# Hypertension Detector Based on Internet of Things

**Oktavia Dewi Suryaatmaja[1], Koesmarijanto Koesmarijanto*[2], Dianthy Marya[3]**

[1,3] Digital Telecommunication Network Study Program, Department of Electrical Engineering, State Polytechnic of Malang, 65141, Indonesia.

[2] Telecommunication Engineering Study Program, Department of Electrical Engineering, State Polytechnic of Malang, 65141, Indonesia.

[1]oktaviadewisuryaatmaja@gmail.com, [2]koesmarijanto@polinema.ac.id,[3]dianthy@polinema.ac.id

***Abstract*— Hypertension, a leading global non-communicable disease and a major risk factor for cardiovascular mortality, requires continuous monitoring due to its often asymptomatic nature. This research aims to design and implement an Internet of Things (IoT)-based system for detecting and managing hypertension. The system comprises a prototype detector device for measuring blood pressure and a complementary mobile application for data communication. The development follows the Research and Development (R&D) methodology. The application facilitates remote patient monitoring by connecting doctors and patients, featuring data logging, chat, call, and video call functionalities for timely medical intervention. The prototype blood pressure detector was validated through ten tests on different subjects, comparing its systolic and diastolic readings against a standard sphygmomanometer. Results indicate an average error of 16.9% for systolic pressure and 17.2% for diastolic pressure, corresponding to accuracy rates of 83% and 82.8%, respectively. The observed variations in error are attributed to differing patient physiological conditions during measurement. The study concludes that the developed IoT-based system offers a viable solution for remote hypertension monitoring, enabling proactive healthcare management by bridging patients and doctors through integrated hardware and software.**

***Keywords*— Hypertension, Internet of Things, Blood Pressure Measurement, Remote Patient Monitoring, e-Health.**

## I. INTRODUCTION

Hypertension, defined as a chronic elevation of systemic arterial blood pressure, is a predominant global health crisis and a leading modifiable risk factor for cardiovascular disease (CVD), stroke, and premature mortality [1]. Its pathological persistence places excessive hemodynamic stress on vital organs, leading to complications such as heart failure, renal impairment, and vascular dementia, thereby substantially diminishing patient quality of life and imposing significant economic burdens on healthcare systems [2]. Often termed the "silent killer," hypertension typically progresses asymptomatically, resulting in underdiagnosis until a catastrophic event like myocardial infarction or hemorrhagic stroke occurs [3]. Contemporary epidemiological models project that the global prevalence of hypertension will exceed 1.5 billion individuals by 2025, highlighting an urgent need for innovative, scalable, and proactive management strategies [4].

Conventional diagnostic and monitoring protocols, which rely on sporadic measurements in clinical settings are fraught with limitations. These include the inability to detect nocturnal hypertension or masked hypertension, the presence of white-coat effect, and a lack of granular data on diurnal blood pressure variability—a strong prognostic indicator [5]. Consequently, there is a paradigm shift towards Ambulatory Blood Pressure Monitoring (ABPM) and Home Blood Pressure Monitoring (HBPM), which are endorsed by major clinical guidelines for their superior predictive accuracy [6]. The integration of these concepts with the Internet of Things (IoT) framework presents a transformative opportunity. IoT in healthcare, or the Internet of Medical Things (IoMT), enables the creation of interconnected ecosystems where medical sensors, data gateways, cloud platforms, and end-user applications communicate seamlessly to facilitate remote patient monitoring (RPM) and telemedicine [7], [8].

The core of an IoT-based hypertension detector is a digitally-enabled sphygmomanometer. Modern implementations often utilize the oscillometric method, where an air pressure sensor, such as the MPX5700AP, accurately measures the pressure oscillations within an inflatable cuff to derive systolic and diastolic values algorithmically [9]. A microcontroller unit (MCU) with embedded wireless capability, such as the ESP8266-based Wemos D1 R1, processes this sensor data, manages cuff inflation via a motor driver (e.g., L298N module) controlling a DC pump, and handles network communication [10], [11]. The acquired blood pressure readings are then transmitted via Wi-Fi to a cloud backend service like Google Firebase. Firebase provides a real-time database and authentication services, ensuring secure, synchronized, and instantly accessible data storage [12].

On the consumer-facing end, a dedicated mobile application, developed using platforms like Android Studio, serves as a digital health dashboard. It visualizes historical trends, provides medication reminders, and educates the user [13]. Crucially, a complete IoMT system extends beyond simple data logging to enable clinical decision support. By incorporating a physician portal, the system allows healthcare providers to remotely access patient data, identify trends indicative of poor control, and intervene proactively through integrated teleconsultation features such as secure messaging or video calls [14], [15]. This closed-loop system—from automated measurement to clinical feedback—represents the next evolution in chronic disease management.

Despite these technological advancements, many existing solutions are fragmented, offering either a standalone device or

*Corresponding author

a simple app without true clinical integration. This study aims to design, implement, and evaluate a holistic, IoT-based hypertension detection and management system. The proposed system integrates a custom-built, automated blood pressure monitor with a dual-interface platform (patient app and clinician web portal), connected via a Firebase cloud infrastructure. The primary objective is to create a validated prototype that demonstrates improved accessibility to regular monitoring, enhances patient engagement through data visualization, and facilitates timely medical intervention, ultimately contributing to better hypertension control and reduced complication rates.

### *A. Android Studio*

Android studio is the official IDE (Integrated Development Environment) for developing Android applications and is open source or free. The launch of Android Studio was announced by Google on May 16 2013 at the Google I/O Conference event for 2013. Since then, Android Studio has replaced Eclipse as the official IDE for developing Android applications.

Android studio is an integrated development environment - Integrated Development Environment (IDE) for android application development, based on IntelliJ IDEA. Apart from being an IntelliJ code editor and powerful developer tools, Android Studio offers more features to increase productivity when creating Android applications, for example:

- Project based on Gradle Build.
- Fast re-factory and bug fixing.
- A new tool called "Lint" is claimed to be able to quickly monitor application speed, usability and compatibility.
- Supports Pro-guard and App-signing for security.
- Having an Android application GUI is easier.
- Supported by Google Cloud Platform for every application developed [16].

### *B. Firebase*

Firebase has a main product, namely providing a real-time database with a backend as a service. This serviceprovides an application development API that allows application data to be synchronized across clients and stored in Firebase's cloud. Firebase provides libraries for various client platforms that allow integration with Android, iOs, JavaScript, Java, Objective-C, and Node Js applications and can also be called a DbaaS (Database as a Service) service with a real-time concept. Firebase is used to make it easier to add features to be built by developers [17].

### *C. Wemos D1 R1*

Wemos D1 R1 is a board that uses ESP8266 as a Wi-Fi module and is designed to resemble an Arduino Uno. The advantages of the Wemos D1 R1 are that it is open source, compatible with Arduino, can be programmed using Arduino IDE, pinout compatible with Arduino Uno, can stand alone without using another microcontroller, has a 32- bit processor with a speed of 80 MHz, High Level Language, can be programmed with Python and Lua programming languages [18].


Figure 1. Wemos D1 R1[18]

The specifications of the Wemos D1 R1 microcontroller are shown in Table I.

TABLE I
WEMOS D1 R1 SPECIFICATIONS

| Category | Specifications |
|---|---|
| Microcontroller | ESP8266 Tensilica 32-bit |
| Serial to USB Converter | CH340G |
| Operating Voltage | 3.3 – 5V |
| Input Voltage | 7 – 12V |
| Digital I/O Pins | 11 |
| PWM I/O Pins | 10 |
| Analog Input Pins | 1 (10-bit) |
| DC Current per I/O Pin | 12mA (Max) |
| Hardware Serial Ports | 1 |
| Flash Memory | 4 Mbytes |
| Instruction RAM | 64 Kbytes |
| Data RAM | 96 Kbytes |
| Clock Speed | 80 MHz |
| Network | IEEE 802.11 b/g/n WiFi |
| Built-in LED | Attached to digital pin 13 |
| USB Connector Style | Micro-B Female |
| Board Dimensions (PCB) | 69 x 53mm (2.7 x 2.1”) |
| Datasheet | ESP8266EX |

### *D. Sensor MPX5700AP*

Figure 2. MPX5700AP [19]

The MPX5700AP sensor is a single port absolute silicon pressure sensor with integrated pressure in a 6 pin SIP package which is a Manifold Absolute Pressure (MAP), which is a pressure sensor that can read air pressure in a manifold. The MPX5700AP sensor is equipped with a signal conditioning circuit and temperature calibrator. Bipolar processing inside the transistor provides an accurate and high-level analog output signal that is proportional to the applied pressure, so the MPX5700AP sensor has 2.5% maximum error, pressure 15 kPa to 700 kPa, voltage rangebetween 4.75V – 5.25V, sensitivity 1.0 kPa or equivalent to 0.145 psi, and a temperature range of -40oC – 125oC. Thissensor is used to determine the air pressure in the cuff on the tensimeter [19].

*E. Motor DC*

A direct current motor or device that converts electrical energy into mechanical energy. DC motors are one of the two basic types of motors, the other being AC motors. A DC motor consists of several parts, including the stator, armature, rotor, and commutator with brushes. The opposite polarity direction between the two magnetic fields in the motor causes the motor to rotate. DC motors are the simplest type of motor and are commonly used in household appliances such as electric razors and electric windows in cars. DC motors are equipped with permanent magnets or electromagnetic coils that produce a magnetic field.

The working principle of a DC motor lies in thecurrent flowing through an armature, or coil placed betweenthe north and south poles of a magnet. The magnetic field produced by the coil will interact with other magnetic fieldsand produce torque on the motor. In a DC motor, the magnet is formed on the stator, while the armature is placed on the rotor and the commutator diverts current from one coil to another. The commutator functions to connect the stationary power source to the armature through the use of brushes or conductive rods. DC motors operate at a fixed speed for a fixed voltage and there is no slip [20].

*F. Modul L298N*

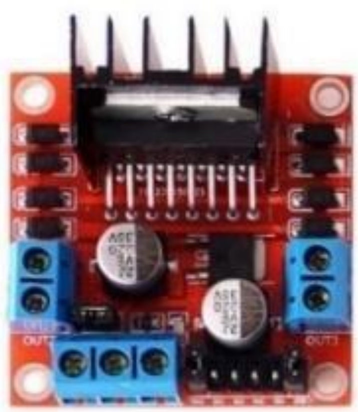

Figure 3. Modul L298N

L298N is a DC motor driver module that is most widely used or used in the world of electronics which is used to control the speed and direction of rotation of a DC motor. IC L298 is an H-bridge type IC which is capable ofcontrolling inductive loads such as relays, solenoids, DC motors and stepper motors. The L298 IC consists of transistor logic (TTL) with a Nand gate which functions tomake it easier to determine the direction of rotation of a DC motor or stepper motor. Table II is a specification regardingthe L298N driver module.

TABLE II
L298N DRIVER MODULE SPECIFICATIONS

| **Driver power supply** | **+5V ~ +46V** |
|---|---|
| Driver Io | 2A |
| Logic power output | +5V ~ +7V (*internal supply* +5V) |
| Logic current | 0 – 36 mA |
| Controlling level | *Low*: -0.3V – 1.5V<br>*High*: 2.3V - Vss |
| Enable signal level | *Low*: -0.3V – 1.5V<br>*High*: 2.3V – Vss |
| Max power | 25W (*Temperature* 75ºC) |
| Working temperature | -25C ~ +130C |
| Dimension | 60mm*54mm |
| Driver weight | 48g |

## II. METHOD

*A. Research Stages*

This research requires several research stages starting from the design process to producing prototypes and applications that patients will use during the treatment period. The systematic stages of this research are presented in the form of a flowchart as shown in Fig 4.

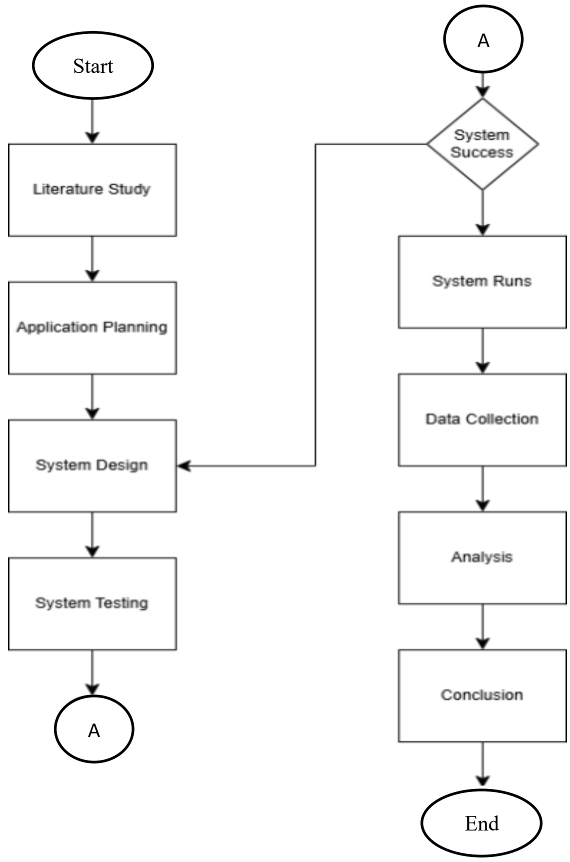


Figure 4. Research stages

The following is a description of Fig. 4:

- The first stage, conduct literature studies in reference journals that have been published by various sources and observe problems in the field and determine the object and parameters of the problem.
- The second stage is carrying outapplication planning. This planning includes designing theapplication system that will be created.
- The third stage, namely designing the system. System design takes the form of designing the application and

tensimeter tool that will be made.

- The fourth stage, namely conducting system testing, whether the applications and tools created are in accordance with the design or not. If the system is successful, then applications and tools can run. If not, it will return to the designprocess.
- The fifth stage, data collection is carried out. The data taken is based on testing on applications and tools.
- Thesixth stage, carry out analysis of the data that has been obtainedwhen testing the application.
- The seventh stage, makeconclusions based on the analysis data obtained.

*B. Block Diagram*

In this research there are two block diagrams, including system diagrams for tools and applications. The system diagram for the tool will be shown in Fig. 5.

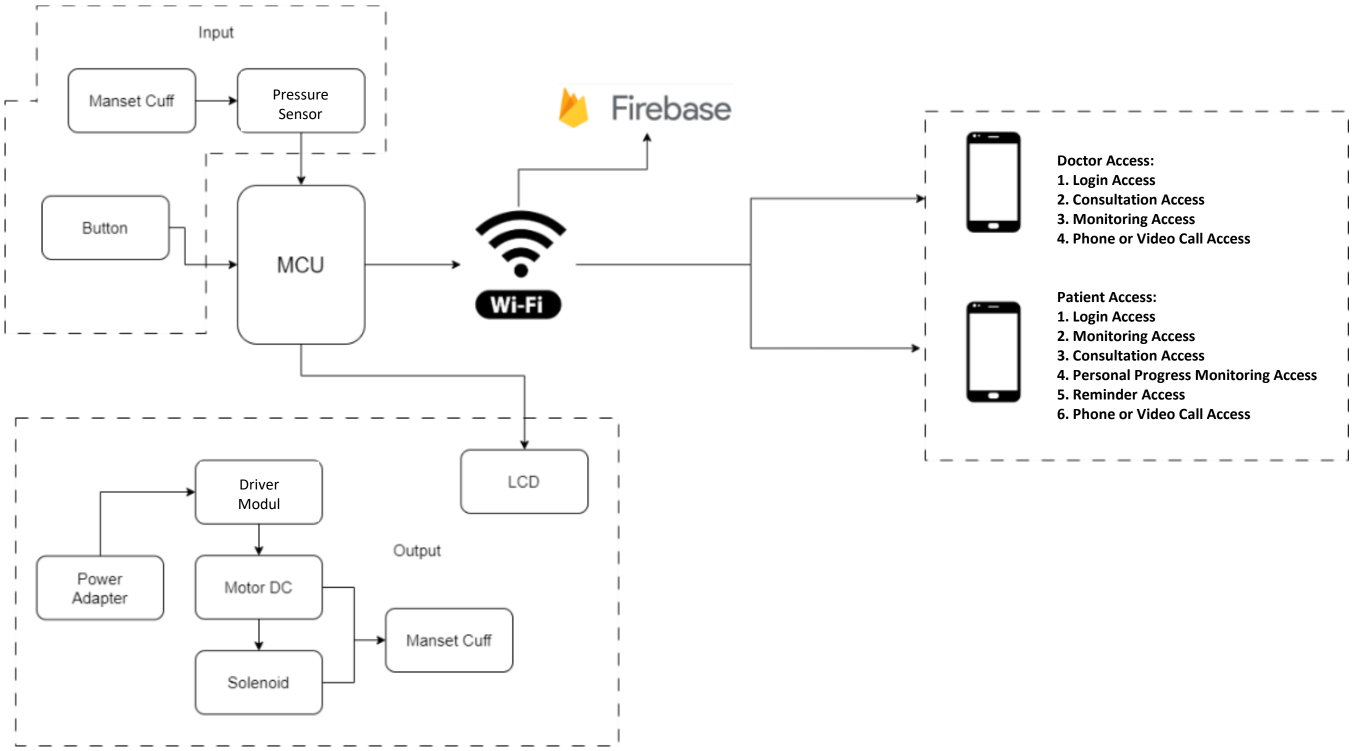


Figure 5. Tools Diagram

Based on the block diagram in Fig. 5, each system block has the following functions.

- Wemos D1 R1 will be used as a microcontroller to process data.
- The sensor as input is used to read the value of the pressure applied from within the cuff or arm cuff.
- A DC motor and solenoid will be used to pump the cuff that will be used. After that, the measurement results will be displayed on the LCD.
- Power adapter is used to supply dc motors.
- There is Wi-Fi as a connection for the application to be designed.
- Then, firebase as support for creating applications. Firebase is used to connect monitoring data from each access, both from monitoring access, doctors and patients. During patient access, a form will be available containing personal development data. Data on patient access is much simpler. The aim is so that patients can understand every question asked in the form. The data sent by the patient will be read automatically by the doctor, so that the doctor can provide treatment for the patient's condition. The patient's existing monitoring access will later display data from blood pressure measurements carried out via the Hypertension Detector device.
- Apart from that, in patient and doctor access there is also a consultation feature, where patients can directly connect with doctors via chat, call or video call features.

*C. User System Flowchart*

There are 3 flowcharts in this research, including tensimetersystem, doctor and patient flowcharts.

*1) Tensimeter System Flowchart:*

The tensimeter system flowchart that will be created has the following algorithm.

- The system starts from start.
- Connect the power adapter to the relay module to start the dc motor.
- Press the button to start the calculation.
- If yes, then the dc motor will start pumping the cuff. If not, then the dc motor cannot pump successfully.

- After the dc motor starts pumping, the systole and diastole values will automatically be displayed on the LCD screen.

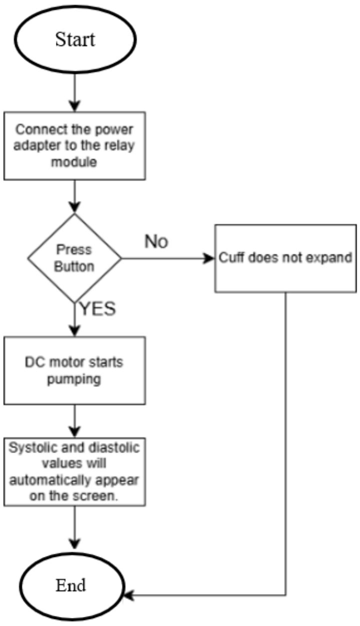


Figure 6. Tensimeter flowchart

*2) Doctor Access Flowchart:*

There are 3 flowcharts in this research, including tensimeter system, doctor and patient flowcharts.

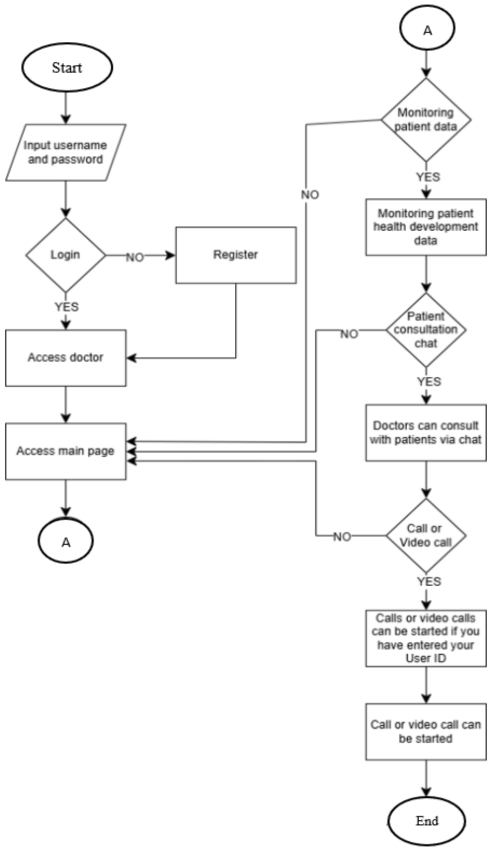


Figure 7. Doctor access

Fig. 7 explains the flow diagram of the doctor access. After the application starts, the user log in. If you don't have anaccount, you can register first. If you already have an account,you can log in. Then, you can choose doctor access. On this page, there are buttons for monitoring patient data, patient consultation chat, and call or video call. If the patient data monitoring button is clicked, it will display the patient's progress condition sent by the patient himself. But if not, it will return to the main access page. Next, if the patient consultationchat button is clicked, it will display a chat column for doctorsto consult with patients via chat. Apart from that, consultationswith patients can also be carried out via the call or video call feature.

*3) Patient Access Flowchart:*

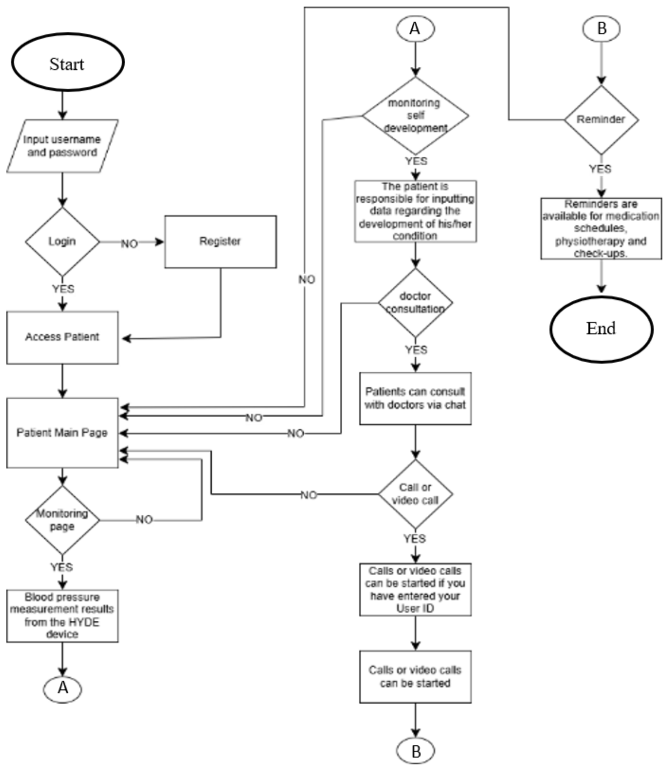


Figure 8. Patient access

Fig. 8 explains the flow diagram of the patient access. After the application starts, the user log in. If you don't have an account, you can register first. If you already have an account, you can log in. Then, you can choose patient access. On this page there are monitoring buttons, monitoring personal progress, doctor consultation chat, call or video call and reminders. If the monitoring button is clicked, this menu will be connected to the hypertension detector prototype, where the measurement results data will be displayed automatically on the systole and diastole values. Later, the status of the application will automatically change according to the value entered. The results of the status reading are in the form of normal, low or high blood pressure measurements. If the self-monitoring progress button is clicked, a page will be displayed for the patient to input the progress of their condition in real time to the doctor. If not, it will return to the main access page.

If the doctor consultation chat button is clicked, it will display a chat column for patients to consult with a doctor, either via chat, call or video call. If not, it will return to the main access page. If the reminder button is clicked, it will display a reminder page for the patient's schedule for taking medication, doing physiotherapy and also checkups. If not, it will return to the main access page.

## III. RESULTS AND DISCUSSION

### A. Application HYDE

[redacted]


Figure 9. Splash screen

When you first open the application, a splash screen will appear that says HYDE (Hypertension Detector) or the name of the application being created. After that, the display will automatically switch to the sign-in page.

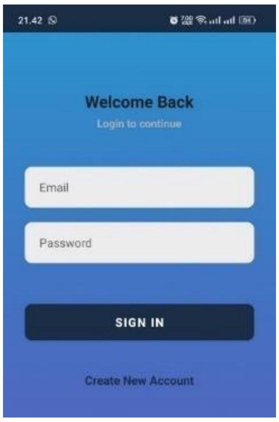


Figure 10. Sign in

On the first page a sign in menu will appear, where we need to enter the registered Email and Password. If you don't have an account, you can create an account first in the Create New Account section. After that, we can log in with this account.


Figure 11. Sign up page

Fig.11 shows the sign-up page, where we need to enter image, name, email, password and confirm password to create an account.


Figure 12. Access page

On this page there are 2 accesses, namely doctors and patients. Each page has a different menu. The menu on the page will be explained in Fig. 13 and Fig. 14.


Figure 13. Doctor page

There are 3 buttons on the doctor's page, namely the Patient Data Monitoring button, the patient consultation chat button, and the call or video call button. Monitoring patient data onthe doctor's page will display monitoring data sent by patients. This data will automatically be entered on the doctor's page, so that the doctor can monitor this data.

Next, there is a patient consultation chat and call or video call menu. In this menu, doctors can consult with patients via the chat, call or video call features available. This menu will make it easier for patients when they need something urgent to consult a doctor.


Figure 14. Patient page

Lastly, there is the patient page. This page has more access than doctors. On the patient page there are 5 menus, including the Monitoring menu which will contain data on the results of checking measurements from the digital tensimeter. Next, the Monitoring Personal Progress menu which contains a form regarding the patient's development data in one week. Apart from that, there is a Patient Consultation Chat and Call or Video Call menu, where in this menu patients can consult a doctor via the available chat, call or video call feature. The goal is to make it easier for patients to ask about emergency conditions, so that patients can receive treatment more quickly. And the fourth menu is the Reminder menu.

## B. *Tensimeter Tools*

The tensimeter circuit is shown in Fig. 15. The tensimeteris made using Wemos D1 R1 as a microcontroller, and a power adapter as output power to run a dc motor.

The MPX5700AP sensor is connected to the Wemos D1 R1 microcontroller via pin A0. Then, it is uploaded using program code supported by the library of the microcontroller itself. The process of uploading the program code to the Wemos D1 R1 uses a microUSB or Universal Serial Bus cable that connects the laptop to the Arduino IDE as software. The solenoid and dc motor are connected to VCC and COM in the relay module. The module used here is the L298N module, where in this module there are 2 voltage inputs. The 12V voltage is connected to VCC which comes from the power adapter, while the 5V voltage is connected to VCC on the Wemos D1 R1. Because Wemos D1 R1 only requires a voltage of 5V, the voltage from the driver module is also 5V. Measurements are taken by attaching the cuff to the left arm. After that, press the button to start the calculation. Before using the cuff, make sure there is no air in the cuff, so that measurements can be taken optimally. Then, press the button again according to its original position. The measurement results will automatically be displayed on the LCD. Testing will be carried out 10 times.

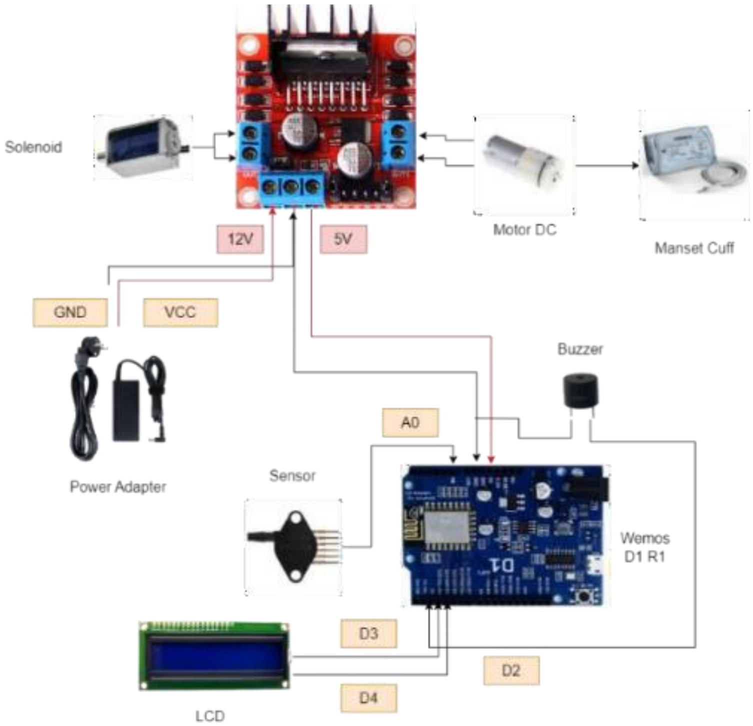


Figure 15. Tensimeter circuit

TABLE III
READING IF PRESURE SENSOR VALUES

| mmHg | Dataadc | Tensimeter |
|---|---|---|
| 0 | 297 | 0 |
| 20 | 303 | 19 |
| 40 | 307 | 44 |
| 60 | 313 | 61 |
| 80 | 321 | 80 |
| 100 | 326 | 99 |
| 120 | 328 | 124 |
| 140 | 338 | 142 |
| 160 | 344 | 159 |

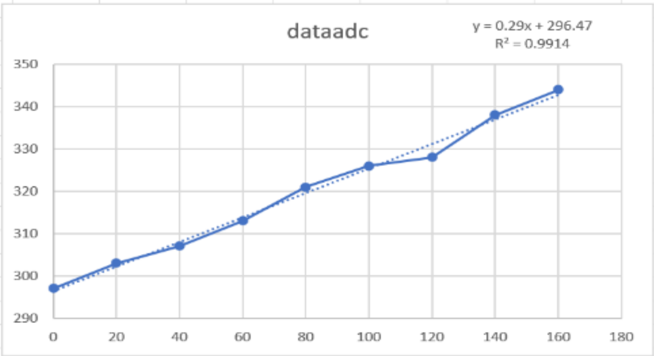


Figure 16. Calibration Chart

Before using the tensimeter, it is necessary to calibrate using the cuff and pump used by the manual tensimeter. The ADC number from the sensor will later be recorded in Excel and produce a formula that will be used as a benchmark for measuring systole and diastole values. The results of the sensor calibration are shown in Table 1. Once the data has been obtained, create a graph in Excel as shown in Figure 16. Then enter the Y value into the coding as in Fig. 17.

```
firebase_write.ino
void loop(){
  dataadc = analogRead(A0);      //pembacaan sensor pada pin A0
  mmhgx = (dataadc - 296.47) / 0.29;     //pembacaan sensor - nilai Y pada excel
  Serial.println(dataadc);
```

Figure 17. Arduino coding

Systole and diastole values will automatically appear on the LCD screen, as shown in Fig. 18.


Figure 18. Measurement results

## C. Accuracy Test Results

Accuracy was carried out 10 times with the results of testing the systole and diastole values on the prototype being compared with the systole and diastole values on a conventional digital sphygmomanometer.

TABLE IV
ACCURACY TEST RESULTS

| Prototype Testing Result | | Tensimeter (Arm Style) | | Difference (mmHg) | |
|---|---|---|---|---|---|
| Systole | Diastole | Systole | Diastole | Systole | Diastole |
| | | 191 | 65 | 93 | 7 |
| | | 101 | 74 | 5 | 15 |
| | | 105 | 84 | 9 | 26 |
| 96 | 58 | 102 | 61 | 6 | 3 |
| | | 101 | 63 | 5 | 5 |
| | | 219 | 98 | 90 | 28 |
| | | 139 | 98 | 10 | 23 |
| | | 108 | 78 | 21 | 3 |
| | | 136 | 63 | 7 | 12 |
| 129 | 75 | 129 | 76 | 0 | 1 |
| | | 167 | 75 | 38 | 23 |
| | | 146 | 88 | 17 | 10 |
| | | 101 | 77 | 28 | 21 |
| | | 128 | 66 | 1 | 32 |
| 129 | 98 | 125 | 98 | 4 | 0 |
| | | 112 | 94 | 3 | 23 |
| | | 112 | 74 | 3 | 3 |
| | | 108 | 67 | 7 | 4 |
| 115 | 71 | 101 | 82 | 14 | 10 |
| | | 115 | 76 | 0 | 5 |
| | | 136 | 98 | 1 | 16 |
| | | 108 | 77 | 29 | 5 |
| | | 136 | 61 | 1 | 21 |
| 137 | 82 | 129 | 70 | 8 | 12 |
| | | 139 | 84 | 2 | 2 |
| | | 101 | 88 | 7 | 14 |
| | | 108 | 81 | 0 | 7 |
| | | 105 | 94 | 3 | 20 |
| | | 105 | 74 | 3 | 0 |
| 108 | 74 | 108 | 74 | 0 | 0 |
| | | 167 | 94 | 63 | 25 |
| | | 105 | 98 | 1 | 29 |
| | | 105 | 61 | 1 | 8 |
| | | 115 | 84 | 11 | 15 |
| 104 | 69 | 108 | 71 | 4 | 2 |
| | | 112 | 84 | 1 | 28 |
| | | 110 | 71 | 3 | 15 |
| | | 106 | 84 | 7 | 28 |
| | | 121 | 61 | 8 | 5 |
| 113 | 56 | 115 | 60 | 2 | 4 |
| | | 198 | 57 | 92 | 4 |
| | | 125 | 79 | 19 | 18 |
| | | 105 | 84 | 1 | 23 |
| | | 112 | 84 | 6 | 23 |
| 106 | 61 | 101 | 64 | 5 | 3 |
| | | 277 | 77 | 169 | 3 |
| | | 101 | 94 | 7 | 20 |
| | | 246 | 66 | 138 | 8 |
| 108 | 74 | 125 | 84 | 17 | 10 |
| | | 108 | 76 | 0 | 2 |
| Average Error Percentage | | | | 16.9% | 17.2% |

Based on the results of the accuracy table above, it can be seen that the accuracy is different for each systole and diastole value. Then the research tries to carry out calculations using a formula to get the difference value:

$$Difference = Tensimeter - Prototype\ testing\ result \quad (1)$$

$$Average = \frac{Total\ difference}{Total\ testing}$$

$$Average\ eror\ percentage = \frac{Average\ difference}{Average\ tensimeter\ value} \times 100\% \quad (2)$$

$$Systole = \frac{19.4}{114.5} \times 100\% = 16.9\%$$

$$Dystole = \frac{12.4}{71.8} \times 100\% = 17.2\%$$

$$Sensor\ Acuracy\ Percentage = 100\% - average\ error\ percentage \quad (3)$$

$$Systole = 100\% - 16.9\% = 83\%$$

$$Dystole = 100\% - 17.2\% = 82.8\%$$

A large difference value was obtained in each experiment. This is due to the influence of the type of sensor used during calibration. The sensor used has a sensitivity value of 6.4 kPa. Meanwhile, the sensitivity of the sensor greatly influences the measurement results. Sensor sensitivity is the sensor's ability to detect changes in output compared to the unit change in input. Sensitivity will show how sensitive the sensor is to the quantity being measured. The greater the sensitivity value of a sensor, the more sensitive the sensor is to its input, so the better the quality of the sensor.

## IV. CONCLUSION

Based on the research regarding the Hypertension Detector Based on Internet of Things, it can be concluded that the application is successfully supported by a monitoring form available for patients connected via Firebase. The data sent by the tool automatically appears on the patient page, displaying systole and diastole values along with measurement status categorized as low, normal, or high. The tensimeter tool, developed using a pressure sensor, underwent testing 10 times with 5 data samples each. The results indicate an average difference of 19.2 mmHg for systolic pressure and 12.4 mmHg for diastolic pressure when compared to a standard arm-style sphygmomanometer. The discrepancies in experimental data were influenced by varying patient conditions. The average error percentage in detecting values was 16.9% for systolic pressure and 17.2% for diastolic pressure. These significant differences observed in experiments are attributed to the low sensitivity of the sensor used (6.4 kPa), as sensor sensitivity plays a critical role in measurement accuracy.

## REFERENCES


[1] B. Zhou et al., "Worldwide trends in hypertension prevalence and progress in treatment and control from 1990 to 2019: a pooled analysis of 1201 population-representative studies with 104 million participants," *The Lancet*, vol. 398, no. 10304, pp. 957–980, Sep. 2021.

[2] P. K. Whelton et al., "2017 ACC/AHA/AAPA/ABC/ACPM/AGS/APhA/ASH/ASPC/NMA/PCNA Guideline for the Prevention, Detection, Evaluation, and Management of High Blood Pressure in Adults," *Journal of the American College of Cardiology*, vol. 71, no. 19, pp. e127–e248, May 2018.

[3] G. S. Stergiou et al., "2021 European Society of Hypertension practice guidelines for office and out-of-office blood pressure measurement," *Journal of Hypertension*, vol. 39, no. 7, pp. 1293–1302, Jul. 2021.

[4] World Health Organization, "Global Report on Hypertension: The Race Against a Silent Killer," WHO, Geneva, Switzerland, 2023.

[5] J. A. Staessen, Y. Li, and T. Richart, "Ambulatory Blood Pressure Monitoring in Clinical Practice," *European Heart Journal*, vol. 42, no. 34, pp. 3293–3295, Sep. 2021.

[6] T. G. Pickering et al., "Recommendations for Blood Pressure Measurement in Humans and Experimental Animals: Part 1: Blood Pressure Measurement in Humans," *Hypertension*, vol. 45, no. 1, pp. 142–161, Jan. 2005.

[7] S. R. Islam, D. Kwak, M. H. Kabir, M. Hossain, and K.-S. Kwak, "The Internet of Things for Health Care: A Comprehensive Survey," *IEEE Access*, vol. 3, pp. 678–708, Jun. 2015.

[8] J. Qi, P. Yang, G. Min, O. Amft, F. Dong, and L. Xu, "Advanced Internet of Things for Personalised Healthcare Systems: A Survey," *Pervasive and Mobile Computing*, vol. 41, pp. 132–149, Oct. 2017.

[9] W. Aritonang, I. A. Bangsa, and R. Rahmadewi, "Implementation of the DS18B20 Temperature Sensor and the MPX5700AP Pressure Sensor Using an Arduino Microcontroller on a Stress Level Detector," *Jurnal Ilmiah Wahana Pendidikan*, vol. 7, no. 1, pp. 56–63, Feb. 2021.

[10] F. A. Deswar and R. Pradana, "Server Room Temperature Monitoring Using Wemos D1 R1 Based on Internet of Things (IoT)," *Jurnal Ilmiah : Technologia*, vol. 12, no. 1, pp. 25–32, Jan. 2021.

[11] R. Ashshiddiq and B. Rahmadya, "Design of an Automatic Internet of Things-Based Blood Pressure Measurement Device," *Journal of Computer Hardware, Signal Processing, Embedded Systems and Networks*, vol. 4, no. 1, pp. 12–18, Apr. 2023.

[12] P. E. A. Kaunang, S. R. U. A. Sompie, and A. S. M. Lumenta, "Implementation of Google Internet of Things Core for Monitoring Car Tire Air Volume," *Jurnal Teknik Elektro dan Komputer*, vol. 9, no. 3, pp. 163–170, Dec. 2020.

[13] "Android Developers," Google. [Online]. Available: https://developer.android.com/studio. [Accessed: Apr. 1, 2023].

[14] A. I. Alaboudi et al., "Perceptions and Experiences of Remote Patient Monitoring in Healthcare: A Systematic Review and Thematic Synthesis," *Journal of Medical Internet Research*, vol. 25, e45129, 2023.

[15] M. A. Alshehri, S. M. Alghamdi, and A. S. Alghamdi, "Mobile Health Applications for Hypertension Management: A Systematic Review of Functionality and Quality," *JMIR mHealth and uHealth*, vol. 10, no. 7, e35817, Jul. 2022.

[16] P. E. A. Kaunang, S. R. U. A. Sompie, and A. S. M. Lumenta, "Implementasi Google Internet of Things Core pada monitoring volume ban angin mobil," *J. Tek. Elektro dan Komputer*, vol. 9, no. 3, pp. 163–170, Dec. 2020.

[17] F. A. Deswar and R. Pradana, "Monitoring suhu pada ruang server menggunakan Wemos D1 R1 berbasis Internet of Things (IoT)," *J. Ilmiah : Technologia*, vol. 12, no. 1, pp. 25–32, Jan. 2021.

[18] W. Aritonang, I. A. Bangsa, and R. Rahmadewi, "Implementation of the DS18B20 temperature sensor and the MPX5700AP pressure sensor using an Arduino microcontroller on a stress level detector," *J. Ilmiah Wahana Pendidikan*, vol. 7, no. 1, Feb. 2021.

[19] R. Ashshiddiq and B. Rahmadya, "Rancang alat pengukur tekanan darah otomatis berbasis Internet of Things," *J. Comput. Hardw., Signal Process., Embed. Syst. and Netw.*, vol. 4, no. 1, Apr. 2023.

[20] A. Siswanto et al., "Meja tulis adjustable dengan konsep smart furniture," *J. Ilmiah Widya Teknik*, vol. 19, no. 2, 2020.