

RANCANG BANGUN *ROY'S FLOOD ALARM DETECTOR* BERBASIS ESP32 CAM

Oleh:

Roy Andi Subagia

Inggit Marodiyah

Teknik Industri

Universitas Muhammadiyah Sidoarjo

January, 2025



Pendahuluan

Desa Kedungbanteng ialah suatu desa yang berlokasi di Kecamatan Tanggulangin Kabupaten Sidoarjo dengan luas area 137,17 hektar. Desa Kedungbanteng berada di lahan rendah yang dikelilingi oleh beberapa sungai besar, termasuk Sungai Brantas dan Sungai Porong. Karena itu, desa ini sangat rawan atau rentan terkena bencana banjir, terutama selama musim hujan. Di samping itu, banjir di Desa Kedungbanteng adalah kejadian yang berlangsung setiap tahun.

Roy's Flood Alarm Detector (RFAD) adalah alat pendeteksi banjir berbasis ESP32-CAM yang menggunakan sensor ultrasonik untuk memantau ketinggian air secara *real-time*. Alat ini dirancang untuk memberikan peringatan dini saat ketinggian air mencapai ambang batas tertentu, sehingga dapat membantu dalam mitigasi risiko banjir. Dengan integrasi ESP32-CAM, RFAD juga mampu mengirimkan foto situasi terkini melalui konektivitas jaringan, menjadikannya solusi modern dalam sistem deteksi banjir.

Pertanyaan Penelitian (Rumusan Masalah)

- Bagaimana merancang sistem peringatan dini banjir yang efektif dengan menggunakan mikrokontroler ESP32-CAM sebagai pengendali utama?
- Bagaimana sistem dapat memberikan informasi secara *real-time* dan akurat kepada pengguna melalui aplikasi Telegram?

Metode

A. Teknik Pengumpulan Data

- Survey Pustaka
- Observasi
- Wawancara

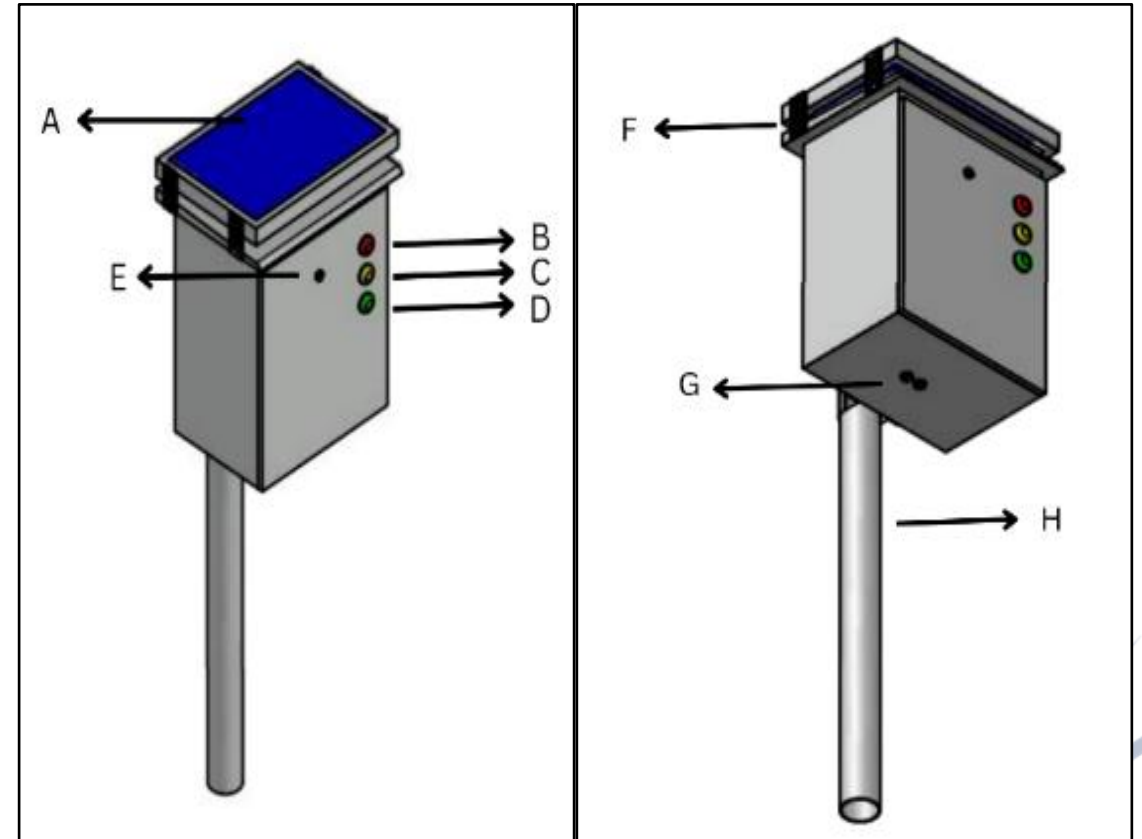
B. Metode Pengolahan Data

- Diagram Blok Sistem
- *Fuzzy Logic*
- *Packet Loss*

Hasil

- **Keterangan:**

- A. Panel Solar Berfungsi untuk mengisi daya Baterai dengan menggunakan kekuatan sinar matahari.
- B. Lampu Indikator Merah dan Buzzer Berfungsi untuk memberikan sinyal darurat jika akan terjadi luapan air di area sekitar alat tersebut serta buzzer akan berbunyi
- C. Lampu Indikator Kuning Menandakan level air dalam level siaga yang artinya akan terjadi luapan air yang mengakibatkan banjir
- D. Lampu Indikator Hijau Berfungsi untuk menunjukkan level air dalam kondisi aman
- E. ESP 32 Cam Berfungsi untuk pemantauan jarak jauh dengan cara mengirim pesan dan gambar real time kondisi area sekitar alat tersebut melalui telegram
- F. Plat Penahan Solar Panel Berfungsi untuk menahan solar panel
- G. Sensor Ultrasonik Berfungsi untuk mengukur level air dari status aman, siaga, hingga darurat
- H. Besi Penyangga Berfungsi untuk menyangga alat tersebut agar dapat mengukur ketinggian air



Pembahasan

- **Pemrograman Arduino**

Software Arduino, biasanya disebut Arduino IDE (Integrated Development Environment), adalah perangkat lunak open-source yang digunakan untuk menulis, mengedit, dan mengunggah kode ke papan mikrokontroler Arduino, seperti Arduino Uno

- **Kode Pemrograman**

```
CameraWebServer.ino  app_httpd.cpp  camera_index.h  camera_pins.h
1  #include <WiFiClientSecure.h>
2  #include "soc/soc.h"
3  #include "soc/rtc_cntl_reg.h"
4  #include "esp_camera.h"
5  #include <UniversalTelegramBot.h>
6  #include <ArduinoJson.h>
7  #include <NewPing.h>
8  #include <WiFiManager.h>
9
10 #define FLASH_LED_PIN 4
11 #define Danger 15 //pin out danger
12 #define Siaga 14 //pin out siaga
13 #define Normal 2 //pin out aman
14 #define TRIGGER_PIN 12
15 #define ECHO_PIN 13
16 #define MAX_DISTANCE 100 // Maksimum jarak yang akan diukur (dalam centimeter)
17 unsigned int distance;
18 unsigned int jarak;
19 NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE);

CameraWebServer.ino  app_httpd.cpp  camera_index.h  camera_pins.h
21 bool flashState = LOW;
22 bool levelAir = false;
23 bool sendPhoto = false;
24 bool wm_nonblocking = false; // change to true to use non blocking
25 WiFiManager wm;
26 String chatId = "1970034271";
27 String BOTtoken = "7068110760:AAHDX9oz8kmTixghSxaQinYC0JCXb5gfc8";
28 WiFiClientSecure clientTCP;
29 UniversalTelegramBot bot(BOTtoken, clientTCP);
30
31 //CAMERA_MODEL_AI_THINKER
32 #define PWDN_GPIO_NUM 32
33 #define RESET_GPIO_NUM -1
34 #define XCLK_GPIO_NUM 0
35 #define SIOC_GPIO_NUM 26
36 #define SIOC_GPIO_NUM 27
37
38 #define Y9_GPIO_NUM 35
39 #define Y8_GPIO_NUM 34
40 #define Y7_GPIO_NUM 39

CameraWebServer.ino  app_httpd.cpp  camera_index.h  camera_pins.h
41 #define Y6_GPIO_NUM 36
42 #define Y5_GPIO_NUM 21
43 #define Y4_GPIO_NUM 19
44 #define Y3_GPIO_NUM 18
45 #define Y2_GPIO_NUM 5
46 #define VSYNC_GPIO_NUM 25
47 #define HREF_GPIO_NUM 23
48 #define PCLK_GPIO_NUM 22
49
50
51 int botRequestDelay = 1000; // mean time between scan messages
52 long lastTimeBotRan; // last time messages' scan has been done
53
54 void handleNewMessages(int numNewMessages);
55 String sendPhotoTelegram();
56 String getReadings(){
57     String status="";
58     unsigned int distance = sonar.ping_cm();
59     if(distance>=1 && distance <=20){
60         status="DARURAT";

CameraWebServer.ino  app_httpd.cpp  camera_index.h  camera_pins.h
61     }else if(distance>20 && distance<=40){
62         status= "Siaga";
63     }
64     else if(distance>40 && distance<=99){
65         status= "Aman";
66     }else{
67         status="Sensor No Detected";
68     }
69     distance = 100-distance;
70     String message = "PERINGATAN!!! Status Level Air Kondisi " + String(status) + "\n";
71     message += "Ketinggian Air: " + String (distance) + " CM \n";
72     return message;
73 }
74
75 void setup(){
76     WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0);
77     Serial.begin(115200);
78     pinMode(FLASH_LED_PIN, OUTPUT);
79     pinMode(Danger, OUTPUT);
80     pinMode(Siaga, OUTPUT);
81     pinMode(Normal, OUTPUT);

CameraWebServer.ino  app_httpd.cpp  camera_index.h  camera_pins.h
82
83     digitalWrite(Danger, HIGH);
84     digitalWrite(Siaga, HIGH);
85     digitalWrite(Normal, HIGH);
86
87     flashState = !flashState;
88     digitalWrite(FLASH_LED_PIN, flashState);
89     delay(400);
90     flashState = !flashState;
91     digitalWrite(FLASH_LED_PIN, flashState);
92
93     WiFi.mode(WIFI_STA);
94     wm.resetSettings();
95     if(wm_nonblocking) wm.setConfigPortalBlocking(false);
96     wm.setConfigPortalTimeout(100);
97     wm.autoConnect("Rays Flood Alarm", "12345678");
98     clientTCP.setCACert(TELEGRAM_CERTIFICATE_ROOT); // Add root certificate for api.telegram.org
99     Serial.print("ESP32-CAM IP Address: ");
100     Serial.println(WiFi.localIP());
101 }
```

Pembahasan

- Kode Pemrograman

```
CameraWebServer.ino  app_httpd.cpp  camera_index.h  camera_pins.h  CameraWebServer.ino  app_httpd.cpp  camera_index.h  camera_pins.h  CameraWebServer.ino  app_httpd.cpp  camera_index.h  camera_pins.h  CameraWebServer.ino  app_httpd.cpp  camera_index.h  camera_pins.h  CameraWebServer.ino  app_httpd.cpp  camera_index.h  camera_pins.h
103 camera_config_t config;
104 config.ledc_channel = LEDC_CHANNEL_0;
105 config.ledc_timer = LEDC_TIMER_0;
106 config.pin_d0 = Y2_GPIO_NUM;
107 config.pin_d1 = Y3_GPIO_NUM;
108 config.pin_d2 = Y4_GPIO_NUM;
109 config.pin_d3 = Y5_GPIO_NUM;
110 config.pin_d4 = Y6_GPIO_NUM;
111 config.pin_d5 = Y7_GPIO_NUM;
112 config.pin_d6 = Y8_GPIO_NUM;
113 config.pin_d7 = Y9_GPIO_NUM;
114 config.pin_xclk = XCLK_GPIO_NUM;
115 config.pin_pclk = PCLK_GPIO_NUM;
116 config.pin_vsync = VSYNC_GPIO_NUM;
117 config.pin_href = HREF_GPIO_NUM;
118 config.pin_sscb_sda = SIOD_GPIO_NUM;
119 config.pin_sscb_scl = SIOC_GPIO_NUM;
120 config.pin_pwdn = PWDN_GPIO_NUM;
121 config.pin_reset = RESET_GPIO_NUM;
122 config.xclk_freq_hz = 20000000;

123 config.pixel_format = PIXFORMAT_JPEG;
124
125 //init with high specs to pre-allocate larger buffers
126 if(psramFound()){
127     config.frame_size = FRAMESIZE_UXGA;
128     config.jpeg_quality = 10; //0-63 lower number means higher quality
129     config.fb_count = 2;
130 } else {
131     config.frame_size = FRAMESIZE_SVGA;
132     config.jpeg_quality = 12; //0-63 lower number means higher quality
133     config.fb_count = 1;
134 }
135
136 // camera init
137 esp_err_t err = esp_camera_init(&config);
138 if (err != ESP_OK) {
139     Serial.printf("Camera init failed with error 0x%x", err);
140     delay(1000);
141     ESP.restart();
142 }

143 // Drop down frame size for higher initial frame rate
144 sensor_t * s = esp_camera_sensor_get();
145 s->set_framesize(s, FRAMESIZE_CIF);
146 String readings = getReadings();
147 bot.sendMessage(chatId, readings, "");
148 //sendPhotoTelegram();
149 }
150
151 void loop(){
152     unsigned int jarak = sonar.ping_cm();
153     if(jarak>=20 && jarak<=40){// siaga
154         Serial.println("status siaga");
155         digitalWrite(Danger, HIGH);
156         digitalWrite(Siaga, LOW);
157         digitalWrite(Normal, HIGH);
158         LevelAir = true;
159         delay(200);
160     }else if(jarak>=1 && jarak<=20){//danger
161         Serial.println("status darurat");
162         digitalWrite(Danger, LOW);
163         digitalWrite(Siaga, HIGH);
164         digitalWrite(Normal, HIGH);
165         LevelAir = true;
166         delay(200);
167     }else{//aman
168         digitalWrite(Danger, HIGH);
169         digitalWrite(Siaga, HIGH);
170         digitalWrite(Normal, LOW);
171         LevelAir = false;
172         delay(200);
173     }
174     if (sendPhoto){
175         Serial.println("Preparing photo");
176         sendPhotoTelegram();
177         sendPhoto = false;
178     }
179 }

184 if(LevelAir){
185     bot.sendMessage(chatId, "Pengumuman Warga KedungBanteng!!!", "");
186     String readings = getReadings();
187     bot.sendMessage(chatId, readings, "");
188     sendPhotoTelegram();
189     LevelAir = false;
190 }
191
192 if (millis() > lastTimeBotRan + botRequestDelay){
193     int numNewMessages = bot.getUpdates(bot.last_message_received + 1);
194     while (numNewMessages){
195         Serial.println("got response");
196         handleNewMessages(numNewMessages);
197         numNewMessages = bot.getUpdates(bot.last_message_received + 1);
198     }
199     lastTimeBotRan = millis();
200 }
201 delay(1000);
202 }
```

Pembahasan

- Kode Pemrograman

```
CameraWebServer.ino  app_httpd.cpp  camera_index.h  camera_pins.h  CameraWebServer.ino  app_httpd.cpp  camera_index.h  camera_pins.h  CameraWebServer.ino  app_httpd.cpp  camera_index.h  camera_pins.h  CameraWebServer.ino  app_httpd.cpp  camera_index.h  camera_pins.h
204 String sendPhotoTelegram(){
205   const char* myDomain = "api.telegram.org";
206   String getAll = "";
207   String getBody = "";
208
209   camera_fb_t * fb = NULL;
210   fb = esp_camera_fb_get();
211   if(!fb) {
212     Serial.println("Camera capture failed");
213     delay(1000);
214     ESP.restart();
215     return "Camera capture failed";
216   }
217
218   Serial.println("Connect to " + String(myDomain));
219
220   if (clientTCP.connect(myDomain, 443)) {
221     Serial.println("Connection successful");
222
223     String head = "--RandomWerdTutorials\r\nContent-Disposition: form-data; name='chat_id'; \r\n\r\n" +
224     String tail = "\r\n--RandomWerdTutorials--\r\n";
225
226     uint16_t imagelen = fb->len;
227     uint16_t extralen = head.length() + tail.length();
228     uint16_t totallen = imagelen + extralen;
229
230     clientTCP.println("POST /bot+"BOTToken+"/sendPhoto HTTP/1.1");
231     clientTCP.println("Host: " + String(myDomain));
232     clientTCP.println("Content-Length: " + String(totallen));
233     clientTCP.println("Content-Type: multipart/form-data; boundary=RandomWerdTutorials");
234     clientTCP.println();
235     clientTCP.print(head);
236
237     uint8_t *fbBuf = fb->buf;
238     size_t fbLen = fb->len;
239     for (size_t n=0;n<fbLen;n+=1024) {
240       if (n+1024<fbLen) {
241         clientTCP.write(fbBuf, 1024);
242         fbBuf += 1024;
243
244       else if (fbLen%1024>0) {
245         size_t remainder = fbLen%1024;
246         clientTCP.write(fbBuf, remainder);
247       }
248     }
249     clientTCP.print(tail);
250
251     esp_camera_fb_return(fb);
252
253     int waitTime = 10000; // timeout 10 seconds
254     long startTimer = millis();
255     boolean state = false;
256
257     while ((startTimer + waitTime) > millis()){
258       Serial.print(".");
259       delay(100);
260       while (clientTCP.available()) {
261         char c = clientTCP.read();
262         if (state==true) getBody += String(c);
263
264         if (c == '\n') {
265           if (getAll.length()==0) state=true;
266           getAll = "";
267         }
268         else if (c != '\r')
269           getAll += String(c);
270         startTimer = millis();
271       }
272       if (getBody.length()>0) break;
273     }
274     clientTCP.stop();
275     Serial.println(getBody);
276   }
277   else {
278     getBody="Connected to api.telegram.org failed.";
279     Serial.println("Connected to api.telegram.org failed.")
280   }
281   return getBody;
282 }
283
284 void handleNewMessages(int numNewMessages){
285   Serial.print("Handle New Messages: ");
286   Serial.println(numNewMessages);
287
288   for (int i = 0; i < numNewMessages; i++){
289     // Chat id of the requester
290     String chat_id = String(bot.messages[i].chat_id);
291     if (chat_id != chatId){
292       bot.sendMessage(chat_id, "Unauthorized user", "");
293       continue;
294     }
295
296     // Print the received message
297     String text = bot.messages[i].text;
298     Serial.println(text);
299
300     String fromName = bot.messages[i].from_name;
301     if (text == "/photo") {
302       sendPhoto = true;
303       Serial.println("New photo request");
304     }
305   }
306 }
```

Pembahasan

- **Kode Pemrograman**

```
CameraWebServerino  app_httpd.cpp  camera_index.h  camera_pins.h
304 }
305 if (text == "/flash") {
306     flashState = !flashState;
307     digitalWrite(FLASH_LED_PIN, flashState);
308     Serial.println("Change flash LED state");
309 }
310 if (text == "/readings"){
311     String readings = getReadings();
312     bot.sendMessage(chatId, readings, "");
313 }
314 if (text == "/start"){
315     String welcome = "Selamat Datang Di Bot Monitoring Kebanjiran\n";
316     welcome += "/photo : mengambil foto baru\n";
317     welcome += "/readings : Pembacaan Level air\n\n";
318     welcome += "Anda akan menerima pesan otomatis saat siaga dan darurat.\n";
319     bot.sendMessage(chatId, welcome, "Markdown");
320 }
321 }
322 }
```

Temuan Penting Penelitian

- Akurasi Deteksi Ketinggian Air
- Pemantauan Visual Real-Time
- Sistem Notifikasi Otomatis
- Biaya dan Efisiensi Operasional
- Respons Cepat dan Keandalan
- Tantangan Jaringan dan Lingkungan
- Potensi Pengembangan Lanjutan

Manfaat Penelitian

- **Deteksi Dini Banjir**

Alat ini dapat mendeteksi kenaikan permukaan air secara otomatis sehingga memungkinkan peringatan dini sebelum terjadi banjir yang lebih parah.

- **Monitoring Real-Time**

Dengan ESP32 Cam dan sensor ultrasonik, sistem dapat memantau kondisi lingkungan secara terus-menerus dan memberikan data secara real-time kepada petugas atau masyarakat

- **Meningkatkan Respons Darurat**

Dengan adanya sistem peringatan dini yang otomatis, proses evakuasi dan penanganan bencana dapat dilakukan lebih cepat, sehingga mengurangi potensi kerugian dan korban

- **Pengembangan Teknologi Lokal**

Penelitian ini mendorong inovasi dalam pengembangan teknologi berbasis Internet of Things (IoT) untuk mitigasi bencana, yang bisa dikembangkan lebih lanjut sesuai dengan kebutuhan lokal

Referensi

- [1] N. Adelia, N. Agustin, I. D. Rahmawati, and U. M. Sidoarjo, “Strategi pemerintah desa dalam upaya penanggulangan banjir di desa kedungbanteng,” vol. 2, no. 7, 2024.
- [2] I. Marodiyah, A. S. Cahyana, and I. R. Nurmalasari, “Risk Mitigation in Sugarcane Planting Using Failure Mode and Effect Analysis Method,” *Int. J. Multidiscip.*, vol. 1, no. 1, pp. 102–107, 2024, doi: 10.61796/ijmi.v1i1.79.
- [3] Y. Septiana, “Design of prototype decision support system for flood detection based on ultrasonic sensor,” *MATEC Web Conf.*, vol.197, pp. 1–4, 2018, doi: 10.1051/mateconf/201819703017.
- [4] A. F. Akbar, W. Cahyadi, and W. Muldayani, “Prototype Sistem Peringatan Dini Berbasis Iot (Internet of Things) untuk Mitigasi Bencana Tsunami di Pantai Pancer, Puger, Jember,” *J. Arus Elektro Indones.*, vol. xx, No. xx, no. 1, pp. 11–16, 2023.
- [5] T. Dzulfikar *et al.*, “Hazard Risk Analysis and Cathodic Maintenance Timing With Total Productive Maintenance Approach Analisis Risiko Bahaya dan Penentuan Waktu Perawatan Katodik Dengan Pendekatan Total Productive Maintenance,” vol. 8, no. 1, pp. 60–69, 2024.
- [6] E. P. Tenda, A. V. Lengkong, and K. F. Pinontoan, “Sistem Peringatan Dini Banjir Berbasis IoT dan Twitter,” *CogITO Smart J.*, vol. 7, no. 1, pp. 26–39, 2021, doi: 10.31154/cogito.v7i1.284.26-39.
- [7] Nur Atikah, Tuti Hartati, Agus Bahtiar, Kaslani, and Odi Nurdiawan, “Sistem Image Capturing Menggunakan ESP32-Cam Untuk Memonitoring Objek Melalui Telegram,” *KOPERTIP J. Ilm. Manaj. Inform. dan Komput.*, vol. 6, no. 2, pp. 49–53, 2022, doi: 10.32485/kopertip.v6i2.141.
- [8] R. D. Pratama, S. Samsugi, and J. P. Sembiring, “Alat Deteksi Ketinggian Air Menggunakan Sensor Ultrasonik Dengan Database,” *J. Tek. dan Sist. Komput.*, vol. 3, no. 1, pp. 45–55, 2022, doi: 10.33365/jtikom.v3i1.1878.
- [9] Rizky Wahyu Pradana, Ganjar Febriyani Pratiwi, and Tri Nur Arifin, “Rancang Bangun Sistem Pemantau Ketinggian Air Otomatis Menggunakan Sensor Ultrasonik (Hc-Sr04) Berbasis Arduino Uno Dengan Antarmuka Komputer Berbasis Microsoft Visual Basic 6.0,” *J. Tek. dan Sci.*, vol. 3, no. 1, pp. 13–24, 2024, doi: 10.56127/jts.v3i1.1212.

Referensi

- [10] R. Dias Valentin, M. Ayu Desmita, and A. Alawiyah, “Implementasi Sensor Ultrasonik Berbasis Mikrokontroler Untuk Sistem Peringatan Dini Banjir,” *Jimel*, vol. 2, no. 2, pp. 2723–598, 2021.
- [11] B. Sugandi and J. Armentaria, “Sistem Penyiraman Tanaman Otomatis Menggunakan Metode Logika Fuzzy,” *J. Appl. Electr. Eng.*, vol. 5, no. 1, pp. 5–8, 2021, doi: 10.30871/jaee.v5i1.2991.
- [12] A. F. Waluyo and T. R. Putra, “Peringatan Dini Banjir Berbasis Internet Of Things (IOT) dan Telegram,” *Infotek J. Inform. dan Teknol.*, vol. 7, no. 1, pp. 142–150, 2024, doi: 10.29408/jit.v7i1.24109.
- [13] E. Febriansyah and E. Nirmala, “Perancangan Sistem Informasi Jual Beli Properti Menggunakan Chat Bot Telegram Yang Terintegrasi Dengan Web Menggunakan Metode Prototype,” *JORAPI J. Res. Publ. Innov.*, vol. 1, no. 2, pp. 279–284, 2023, [Online]. Available: <https://jurnal.portalpublikasi.id/index.php/JORAPI/index>
- [14] F. Saputra, “Prototype Sistem Monitoring Ketinggian Air Menggunakan Telegram,” *J. ICTEE*, vol. 4, no. 2, p. 1, 2023, doi: 10.33365/jictee.v4i2.3161.
- [15] W. A. Wicaksono and L. M. Silalahi, “Rancang Bangun Alat Pendeteksi Banjir Menggunakan Arduino Dengan Metode Fuzzy Logic,” *J. Teknol. Elektro*, vol. 11, no. 2, p. 93, 2020, doi: 10.22441/jte.2020.v11i2.005.
- [16] M. Syahdi Nasution, Muhammad Amin, and Wirda Fitriani, “Smart Sistem Iot Pemberi Pakan Ikan Dengan Menggunakan Metode Time Scheduling Berbasis Mikrokontroller,” *J. Zetroem*, vol. 5, no. 2, pp. 161–164, 2023, doi: 10.36526/ztr.v5i2.3082.
- [17] M. F. Firmansyah and R. P. Astutik, “Prototipe Sistem Peringatan dan Kontrol Jaring Otomatis Dengan Metode Fuzzy Untuk Mitigasi Risiko Lepasnya Ikan Saat Banjir di Tambak Berbasis IoT,” *J. Ris. Rekayasa Elektro*, vol. 6, no. 1, p. 37, 2024, doi: 10.30595/jrre.v6i1.21783.
- [18] Satria Turangga, Martanto, and Yudhistira Arie Wijaya, “Analisis Internet Menggunakan Paramater Quality of Service Pada Alfamart Tuparev 70,” *JATI (Jurnal Mhs. Tek. Inform.*, vol. 6, no. 1, pp. 392–398, 2022, doi: 10.36040/jati.v6i1.4693.

