Jurnal ELTIKOM :
Jurnal Teknik Elektro, Teknologi Informasi dan Komputer

Vol. xx, No. yy, month year, page. ab-cd
ISSN 2598-3245 (Print), ISSN 2598-3288 (Online)
DOI
Available online at http://eltikom.poliban.ac.id

IMPLEMENTATION OF INTERNET OF THINGS IN AUTOMATIC PILL DISPENSER TO RAISE ADHERANCE

Andrew Kurniawan Gianto, Sibgah Rabbani Kusuma, Theresia Herlina Rochadiani
[1,2,3)] Departement of Informatics, Pradita University, Tangerang, Indonesia

e-mail: andrew.kurniawan@student.pradita.ac.id, sibgah.rabbani@student.pradita.ac.id, theresia.herlina@pradita.ac.id

Received: xx month year – Revised: xx month year – Accepted: xx month year

ABSTRACT

In the daily lives of society, people may have hectic and busy schedules. Sometimes, this busy lifestyle can make them forget to maintain their health, including taking medication to their prescribed schedule. One of many solutions to deal with this problem is to create an automatic medication dispenser that will assist people in preparing efficient medicine containers and reminding them when it is time to take their medicine. This dispenser makes use of modern innovations, such as a smart reminder system, mobile app notifications, and automated dosage setting. Integrating hardware and software design, as well as functionality testing on elderly people samples, are among the development methodologies used. The research findings suggest that this medicine dispenser is helpful in enhancing, especially elderly, people adherence to medication regimens and lowering the probability of dose mistakes. This study concludes that automatic medicine dispenser technology can be an effective and practical option to promote the health of the elderly, enhance their quality of life, and lessen the strain on caregivers.

Keywords: *Automatic pill dispenser, Reminder, Elderly, Advance technology, Health*

I. Introduction

In the daily lives of society, people can have hectic and busy schedules. This busy lifestyle often causes people to forget to maintain their health, including taking medication according to the prescribed schedule[1]. Good health is crucial in preventing various illnesses[2], yet the reality is that many people often forget or are inconsistent in taking their medication[3], especially for those with specific medical conditions.

A study on the compliance level of patients at the Lempake Samarinda Community Health Center taking medications found that medication adherence varied by age (45.5%, 34.1%, and 24.4%)[4]. Patients over 45 years old (39.2%), women (34.2%), with elementary school education (14.6%), and a treatment duration of less than 5 years (34.1%) had high levels of adherence.

Non-adherence to medication is a serious health issue and can lead to fatal consequences[5]. One of the main causes of non-adherence is patients forgetting to take their medication or taking it at the wrong time[1]. This can be due to various factors such as busyness, forgetfulness, or not having an organized place to store medication.

Regular medication consumption is critical to sustaining health, particularly for people who have chronic medical problems that need continuous medication management[6], [7]. Unfortunately, many people continue to struggle with reminders and taking medication on time. Many of us forget or take medicine outside of the authorized timetable, which might impair the effectiveness of treatment.

One solution that can be done to remind people to take their medication on a regular and prescribed schedule is a Smart Automatic Pill Dispenser that is linked to the Internet of Things (IoT) [8], [12], [13], [14], [15], [16].

The Smart Automatic Pill Dispenser serves to remind users of their medication routine. When the time arrives, the automated dispenser will alert the user with notifications from their devices. Furthermore, this dispenser may display information about the amount of medication remaining, helping consumers to ensure they have an appropriate supply.

The primary objective for this Automatic Pill Dispenser is to remind people to take their medications regularly without the possibility of missing pills. It also helps them to avoid accidental over/under dose. In other words, to improve medication adherence and the quality of life for people who need constant medication management.

II. RESEARCH METHODS

A. Research Framework

Research Framework is a tool that provides an underlying structure or model to support collective research efforts. It's a precise representation of the structure of a research project plan. In essence, a research framework guides the direction of the study, suggests relationships between key concepts, and discusses relevant theories based on your literature review

.

The Research Framework for this project includes 4 steps illustrated in figure 1.

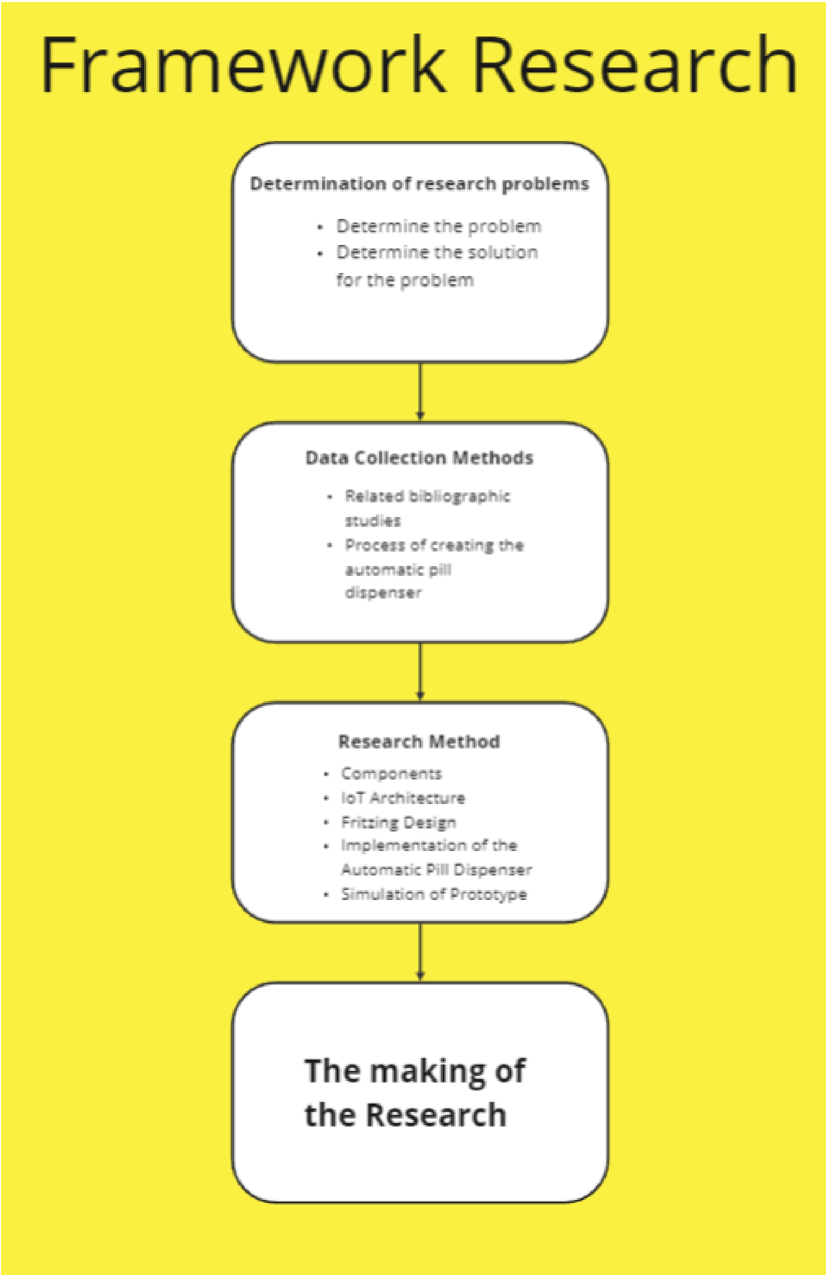

Figure 1.Research Framework

B. Data

The research data comes from related bibliographic studies that have previously been done to gain in-depth insights into the observation that is under investigation such as the comparison of pill dispensers with different components, average medical adherence in Indonesia, and important features that need to be implemented in an automatic pill dispenser. This study also involves the process of creating an automatic pill dispenser based on existing scientific knowledge which starts by choosing the right components, making the IoT Architecture, and creating the fritzing design model for the project.

C. Components

To achieve its objectives, this research utilizes a set of materials and tools that have been specifically described. Each component has their own purpose in this project.

The microcontroller that is used for this project is the ESP-32 dev kit V1 as shown in Figure 1. This serves to connect sensors, actuators, and other components together.

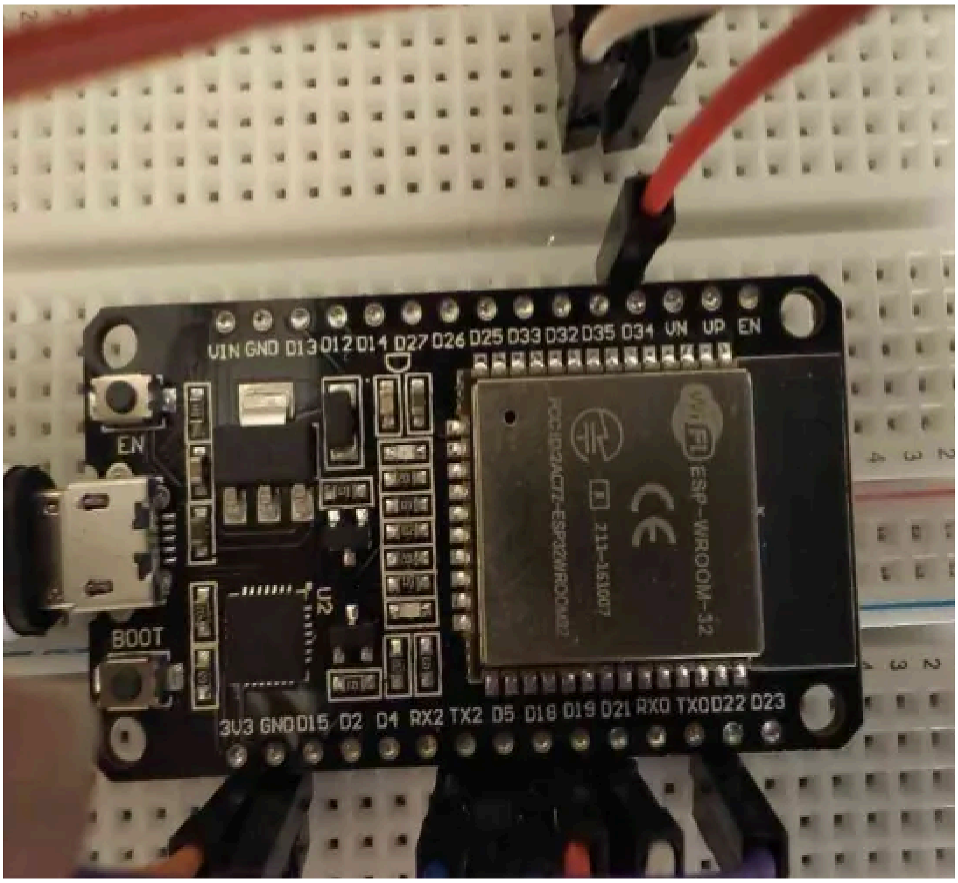

Figure 2. ESP-32 Dev Kit V1

Our project also requires sensors for data collection as well. The sensor that is used is the ultrasonic sensor as shown in figure 3.

The ultrasonic sensor operates by emitting sound waves at a frequency higher than the human ear can detect. The sensor functions to prepare medication from the dispenser by detecting the presence and proximity of an object, such as a hand, through these sound waves.

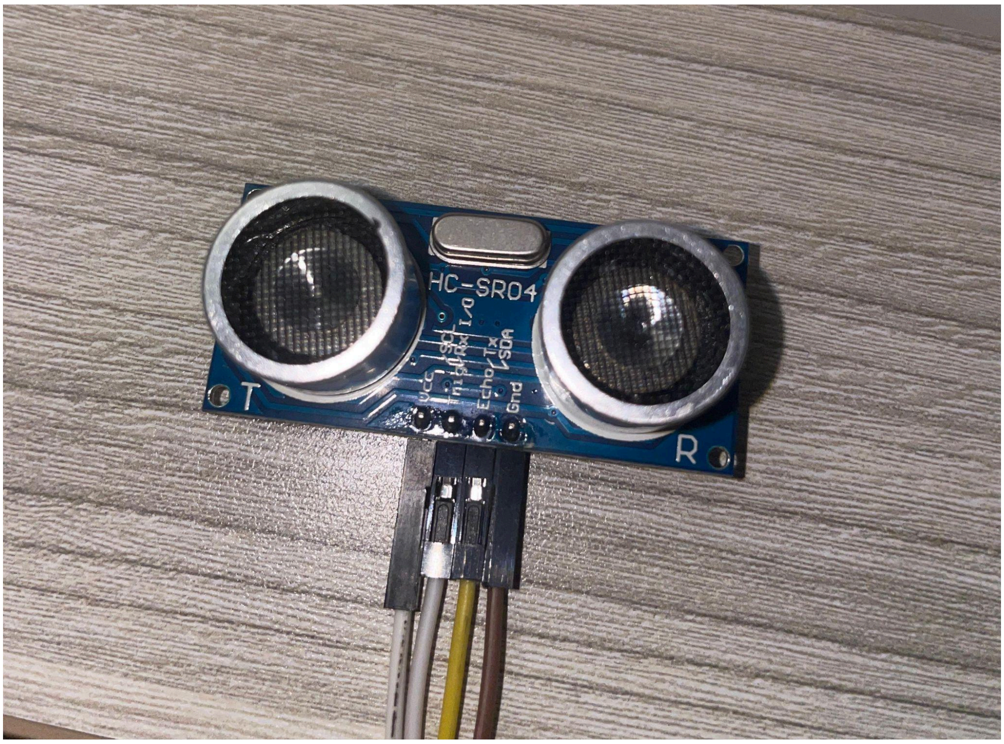

Figure 3. Ultrasonic Sensor

An actuator is also needed to create movement based from a signal for the automatic pill dispenser to operate. The actuator that is used is a motor stepper 28BYJ and drive board ULN2003 as shown in figure 4. It functions to move mechanical parts within the dispenser, transferring medication pills from the storage to a location accessible for the user to retrieve.

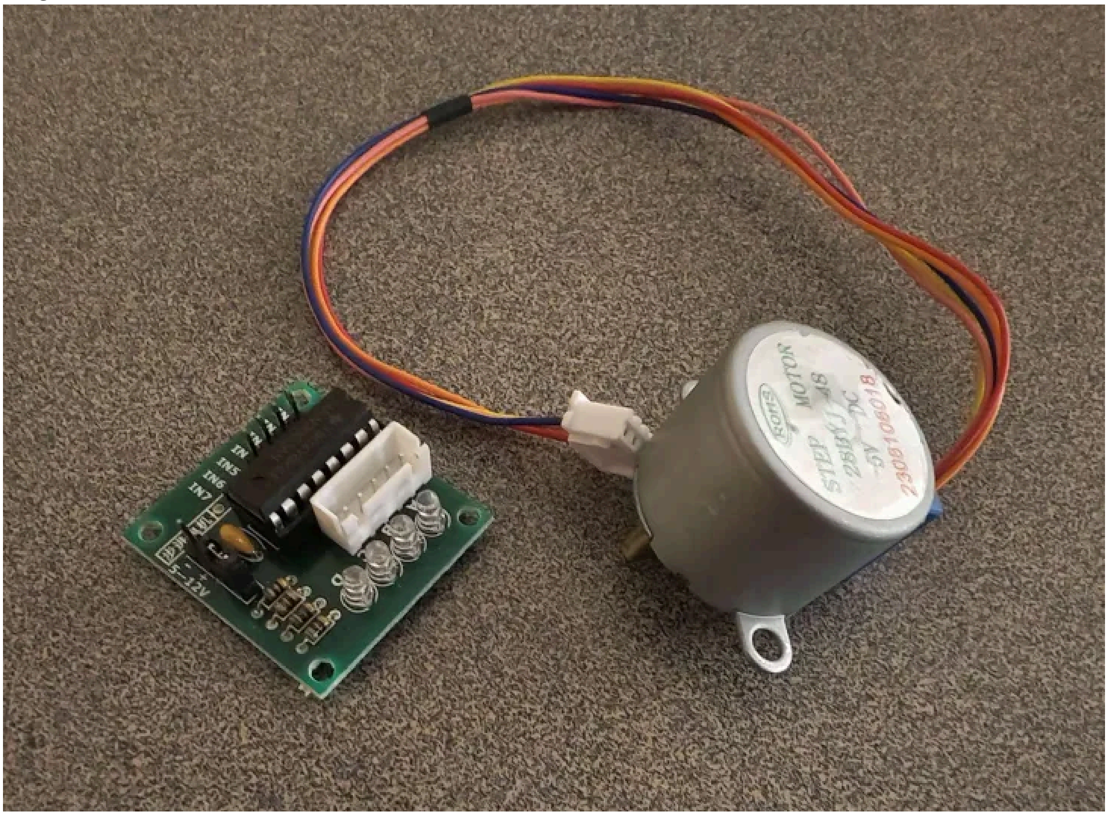

Figure 4. 28BYJ and ULN2003

The real-time clock (RTC) module, as shown in figure 5, can be utilized in this project to notify or alert when the pill is ready within a specified time.

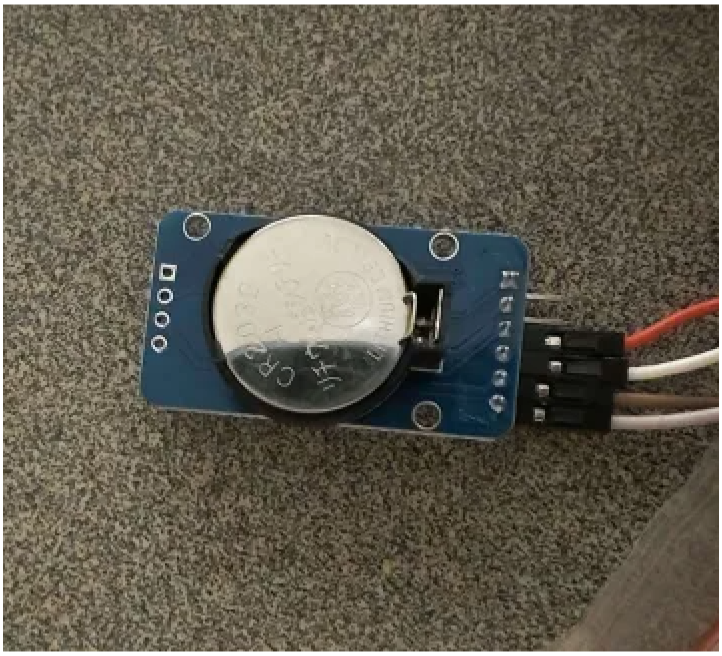

Figure 5. Real-time Clock Module

The LCD 1602 I2C, as shown in figure 6, is used to display the time, amount of activations, and information of when the pill is ready to be taken.

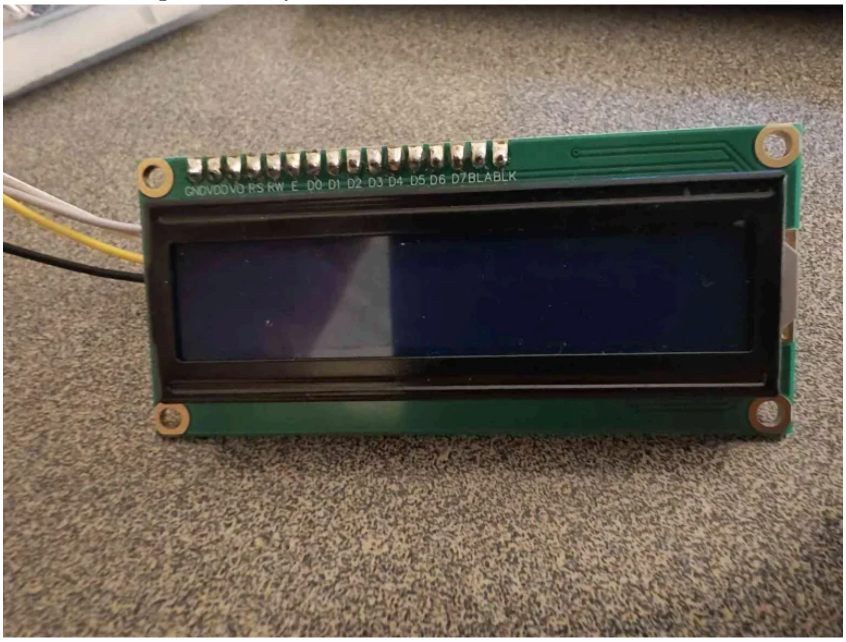

Figure 6. LCD 1602 I2C

Other components that are also used for this project include the Breadboard.

D. IoT Architecture

The IoT architecture includes devices (sensors, actuators), connectivity (network layer), data processing (middleware), applications (software), security, and management layers. These components work together to enable data collection, processing, and control in IoT systems.

IoT architectures have various models that have specific use cases and requirements, each with its own configuration of layers and components[9].

In a three-layer IoT architecture, sensors and actuators gather data, which is then sent through the network layer to a central server or cloud platform for analysis[10], [11]. Subsequently, applications utilize this processed data for various purposes, such as monitoring, control, or decision-making. A four-layer architecture introduces a middleware layer between the network and application layers. This middleware layer manages tasks like data processing, routing, and integration, facilitating seamless communication between devices and applications. Lastly, a five-layer architecture, security and privacy considerations are integrated as a distinct layer. This dedicated layer ensures the safeguarding of data, devices, and networks against unauthorized access, tampering, or cyber threats, forming a secure foundation for IoT deployments.

Based on the requirements and the components that the project uses, the IoT architecture that is needed to be used is the 3-layers architecture, which consists of the perception layer, network layer, and application layer as illustrated in figure 7. The first layer, Perception Layer, includes sensors and actuators such as Ultrasonic sensor, Drive Motor 28BYJ and Driver ULN2003. The second layer, which is the network layer, which is the connecting of devices and transferring data. The third layer, Application layer, involves connection of cloud and servers with Blynk, an application server platform that is used for controlling the Automatic Pill Dispenser remotely.

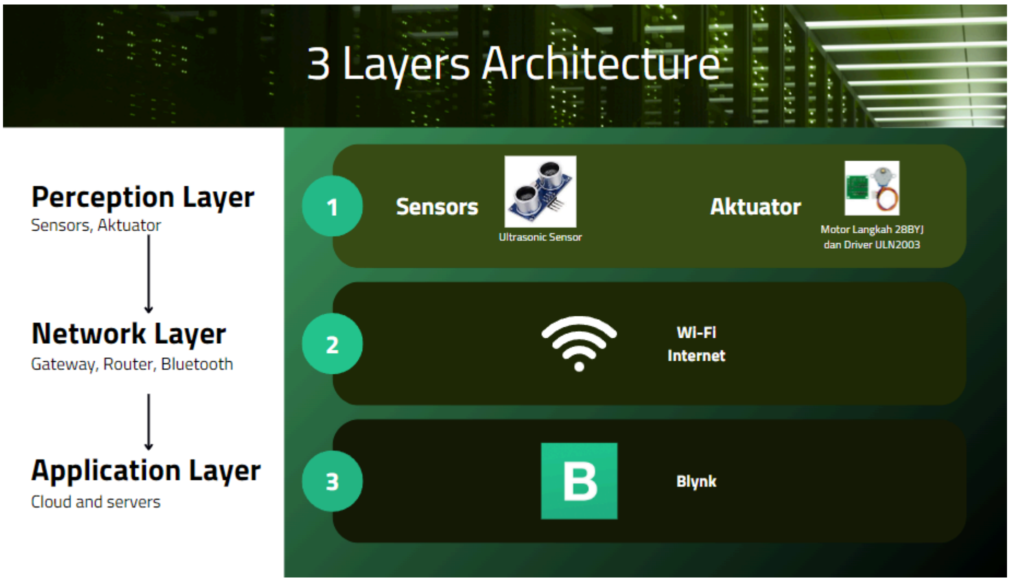

Figure 7. 3 Layers IoT Architecture

E. Automatic Pill Dispenser Design

To understand the Automatic Pill Dispenser design, we use a fritzing application for the design as illustrated in figure 8. At first, link the RTC's SCL pin to the ESP32's SCL pin, the RTC's SDA pin to the ESP32's SDA pin, the RTC's VCC pin to the ESP32's 3V pin, and the RTC's GND pin to the ESP32's GND pin. The LCD's SCL pin should be connected to the ESP32's SCL pin, its SDA pin to the ESP32's SDA pin, its VCC pin to the ESP32's 3V pin, and its GND pin to the ESP32's GND pin as the next step. Next, attach the motor driver 28BYJ48's IN1 pin to the ESP32's D5 pin, its IN2 pin to the ESP32's D18 pin, its IN3 pin to the ESP32's D19 pin, and its IN4 pin to the ESP32's TX2 pin. Furthermore, link the motor driver 28BYJ48's POWER pin to the USB to TTL's 5V pin and its GND pin to the same USB to TTL pin. The ultrasonic sensor's VCC pin should be connected to the 3V pin on the ESP32, its Trigger pin should be connected to the D4 pin, its Echo pin should be connected to the D2 pin, and its GND pin should be connected to the GND pin on the ESP32.

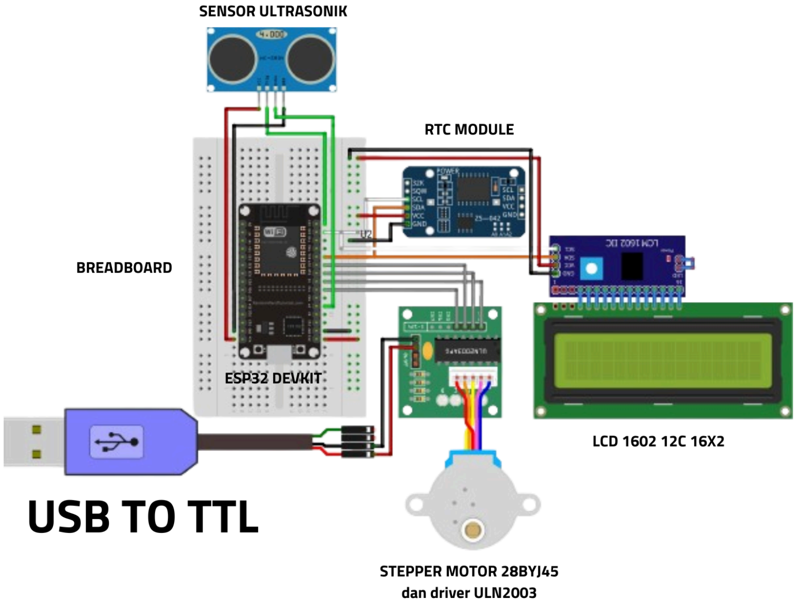

Figure 8. Design in Fritzing Application

III. Result and Discussions

A. 3D Designing

Using the Blender application, a 3D model of the automatic pill dispenser was managed to be built with the wanted requirements as shown in Figure 10. In order to create the model, an existing design from the Instructables website, illustrated in figure 9, was modified and tailored to the particular requirements of the project. Although the initial design provided a strong framework, it was improved in a number of ways to increase its usefulness and functionality. The approach started with a thorough examination of the original design and a description of the intended changes. The 3D modeling program Blender was then used to put these modifications into practice. This was very helpful in assessing the design's ergonomics and making the required modifications.

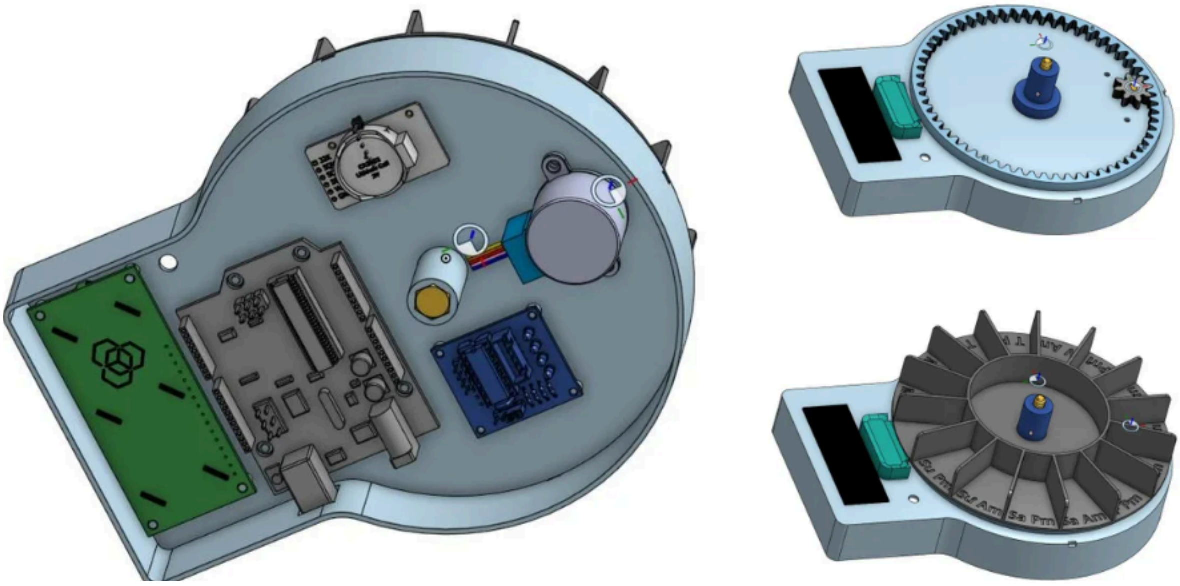

Figure 9. Original 3D Design [17]

Figure 10. 3D Designing in Blender

B. *Device Program Code*

This C++ code uses an ESP32 microcontroller to power an Internet of Things pill dispenser. It makes use of the Stepper library to operate a stepper motor, the RTClib for real-time clock capabilities, the LiquidCrystal_I2C library to operate an LCD display, and the Blynk library for Internet of Things connectivity. The code generates instances for the stepper motor, RTC, LCD, and Blynk timer in addition to configuring constants, WiFi credentials, stepper motor connections, and ultrasonic sensor pins. The setup() function sets up the stepper motor, RTC, LCD, and ultrasonic sensor in addition to initializing serial communication and establishing a Blynk connection. In order to track closeness and refresh the LCD, two functions—checkDistance() and displayTime()—are programmed to execute once per second.

```cpp
void setup()
{
  // Start the serial communication
  Serial.begin(115200);
  // Initialize the Blynk connection
  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
  // Initialize the stepper motor
  stepper.setSpeed(5); // Set the speed to 5 RPM
  // Initialize the RTC
  if (!rtc.begin()) {
    Serial.println("Couldn't find RTC");
    while (1);
  }
  if (rtc.lostPower()) {
    // Set the RTC to the current date & time if it lost power
    rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
  }

  // Initialize the LCD
  lcd.init();
  lcd.backlight();
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Pill dispenser");
  lcd.setCursor(0, 1);
  lcd.print("Sibgah, Andrew");
  delay(2000);
  // Initialize the ultrasonic sensor
  pinMode(TRIGGER_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
  // Setup a function to be called every second
  timer.setInterval(1000L, checkDistance);
  timer.setInterval(1000L, displayTime);
}

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

```cpp
void displayTime() {
  DateTime now = rtc.now();

  // Display the current time on the LCD
  lcd.clear();
  lcd.setCursor(0, 0);
  if (canMoveStepper()) {
    lcd.print("Obat sudah siap");
    Blynk.logEvent("obat", "Obat sudah siap nih!, Ambil obatnya yuk");
  } else {
    lcd.print("Time: ");
    lcd.setCursor(0, 1);
    lcd.print(now.hour());
    lcd.print(':');
    if (now.minute() < 10) {
      lcd.print('0');
    }
    lcd.print(now.minute());
    lcd.print(':');
    if (now.second() < 10) {
      lcd.print('0');
    }
    lcd.print(now.second());
  }
}

void checkDistance() {
  // Trigger the sensor by setting the trigger pin high for 10 microseconds
  digitalWrite(TRIGGER_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIGGER_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIGGER_PIN, LOW);

  // Read the echo pin and calculate the distance
  long duration = pulseIn(ECHO_PIN, HIGH);
  float distance = duration * 0.034 / 2;

  // Check if the distance is below 15 cm and can move the stepper
  if (distance < 7 && canMoveStepper()) {
    moveStepper();
  }
}
```

Figure 11. Set up, loop, and function Code

The timer and Blynk processes are continuously executed via the loop() method. A number of BLYNK_WRITE() functions manage communication with the Blynk app, enabling remote control of the stepper motor, move limit setting, ESP32 reset, move count resetting, and time window adjustments for pill distribution.

```
BLYNK_WRITE(V0) // Button on V0 for moving the stepper
{
  int pinValue = param.asInt();
  if (pinValue == 1) { // If button is pressed
    if (canMoveStepper()) {
      moveStepper();
    } else {
      lcd.clear();
      lcd.setCursor(0, 0);
      lcd.print("Belum waktunya");
      lcd.setCursor(0, 1);
      lcd.print("minum obat");
      delay(1000);
    }
  }
}

BLYNK_WRITE(V1) // Button on V1 for resetting the move count
{
  int pinValue = param.asInt();
  if (pinValue == 1) { // If button is pressed
    resetMoveCount();
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Move Count");
    lcd.setCursor(0, 1);
    lcd.print("Reset");
    delay(2000); // Display reset message for 2 seconds
    displayTime();
  }
}
```

```
BLYNK_WRITE(V6) // Slider on V6 for setting morning start time
{
  morningStart = param.asInt(); // Set morningStart to slider value
}

BLYNK_WRITE(V7) // Slider on V7 for setting morning end time
{
  morningEnd = param.asInt(); // Set morningEnd to slider value
}

BLYNK_WRITE(V8) // Slider on V8 for setting evening start time
{
  eveningStart = param.asInt(); // Set eveningStart to slider value
}

BLYNK_WRITE(V9) // Slider on V9 for setting evening end time
{
  eveningEnd = param.asInt(); // Set eveningEnd to slider value
}

BLYNK_WRITE(V4) // Slider on V4 for setting the move limit
{
  moveLimit = param.asInt(); // Set moveLimit to slider value
}

BLYNK_WRITE(V5) // Button on V5 for resetting the ESP32
{
  int pinValue = param.asInt();
  if (pinValue == 1) { // If button is pressed
    ESP.restart(); // Restart the ESP32
  }
}
```

Figure 12. Declaring virtual pins for Blynk

The canMoveStepper() function determines whether the move count is below a predetermined limit and whether the current time is inside predefined intervals. MoveCount and Last Move date are updated, the motor is operated, and success notifications are displayed by the moveStepper() function. The move count is reset and the last move date is updated via the resetMoveCount() function. The checkDistance() function uses the ultrasonic sensor to detect proximity and, if necessary, drive the stepper motor. The displayTime() function shows the current time on the LCD.

```
bool canMoveStepper() {
  DateTime now = rtc.now();

  // Reset move count at midnight
  if (now.day() != lastMoveDate.day()) {
    moveCount = 0;
  }

  // Check if current time is within allowed intervals and move count is less than moveLimit
  bool morningWindow = now.hour() >= morningStart && now.hour() < morningEnd;
  bool eveningWindow = now.hour() >= eveningStart && now.hour() < eveningEnd;

  if ((morningWindow || eveningWindow) && moveCount < moveLimit) {
    return true;
  }
  return false;
}

void moveStepper() {
  // Display "Processing" message on LCD
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Processing");

  // Move the stepper motor
  int steps = -1024;
  stepper.step(steps); // Move 1024 steps (full revolution for 28BYJ-48)

  // Increment move count and update last move date
  moveCount++;
  lastMoveDate = rtc.now();

  // Display success message and then return to the main display
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Sukses!");
  delay(2000);
  displayTime();
}

void resetMoveCount() {
  moveCount = 0;
  lastMoveDate = rtc.now(); // Update last move date to now
}
```

```
void displayTime() {
  DateTime now = rtc.now();

  // Display the current time on the LCD
  lcd.clear();
  lcd.setCursor(0, 0);
  if (canMoveStepper()) {
    lcd.print("Obat sudah siap");
    Blynk.logEvent("obat", "Obat sudah siap nih!, Ambil obatnya yuk");
  } else {
    lcd.print("Time: ");
    lcd.setCursor(0, 1);
    lcd.print(now.hour());
    lcd.print(':');
    if (now.minute() < 10) {
      lcd.print('0');
    }
    lcd.print(now.minute());
    lcd.print(':');
    if (now.second() < 10) {
      lcd.print('0');
    }
    lcd.print(now.second());
  }
}

void checkDistance() {
  // Trigger the sensor by setting the trigger pin high for 10 microseconds
  digitalWrite(TRIGGER_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIGGER_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIGGER_PIN, LOW);

  // Read the echo pin and calculate the distance
  long duration = pulseIn(ECHO_PIN, HIGH);
  float distance = duration * 0.034 / 2;

  // Check if the distance is below 15 cm and can move the stepper
  if (distance < 7 && canMoveStepper()) {
    moveStepper();
  }
}
```

Figure 13. Declaring functions for stepper

In order to develop an automated pill dispenser, this code combines motor control, ultrasonic sensing, remote interface, and real-time clock capabilities. It offers ease and promotes medication adherence by guaranteeing that medication is given at the appropriate times.

C. Blynk Application

Using the Blynk application, a user-friendly interface was made for controlling and monitoring the automatic pill dispenser. The application consists of three main pages: the Home Page, the Set Up Page, and the Reset Page.

The Home Page, illustrated in figure 14, features a button that is connected to virtual pin 0 and triggers the dispensing mechanism of the device. This provides a simple and intuitive way for users to dispense their medication at the touch of a button.

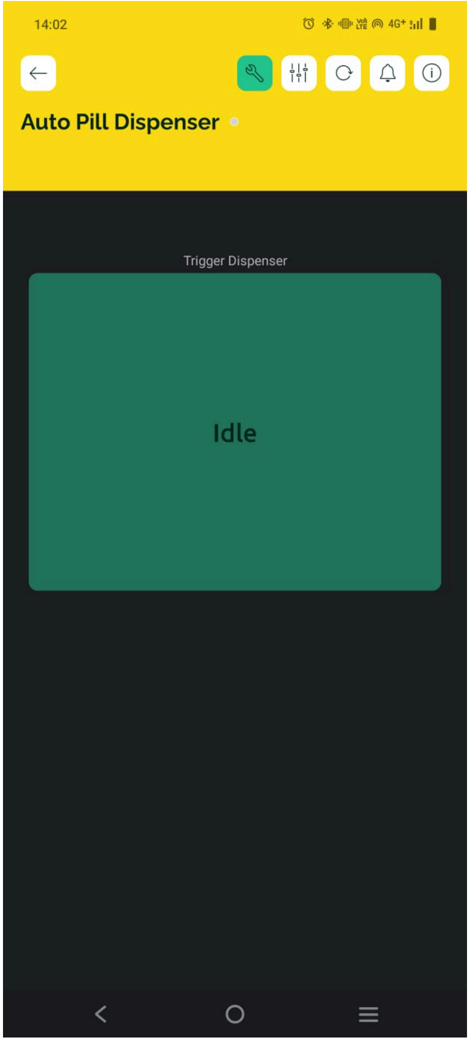

Figure 14. Home Page

The Set Up Page, as shown in figure 15, is designed to allow users to customize the operation of the pill dispenser according to their needs. It includes a step slider that is connected to virtual pin 4 and can be used to increase or decrease the pill counter per day, allowing users to adjust the dosage as required. Additionally, it has another set of step sliders connected to virtual pins 6, 7, 8, 9 and provides the option to set up time limits for day and night, ensuring that medication is dispensed at appropriate times.

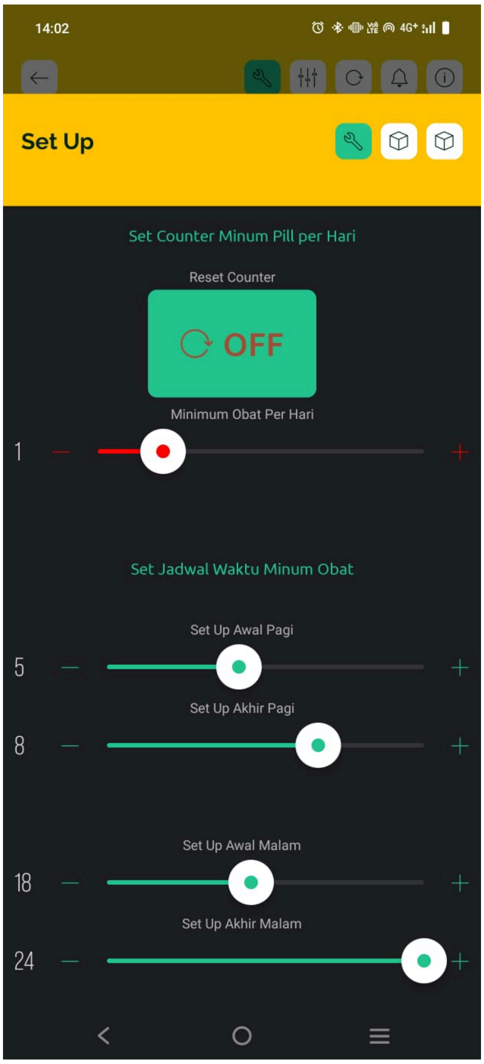

Figure 15. Set Up Page

The Reset Page, as shown in figure 16, includes a button connected to virtual pin 5 and resets the ESP, effectively restarting the program. This feature is particularly useful for debugging purposes, allowing for quick and easy troubleshooting when necessary.

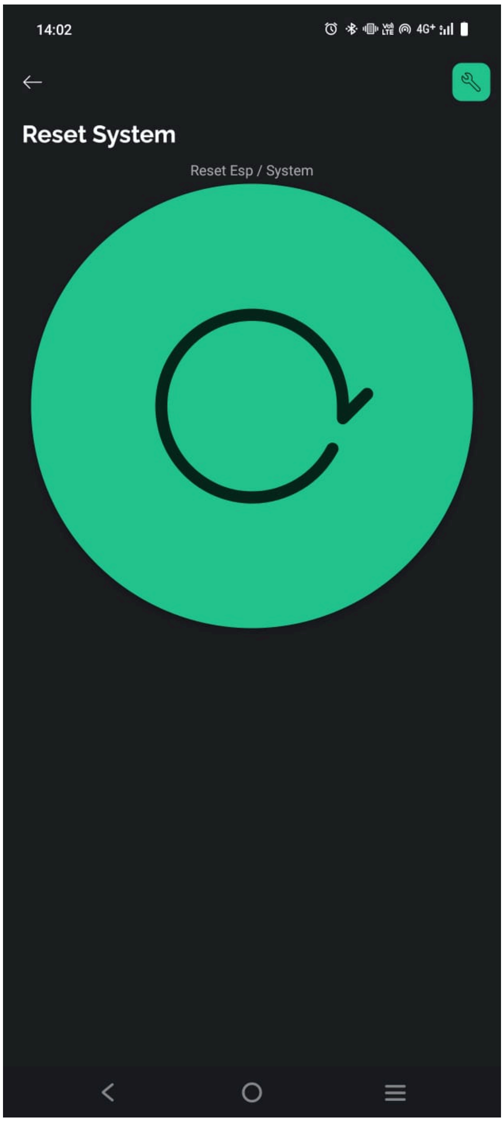

Figure 16. Reset Page

This Blynk application plays a crucial role in enhancing the functionality and user experience of the automatic pill dispenser, providing an intuitive interface for customization, control, and maintenance of the device.

D. Real-world applications

In real-world applications, the automatic pill dispenser operates in a series of states, illustrated in figure 17, that ensure efficient and accurate dispensing of medication. Initially, in the Idle State, the dispenser is idle, waiting for the next pill schedule and displaying the current time. When it's time for medication, the dispenser enters the Ready State, displaying a notification that the pill is ready to be taken and standing by for any input, either from a virtual button on the Blynk application or detection of an object within 7 cm range via the ultrasonic sensor. This flexibility in triggering mechanisms caters to the varying abilities and preferences of users. Upon receiving input, the dispenser activates and enters the Processing State, displaying a processing message while the stepper motor is moving to dispense the pill, providing real-time feedback to the user about the dispensing process. Once the pill has been dispensed, the dispenser enters the Completion State, completing the processing and not receiving any more input of the virtual button or ultrasonic sensor until the next pill schedule, preventing accidental dispensing of extra medication.

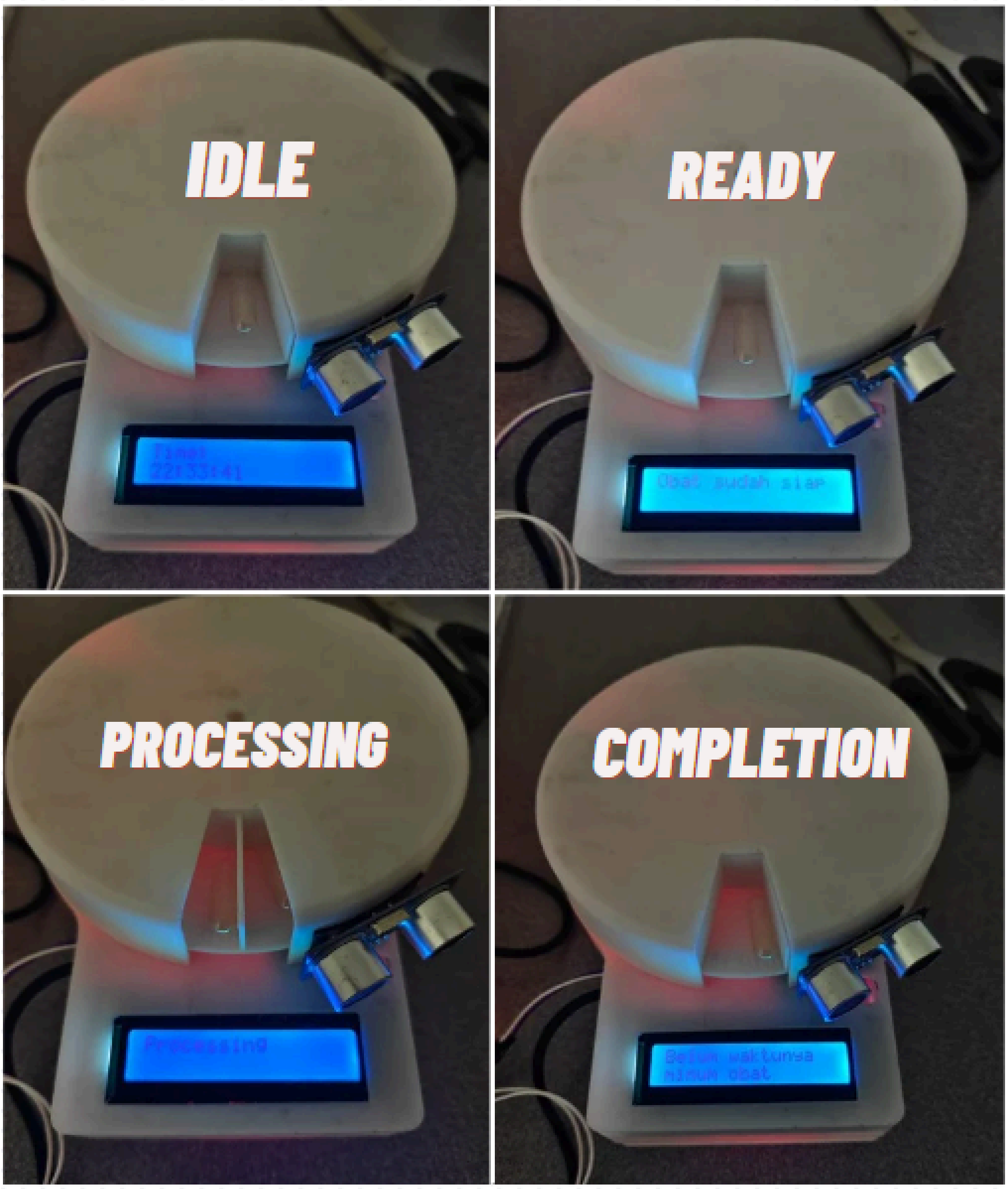

Figure 17. Four states of pill dispenser

IV. CONCLUSION

The automated pill dispenser was successfully built and deployed using the ESP32 and the Blynk program, as illustrated in this article. Several important innovations enhanced the dispenser's operation and appearance. These include an interactive virtual button, a pill counter for managing dose, a time setting tool for scheduling medication times, and a reset ESP button for system difficulties. One critical component of this system is the notification that consumers receive via the Blynk app when their scheduled medicine is ready to be administered. This function, in addition to enhancing the user experience, ensures that medications are taken on time. The automated pill dispenser, a significant advance in home healthcare technology, demonstrates how Internet of Things (IoT) devices may improve patient health and prescription adherence.

References

[1] A. S. Gadkari and C. A. McHorney, "Unintentional non-adherence to chronic prescription medications: How unintentional is it really?," *BMC Health Serv Res*, vol. 12, no. 1, p. 98, 2012, doi: 10.1186/1472-6963-12-98.

[2] T. van der Voordt and P. A. Jensen, "The impact of healthy workplaces on employee satisfaction, productivity and costs," *Journal of Corporate Real Estate*, vol. 25, no. 1, pp. 29–49, Jan. 2023, doi: 10.1108/JCRE-03-2021-0012.

[3] J. Pitanupong and J. Sammathit, "Knowledge and attitudes on medication adherence and residual symptoms in individuals with depression: a survey at a University Hospital," *BMC Psychiatry*, vol. 23, no. 1, p. 210, 2023, doi: 10.1186/s12888-023-04706-y.

[4] N. Hijriyati *et al.*, "TINGKAT KEPATUHAN MINUM OBAT PASIEN HIPERTENSI DI PUSKESMAS LEMPAKE SAMARINDA," 2022.

[5] A. Kassem, W. Antoun, M. Hamad, and C. El-Moucary, "A comprehensive approach for a smart medication dispenser," in *International Journal of Computing and Digital Systems*, University of Bahrain, Mar. 2019, pp. 131–141. doi: 10.12785/ijcds/080205.

[6] J. Philip, F. Mary Abraham, K. Kurian Giboy, B. J. Feslina, and T. Rajan, "Automatic Medicine Dispenser using IoT," 2020. doi: 10.17577/IJERTV9IS080152.

[7] A. Gast and T. Mathes, "Medication adherence influencing factors—an (updated) overview of systematic reviews," *Syst Rev*, vol. 8, no. 1, p. 112, 2019, doi: 10.1186/s13643-019-1014-8.

[8] S. Nazir, Y. Ali, N. Ullah, and I. García-Magariño, "Internet of Things for Healthcare Using Effects of Mobile Computing: A Systematic Literature Review," *Wireless Communications and Mobile Computing*, vol. 2019. Hindawi Limited, 2019. doi: 10.1155/2019/5931315.

[9] J. Maktoubian and K. Ansari, "An IoT architecture for preventive maintenance of medical devices in healthcare organizations," *Health Technol (Berl)*, vol. 9, no. 3, pp. 233–243, 2019, doi: 10.1007/s12553-018-00286-0.

[10] M. Milenkovic, "Internet of Things: System Reference Architecture," 2022. doi: https://doi.org/10.48550/arXiv.2204.01872.

[11] R. Saemaldahr, B. Thapa, K. Maikoo, and E. B. Fernandez, "Reference Architectures for the IoT: A Survey," in *Innovative Systems for Intelligent Health Informatics*, F. Saeed, F. Mohammed, and A. Al-Nahari, Eds., Cham: Springer International Publishing, 2021, pp. 635–646. doi: https://doi.org/10.1007/978-3-030-70713-2_58.

[12] M. M. Krishna, M. E. Sri, and M. Nithya, "AN IOT BASED SMART PILL DISPENSER WITH HEALTH MONITORING," 2023. [Online]. Available: www.irjet.net

[13] K. Karthikeyan, E. D. Babu, S. Ranjith, and S. Arunkumar, "Smart Pill Dispenser for Aged Patients," 2021 International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA), pp. 1–5, 2021, [Online]. Available: https://api.semanticscholar.org/CorpusID:246037348

[14] F. F. Raisa, N. T. Orthy, M. D. I. S. Nibir, H. M. Faysal, and A. S. N. Huda, "Improving Healthcare through an IoT Based Automated Medication Dispenser," 2023 26th International Conference on Computer and Information Technology (ICCIT), pp. 1–6, 2023, [Online]. Available: https://api.semanticscholar.org/CorpusID:268044678

[15] B. V. P. Rajan et al., "Smart Pill Box With Reminder To Consume And Auto-Filling Process Using IOT," 2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), pp. 40–44, 2021, [Online]. Available: https://api.semanticscholar.org/CorpusID:245403223

[16] S. A. Aziz, A. A. B. Sajak, R. Ramly, and M. H. Zulfakar, "IoT Automated Pill Dispenser for Elderly Care," in Proceedings of Seventh International Congress on Information and Communication Technology, X.-S. Yang, S. Sherratt, N. Dey, and A. Joshi, Eds., Singapore: Springer Nature Singapore, 2023, pp. 13–27.

[17] C. Donart, "Automatic Pill Dispenser Instructables." Accessed: May 20, 2024. [Online]. Available: https://www.instructables.com/Automatic-Pill-or-Medication-Dispenser/