    static bool diastoleDetected = false;
    static int stableCount = 0;

    if (pressure_mmHg >= 150 && !systoleDetected) {
        // Mulai deflasi ketika tekanan >= 150 mmHg
        deflateCuff();
    }

    if (isMeasuring) {
```

```

    // Deteksi sistole
    if (!systoleDetected && pressure_mmHg < previousPressure &&
previousPressure < 150) {
        stableCount++;
        if (stableCount > 2) {
            systolePressure = previousPressure;
            systoleDetected = true;
            Serial.print("Systole detected: ");
            Serial.println(systolePressure);
        }
    }

    // Deteksi diastole
    if (systoleDetected && !diastoleDetected && pressure_mmHg > 50 &&
pressure_mmHg < 100) {
        stableCount++;
        if (stableCount > 5) {
            diastolePressure = previousPressure;
            diastoleDetected = true;
            isMeasuring = false; // Pengukuran selesai
            // Matikan solenoid valve
            digitalWrite(SOLENOID_PIN1, LOW);
            digitalWrite(SOLENOID_PIN2, LOW);
            buttonPressed = false; // Reset status tombol
            Serial.print("Diastole detected: ");
            Serial.println(diastolePressure);
        }
    }

    previousPressure = pressure_mmHg;
}
}

```

Gambar 4. 11 Sintaks fungsi *detectSystoleDiastole*

### 4.3 Pengujian Perangkat

Pada bagian ini dilakukan pengujian untuk melihat apakah sistem sudah sesuai dengan apa yang diharapkan.

#### 4.3.1 Pengujian Kalibrasi Sensor MPX5700AP

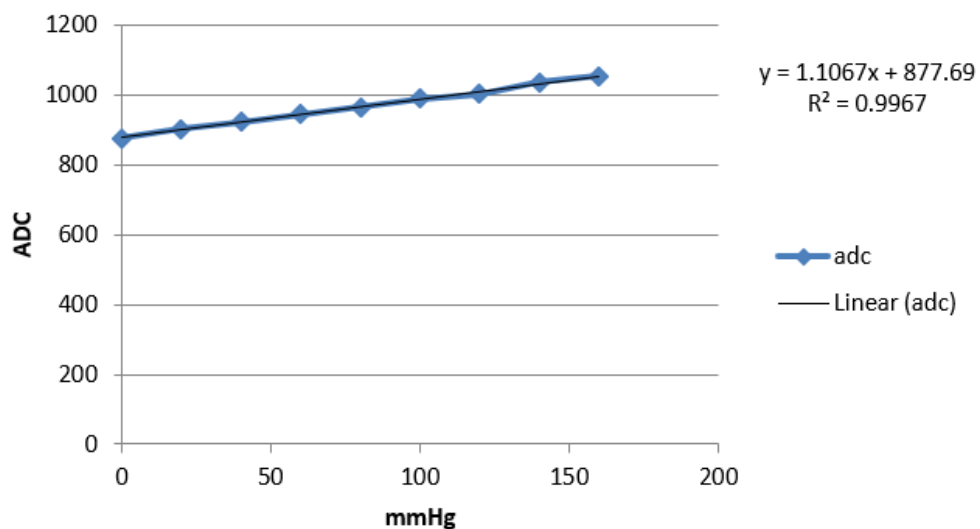
Kalibrasi pada sensor dibutuhkan untuk meminimalisir kesalahan pada pengukuran tekanan darah. Pada tabel 4.1 kalibrasi dilakukan dengan menggunakan *sphygmomanometer* untuk menyesuaikan nilai ADC dengan satuan mmHg yang dimulai dari 0 sampai 160 mmHg.

Tabel 4. 1 kalibrasi sensor

mmHg	ADC
0	875
20	903

40	923
60	944
80	966
100	990
120	1003
140	1037
160	1055

Setelah menyesuaikan nilai mmHg dan ADC kemudian didapat rumus dengan menggunakan persamaan regresi linear yang akan digunakan pada program untuk mengubah nilai ADC yang terbaca dari sensor menjadi nilai mmHg. Variabel  $y$  merupakan nilai ADC sedangkan  $x$  merupakan tekanan dalam mmHg. Nilai 1.1067 merupakan gradien atau kemiringan garis, yang menunjukkan seberapa banyak nilai ADC berubah untuk setiap perubahan 1 mmHg. Nilai 877.69 merupakan *intercept* yang merupakan nilai perubahan pada variabel  $y$  saat variabel  $x$  bernilai 0. nilai  $R^2$  adalah 0.9967 yang menunjukkan tingkat akurasi dari model linear ini sesuai dengan data pengukuran dari sensor. Grafik kalibrasi linear bisa dilihat pada Gambar 4. 12.



Gambar 4. 12 grafik kalibrasi linear sensor

### 4.3.2 Pengujian Sensor MPX5700AP dan Tensimeter Digital

Pengujian dilakukan untuk mengukur seberapa akurat sensor MPX5700AP dalam mengukur tekanan darah. Pengujian dilakukan dengan membandingkan nilai *systole* dan *diastole* yang diukur dari sensor MPX5700AP dengan alat tensimeter digital. Gambar 4.13 merupakan perbandingan pertama yang dilakukan.



Gambar 4. 13 Pengujian sensor

Tabel 4.1 adalah hasil dari pengukuran tekanan darah yang dilakukan dengan membandingkan nilai *systole* dan *diastole* yang diukur dari sensor MPX5700AP dengan alat tensimeter digital.

Tabel 4. 2 Pengujian Sensor

Percobaan	MPX5700AP(mmHg)		Tensimeter Digital(mmHg)		Error(%)	
	<i>Systole</i> (X1)	<i>Diastole</i> (Y1)	<i>Systole</i> (X2)	<i>Diastole</i> (Y2)	<i>systole</i>	<i>diastole</i>
1	136	91	135	83	0,74	9,63
2	134	88	131	82	2,29	7,31
3	126	80	126	79	0	1,26
4	107	70	111	75	3,60	6,67
5	123	75	115	81	6,95	7,40
6	108	67	109	77	0,91	12,98

7	124	87	117	85	5,98	2,35
8	134	78	125	81	7,20	3,70
9	126	76	121	80	4,13	5,00
10	115	78	120	81	4,16	3,70
<i>Average</i>					3,59	6,00

Untuk mendapatkan nilai *error* dari *systole* dan *diastole* , digunakan persamaan yang bisa dilihat pada persamaan 4. 1 dan 4. 2 .

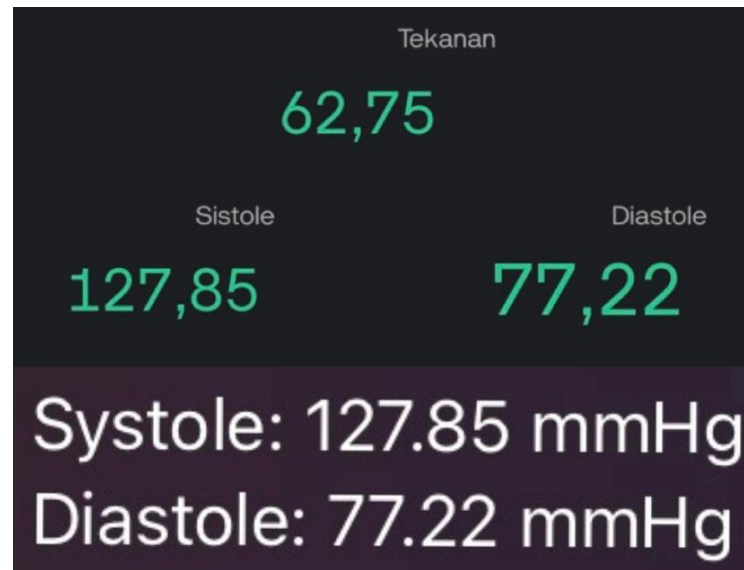
$$error\ systole = \frac{x1 - x2}{x2} . 100\% \quad (4.1)$$

$$error\ diastole = \frac{y1 - y2}{y2} . 100\% \quad (4.2)$$

Hasil pengujian pada tabel 4.2 menunjukkan bahwa terdapat rata-rata *error* 3,59% untuk nilai *systole* dan 6,00% pada nilai *diastole*.

### 4.3.3 Pengujian pada Blynk dan Telegram

Pengujian ini dilakukan untuk memastikan apakah nilai tekanan darah berhasil dikirimkan ke Blynk dan Telegram. Pengujian dilakukan dengan mengukur tekanan darah langsung, setelah tekanan darah berhasil diukur, maka hasil akan langsung terkirim ke Blynk dan Telegram. Perlu diingat pengujian harus dilakukan menggunakan koneksi internet agar data dapat terkirim ke Blynk dan Telegram.



Gambar 4. 14 Pengujian pada Blynk dan Telegram

Tabel 4. 3 Pengujian pada Blynk dan Telegram

No	Parameter	status
1	Blynk	Berhasil
2	Telegram	Berhasil

#### 4.3.4 Pengujian pada Pengguna

Pengujian pada pengguna dilakukan dengan tujuan untuk mengetahui pemahaman pengguna terhadap sistem yang dibangun serta seberapa mudah sistem dapat digunakan oleh pengguna. Pengujian dilakukan dengan memberikan pertanyaan kepada pengguna tentang review dari sistem yang dibangun. Hasil pengujian dapat dilihat pada Tabel 4.4.

Tabel 4. 4 Pengujian Pada Pengguna

Pertanyaan	Jawaban
Apakah sistem memudahkan Anda melakukan pengukuran mandiri ?	Sangat memudahkan karena hasil pengukuran tercatat langsung di <i>handphone</i> .
Apakah sistem ini dibutuhkan?	Ya, karena tidak perlu catat hasil pengukuran di buku karena bisa otomatis.
Apakah sistem ini sulit digunakan?	Sistem mudah digunakan hanya tinggal membuka aplikasi.

Dari jawaban Ibu Enny sebagai pengguna bisa disimpulkan bawah sistem sangat memudahkan bagi pengguna karena tidak perlu repot untuk mencatat hasil pengukuran karena hasil akan otomatis terkirim pada Telegram pengguna. Sistem juga dirancang untuk mudah digunakan oleh pengguna agar tidak banyak proses yang perlu dilakukan sebelum melihat hasil pengukurannya.



Gambar 4. 15 Pengujian pada Pengguna

#### 4.4 Evaluasi

Berdasarkan hasil pengujian, alat pemantauan tekanan darah berbasis IoT yang dikembangkan berfungsi dengan baik dan memberikan hasil yang memuaskan. Alat ini mampu mengukur tekanan darah dengan rata-rata kesalahan sebesar 3,59% pada pengukuran tekanan sistolik dan 6,00% pada tekanan diastolik, jika dibandingkan dengan tensimeter komersial yang ada di pasaran.

Namun, hasil dan keakuratan pengukuran dapat dipengaruhi oleh berbagai faktor eksternal, seperti aktivitas fisik pengguna sebelum pengukuran, sehingga hasilnya mungkin mengalami sedikit fluktuasi dari yang diharapkan. Selain itu, tekanan darah dapat berubah dalam waktu singkat secara alami.

Alat ini memiliki kelebihan yaitu hasil pengukurannya yang tercatat pada aplikasi Blynk dan Telegram yang mana pengguna tidak perlu mencatat hasil pengukurannya secara manual.

Selain kelebihan, alat ini juga punya kelemahan. Salah satu kelemahan utama perangkat ini adalah ketergantungannya pada koneksi internet. Karena alat ini mengirimkan data ke platform Blynk dan Telegram melalui jaringan internet, maka perangkat tidak akan berfungsi dengan baik jika koneksi internet tidak stabil. Kelemahan lainnya yaitu sensor yang kurang stabil saat membaca pengukuran, terkadang terjadi *noise* atau gangguan elektromagnetik dari komponen lain yang menyebabkan fluktuasi tegangan sehingga mengganggu kinerja dari sensor dan pengukuran terkadang tidak terbaca.

Secara keseluruhan, perangkat ini merupakan solusi yang baik untuk memantau tekanan darah secara berkala. Meskipun terdapat kelemahan utama dalam hal ketergantungan pada koneksi internet, hasil pengujian menunjukkan bahwa perangkat ini tetap dapat memberikan data yang akurat dan andal setelah proses kalibrasi dilakukan dengan tepat.

## **BAB V**

### **KESIMPULAN**

#### **5.1 Kesimpulan**

Dari penelitian ini dapat disimpulkan bahwa sistem pemantauan tekanan darah berbasis Internet of Things (IoT) dengan menggunakan sensor MPX5700AP berhasil dirancang dan diterapkan untuk memantau tekanan darah secara rutin. Hasil pengujian menunjukkan bahwa sistem mampu mendeteksi pengukuran tekanan darah dengan akurat, serta memungkinkan pengguna untuk memantau hasilnya tanpa perlu mencatat secara manual. Selain memudahkan pengguna, teknologi ini juga dapat membantu tenaga medis saat pengguna melakukan kontrol, karena hasil pengukuran rutin dapat dilihat pada aplikasi. Dengan dibuatnya sistem ini diharapkan dapat membantu pengguna menjaga kesehatannya dengan mengurangi risiko hipertensi dan hipotensi.

#### **5.2 Saran**

Berdasarkan penelitian yang telah melalui berbagai tahapan, terdapat beberapa hal yang dapat dikembangkan untuk penelitian selanjutnya, antara lain :

a. Ketergantungan pada koneksi internet

Untuk penelitian saat ini sistem sangat bergantung dengan adanya koneksi internet karena pengiriman hasil pengukuran ke Blynk dan Telegram. Penelitian selanjutnya diharapkan untuk mengembangkan alat yang tetap berfungsi saat tidak ada koneksi internet, contohnya dengan membuat penyimpanan data sementara yang dapat diunggah saat koneksi internet telah tersedia.

b. Keakuratan pembacaan pada sensor

Untuk penelitian selanjutnya diharapkan untuk meningkatkan kekuatan pembacaan pada sensor dengan menggunakan sensor yang lebih presisi dan metode kalibrasi yang lebih baik agar kesalahan dalam pengukuran dapat diminimalisir.

c. Notifikasi baterai

Penelitian selanjutnya diharapkan untuk menambahkan fitur notifikasi baterai pada alat, sehingga saat daya baterai sudah sedikit pengguna dapat mengganti atau melakukan pengisian ulang pada baterai.

## DAFTAR PUSTAKA

- Ardutech. (2020). *Mengenal ESP32 Development Kit untuk IoT (Internet of Things)*. Www.Ardutech.Com.<https://www.ardutech.com/mengenal-esp32-development-kituntuk-iot-internet-of-things/>
- Ashshiddiq, R., & Rahmadya, B. (2023). Rancang Alat Pengukur Tekanan Darah Otomatis Berbasis Internet Of Things. *CHIPSET*, 4(01), 23–35. <https://doi.org/10.25077/chipset.4.01.23-35.2023>
- Cahyana, A., Hariyawan, M. Y., Indani, W., & Ramadona, S. (2023). Jurnal Politeknik Caltex Riau Sistem Cerdas Pemantau Kesehatan Pasien Lanjut Usia Berbasis IoT (Hardware). In *Jurnal ELEMENTER* (Vol. 9, Issue 1). <https://jurnal.pcr.ac.id/index.php/elementer>
- DARWONGSO, A., UGHI, F., & HIDAYAT, S. W. (2019). Simulator Tekanan Darah: Minimalisasi Pengaruh Laju Inflasi dan Deflasi Terhadap Simulasi Osilasi. *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, 7(1), 165. <https://doi.org/10.26760/elkomika.v7i1.165>
- Fitriani, N., & Nilamsari, N. (2017). FAKTOR-FAKTOR YANG BERHUBUNGAN DENGAN TEKANAN DARAH PADA PEKERJA SHIFT DAN PEKERJA NON-SHIFT DI PT. X GRESIK. *Journal of Industrial Hygiene and Occupational Health*, 2(1). <https://doi.org/10.21111/jihoh.v2i1.948>
- Irianto, K. D. (2023). Pre-SEMMS: A Design of Prepaid Smart Energy Meter Monitoring System for Household Uses Based on Internet of Things. *Journal of Electronics, Electromedical Engineering, and Medical Informatics*, 5(2), 69–74. <https://doi.org/10.35882/jeemi.v5i2.282>
- Isyanto, H., Syahrul Wahid, A., & Ibrahim, W. (2022). *Desain Alat Monitoring Real Time Suhu Tubuh, Detak Jantung dan Tekanan Darah secara Jarak Jauh melalui Smartphone berbasis Internet of Things Smart Healthcare*. 5(1).
- K. K. RI. (2019, May 17). *Hipertensi Penyakit Paling Banyak Diidap Masyarakat*. <https://sehatnegeriku.kemkes.go.id/baca/umum/20190517/5130282/hipertensi-penyakit-paling-banyak-diidap-masyarakat/>

- Nur Fariz, M., & jamaaluddin. (2021). Rancang Bangun Monitoring Tekanan Darah, Gula Darah Non-Invasif, Berat, Dan Suhu Badan Berbasis Internet Of Things. In *Procedia of Engineering and Life Science* (Vol. 1, Issue 1).
- Prayoga, D. (2020). *Desain Dan Implementasi Sistem Monitoring Detak Jantung Dan Suhu Tubuh Dengan Nodemcu Dan Thingspeak Berbasis Iot*.
- Ratna, S., Kalimantan, I., Al, M. A., & Banjarmasin, B. (2020). *SISTEM MONITORING KESEHATAN BERBASIS INTERNET OF THINGS (IoT)*.
- Rihhadatulaisy, Z. H., & Irianto, K. D. (2024). Designing an Automatic Room Temperature Control System for Smart Homes for the Elderly Using IoT. *International Journal Software Engineering and Computer Science (IJSECS)*, 4(2), 758–766. <https://doi.org/10.35870/ijsecs.v4i2.2844>
- Stikes, R. (2019). *FAKTOR-FAKTOR PENYEBAB TERJADINYA HIPERTENSI PADA LANSIA DI UPT PSTW KHUSNUL KHOTIMAH. XIII(5)*.
- Sulista, A., Nehru, N., & Fuady, S. (2021). Rancang Bangun Alat Monitoring Tekanan Darah Berbasis Internet of Things (IoT). *Jurnal Engineering*, 3(1). <https://doi.org/10.22437/jurnalengineering.v3i1.11666>
- Syani, M., Santika Nurathilla, C., Ahmad Firdaus, E., Andrijana Kusuma, K., & TEDC Bandung, P. (2023). *Penerapan Mini Internet Of Things (IoT) Board Berbasis Mikrokontroler Untuk Monitoring Kesehatan Lansia*. 17, 2614–5405. <https://doi.org/10.25134/nuansa>
- Widyaiswara Hairil, T., & Islamiyati, A. (2015). *Penaksiran Parameter Model Kalibrasi Linier yang Berdistribusi Skew-Normal dengan Algoritma-EM* (Vol. 12, Issue 1).