JURNAL EMACS
e-ISSN: 2686-2573
[Engineering, MAthematics and Computer Science] Vol.3 No.3 September 2021: 127-136 DOI: 10.21512/emacsjournal.v3i3.7705

Purwarupa Sistem Kendali Kemudi Kendaraan Roda Empat menggunakan Giroskop pada Realitas Virtual Berbasis Mikrokontroler ESP-WROOM-32

Ahmad Ryan Rivai[1], Bheta Agus Wardijono[2*]

[1,2] Sistem Komputer, STMIK Jakarta STI&K,
Jakarta, Indonesia 11480
ahmadryanrivai@gmail.com; bheta@jak-stik.ac.id

*Correspondence: bheta@jak-stik.ac.id

Abstrak - *Dalam penelitian ini akan dibahas mengenai purwarupa sistem kendali kemudi kendaraan roda empat menggunakan giroskop pada realitas virtual, ini merupakan sistem yang memungkinkan untuk bisa mengemudikan dengan menggunakan gerak ayunan tangan melalui perantara sensor giroskop. Mikro-kontroler yang digunakan yaitu ESP-WROOM-32 karena sudah memiliki fungsionalitas lengkap seperti WiFi & Bluetooth. Giroskop yang terdapat pada sensor terpadu MPU-9250 digunakan sebagai pendeteksi gerak, kemudian datanya setelah diolah mikrokontroler dapat digunakan untuk memanipulasi objek yang berada dalam realitas virtual dan objek yang digunakan yaitu kendaraan roda empat sehingga memungkinkan alat ini mengemudikan kendaraan tersebut.*

Kata Kunci: *ESP-WROOM-32; MPU-9250; Realitas Virtual; Kemudi*

I. PENDAHULUAN

Dengan pesatnya perkembangan teknologi informasi saat ini, ditandai dengan semakin cepatnya komputasi serta besarnya ukuran memori yang dimiliki komputer sekarang, telah membuka gerbang baru bagi teknologi yang sebelumnya tak terpikirkan oleh manusia untuk terwujud. Salah satu dari teknologi itu adalah realitas virtual maupun realitas augmentasi, teknologi ini memungkinkan manusia untuk membuat simulasi dari realitas yang sebenarnya. Contoh penggunaannya ada pada bidang aviasi, antariksa, medis, teknik sipil, bahkan militer.

Namun dengan adanya teknologi baru ada pula masalah baru, diperlukan peralatan serta algoritma yang memungkinkan itu bisa terjadi. Dalam hal peralatan, salah satunya diperlukan sensor yang bisa menjadi penghubung antara data dari dunia nyata ke dalam realitas tersebut, salah satunya bisa menggunakan MPU-9250 untuk mendapatkan data pergerakan dari akselerometer, giroskop, dan magnetometer. Kemudian, data tersebut akan diproses oleh algoritma yang telah ditentukan agar bisa disimulasikan.

Dengan menggunakan mikrokontroler ESP-WROOM-32 data tersebut bisa digunakan untuk mengendalikan kemudi kendaraan pada simulasi realitas virtual yang nantinya bisa diaplikasikan pada keadaan yang sebenarnya. Masalah bisa diidentifikasi sebagai bagaimana cara menghubungkan data pergerakan dari sensor pada dunia nyata ke dalam realitas virtual, sehingga dapat mengendalikan kemudi kendaraan roda empat di dalam realitas tersebut

II. METODE PENELITIAN

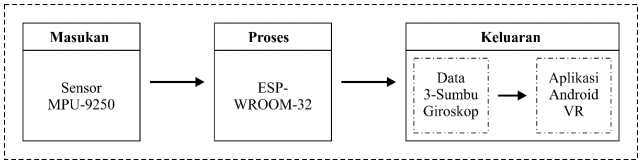
2.1 Gambaran Umum

Sistem ini dirancang untuk mensimulasikan sistem kendali kemudi kendaraan roda empat menggunakan giroskop pada realitas virtual. menggunakan ESP-WROOM-32 sebagai otak pengoperasiannya dan ditunjang dengan komponen sensor MPU-9250 sebagai pendeteksi gerakan pada 3-sumbu giroskop yang akan diproses oleh mikrokontroler dan ditampilkan melalui aplikasi Unreal Engine pada perangkat Android.

2.2 Analisa Perancangan

Tahap perancangan pada alat ini secara garis besar dibagi menjadi beberapa tahap, yaitu : Diagram Blok Rangkaian, Prinsip Kerja Rangkaian, Rangkaian MPU-9250, Rangkaian Pemroses, Rangkaian Keseluruhan, Tampilan Aplikasi VR, *Flowchart* dan Perancangan Program.

2.3 Diagram Blok Rangkaian



Gambar 1: Blok diagram alat

Berdasarkan gambar 1, rangkaian sistem ini menerima masukan dari MPU-9250 lalu akan diproses oleh ESP-WROOM-32 dan menghasilkan keluaran berupa data 3-sumbu giroskop yang akan digunakan pada aplikasi VR Android. ESP-WROOM-32 bekerja dengan memberikan tegangan 3.3V melalui USB yang dihubungkan pada power bank.

Dari diagram blok tersebut dapat dilihat cara kerja setiap blok rangkaian yang ada, berikut uraian cara kerjanya:

1. Sensor MPU-9250 sebagai pendeteksi pergerakan 3-sumbu giroskop.
2. ESP-WROOM-32 sebagai pemrosesan dari blok masukan.
3. Data 3-sumbu giroskop sebagai data output pergerakan sumbu yang akan digunakan pada aplikasi Android VR.
4. Aplikasi Android VR sebagai pengguna data dari 3-sumbu giroskop yang kemudian digunakan sebagai kendali kemudi pada realitas virtual.

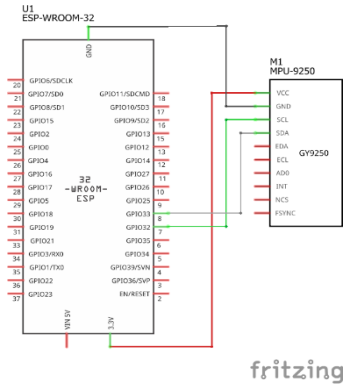
2.4 Prinsip Kerja Rangkaian

Pada penelitian pembuatan purwarupa alat ini dibuat dengan memanfaatkan mikrontroler ESP-WROOM-32 yang diguna-kan untuk memproses data yang diterima oleh sensor MPU-9250. ESP-WROOM-32 juga digunakan untuk mengendalikan sumber daya maupun komponen yang terhubung agar dapat melakukan pemrosesan dan pengolahan data. Mikrokontroler ESP-WROOM-32 dapat bekerja dengan menghubungkan-nya melalui port *micro* USB yang dimilikinya, sehingga memperoleh tegangan yang diperlukan untuk bekerja sebesar 3,3V.

Sensor MPU-9250 digunakan sebagai masukan untuk ESP-WROOM-32 sehingga pada saat sensor melakukan pembacaan, maka data akan dikirimkan ke mikrokontroler untuk melakukan pengendalian kemudi pada aplikasi Android VR. Menggunakan IP (*Internet Protocol*) *address* alat ini dihubungkan dengan aplikasi melalui WiFi, untuk kemudian memperoleh data yang akan digunakan untuk mengendalikan kemudi pada kendaraan roda empat di dalam realitas virtual.

2.5 Rangkaian MPU-9250

MPU-250 adalah sensor pendeteksi gerakan pada 3-sumbu yang berasal dari akselerometer, giroskop, dan magnetomer di dalam chip-nya, memiliki keluaran berupa sinyal digital dengan konversi dan perhitungan oleh sensor itu sendiri melaui DMP. Sensor ini bekerja sebesar 3,3 – 5 VDC (*Direct Current*) dan memiliki range pengukuran yang luas yaitu 2g sampai 16g untuk akselerometer, ±250°/det sampai ±2000°/det untuk Giroskop, dan ±4800µT untuk magnetometer. Komunikasi dan sinyal data menggunakan I^2C untuk komunikasi antara MCU dan MPU-9250. Rangkaian MPU-9250 dapat digunakan seperti gambar 2.



Gambar 2: Rangkaian MPU-9250

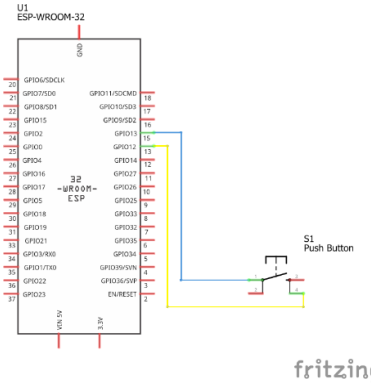
Bisa dilihat pada gambar 2, pin-pin pada sensor MPU-9250 dihubungkan dengan pin-pin pada ESP-WROOM-32. Pin VCC berupa pin bertegangan positif, pada sensor dihubungkan ke pin 3.3V pada ESP-WROOM-32, pin GND berupa arus listrik memerlukan jalur kembali ke ground, pada sensor dihubungkan ke pin GND pada ESP-WROOM-32, pin SCL berupa *Serial Clock* sebagai penghantar sinyal *clock* pada sensor dihubungkan ke pin GPIO32 pada ESP-WROOM-32, dan pin SDA berupa *Serial Data* sebagai jalur transmisi data pada sensor dihubungkan ke pin GPIO33 pada ESP-WROOM-32. Konfigurasi pin sensor MPU-9250 dapat dilihat pada tabel 2.

Tabel 2: Konfigurasi pin sensor MPU-9250

MPU-9250	ESP-WROOM-32
VCC	3,3V
SCL	GPIO32
SDA	GPIO33
GND	GND

2.6 Rangkaian Push Button

Push Button adalah saklar yang berupa tombol dan berfungsi sebagai pemutus atau penyambung arus listrik dari sumber arus ke beban listrik. Pada mikrokontroler *push button* dapat digunakan sebagai input. Rangkaian *push button* dapat digunakan seperti gambar 3.



Gambar 3: Rangkaian *push button*

Pada gambar 3, pin-pin pada *push button* dihubungkan dengan pin-pin pada ESP-WROOM-32. Pin Kaki dihubungkan ke GPIO13 dan pin Kaki 4 pada *push button* dihubungkan ke GPIO12. Pada kode program, pin GPIO12 akan atur sebagai *output High* sementara pin GPIO13 akan diatur sebagai *input* maka ketika ditekan akan

menyebabkan pin GPIO13 membaca *High*. Konfigurasi pin *push button* dapat dilihat pada tabel 3.

Tabel 3: Konfigurasi pin *push button*

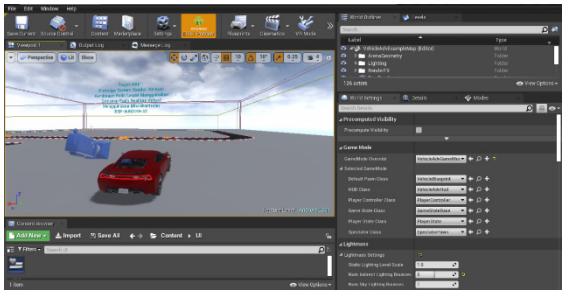
Push Button	ESP-WROOM-32
Kaki 1	GPIO13
Kaki 2	-
Kaki 3	-
Kaki 4	GPIO12

2.7 Analisa Perancangan

Setelah program berjalan, maka ESP-WROOM-32 akan menghubungkan ke AP (*Access Point*) yang telah ditentukan, jika gagal terhubung dengan AP maka akan terus mencoba menghubungkan kembali sampai berhasil terhubung dan mendapatkan alamat IP. Lalu alamat IP ini dapat digunakan untuk menghubungkan dengan ESP-WROOM-32 untuk mendapatkan data dari sensor MPU-9250 berupa pergerakan 3-sumbu giroskop melalui protokol WebSocket.

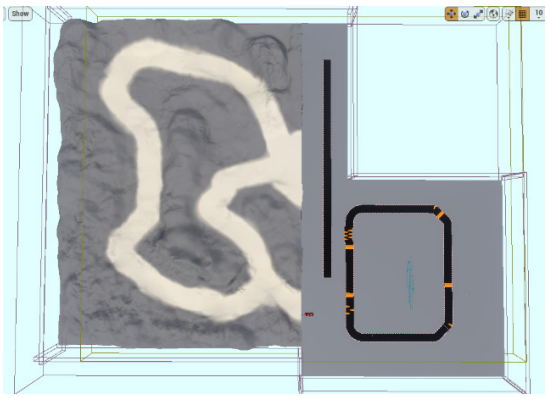
Pada saat rangkaian modul sensor MPU-9250 aktif maka data yang dikirimkan berupa sudah berupa data digital sehingga tidak perlu konversi seperti ADC (*Analog to Digital Converter*) lagi, lalu data yang diterima akan di proses ke mikrokontroler untuk dikirimkan menuju aplikasi Android VR untuk bisa mengendalikan kemudi kendaraan roda empat pada realitas virtual.

2.8 Tampilan Unreal Editor 4



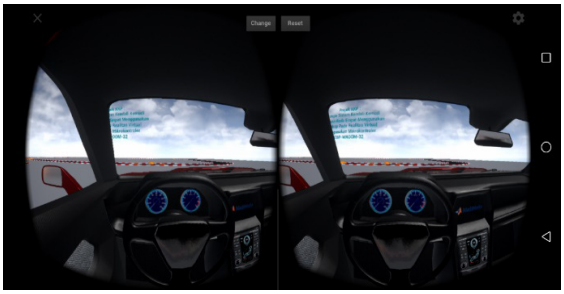
Gambar 4: Tampilan Unreal Editor 4.20

Unreal editor digunakan sebagai penyusun dari aplikasi Android VR, disini dilakukan pembuatan *level* atau *map* yang akan dipakai dalam simulasi, konfigurasi objek kendaraan roda empat yang digunakan, serta berbagai mekanisme yang digunakan untuk koneksi antara mikrokontroller ESP-WROOM-32 dengan Android VR. Pada gambar 5 merupakan jalur yang digunakan sebagai simulasi nantinya.



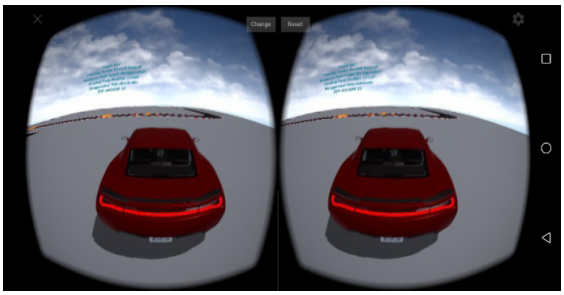
Gambar 6: Bentuk jalur yang digunakan

2.9 Tampilan Aplikasi Android VR



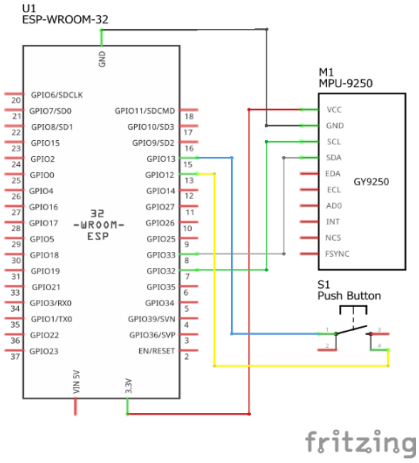
Gambar 7: Mode interior pada Android VR

Pada mode tampilan yang digunakan pada aplikasi Android VR terdapat dua macam yaitu mode interior dan eksterior, bisa dipilih menggunakan tombol *change* yang terdapat pada bagian tengah atas dari aplikasi Android VR. Pada mode interior sudut pandang seperti yang digunakan pengemudi mobil pada umumnya, sedangkan pada mode eksterior sudut pandangnya berada dibagian luar belakang kendaraan. Selain itu terdapat tombol *reset* yang berfungsi untuk mereset posisi kendaraan pada posisi semula ketika pertama kali simulasi dimulai.



Gambar 8: Mode eksterior pada Android VR

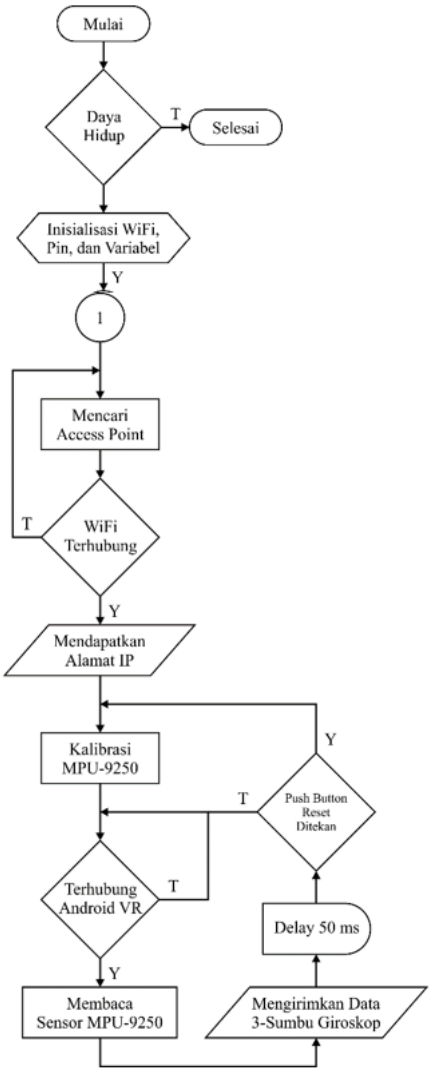
2.10 Tampilan Aplikasi Android VR



Gambar 9: Rangkaian keseluruhan alat

Rangkaian pada gambar 9 merupakan semua rangkaian dari masing-masing bagian yang saling terhubung menjadi satu pada purwarupa sistem kendali kemudi kendaraan roda empat menggunakan giroskop pada realitas virtual berbasis mikrokontroler ESP-WROOM-32, keseluruhan komponen bekerja dan memiliki fungsi masing-masing dan membentuk suatu kesatuan yang utuh. Mikrokontroler ESP-WROOM-32 sebagai otak yang akan pemroses semua input dari sensor MPU-9250 maupun *push button* untuk kemudian dikirimkan pada aplikasi Android VR.

2.11 Diagram Alur Program



Gambar 10: Diagram alur rangkaian

Dalam membuat atau menyusun program pada ESP-WROOM-32, langkah yang harus dilakukan adalah dengan menyusun suatu diagram alur atau *flowchart*, langkah ini akan digunakan sebagai acuan dari pemrograman pada mikrokontroler. Diagram alur dapat dilihat pada gambar 10, berdasarkan diagram alur tersebut dapat dijelaskan bahwa sistem ini pertama kali akan menginisialisasi semua perangkat yang terdiri dari WiFi dan sensor MPU-9250 yang terhubung pada mikrokontroler.

Setelah program berjalan maka ESP-WROOM-32 akan menghubungkan ke AP yang telah ditentukan, jika gagal terhubung dengan AP maka akan terus mencoba menghubungkan kembali sampai berhasil terhubung dan mendapatkan alamat IP, lalu alamat IP ini dapat digunakan untuk mengakses sensor MPU-9250 pada mikrokontroler melalui aplikasi Android VR. Kemudian aplikasi tersebut akan mengambil data melalui sensor MPU-9250 berupa data pergerakan dari giroskop dalam bentuk derajat yang sebelumnya telah diproses oleh mikrokontroler. Jika sensor MPU-9250 mengalami perubahan gerakan maka data yang diperoleh akan berubah setiap 50 milidetik atau dengan siklus ±20Hz, lalu dikirimkan lagi ke aplikasi Android VR.

2.12 Pembuatan Program

Proses pembuatan program dilakukan dengan cara menuliskan program ke dalam memori dari mikrokontroler ESP-WROOM-32 dengan menggunakan software kode editor yaitu VS (*Visual Studio*) Code yang merupakan kode editor yang cukup fleksibel sehingga banyak digunakan untuk berbagai macam keperluan, salah satunya untuk memprogram mikrokontroler sejenis Arduino. Gambar 11 merupakan tampilan software VS Code, tapi perlu diingat untuk menggunakan VS Code untuk memprogram Arduino tetap diperlukan IDE Arduino sebagai pendukungnya.

Gambar 11: Tampilan Visual Studio Code

Setelah membuka VS Code, langkah selanjutnya yaitu menuliskan kode program pada lembar kerja yang telah tersedia. Pada gambar 12 merupakan bagian kode program untuk mendapatkan derajat rotasi 3-sumbu giroskop untuk kemudian dikirimkan kepada aplikasi Android VR.

```
if (imu.dmpUpdateFifo() == INV_SUCCESS) {
  EulerAngles tempEuler;
  int steer = 20;

  imu.computeEulerAngles();

  tempEuler.roll = imu.roll;
  tempEuler.pitch = imu.pitch;
  tempEuler.yaw = imu.yaw;

  // forward-backward
  tempEuler.roll = steering(tempEuler.roll, steer);
  // left-right
  tempEuler.pitch = steering(tempEuler.pitch, steer) * -1;
  // left-right
  tempEuler.yaw = steering(tempEuler.yaw, steer);

  output = String(tempEuler.roll) + ",";
  output += String(tempEuler.pitch) + ",";
  output += String(tempEuler.yaw);

  Serial.println("send!");
  webSocket.sendTXT(0, output);
}
```

Gambar 12: Bagian kode untuk giroskop

Dilakukan pemilihan *if* untuk mengecek sekaligus memperbarui data dari memori FiFo sensor MPU-9250, kemudian data hasil pembaruan disimpan pada variabel tempEuler dengan tipe data EulerAngles, data yang disimpan berupa *roll*, *pitch*, dan *yaw*. Kemudian, data tersebut proses kembali oleh fungsi *steering* untuk merubahnya menjadi format data yang diperlukan untuk kemudi. Pada *pitch* terdapat perkalian terhadap minus satu itu dilakukan untuk membalik arah rotasi dari pergerakan. Setelah itu data simpan pada variabel *ouput* untuk selanjutnya dikirim melalui *method* sendTXT dari WebSocket.

Pada gambar 13 merupakan bagian kode program fungsi *steering* untuk memformat data dari giroskop.

```
float steering(float &degree, int range, int bias = 5) {
  if (degree <= (range + bias) || degree >= (360 - (range
+ bias)))
    return 0.0f;
  else if (degree < 180)
    return 1.0f;
  else
    return -1.0f;
}
```

Gambar 13: Bagian kode untuk fungsi steering

Fungsi *steering* mengembalikan data berupa *float*, mempunyai tiga buah parameter yaitu *degree* berupa *float* dengan *reference*, *range* dan *bias* berupa *integer*. Fungsi ini bertujuan untuk merubah nilai dari giroskop bila *degree* melebihi *range* ditambah *bias* maka akan mengembalikan nilai 1, sedangkan bila *degree* melebihi 360 derajat dikurangi *range* ditambah *bias* maka akan mengembalikan nilai -1, dan jika tidak dari kedua itu mengembalikan nilai 0.

Pada gambar 14 merupakan bagian kode program fungsi inisialiasasi sensor MPU-9250 untuk mengecek kondisi serta mengatur konfigurasi sensor.

```
void initIMU(MPU9250_DMP &_imu) {
  MPU9250_DMP *p = &_imu, imuInit;
  p = NULL;

  if (imuInit.begin(SDA_PIN, SCL_PIN) != INV_SUCCESS)
    while (1) {
      Serial.println("Unable to communicate with MPU-9250");
      Serial.println("Check connections, and try again.\n");
      delay(5000);
    }
  else
    Serial.println("MPU-9250 status green!");

  imuInit.dmpBegin(
      DMP_FEATURE_GYRO_CAL |
      DMP_FEATURE_SEND_CAL_GYRO |
      DMP_FEATURE_6X_LP_QUAT,
      20);

  imuInit.setLPF(188);

  _imu = imuInit;
}
```

Gambar 14: Bagian kode inisialiasasi MPU-9250

Fungsi initIMU mempunyai parameter yaitu _imu berupa tipe data MPU9250_DMP dengan *reference*. Kemudian disimpan pada variabel *pointer* p lalu diatur menjadi NULL untuk mereset nilainya, dan juga dibuat variabel imuInit. Setelah itu dipanggil *method begin* untuk mengecek jika sensor telah terhubung dengan mikrokontroler berdasarkan nilai pin yang kirimkan yaitu SDA_PIN, dan SCL_PIN. Selanjutnya dipanggil *method* dmpBegin untuk mengkonfigurasi DMP dari sensor MPU-9250, nilai 20 pada parameter berfungsi untuk mengatur *refresh rate* dari memori FiFo sebesar 20Hz. Lalu dipanggil *method* setLPF untuk mengatur *Low Pass Filter* pada sensor sebesar 188Hz. Setelah itu nilai imuInit disimpan pada variabel _imu.

Pada gambar 15 merupakan bagian kode program fungsi pushOnce untuk *push button* agar bisa mereset posisi giroskop pada keadaan semula.

```
void pushOnce(int pin, bool &_lock) {
  if (digitalRead(pin) && !_lock) {
    Serial.println("\nRecalibrate Gyro ");
    initIMU(imu);
    _lock = true;
  } else if (!digitalRead(pin))
    _lock = false;
}
```

Gambar 15: Bagian kode untuk fungsi pushOnce

Fungsi pushOnce mempunyai dua buah parameter yaitu pin berupa *integer*, _lock berupa *boolean* dengan *reference*. Fungsi ini selain bertujuan untuk mereset posisi giroskop seperti semula juga membuat *push button* bersifat *once action one push*, jadi meskipun tombol ditekan terus menerus hanya akan ada satu aksi saja atau tidak terjadi pengulangan aksi yang bisa menyebabkan *bug*.

Pada gambar 16 merupakan bagian kode program fungsi taskWifi untuk menghubungkan mikrokontroler dengan AP yang telah ditentukan.

```
void taskWifi(void *param) {
  bool initialSetup = 0;

  while (1) {
    for (int i = 0; i < NUM_NETWORKS; i++) {
      delay(500);
      Serial.print("\nConnecting to ");
      Serial.print(ssidTab[i]);
      WiFi.begin(ssidTab[i], passwordTab[i]);

      for (int j = 0; j < 10; j++) {
        if (WiFi.status() != WL_CONNECTED) {
          delay(500);
          Serial.print(".");
        } else {
          if (initialSetup == 0) {
            server.begin();
            webSocket.begin();
            Serial.println("Connected!");
            webSocket.onEvent(onWebSocketEvent);
            initialSetup = 1;
          }
          Serial.println("done");
          Serial.print("IP address: ");
          Serial.println(WiFi.localIP());

          while (WiFi.status() == WL_CONNECTED) {
            xSemaphoreTake(sem, 5);
            if (xSemaphoreTake(mtx, 5) == pdTRUE) {
              webSocket.loop();
              xSemaphoreGive(mtx);
            }
} } } } } }
```

Gambar 16: Bagian kode untuk fungsi taskWifi

Fungsi taskWifi mempunyai sebuah parameter yaitu param berupa *void* dengan *pointer*. Fungsi ini merupakan *task* yang digunakan untuk menjalankan proses secara *multitasking*. Kemudian dibuat variabel initialSetup berupa *Boolean* dengan nilai awal 0 (atau *false*). Setelah itu akan masuk kedalam perulangan terus menerus *while*, di dalam perulangan ini akan dilakukan proses untuk menghubungkan ke AP melalui *method begin* dari WiFi, jika telah tersambung maka akan memanggil fungsi untuk menjalankan *web server* dan *web socket* melalui *method begin*, sedangkan bila tidak tersambung maka proses ini akan mengulangi terus hingga AP tersambung.

Pada gambar 17 merupakan bagian kode program fungsi onWebSocketEvent untuk untuk menangani *event* dari WebSocket.

```
void onWebSocketEvent(uint8_t num, WStype_t type, uint8_t *payload,
size_t length) {

  switch (type) {
  case WStype_DISCONNECTED: {
    wsconnected = false;
    Serial.printf("[%u] Disconnected\r\n", num);
  } break;

  case WStype_CONNECTED: {
    wsconnected = true;
    Serial.printf("\r\n[%u] Connection from Network \r\n", num);
  } break;

  case WStype_TEXT: {
    Serial.printf("[%u] Text:\r\n", num);
    for (int i = 0; i < length; i++)
      Serial.printf("%c", (char)(*(payload + i)));
    Serial.println();
  } break;

  case WStype_BIN:
  case WStype_ERROR:
  case WStype_FRAGMENT_TEXT_START:
  case WStype_FRAGMENT_BIN_START:
  case WStype_FRAGMENT:
  case WStype_FRAGMENT_FIN:
  default:
    break;
  }
}
```

Gambar 17: Bagian kode onWebSocketEvent

Fungsi onWebSocketEvent mempuny-ai empat buah parameter yaitu num berupa uint8_t, *type* berupa WStype_t, *payload* berupa uint8_t dengan *pointer*, dan *length* berupa size_t. Peran utamanya adalah memberi nilai pada variabel wsconnected ketika WebSocket terjadi koneksi dengan nilai *true* dan ketika diskoneksi dengan nilai *false*.

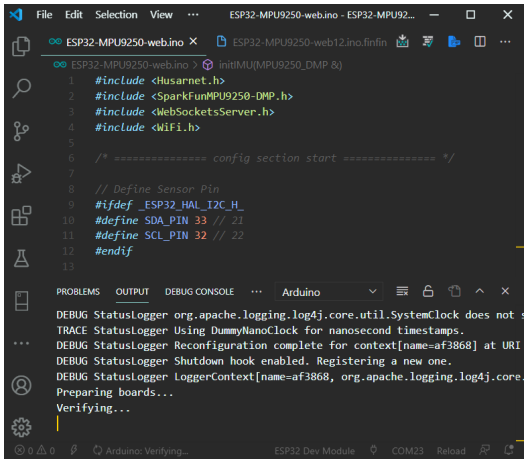
Pada gambar 18 merupakan bagian kode program fungsi taskHTTP untuk untuk menjalankan webserver.

```
void taskHTTP(void *param) {
  String header;

  while (1) {
    while (WiFi.status() != WL_CONNECTED)
      delay(500);
    delay(10);

    client = server.available();

    if (client) {
      Serial.println("New Client.");
      String currentLine = "";
      Serial.printf("connected: %d\r\n", client.connected());
      while (client.connected()) {
        if (client.available()) {
          char c = client.read();
          Serial.write(c);
          header += c;
          if (c == '\n') {
            if (currentLine.length() == 0) {
              client.println("HTTP/1.1 200 OK");
              client.println("Content-type:text/html");
              client.println("Connection: close");
              client.println();
              client.println(html);
              break;
            } else
              currentLine = "";
          } else if (c != '\r')
            currentLine += c;
        }
      }

      header = "";

      client.stop();
      Serial.println("Client disconnected.");
      Serial.println("");
    } else
      delay(200);
  }
}
```

Gambar 18: Bagian kode untuk fungsi taskHTTP

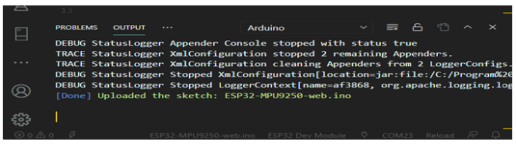
Fungsi taskHTTP mempunyai sebuah parameter yaitu param berupa *void* dengan *pointer*. Fungsi juga merupakan *task* yang digunakan untuk menjalankan proses secara *multitasking*. Webserver disini digunakan untuk melakukan *debugging* terhadap data yang dikirimkan dari mikrokontroler yang berasal dari sensor MPU-9250.

Setelah kode program selesai dibuat, selanjutnya dilakukan *Compile*/*Verify* program untuk memeriksa ada atau tidaknya kesalahan penulisan kode program. Proses *Compile*/*Verify* dapat dilihat pada gambar 19.



Gambar 19: Proses Compile/Verify pada VS Code

Setelah program ter-*Compile*/*Verify* dan apabila tidak terjadi kesalahan maka program dapat diunggah ke ESP-WROOM-32. Sebelum mengunggah program terlebih dahulu dilakukan pengaturan jenis *board* pada mikrokontroler, lalu memilih *ESP32 Dev Module,* selanjutnya dilakukan pengaturan *port* yang digunakan oleh mikrokontroler dalam kasus ini COM23 (nomor *port* bisa berbeda-beda pada tiap komputer). Setelah pengaturan selesai, maka program siap diunggah. Proses unggah program dapat dilihat pada gambar 20.



Gambar 20: Proses unggah program

III. HASIL DAN PEMBAHASAN

3.1 Prosedur Pengujian

Proses pengujian yang akan dilakukan pada alat ini bertujuan untuk mengetahui prinsip kerja dari alat dan program yang telah dibuat sesuai dengan yang akan dicapai sehingga diperoleh suatu kebenaran dari masing bagian - bagian unit yang saling terhubung dalam sistem tersebut.

3.2 Pengujian Alat

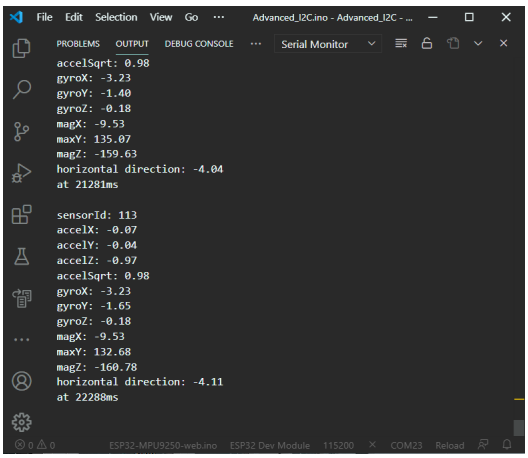
Pada pengujian alat yang akan dilakukan pada purwarupa sistem kendali kemudi kendaraan roda empat menggunakan giroskop pada realitas virtual merupakan uji coba program pada ESP-WROOM-32 untuk dapat mengetahui kinerja dari alat dan program yang telah dibuat sehingga diperoleh suatu kebenaran dari masing-masing bagian dalam suatu sistem tersebut. Pengujian yang dilakukan dibagi menjadi empat bagian yaitu uji teknis, uji program, uji fungsional, dan analisa percobaan. Uji teknis meliputi pengukuran spesifikasi besaran listrik yang bekerja pada komponen. Kemudian, uji fungsional meliputi kinerja dan respon tanggap dalam memberikan data hasil pergerakan dari 3-sumbu pada giroskop ke dalam aplikasi Android VR. Sedangkan analisa percobaan adalah penelaahan dari percobaan yang telah dilakukan.

3.3 Pengujian Teknis

Pada pengujian teknis merupakan pengujian pada tiap rangkaian secara bergantian, seperti dengan melakukan pengukuran spesifikasi besaran listrik yang bekerja pada komponen, dilakukan dengan menggunakan multimeter yang berfungsi untuk mengetahui besaran tegangan atau voltage. Pengukuran dengan mengamati serial monitor, dilakukan pada saat rangkaian pada tiap blok saling terhubung. Rangkaian ini terdiri dari rangkaian sensor MPU-9250, dan rangkaian Push Button yang dilakukan dengan pengukuran pada titik tertentu.

3.4 Pengujian Rangkaian MPU-9250

Pengujian sensor MPU-9250 menggunakan serial monitor untuk mengetahui keluaran yang dihasilkan yaitu data akselerometer, giroskop, dan magnetometer dengan menghubungkan dan memprogram konektivitas pada pin yang terdapat pada sensor MPU-9250 dengan ESP-WROOM-32.



Gambar 21: Hasil keluaran dari MPU-9250

Pada gambar 21, sensor MPU-9250 berhasil terhubung dan keluaran diproses oleh mikrokontroler. Hasil keluaran yang diperoleh merupakan data mentah dari sensor MPU-9250, jadi perlu diproses kembali agar data giroskop dapat menunjukkan derajat kemiringannya.

Pada tes ini perubahan data terjadi setiap 1 detik sekali yang akan kemudian ditampilkan pada serial monitor. Pengujian ini dilakukan dengan membandingan kemiringan dari girosokop terhadap kompas digital yang terdapat pada *smartphone*. Pada tabel 4 menunjukan hasil keluaran derajat kemiringan dari 3-sumbu giroskop pada sensor MPU-9250.

Tabel 1: Hasil giroskop sensor MPU-9250

No	Arah Kompas	Sumbu X (θ)	Sumbu Y (θ)	Sumbu Z (θ)
1	Utara	0.15	0.19	0.02
2	Timur Laut	46.27	45.32	44.56
3	Timur	90.93	91.67	90.12
4	Tenggara	134.98	135.82	135.78
5	Selatan	180.43	180.93	181.02
6	Barat Daya	225.22	224.55	225.09
7	Barat	269.21	270.33	270.22
8	Barat Laut	315.87	316.28	315.67

Posisi sumbu X, Y, dan Z diselaraskan terhadap bidang horizontal ketika dilakukan pengukuran, sehingga akan memudahkan ketika dilakukan pembandingan dengan derajat kompas digital pada *smartphone*.

3.5 Pengujian Rangkaian Push Button

Pengujian rangkaian *push button* dapat menggunakan serial monitor untuk mengetahui input masukan ketika tombol dari *push button* ditekan. Keluaran yang dihasilkan berupa tulisan pada serial monitor yang menandakan *push button* telah ditekan. Berikut gambar 22 menunjukan output dari serial monitor ketika *push button* ditekan.

Gambar 22: Hasil keluaran dari push button

Pada gambar 22, rangkaian *push button* berhasil terhubung dan keluaran diproses oleh mikrokontroler. Hasil keluaran yang diperoleh merupakan tulisan yang menandakan *push button* telah ditekan, ini bisa terjadi karena ketika *push button* ditekan maka output *High* berada pada pin GPIO12 mengalir menuju GPIO13, sehingga Input menjadi *High* dan akan menampilkan tulisan tersebut. Berikut gambar 23 pengujian dengan multitester ketika *push button* tidak ditekan dan gambar 24 ketika ditekan.

Gambar 23: Tegangan pada GPIO12 saat *push button* tidak ditekan

Gambar 24: Tegangan pada GPIO12 saat *push button* ditekan

Pada saat rangkaian *push button* di uji terdapat terdeteksi nilai tegangan dengan menggunakan multimeter. Hasil uji dapat melihat pada tabel 2.

Tabel 2: Hasil pengujian rangkaian *push button*

Kondisi Push Button	Tegangan pin GPIO12	Keterangan
Tidak ditekan	3.31 V	High
Ditekan	0.00 V	Low

3.6 Pengujian Program

Pada pengujian program dilakukan uji coba terhadap output logis dari program apakah telah sesuai dengan apa yang tentukan atau tidak. Pengujian ini dilakukan pada bagian yang cukup krusial dari kode program yaitu pada fungsi *steering* karena bagian inilah yang menyebabkan kendaraan roda empat dalam realitas virtual dapat digerakkan maju, mundur, kiri, dan kanan. Untuk tabel hasil pengujiannya bisa dilihat pada tabel 3, pengujian ini

lakukan dengan menggunakan nilai *random* dari 0 sampai 360 sebanyak 30 kali percobaan. Untuk *response time* yang terdapat pada tabel merupakan hasil dari frekuensi *sampling* yang diterapkan pada sensor sebesar 20 Hz atau 50 *millisecond* setiap kali dilakukan pengambilan data dari giroskop.

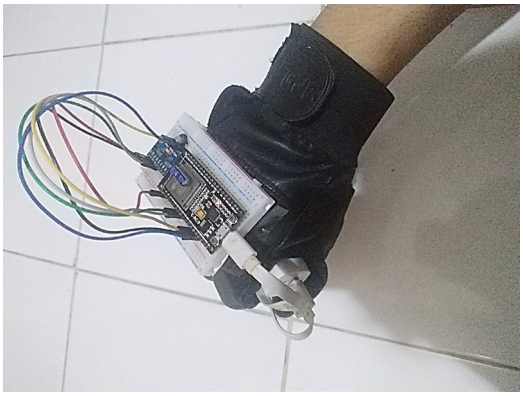
Tabel 3: Hasil output fungsi steering

No.	Nilai Random (θ)	Output	Response Time (ms)
1	0	0	50
2	20	0	50
3	50	1	50
4	59	1	50
5	128	1	50
6	44	1	50
7	14	0	50
8	341	0	50
9	278	-1	50
10	143	1	50
11	32	1	50
12	218	-1	50
13	167	1	50
14	83	1	50
15	44	1	50
16	188	-1	50
17	107	1	50
18	191	-1	50
19	98	1	50
20	347	0	50
21	29	1	50
22	176	1	50
23	56	1	50
24	179	1	50
25	104	1	50
26	311	-1	50
27	68	1	50
28	215	-1	50
29	53	1	50
30	353	0	50

Output yang dihasilkan pada fungsi program yaitu bernilai 1 jika derajat input melebihi 20° sampai 180° sedangkan bila derajat input kurang dari 360° dikurangi 20° sampai 180° maka akan menghasilkan nilai -1 namun apabila tidak keduanya maka akan bernilai 0.

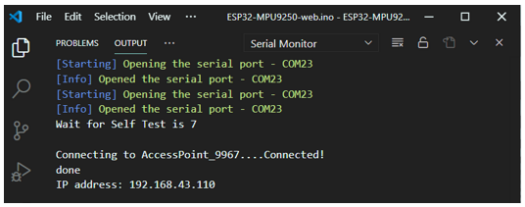
3.7 Pengujian Fungsional

Pada pengujian fungsional yang dilakukan yaitu untuk mengetahui inisialisasi dari setiap kinerja rangkaian apakah telah berjalan dan berfungsi dengan baik sesuai yang diharapkan atau tidak. Pengujian dilakukan untuk mengetahui seluruh rangkaian alat berfungsi dan berjalan sesuai kebutuhan atau tidak. Pengujian di awali dengan menghubungkan mikrokontroler ESP-WROOM-32 ke catu daya dengan tegangan 5V untuk mengaktifkan rangkaian. Maka selanjutnya dilakukan proses pengujian. Gambar 25 menunjukkan ketika alat sedang dalam keadaan aktif.



Gambar 25: Alat dalam keadaan aktif

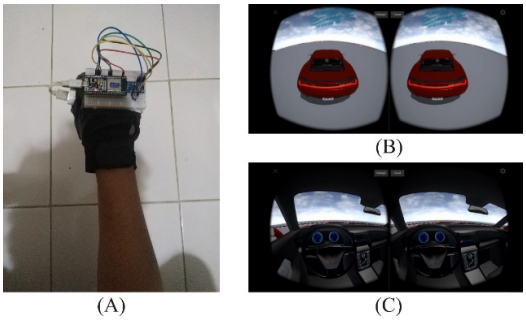
Sebelum mencoba alat, terlebih dahulu mengecek apakah WiFi sudah terhubung ke AP yang telah ditentukan, setelah program dijalankan dan pastikan AP telah menyala. Gambar 26 menunjukan ketika WiFi telah terhubung ke AP.



Gambar 26: ESP-WROOM-32 terhubung ke AP

Kemudian, setelah terhubung maka akan mendapatkan alamat IP yang dapat digunakan untuk menghubungkannya dengan aplikasi Android VR, setelah itu aplikasi dapat digunakan jika *device* sudah terhubung pada AP yang sama seperti yang digunakan mikrokontroler ESP-WROOM-32.

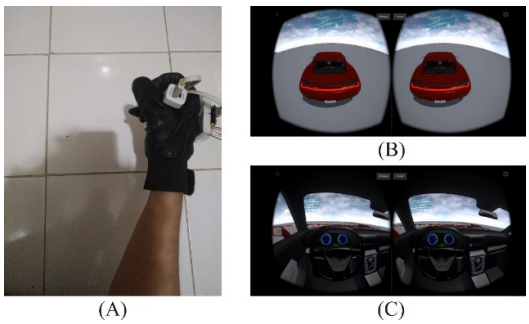
Langkah selanjutnya melakukan pengujian di dalam aplikasi Android VR untuk mengetahui apakah sensor MPU-9250 yang terpasang pada mikrokontroler ESP-WROOM-32 ini sudah berfungsi dengan baik untuk mengendalikan kemudi kendaraan roda empat menggunakan giroskop pada realitas virtual.



(A) (B) (C)

Gambar 27: Posisi siaga alat (A), mode eksterior (B), mode interior (C)

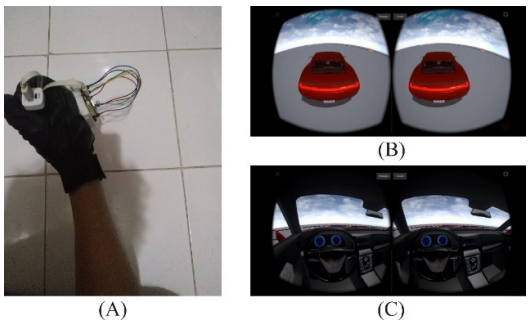
Pada gambar 27 posisi siaga, alat siap untuk menerima input pergerakan dari tangan pengguna, untuk bisa mengendalikan kemudi kendaraan roda empat pada realitas virtual melalui aplikasi Android VR.



(A) (B) (C)

Gambar 28: Posisi maju alat (A), mode eksterior (B), mode interior (C)

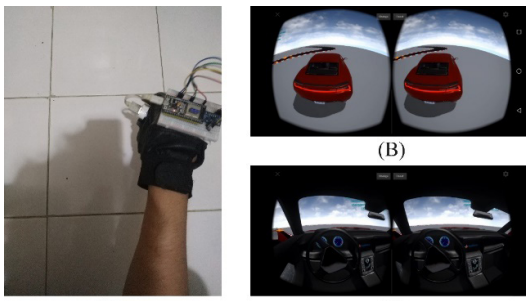
Pada gambar 28 posisi maju, tangan pengguna digerakan ke atas sebesar 20 derajat, alat akan menggerakan maju kendaraan roda empat dalam realitas virtual seperti layaknya pengemudi yang menginjak pedal gas untuk maju pada kendaraan roda empat umumnya.



(A) (B) (C)

Gambar 29: Posisi mundur alat (A), mode eksterior (B), mode interior (C)

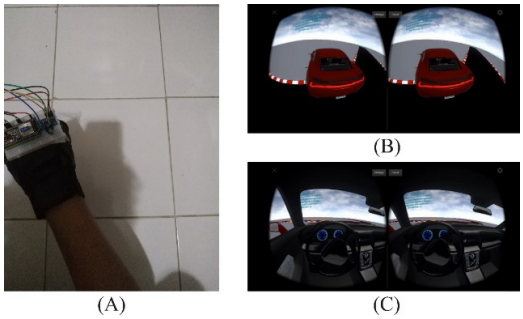
Pada gambar 29 posisi mundur, tangan pengguna digerakan ke bawah sebesar 20 derajat, alat akan menggerakan mundur kendaraan roda empat dalam realitas virtual seperti layaknya pengemudi yang menginjak pedal gas untuk mundur pada kendaraan roda empat umumnya.



(A) (B) (C)

Gambar 30: Posisi kanan alat (A), mode eksterior (B), mode interior (C)

Pada gambar 30 posisi kanan, tangan pengguna digerakan ke kanan sebesar 20 derajat, alat akan menggerakan ke kanan kendaraan roda empat dalam realitas virtual seperti layaknya pengemudi yang menggerakan setir untuk berbelok kanan pada kendaraan roda empat umumnya.



(A) (B) (C)

Gambar 31: Posisi kiri alat (A), mode eksterior (B), mode interior (C)

Pada gambar 31 posisi kiri, tangan pengguna digerakan ke kiri sebesar 20 derajat, alat akan menggerakan ke kiri kendaraan roda empat dalam realitas virtual seperti layaknya pengemudi yang menggerakan setir untuk berbelok kiri pada kendaraan roda empat umumnya.

Pada alat ini juga terdapat *push button* yang berfungsi sebagai tombol rekalibrasi untuk giroskop. Ketika ditekan dalam kondisi alat sedang aktif mengirimkan data menuju aplikasi Android VR, maka LED bawaan ESP-WROOM-32 akan mati sejenak kemudian menyala kembali yang juga menandakan proses rekalibrasi giroskop berhasil dilakukan.

3.8 Analisa Keseluruhan

Secara keseluruhan suatu alat terdiri dari 3 bagian yaitu masukan, pemroses dan keluaran. Masukan pada alat ini berupa sensor MPU-9250 yang berfungsi untuk memberikan data pergerakan 3-Sumbu dari Giroskop ke pemroses data. Pemroses data merupakan bagian penting dari rangkaian tersebut karena memproses data dari masukan yang diterima untuk kemudian diolah menjadi keluaran sesuai dengan kebutuhan, pemroses data tersebut menggunakan ESP-WROOM-32. Data hasil proses yang diperoleh lalu dikirimkan pada aplikasi Android VR, kemudian data dari 3-sumbu giroskop tersebut digunakan untuk mengendalikan kemudi kendaraan roda empat dalam realitas virtual.

IV. KESIMPULAN

Berdasarkan dari percobaan purwarupa sistem kendali kemudi kendaraan roda empat menggunakan giroskop pada realitas virtual berbasis mikrokontroler ESP-WROOM-32, dapat diambil kesimpulan sebagai berikut :

1. Sistem dapat mengindentifikasi pergerakan dari giroskop pada sensor MPU-9250, sehingga dapat digunakan sebagai pengendali kemudi kendaraan roda empat pada realitas virtual.
2. Sistem dapat melakukan interaksi antara dunia nyata dengan realitas virtual dengan melalui sensor MPU-9250 sebagai pendeteksi pergerakan yang terjadi.

Adapun saran agar alat ini bisa dikembangkan untuk kedepannya dapat dilakukan suatu penyempurnaan lebih lanjut, yaitu sebagai berikut :

1. Sensor dapat diganti dengan sensor yang lebih mumpuni seperti 10 DOF IMU agar pergerakan giroskopnya lebih akurat, namun dengan harga yang lebih tinggi karena sepadan dengan fitur yang diperoleh.
2. Memanfaat akselerometer dan magnetometer yang terdapat pada sensor, agar dapat melakukan deteksi pergerakan seperti maju, mundur, kiri, dan kanan.
3. Jika ingin membuat pengembangan lebih lanjut bisa dibuat *user interface* yang lebih mumpuni sehingga user bisa dengan mudah dan nyaman dalam menggunakannya.

DAFTAR PUSTAKA

Nugroho, S.A., Suryawan I K.D., Wardana I N.K. (2015). *Penerapan mikrokontroler sebagai sistem kendali perangkat listrik berbasis Android*. Denpasar. *Jurnal Eksplora Infor-matik*, vol. 4, no. 2, pp. 135-144.

Sosa, M.S.H. (2019). *Perancangan Prototipe Sistem Smarthome Berbasis IoT dengan Smartphone Menggunakan NodeMCU*. Medan. Universitas Sumatera Utara.

Gangenes, E.R. (2016). *Motion capturing machine learning. Bluetooth motion sensors*. UiT Norges Arktiske Uni-versitet. MS thesis.

Hendri, H. (2017). *Sistem Kunci Pintu Otomatis Menggunakan RFID (Radio Frequency Identification) Berbasis Mikrokontroler Arduino Uno R3*. Universitas Putra Indonesia YPTK Padang.

Firman, B. (2016). *Implementasi Sensor IMU MPU6050 Berbasis Serial I2C pada Self-Balancing Robot*. Institut Sains & Teknologi AKPRIND Yogyakarta,.

Panambang, P. (2017). *Rancang Bangun Pembangkit World Dinamis Pada 3D Virtual Game The Labyrinth Menggunakan Algoritma Recursive Division*. Institut Teknologi Sepuluh Nopember.

Nurbadi, M.S. (2018). *Aplikasi Berbasis Virtual Reality untuk Mendukung Proses Pembelajaran Organ Pencernaan Manusia*. Universitas Islam Indonesia, Yogyakarta.

Hutami, A.A. (2019), *Pembuatan Aplikasi Android Augmented Reality Sebagai Media Informasi Politeknik Negeri Sriwijaya*. Politeknik Negeri Sriwijaya, 2019.

Santoso, A.B. (2017). *Pemanfaatan Model Sistem RFID Sebagai Alat Bantu Dalam Penulisan Berita Acara Perkuliahan Berbasis Web Dengan Teknik Pemrograman Berorientasi Objek*. Jurnal Lentera ICT 3.1.