

BAB IV

HASIL DAN PEMBAHASAN

Pada Bab ini menyajikan hasil dan penelitian yang dilakukan mengenai perancangan kapal Tanpa Awak berbasis ESP32-CAM, hasil penelitian ini di peroleh melalui beberapa rangkaian seperti pengujian *system*, baik dalam lingkungan yang dapat terkendali di kolam rumah

Pembahasan bab ini meliputi kinerja *system* navigasi kapal, *streaming* video dari ESP32—CAM, jangkauan sinyal komunikasi, selain itu dilakukkan analisis terhadap kendala yang dapat ditemukan selama pengujian serta usulan pengembangan lebih lanjut

4.1 Perancangan Dan Perakitan Kapal Area Dengan Menggunakan ESP32-CAM

Kapal Area ini dirancang secara real-time dengan teknologi IOT. Data berupa gambar ataupun video dari lingkungan sekitar yang akan dikirimkan melalui jaringan WI-FI hotspot dari smartphone dari jarak jauh dengan menggunakan aplikasi berbasis web

4.1.1 Perancangan Body Kapal

Pada perancangan kapal ini menggunakan bahan ringan dan tahan air yaitu dengan menggunakan strophom, wadah untuk melindungi komponen komponen yang ada di dalam agar tidak terkena air dan roda yang di rancang khusus untuk menjalan boat di atas perairan



Gambar 4. 1. Body Kapal

4.1.2 Perakitan Sistem Elektronik

Pada perakitan sistem komponen yaitu memasang ESP32-CAM di bagian depan kapal kemudian di sambungkan motor DC ke Motor Driver untuk mengontrol gerakan kapal dan memasang baterai sebagai sumber daya utamanya



Gambar 4. 2. Perakitan Sistem Elektronik

4.2 Pemrograman Esp 32-cam

ESP32-CAM dapat menerjemahkan perintah tersebut dengan mengontrol motor DC melalui *Driver* motor L298N untuk menggerakkan kapal sesuai dengan arah yang di inginkan oleh pengguna, berikut gambar dan penjelasan program Arduino IDE sebagai berikut:

```
#include "esp_camera.h"  
#include <Arduino.h>  
#include <WiFi.h>  
#include <AsyncTCP.h>  
#include <ESPAsyncWebServer.h>  
#include <iostream>  
#include <sstream>
```

Gambar 4. 3, Codingan Include

Pada Gambar 4. 3 **Esp_ Camera.h** sebagai *library* untuk mengontrol kamera ESP 32-CAM. **Wifi.h** = digunakan untuk sebagai koneksi wifi. **AsyncTCP.h** dan **ESPAsyncWebServer.h** digunakan untuk server Websocket yang dapat menangani

komunikasi antara ESP32-CAM dan pengguna Instream dan sstream yang digunakan untuk manipulasi string dalam c++

```
struct MOTOR_PINS
{
    int pinEn;
    int pinIN1;
    int pinIN2;
};

std::vector<MOTOR_PINS> motorPins =
{
    {12, 13, 15}, //RIGHT_MOTOR Pins (EnA, IN1, IN2)
    {12, 14, 2}, //LEFT_MOTOR Pins (EnB, IN3, IN4)
};

#define LIGHT_PIN 4
```

Gambar 4. 4. Codingan Motor

Pada Gambar 3.6 Motor yang dikendalikan dengan 2 set pin. satu untuk motor kanan dan kiri, ENA/ENB merupakan pin untuk kecepatan motor sedangkan IN1,IN2,IN3,IN4 untuk menentukan arah Gerak motor.

```
const char* ssid = "camera WIFI";
const char* password = "123456789";
```

Gambar 4. 5. SSID dan Password

Pada Gambar 4. 5 Untuk menentukan SSID dan Password agar dapat terhubung ke jaringan hotspot dari smartphone

```
AsyncWebServer server(80);
AsyncWebSocket wsCamera("/Camera");
AsyncWebSocket wsCarInput("/CarInput");
uint32_t cameraClientId = 0;
```

Gambar 4. 6. Server, Kamera, Carinput

Pada Gambar 4. 6 Server yang berjalan di port 80 dapat memungkinkan untuk mengakses melalui browser, WsCamera dapat menangani komunikasi gambar dari ESP32-CAM ke klien , wsCarInput yang dapat menangani perintah control kendaraan

```

void rotateMotor(int motorNumber, int motorDirection)
{
  if (motorDirection == FORWARD)
  {
    digitalWrite(motorPins[motorNumber].pinIN1, HIGH);
    digitalWrite(motorPins[motorNumber].pinIN2, LOW);
  }
  else if (motorDirection == BACKWARD)
  {
    digitalWrite(motorPins[motorNumber].pinIN1, LOW);
    digitalWrite(motorPins[motorNumber].pinIN2, HIGH);
  }
  else
  {
    digitalWrite(motorPins[motorNumber].pinIN1, LOW);
    digitalWrite(motorPins[motorNumber].pinIN2, LOW);
  }
}

```

Gambar 4. 7. Coding Rotasi Motor

Pada Gambar Gambar 4. 7 Fungsinya dapat mengatur arah motor berdasarkan input pengguna, jika FORWARD untuk maju, pin IN1 = HIGH dan IN2 = LOW, sedangkan BACKWARD (mundur), pin IN1 = LOW dan IN2 = HIGH , fungsi STOP untuk kedua pin LOW

```

void moveCar(int inputValue)
{
  Serial.printf("Got value as %d\n", inputValue);
  switch(inputValue)
  {
    case UP:
      rotateMotor(RIGHT_MOTOR, FORWARD);
      rotateMotor(LEFT_MOTOR, FORWARD);
      break;
    case DOWN:
      rotateMotor(RIGHT_MOTOR, BACKWARD);
      rotateMotor(LEFT_MOTOR, BACKWARD);
      break;
    case LEFT:
      rotateMotor(RIGHT_MOTOR, FORWARD);
      rotateMotor(LEFT_MOTOR, BACKWARD);
      break;
    case RIGHT:
      rotateMotor(RIGHT_MOTOR, BACKWARD);
      rotateMotor(LEFT_MOTOR, FORWARD);
      break;
    case STOP:
      rotateMotor(RIGHT_MOTOR, STOP);
      rotateMotor(LEFT_MOTOR, STOP);
      break;
    default:
      rotateMotor(RIGHT_MOTOR, STOP);
      rotateMotor(LEFT_MOTOR, STOP);
  }
}

```

Gambar 4. 8. Codingan Move

Pada Gambar 4. 8 menunjukkan bahwa Fungsinya untuk mengontrol pergerakan keseluruhan kendaraan berdasarkan input tombol dari WebSocket

```
void onCameraWebSocketEvent(AsyncWebSocket *server,
                             AsyncWebSocketClient *client,
                             AwsEventType type,
                             void *arg,
                             uint8_t *data,
                             size_t len)
{
    switch (type)
    {
        case WS_EVT_CONNECT:
            Serial.printf("websocket client #%u connected from %s\n", client->id(), client->remoteIP().toString().c_str());
            cameraClientId = client->id();
            break;
        case WS_EVT_DISCONNECT:
            Serial.printf("websocket client #%u disconnected\n", client->id());
            cameraClientId = 0;
            break;
        case WS_EVT_DATA:
            break;
        case WS_EVT_PONG:
        case WS_EVT_ERROR:
            break;
        default:
            break;
    }
}
```

Gambar 4. 9. Camera Web

Pada Gambar 4. 9 Menunjukkan MoveCar yang digunakan untuk mengontrol arah Gerakan kendaraan , Speed untuk kecepatan Motor dan Light untuk mengatur kecerahan lampu

```

void setup(void)
{
  setupPinModes();
  Serial.begin(115200);

  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi..");
  }
  //Serial.println(WiFi.localIP());
  // Print ESP32 Local IP Address
  //Serial.println(WiFi.localIP());

  //WiFi.softAP(ssid, password);
  //IPAddress IP = WiFi.softAPIP();
  //Serial.print("AP IP address: ");
  //Serial.println(IP);

  server.on("/", HTTP_GET, handleRoot);
  server.onNotFound(handleNotFound);

  wsCamera.onEvent(onCameraWebSocketEvent);
  server.addHandler(&wsCamera);

  wsCarInput.onEvent(onCarInputWebSocketEvent);
  server.addHandler(&wsCarInput);

  server.begin();
  Serial.println("HTTP server started");

  setupCamera();
}

```

Gambar 4. 10. Codingan Wifi dan Camera

Pada Gambar 4. 10 Digunakan untuk menghubungkan ESP32-CAM ke WIFI , Menjalankan server HTTP dan WebSocket dan menginisialisasi kamera

4.3 Pengujian Sistem Kendali antara Kapal dan pengguna

Pada Pengujian ini merupakan salah satu aspek dalam penelitian ini karena dapat berkaitan dengan efektivitas kendali kapal dan transmisi video streaming yang dikirimkan melalui ESP32-CAM.

4.3.1 Pengujian Jarak

tujuan pengujian yaitu untuk mengevaluasi seberapa jauh boat dapat beroperasi dengan stabil sesuai maksimal batas jangkauan sinyal

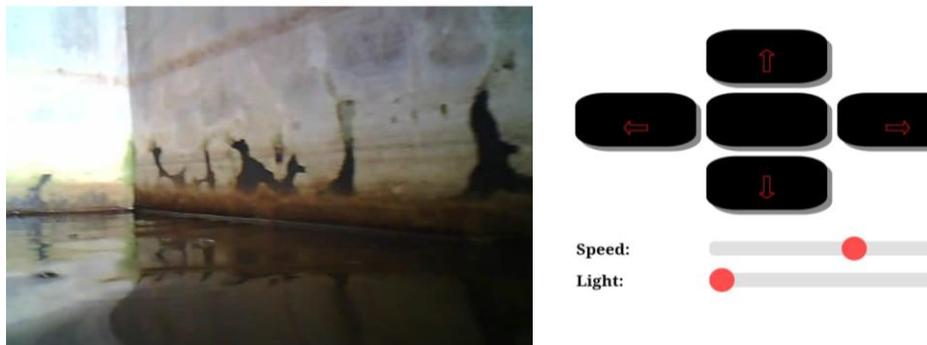
Tabel 4. 1. Pengujian Jarak Kapal

Jarak (Meter)	Respon Perintah	Kecepatan
1	Bekerja dengan sangat baik	Normal
2	Bekerja dengan sangat baik	Normal
3	Bekerja dengan sangat baik	Normal
4	Bekerja dengan baik	Normal
5	Bekerja dengan baik	Normal
6	Bekerja dengan baik	Normal
7	Bekerja dengan baik	Normal
8	Bekerja dengan baik	Normal
9	Bekerja dengan baik	Normal
10	Bekerja dengan baik	Normal
11	Bekerja dengan baik	Normal
12	Bekerja dengan baik	Normal
13	Bekerja dengan baik	Normal
14	Bekerja dengan baik	Normal
15	Bekerja dengan baik	Mulai lambat
16	Kurang respon	lambat
17	Kurang respon	lambat
18	Kurang respon	lambat
19	Kurang tidak baik	Lambat dan delay
20	Kurang tidak baik	Lambat dan delay
21	Kurang tidak baik	Lambat dan delay
22	Kurang tidak baik	Lambat dan delay
23	Respon tidak baik	Lambat dan delay
24	Respon tidak baik	Lambat dan delay
25	Respon tidak baik	Lambat dan delay

Dapat di simpulkan bahwa respon kapal jika jarak lebih dari 20 meter maka respon menjadi semakin lambat, secara menyeluruh boat bergerak dengan normal dan berhasil di jalankan sesuai dengan harapan.

4.3.2 Pengujian Camera ESP32-CAM

Pengujian kamera pada penelitian ini berperan karena ESP32-CAM berperan sebagai sensor utama untuk menangkap dan mengirimkan data visual secara real – time, pengujian ini bertujuan untuk mengevaluasi kualitas gambar atau video yang di hasilkan oleh ESP32-CAM



Gambar 4. 11. Pengujian Kamrea ESP32-CAM

4.4 Pengujian Stabilitas Dan Navigasi Kapal

Pada pengujian ini bertujuan untuk mengetahui stabilitas dan navigasi kapal untuk menunjukkan bahwa kapal dapat beroperasi dengan baik di daerah perairan.

4.4.1 Pengujian Stabilitas

Pengujian stabilitas kapal mampu mengapung dan menjaga keseimbangan di atas permukaan air secara horizontal tanpa adanya miring sedikit pun, daya apung kapal tidak tenggelam meskipun membawa beban elektronik yaitu ESP32-CAM, Motor Driver, Motor DC, Baterai)

4.4.2 Pengujian Navigasi

Pada pengujian navigasi merupakan salah satu aspek yang dimana dapat menentukan sejauh mana boat dapat bergerak secara akurat dan dapat dikendalikan dengan baik di lingkungan perairan. Sistem navigasi kapal dengan

menggunakan ESP32-CAM di uji untuk memastikan bahwa boat dapat bergerak sesuai dengan arah yang di inginkan untuk mempertahankan jalur yang stabil, serta respon kendali yang baik

Tabel 4. 2. Pengujian Navigasi Kapal

Perintah	Keterangan
Maju	Kapal dapat Berjalan Maju
Mundur	Kapal Dapat Berjalan Mundur
Kiri	Kapal Dapat Berbelok ke arah kiri
Kanan	Kapal Dapat Berbelok Ke arah Kanan

Berdasarkan Hasil pengujian yang telah di lakukan, kapal dapat dikendalikan dengan baik dengan menggunakan smartphone yang terhubung melalui ESP32-CAM, yang dimana koneksi tersebut dapat memungkinkan smartphone dapat menjankan perintah sesuai dengan keinginan pengguna , ESP32-CAM menerima perintah dan di proses oleh sistem control boat setelah perintah di terima, sistem tersebut dapat mengatur driver L298N untuk menjalankan kedua motor DC.

4.5 Pengujian Kecepatan Pada Kapal

Pada Pengujian ini dilakukan untuk mengetahui kecepatan kapal dapat bergerak dalam satuan meter perdetik (m/s) yang mampu dikendalikan secara jarak jauh dengan menggunakan antarmuka web

Tabel 4. 3. Pengujian Kecepatan

Percobaan	Jarak	Waktu Tempuh (Detik)	Kecepatan (m/s)
1	4	15	0.27
2	4	12	0.34
3	4	9	0.45

4.6 Hasil Pengujian Sistem Secara Keseluruhan

Hasil Analisa pada pengujian kapal berbasis ESP32-CAM dengan system control WIFI . Pada kapal ini menggunakan Esp32-CAM sebagai sistem kendali kapal yang digunakan sebagai penerima perintah yang dikirim melalui smartphone, motor DC di fungsikan sebagai penggerak boat yang dapat dikontrol melalui smartphone dengan menggunakan browser. ESP32-CAM juga dapat memberikan penerangan atau menyalakan flash dan juga dapat merekam video dengan menggunakan kamera OV2640

Menguji Secara Keseluruhan memiliki tujuan yang dimana dapat melihat hasil kerja dari rangkaian kapal dan aplikasi pada web browser secara menyeluruh, Perancangan sistem boat dengan pengujian memberikan input yang sesuai dengan fungsi yang dimana dapat membantu tim relawan seperti yang berbahaya, pencarian korban yang terkena bencana alam dan Lorong yang tidak dapat di lalui oleh para tim relawan di daerah perairan dan tentunya sangat berbahaya di dipaksa oleh tim relawan