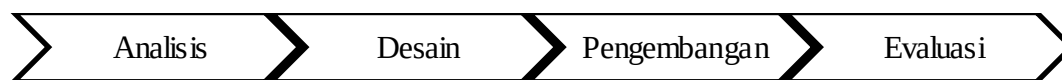


BAB III

METODE PENELITIAN

Metode penelitian *design and development* merupakan pendekatan penelitian yang menitikberatkan pada proses perancangan, pengembangan, serta evaluasi produk atau program dengan tujuan meningkatkan kualitas pembelajaran. Menurut Richey dan Klein (2007), metode ini diartikan sebagai penelitian terstruktur yang mengkaji proses desain, pengembangan, dan evaluasi, dengan tujuan membangun dasar ilmiah untuk menciptakan produk baik instruksional maupun non-instruksional yang baru atau telah disempurnakan sesuai kebutuhan pengembangannya.

Metode penelitian *design and development* merupakan kerangka kerja yang komprehensif yang digunakan sebagai pedoman dalam melakukan penelitian berbasis alat. Pada Gambar 3.1, dijelaskan fase-fase tahapan dalam melakukan metode penelitian *design and development*.



Gambar 3.1 Fase-Fase Proyek pada Metode *Design and Development*

Pada saat penelitian dimulai, proses *Analysis* pada metode *design and development* menganalisa dan meninjau masalah dan kebutuhan yang diperlukan untuk menyelesaikan penelitian. Studi literatur akan dilaksanakan untuk mengetahui isu yang sedang berlangsung. Dari studi literatur yang telah dilakukan, rumusan masalah dan kebutuhan tentang cara merancang sebuah sistem kendali pergerakan *Pan-Tilt* pada sebuah ROV dihasilkan.

Dengan cakupan perancangan desain sistem yang begitu luas, penelitian ini membagi tahap perancangan desain menjadi empat jenis perancangan yaitu: perancangan desain *frame* ROV, perancangan desain *Pan-Tilt*, dan perancangan *wiring* ROV, dan perancangan desain aplikasi.

Setelah seluruh proses perancangan telah selesai, penelitian akan memasuki tahap pengembangan (*development*) dimana seluruh rancangan desain akan dibuat dan dirakit menjadi satu buah kesatuan sistem ROV. Proses pengembangan

melibatkan proses pembangunan rangka ROV, proses pembangunan mekanisme *Pan-Tilt*, proses pembuatan skema elektronik ROV, serta proses pembuatan aplikasi *mobile*

Terakhir, evaluasi dilaksanakan dengan menguji dua buah pengujian *black box*, yaitu: pengujian unit dan pengujian sistem. Pengujian unit merupakan pengujian yang dilakukan untuk mengetahui kinerja komponen sebelum diimplementasikan ke ROV seperti kinerja motor DC dan kinerja pergerakan *Pan-Tilt*. Sementara itu, pengujian sistem merupakan pengujian yang dilakukan setelah ROV dibuat. Pengujian sistem mencakup pengujian kinerja visual ROV dan pengujian pergerakan ROV.

3.1 Analisis

Tahapan analisis merupakan proses identifikasi masalah dan kebutuhan yang diperlukan agar penelitian dapat berjalan dengan optimal. Identifikasi masalah dilakukan dengan melakukan studi literatur tentang permasalahan yang dihadapi ketika melakukan pekerjaan bawah air serta studi literatur tentang pembuatan ROV pada penelitian-penelitian sebelumnya.

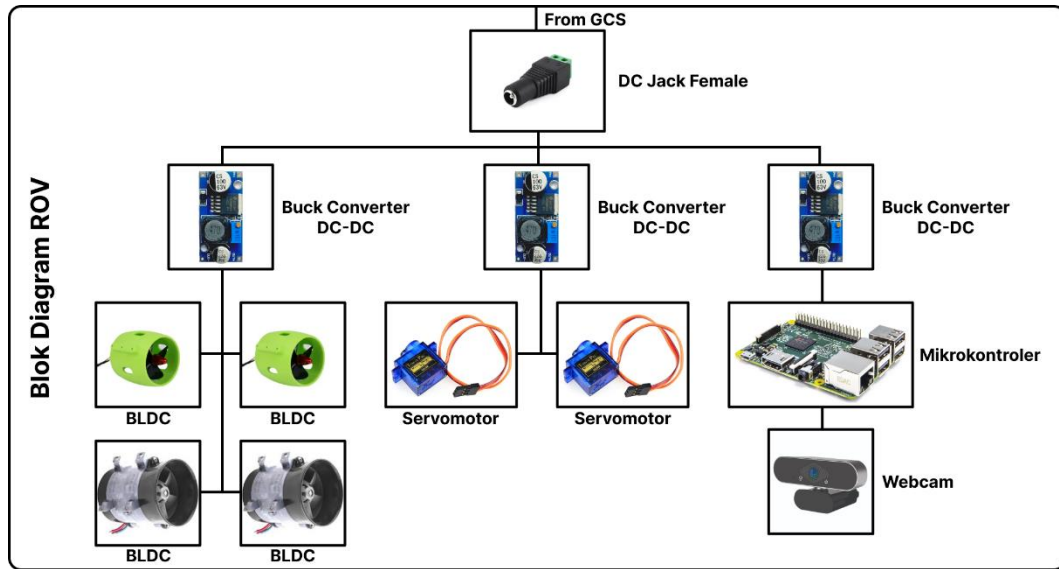
Sementara itu, guna melancarkan proses desain dan pengembangan ROV, dibutuhkan data-data yang memuat tentang alat dan bahan yang dibutuhkan untuk membangun sebuah ROV. Kebutuhan sistem ROV dijabarkan pada sub-bab berikutnya yang memuat tentang proses desain sistem ROV.

3.2 Desain

Perancangan desain sistem melibatkan proses desain dan pembuatan dari penelitian yang diteliti. Perancangan desain sistem memuat diagram arsitektur, blok diagram, alur kerja, *use case diagram*, dan desain-desain ROV serta aplikasi *mobile*.

3.2.1 Blok Diagram Sistem

Sistem ini terdiri dari beberapa komponen utama yang bekerja secara terintegrasi untuk mengoperasikan ROV. Gambar 3.2 menjabarkan blok diagram yang telah dibuat.



Gambar 3.2 Blok Diagram Sistem ROV

Sumber *input* kontrol dan visual berasal dari aplikasi *smartphone* yang dibuat menggunakan *software* Unity. Aplikasi ini berfungsi untuk melakukan manuver ROV serta menampilkan tampilan yang ditangkap oleh kamera *webcam*. Sistem kontrol ROV pada aplikasi ini bekerja dengan mengirimkan data JSON melalui protokol komunikasi MQTT. Bersamaan dengan pengiriman data tersebut, aplikasi ini juga menerima data gambar secara berturut-turut melalui protokol komunikasi HTTP lalu menampilkan gambar-gambar tersebut pada layar sehingga menyerupai *video streaming*. Dalam proses pengiriman dan penerimaan data, *smartphone* dihubungkan dengan Raspberry PI melalui router sebagai *access point*.

Ground Control System (GCS) merupakan unit yang bertugas untuk mengontrol sistem ROV. GCS yang digunakan pada penelitian ini terdiri dari 4 buah komponen utama yaitu: *router*, *smartphone*, *power supply unit*, dan *buck converter*.

Router berfungsi sebagai unit utama yang menghubungkan perangkat Raspberry PI dengan *smartphone* yang digunakan sehingga protokol komunikasi MQTT dan HTTP dapat berjalan dengan lancar pada setiap perangkat. Router memberikan IP tetap untuk Raspberry PI sehingga Raspberry PI memiliki IP tetap dan juga dapat diakses oleh *device* dengan jaringan yang sama. Hal tersebut dapat

memberikan fleksibilitas dalam proses *debugging* karena memungkinkan Raspberry PI untuk diakses tanpa memerlukan monitor, atau peripheral lainnya.

Xiaomi POCO M5S adalah perangkat *smartphone* yang digunakan untuk uji coba pada penelitian ini. *Smartphone* tersebut berfungsi sebagai unit pengendali utama yang bertugas mengirim *input* ke Raspberry Pi via protokol komunikasi MQTT.

Sistem ROV dilengkapi dengan *power supply unit* (PSU) dengan kekuatan daya sebesar 720 watt (24 V 30 A). Menggunakan tegangan AC sebagai *input*, PSU tersebut menyuplai tenaga untuk beberapa komponen seperti: router, Raspberry Pi, dan motor DC. Sistem *powering*, pada ROV ini dibuat sedemikian rupa agar tidak terlalu memakan ruang dan relatif mudah untuk dibawa. Sistem *powering* terdiri dari PSU, modul LM2956, dan router TL-WR840N dengan wadah yang dibuat sebagai *mounting* berbahan dasar PVC board dengan tebal 2 mm.

LM2956 sebagai *buck converter* DC-DC merupakan sebuah modul penurun tegangan DC ke DC dengan *input* tegangan 4-40 V dan mengeluarkan *output* 3-37 V dengan keluaran arus maksimal 3 A. Modul ini digunakan agar GCS dapat disederhanakan dengan menggunakan satu jenis colokan steker AC saja.

Raspberry PI sebagai unit utama pengendali ROV mempunyai tugas yang signifikan terhadap perilaku ROV. Pada mikrokontroler ini, program-program yang dibuat mempunyai tugas-tugas utama seperti: menyalakan webcam, melakukan *streaming* melalui HTTP, menjalankan *mosquitto client* dan *mosquitto broker* untuk menerima data melalui MQTT, melakukan *parsing* terhadap data JSON yang diterima, menggerakkan *servomotor*, dan menggerakkan motor *thruster*. Pertama kali pada saat program berjalan, Raspberry PI akan menjalankan webcam dan melakukan *streaming over IP* dengan menggunakan *library* MJPG-Streamer serta memulai MQTT *Client* dan MQTT *Broker* menggunakan *library* *mosquitto*. Kemudian, Raspberry PI akan menginisialisasi ESC terlebih dahulu untuk mempersiapkan motor penggerak. Selanjutnya, Raspberry PI akan melakukan *parsing* terhadap data JSON yang diterima lalu data JSON tersebut akan digunakan sebagai variabel untuk mengatur unit motor penggerak sekaligus juga mengatur gerakan *servomotor Pan-Tilt*.

Mekanisme *Pan-Tilt* pada penelitian ini dibuat sesederhana mungkin dengan menggunakan dua buah *servomotor*. Servo-servo tersebut merepresentasikan dua buah *degree of freedom* yaitu: *pitch* dan *yaw* pada kendali kamera *Pan-Tilt* ROV. Data perpindahan sudut *Pan-Tilt* diproses dengan data yang diperoleh dari JSON yang telah di-*parsing* sebelumnya. Pada ujung *Pan-Tilt*, dipasangkan sebuah kamera webcam yang berfungsi untuk menangkap gambar.

3.2.2 Diagram Arsitektur Sistem

Pada Gambar 3.3 di bawah, kendali ROV dilakukan melalui sebuah aplikasi yang dikembangkan menggunakan *software* Unity dan dijalankan pada smartphone. Smartphone terhubung ke router TP-Link WR840N melalui koneksi Wi-Fi. Router ini berfungsi sebagai penghubung utama antara sistem kendali pada smartphone dengan unit ROV.

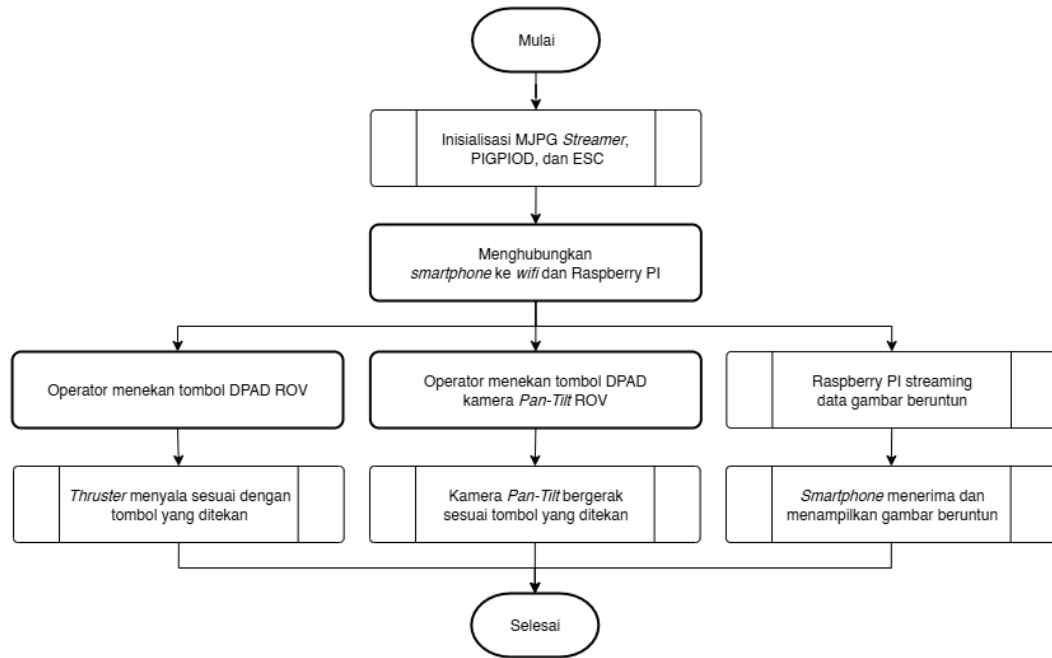


Gambar 3.3 Arsitektur Diagram ROV

Di sisi ROV, Raspberry Pi terhubung ke router TP-Link WR840N menggunakan koneksi Ethernet. Raspberry Pi berperan sebagai pusat pemrosesan perintah yang diterima dari aplikasi Unity di smartphone melalui jaringan lokal yang difasilitasi oleh router. Dengan arsitektur ini, komunikasi antara smartphone dan ROV dapat dilakukan secara *real-time* melalui jaringan Wi-Fi, dengan router TP-Link WR840N sebagai penghubung utama.

3.2.3 Alur Kerja Sistem

Alur kerja sistem merupakan sebuah proses sistem ROV dalam melakukan fungsi utamanya. Didalamnya terdapat entitas-entitas *input* dan proses mekanisme sistem ROV. Alur kerja Sistem ROV sesuai dengan Gambar 3.4 yang tertera dibawah ini.

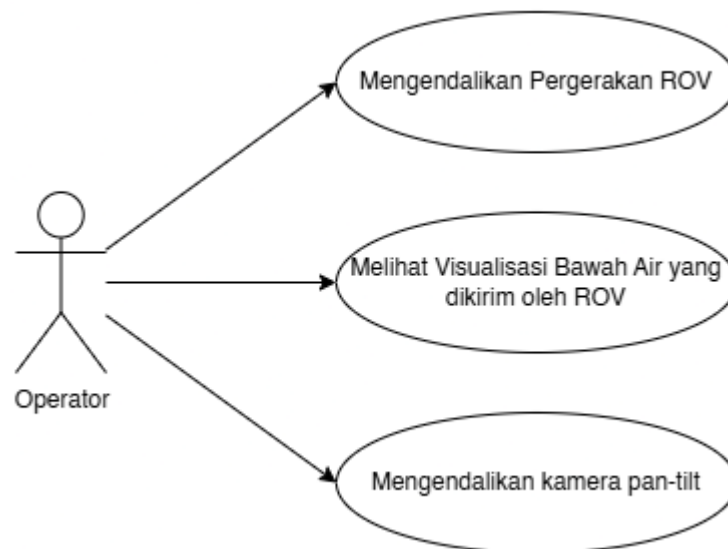


Gambar 3.4 Alur Kerja Sistem ROV

Sistem ROV dimulai dengan menyalakan seluruh komponen yang terhubung, yaitu: router dan Raspberry PI. Setelah seluruh komponen ROV menyala, langkah berikutnya adalah menghubungkan smartphone ke jaringan Wi-Fi router yang sama dengan Raspberry PI. Pada aplikasi *smartphone*, terdapat antarmuka untuk menghubungkan *smartphone* dengan Raspberry PI yang telah menyala dan menjalankan MQTT Broker. Operator dapat menekan tombol yang ada di bagian bawah layar untuk menggerakkan *thruster* dan menekan tombol di bagian atas layar untuk menggerakkan kamera *Pan-Tilt*. Selain itu, pada layar yang sama, operator dapat melihat tampilan gambar beruntun yang ditangkap oleh kamera *webcam*.

3.2.4 Use Case Diagram Sistem

Merujuk pada Gambar 3.5, operator memiliki tiga peran utama. Pertama, operator mengendalikan pergerakan *thruster* sesuai dengan *input* pada aplikasi. Kedua, operator menerima dan melihat visual yang dikirimkan oleh ROV, berupa gambar atau video yang memberikan gambaran langsung ke aplikasi berbasis *mobile*. Ketiga, operator dapat mengendalikan kamera *Pan-Tilt* dengan menggunakan *virtual d'pad* pada antarmuka aplikasi, untuk mengubah sudut pandang kamera guna memfokuskan pada objek atau area tertentu. Ketiga peran ini bekerja bersama untuk memungkinkan operator mengoperasikan dan memantau ROV secara efektif.



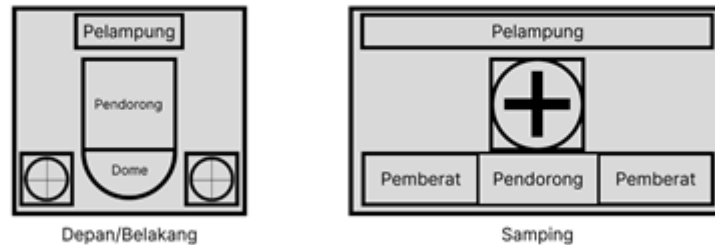
Gambar 3.5 Use Case Diagram Sistem ROV

3.2.5 Desain Rangka ROV

Sebuah ROV menerapkan hukum Archimedes netral dimana sebuah benda akan melayang ketika gaya gravitasi benda setara dengan gaya apung yang dihasilkan. Dengan konsep tersebut, sebuah ROV diharapkan bisa melayang di dalam air dan bermanuver dengan lancar. Titik apung dan titik gravitasi pada ROV menjadi hal yang krusial karena berpengaruh terhadap keseimbangan dan kontrol ROV. ROV yang baik mempunyai desain *frame* dengan titik gravitasi dan titik

apung yang seimbang sehingga tidak akan menyebabkan guncangan pada ROV yang membuat ROV tidak terkendali.

Desain rangka yang direncanakan dalam penelitian ini dapat dilihat pada gambar 3.6 berikut ini.

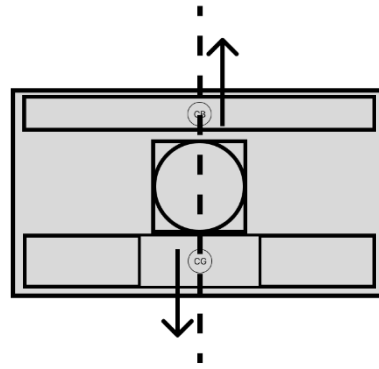


Gambar 3.6 Rancangan Desain Rangka ROV

Referensi dari desain tersebut diambil dari penelitian oleh Hakim (2014). Pada gambar tersebut, dua buah motor pendorong diletakan di bagian bawah ranka ROV agar mendapatkan titik gravitasi yang berada di tengah rangka ROV. Selain itu, agar rangka tetap seimbang ketika didorong oleh motor pendorong, pemberat diberikan sebagai beban di bagian dasar ROV.

Bagian atap rangka ROV dipasangkan sebuah pelampung untuk mempertahankan ROV agar tidak tenggelam ketika menggunakan besi sebagai bahan dasar pembuatan rangka ROV. Pelampung berguna untuk menstabilkan berat ROV agar nantinya ROV dapat melayang di dalam air. Pelampung dipasang secara vertikal pada atas-tengah rangka agar mendapatkan titik apung yang seimbang dengan titik gravitasi sistem

Gambar 3.7 menjabarkan hasil titik gravitasi dan titik apung yang menghasilkan kesetimbangan hidrostatik pada ROV.



Gambar 3.7 Keseimbangan Hirostatis

Pada gambar tersebut, titik apung (*center of buoyancy*) akan dimiliki oleh bagian tengah dari pelampung yang dipasang vertikal pada bagian atas sistem ROV sedangkan titik gravitasinya (*center of gravity*) berada pada bagian dasar sistem sehingga menghasilkan keseimbangan hidrostatik yang stabil.

3.2.6 Desain Kamera *Pan-Tilt*

Kamera *Pan-Tilt* terdiri dari dua jenis *servomotor* yang menggerakkan kamera secara vertikal dan horizontal dan sebuah kamera webcam. Kamera akan dipasang pada *bracket Pan-Tilt* dengan desain standar yang telah disesuaikan. Untuk melindungi kamera dan perangkat elektronik lainnya, kamera dan *bracket* akan dibungkus dengan sebuah wadah yang dirancang khusus yang diharapkan mampu menahan tekanan air sekaligus memberikan visual yang baik. *Bracket* ini merupakan produk yang mudah ditemukan di pasar Indonesia, sehingga mempermudah proses pengadaan alat. Gambar 3.8 adalah tampilan desain braket yang akan digunakan.



Gambar 3.8 *Bracket Servomotor Pan-Tilt*

Kamera yang digunakan adalah webcam XIAOVV dengan spesifikasi resolusi kamera 2 MP 30 FPS seperti yang terlihat pada Gambar 3.9. Kamera XIAOVV adalah kamera webcam dengan dimensi panjang 100, lebar 25 mm, tinggi 50 mm dan mempunyai berat sebesar 105g. Kamera ini akan dipasangkan ke *bracket servomotor Pan-Tilt* agar bisa digerakan dengan leluasa oleh operator. Webcam XIAOVV menggunakan kabel USB sehingga tidak perlu melakukan instalasi dan konfigurasi tambahan untuk menggunakannya.

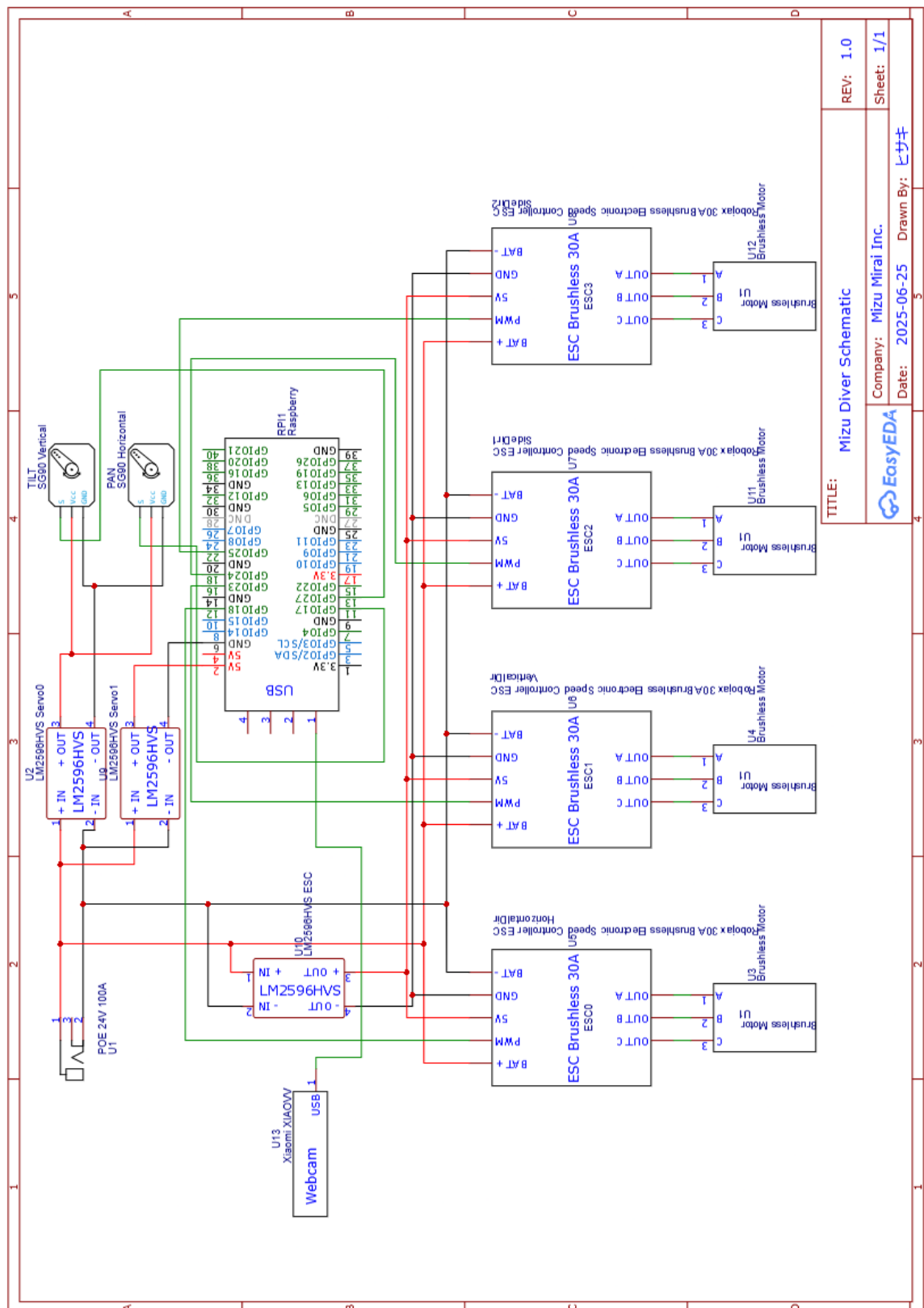


Gambar 3.9 Kamera Xiaomi XIAOVV

Kamera tersebut mempunyai tugas untuk menangkap gambar secara terus-menerus untuk diteruskan ke *smartphone*. Namun, resolusi yang dikirimkan akan dikirimkan diatur menjadi 640x480 dengan *frame rate* 24 FPS untuk menghemat *bandwidth* dan meminimalisir lag.

3.2.7 Perancangan *Wiring* ROV

Perancangan rangkaian elektronika atau *wiring* sistem ROV memuat keseluruhan daya listrik yang mengalir pada sistem ROV. Rangkaian tersebut terbagi menjadi dua jenis rangkaian yaitu: rangkaian GCS, dan rangkaian sistem ROV itu sendiri. Rangkaian *wiring* ROV terlihat pada Gambar 3.10 di bawah ini.



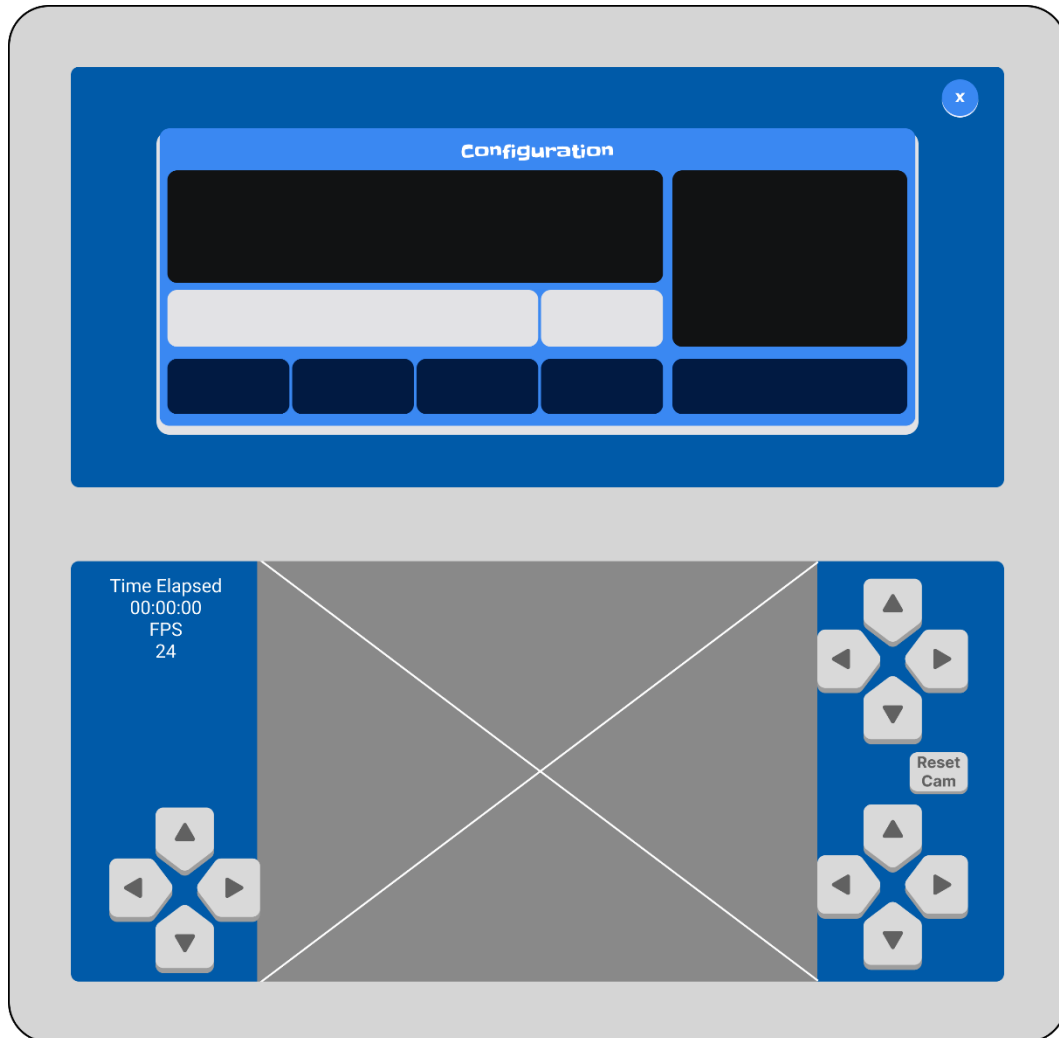
Gambar 3.10 Skema Wiring Diagram Sistem ROV

Rangkaian yang dibuat memuat modul-modul yang digunakan pada perangkat ROV. Pada GCS, sumber listrik AC terhubung dengan PSU yang mengubah listrik AC tersebut menjadi listrik DC. PSU terhubung dengan DC *jack male* dan *buck converter* yang berfungsi untuk menyalakan router. Sumber daya disalurkan dari kabel PoE ke tiga buah modul DC *Buck Converter* yang terhubung dengan Raspberry PI, *servomotor*, dan *brushless* DC motor. Raspberry PI sebagai otak dari ROV ini memakai 6 buah pin untuk mengatur sinyal PWM yang dikirimkan ke servo dan motor dan juga memakai 2 buah pin untuk menenagai Raspberry PI itu sendiri. Motor yang digunakan adalah motor *brushless* DC sehingga memerlukan komponen tambahan untuk mengatur PWM dari motor tersebut yaitu *Electronic Speed Controller* (ESC). ESC mendapatkan daya dari modul LM2597HVS untuk mengontrol PWM pada motor DC sekaligus meneruskan daya secara langsung dari PoE ke motor DC.

3.2.8 Desain Aplikasi berbasis *Mobile*

Aplikasi *mobile* untuk menggerakkan ROV dibuat dengan menggunakan *software* Unity dengan bahasa pemrograman C#. Aplikasi yang diberi nama “Mizu Diver Controller” ini berfungsi untuk mengirim *input* operator dari *smartphone* untuk diteruskan ke Raspberry Pi dengan menggunakan protokol komunikasi MQTT. Aplikasi ini juga mempunyai kemampuan untuk menerima data streaming yang dikirim oleh Raspberry Pi.

Tujuan dibuatnya aplikasi Mizu Diver Controller yang utama adalah mengendalikan ROV dan menampilkan hasil *streaming* pada layar *smartphone*. Oleh karena itu, pertama-tama *mockup* dari aplikasi Mizu Driver Controller dibuat menggunakan *software* desain berbasis web bernama Figma seperti yang terlihat pada Gambar 3.11.



Gambar 3.11 Desain *Mockup* Aplikasi “Mizu Diver Controller” di Figma

Mockup aplikasi ini terdiri dari dua tampilan utama yaitu: menu konfigurasi dan tampilan kontrol utama dari ROV. Desain dari menu konfigurasi dirancang seminimal mungkin agar operator dapat dengan mudah melakukan konfigurasi pertama pada aplikasi. Di samping itu, desain tampilan menu dibuat sesimpel mungkin dan hanya berfokus pada tombol-tombol pergerakan utama.

Dengan menggunakan Figma, proses pembuatan *user interface* dari aplikasi yang dibuat menjadi lebih mudah. Bahkan dengan menggunakan Figma, proses pembuatan aset-aset gambar dan UI yang digunakan dalam aplikasi dapat dibuat dengan lebih mudah dan efisien karena Figma dapat mengekspor seluruh desain yang dibuat dalam bentuk PNG. Gambar-gambar yang telah dibuat diimpor seluruhnya diimpor ke Unity sebagai aset.

3.3 Pengembangan

Setelah proses desain selesai dilakukan, penelitian memasuki tahap pengembangan sistem yang bertujuan untuk membangun dan merakit komponen-komponen sehingga menjadi kesatuan sistem ROV. Sub-bab ini memuat perancangan alat, bahan, serta prosedur yang akan digunakan dalam proses pembuatan seluruh unit ROV. Berikut ini adalah rancangan langkah-langkah yang akan dilaksanakan pada saat proses pembuatan ROV.

3.3.1 Pembuatan Rangka ROV

Pengembangan ROV dimulai dengan menentukan dimensi ruang yang digunakan sebagai dasar dalam membuat rangka ROV. Proses pertama dalam membuat rangka adalah dengan memotong aluminium siku dan menghubungkannya dengan menggunakan paku rivet agar membuat bentuk kubus.

3.3.2 Pemasangan *Brushless Motor DC*

Jika rangka ROV telah dibuat, langkah selanjutnya adalah memasang *brushless motor DC* ke rangka ROV. Terdapat dua jenis penempatan baut pada motor DC yang digunakan yaitu: model 4 baut dan model 3 baut. Motor DC dengan model 4 baut ukuran M5 dipasangkan pada rangka ROV secara vertikal untuk naik dan turun dan horizontal untuk pergerakan maju dan mundur jenis M5 sedangkan motor DC dengan model 3 baut ukuran M3 dipasangkan dengan posisi horizontal kiri kanan untuk pergerakan menyamping dan memutar.

3.3.3 Perakitan Kamera *Pan-Tilt*

Alat dan bahan yang dibutuhkan pada proses perakitan kamera *Pan-Tilt* diantaranya: kamera webcam, kit *Pan-Tilt* siap pasang, dua buah *servomotor SG90*, baut untuk mengencangkan perangkat, dan beberapa kabel *zip ties* untuk menahan webcam pada tempatnya. Proses perakitan kamera *Pan-Tilt* bertujuan untuk mengintegrasikan webcam ke mekanisme *Pan-Tilt* dan juga memastikan agar *Pan-Tilt* kokoh ketika digerakan oleh *servomotor*.

3.3.4 Pembuatan aplikasi *mobile* dan *Scripting*

Penelitian ini akan menggunakan bahasa pemrograman C# dan Python untuk *scripting* karena menggunakan dua jenis sistem yang berbeda yaitu aplikasi

mobile dan *smartphone*. Aplikasi *mobile* pada penelitian ini dibuat dengan menggunakan *software* Unity dengan beberapa *library* pembantu yang berguna untuk berjalannya sistem ROV ketika ROV digerakkan.

Program Python akan digunakan pada Raspberry Pi untuk meng-*handle* MQTT pada Raspberry Pi, menjalankan kamera, menjalankan konfigurasi pin pada Raspberry Pi, mengatur posisi *servomotor Pan-Tilt*, dan mengatur kendali motor DC. Program Python dibuat modular dengan tergantung dengan fungsi tugas yang diberikan pada setiap program. Seluruh fungsi diatur dari *script* “main.py” yang isinya seperti yang ada pada lampiran 9. Lampiran 10 merupakan program untuk memeriksa *library* pigpiod dalam kondisi *running* saat program dimulai. Pigpiod merupakan *library* untuk mengatur pin pada Raspberry Pi. Apabila pigpiod telah berhasil dijalankan program *mjpeg_streamer.py* pada lampiran 11 akan memulai *streaming* MJPG sehingga *smartphone* dapat menampilkan visual kamera webcam. Jika kamera telah berhasil dijalankan, program *mqtt_client_handler.py* pada lampiran 12 akan mulai membuka protokol koneksi MQTT untuk menerima data dalam bentuk JSON. Program *mqtt_client_handler.py* juga memiliki fungsi dari *servo_controller.py* dan *motor_controller.py*.

3.3.5 Integrasi Sistem

Tahap akhir dari proses pengembangan sistem ROV adalah dengan mengintegrasikan keseluruhan unit sistem yang telah dibuat. Komponen-komponen elektronik dan daya serta kamera *Pan-Tilt* dipasangkan pada rangka ROV dan juga dihubungkan dengan motor DC yang telah dipasang pada rangka ROV sebelumnya. Sistem *powering* berasal dari PoE yang menghantarkan daya sekaligus jaringan dari *ground control system* (GCS) ke ROV.

3.4 Evaluasi

Pengujian sistem merupakan tahap evaluasi yang dilakukan setelah menggabungkan seluruh sistem ROV yang telah ada untuk dirakit menjadi sebuah ROV komplit siap pakai. Pengujian ini melibatkan penyusunan, perakitan, dan pengujian seluruh rangka dan komponen elektronik hingga menjadi satu-kesatuan. Pengujian unit pada sistem ROV dilakukan untuk memastikan bahwa setiap komponen dalam sistem berfungsi dengan baik secara individu sebelum

implementasi penuh. Pengujian unit dilakukan secara bertahap untuk memastikan setiap komponen dalam sistem ROV dapat bekerja dengan baik sebelum dilakukan pengujian integrasi. Hasil dari pengujian ini memberikan data yang menjadi dasar untuk langkah evaluasi dan pengembangan lebih lanjut.

Pengujian *black box* digunakan sebagai metode pengujian dalam penelitian ini. Pengujian *black box* adalah pengujian fungsionalitas dari suatu sistem tanpa menilik langsung ke dalam program atau fungsi teknis lainnya. Pengujian ini dirasa cocok karena merupakan simplifikasi dari pengujian *hardware* dan *software*.

3.4.1 Pengujian Unit

Pengujian unit pada sistem ROV dilakukan untuk memastikan bahwa setiap komponen dalam sistem berfungsi dengan baik secara individu sebelum integrasi dan implementasi penuh. Pengujian unit dilakukan secara bertahap untuk memastikan setiap komponen dalam sistem ROV dapat bekerja dengan baik sebelum dilakukan pengujian integrasi. Pengujian *black box* terhadap unit-unit yang diuji antara lain: pengujian *black box* terhadap aplikasi berbasis *mobile*, pengujian *black box* terhadap gerakan BLDC motor, pengujian *black box streaming* kamera, dan pengujian *black box* gerakan *Pan-Tilt*

1. Evaluasi Kinerja Pengiriman Data via Aplikasi

Aplikasi berbasis *mobile* yang dibuat menggunakan Unity sebagai *software* utama. Pengujian pada aplikasi ini bertujuan untuk mengamati kesesuaian data yang dikirim dari *smartphone* ke Raspberry Pi dengan skenario pengujian serta hasil yang diharapkan seperti yang tertera pada Tabel 3.1.

Tabel 3.1 Tabel Skenario Pengujian Kinerja Pengiriman Data

Skenario Pengujian	Ekspektasi Hasil
D-Pad 1 Tombol Kiri Ditekan	Data rotasi ke kiri dikirimkan
D-Pad 1 Tombol Kanan Ditekan	Data rotasi ke kanan dikirimkan
D-Pad 1 Tombol Atas Ditekan	Data bergerak naik dikirimkan

Skenario Pengujian	Ekspektasi Hasil
D-Pad 1 Tombol Bawah Ditekan	Data bergerak turun dikirimkan
D-Pad 2 Tombol Kiri Ditekan	Data bergerak menyamping ke kiri dikirimkan
D-Pad 2 Tombol Kanan Ditekan	Data bergerak menyamping ke kanan dikirimkan
D-Pad 2 Tombol Atas Ditekan	Data bergerak maju dikirimkan
D-Pad 2 Tombol Bawah Ditekan	Data bergerak mundur dikirimkan
D-Pad 3 Tombol Kiri Ditekan	Data kamera bergerak ke kiri dikirimkan
D-Pad 3 Tombol Kanan Ditekan	Data kamera bergerak ke kanan dikirimkan
D-Pad 3 Tombol Atas Ditekan	Data kamera bergerak ke atas dikirimkan
D-Pad 3 Tombol Bawah Ditekan	Data kamera bergerak ke bawah dikirimkan
Tombol Reset Kamera Ditekan	Data reset posisi kamera dikirimkan

Tabel tersebut menyajikan skenario pengujian pengiriman data via aplikasi berbasis *mobile*. Setiap skenario pengujian menggambarkan aksi spesifik yang dilakukan oleh pengguna, seperti menekan tombol tertentu pada D-Pad 1, D-Pad 2, D-Pad 3, maupun tombol reset kamera.

Kolom pertama berisi Skenario Pengujian, yang menjelaskan aksi atau interaksi pengguna pada antarmuka kontrol. Kolom kedua berisi Ekspektasi Hasil, yaitu output atau respons sistem yang diharapkan muncul sebagai hasil dari aksi tersebut. Ekspektasi hasil difokuskan pada pengiriman data kontrol yang sesuai untuk menggerakkan bagian-bagian tertentu dari ROV seperti arah gerak utama (maju, mundur, menyamping), arah rotasi (yaw), serta kontrol *Pan-Tilt* kamera.

Selain pengujian pengiriman data MQTT, pengujian untuk mengukur FPS dari *stream* yang diterima juga diukur untuk mengetahui kinerja kualitas *streaming* pada *smartphone*. Pengukuran FPS dilakukan pada sisi aplikasi *smartphone* dengan menghitung jumlah *frame* yang menggantikan *texture* dari gambar beruntun selama 10 detik. Dalam memudahkan proses pelacakan waktu, aplikasi akan dilengkapi

dengan fitur *timestamp* sehingga pengukuran dapat lebih terperinci. Layar *smartphone* akan didokumentasikan melalui *screenshot* dengan interval 1 detik. Tabel 3.2 berikut menyajikan skenario pengujian berdasarkan interval waktu yang akan dilakukan pada pengukuran FPS.

Tabel 3.2 Tabel Skenario Pengujian Kinerja *Streaming*

Interval (s)	Ekspektasi FPS
1	>24
2	>24
3	>24
4	>24
5	>24
6	>24
7	>24
8	>24
9	>24
10	>24

Kolom pertama merepresentasikan titik waktu di mana data FPS direkam, dengan pencatatan yang dilakukan setiap satu detik selama durasi pengujian. Sementara itu, “Ekspektasi FPS” berfungsi sebagai nilai acuan atau benchmark performa, yang dalam penelitian ini ditetapkan pada 24 FPS. Nilai ini dipilih karena merupakan *frame rate* standar yang mampu menyajikan pergerakan visual yang halus sekaligus efisien dalam penggunaan *bandwidth* jaringan. Dengan membandingkan nilai FPS aktual yang tercatat pada setiap interval terhadap *benchmark* ini, kinerja dan konsistensi video *stream* dari ROV dapat dianalisis secara objektif.

2. Pengujian Kinerja Motor DC

Motor DC yang akan dipasang pada sistem ROV diuji coba terlebih dahulu untuk meninjau kinerja yang terlihat ketika motor DC berputar dan memberikan gaya dorong. Proses pengujian motor DC berlangsung dengan menggunakan mikrokontroler Arduino yang memberikan sinyal PWM ke ESC secara langsung untuk menggerakkan motor DC. Pengujian *black box* yang digunakan untuk menguji kinerja dari motor DC seperti yang terlampir pada Tabel 3.3 di bawah ini.

Tabel 3.3 Tabel Skenario Pengujian Motor DC

Skenario Pengujian	Ekspektasi Hasil
Motor DC diberi sinyal PWM 1500	Perputaran dan kecepatan motor netral
Motor DC diberi sinyal PWM 1300	Motor berputar searah jarum jam dengan kecepatan pelan
Motor DC diberi sinyal PWM 1000	Motor berputar searah jarum jam dengan kecepatan cepat
Motor DC diberi sinyal PWM 1700	Motor berputar berlawanan arah jarum jam dengan kecepatan pelan
Motor DC diberi sinyal PWM 2000	Motor berputar berlawanan arah jarum jam dengan kecepatan cepat

Pada skenario pengujian tersebut, secara berturut-turut skenario pengujian adalah dengan memberikan masukan sinyal PWM ke ESC sebanyak 1000 hingga 2000 μ s dan melihat kinerja yang diberikan motor DC ketika diberi *input-input* tersebut. Hasil yang diharapkan ketika *input-input* pada skenario pengujian diberikan adalah motor DC dapat berada pada posisi netral, berputar searah jarum jam (CW) dan berputar berlawanan arah jarum jam (CCW) dengan kecepatan putaran yang diatur oleh nilai PWM yang dimasukkan.

3.4.2 Pengujian Sistem

Pengujian integrasi adalah tahap penting dalam pengembangan sistem, termasuk pada proyek ROV (*Remotely Operated Vehicle*). Setelah pengujian unit selesai dilakukan pada setiap komponen, tahap pengujian integrasi bertujuan untuk memastikan bahwa semua komponen yang telah diuji secara individu dapat bekerja dengan baik ketika digabungkan dalam satu sistem utuh. Pengujian ini dilakukan untuk mengevaluasi interaksi antara berbagai subsistem yang ada dalam ROV. Pengujian *black box* sistem ROV berfokus pada pengaruh

1. Pengujian Kinerja Visual ROV

Evaluasi kinerja visual ROV bertujuan untuk meninjau pengujian yang telah dilaksanakan berdasarkan pengujian *black box* dengan teknik uji fungsionalitas sistem. Skenario pengujian melibatkan pengukuran responsivitas dan kecepatan

mekanisme kamera *Pan-Tilt* secara kuantitatif dengan menggunakan perangkat *smartphone* dalam memberikan *input* ke Raspberry Pi yang akan diolah menjadi gerakan *Pan-Tilt*. Pengujian dilakukan dengan memberikan input perintah gerak hingga kamera mencapai batas maksimum pada setiap sumbu (kiri, kanan, atas, bawah). Untuk setiap skenario, akan diukur waktu yang dibutuhkan dan perubahan sudut total yang ditempuh. Dari data tersebut, akan dihitung kecepatan sudut (rad/s) untuk menganalisis performa sistem. Pengujian untuk setiap skenario akan diulang sebanyak 20 kali untuk mendapatkan data yang valid dan reliabel, yang kemudian akan dihitung nilai rata-ratanya. Hasil yang diharapkan ketika pengujian sedang berlangsung adalah gerakan *Pan-Tilt* dan visual yang ditangkap kamera dapat bergerak sesuai dengan *input* yang diberikan dengan responsif. Berikut adalah Tabel 3.4 yang merupakan tabel skenario *black box* yang digunakan dalam pengujian.

Tabel 3.4 Tabel Skenario Pengujian Kinerja Visual ROV

Skenario Pengujian	Hasil yang diharapkan
D-Pad 3 Tombol Kiri Ditekan	Pan-Tilt melakukan rotasi ke kiri dan tampilan bergeser ke kiri
D-Pad 3 Tombol Kanan Ditekan	Pan-Tilt melakukan rotasi ke kanan dan tampilan bergeser ke kanan
D-Pad 3 Tombol Naik Ditekan	Pan-Tilt melakukan rotasi ke atas dan tampilan bergeser ke atas
D-Pad 3 Tombol Turun Ditekan	Kamera bergerak turun dan gambar bergerak turun

Tabel 3.3 tersebut merinci setiap skenario yang dirancang untuk pengujian fungsionalitas sistem visual ROV dengan metode *black box*. Kolom pertama, "Skenario Pengujian", menjabarkan setiap aksi spesifik yang dilakukan oleh operator. Aksi ini berupa interaksi penekanan tombol-tombol yang relevan pada D-Pad 3 di antarmuka aplikasi, yang berfungsi sebagai *input* untuk memicu mekanisme kamera *Pan-Tilt*. Setiap skenario mensimulasikan perintah gerak pada kedua sumbu, yaitu pergerakan horizontal (*pan*) dan vertikal (*tilt*), untuk

memastikan cakupan pengujian yang menyeluruh. Sementara itu, kolom kedua, "Hasil yang Diharapkan", mendefinisikan kriteria keberhasilan atau ekspektasi keluaran untuk setiap skenario pengujian yang dilakukan. Ekspektasi ini mencakup dua aspek utama yang harus terpenuhi: respons fisik dari pergerakan mekanisme kamera sesuai dengan arah input yang diberikan, dan respons visual yang sinkron pada tampilan gambar di layar *smartphone*. Kolom ini menjadi acuan utama untuk memvalidasi bahwa setiap perintah dari operator dapat diterjemahkan menjadi aktuasi mekanik dan umpan balik visual yang sesuai dengan rancangan sistem.

2. Pengujian Koordinasi *Thruster*

Pengujian koordinasi *thruster* merupakan tahap evaluasi krusial yang bertujuan untuk memverifikasi bahwa setiap sinyal input dari aplikasi operator berhasil diterjemahkan menjadi kombinasi aktivasi motor DC yang benar. Metode pengujian yang digunakan adalah observasi fungsional, di mana setiap skenario pergerakan ROV (maju, mundur, naik, turun, lateral, dan rotasi) akan diaktifkan secara berkelanjutan selama 10 detik.

Selama interval pengujian tersebut, koordinasi dan arah dorong dari setiap motor DC akan diamati dan dicatat pada setiap detiknya untuk memastikan konsistensi dan kesesuaian respons terhadap perintah yang diberikan. Hal ini penting untuk memvalidasi bahwa logika pergerakan yang dirancang telah terimplementasi dengan benar pada sistem pendorong. Tabel 3.5 di bawah ini menyajikan rincian skenario pengujian yang akan dilakukan beserta ekspektasi hasil dari koordinasi *thruster* untuk setiap perintah gerak.

Tabel 3.5 Tabel Skenario Pengujian Koordinasi Motor DC

Skenario Pengujian	Ekspektasi Hasil
Tombol Maju Ditekan	Motor DC horizontal (maju/mundur) mendorong ke belakang
Tombol Mundur Ditekan	Motor DC horizontal (maju/mundur) mendorong ke depan
Tombol Kiri (Lateral) Ditekan	Motor DC horizontal (kiri/kanan) keduanya mendorong ke kanan

Skenario Pengujian	Ekspektasi Hasil
Tombol Kanan (Lateral) Ditekan	Motor DC horizontal (kiri/kanan) keduanya mendorong ke kiri
Tombol Kiri (Rotasi) Ditekan	Motor DC horizontal (kiri/kanan) bagian belakang mendorong ke kanan
Tombol Kanan (Rotasi) Ditekan	Motor DC horizontal (kiri/kanan) bagian belakang mendorong ke kiri
Tombol Naik Ditekan	Motor DC vertikal (naik/turun) bagian belakang mendorong ke bawah
Tombol Turun Ditekan	Motor DC vertikal (naik/turun) bagian belakang mendorong ke atas

Pada kolom pertama, skenario yang diujikan adalah dengan menekan tombol *input* pada D-Pad 1 dan D-Pad 2 yang berfungsi untuk mengatur pergerakan koordinasi motor *thruster*. Ekspektasi hasil pengujian ini terdapat pada kolom kedua yaitu motor DC diharapkan dapat bergerak sesuai dengan tombol *input* pada masing-masing D-Pad.