Prototype Palang Pintu Kereta Api Otomatis Berbasisi IoT

Yuniarti Lestari [a,1*], Umar Ghoni [b,2], Agung Rimandita [b,3]

[a,b] Teknik Informatika, Universitas Muhammadiyah Brebes, Jl. Pangeran Diponegoro Grengseng No.184 Paguyangan, Brebes 52276, Indonesia
[1] yuniarti.lestari@umbs.ac.id *; [2] umar.ghoni@umbs.ac.id; [3] agungrimandita@gmail.com
* yuniarti.lestari@umbs.ac.id

Submission:25/04/2025, Revision: 25/04/2025, Accepted: 29/04/2025

Abstract

Trains often pass through residential areas and roads, so a gate is needed as a signal for drivers and pedestrians to stop when the train passes. However, many doorstops still operate manually, increasing the risk of accidents due to operator negligence or impatience of road users. To overcome this problem, an IoT-based train doorstop automation system is needed that can be controlled remotely, to increase safety and efficiency. The automatic train doorstop prototype in this research utilizes ESP32, infrared sensors, servo motors, and the Telegram application. System development follows the Extreme Programming method which consists of four stages: planning, design, programming and testing. The test results in this research show that the infrared sensor can detect objects from a distance of 1 to 7 cm and works with a detection angle of between 10 and 100 degrees. When a toy train object is brought close to the sensor, the indicator LED lights up red, indicating the presence of the object, and the servo immediately moves to close the train crossbar up to 100 degrees. After 5 seconds, the servo returns to its initial position of 10 degrees, and the LED turns yellow, indicating that the object is no longer there.

Keywords: railway crossing, Extreme Programming, Internet Of thing, ESP 32, Telegram

Abstrak

Kereta api sering melewati pemukiman dan jalan raya, sehingga diperlukan palang pintu sebagai tanda bagi pengendara dan pejalan kaki untuk berhenti saat kereta melintas. Namun, banyak palang pintu yang masih beroperasi secara manual, meningkatkan risiko kecelakaan akibat kelalaian operator atau ketidaksabaran pengguna jalan. Untuk mengatasi masalah ini, diperlukan sistem otomatisasi palang pintu kereta api berbasis IoT yang dapat dikendalikan dari jarak jauh, guna meningkatkan keselamatan dan efisiensi. *Prototype* palang pintu kereta api otomatis dalam penelitian ini memanfaatkan *ESP*32, sensor *infrared*, motor *servo*, dan aplikasi Telegram. Pengembangan sistem mengikuti metode Extreme Programming yang terdiri dari empat tahapan: perencanaan, desain, pemrograman, dan pengujian. Hasil pengujian dalam penelitian ini menunjukkan bahwa sensor *infrared* dapat mendeteksi objek dari jarak 1 hingga 7 cm dan bekerja dengan sudut deteksi antara 10 hingga 100 derajat. Ketika objek mainan kereta api didekatkan ke sensor, LED indikator menyala merah, menandakan adanya objek, dan *servo* langsung bergerak untuk menutup palang kereta api hingga 100 derajat. Setelah 5 detik, *servo* kembali ke posisi awal 10 derajat, dan *LED* berubah menjadi kuning, menandakan bahwa objek sudah tidak ada.

Kata kunci: palang pintu kereta api, *Extreme Programming*, *Internet Of thing*, *ESP32*, Telegram.

1. Pendahuluan (10pt, tebal)

Kereta Api adalah sarana transportasi berupa kendaraan dengan tenaga gerak diesel, baik berjalan sendiri maupun dirangkaikan dengan kendaraan lainnya, untuk mengangkut kargo atau penumpang. Kereta Api hanya dapat bergerak atau berjalan pada lintasan tertentu yaitu rel yang terbuat dari baja [1]. Rel kereta api terkadang melewati pemukiman umum atau jalan raya. Sehingga dibutuhkan tanda bagi para pengendara atau pejalan kaki yang melewati atau berhenti ketika kereta api melintas. Palang pintu kereta api merupakan sarana untuk mengatur atau mencegah pengendara kendaraan bermotor maupun pejalan kaki melintasi rel kereta api saat ada kereta api yang melintasi palang pintu kereta.

Belum banyaknya palang pintu yang digunakan dan pengoperasiannya masih manual. Sehingga rawan terjadi kecelakaan karena kelalaian operator palang pintu ataupun pengendara atau pejalan kaki yang kurang sabar menunggu palang pintu dibuka. Sebagai contoh di Kota Jember daerah di provinsi Jawa timur dari 346 perlintasan, ada 253 yang tidak dilengkapi palang pintu dan tidak dijaga petugas. Kasus angka kecelakaan di palang pintu pelintasan kereta api di Jawa Timur (Jatim) pada tahun 2022 meningkat 21,5 persen. Jumlah itu meningkat bila dibandingkan dengan tahun 2021. Berdasarkan Data Polda yang didapatkan sepanjang tahun 2022, sebanyak 175 kasus kecelakaan terjadi di pelintasan kereta api, dan jumlah korban yang meninggal sebanyak 105 orang [2].

Berdasarkan permasalahan diatas penulis akan melakukan penelitian tentang perancangan *prototype* palang pintu kereta api otomatis menggunakan *ESP32* dan sensor *infrared* serta motor *servo* sebagai penggeraknya. *ESP32* adalah mikrokontroler terintegrasi yang memiliki fitur lengkap dan kinerja tinggi [3]. Mikrokontroler ini merupakan pengembangan dari *ESP8266. ESP32* memiliki dua prosesor komputasi, satu prosesor untuk mengelola jaringan wifi dan Bluetooth, serta satu prosesor lainnya untuk menjalankan aplikasi [4]. Dilengkapi dengan memori *RAM* yang cukup besar untuk menyimpan data. *Prototype* di bangun menggunakan *ESP32* dengan sensor *Infrared*. Sensor *infrared* digunakan untuk mendeteksi kedatangan kereta. Sensor tersebut juga digunakan Firdaus, Aryo, and Utomo untuk membuat miniatur palang pintu kereta api menggunakan *arduino*. Akan tetapi miniature yang dibuat belum terintegrasi dengan *IoT* dan hanya sebuah miniatur saja [5]. Sebagai contoh, dalam penelitian yang dilakukan oleh Shendy Pratama, miniature yang dibuat masih menggunakan satu motor servo dan satu sensor, sehingga hanya ada satu penghalang jalur kendaraan dari satu sisi saja [6]. Sementara itu, penelitian yang dilakukan oleh Aranthia laudira masih menggunakan Arduino atmega, dimana mikrokontroler ini belum dilengkapi dengan wifi sebagai media *IoT* [7]. *IoT* sendiri adalah singkatan dari "*Internet of Things*", yang merujuk pada jaringan objek fisik yang dilengkapi dengan sensor, perangkat lunak, dan teknologi lainnya untuk saling terhubung dan bertukar data dengan perangkat dan sistem lain melalui internet [8,9,10].

Sebagai media penghubung *IoT* ke *ESP32*, peneliti menggunakan telegram sebagai media operatornya. Telegram merupakan aplikasi yang digunakan sebagai sosial media selain itu telegram memiliki fitur *chatbot* yang dapat digunakan untuk media komunikasi ke *ESP 32* [11]. Dimana di *bot* dibuat 2 metode kontrol yaitu manual dan *automatic.* Dimana jika menggunakan manual kontrol sepenuhnya menggunakan *bot* telegram sedangkan jika automatic *prototype* akan bekerja secara otomatis tanpa kontrol dari *bot* telegram tetapi notifikasi tetap akan muncul di notif telegram.

2. Metode Penelitian

Metode penelitian yang digunakan dalam penelitian ini adalah adalah metode kuantitatif dengan pendekatan experiment. Metode penelitian kuantitatif merupakan metode melalui pengamatan secara sistematis dengan cara mengumpulkan data yang berasal dari hasil pengukuran [12,13].

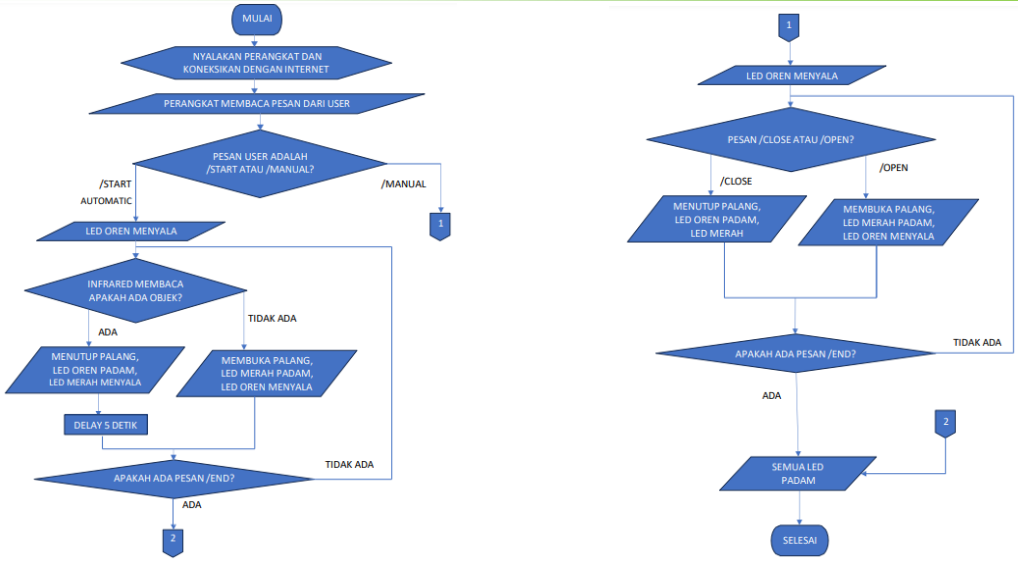
Sedangkan untuk metode pengembangannya menggunakan metode Extreme Programming. *Extreme Programming* (XP) adalah metodologi dalam pengembangan rekayasa perangkat lunak dan juga merupakan satu dari beberapa *agile software* development methodologies yang berfokus pada coding sebagai aktivitas utama di semua tahap pada siklus pengembangan perangkat lunak (*software development lifecycle*). Metodologi ini mengedepankan proses pengembangan yang lebih responsive terhadap kebutuhan customer (”*agile*”) dibandingkan dengan metode-metode tradisional lainnya [14]. Peneliti menggunakan *Extreme Programming* sebagai metode penngembanganya yang mempunyai empat tahapan yang meliputi *planning, design, coding dan testing* [15].

3. Hasil dan Pembahasan

3.1. *Planning* /perencanaan

Peneliti melakukan penelitian menyeluruh tentang kebutuhan dan persyaratan untuk pembuatan *prototype* palang pintu kereta api. Dimulai dengan pemilihan mikrokontroler menggunakan *esp32*, sensor menggunakan sensor *infrared, servo sg90* digunakan sebagai media untuk memutar objek. Peneliti juga mempertimbangkan berbagai aspek seperti mekanisme operasi, material yang digunakan, sistem penggerak dan sistem kontrol.

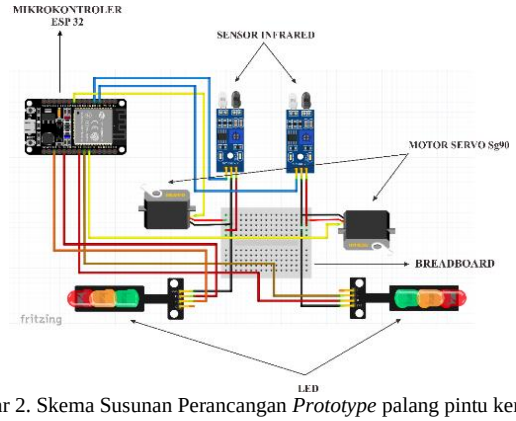
Alat yang dirancang memiliki 2 cara kerja yaitu otomatis dan manual. Dimana jika pengendalian dilakukan secara manual palang pintu di kendalikan menggunakan perintah dari bot telegram yang dikirimkan ke *ESP 32*. Sedangkan jika menggunakan mode otomatis maka kerja alat tergantung pada *prototype* tanpa ada kendali dari *bot* telegram, palang pintu akan menutup atau membuka tergantung inputan yang didapatkan dari sensor infrared.



Gambar 1. *flowchart* palang pintu kereta api

3.2. *Design*

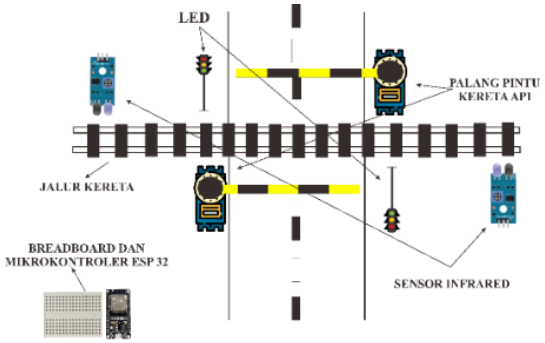
Tahap *design* yang dilakukan dalam penelitian ini meliputi perancangan *prototype* palang pintu kereta api dan perancangan skenario penempatannya. Gambar 2 adalah skema perancangan *prototype* alat dapat dilihat dibawah ini.



Gambar 2. Skema Susunan Perancangan *Prototype* palang pintu kereta api

Konsep susunan perancangan *prototype* palang pintu kereta api yang telah dibuat menggunakan aplikasi *fritzing* sebagai media pembuatannya.

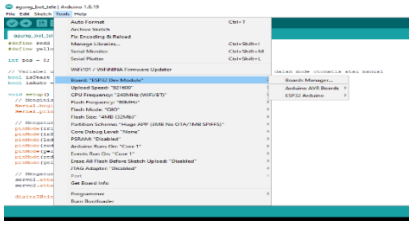
Langkah selanjutnya adalah merancang posisi *prototype* untuk sistem palang pintu kereta. Dalam rancangan ini, *servo* untuk menggerakkan palang pintu kereta akan diposisikan di kedua sisi jalur kereta. Selain itu, akan terdapat dua *LED* dan dua sensor yang akan ditempatkan di masing-masing sisi jalur kereta, berdekatan dengan motor *servo.* Hal ini bertujuan untuk memungkinkan deteksi kereta dan indikasi status palang pintu secara efektif di setiap sisi jalur kereta. Skema penempatan *prototype* palang pintu kereta api dapat dilihat dibawah 3.



Gambar 3. Skema perancangan posisi *prototype*

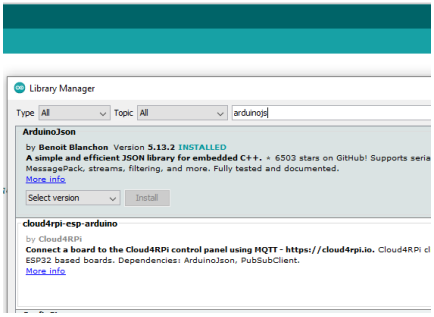
3.3. *Coding*

Proses penulisan kode program di *Arduino IDE* 1.8.19 perlu beberapa penyesuain board pada *Arduino IDE* untuk bisa mengkoneksikan dengan mikrokontroler yang akan digunakan. *Board* yang digunakan pada *Arduino IDE* menggunakan *ESP32 Dev Module* dan *library* menggunakan *library ArduinoJson*. Penyesuaian *board* dan *library* dapat dilihat pada gambar 4.



Gambar 4. penyesuaian *board Esp Dev module*

Penyesuaian *board* diperlukan untuk memastikan pengaturan dan konfigurasi *board* sesuai dengan *hardware* yang di peneliti gunakan.



Gambar 5. library *arduinoJson*

Ketika semua *library* dan *board* sudah disesuaikan dan dipasang selanjutnya adalah proses memasukkan kode program ke dalam aplikasi *Arduino IDE*. Adapun seluruh kode program yang dituliskan pada board *Arduino IDE* pada penelitian ini dapat dijelaskan satu persatu pada uraian dibawah ini.

Kode Program 1 Memanggil CTBot

```
1. #include "CTBot.h"
```

Kode program diatas berfungsi untuk memanggil *library CTBot. Library ArduinoJson* akan otomatis terakses melalui *library CTBot*, sehingga tidak perlu memasukan kode lain untuk memanggil *library* tersebut

Kode program 2 library wifi manager

```
1. #include <WiFiManager.h>
```

Kode program diatas digunakan untuk mengimpor library WiFi Manager ke dalam program, tanpa perlu merubah wifi yang terhubung di codingan lagi.

Kode program 3 servo

```
1. #include <ESP32Servo.h>
```

Digunakan untuk mengatur dan mengendalikan servo motor menggunakan mikrokontroler ESP32.

Kode Program 4 id dan token

```
#definetoken"6997785193:AAEQTBNMHpRVlW4EbxpI6nzCj5yLV0at6_I"
#define chat_id 1130312470
```

Penejelasan program

Token yang ditulis pada program 5 diambil dari GetIDs Bot dan BotFather. Token dan id diperlukan untuk membuat telegram dimana itu adalah syarat untuk membuat bot kita sendiri. Untuk penjelasannya dapat dilihat pada gambar 4 diatas.

Kode program 5 Definisi pin sensor infrared

```
1. #define ir1 34
2. #define ir2 35
```

Kode program diatas untuk menghubungkan output sensor inframerah ke pin 34 dan 35

Kode Program 6 Definisi pin LED

```
1. #define led 2
2. #define red1 5
3. #define yellow1 18
4. #define red2 4
```

```
5. #define yellow2 15
```

Kode program diatas untuk menghubungkan input output pada pin yang dihubungkan ke LED dari pin 5 dan 4 ke led merah, pin18 dan 15 ke orange.

Kode program 7 otomatis atau manual

```
1. bool isStart = false;
2. bool isAuto = false;
```

Kode di atas digunakan untuk menentukan apakah program telah dimulai dan apakah sedang dalam mode otomatis atau manual

Kode program 8 debugging

```
1. Serial.begin(115200);
2. Serial.println("Starting TelegramBot...");
```

Menginisialisasi serial untuk debugging sebesar 115200 bits per detik.

Kode program 9 Mengatur pin LED dan sensor inframerah sebagai input/output

```
1. pinMode(ir1, INPUT);
2. pinMode(ir2, INPUT);
3. pinMode(led, OUTPUT);
4. pinMode(red1, OUTPUT);
5. pinMode(yellow1, OUTPUT);
6. pinMode(red2, OUTPUT);
7. pinMode(yellow2, OUTPUT);
```

Kode program di atas digunakan untuk Mengatur pin LED dan sensor inframerah sebagai input/output. Pengaturan INPUT menunjukkan bahwa pin akan digunakan untuk menerima sinyal masukan, sedangkan OUTPUT menunjukkan pin akan mengirimkan sinyal keluar.

Kode program 10 pin servo 1 dan 2

```
1. servo1.attach(19);
2. servo2.attach(26);
```

Perintah ini menghubungkan (servo1) ke pin 19 pada mikrokontroler Arduino dan (servo2) ke pin 26 pada mikrokontroler Arduino. Dengan ini pin 19 dan 26 digunakan untuk mengirim sinyal kontrol yang diperlukan oleh kedua servo tersebut.

Kode Program 11 koneksi wifi

```
1. bool res = wm.autoConnect("CG BOT TELE");
```

Menggunakan WiFiManager untuk mengelola koneksi WiFi

Kode Program 12 menahan program

```
1. if (!res) {
2. Serial.println("Failed to connect or configure WiFi");
3. } else {
4. // WiFi terhubung
5. Serial.println("Connected to WiFi");
6. Serial.print("SSID: ");
7. Serial.println(WiFi.SSID());
8. Serial.print("IP Address: ");
9. Serial.println(WiFi.localIP());     }
```

Kode program tersebut digunakan untuk Menahan program jika gagal terhubung ke WiFi

Kode Program 13 menguji tes koneksi telegram

```
1. if (myBot.testConnection()) {
2. Serial.println("\ntestConnectionBot OK");
3. } else {
4. Serial.println("\ntestConnectionBot NOK");          }}
```

Kode ini bertujuan untuk menguji koneksi bot Telegram. Jika testConnection() mengembalikan true (koneksi berhasil), maka akan dicetak pesan "testConnectionBot OK". Jika testConnection() mengembalikan false (koneksi gagal), maka akan dicetak pesan "testConnectionBot NOK".

Kode Program 14 Mengatur token bot Telegram

```
1. myBot.setTelegramToken(token);
```

Perintah digunakan untuk mengatur token bot, dimana token ini diperlukkan untuk menghubungkan bot yang telah dibuat ke server telegram.

Kode Program 15 Pesan Bot Telegram

```
1. if (CTBotMessageText == myBot.getNewMessage(msg)) {
2. // Memeriksa apakah pesan berasal dari CHAT_ID yang diotorisasi
3. if (msg.sender.id != chat_id) {
```

```
4. myBot.sendMessage(msg.sender.id, "Access Denied");
5. return;
```

Kode ini membandingkan isi pesan yang diterima (msg) dengan pesan yang diharapkan (CTBotMessageText). Setelah memverifikasi isi pesan, kode memeriksa apakah ID pengirim (msg.sender.id) tidak sama dengan ID yang diotorisasi (chat_id). Hal ini bertujuan untuk memastikan bahwa pengirim pesan memiliki izin akses untuk menjalankan perintah tertentu. Jika ID pengirim tidak sesuai dengan chat_id, bot akan mengirim pesan "Access Denied" kepada pengirim pesan menggunakan myBot.sendMessage(msg.sender.id, "Access Denied");.

Kode Program 16 Memulai program dalam mode otomatis

```
1. digitalWrite(led, HIGH);
2. isStart = true;
3. isAuto = true;
4. myBot.sendMessage(msg.sender.id, "Program is executed automatically");
5. myBot.sendMessage(msg.sender.id, "/manual switches to manual mode");
6. myBot.sendMessage(msg.sender.id, "/end to terminate program");
7. digitalWrite(led, LOW);
```

Penjelasan Kode Program :

1. Mengaktifkan atau menghidupkan LED.
2. yaitu Menetapkan variabel isStart menjadi true. Ini memberi tahu program bahwa perintah "/start" telah diterima dan program telah dimulai.
3. Menetapkan variabel isAuto menjadi true. Ini menunjukkan bahwa program sedang beroperasi dalam mode otomatis.
4. Memberitahu pengguna bahwa program telah dimulai dan berjalan secara otomatis.
5. Memberikan opsi kepada pengguna untuk beralih ke mode manual dengan perintah "/manual".
6. Memberikan opsi kepada pengguna untuk mengakhiri program dengan perintah "/end".

Kode Program 17 Memulai program dalam mode manual

```
1. digitalWrite(led, HIGH);
2. isStart = true;
3. isAuto = false;
4. myBot.sendMessage(msg.sender.id, "Program is executed manually");
5. myBot.sendMessage(msg.sender.id, "/close to close cross gate");
6. myBot.sendMessage(msg.sender.id, "/open to open cross gate");
7. myBot.sendMessage(msg.sender.id, "/auto switches to auto mode");
8. digitalWrite(led, LOW);
```

Penjelasan kode Program :

1. Mengaktifkan atau menghidupkan LED
2. yaitu Menetapkan variabel isStart menjadi true. Ini memberi tahu program bahwa perintah "/start" telah diterima dan program telah dimulai.
3. Mengatur variabel isAuto menjadi false, menunjukkan bahwa sistem tidak dalam mode otomatis saat ini.
4. Mengirim pesan kepada pengirim pesan msg.sender.id melalui objek myBot, memberitahukan bahwa program sedang dieksekusi secara manual.
5. Mengirimkan pesan kepada pengirim pesan msg.sender.id untuk memberi instruksi bahwa dengan menggunakan perintah /close, mereka dapat menutup gerbang (cross gate)
6. Mengirimkan pesan kepada pengirim pesan msg.sender.id untuk memberi instruksi bahwa dengan menggunakan perintah /open, mereka dapat membuka gerbang (cross gate).
7. Mengirimkan pesan kepada pengirim pesan msg.sender.id untuk memberi instruksi bahwa dengan menggunakan perintah /auto, mereka dapat beralih ke mode otomatis.
8. Mematikan LED atau perangkat output lainnya yang terhubung ke pin led dengan nilai rendah (OFF).

Kode Program 18 Menghentikan program

```
1. digitalWrite(led, HIGH);
2. digitalWrite(red1, LOW);
3. digitalWrite(red2, LOW);
4. digitalWrite(yellow1, LOW);
5. digitalWrite(yellow2, LOW);
6. isStart = false;
7. isAuto = false;
8. myBot.sendMessage(msg.sender.id, "Program is terminated");
```

```
9. myBot.sendMessage(msg.sender.id, "/start to start program");
10. digitalWrite(led, LOW);
```

Penjelasan Program :

1. Mengaktifkan atau menghidupkan LED
2. Mematikan output yang terhubung ke pin red1 dengan nilai rendah (LOW)
3. Mematikan output yang terhubung ke pin red2 dengan nilai rendah (LOW).
4. Mematikan output yang terhubung ke pin yellow1 dengan nilai rendah (LOW).
5. Mematikan output yang terhubung ke pin yellow2 dengan nilai rendah (LOW).
6. Mengatur variabel isStart menjadi false, menandakan bahwa program telah dihentikan.
7. Mengatur variabel isAuto menjadi false, menunjukkan bahwa sistem tidak dalam mode otomatis.
8. Mengirim pesan kepada pengirim pesan msg.sender.id melalui objek myBot, memberitahukan bahwa program telah dihentikan.
9. Mengirim pesan kepada pengirim pesan msg.sender.id untuk memberi instruksi bahwa dengan menggunakan perintah /start, mereka dapat memulai program kembali.
10. Mematikan LED atau perangkat yang terhubung ke pin led dengan nilai rendah (LOW).

Kode Program 19 Jika program dalam mode manual

```
1. digitalWrite(led, HIGH);
2. digitalWrite(red1, LOW);
3. digitalWrite(red2, LOW);
4. digitalWrite(yellow1, HIGH);
5. digitalWrite(yellow2, HIGH);
```

Penjelasan program :

1. Mengaktifkan atau menghidupkan LED
2. Mematikan output yang terhubung ke pin red1 dengan nilai rendah (LOW)
3. Mematikan output yang terhubung ke pin red2 dengan nilai rendah (LOW).
4. Menghidupkan output yang terhubung ke pin yellow1 dengan nilai tinggi (HIGH).
5. Menghidupkan output yang terhubung ke pin yellow2 dengan nilai tinggi (HIGH).

Kode Program 20 Menutup palang perlintasan

```
1. digitalWrite(led, HIGH);
2. digitalWrite(red1, HIGH);
3. digitalWrite(red2, HIGH);
4. digitalWrite(yellow1, LOW);
5. digitalWrite(yellow2, LOW);
6. for (pos = 100; pos >= 10; pos--) {
7. servo1.write(pos);
8. servo2.write(pos);
9. delay(15); }
10. myBot.sendMessage(msg.sender.id, "Cross gate is closed");
11. digitalWrite(led, LOW);
```

Penjelasan Program :

Kode program di atas berfungsi untuk mengontrol beberapa perangkat (lampu LED dan servo motor). servo1.write(pos); dan servo2.write(pos)yaitu Mengatur posisi servo motor servo1 dan servo2 sesuai dengan nilai pos yang sedang berubah. Dan pada delay(15); yaitu Memberikan jeda (delay) selama 15 milidetik (ms) setiap kali nilai posisi servo diperbarui. Hal ini digunakan untuk mengontrol kecepatan atau interval perubahan posisi servo.

Kode Program 21 Membuka palang perlintasan

```
1. digitalWrite(led, HIGH);
2. digitalWrite(red1, LOW);
3. digitalWrite(red2, LOW);
4. digitalWrite(yellow1, HIGH);
5. digitalWrite(yellow2, HIGH);
6. for (pos = 10; pos <= 100; pos++) {
7. servo1.write(pos);
8. servo2.write(pos);
9. delay(15); }
10. myBot.sendMessage(msg.sender.id, "Cross gate is opened");
11. digitalWrite(led, LOW);
```

Penjelasan Program :

Digunakan untuk mengontrol perangkat fisik seperti lampu dan servo motor berdasarkan kondisi tertentu. Saat kondisi dibuka (opened), lampu kuning dinyalakan (yellow1 dan yellow2 HIGH), dan servo motor dijalankan dari posisi 10^0 ke 100^0 derajat. Setelah proses selesai, pesan dikirimkan dan lampu LED dimatikan.

Kode Program 22 Mengubah program ke mode otomatis

```
1. digitalWrite(led, HIGH);
2. isStart = true;
3. isAuto = true;
4. myBot.sendMessage(msg.sender.id, "Program is executed automatically");
5. myBot.sendMessage(msg.sender.id, "/manual switches to manual mode");
6. digitalWrite(led, LOW);
7. digitalWrite(led, LOW);
```

Penjelasan Program :

Perintah diatas untuk mengaktifkan lampu LED saat memulai program. Mengatur variabel isStart dan isAuto ke true untuk menandakan bahwa program sedang berjalan dalam mode otomatis. Mengirimkan dua pesan teks kepada pengirim pesan, sebagai notifikasi atau instruksi terkait dengan operasi program. Mematikan lampu LED setelah selesai mengirim pesan-pesan tersebut

Kode Program 23 Membaca sensor inframerah

```
1. Serial.print("state ir1 = ");
2. Serial.println(digitalRead(ir1));
3. Serial.print("state ir2 = ");
4. Serial.println(digitalRead(ir2));
5. Serial.println("");
```

Penjelasan Program :

Kode ini digunakan untuk memantau dan mencetak status dari dua sensor infrared (ir1 dan ir2) ke port serial. Setiap sensor diidentifikasi dengan label "state ir1" atau "state ir2" diikuti oleh nilai sensor tersebut.

Kode Program 24 Jika ada objek di dekat sensor, menutup palang perlintasan

```
1. if (digitalRead(ir1) == 0 || digitalRead(ir2) == 0) {
2. digitalWrite(red1, HIGH);
3. digitalWrite(red2, HIGH);
4. digitalWrite(yellow1, LOW);
5. digitalWrite(yellow2, LOW);
6. // Menutup palang perlintasan
7. for (pos = 100; pos >= 10; pos--) {
8. servo1.write(pos);
9. servo2.write(pos);
10. delay(15);        }
11. myBot.sendMessage(msg.sender.id, "Cross gate is closed");
12. delay(5000);     }
```

Penjelasan Program :

1. DigitalWrite(red1, HIGH); dan digitalWrite(red2, HIGH); akan mengatur pin red1 dan red2 menjadi HIGH, digunakan untuk mengaktifkan LED.
2. DigitalWrite(yellow1, LOW); dan digitalWrite(yellow2, LOW); akan mengatur pin yellow1 dan yellow2 menjadi LOW. Digunakan untuk mematikan LED. Jadi kode ini mengatur kondisi di mana jika salah satu dari sensor infrared (ir1 atau ir2) mendeteksi sesuatu (nilai 0), maka lampu merah (red1 dan red2) akan dinyalakan, sementara lampu kuning (yellow1 dan yellow2) akan dimatikan.
3. servo1.write(pos); dan servo2.write(pos); adalah perintah untuk mengatur posisi servo motor servo1 dan servo2 sesuai dengan nilai pos saat ini.
4. delay(15); adalah perintah untuk menunda eksekusi program selama 15 milidetik. Ini membantu dalam memberikan waktu yang cukup bagi servo untuk mencapai posisi yang diinginkan sebelum langkah berikutnya dalam loop.
5. myBot.sendMessage(msg.sender.id, "Palang pintu terbuka");: Kode ini mengirim pesan "Cross gate is closed" kepada pengguna yang mengirimkan pesan (msg.sender.id) melalui bot (myBot). Pesan ini memberi tahu pengguna bahwa palang perlintasan sedang ditutup.
6. delay(5000);: Ini adalah fungsi delay yang menghentikan eksekusi program selama 5000 milidetik (5 detik). Ini memberi waktu bagi palang perlintasan untuk tetap tertutup selama beberapa saat sebelum dibuka kembali.

Jadi secara keseluruhan, loop ini akan menggerakkan kedua servo motor dari posisi 100 derajat hingga 10 derajat dengan mengurangi posisi sebesar satu derajat pada setiap iterasi, sambil memberikan jeda 15 milidetik setiap kali posisi servo diperbarui.

Kode Program 25 Membuka palang perlintasan

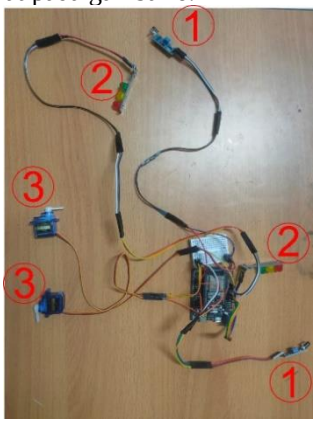
```
for (pos = 10; pos <= 100; pos++) {
servo1.write(pos);
servo2.write(pos);
delay(15);        }
myBot.sendMessage(msg.sender.id, "Cross gate is opened");
delay(200);
```

Penjelasan Program :

1. servo1.write(pos); dan servo2.write(pos); mengatur posisi dari servo motor servo1 dan servo2 sesuai dengan nilai pos saat ini.
2. delay(15); memberi jeda 15 milidetik setelah setiap perubahan posisi servo untuk memastikan servo mencapai posisi yang diinginkan dengan baik
3. Setelah palang perlintasan dibuka sepenuhnya (setelah loop for selesai), baris ini mengirim pesan "Palang Kereta Terbuka" kepada pengguna melalui bot Telegram. Ini memberitahu pengguna bahwa palang perlintasan telah dibuka.
4. Delay (200) milidetik digunakan untuk memberi jeda sebelum program kembali ke awal loop atau melakukan operasi lainnya.

Secara keseluruhan, kode ini mengatur sebuah sistem otomatisasi yang menggunakan *servo* motor untuk mengontrol pembukaan dan penutupan palang perlintasan. Selain itu, melalui integrasi dengan *bot* Telegram, program ini juga memberikan informasi tentang status palang perlintasan kepada pengguna secara real-time.

Pada pengkodingan terdapat proses perakitan alat-alat yang akan dihubungkan dengan mikrokontroler *ESP 32*. Proses perakitan alat dapat dilihat pada gambar 6.



Gambar 6. perakitan alat *prototype* palang pintu kereta api

Perangkat perangkat tersebut terhubung dengan mikrokontroler *ESP 32* melalui pin sesuai kebutuhan masing-masing. Penjelasannya sebagai berikut:

1. Sensor *infrared* terhubung dengan dua pin, yaitu pin D4 dan D35 sebagai input dan outputnya.
2. *LED* terhubung dengan 4 pin yaitu pin D2, D4, D5, D18 sebagai *input* dan *outputnya.*
3. Sensor *infrared* dan *LED* juga terhubung pin *gnd* satu sama lain untuk menciptakan jalur pengembalian arus, sehingga sensor infrared dapat bekerja dengan benar sebagai indikator visual.
4. Motor *servo* terhubung dengan 2 pin yaitu pin D26 dan D19 sebagai *input* dan *outputnya.*

Pada tahap pengkodean diatas, inputan ID Telegram dan token peneliti digunakan sebagai dasar untuk membuat *bot* Telegram sendiri. Token pada Aplikasi Telegram adalah kode rahasia yang diberikan oleh Telegram saat pembuatan *bot* untuk mengotentikasi dan memberi izin akses terhadap *bot* tersebut. Alur pembuatan bot telegram akan dijelaskan sebagai berikut :

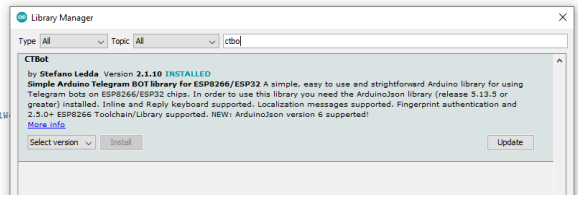
Pembuatan *bot* Telegram untuk menciptakan layanan otomatis yang dapat berinteraksi dengan pengguna melalui *platform* Telegram dapat dilihat pada gambar 7.

Gambar 7 membuat *bot* telegram

Fungsi *Botfather* adalah untuk membuat bot baru yang akan peneliti gunakan untuk media komunikasi dengan alat yang akan dibuat. Sedangkan *Getid* digunakan peneliti untuk mengetahui token id telegram peneliti untuk nantinnya token tersebut dibutuhkan untuk menghubungkan *telegram* dengan alat yang akan di buat.

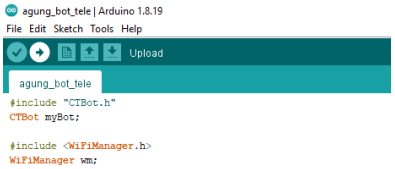
Gambar 8 cara mengetahui token id telegram

Setelah *bot* telegram telah dibuat selanjutnya bisa dilakukan penginstalan *library CTBot*, agar pada proses pengkodean id dan token bot yang telah dibuat dapat di baca oleh *Arduino IDE*. Library *CTBot* dapat dilihat pada gambar 9.



Gambar 9 *library CTBot*

Langkah terakhir seteleh pengkodean dan pembuatan bot adalah proses upload pada board Arduino IDE. Penguplodan adalah langkah-langkah yang diperlukan untuk mentransfer kode program dari komputer ke papan Arduino agar mikrokontroler pada papan tersebut dapat menjalankan kode yang telah ditulis. Proses upload dapat dilihat pada gambar 10.



Gambar 10 Proses Upload pada board Arduino IDE

3.4. Testing

Tahap terakhir dalam pembuatan prototype palang pintu kereta api otomatis setelah selesai melakukan proses pengkodean adalah tahap pengujian alat. Tahap pengujian yang dilakukan meliputi pengujian sensor *infrared*, pengujian motor *servo*, pengujian *prototype*, pengujian *IoT* menggunakan Telegram yang terhubung dengan *prototype* palang pintu kereta api. Tahap *testing* pengujian alat dilakukan untuk mengetahui apakah semua alat sudah bekerja dengan baik atau belum. Beberapa testing yang dilakukan dapat dilihat pada tabel berikut :

1. Pengujian sensor

Pengujian sensor dilakukan untuk menguji apakah sensor dapat mendeteksi objek yang telah ditentukan pada jarak tertentu sesuai keinginan peneliti. Tabel pengujian sesnsor infrared dapat dilihat dibawah ini.

Tabel 1 Tabel Pengujian Sensor

No	Jarak (cm)	Status Deteksi
1	1 cm	Terdeteksi
2	3 cm	Terdeteksi
3	5 cm	Terdeteksi
4	6 cm	Terdeteksi
5	7 cm	Terdeteksi

Keterangan status deteksi :
Sensor berhasil mendeteksi objek pada jarak yang ditentukan. Ini berarti sensor bekerja dengan baik dalam jarak tersebut dan dapat memberikan sinyal kepada mikrokontroler *ESP 32.*

2. Pengujian Servo

Berikut adalah tabel pengujian servo dengan sudut 30 derajat, 60 derajat, 75 derajat, 90 derajat dan 100 derajat, serta status keberhasilan pengujian untuk setiap sudut:

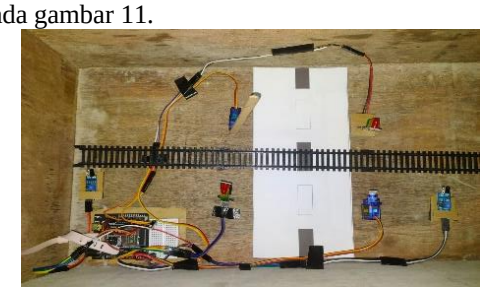
Tabel 2 Tabel Pengujian *Servo Sg90*

No	Sudut (derajat)	Status Pengujian
1	100 derajat	Berhasil
2	90 derajat	Berhasil
3	75 derajat	Berhasil
4	60 derajat	Berhasil
5	30 derajat	Berhasil

Keterangan Status Pengujian :
Servo berhasil bergerak mengikuti perintah sudut yang diberikan peneliti.

3. Pengujian *Prototype*

Pengujian semua komponen *prototype* saat terhubung ke mikrokontroler *ESP 32.* P*rototype* palang pintu kereta api dapat dilihat pada gambar 11.



Gambar 11 *Prototype* palang pintu kereta api

Pengujian semua komponen setelah alat sudah terhubung ke *ESP 32*, dimana masing masing alat yang terhubung ke ESP 32 meliputi 2 sensor infrared sebagai pendeteksi objek, 2 *servo Sg90* sebagai sistem penggerak palang pintu kereta api, 2 *LED* sebagai indikator. Pengujian dilakukan dengan menggerakkan objek mainan kereta api menuju sensor dengan tangan. Sensor berhasil mendeteksi objek, dan indikator *LED* menyala merah sebagai tanda adanya objek yang terdeteksi. Pada saat itu, *servo* langsung bergerak hingga 100 derajat untuk menutup palang kereta api. Setelah 5 detik, *servo* akan kembali ke posisi 10 derajat, dan indikator *LED* berubah menjadi kuning menandakan bahwa objek sudah tidak ada.

4. Pengujian Notifikasi Pesan *bot* Telegram

Peneliti menguji apakah notifikasi sudah bisa muncul pada bot aplikasi telegram dan pengendali manual sudah bisa dilakukan atau masih ada kendala. Hasil pengujian dapat dilihat pada gambar 12.

Gambar 12 Pengujian *Bot* Telegram

Hasil pengujian bot Telegram pada *prototype* palang pintu kereta api menunjukkan bahwa notifikasi berhasil dikirim ketika palang terbuka atau tertutup. Selain itu, dalam mode manual, di mana pengendalian *prototype* dilakukan melalui bot, *prototype* berfungsi dengan lancar.

4. Kesimpulan

Berdasarkan hasil dari penelitian ini, *Prototype* palang pintu kereta api otomatis berbasis *IoT* berhasil dibuat menggunakan sensor *infrared, servo Sg90* dan Mikrokontroler *ESP 32*. pengguna dapat menggunakan

prototype dengan langsung atau menggunakan aplikasi Telegram sebagai pengendali melalui *Bot* Telegram yang sudah dibuat. Hasil testing 4 tahap pengujian mulai dari sensor, servo, pengujian semua komponen dan pengujian bot telegram semua mendapatkan hasil yang baik seperti yang ditunjukan pada sub bab *testing* 3.4.

Hasil yang diperoleh saat pengujian *prototype* yaitu, alat sudah dapat mendeteksi mainan kereta pada jarak 1 - 5cm, *servo* juga sudah dapat bergerak 100 derajat ketika sensor merespon objek yang lewat. Sebagai indikator *LED* bekerja dengan baik dimana ketika sensor mendeteksi objek lampu *LED* berwarna merah dan ketika sedang tidak ada objek lampu *LED* berwarna kuning. *Prototype* yang dibuat juga sudah bisa mengirim notifikasi pesan dan pengendali secara otomatis dan manual juga berjalan dengan baik sesuai keinginan peneliti.

5. Daftar Pustaka

[1] Muliani, Baiq Nurul. "Peningkatkan Kemampuan Kognitif dalam Mengenal Lambang Bilangan melalui Media Model Kereta Api." *Pandawa* 1.1 (2019): 20-39. [Online]. Available: https://ejournal.stitpn.ac.id/index.php/pandawa

[2] Pratama, Wildan. “Angka Kecelakaan di Perlintasan Kereta Api Jatim Tahun 2022 Meningkat 21 Persen,” https://www.suarasurabaya.net/kelanakota/2023/angka-kecelakaan-di-perlintasan-kereta-api-jatim-tahun- 2022-meningkat-21-persen/, Jan. 2023, Accessed: Apr. 18, 2024. [Online]. Available: https://www.suarasurabaya.net/kelanakota/2023/angka-kecelakaan-di-perlintasan-kereta-api-jatim-tahun- 2022-meningkat-21-persen/

[3] Alfiahamzah, Muhamad., Najmuddin., Ansori, Yulian. “RANCANG BANGUN SMART HOME DENGAN GOOGLE ASSISTANT BERBASIS INTERNET OF THINGS MENGGUNAKAN METODE PROTOTYPING DI PERUMAHAN CIUJUNG INDAH”. SCIENTICA (Jurnal Ilmiah Sain dan Teknologi) 2024, 2 (12): 871-878 ISSN 3021-8209. https://jurnal.kolibi.org/index.php/scientica/article/download/3473/3346/12985

[4] Nizam, Muhammad., Yuana, Haris., Wulansari, Zunita. “MIKROKONTROLER ESP 32 SEBAGAI ALAT MONITORING PINTU BERBASIS WEB”. JATI (Jurnal Mahasiswa Teknik Informatika) Vol. 6 No. 2, September 2022. https://ejournal.itn.ac.id/index.php/jati/article/download/5713/3438/

[5] PRATAMA, SHENDY. *PALANG PINTU KERETA API OTOMATIS BERBASIS ARDUINO UNO*. 2019. PhD Thesis. Politeknik Negeri Sriwijaya.

[6] LAUDIRA, Aranthia. *Palang Pintu Perlintasan Kereta Api Otomatis Berbasis Arduino Uno*. 2021. PhD Thesis. Universitas Islam Riau.

[7] FIRDAUS, Muhammad Azzam; UTOMO, Aryo Baskoro. Miniatur palang pintu kereta api otomatis dengan menampilkan kecepatan kereta serta waktu tunggu menggunakan arduino. *Jurnal Teknik Elektro*, 2016, 8.1: 12-17.

[8] AWAL, Hasri. Perancangan prototype smart home dengan konsep internet of thing (iot) berbasis web server. *Majalah Ilmiah UPI YPTK*, 2019, 65-79.

[9] Ilhami, F., Sokibi, P., Amroni (2019). Perancangan Dan Implementasi Prototype Kontrol Peralatan Elektronik Berbasis Internet Of Things Menggunakan Nodemcu. Jurnal Digit Vol. 9, No.2Nov 2019, pp.143~155 ISSN : 2088-589X. https://doi.org/10.51920/jd.v9i2.115

[10] Selay, A., Andigha, G. D., Alfarizi, A., Bintang Wahyudi, M. I., Falah, M. N., Khaira, M., & Encep, M. (2022). Internet Of Things. *Karimah Tauhid*, *1*(6), 860–868. https://doi.org/10.30997/karimahtauhid.v1i6.7633.

[11] Tisna, D. R., Maharani, T., Nugroho, K. T. (2024). Pemanfaatan Chatbot Telegram Untuk Monitoring Dan Kontrol Kualitas Air Menggunakan Esp32. JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika) Vol. 9, No. 3, September 2024, Pp. 1292-1306 ISSN: 2540-8984. https://doi.org/10.29100/jipi.v9i3

[12] SHAHROSI, Muhammad Yasmin Nur; HARIJANTO, Alex; NURAINI, Lailatul. Rancang Bangun Prototype Sistem Monitoring Suhu, Kelembaban dan Intensitas Cahaya pada Tanaman Cabai Berbasis IoT. *STRING (Satuan Tulisan Riset dan Inovasi Teknologi)*, 2023, 7.3: 300-309. doi: 10.30998/string.v7i3.15139.

[13] Sinaga, R. N., Sianturi, R., & Sinaga, C. V. R. (2024). Sistem Pendukung Keputusan untuk Pemilihan Smartphone dengan menggunakan Metode Maut (Multi Attribute Utility Theory). *Innovative: Journal Of Social Science Research*, *4*(6), 1405-1420.

[14] GUMELAR, T.; ASTUTI, R. Trio Sunarni, A., & Studi Magister Ilmu Komputer, P.(2017). *SISTEM PENJUALAN ONLINE DENGAN METODE EXTREME PROGRAMMING*, 9.2.

[15] Crab, A. V., & Aquatic, H. (2025). *Pengembangan Frontend Sistem Dashboard Berbasis Internet Of Things Dan Manajemen Distribusi Vertical Crab House Aquatic*. *12*(1), 1613–1622.