

**PERANCANGAN SISTEM KENDALI MEKANISME
PEMBUKA PINTU PADA ROBOT ASISTEN MEDIS
BERBASIS ANDROID**

TUGAS AKHIR

**Diajukan Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Teknik Mesin**



Disusun Oleh :

Nama : Rahmat Irvan Rinaldy

No. Mahasiswa : 17525060

NIRM : 2017023615

**JURUSAN TEKNIK MESIN
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA**

2021

PERNYATAAN KEASLIAN

Saya yang bertanda tangan di bawah ini,

Nama : Rahmat Irvan Rinaldy

NIM : 17525060

Program Studi : S1, Teknik Mesin

Institusi : Universitas Islam Indonesia

Judul Laporan : Perancangan Sistem Kendali Mekanisme Pembuka Pintu pada *Robot* Asisten Medis Berbasis *Android*

Dengan ini saya menyatakan, semua yang saya tulis pada Tugas Akhir ini adalah hasil karya saya sendiri kecuali kutipan atau ringkasan yang saya ambil sebagai referensi dan telah saya cantumkan sumber-sumbernya. Apabila di kemudian hari pengakuan saya terbukti tidak benar, maka saya bersedia mengikuti hukuman ataupun sanksi yang diberikan sesuai hukum yang berlaku.

Yogyakarta, 19 Agustus 2021



Rahmat Irvan Rinaldy
17525060

LEMBAR PENGESAHAN DOSEN PEMBIMBING

**PERANCANGAN SISTEM KENDALI MEKANISME
PEMBUKA PINTU PADA ROBOT ASISTEN MEDIS
BERBASIS ANDROID**

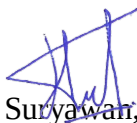
TUGAS AKHIR

Disusun Oleh :

Nama : Rahmat Irvan Rinaldy
No. Mahasiswa : 17525060
NIRM : 2017023615

Yogyakarta, 9 Agustus 2021

Pembimbing,



Donny Suryawan, S.T., M.Eng

LEMBAR PENGESAHAN DOSEN PENGUJI

PERANCANGAN SISTEM KENDALI MEKANISME PEMBUKA PINTU PADA ROBOT ASISTEN MEDIS BERBASIS ANDROID

TUGAS AKHIR

Disusun Oleh :

Nama : Rahmat Irvan Rinaldy
No. Mahasiswa : 17525060
NIRM : 2017023615

Tim Penguji

Donny Suryawan, S.T., M.Eng.

Ketua



Tanggal : 17 Agustus 2021

Purtojo, S.T., M.Sc.

Anggota I



Tanggal : 14 Agustus 2021

Muhammad Ridlwan, S.T., M.T.

Anggota II



Tanggal : 9 Agustus 2021

Mengetahui

Ketua Jurusan Teknik Mesin



Dr. Eng. Risdiyono, S.T., M.Eng.

HALAMAN PERSEMBAHAN

Allhamdulillah kupanjatkan kepada Allah SWT, atas segala rahmat dan juga kesempatan dalam menyelesaikan laporan Tugas Akhir ini dengan segala kekurangannya. Segala syukur kuucapkan padaMu ya Rabb, karena sudah menghadirkan orang-orang yang berarti di sekeliling saya, yang selalu memberi semangat dan doa sehingga laporan Tugas Akhir ini dapat diselesaikan dengan baik.

Karya tulis sederhana ini kupersembahkan untuk :

Kedua orang tua yang senantiasa memberikan dukungan dan doa hingga Tugas Akhir ini selesai.

Adik-adikku, yang selalu menghibur di tengah suntuknya mengerjakan laporan Tugas Akhir ini.

Dosen pembimbing, yang dengan sabar memberikan masukan, nasihat dan ilmu sehingga pengerjaan laporan Tugas Akhir ini dapat berjalan dengan lancar.

Teman-teman Teknik Mesin UII, yang selalu memberi bantuan, nasihat, doa dukungan dan lainnya hingga selesainya Tugas Akhir ini.

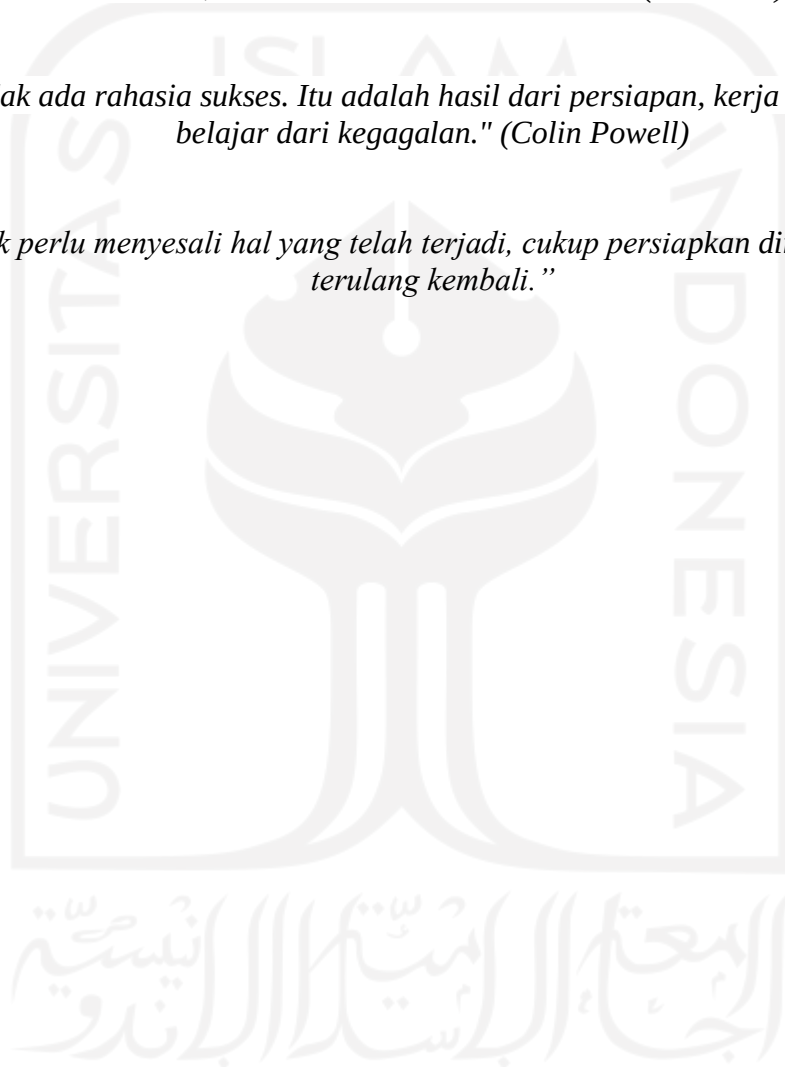
HALAMAN MOTTO

*“Sesungguhnya dibalik kesulitan itu ada kemudahan”
(QS. Al-Insyiroh : 6)*

*"Jika kamu benar-benar ingin melakukan sesuatu, kamu akan menemukan cara.
Jika tidak, kamu akan menemukan alasan." (Jim Rohn)*

*"Tidak ada rahasia sukses. Itu adalah hasil dari persiapan, kerja keras, dan
belajar dari kegagalan." (Colin Powell)*

*“Tak perlu menyesali hal yang telah terjadi, cukup persiapkan diri agar tak
terulang kembali.”*



KATA PENGANTAR ATAU UCAPAN TERIMA KASIH

Dengan menyebut nama Allah SWT. yang maha pengasih lagi maha penyayang. Puji syukur kehadirat Allah SWT. karena berkat rahmat dan karunia –Nya lah Penulis dapat menyelesaikan Laporan Tugas Akhir ini. Laporan Tugas Akhir ini disusun sebagai salah satu syarat untuk menyelesaikan program pendidikan Strata Satu Teknik Mesin di Universitas Islam Indonesia.

Selesainya Laporan Tugas Akhir ini tidak terlepas dari dukungan dan bantuan dari banyak pihak. Penulis berterima kasih kepada semua pihak yang telah ikut berkontribusi dalam penyelesaian Laporan Tugas Akhir ini dan secara khusus pada kesempatan ini penulis menyampaikan ucapan terima kasih kepada yang terhormat:

1. Allah SWT, Tuhan semesta alam yang telah memberikan rahmat dan hidayah-Nya kepada penulis, dan tak lupa kepada Nabi besar Muhammad SAW sebagai nabi akhir zaman dan suri tauladan bagi umat Islam.
2. Bapak dan Ibu serta seluruh keluarga tercinta yang senantiasa selalu memberikan dukungan dan doa dalam menempuh pendidikan.
3. Bapak Donny Suryawan, S.T., M.Eng selaku pembimbing tugas akhir yang telah meluangkan waktunya untuk membimbing penulis.
4. Muhammad Al Qadim dan A Rahman teman seperjuangan kuliah yang sering memberi masukan dan dukungan kepada penulis dalam proses pengerjaan laporan tugas akhir ini.
5. Seluruh mahasiswa Teknik Mesin FTI UII dan semua pihak yang telah berjasa membantu penulis menyelesaikan tugas akhir dan tidak bisa disebutkan satu persatu oleh penulis.

Semoga segala dukungan dan bantuan yang diberikan sehingga penulis dapat menyelesaikan penulisan Laporan Tugas Akhir ini dicatat oleh Allah SWT sebagai amal ibadah. Aamiin.

Penulis menyadari bahwa masih banyak kekurangan dari laporan ini, oleh karena itu penulis mohon maaf untuk kesalahan-kesalahan yang terdapat pada penulisan laporan Tugas Akhir ini, baik dari materi maupun teknik penyajiannya, mengingat kurangnya pengetahuan dan pengalaman penulis, oleh karena itu kritik

dan saran yang sifatnya membangun demi kesempurnaan akan penulis terima dengan senang hati. Akhir kata semoga laporan ini dapat bermanfaat bagi kita semua.

Yogyakarta, 9 Agustus 2021

Penulis



ABSTRAK

Penelitian ini bertujuan untuk merancang sistem kendali mekanisme pembuka pintu pada robot medis sehingga robot dapat dikendalikan oleh operator untuk membuka/menutup pintu dan masuk ke dalam ruangan untuk menjalankan tugas sebagai asisten petugas medis.

Perancangan sistem kendali menggunakan aplikasi pada Android sebagai alat pengendali mekanisme yang mana menggunakan konektivitas Wi-Fi untuk mengirim data perintah menuju ESP32 board. Data yang diterima oleh ESP32 melalui komunikasi via server kemudian akan diolah untuk menggerakkan aktuator pada mekanisme pembuka pintu sehingga mekanisme dapat berfungsi dengan semestinya. Aktuator yang akan digerakkan oleh ESP32 board terdiri dari 2 motor Power Window Universal dan 2 motor Servo MG 996R.

Hasil akhir dari perancangan menunjukkan bawah aplikasi Android yang dirancang mampu untuk mengendalikan robot untuk membuka pintu dari jarak jauh dengan persentase keberhasilan 100% dengan catatan 50% lancar dan 50% kurang lancar, sedangkan untuk menutup pintu persentase keberhasilan 100% dengan catatan 60% lancar dan 40% kurang lancar. Waktu rata-rata yang dibutuhkan robot untuk membuka pintu adalah 71,3 detik dan waktu rata-rata untuk menutup pintu adalah 23,3 detik.

Kata kunci : Robot asisten medis, membuka pintu, ESP32, Android, Wi-Fi

DAFTAR ISI

Halaman Judul	i
Lembar Pengesahan Dosen Pembimbing	ii
Lembar Pengesahan Dosen Penguji	iii
Halaman Persembahan	iv
Halaman Motto	v
Kata Pengantar atau Ucapan Terima Kasih	vi
Abstrak	viii
Daftar Isi	ix
Daftar Tabel	xii
Daftar Gambar	xiii
Bab 1 Pendahuluan	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian atau Perancangan	3
1.5 Manfaat Penelitian atau Perancangan	3
1.6 Sistematika Penulisan	4
Bab 2 Tinjauan Pustaka	5
2.1 Kajian Pustaka	5
2.2 Landasan Teori	8
2.2.1 Sistem Kendali	8
2.2.2 <i>Android</i>	8
2.2.3 <i>Arduino</i>	8
2.2.4 <i>ESP32 Board</i>	8
2.2.5 <i>Wireless communication</i>	9
2.2.6 <i>Bluetooth</i>	9
2.2.7 <i>Wi-Fi</i>	9
2.2.8 <i>Smartphone</i>	9
2.2.9 Robot semi otomatis	9
2.2.10 <i>Module BTS 7960</i>	9

2.2.11	<i>Bluetooth Module HC-05</i>	10
2.2.12	<i>MIT App Inventor</i>	10
Bab 3	Metode Penelitian	11
3.1	Alur Perancangan.....	11
3.2	Peralatan dan Bahan.....	12
3.3	Perancangan	17
3.3.1	Kriteria Perancangan	17
3.3.2	Perancangan Perangkat Keras	19
3.3.3	Perancangan Perangkat Lunak	22
3.3.4	Pemilihan Mikrokontroler	28
Bab 4	Hasil dan Pembahasan	29
4.1	Hasil Perancangan.....	29
4.2	Hasil Pengujian	37
4.2.1	Pengujian membuka pintu	37
4.2.2	Pengujian waktu yang diperlukan robot untuk membuka pintu hingga masuk ke dalam ruangan	40
4.2.3	Pengujian menutup pintu	42
4.2.4	Pengujian waktu yang dibutuhkan robot untuk menutup pintu.....	45
4.2.5	Pengujian dengan operator yang belum terlatih	47
4.3	Analisis dan Pembahasan.....	52
4.3.1	Pengujian dengan operator melihat robot secara langsung	52
4.3.2	Pengujian dengan operator melihat visual kamera robot dari jarak jauh	54
4.3.3	Pengujian dengan operator yang belum terlatih	58
4.3.4	Perbandingan robot asisten medis UII dengan robot asisten medis lainnya	60
Bab 5	Penutup.....	61
5.1	Kesimpulan	61
5.2	Saran atau Penelitian Selanjutnya	62
Daftar Pustaka		63
LAMPIRAN 1		65
LAMPIRAN 2		66

LAMPIRAN 3	69
LAMPIRAN 4	70



DAFTAR TABEL

Tabel 2-1 Ringkasan kajian pustaka.....	7
Tabel 2-2 Spesifikasi Singkat <i>Module BTS 7960</i>	10
Tabel 3-1 Alur gerakan robot ketika membuka pintu	17
Tabel 3-2 Perbandingan <i>Arduino</i> dan <i>ESP32</i>	28
Tabel 4-1 Fungsi tiap tombol pada aplikasi <i>Android</i>	30
Tabel 4-2 Perubahan alur gerakan robot ketika membuka pintu.....	35
Tabel 4-3 Data pengujian membuka pintu secara langsung.....	38
Tabel 4-4 Data pengujian membuka pintu dari jarak jauh	39
Tabel 4-5 Data pengujian waktu yang dibutuhkan robot untuk membuka pintu hingga masuk ke dalam ruangan secara langsung	40
Tabel 4-6 Data pengujian waktu yang dibutuhkan robot untuk membuka pintu hingga masuk ke dalam ruangan dari jarak jauh	41
Tabel 4-7 Data pengujian menutup pintu secara langsung.....	43
Tabel 4-8 Data pengujian menutup pintu dari jarak jauh	44
Tabel 4-9 Data pengujian waktu yang dibutuhkan robot untuk menutup pintu secara langsung.....	45
Tabel 4-10 Data pengujian waktu yang dibutuhkan robot untuk menutup pintu dari jarak jauh	46
Tabel 4-11 Data pengujian robot membuka pintu yang dikendalikan operator A dan operator B	48
Tabel 4-12 Data pengujian waktu yang dibutuhkan robot untuk membuka pintu hingga masuk ke dalam ruangan yang dikendalikan oleh operator A dan B	49
Tabel 4-13 Data pengujian robot menutup pintu yang dikendalikan operator A dan operator B	50
Tabel 4-14 Data pengujian waktu yang dibutuhkan robot untuk menutup pintu yang dikendalikan oleh operator A dan B	51
Tabel 4-15 Perbandingan robot asisten medis yang sudah ada	60

DAFTAR GAMBAR

Gambar 1-1 Grafik Perkembangan Kasus COVID-19 Indonesia	1
Gambar 3-1 Alur Perancangan	11
Gambar 3-2 <i>Arduino</i>	12
Gambar 3-3 <i>ESP32 Board</i>	13
Gambar 3-4 Motor Driver BTS 7960	13
Gambar 3-5 <i>Bluetooth</i> modul HC-05	14
Gambar 3-6 Tampilan awal <i>Arduino IDE</i>	14
Gambar 3-7 Tampilan awal MIT App Inventor	15
Gambar 3-8 <i>Limit switch</i>	15
Gambar 3-9 Motor servo MG 996 R	16
Gambar 3-10 Motor <i>Power Window Universal</i>	16
Gambar 3-11 Desain mekanisme pembuka pintu pada robot asisten medis	19
Gambar 3-12 Ilustrasi rangkaian sistem kendali mekanisme pembuka pintu menggunakan <i>Arduino</i>	21
Gambar 3-13 Ilustrasi rangkaian sistem kendali mekanisme pembuka pintu menggunakan <i>ESP32</i>	21
Gambar 3-14 Skema Alur Kerja Aplikasi Remot pada <i>Android</i> untuk penggunaan <i>Arduino</i>	23
Gambar 3-15 Skema Alur Kerja Aplikasi Remot pada <i>Android</i> untuk penggunaan <i>ESP32</i>	24
Gambar 3-16 Skema Alur Kerja pada <i>Arduino</i>	26
Gambar 3-17 Skema Alur Kerja pada <i>ESP32</i>	27
Gambar 4-1 Tampilan Aplikasi Remot	29
Gambar 4-2 Pembacaan alamat server yang diterima <i>ESP32</i> dari <i>Android</i> menggunakan serial monitor pada <i>Arduino IDE</i>	30
Gambar 4-3 Lengan Robot yang akan dikendalikan	31
Gambar 4-4 Posisi peletakan motor <i>power window universal</i>	31
Gambar 4-5 Posisi peletakan motor servo	32
Gambar 4-6 Posisi peletakan <i>limit switch</i>	32
Gambar 4-7 Rangkaian Sistem Kendali	33

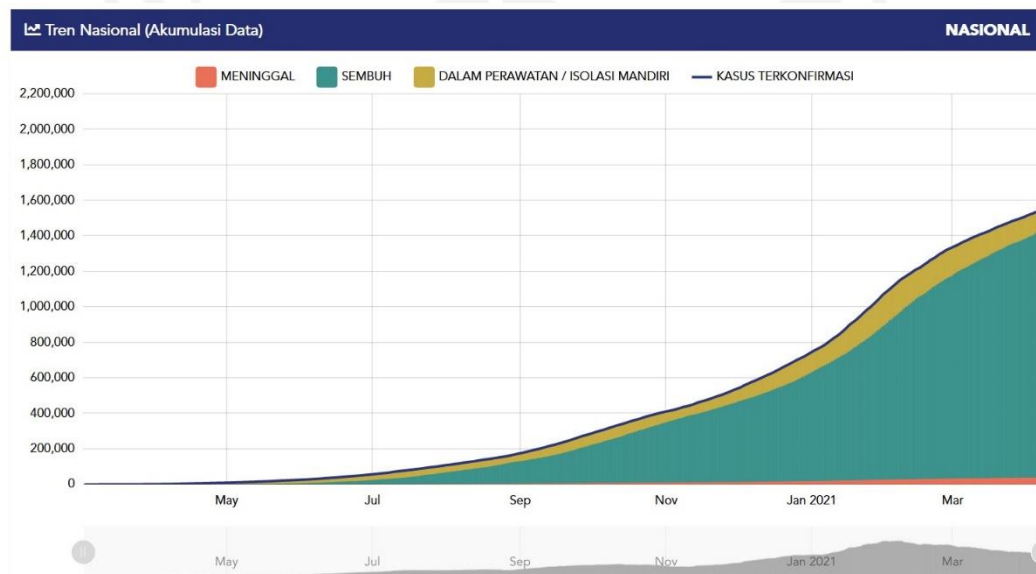
Gambar 4-8 Percobaan mengendalikan lengan robot untuk membuka pintu.....	34
Gambar 4-9 Diagram percobaan robot membuka pintu secara langsung.....	38
Gambar 4-10 Diagram percobaan robot membuka pintu dari jarak jauh.....	39
Gambar 4-11 Grafik waktu yang dibutuhkan robot untuk membuka pintu dan masuk ke dalam ruangan secara langsung.....	41
Gambar 4-12 Grafik waktu yang dibutuhkan robot untuk membuka pintu dan masuk ke dalam ruangan dari jarak jauh	42
Gambar 4-13 Diagram percobaan robot menutup pintu secara langsung	43
Gambar 4-14 Diagram percobaan robot menutup pintu dari jarak jauh.....	44
Gambar 4-15 Grafik waktu yang dibutuhkan robot untuk menutup pintu secara langsung.....	46
Gambar 4-16 Grafik waktu yang dibutuhkan robot untuk menutup pintu dari jarak jauh	47
Gambar 4-17 Persentase keberhasilan robot membuka pintu yang dikendalikan operator A dan operator B	48
Gambar 4-18 Grafik data waktu yang dibutuhkan operator A dan B untuk mengendalikan robot membuka pintu hingga masuk ke dalam ruangan	50
Gambar 4-19 Persentase keberhasilan robot menutup pintu yang dikendalikan operator A dan operator B	51
Gambar 4-20 Grafik data waktu yang dibutuhkan operator A dan B untuk mengendalikan robot menutup pintu	52
Gambar 4-21 Masalah roda robot terangkat ketika proses membuka pintu	53
Gambar 4-22 Posisi awal kamera	55
Gambar 4-23 Posisi kamera setelah dilakukan modifikasi.....	55
Gambar 4-24 Mekanisme pembuka pintu robot tersangkut pada kusen pintu	57

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Pada awal tahun 2020 hingga sekarang, dunia mengalami pandemi wabah *COVID-19* tidak terkecuali Indonesia. Virus ini dikatakan berbahaya karena dapat dengan cepat menyebar dan menjangkit banyak orang melalui kontak fisik antara orang sehat dengan orang yang telah terjangkit virus. Hingga laporan ini ditulis, tercatat sudah lebih dari 1,5 juta kasus positif dengan lebih dari 40 ribu orang meninggal di Indonesia [1].



Gambar 1-1 Grafik Perkembangan Kasus COVID-19 Indonesia

Dari grafik pada Gambar 1-1 terlihat bahwa jumlah kasus positif yang terkonfirmasi masih terus bertambah tiap harinya, untuk menahan penyebaran virus yang sangat cepat, maka pasien yang terjangkit virus *COVID-19* dan bergejala berat diharuskan melakukan karantina diri di ruang isolasi rumah sakit. Namun, jumlah pasien yang terlampaui banyak, membuat tenaga medis kewalahan dalam menangani pasien virus *COVID-19*. Selain mengurus tenaga, petugas medis juga terancam terpapar virus *COVID-19* karena seringnya berinteraksi dengan pasien yang terjangkit virus, tercatat sudah ada sebanyak 237 dokter di Indonesia yang meninggal per 2 Januari 2021[2].

Untuk membantu tenaga medis dalam merawat pasien dan juga meminimalisir kontak antara pasien dengan petugas medis agar penyebaran virus dapat dikendalikan serta meminimalisir korban jiwa, maka dibuatlah robot asisten medis yang bertugas untuk mengantarkan barang, makanan dan obat ke pasien yang sedang dirawat di ruang isolasi. Namun robot asisten medis yang sudah ada sekarang belum tersedia fitur atau mekanisme untuk dapat membuka pintu, sehingga pasien yang harus mengambil keluar atau membukakan pintu. Hal ini menjadi masalah jika pasien yang dirawat dalam kondisi yang tidak memungkinkan untuk mengambil sendiri keluar ruang isolasi. Selain itu juga proses karantina menjadi kurang efektif. Solusi lain untuk masalah ini sebenarnya sudah ada yaitu dengan membuat pintu ruangan isolasi tersebut menjadi otomatis terbuka dan tertutup, namun biaya yang diperlukan sangat besar. Oleh karena itu perlu adanya penelitian lebih lanjut terkait pengembangan mekanisme pembuka pintu pada robot medis yang sudah ada agar dapat meningkatkan mobilitas robot dalam membantu petugas medis dalam melayani pasien karantina COVID-19.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang dijelaskan pada bab 1.1 maka dapat ditentukan rumusan masalah pada perancangan sistem kendali mekanisme pembuka pintu pada robot asisten medis berbasis *Android*, sebagai berikut :

1. Bagaimana membuat aplikasi *Android* untuk menjadi remot pengendali mekanisme pembuka pintu pada robot medis ?
2. Bagaimana merancang sistem kendali untuk mekanisme pembuka pintu ?
3. Bagaimana membuat program pada mikrokontroler untuk menerima data dari *Android* agar mekanisme pembuka pintu pada robot medis dapat menjalankan tugasnya untuk membuka pintu ?

1.3 Batasan Masalah

Pada tugas akhir ini, dilakukan dari perancangan, pembuatan *prototype* hingga melakukan simulasi, namun dalam proses perancangan dan pembuatan dibatasi oleh :

1. Terbatas pada perancangan sistem kendali, tidak membahas desain mekanisme
2. Perancangan sistem kendali terbatas pada sistem operasi *Android*
3. Bahasa pemrograman yang digunakan terbatas pada bahasa pemrograman yang digunakan *software Arduino IDE*
4. Perancangan aplikasi *remot control Android* terbatas pada sistem kendali mekanisme pembuka pintu, tidak termasuk pengendali robot asisten medis
5. Sistem kendali mekanisme terbatas pada pintu yang dibuka dengan cara ditarik
6. Perancangan dilakukan dengan mempertimbangkan biaya yang seminimal mungkin.

1.4 Tujuan Penelitian atau Perancangan

Tujuan dari perancangan yang dilakukan pada tugas akhir ini yaitu :

1. Merancang dan membuat aplikasi *Android* yang mampu mengendalikan mekanisme pembuka pintu pada robot asisten medis
2. Merancang sistem kendali untuk mekanisme pembuka pintu
3. Membuat program pada mikrokontroler untuk mengelola data yang dikirimkan oleh *Android* sehingga mekanisme dapat dikendalikan oleh *Android*.

1.5 Manfaat Penelitian atau Perancangan

Dengan dirancangnya mekanisme lengan pembuka pintu pada robot asisten medis, maka dapat membantu petugas medis dalam mengantarkan kebutuhan pasien karantina COVID-19 melalui robot yang dikendalikan dari jauh ke ruang isolasi, meminimalisir kontak langsung dengan pasien sehingga dapat menekan penyebaran virus COVID-19.

1.6 Sistematika Penulisan

Secara garis besar laporan ini terdiri dari 5 bab, yang mana dari 5 bab tersebut dibagi lagi menjadi beberapa subbab dengan sistematika penulisan sebagai berikut :

1. Bab 1 yaitu pendahuluan, bab ini terdiri dari latar belakang, rumusan masalah, batasan masalah, tujuan perancangan, manfaat perancangan dan sistematika penulisan.
2. Bab 2 membahas tentang tinjauan pustaka yang terdiri dari landasan teori sistem dan program yang digunakan dalam membuat tugas akhir.
3. Bab 3 membahas tentang metode penelitian yang mana meliputi alur perancangan, alat dan bahan yang digunakan serta proses perancangan
4. Bab 4 membahas mengenai hasil dari perancangan yang telah dilakukan dan analisis tentang data-data pengujian yang sudah dilakukan
5. Bab 5 yaitu penutup, yang berisi tentang kesimpulan dan saran

BAB 2

TINJAUAN PUSTAKA

2.1 Kajian Pustaka

Android adalah sistem operasi pada *smartphone* berbasis Linux yang bersifat *open source*, sehingga memungkinkan pengguna dapat membuat aplikasi sendiri atau melakukan modifikasi pada sistem *Android* itu sendiri, oleh karena itu sudah cukup banyak yang melakukan penelitian terkait aplikasi *Android*, salah satunya pada penelitian yang dilakukan oleh Rizal Khuzzai pada tahun 2014 yang berjudul “*Prototipe Sistem Kendali Kran Elektrik Pada Meteran Air PDAM Berbasis Aplikasi Android*” yang mana pada penelitian tersebut Rizal Khuzzai membuat alat pengendali dan memantau kran elektrik pada meteran air PDAM dengan memanfaatkan aplikasi *Android* [3].

Salah satu alasan *Android* banyak digunakan sebagai pengendali karena pada *smartphone Android* sekarang sudah dilengkapi dengan berbagai macam teknologi terbaru, seperti teknologi komunikasi nirkabel yang membuat kendali jarak jauh semakin mudah. Teknologi komunikasi nirkabel yang sudah pasti ada pada *smartphone Android* sekarang diantaranya yaitu teknologi *Bluetooth* dan *Wi-Fi* yang memanfaatkan sinyal radio untuk melakukan komunikasi. Penelitian mengenai penggunaan *Bluetooth* sebagai media pengiriman data secara nirkabel sudah cukup banyak dilakukan. Salah satu contoh penelitian yang memanfaatkan *Bluetooth* yaitu penelitian yang dilakukan oleh Tito Aminullah pada tahun 2020 dengan judul “*Rancang Bangun Drone Pembersih Sampah Menggunakan Android UNO Sebagai Pengendali*”, pada penelitian tersebut dirancang sebuah *drone* dengan memanfaatkan *Arduino* dan *Bluetooth* sebagai pengendalinya yang mana *drone* tersebut bertugas membersihkan sampah-sampah pada permukaan air [4]. Selain *Bluetooth*, penelitian menggunakan *Wi-Fi* juga telah banyak dilakukan, salah satunya penelitian yang dilakukan oleh Aziz Isrofi, Shoffin Nahwa Utama dan Oddy Virgantara Putra pada tahun 2021 yang berjudul “*Rancang Bangun Robot Pemotong Rumput Otomatis Menggunakan Wireless Kontroler Modul Esp32-Cam Berbasis Internet Of Things (Iot)*”, pada penelitian tersebut dirancang

sebuah robot pemotong rumput otomatis yang dapat dikontrol menggunakan *Smartphone* melalui jaringan *Wi-Fi* [5].

Sistem kendali secara nirkabel dengan menggunakan *Android* sebagai pengendali membutuhkan suatu mikrokontroler untuk menerima dan mengolah data yang dikirimkan oleh *Android* melalui jaringan *Bluetooth* maupun *Wi-Fi*, salah satu mikrokontroler yang sering digunakan adalah *Arduino*. *Arduino* adalah sebuah papan rangkaian elektronik yang bersifat *open source*. *Arduino* sudah banyak digunakan dalam dunia robotik untuk mengelola data pada robot sehingga robot dapat berfungsi dengan sempurna, salah satu penelitian yang memanfaatkan *Arduino* adalah penelitian yang dilakukan oleh Ulinnuha latifa dan Joko Slamet Saputro pada tahun 2018 dengan judul “*Perancangan Robot ARM Gripper Berbasis Arduino UNO Menggunakan Antarmuka Labview*”, pada penelitian tersebut dirancang sebuah robot *ARM* dengan menggunakan *Arduino* untuk menggerakkan aktuator (motor servo) pada robot *ARM* yang telah dirancang [6].

Dunia robotik saat ini sudah semakin maju, sudah banyak jenis robot yang dikembangkan untuk membantu tugas manusia dalam berbagai bidang seperti medis, pertanian, pelayanan dan lainnya. Penelitian yang belakangan ini cukup banyak dilakukan yaitu penelitian mengenai robot asisten medis, salah satunya adalah penelitian yang berjudul “*Rancang Bangun Robot Pelayan Medis untuk Pasien Karantina COVID-19 dengan Kendali berbasis Android*” yang diteliti oleh Donny Suryawan dan R.M. Sisdarmanto Adinandra. Pada penelitian tersebut dirancang dan dibuat robot semi otomatis untuk membantu petugas medis dalam melayani pasien karantina COVID-19 [7], namun robot yang dibuat belum mampu untuk membuka pintu sendiri, sehingga berpotensi menghambat kinerja dari robot dalam menjalankan tugasnya untuk mengantarkan barang ke dalam ruangan isolasi pasien COVID-19 yang sedang dikarantina.

Mengacu dari kelima penelitian tersebut, maka dilakukan penelitian tentang perancangan sistem kendali mekanisme pembuka pintu pada robot asisten medis berbasis *Android* yang memanfaatkan koneksi *Wi-Fi* pada *Android* sebagai pengendali dari mekanisme pembuka pintu pada robot asisten medis sehingga dapat membuka pintu dan meningkatkan kinerja robot dalam membantu petugas medis dalam melayani pasien.

Tabel 2-1 Ringkasan kajian pustaka

No	Tahun	Judul	Penulis	Kontribusi
1	2014	Prototipe Sistem Kendali Kran Elektrik Pada Meteran Air PDAM Berbasis Aplikasi <i>Android</i>	Rizal Khuzzai	Sebagai acuan untuk membuat aplikasi <i>Android</i> menggunakan MIT App Inventor
2	2018	Perancangan Robot <i>ARM Gripper</i> Berbasis <i>Arduino UNO</i> Menggunakan Antarmuka <i>Labview</i>	Ulinuha latifa dan Joko Slamet Saputro	Sebagai acuan dalam merancang rangkaian dan perintah pada <i>Arduino</i> untuk menggerakkan aktuator
3	2020	Rancang Bangun <i>Drone</i> Pembersih Sampah Menggunakan <i>Android UNO</i> Sebagai Pengendali	Tito Aminullah	Sebagai acuan dalam merancang rangkaian <i>Bluetooth</i> dan pengaplikasiannya pada <i>Arduino</i>
4	2021	Rancang Bangun Robot Pelayan Medis untuk Pasien Karantina COVID-19 dengan Kendali berbasis <i>Android</i>	Donny Suryawan dan R.M. Sisdarmanto Adinandra	Sebagai acuan penempatan sistem kendali dan sumber daya untuk sistem kendali
5	2021	Rancang Bangun Robot Pemotong Rumput Otomatis Menggunakan <i>Wireless</i> Kontroler Modul <i>Esp32-Cam</i> Berbasis <i>Internet Of Things (Iot)</i>	Aziz Isrofi, Shoffin Nahwa Utama dan Oddy Virgantara Putra	Sebagai acuan dalam penggunaan <i>Wi-Fi</i> sebagai jalur komunikasi dan merancang rangkaian menggunakan modul <i>ESP32</i>

2.2 Landasan Teori

2.2.1 Sistem Kendali

Sistem kendali merupakan sebuah sistem yang dibuat untuk mengendalikan suatu komponen dari sistem itu sendiri, dalam hal ini yaitu mengendalikan perangkat keras (*hardware*) melalui perangkat lunak (*software*) [8].

2.2.2 Android

Android adalah sebuah sistem operasi pada *Smartphone* yang bersifat *open source* berbasis Linux. *Android* menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri [4].

2.2.3 Arduino

Papan *Arduino* merupakan papan rangkaian elektronik yang *open source* yang komponen utamanya adalah *microcontroller 8 bit* dengan merk ATmega, papan *Arduino* banyak macamnya tergantung spesifikasinya, seperti *Arduino UNO* yang menggunakan ATmega 328, *Arduino mega 2560* yang menggunakan ATmega2560 dan masih banyak lagi. Selain papan *microcontroller Arduino* juga merupakan *software* yang berfungsi sebagai *editor program*, *compiler*, dan *uploader* sehingga dapat memasukkan perintah pada papan *Arduino*, *software Arduino* ini diberi nama *Arduino IDE* [9].

2.2.4 ESP32 Board

ESP32 Board atau papan *ESP32* adalah papan rangkaian elektronik *open source* dari sistem Espressif, sama seperti *Arduino* papan *ESP32* banyak macamnya tergantung dengan spesifikasinya, namun perbedaan *ESP32* dengan *Arduino* yaitu biasanya pada papan *ESP32* sudah dilengkapi dengan *Wi-Fi* dan *Bluetooth*. Untuk memrogram papan *ESP32* dapat dilakukan dengan beberapa *software* diantaranya yaitu *ESP-IDF*, *Arduino IDE*, *MicroPython* dan *Mongoose OS* [10].

2.2.5 Wireless communication

Komunikasi *wireless* bekerja dengan sinyal elektromagnetik yang dipancarkan oleh suatu perangkat pengirim sinyal yang kemudian sinyal tersebut diterima oleh perangkat penerima sehingga menciptakan jembatan komunikasi *wireless* (tanpa kabel) antara perangkat pengirim dan perangkat penerima. [11].

2.2.6 Bluetooth

Bluetooth adalah salah satu teknologi komunikasi nirkabel yang dapat menghubungkan atau melakukan kegiatan tukar-menukar informasi dengan memanfaatkan transmisi radio [11].

2.2.7 Wi-Fi

Wi-Fi merupakan kependekan dari *Wireless Fidelity* yaitu suatu media komunikasi data nirkabel yang menggunakan sinyal radio dengan frekuensi tertentu sebagai media penghantar data komunikasinya [12].

2.2.8 Smartphone

Smartphone sudah sangat berkembang pesat. Pengguna *smartphone* dapat melakukan banyak hal dengan *smartphone*, mulai dari menjadi alat komunikasi, tempat membaca berita, sebagai alarm hingga dapat menjadi *remote control* untuk mengendalikan robot [13].

2.2.9 Robot semi otomatis

Robot semi otomatis adalah yang dalam melakukan fungsinya menerima instruksi dari luar atau operator. Robot semi otomatis memiliki sistem yang lebih sederhana dibandingkan dengan robot otomatis karena sistemnya yang terbuka dan tidak banyak data informasi yang harus olah [14].

2.2.10 Module BTS 7960

Module BTS 7960 merupakan salah satu perangkat elektronik yang menerapkan sistem *high current H bridge* untuk aplikasi penggerak motor dengan

arus tinggi. *Module BTS 7960* sudah dilengkapi dengan sensor arus, perlindungan terhadap suhu berlebih, tegangan berlebih, tegangan kurang, arus berlebih agar tidak terjadi korsleting dan merusak komponen elektronik lainnya [15].

Tabel 2-2 Spesifikasi Singkat *Module BTS 7960*

Spesifikasi Singkat	
Tegangan <i>Input</i>	6 ~ 27 V (DC)
Arus Maksimal	43 A
Kemampuan PWM	Hingga 25 kHz
Mode Kontrol	PWM atau Level
Ukuran	50 x 50 x 43 mm
Berat	~ 66 g

2.2.11 *Bluetooth Module HC-05*

Bluetooth module HC-05 merupakan *module* yang beroperasi berdasarkan prinsip *Serial Port Protocol* (SPP). *Module* ini dirancang khusus untuk komunikasi serial nirkabel. *Bluetooth module HC-05* dapat dioperasikan sebagai *master* atau *slave*, pada *module* ini komunikasi yang dilakukan melalui komunikasi serial yang mana membuatnya mudah untuk berinteraksi dengan mikrokontroler atau PC [16]. *Interface* yang digunakan adalah Serial RXD, TXD, VCC dan GND, selain itu *Bluetooth module HC-05* juga sudah dilengkapi dengan LED sebagai indikator koneksi *Bluetooth*.

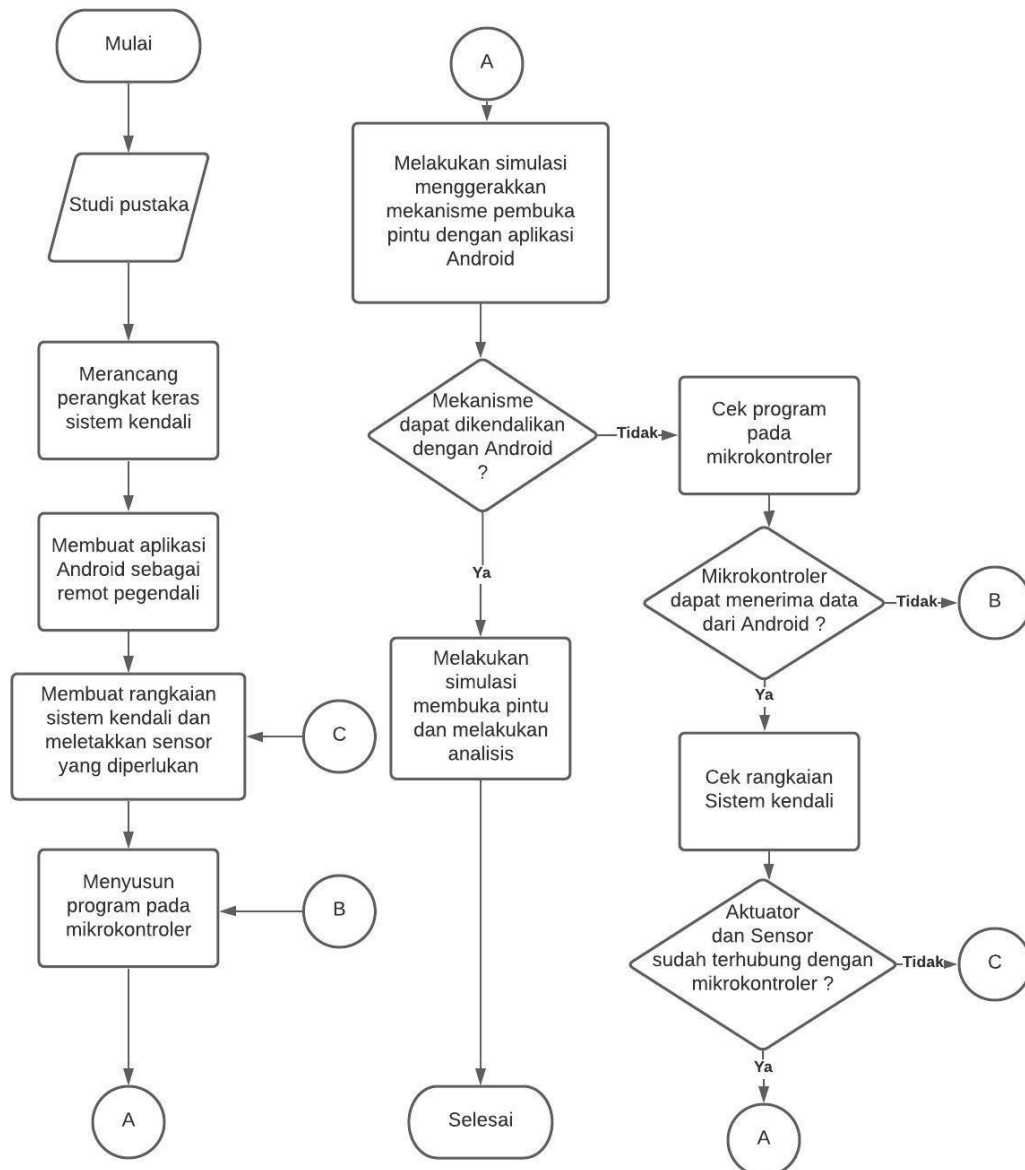
2.2.12 *MIT App Inventor*

MIT App Inventor adalah sebuah sistem berbasis web yang bersifat *open source* yang mana memungkinkan pengguna membuat dan mengembangkan aplikasi pada OS *Android* sesuai kemauannya sendiri. *MIT App Inventor* menggunakan *visual block programming* sebagai pengganti bahasa pemrograman sehingga dapat memudahkan pengguna dalam menyusun perintah yang diinginkan pada aplikasi yang akan dibuat, pengguna hanya perlu melakukan *drag and drop* setiap perintah yang ada tanpa harus menuliskan bahasa pemrogramannya [17]. *MIT App Inventor* tersedia secara gratis, pengguna hanya membutuhkan koneksi internet untuk dapat mengaksesnya.

BAB 3

METODE PENELITIAN

3.1 Alur Perancangan



Gambar 3-1 Alur Perancangan

3.2 Peralatan dan Bahan

A) Peralatan

Peralatan yang digunakan pada perancangan ini terbagi menjadi 2 jenis, yaitu perangkat keras dan perangkat lunak.

a. Perangkat Keras

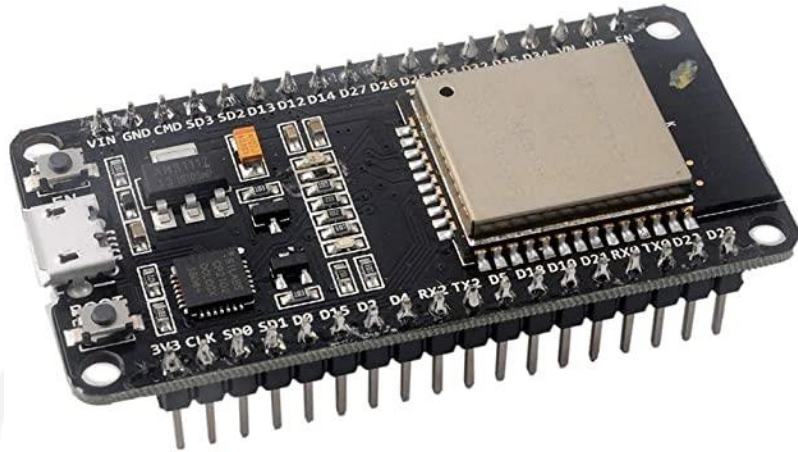
1) *Arduino*



Gambar 3-2 *Arduino*

Arduino adalah salah satu papan rangkaian elektronik yang mudah untuk di program, pada perancangan kali ini *Arduino* digunakan sebagai pengelola data yang diterima dari modul *Bluetooth* dan sensor yang digunakan pada rangkaian sistem kendali serta sebagai pengatur gerakan dari aktuator yang digunakan (motor *Power Window Universal* dan motor servo MG 966R).

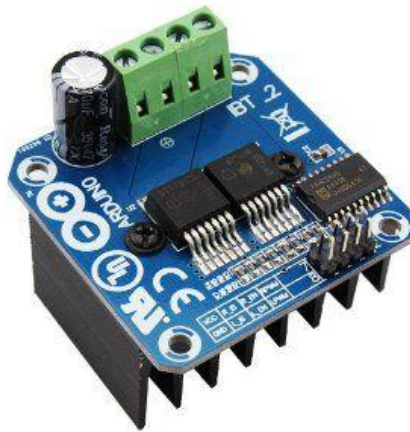
2) *ESP32 Board*



Gambar 3-3 *ESP32 Board*

ESP32 Board adalah salah satu papan rangkaian elektronik yang mudah untuk di program, pada perancangan kali ini *ESP32 Board* berfungsi sebagai alternatif papan *Arduino* sebagai pengelola data yang diterima dari *Android* melalui jaringan *Wi-Fi*.

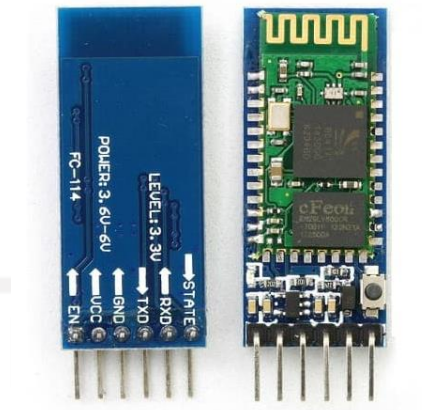
3) *Motor Driver BTS7960*



Gambar 3-4 *Motor Driver BTS 7960*

Motor driver *BTS7960* adalah salah satu perangkat elektronik pengendali motor yang mana digunakan untuk mengendalikan gerakan motor *Power Window Universal* yang gerakannya sesuai dengan perintah yang diberikan oleh *Arduino* atau *ESP32*.

4) *Bluetooth* modul HC-05



Gambar 3-5 *Bluetooth* modul HC-05

Bluetooth modul HC-05 adalah perangkat elektronik yang memanfaatkan jaringan *Bluetooth*, pada perancangan ini *Bluetooth* modul HC-05 digunakan sebagai penerima data yang dikirimkan oleh *Android* melalui jaringan *Bluetooth* yang kemudian dikirim ke *Arduino* untuk dikelola.

b. Perangkat Lunak

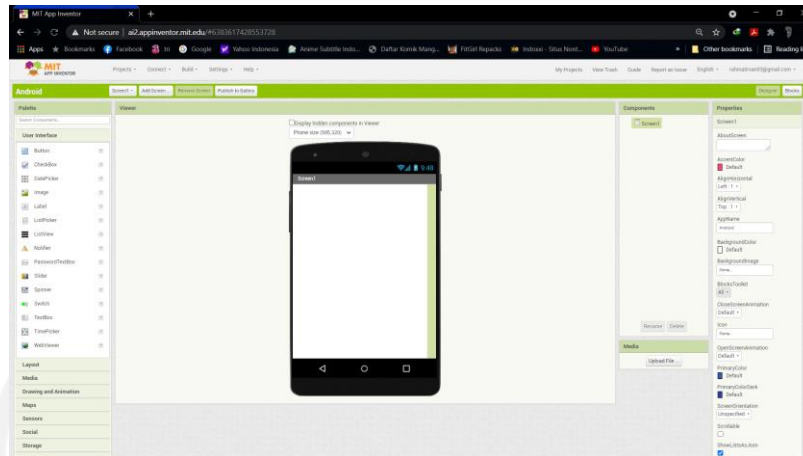
1) *Arduino IDE*



Gambar 3-6 Tampilan awal *Arduino IDE*

Arduino IDE adalah *software* yang digunakan untuk menyusun perintah-perintah pada *Arduino* maupun *ESP32* sehingga dapat mengelola data yang diterima dan mengendalikan mekanisme pembuka pintu pada robot asisten medis.

2) MIT App Inventor



Gambar 3-7 Tampilan awal MIT App Inventor

MIT App Inventor adalah *software* yang digunakan untuk membuat aplikasi yang berfungsi sebagai remot pengendali yang dapat mengirimkan data ke *Arduino* atau *ESP32* sehingga mekanisme pembuka pintu dapat dikendalikan secara nirkabel.

B) Bahan

Bahan yang digunakan dalam perancangan ini adalah sebagai berikut :

a. *Limit switch*



Gambar 3-8 *Limit switch*

Limit switch adalah suatu perangkat elektromekanis yang memiliki tuas aktuator sebagai kontak terminal dari NO ke NC atau sebaliknya NC ke NO, pada perancangan kali ini *Limit switch* berfungsi sebagai sensor pada mekanisme pembuka pintu sebagai batas ketika lengan digerakkan maju dan batas ketika lengan digerakkan mundur.

b. Motor Servo MG 996R



Gambar 3-9 Motor servo MG 996 R

Motor servo MG 996R adalah sebuah aktuator putar yang dapat berputar maksimal 120° di mana pengguna dapat mengatur posisi sesuai yang diinginkan, pada perancangan ini Motor servo MG 996R berfungsi sebagai penggerak *hook* sehingga *hook* dapat “memegang” gagang pintu.

c. Motor Power Window Universal



Gambar 3-10 Motor Power Window Universal

Motor Power Window Universal adalah salah satu aktuator putar yang memiliki torsi cukup besar yaitu 0-90 kg/cm dan menggunakan tegangan 12 V. Pada perancangan ini Motor Power Window Universal berfungsi sebagai penggerak pada lengan mekanisme pembuka pintu sehingga dapat menarik pintu ketika proses membuka pintu berlangsung.

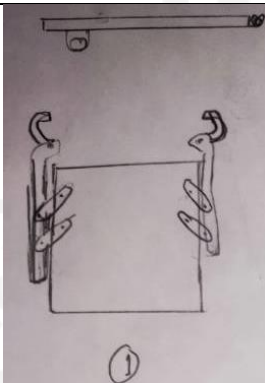
3.3 Perancangan

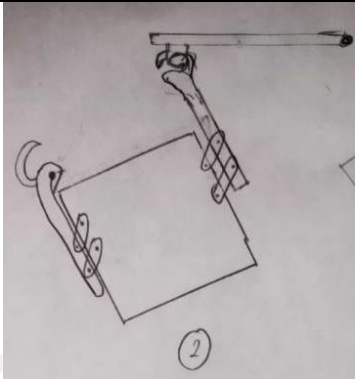
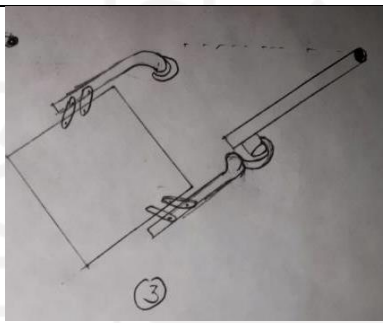
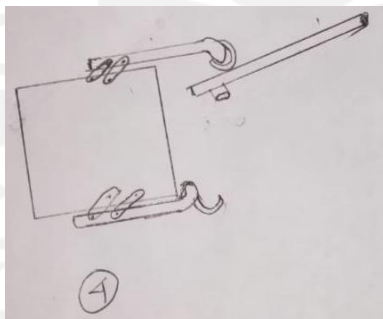
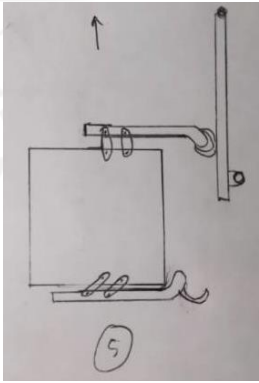
3.3.1 Kriteria Perancangan

Dalam perancangan sistem kendali mekanisme pembuka pintu pada robot asisten medis, terdapat beberapa kriteria perancangan yang harus dipenuhi agar hasil perancangan yang diperoleh adalah rancangan yang paling optimal. Berikut adalah kriteria perancangan sistem kendali mekanisme pembuka pintu pada robot asisten medis :

1. Sistem kendali harus bisa mengendalikan mekanisme pembuka pintu pada robot asisten medis untuk membuka pintu
2. Sistem kendali yang akan dibuat harus dapat dikendalikan dari jarak jauh tanpa kabel dengan menggunakan *Android*
3. Sistem kendali harus bisa mengendalikan alur gerak aktuator sesuai dengan gerakan robot membuka pintu. Alur gerakan robot tersebut dapat dilihat pada Tabel 3-1 berikut :

Tabel 3-1 Alur gerakan robot ketika membuka pintu

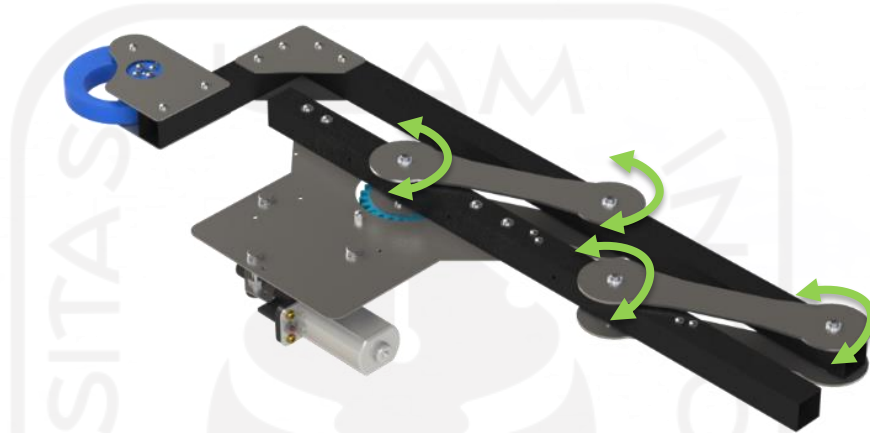
Langkah ke-	Ilustrasi	Keterangan
1		Pertama robot diposisikan menghadap pintu dan berada 30 cm di depan pintu dengan sisi kiri robot sejajar dengan gagang pintu yang akan dibuka.

2		<p>Langkah kedua yaitu robot diposisikan menyerong ke kiri, kemudian mekanisme pada sisi kanan dimajukan dan meraih gagang pintu menggunakan <i>hook</i>.</p>
3		<p>Setelah <i>hook</i> berhasil “memegang” gagang pintu, robot digerakkan berputar $\pm 90^\circ$ sehingga pintu terbuka $\pm 45^\circ$.</p>
4		<p>Langkah selanjutnya yaitu, mekanisme pada sisi kiri akan dimajukan sampai batas maksimum, lalu <i>hook</i> pada sisi kanan akan melepaskan gagang pintu sehingga pintu akan tersangkut pada mekanisme pada sisi kiri robot.</p>
5		<p>Terakhir robot memundurkan mekanisme pada sisi kanan, kemudian mendorong pintu menggunakan mekanisme pada sisi kiri hingga pintu terbuka 90° sehingga robot dapat masuk ke dalam ruangan dengan cara bergerak menyamping ke kiri.</p>

3.3.2 Perancangan Perangkat Keras

Perancangan sistem kendali mekanisme pembuka pintu dimulai dari merancang perangkat keras, di mana perangkat keras tersebut terbagi menjadi 2 bagian yaitu perangkat keras yang berfungsi sebagai yang dikendalikan dan perangkat keras yang berfungsi sebagai pengendali.

1. Perancangan perangkat keras yang akan dikendalikan



Gambar 3-11 Desain mekanisme pembuka pintu pada robot asisten medis

Berdasarkan Gambar 3-11 terlihat bahwa perangkat keras yang akan dikendalikan berupa aktuator dan sensor, di mana untuk satu lengan pembuka pintu, aktuator yang akan dikendalikan adalah 1 motor *power window universal* yang berfungsi menggerakkan lengan maju atau pun mundur untuk menarik pintu, 1 motor servo yang berfungsi sebagai penggerak *hook* sehingga lengan dapat “memegang” gagang pintu sehingga pintu dapat dibuka. Untuk sensor yang digunakan adalah sensor batas yaitu *limit switch* yang berfungsi sebagai batas gerakan lengan agar tidak bergerak berlebihan sehingga merusak komponen-komponen lain.

- a. Aktuator

Motor *power window universal* sebagai penggerak lengan diletakkan pada *gear* rasio yang dilekatkan menggunakan baut dan mur pada struktur robot sehingga dapat menggerakkan lengan untuk maju atau mundur. Sedangkan untuk motor servo yang digunakan adalah motor servo MG 996R yang diletakkan pada ujung lengan robot sehingga dapat menggerakkan *gear* yang terhubung dengan *hook*.

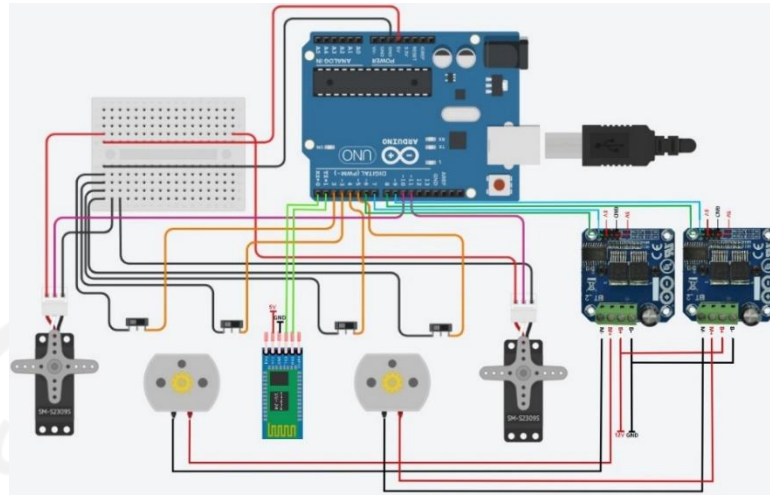
b. Sensor

Selain menggerakkan aktuator, pada sistem kendali yang dibuat juga menggunakan sensor berupa *limit switch*, yang berfungsi sebagai pembatas gerakan pada lengan sehingga dapat meminimalisir kerusakan pada komponen-komponen lengan yang lainnya.

2. Perancangan perangkat keras pengendali

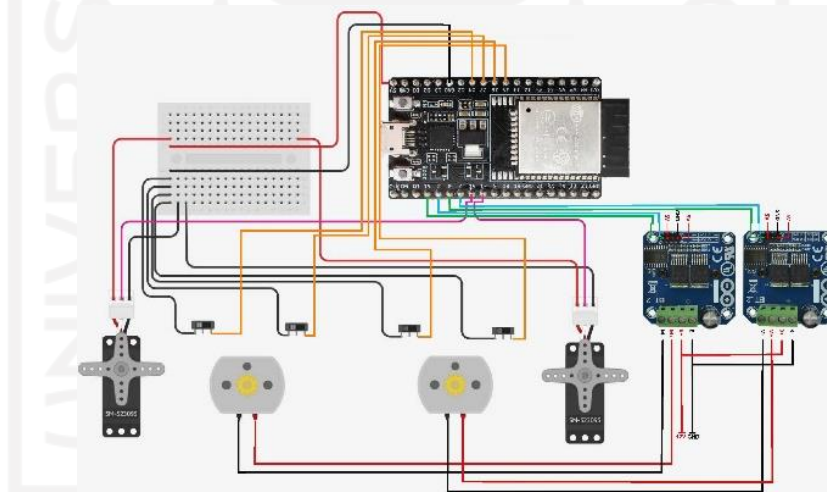
Setelah rancangan perangkat keras yang akan dikendalikan sudah selesai, maka langkah selanjutnya adalah merancang perangkat keras pengendali yang berupa rangkaian sistem kendali. Seperti yang telah dijelaskan sebelumnya, pada perancangan ini motor yang digunakan adalah motor *power window universal*, di mana motor tersebut membutuhkan tegangan 12 Volt dan 2 Ampere untuk berjalan normal, maka dibutuhkan *driver* motor yang dapat menahan beban tersebut. Oleh karena itu dari sekian banyak motor *driver* dipilih motor *driver* BTS 9760 yang sanggup menahan beban hingga 27 Volt dan 43 Ampere, untuk pusat pengendali pada mulanya menggunakan *Arduino* dan *Bluetooth* sebagai media pengiriman data dari *Android* ke *Arduino*, namun pada sistem ini terdapat banyak kekurangan seperti jarak maksimal yang dapat dicapai hanya 10 m dan koneksi antara *Android* dan *Arduino* yang kurang stabil (terkadang mengalami *lost connection*). Oleh karena itu maka dibuatlah alternatif menggunakan *ESP32* dengan *Wi-Fi* sebagai media pengiriman data sehingga jarak yang dapat dicapai menjadi 100 m, bahkan lebih jika menggunakan pemancar *Wi-Fi* tambahan, selain itu koneksi menggunakan *Wi-Fi* terbilang cukup stabil. Berikut adalah rangkaiannya :

a. Rangkaian menggunakan *Arduino*



Gambar 3-12 Ilustrasi rangkaian sistem kendali mekanisme pembuka pintu menggunakan *Arduino*

b. Rangkaian menggunakan *ESP32*



Gambar 3-13 Ilustrasi rangkaian sistem kendali mekanisme pembuka pintu menggunakan *ESP32*

Gambar 3-12 merupakan rangkaian sistem kendali menggunakan *Arduino*, pada rangkaian ini memerlukan modul *Bluetooth* tambahan sedangkan pada Gambar 3-13 terlihat rangkaian sistem kendali dengan menggunakan *ESP32* lebih sederhana karena modul *Wi-Fi* sudah menjadi satu dengan *ESP32* itu sendiri sehingga tidak memerlukan modul tambahan lagi. Pada perancangan ini digunakan PCB lubang yang kemudian disolder dengan pin *header* untuk menghubungkan *Arduino* maupun *ESP32* dengan aktuator beserta sensor sesuai rangkaian.

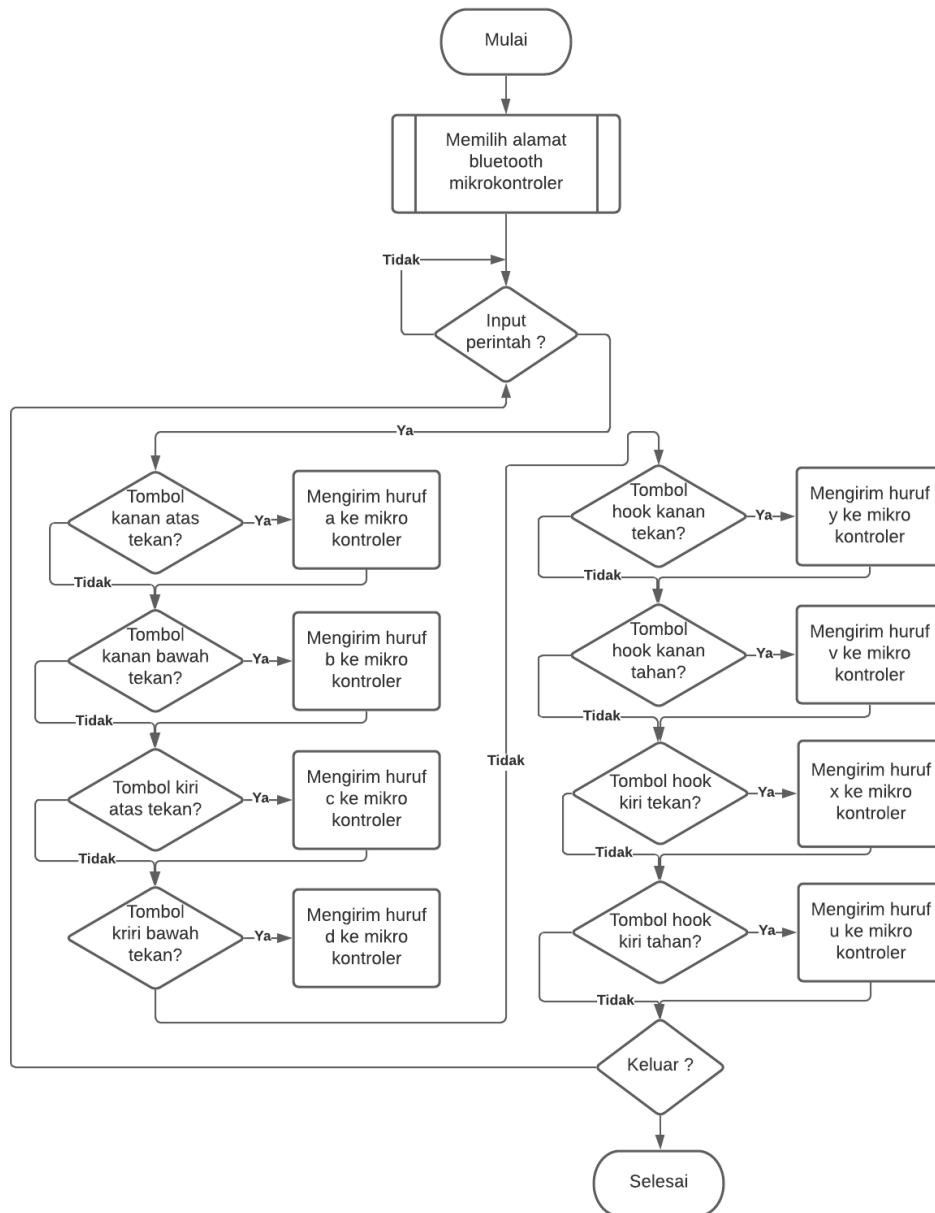
3.3.3 Perancangan Perangkat Lunak

Perancangan perangkat lunak untuk sistem kendali mekanisme pembuka pintu dibagi menjadi 2 proses, yaitu merancang aplikasi remot kontrol dan merancang program perintah pada *Arduino* ataupun *ESP32*. Berikut adalah langkah-langkah yang dilakukan pada proses perancangan perangkat lunak sistem kontrol pada mekanisme pembuka pintu robot asisten medis :

1. Membuat Aplikasi remot kontrol untuk *Android*

Aplikasi remot *Android* untuk sistem kendali mekanisme pembuka pintu pada robot medis dirancang menggunakan bantuan *software* MIT App Inventor, di mana langkah pertamanya adalah membuat tampilan aplikasi seperti tata letak tombol, bentuk tombol dan lainnya sesuai yang dibutuhkan yang dilanjutkan dengan menyusun program yang akan berjalan pada tiap-tiap tombol yang sudah dibuat. Berikut adalah skema alur kerja aplikasi remot *Android* :

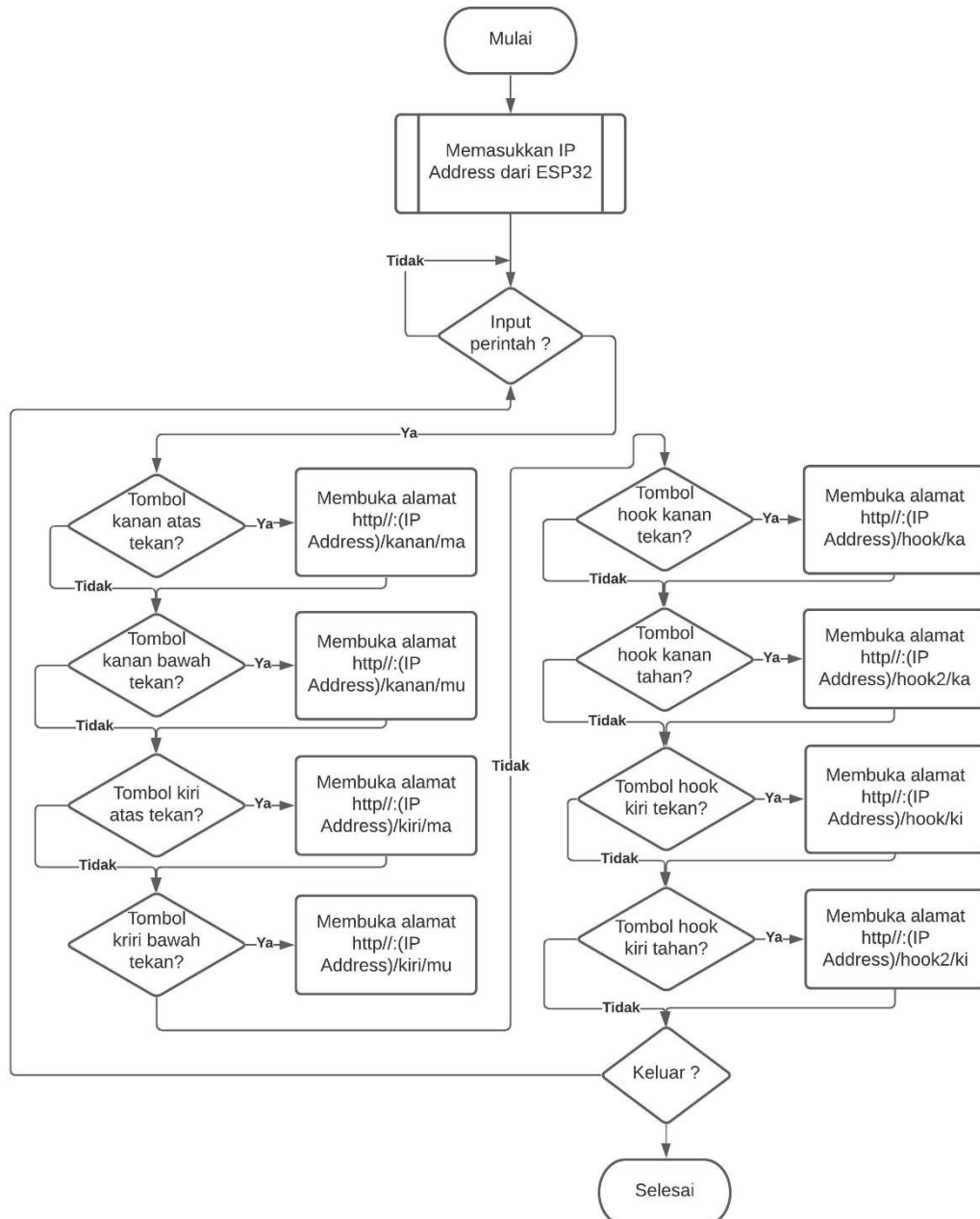
a. Alur kerja aplikasi *Android* untuk *Arduino* dengan koneksi *Bluetooth*



Gambar 3-14 Skema Alur Kerja Aplikasi Remot pada *Android* untuk penggunaan *Arduino*

Gambar 3-14 merupakan alur skema kerja transfer data antara remot pada *Android* dengan *Arduino*. Pada sistem ini transfer data dilakukan dengan cara komunikasi serial, di mana ketika tombol di remot pada *Android* di tekan maka *Android* akan mengirimkan data berupa data serial (huruf a, b, c dapat dilihat diagram alir pada Gambar 3-14) yang kemudian akan diterima oleh modul *Bluetooth* dan diteruskan ke *Arduino* untuk diolah.

b. Alur kerja aplikasi *Android* untuk *ESP32* dengan koneksi *Wi-Fi*



Gambar 3-15 Skema Alur Kerja Aplikasi Remot pada *Android* untuk penggunaan *ESP32*

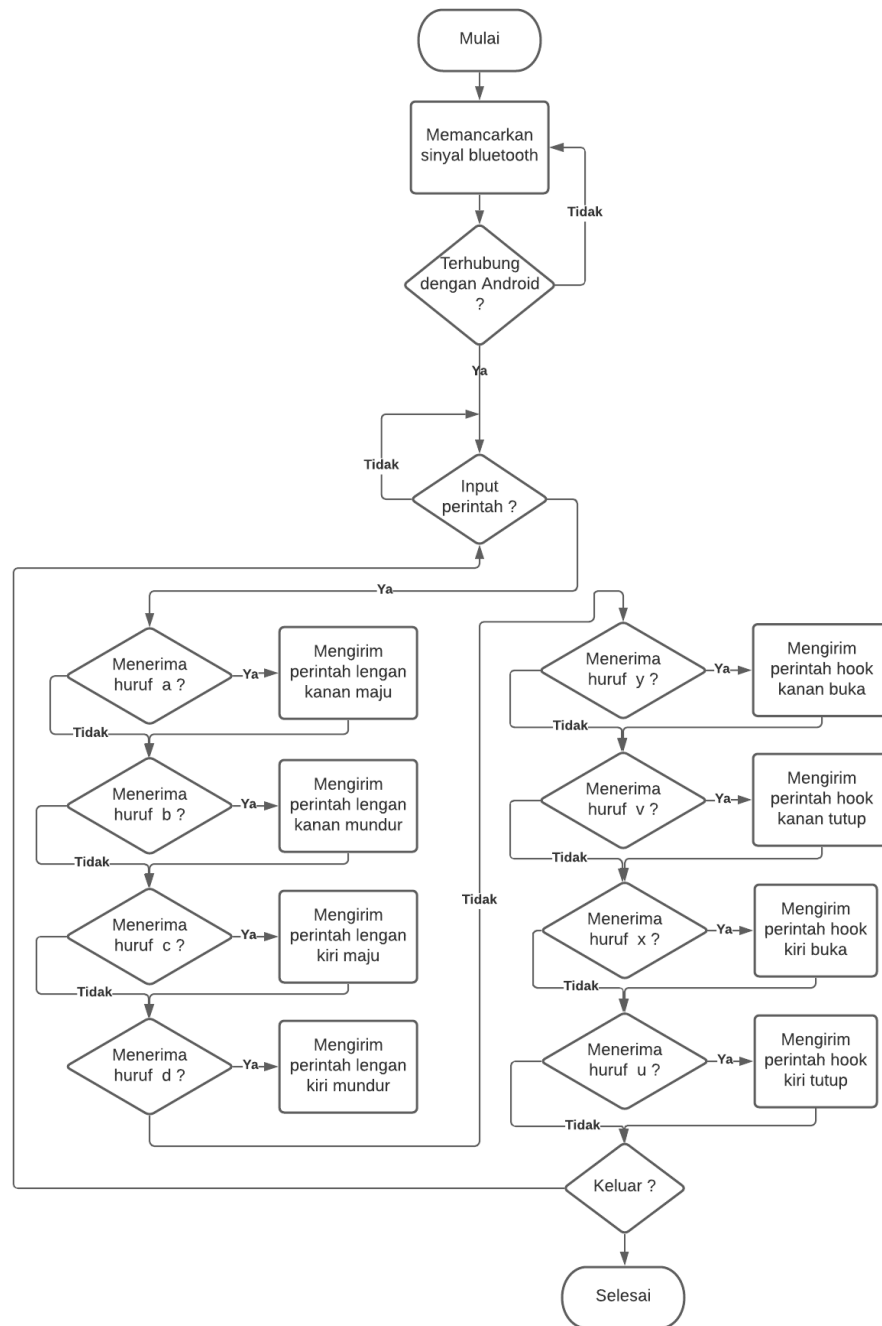
Berubahnya media transfer data dari menggunakan *Bluetooth* dengan *Arduino* menjadi *Wi-Fi* dengan *ESP32* membuat alur skema aplikasi remot di *Android* ikut berubah, hal ini terlihat pada Gambar 3-15 yang menampilkan alur kerja aplikasi remot pada *Android*, pada gambar tersebut terlihat bahwa transfer data tidak lagi menggunakan komunikasi serial melainkan berubah menggunakan sistem server, hal ini terlihat dari perintah yang dilakukan *Android* ketika tombol ditekan adalah

membuka alamat server yang telah dibuat, tidak seperti sebelumnya ketika menggunakan *Bluetooth* di mana ketika tombol ditekan, maka *Android* akan mengirimkan data berupa huruf-huruf. Pada dasarnya terdapat berbagai cara untuk melakukan transfer data antara *Android* dengan *ESP32* melalui jaringan *Wi-Fi* seperti *peer to peer network*, sistem server dan lainnya, namun pada perancangan ini dipilih metode sistem server karena cukup mudah untuk mengaksesnya dan juga sudah cukup banyak penelitian sebelumnya yang menggunakan metode sistem server pada *ESP32*.

2. Menyusun Program Perintah pada *Arduino*

Sistem kendali mekanisme pembuka pintu yang dirancang menggunakan mikrokontroler sebagai pusat kendali, di mana terdapat dua alternatif mikrokontroler yang dapat digunakan, yaitu *Arduino* dan *ESP32*. Kedua alternatif mikrokontroler yang dapat digunakan dapat diprogram dengan berbagai jenis *software*, pada perancangan ini digunakan *software Arduino IDE* dengan bahasa pemrograman C karena kemudahan dalam penggunaan dan sudah mendukung kedua mikrokontroler yang akan digunakan. Berikut adalah skema alur kerja sistem kendali mekanisme pembuka pintu pada robot asisten medis yang dirancang :

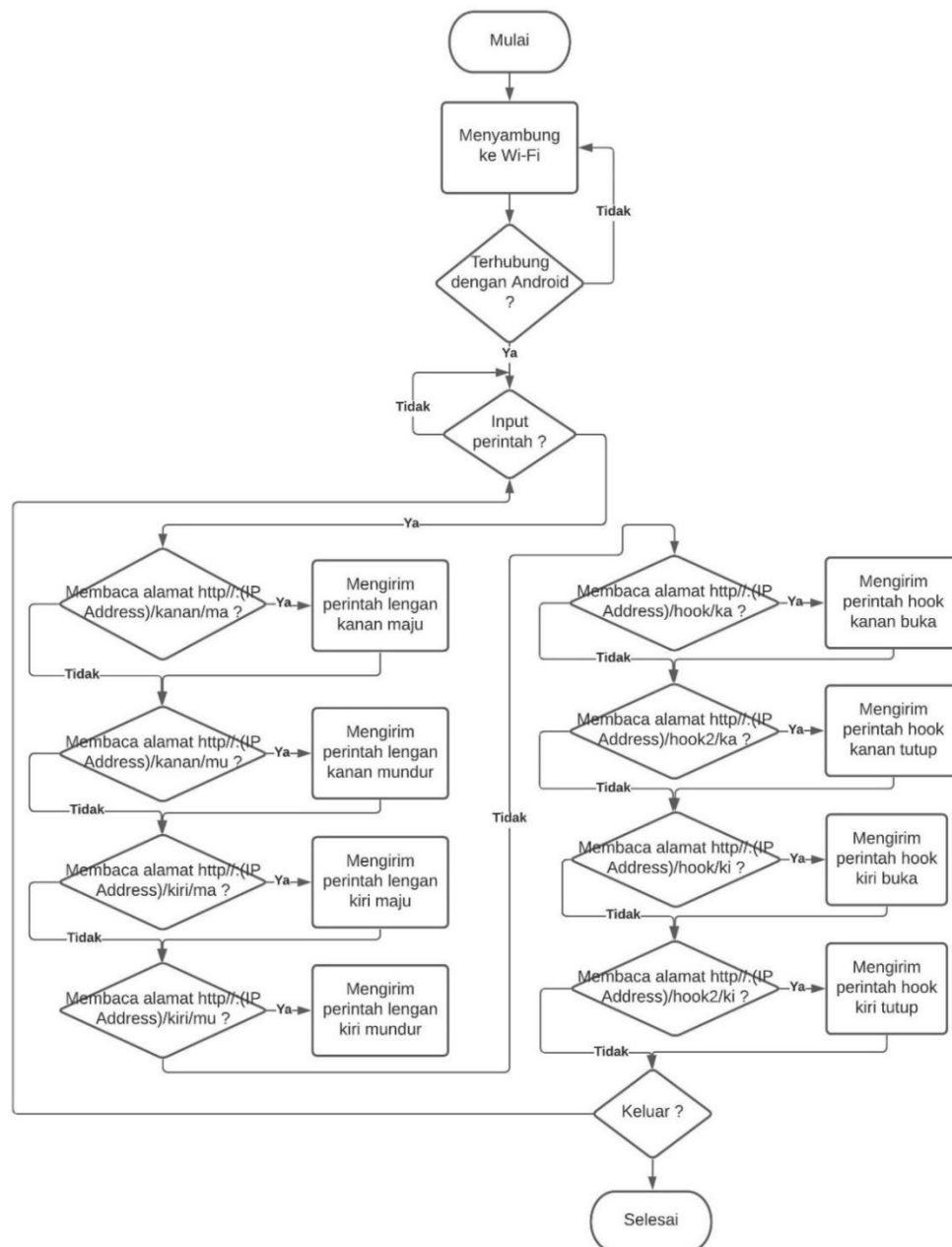
a. Skema alur kerja untuk penggunaan *Arduino*



Gambar 3-16 Skema Alur Kerja pada *Arduino*

Data yang dikirim oleh *Android* akan ditangkap oleh modul *Bluetooth* yang kemudian diolah oleh *Arduino* seperti pada skema alur kerja pada Gambar 3-16, data yang diterima akan menjadi parameter yang digunakan untuk memberi perintah pada aktuator maupun sensor sehingga mekanisme dapat menjalankan tugasnya untuk membuka pintu.

b. Skema alur kerja untuk penggunaan *ESP32*



Gambar 3-17 Skema Alur Kerja pada *ESP32*

Skema alur kerja pada *ESP32* tidak jauh berbeda dengan skema alur kerja pada *Arduino*, perbedaannya terletak pada data yang diterima oleh *ESP32* sebagai parameter untuk memberi perintah pada aktuator dan sensor seperti yang terlihat pada Gambar 3-17, data yang diterima tidak lagi berupa data serial melainkan alamat server, hal ini disebabkan oleh berubahnya media transfer data yang awalnya menggunakan *Bluetooth* menjadi *Wi-Fi* di mana metode komunikasi yang dilakukan yaitu menggunakan sistem server.

3.3.4 Pemilihan Mikrokontroler

Terdapat dua alternatif pilihan mikrokontroler yang dapat digunakan pada perancangan sistem kendali mekanisme pembuka pintu pada robot medis, yaitu *Arduino* dan *ESP32*. *Arduino* dan *ESP32* memiliki kelebihan dan kekurangannya masing-masing, perbandingan *Arduino* dan *ESP32* dapat dilihat pada Tabel 3-2 berikut :

Tabel 3-2 Perbandingan *Arduino* dan *ESP32*

No	Spesifikasi	<i>Arduino</i>	<i>ESP32</i>
1	Konektivitas	<i>Bluetooth</i> , memerlukan modul tambahan	<i>Wi-Fi</i> , sudah menjadi satu dengan <i>ESP32 board</i>
2	Kelancaran mengirim data	Lancar	Terkadang mengalami <i>delay</i>
3	Kemudahan untuk tersambung dengan <i>Android</i>	Hanya perlu memilih alamat <i>Bluetooth</i> yang dipancarkan oleh modul <i>Bluetooth</i>	Perlu alat tambahan untuk mengetahui <i>IP Address ESP32</i> (seperti laptop atau menggunakan LCD)
4	Kekuatan Jaringan	Terkadang koneksi <i>Android</i> dan <i>Arduino</i> mengalami <i>lost connection</i>	Koneksi antara <i>Android</i> dan <i>ESP32</i> cukup stabil
5	Jangkauan efektif Pengendali	10 m	100 m
6	Biaya yang diperlukan	Harga <i>Arduino</i> = Rp.80.000 dan butuh tambahan <i>Bluetooth</i> =Rp. 50.000 sehingga totalnya Rp. 130.000	Harga <i>ESP32</i> = Rp. 90.000

Berdasarkan Tabel 3-2 terlihat bahwa *ESP32* unggul pada bagian konektivitas, kekuatan jaringan, jangkauan efektif pengendali dan biaya yang diperlukan sedangkan *Arduino* hanya unggul pada bagian kelancaran mengirim data dan kemudahan untuk tersambung dengan *Android*, dari perbandingan tersebut, jelas *ESP32* memiliki lebih banyak keunggulan dibandingkan *Arduino*. Oleh karena itu dipilihlah *ESP32* sebagai pusat kendali yang akan digunakan pada sistem kendali mekanisme pembuka pintu.

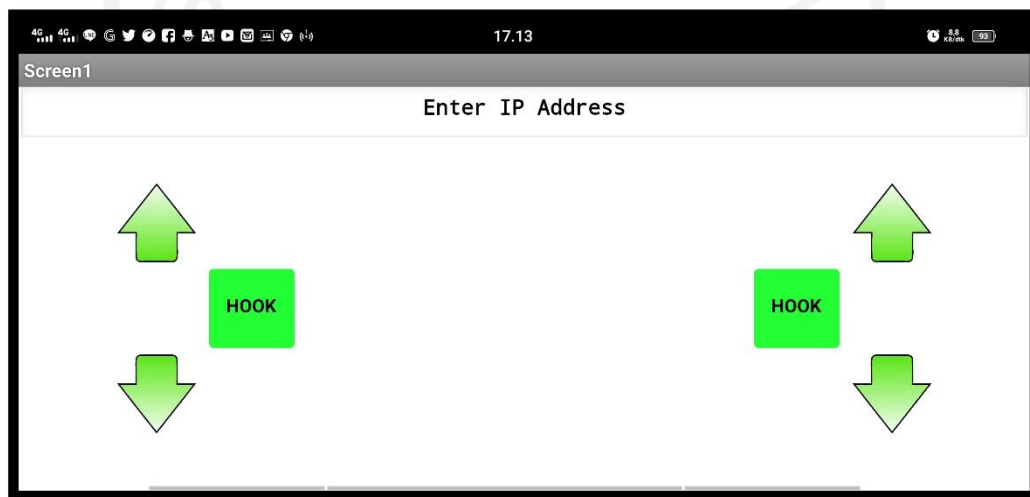
BAB 4

HASIL DAN PEMBAHASAN

4.1 Hasil Perancangan

Hasil dari perancangan yang telah dilakukan yaitu berupa rangkaian sistem pengendali dan aplikasi *Android* yang berperan sebagai remot untuk mengirim perintah pada *ESP32* sebagai pusat sistem kendali yang telah dibuat.

1. Aplikasi *Android*



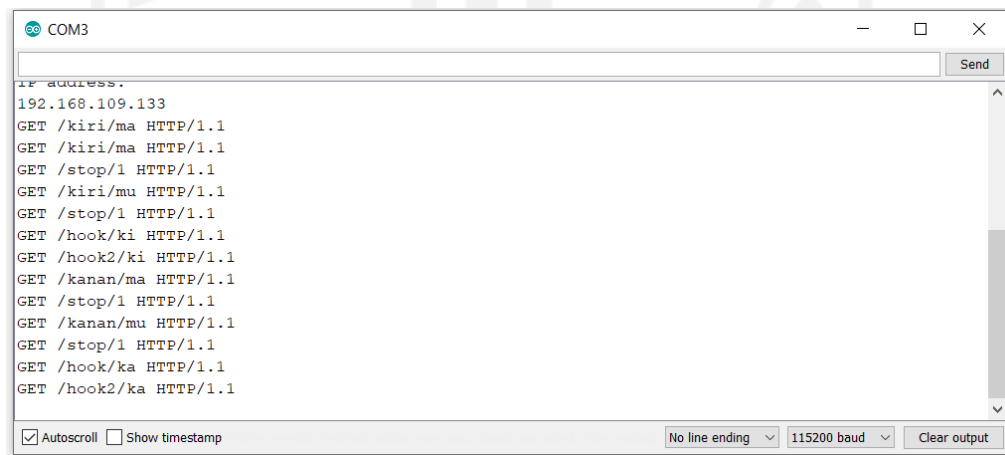
Gambar 4-1 Tampilan Aplikasi Remot

Pada Gambar 4-1 terlihat tampilan pada aplikasi *Android* yang telah dibuat menggunakan *software* MIT App Inventor. Pada aplikasi tersebut terdapat 6 tombol dan 1 *text box* yang dapat digunakan sebagai pengendali mekanisme pembuka pintu pada robot asisten medis, tiap tombol memiliki fungsinya masing-masing. Fungsi pada tiap tombol dapat dilihat pada Tabel 4-1 berikut :

Tabel 4-1 Fungsi tiap tombol pada aplikasi *Android*

No	Tombol	Fungsi
1	Panah atas kiri	Menggerakkan lengan kiri maju
2	Panah bawah kiri	Menggerakkan lengan kiri mundur
3	Tombol <i>hook</i> kiri	(tekan sekali) membuka <i>hook</i> kiri
		(tekan tahan) menutup <i>hook</i> kiri
4	<i>Text box</i>	Tempat memasukkan <i>IP Address ESP32</i>
5	Panah atas kanan	Menggerakkan lengan kanan maju
6	Panah bawah kanan	Menggerakkan lengan kanan mundur
7	Tombol <i>hook</i> kanan	(tekan sekali) membuka <i>hook</i> kanan
		(tekan tahan) menutup <i>hook</i> kanan

Ketika tombol pada Tabel 4-1 ditekan sesuai dengan nomor urutan, maka *ESP32* akan menerima data alamat server yang dikirimkan oleh *Android*. Untuk mengetahui apakah *ESP32* benar menerima data yang dikirim oleh *Android* maka dilakukan pengujian sederhana dengan menampilkan alamat yang dikirimkan *Android* pada serial monitor menggunakan *software Arduino IDE*. Hasil yang ditampilkan serial monitor dapat dilihat pada Gambar 4-2.



Gambar 4-2 Pembacaan alamat server yang diterima *ESP32* dari *Android* menggunakan serial monitor pada *Arduino IDE*

Berdasarkan Gambar 4-2, alamat server yang diterima oleh *ESP32* dari *Android* telah sesuai dengan skema alur kerja *ESP32* pada Gambar 3-17. Oleh karena itu maka, proses transfer data antara *Android* dan *ESP32* dianggap berhasil.

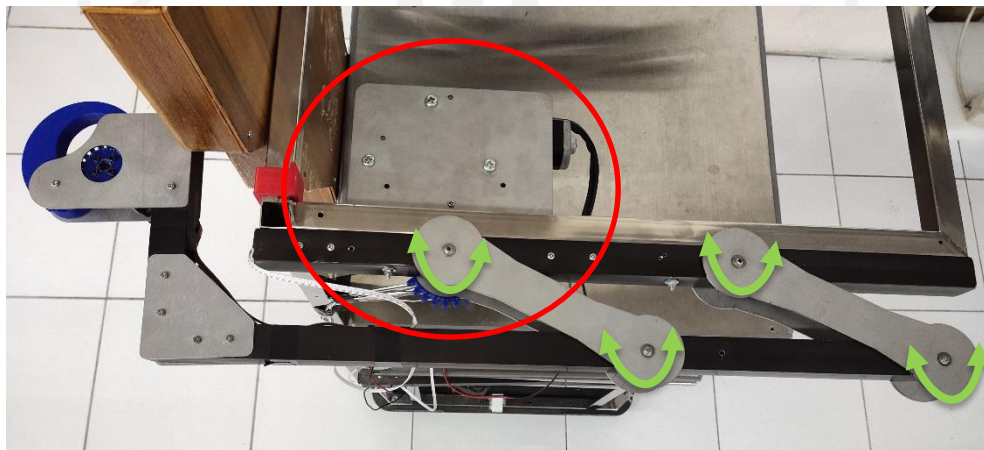
3. Perangkat keras sistem kendali mekanisme pembuka pintu



Gambar 4-3 Lengan Robot yang akan dikendalikan

Gambar 4-3 menunjukkan mekanisme pembuka pintu yang terdiri dari 2 lengan yang di letakkan pada sisi kanan dan sisi kiri robot. Mekanisme dibuat menjadi 2 (kanan dan kiri) berfungsi untuk menyesuaikan gagang pintu yang akan dibuka, jika gagang pintu yang akan dibuka berada di sisi kiri pintu maka lengan kanan yang akan digunakan untuk membuka pintu dan begitu juga sebaliknya, jika gagang pintu berada di sisi kanan pintu maka lengan kiri yang akan digunakan untuk membuka pintu. Berikut adalah bagian-bagian pada sistem kendali mekanisme pembuka pintu :

a. Motor *Power Window Universal*



Gambar 4-4 Posisi peletakan motor *power window universal*

Motor *Power Window Universal* berfungsi sebagai penggerak lengan mekanisme untuk maju ataupun mundur ketika robot membuka pintu, motor *Power Window Universal* diletakkan di bawah rangka robot seperti yang terlihat pada

Gambar 4-4. Pada *bracket* tempat motor melekat dilengkapi dengan *gear ratio* 8:1 untuk meningkatkan torsi yang dihasilkan sehingga dapat menarik pintu dengan sempurna, untuk mengendalikan motor digunakan motor *driver* BTS 7960 yang terhubung dengan *ESP32* sebagai pusat kendali yang mengatur gerakan motor.

b. Motor Servo MG 996 R



Gambar 4-5 Posisi peletakan motor servo

Motor Servo MG 996R berfungsi sebagai penggerak *hook* untuk “memegang” gagang pintu sehingga pintu dapat ditarik oleh mekanisme, Motor Servo MG 996R dihubungkan dengan *hook* menggunakan *gear* 1:1. Motor Servo MG 996R diletakkan pada bagian depan lengan tepat di belakang *hook* seperti pada Gambar 4-5, motor tersebut terhubung dengan *ESP32* sebagai pengendalinya.

c. *Limit switch*



Gambar 4-6 Posisi peletakan *limit switch*

Selain aktuator, pada sistem ini juga terdapat sensor untuk membatasi gerakan lengan agar tidak berlebihan dan merusak komponen lainnya, sensor tersebut adalah *limit switch*. *Limit switch* diletakkan pada bagian bawah lengan seperti yang terlihat pada Gambar 4-6, di mana terdapat 2 *limit switch* untuk 1 lengan yang mana

satu untuk sensor batas gerakan lengan maju dan yang satunya lagi untuk membatasi gerakan lengan mundur.

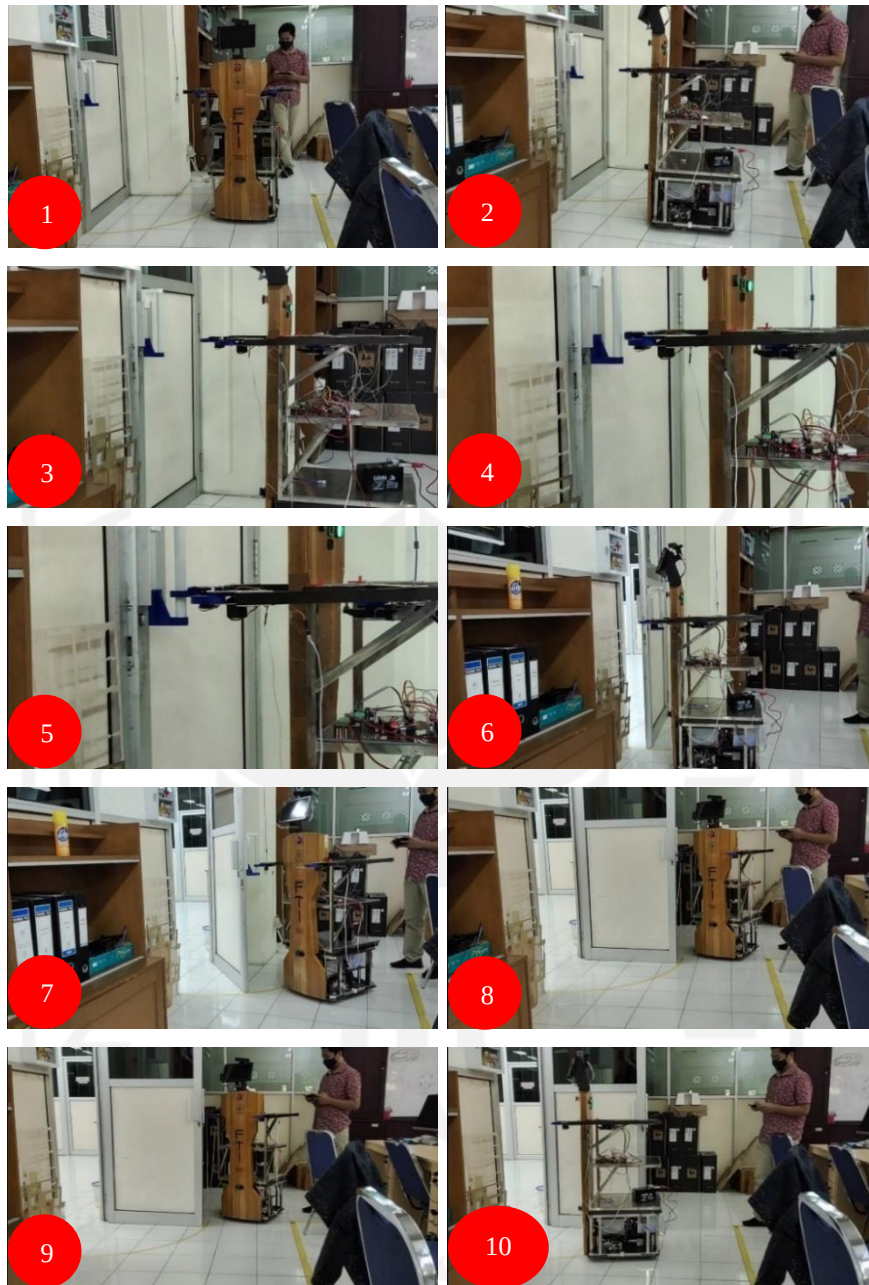
d. Rangkaian sistem kendali



Gambar 4-7 Rangkaian Sistem Kendali

Tanpa sistem kendali, aktuator dan sensor yang telah terpasang pada mekanisme pembuka pintu, tidak dapat langsung digunakan untuk membuka pintu, perlu adanya sistem kendali yang dapat mengendalikan aktuator dan sensor sehingga dapat digunakan untuk membuka pintu, oleh karena itu dibuatlah suatu rangkaian sistem kendali yang terdiri dari *ESP32* dan motor *driver* *BTS 9760*. *ESP32* berperan sebagai penerima sekaligus pengelola data yang dikirimkan oleh aplikasi remot pada *Android* di mana data yang diterima akan digunakan sebagai parameter untuk menggerakkan motor servo *MG 996R* ataupun mengaktifkan motor *driver* yang akan mengendalikan motor *power window universal*. Rangkaian sistem kendali dapat dilihat pada Gambar 3-13 dan hasilnya dapat dilihat pada Gambar 4-7.

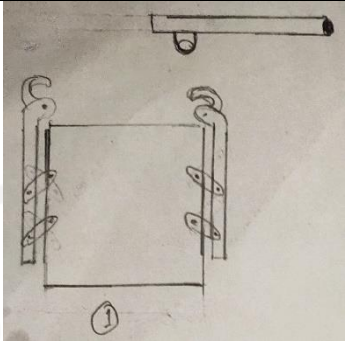
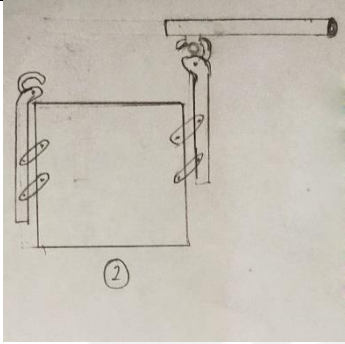
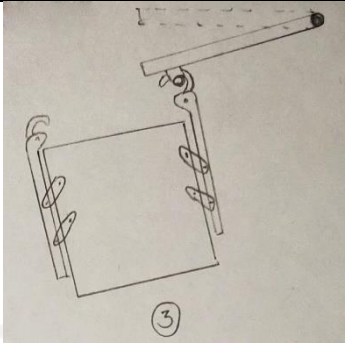
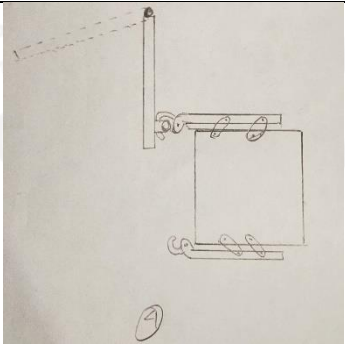
4. Simulasi robot membuka pintu

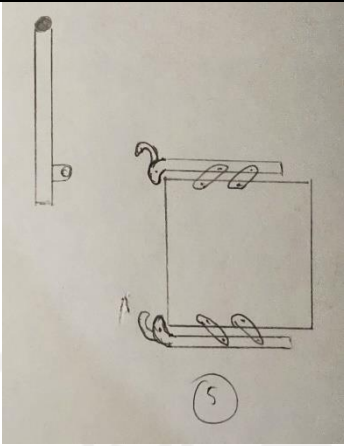
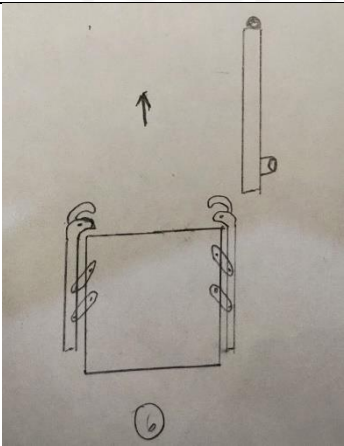


Gambar 4-8 Percobaan mengendalikan lengan robot untuk membuka pintu

Berdasarkan Gambar 4-8 simulasi membuka pintu menggunakan mekanisme pembuka pintu pada robot telah berhasil dilakukan, namun gerakan robot untuk membuka pintu menjadi sangat berbeda dengan kriteria perancangan pada Tabel 3-1, hal ini dikarenakan keterbatasan gerakan robot dan juga tingkat kesulitan pengendalian robot untuk melakukan manuver seperti yang diilustrasikan. Perubahan alur gerakan robot dalam membuka pintu dapat dilihat pada Tabel 4-2 berikut :

Tabel 4-2 Perubahan alur gerakan robot ketika membuka pintu

Langkah ke-	Ilustrasi	Keterangan
1		Pertama robot diposisikan menghadap pintu dan berada kurang lebih 30 cm di depan pintu dengan sisi kanan robot sejajar dengan gagang pintu yang akan dibuka.
2		Langkah kedua yaitu mekanisme pembuka pintu robot digerakkan maju sehingga dapat “memegang” gagang pintu.
3		Setelah <i>hook</i> berhasil “memegang” gagang pintu, mekanisme pembuka pintu digerakkan mundur kembali sehingga pintu terbuka $\pm 20^\circ$.
4		Langkah selanjutnya yaitu, robot bergerak mundur sehingga pintu tertarik hingga terbuka sepenuhnya.

5		<p>Lalu, mekanisme pembuka pintu melepaskan gagang pintu dan bergerak mundur menjauhi pintu.</p>
6		<p>Terakhir, robot digerakkan untuk memasuki ruangan.</p>

4.2 Hasil Pengujian

Setelah hasil rancangan dibuat menjadi sebuah *prototype*, maka dilakukan pengujian robot asisten medis untuk membuka pintu secara langsung. Terdapat 4 jenis pengujian yaitu pengujian membuka pintu, pengujian waktu yang dibutuhkan robot untuk membuka pintu hingga masuk ke dalam ruangan, pengujian menutup pintu dan pengujian waktu yang dibutuhkan robot untuk menutup pintu, selain itu pengujian juga terbagi menjadi 2 kategori yaitu, pengujian membuka pintu dengan operator melihat secara langsung robot asisten medis dan pengujian membuka pintu dengan operator hanya melihat visual kamera robot dari jarak jauh. Berikut adalah hasil pengujian yang telah dilakukan :

4.2.1 Pengujian membuka pintu

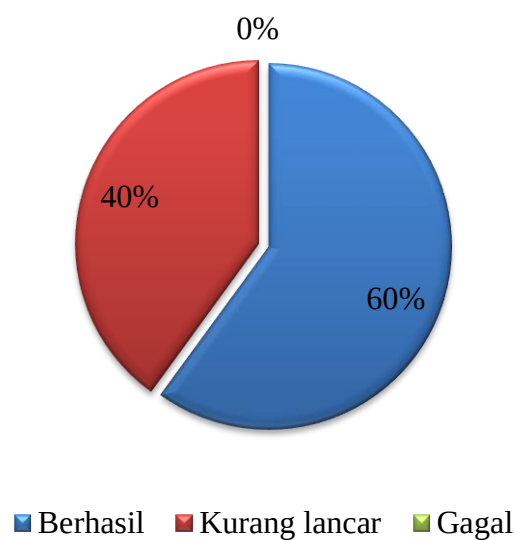
Pada pengujian ini diambil data tingkat keberhasilan robot ketika simulasi membuka pintu, terdapat 3 parameter yang digunakan pada pengambilan data pengujian robot membuka pintu yaitu berhasil, kurang lancar, dan gagal. Robot dinilai berhasil jika pintu berhasil dibuka dengan sempurna tanpa mengalami kendala, dinilai kurang lancar jika pintu berhasil terbuka namun, pada prosesnya robot mengalami kendala dan terakhir robot dinilai gagal jika robot tidak berhasil untuk membuka pintu. Pengambilan data dilakukan secara langsung ketika proses simulasi robot membuka pintu dengan 10 kali percobaan, berikut adalah hasil pengujian yang telah dilakukan :

- a. Data pengujian robot membuka pintu dengan operator melihat secara langsung

Tabel 4-3 Data pengujian membuka pintu secara langsung

Pengujian ke-	Berhasil	Kurang Lancar	Gagal
1	-	✓	-
2	-	✓	-
3	-	✓	-
4	-	✓	-
5	✓	-	-
6	✓	-	-
7	✓	-	-
8	✓	-	-
9	✓	-	-
10	✓	-	-
Total	6	4	0

Berdasarkan Tabel 4-3 terdapat 10 data pengujian robot membuka pintu yang mana hasilnya 6 kali berhasil, 4 kali kurang lancar dan tidak ada yang gagal, jika data tersebut dibuat dalam bentuk diagram maka akan terlihat seperti pada Gambar 4-9 berikut :



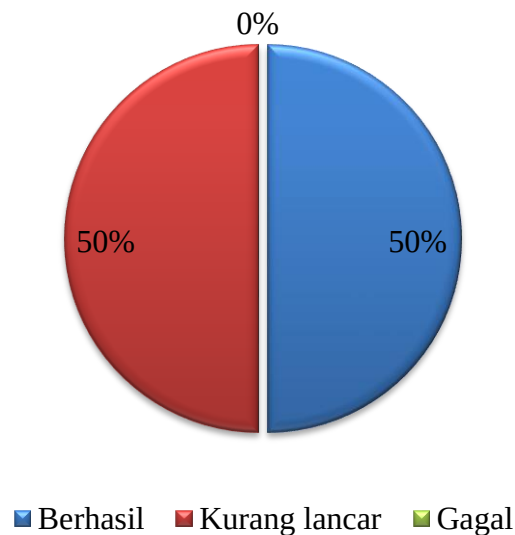
Gambar 4-9 Diagram percobaan robot membuka pintu secara langsung

- b. Data pengujian robot membuka pintu dengan operator melihat visual kamera robot dari jarak jauh

Tabel 4-4 Data pengujian membuka pintu dari jarak jauh

Pengujian ke-	Berhasil	Kurang Lancar	Gagal
1	-	✓	-
2	✓	-	-
3	✓	-	-
4	-	✓	-
5	✓	-	-
6	-	✓	-
7	✓	-	-
8	✓		-
9	-	✓	-
10	-	✓	-
Total	5	5	0

Berdasarkan Tabel 4-4 dari 10 percobaan yang dilakukan robot berhasil membuka pintu dengan lancar sebanyak 5 kali, kurang lancar 5 kali dan tidak ada yang gagal, dalam bentuk diagram data tersebut dapat dilihat pada Gambar 4-10 berikut :



Gambar 4-10 Diagram percobaan robot membuka pintu dari jarak jauh

4.2.2 Pengujian waktu yang diperlukan robot untuk membuka pintu hingga masuk ke dalam ruangan

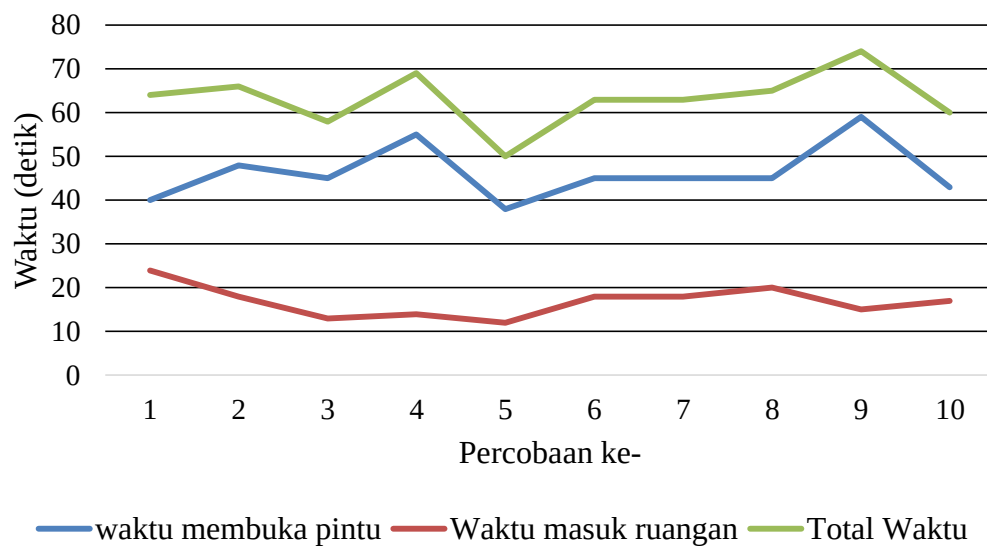
Selain melakukan pengujian tingkat keberhasilan robot membuka pintu, juga dilakukan pengujian waktu yang dibutuhkan robot untuk membuka pintu hingga masuk ke dalam ruangan. Pengambilan data pada pengujian ini dilakukan dengan cara merekam simulasi robot membuka pintu hingga ia masuk ke dalam ruangan, dari video rekaman dilihatlah waktu yang robot membuka pintu hingga masuk ke dalam ruangan. Berikut adalah hasil pengujian yang telah dilakukan :

- a. Data pengujian dengan operator melihat robot secara langsung

Tabel 4-5 Data pengujian waktu yang dibutuhkan robot untuk membuka pintu hingga masuk ke dalam ruangan secara langsung

Pengujian ke-	Waktu membuka pintu (detik)	Waktu masuk ruangan (detik)	Total (detik)
1	40	24	64
2	48	18	66
3	45	13	58
4	55	14	69
5	38	12	50
6	45	18	63
7	45	18	63
8	45	20	65
9	59	15	74
10	43	17	60
Rata-rata	46,3	16,9	63,2

Pada Tabel 4-5 disajikan data hasil pengujian waktu yang dibutuhkan robot untuk membuka pintu, visualisasi data tersebut dalam bentuk grafik dapat dilihat pada Gambar 4-11 berikut :



Gambar 4-11 Grafik waktu yang dibutuhkan robot untuk membuka pintu dan masuk ke dalam ruangan secara langsung

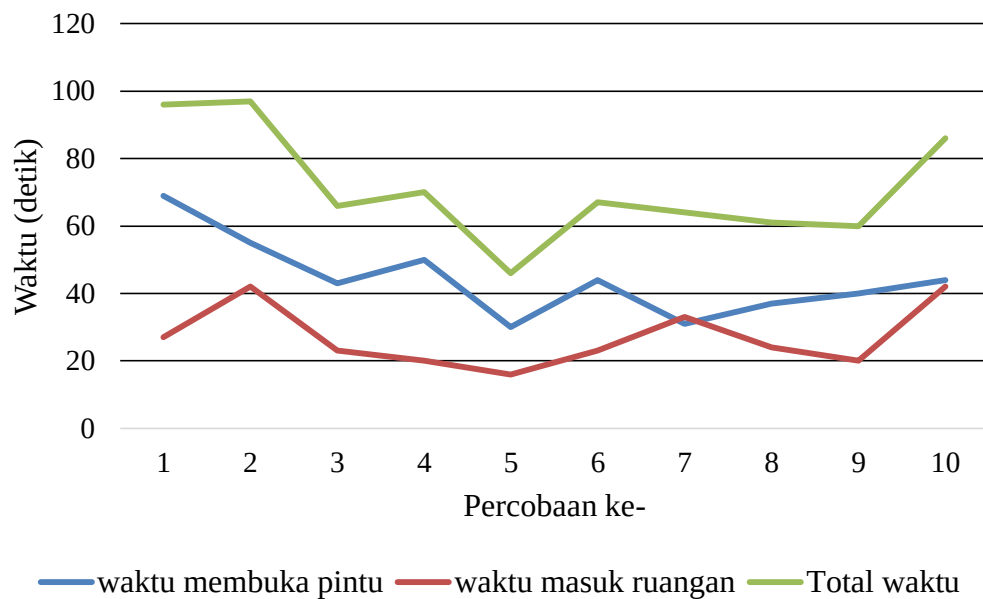
- b. Data pengujian dengan operator melihat visualisasi kamera robot dari jarak jauh

Tabel 4-6 Data pengujian waktu yang dibutuhkan robot untuk membuka pintu hingga masuk ke dalam ruangan dari jarak jauh

Pengujian ke-	Waktu membuka pintu (detik)	Waktu masuk ruangan (detik)	Total (detik)
1	69	27	96
2	55	42	97
3	43	23	66
4	50	20	70
5	30	16	46
6	44	23	67
7	31	33	64
8	37	24	61
9	40	20	60
10	44	42	86
Rata-rata	44,3	27	71,3

Berdasarkan data pada Tabel 4-6 yang telah tersaji, dibuatlah grafik yang dapat mewakili data hasil pengujian yang telah dilakukan. Grafik data pengujian waktu

yang dibutuhkan robot untuk membuka pintu hingga masuk ke dalam ruangan dari jarak jauh dapat dilihat pada Gambar 4-12 berikut :



Gambar 4-12 Grafik waktu yang dibutuhkan robot untuk membuka pintu dan masuk ke dalam ruangan dari jarak jauh

4.2.3 Pengujian menutup pintu

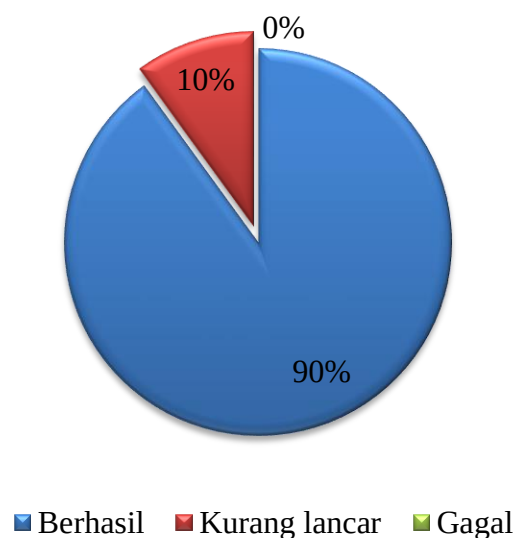
Pengujian menutup pintu dilakukan dengan menilai langsung proses robot menutup pintu ketika simulasi berlangsung. Penilaian dilakukan berdasarkan 3 parameter yaitu berhasil, kurang lancar, dan gagal. Robot dinilai berhasil jika robot berhasil menutup pintu tanpa kendala, dinilai kurang lancar jika robot berhasil menutup pintu, namun mengalami kendala dalam prosesnya dan robot dinilai gagal jika robot tidak berhasil untuk menutup pintu. Pengambilan data dilakukan sebanyak 10 kali percobaan, berikut adalah data hasil pengujian yang dilakukan :

- a. Data pengujian robot menutup pintu dengan operator melihat robot secara langsung

Tabel 4-7 Data pengujian menutup pintu secara langsung

Pengujian ke-	Berhasil	Kurang Lancar	Gagal
1	✓	-	-
2	✓	-	-
3	✓	-	-
4	✓	-	-
5	✓	-	-
6	-	✓	-
7	✓	-	-
8	✓	-	-
9	✓	-	-
10	✓	-	-
Total	9	1	0

Tabel 4-7 merupakan data hasil pengujian robot menutup pintu, terlihat bahwa dalam 10 kali percobaan 9 kali berhasil, 1 kali kurang lancar dan tidak ada yang gagal, data yang sudah di konversi dalam bentuk diagram dapat dilihat pada Gambar 4-13 berikut :



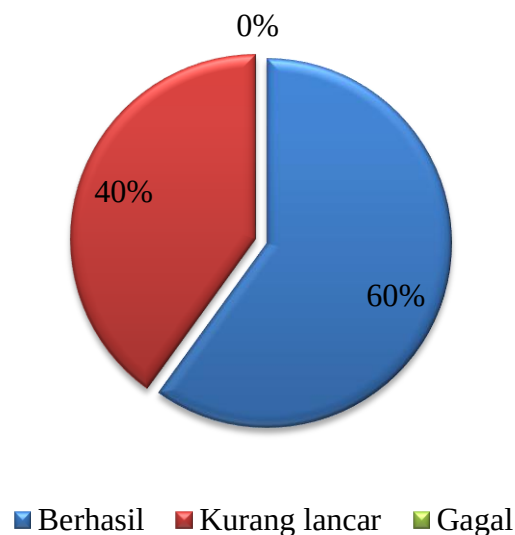
Gambar 4-13 Diagram percobaan robot menutup pintu secara langsung

- b. Data pengujian robot menutup pintu dengan operator melihat visual kamera robot dari jarak jauh

Tabel 4-8 Data pengujian menutup pintu dari jarak jauh

Pengujian ke-	Berhasil	Kurang Lancar	Gagal
1	-	✓	-
2	✓	-	-
3	-	✓	-
4	✓	-	-
5	-	✓	-
6	✓	-	-
7	✓	-	-
8	✓	-	-
9	✓	-	-
10	-	✓	-
Total	6	4	0

Tabel 4-8 merupakan hasil dari pengujian robot menutup pintu di mana operator mengendalikan jarak jauh dan hanya melihat visual kamera dari robot, terlihat bahwa dari 10 kali percobaan, 6 kali berhasil, 4 kali kurang lancar dan tidak ada yang gagal. Jika data tersebut divisualisasikan dalam bentuk diagram maka akan terlihat seperti pada Gambar 4-14 berikut :



Gambar 4-14 Diagram percobaan robot menutup pintu dari jarak jauh

4.2.4 Pengujian waktu yang dibutuhkan robot untuk menutup pintu

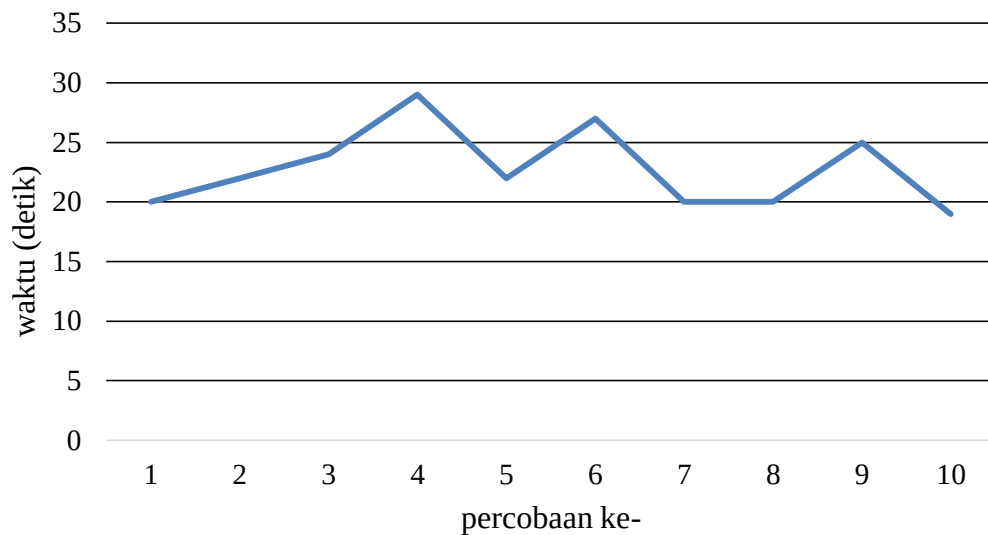
Pengujian waktu yang dibutuhkan robot untuk menutup pintu dilakukan dengan cara merekam simulasi robot menutup pintu, yang kemudian dari video rekaman terlihat waktu yang dibutuhkan robot untuk menutup pintu. Pengambilan data dilakukan sebanyak 10 kali percobaan, berikut adalah hasil pengujian yang dilakukan :

- a. Data pengujian dengan operator melihat robot secara langsung

Tabel 4-9 Data pengujian waktu yang dibutuhkan robot untuk menutup pintu secara langsung

Pengujian ke-	Waktu menutup pintu (detik)
1	20
2	22
3	24
4	29
5	22
6	27
7	20
8	20
9	25
10	19
Rata-rata	22,8

Pada Tabel 4-9 tersaji data waktu yang dibutuhkan robot untuk menutup pintu dari 10 kali pengambilan data yang dilakukan, jika ditampilkan menggunakan grafik maka akan terlihat seperti pada Gambar 4-15 berikut :



Gambar 4-15 Grafik waktu yang dibutuhkan robot untuk menutup pintu secara langsung

b. Data pengujian dengan operator melihat visual kamera robot dari jarak jauh

Tabel 4-10 Data pengujian waktu yang dibutuhkan robot untuk menutup pintu dari jarak jauh

Pengujian ke-	Waktu menutup pintu (detik)
1	29
2	19
3	25
4	22
5	30
6	18
7	21
8	20
9	22
10	27
Rata-rata	23,3

Tabel 4-10 merupakan data hasil pengujian waktu yang diperlukan robot untuk membuka pintu di mana operator tidak melihat robot secara langsung melainkan melihat visual kamera robot dari jarak jauh. Data yang telah tersaji dibuat dalam bentuk grafik untuk memudahkan dalam menganalisis, grafik data pengujian waktu

yang dibutuhkan robot untuk menutup pintu dari jarak jauh dapat dilihat pada Gambar 4-16 berikut :



Gambar 4-16 Grafik waktu yang dibutuhkan robot untuk menutup pintu dari jarak jauh

4.2.5 Pengujian dengan operator yang belum terlatih

Pengujian yang dilakukan sebelumnya, dilakukan dengan operator yang sudah melakukan pelatihan mengendalikan robot selama kurang lebih 1 Minggu, untuk menguji pengaruh operator yang mengendalikan robot asisten medis, maka dilakukan juga pengujian dengan operator yang belum melakukan pelatihan mengendalikan robot asisten medis. Pengujian dilakukan dari jarak jauh dengan memanfaatkan visual kamera dari robot asisten medis. Pengujian dilakukan dengan 2 operator yang belum melakukan pelatihan mengendalikan robot asisten medis, di mana masing-masing operator melakukan 5 kali percobaan. Berikut hasil pengujian yang dilakukan :

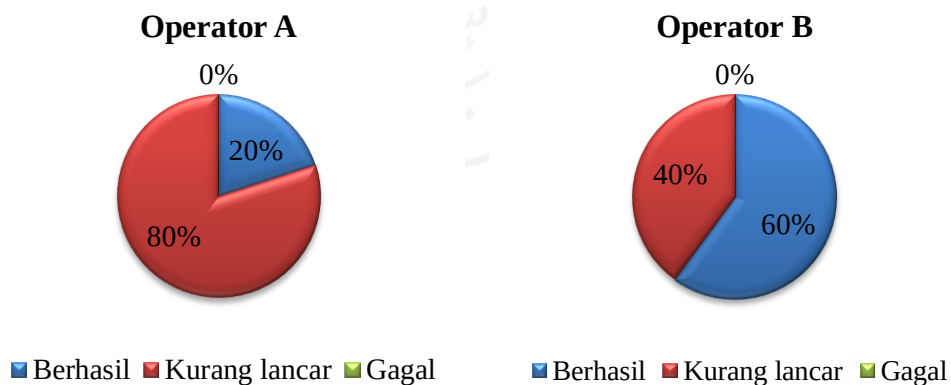
a. Pengujian membuka pintu

Pengujian membuka pintu dilakukan dengan cara mengamati secara langsung proses robot membuka pintu yang dilakukan oleh operator A dan operator B. Penilaian dilakukan menggunakan kriteria penilaian pada pengujian sebelumnya yaitu berhasil, kurang lancar dan gagal, berikut adalah data hasil pengujian yang dilakukan :

Tabel 4-11 Data pengujian robot membuka pintu yang dikendalikan operator A dan operator B

Pengujian ke-	Berhasil	Kurang Lancar	Gagal
Operator A			
1	-	✓	-
2	-	✓	-
3	-	✓	-
4	✓	-	-
5	-	✓	-
Total	1	4	0
Operator B			
1	-	✓	-
2	-	✓	-
3	✓	-	-
4	✓	-	-
5	✓	-	-
Total	3	2	0

Data pada Tabel 4-11 adalah hasil pengujian robot untuk membuka pintu yang dikendalikan oleh operator A dan operator B dengan masing-masing operator melakukan percobaan sebanyak 5 kali. Berdasarkan data yang diperoleh operator A berhasil membuka pintu dengan lancar sebanyak 1 kali, kurang lancar 4 kali dan tidak ada yang gagal, sedangkan operator B berhasil membuka pintu dengan lancar sebanyak 3 kali, kurang lancar 2 kali dan tidak ada yang gagal, untuk melihat persentase keberhasilan dari kedua operator dapat dilihat pada Gambar 4-17 berikut :



Gambar 4-17 Persentase keberhasilan robot membuka pintu yang dikendalikan operator A dan operator B

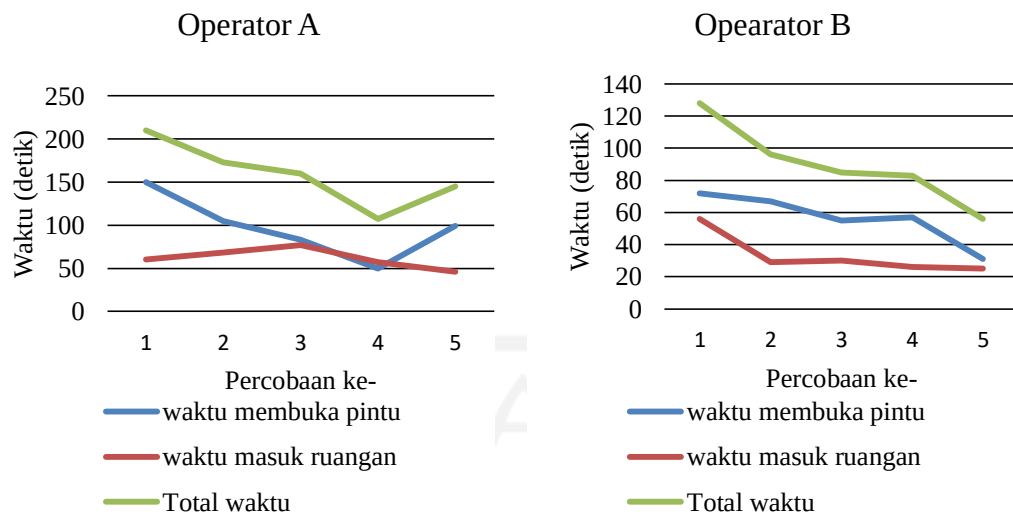
- b. Pengujian waktu yang dibutuhkan untuk membuka pintu hingga masuk ke dalam ruangan

Pengujian waktu yang dibutuhkan robot untuk membuka pintu hingga masuk ke dalam ruangan dilakukan dengan cara merekam proses simulasi robot membuka pintu yang dilakukan oleh operator A dan operator B, yang kemudian dari video rekaman dianalisa waktu yang dibutuhkan robot untuk membuka pintu hingga masuk ke dalam ruangan, berikut adalah data hasil pengujian yang dilakukan :

Tabel 4-12 Data pengujian waktu yang dibutuhkan robot untuk membuka pintu hingga masuk ke dalam ruangan yang dikendalikan oleh operator A dan B

Pengujian ke-	Waktu membuka pintu (detik)	Waktu masuk ruangan (detik)	Total (detik)
Operator A			
1	150	60	210
2	105	68	173
3	83	77	160
4	50	57	107
5	99	46	145
Rata-rata	97,4	61,6	159
Operator B			
1	72	56	128
2	67	29	96
3	55	30	85
4	57	26	83
5	31	25	56
Rata-rata	56,4	33,2	89,6

Berdasarkan Tabel 4-12 operator A memerlukan waktu rata-rata 97,4 detik untuk mengendalikan robot membuka pintu dan rata-rata 61,6 detik untuk mengendalikan robot masuk ke dalam ruangan sehingga total waktu rata-rata yang dibutuhkan operator A untuk mengendalikan robot membuka pintu hingga masuk ke dalam ruangan adalah 159 detik, sedangkan operator B memerlukan waktu rata-rata 56,4 detik untuk membuka pintu dan 33,2 detik untuk masuk ke dalam ruangan sehingga total waktu yang dibutuhkan operator B adalah 89,6 detik, untuk lebih jelasnya dapat dilihat pada grafik pada Gambar 4-18 berikut :



Gambar 4-18 Grafik data waktu yang dibutuhkan operator A dan B untuk mengendalikan robot membuka pintu hingga masuk ke dalam ruangan

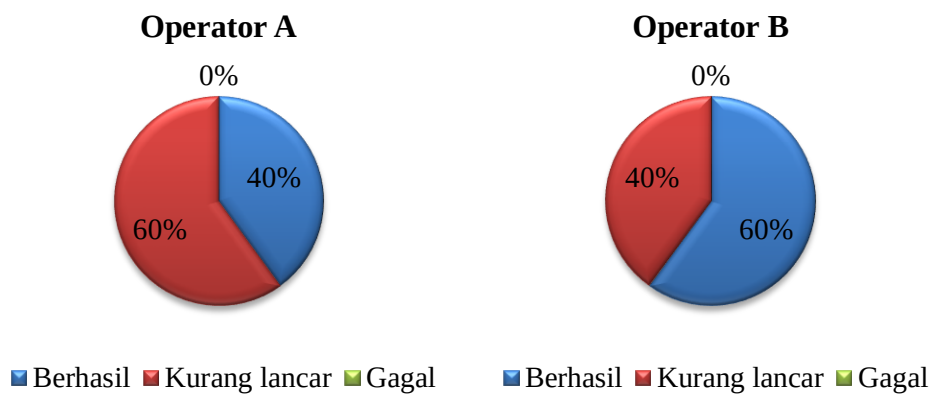
c. Pengujian menutup pintu

Pengujian menutup pintu dilakukan dengan cara pengamatan langsung ketika proses simulasi menutup pintu yang dilakukan oleh operator A dan operator B, di mana kriteria penilaian yang diambil juga sama yaitu berhasil, kurang lancar dan gagal. Berikut adalah hasil pengujian yang dilakukan :

Tabel 4-13 Data pengujian robot menutup pintu yang dikendalikan operator A dan operator B

Pengujian ke-	Berhasil	Kurang Lancar	Gagal
Operator A			
1	-	✓	-
2	✓	-	-
3	✓	-	-
4	-	✓	-
5	-	✓	-
Total	2	3	0
Operator B			
1	-	✓	-
2	✓	-	-
3	✓	-	-
4	✓	-	-
5	-	✓	-
Total	3	2	0

Data pada Tabel 4-13 merupakan data pengujian keberhasilan robot menutup pintu yang dilakukan oleh operator A dan operator B, dari data tersebut terlihat bahwa dari 5 kali percobaan operator A berhasil menutup pintu dengan lancar sebanyak 2 kali dan berhasil dengan kurang lancar 3 kali serta tidak ada yang gagal, sedangkan operator B berhasil menutup pintu dengan lancar sebanyak 3 kali, kurang lancar 2 kali dan tidak ada yang gagal. Persentase dari data pengujian tersebut dapat dilihat pada Gambar 4-19 berikut :



Gambar 4-19 Persentase keberhasilan robot menutup pintu yang dikendalikan operator A dan operator B

d. Pengujian waktu yang dibutuhkan robot untuk menutup pintu

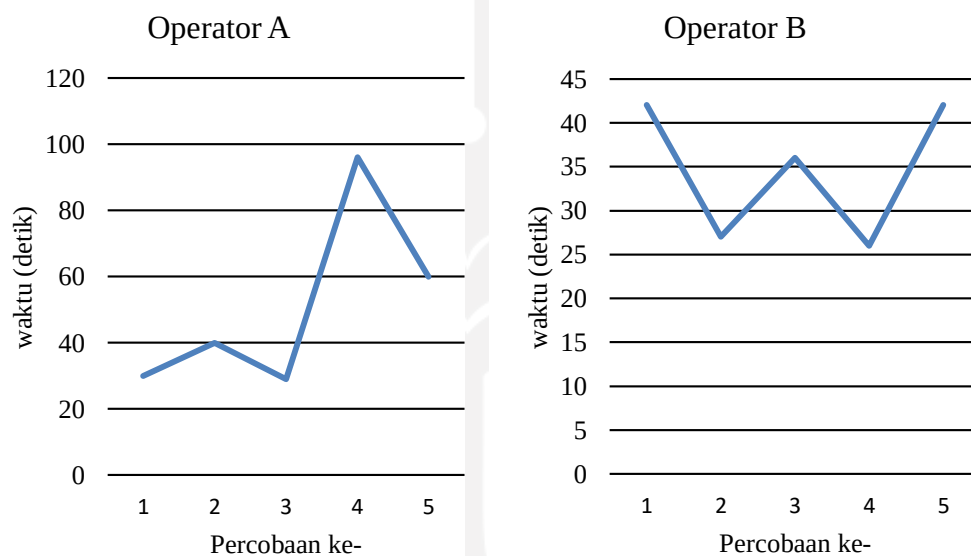
Pengujian waktu yang dibutuhkan robot untuk menutup pintu dilakukan dengan cara merekam proses simulasi robot menutup pintu yang dilakukan oleh operator A dan operator B. Hasil dari video rekaman kemudian dianalisa untuk mendapatkan data waktu yang dibutuhkan robot untuk menutup pintu, berikut adalah data hasil pengujian yang telah dilakukan :

Tabel 4-14 Data pengujian waktu yang dibutuhkan robot untuk menutup pintu yang dikendalikan oleh operator A dan B

Pengujian ke-	Waktu menutup pintu (detik)
Operator A	
1	30
2	40
3	29
4	96
5	60
Rata-rata	51

Operator B	
1	42
2	27
3	36
4	26
5	42
Rata-rata	34,6

Berdasarkan Tabel 4-14 operator A memerlukan waktu rata-rata untuk mengendalikan robot menutup pintu selama 51 detik sedangkan operator B hanya memerlukan waktu rata-rata 34,6 detik untuk mengendalikan robot menutup pintu, untuk lebih jelasnya dapat dilihat pada Gambar 4-20 berikut :



Gambar 4-20 Grafik data waktu yang dibutuhkan operator A dan B untuk mengendalikan robot menutup pintu

4.3 Analisis dan Pembahasan

4.3.1 Pengujian dengan operator melihat robot secara langsung

Pengujian membuka dan menutup pintu bertujuan untuk mengetahui persentase keberhasilan robot ketika proses membuka ataupun menutup pintu, data hasil dari pengujian yang dilakukan dapat dilihat pada Tabel 4-3 dan Tabel 4-7 atau jika dibuat dalam bentuk diagram maka akan terlihat seperti pada Gambar 4-9 dan Gambar 4-13. Pada pengujian ini operator yang mengendalikan robot

menggunakan *Android* berada di dekat robot agar dapat melihat keadaan sekitar robot secara langsung.

Berdasarkan Gambar 4-9 terlihat bahwa robot medis dapat membuka pintu 100% namun persentase robot berhasil membuka pintu dengan lancar sebesar 60% dan 40% lainnya robot dapat membuka pintu namun kurang lancar, kurang lancar yang dimaksudkan adalah ketika robot mengalami masalah dalam proses membuka pintu seperti roda robot terangkat, roda robot tergelincir, ataupun masalah transfer data antara remot dan robot yang terkadang mengalami *delay* ketika proses membuka pintu.



Gambar 4-21 Masalah roda robot terangkat ketika proses membuka pintu

Permasalahan yang cukup fatal dari beberapa masalah yang ditemui adalah terangkatnya roda pada robot (dapat dilihat pada Gambar 4-21) yang mana dapat membuat robot kehilangan keseimbangan dan berpotensi jatuh. Selain itu, jika pada saat robot mengantarkan makanan atau obat atau barang lainnya tentu saja barang yang diletakkan akan jatuh karena posisi robot menjadi miring. Terdapat beberapa alternatif solusi untuk mengatasi masalah tersebut yaitu, mengubah struktur rangka sehingga pembebanan dapat lebih seimbang, mengganti konfigurasi roda menjadi sistem 4 roda dari yang sebelumnya menggunakan 3 roda untuk membuat pergerakan robot menjadi lebih stabil dan menggeser komponen yang berat (seperti baterai) ke belakang struktur sehingga beban pada robot dapat lebih seimbang serta untuk masalah roda tergelincir bisa diatasi dengan mengganti jenis roda yang digunakan dengan roda yang memiliki gaya gesek yang lebih besar.

Persentase keberhasilan robot dalam menutup pintu berdasarkan data yang telah diambil adalah 100% dengan tingkat kelancaran 90% dan tidak lancar 10%,

namun hal ini bukanlah masalah besar karena penyebab yang membuat robot tidak lancar dalam menutup pintu yaitu kesalahan operator dalam menggerakkan robot sehingga lengan robot tersangkut pada gagang pintu, dengan kata lain masalah ini bukanlah masalah yang sering muncul dan dapat diminimalisir dengan melatih operator dalam mengendalikan robot.

Selain untuk mengetahui tingkat keberhasilan robot, juga diambil data waktu yang dibutuhkan robot untuk membuka pintu hingga masuk ke dalam ruangan dan waktu yang diperlukan robot untuk menutup pintu. Data pengujian yang dilakukan dapat dilihat pada Tabel 4-5 dan Tabel 4-9, jika divisualkan dalam bentuk grafik maka akan terlihat seperti pada Gambar 4-11 dan Gambar 4-15.

Berdasarkan grafik pada Gambar 4-11, waktu yang dibutuhkan robot untuk membuka pintu berkisar di antara 40 sampai 50 detik dengan rata-rata 46,3 detik. Selain itu terdapat 2 data menyimpang di mana robot membutuhkan waktu 55 dan 59 detik untuk membuka pintu, hal ini dapat terjadi karena operator kurang terlatih dalam mengendalikan robot dan roda robot yang sesekali tergelincir. Waktu yang dibutuhkan robot untuk masuk ke dalam ruangan berkisar di antara 10 hingga 20 detik dengan rata-rata 16,9 detik. Waktu rata-rata total yang dibutuhkan robot untuk membuka pintu hingga masuk ke dalam ruangan yaitu 63,2 detik.

Data pengujian robot menutup pintu yang dihasilkan cukup bervariasi, hal ini dapat dilihat pada Gambar 4-15. Lamanya waktu yang dibutuhkan yaitu antara 20 detik hingga 30 detik dengan rata-rata 22,8 detik. Lamanya waktu yang dibutuhkan robot sangat dipengaruhi oleh kemampuan operator dalam mengendalikan robot, oleh karena itu data yang dihasilkan saat pengujian menjadi cukup bervariasi.

4.3.2 Pengujian dengan operator melihat visual kamera robot dari jarak jauh

Selain pengujian dilakukan dengan operator berada di dekat robot, juga dilakukan pengujian di mana operator mengendalikan robot dari jarak jauh dan hanya mengandalkan visual kamera dari robot itu sendiri, hal ini dilakukan untuk memperoleh data yang dapat mewakili keadaan yang sebenarnya ketika robot akan digunakan di rumah sakit di mana operator akan mengendalikan robot dari jarak

jauh. Pada awalnya pengujian ini tidak dapat dilakukan karena terdapat masalah yang cukup fatal yaitu posisi kamera pada robot terlalu tinggi sehingga mekanisme pembuka pintu tidak masuk tertangkap kamera seperti pada Gambar 4-22.



Gambar 4-22 Posisi awal kamera

Posisi kamera robot yang terlihat pada Gambar 4-22 menyebabkan mekanisme pembuka pintu tidak terlihat di visual kamera yang ditampilkan aplikasi remot pada *Android* walaupun mekanisme sudah di posisi panjang maksimal, hal ini membuat pengujian membuka dan menutup pintu tidak mungkin dilakukan, oleh sebab itu dilakukan modifikasi untuk mengubah posisi kamera agar mekanisme dapat terlihat dari *Android*, hasil modifikasi dapat dilihat pada Gambar 4-23.



Gambar 4-23 Posisi kamera setelah dilakukan modifikasi

Modifikasi posisi kamera pada Gambar 4-23 dilakukan untuk membuat kamera menangkap video lebih ke bawah, sehingga mekanisme dapat terlihat di *Android*. Hasil modifikasi terbilang berhasil di mana pada kondisi mekanisme dipanjangkan sampai batas maksimal, *hook* pada mekanisme dapat terlihat pada *Android*, hal ini sudah cukup untuk dapat melakukan pengujian membuka dan menutup pintu.

Data hasil pengujian tingkat keberhasilan membuka pintu dapat dilihat pada Tabel 4-4 dan Gambar 4-10. Terlihat bahwa persentase robot berhasil membuka pintu tetap 100% seperti pada pengujian secara langsung namun, tingkat keberhasilan robot membuka pintu dengan lancar menurun 10% dari pengujian sebelumnya menjadi 50% dan 50 % sisanya robot berhasil membuka pintu tetapi kurang lancar. Data pengujian menutup pintu dapat dilihat pada Tabel 4-8 atau diagram pada Gambar 4-14, dari data tersebut terlihat bahwa robot dapat menutup pintu dengan tingkat keberhasilan 100%, di mana 60% lancar dan 40% kurang lancar, robot mengalami penurunan kinerja dibanding dengan pengujian secara langsung yang mana tingkat keberhasilan menutup pintu dengan lancar mencapai 90%. Penurunan kinerja robot ketika membuka dan menutup pintu berkaitan dengan metode pengujian yang mengharuskan operator hanya melihat visual kamera pada robot untuk mengendalikan robot, hal ini membuat operator menjadi lebih sering membuat kesalahan dalam mengendalikan robot karena jarak pandang yang terbatas pada apa yang ditampilkan oleh kamera di *Android*.

Data pengujian waktu yang diperlukan robot untuk membuka pintu hingga masuk ke ruangan dapat dilihat pada Tabel 4-6, terlihat bahwa waktu rata-rata yang dibutuhkan oleh robot untuk membuka pintu adalah 44,3 detik dan waktu rata-rata yang dibutuhkan robot untuk masuk ke dalam ruangan adalah 27 detik sehingga total waktu rata-rata yang dibutuhkan robot adalah 71,3 detik, hal tersebut lebih lama 8,1 detik jika dibandingkan dengan pengujian operator melihat secara langsung. Data yang telah diambil juga dibuat dalam bentuk grafik yang terlihat pada Gambar 4-12, dari grafik tersebut terlihat bahwa fluktuasi data yang terjadi semakin besar dibandingkan dengan grafik data pengujian waktu yang dibutuhkan robot untuk membuka pintu hingga masuk ke dalam ruangan secara langsung, penyebab hal ini hampir sama dengan permasalahan yang telah disebutkan pada bab 4.3.1. di atas, namun pada pengujian ini muncul permasalahan baru yaitu robot

tersangkut pada pintu ketika akan melewati pintu untuk masuk ke dalam ruangan, yang dapat dilihat pada Gambar 4-24.



Gambar 4-24 Mekanisme pembuka pintu robot tersangkut pada kusen pintu

Permasalahan pada Gambar 4-24 dapat terjadi karena terbatasnya jarak pandangan yang dapat dilihat operator karena kamera pada robot hanya dapat melihat ke arah depan sedangkan pada sisi samping dan belakang tidak ada sehingga operator kesulitan untuk mengawasi keadaan sekeliling robot. Hal ini juga yang menyebabkan data yang dihasilkan menjadi sangat fluktuatif, karena waktu yang dibutuhkan robot menjadi bertambah untuk memperbaiki posisinya terlebih dahulu sebelum mencoba lagi masuk ke dalam ruangan. Untuk data pengujian waktu yang dibutuhkan robot untuk menutup pintu dapat dilihat pada Tabel 4-10, di mana pada data tersebut terlihat bahwa waktu rata-rata yang dibutuhkan robot untuk menutup pintu adalah 23,3 detik, hanya berbeda 0,5 detik dibandingkan dengan data pengujian operator melihat robot secara langsung. Grafik data waktu yang dibutuhkan robot untuk menutup pintu dapat dilihat pada Gambar 4-16, terlihat bahwa fluktuasi yang terjadi tidak jauh berbeda dengan grafik pada pengujian dengan operator melihat robot secara langsung, data yang mirip ini terjadi karena gerakan robot untuk menutup pintu cukup sederhana sehingga jarak pandang yang terbatas dari visual kamera pada *Android*, tidak berpengaruh besar pada kinerja robot dalam proses menutup pintu.

4.3.3 Pengujian dengan operator yang belum terlatih

Pengujian sebelumnya dilakukan dengan operator yang sudah melakukan pelatihan mengendalikan robot selama kurang lebih 1 Minggu, untuk menguji pengaruh operator terhadap performa robot asisten medis ketika digunakan untuk membuka dan menutup pintu, maka dilakukan pengujian dengan operator yang belum melakukan pelatihan mengendalikan robot asisten medis. Hasil pengujian keberhasilan membuka pintu dapat dilihat pada Tabel 4-11 atau dapat dilihat pada Gambar 4-17, dari data yang diperoleh terlihat bahwa dari masing-masing 5 kali percobaan, operator A dan B berhasil membuka pintu 100% dengan catatan operator A berhasil dengan lancar 20% dan 80% lainnya kurang lancar, sedangkan operator B berhasil dengan lancar sebesar 60% dan 40% kurang lancar.

Data waktu yang dibutuhkan operator A dan B untuk mengendalikan robot membuka pintu hingga masuk ke dalam ruangan dapat dilihat pada Tabel 4-12 atau grafik pada Gambar 4-18, dari data yang diperoleh terlihat waktu total rata-rata yang dibutuhkan operator A untuk mengendalikan robot membuka pintu hingga masuk ke dalam ruangan adalah 159 detik, sedangkan operator B hanya memerlukan waktu total rata-rata 89,6 detik, sebagai perbandingan operator yang telah melakukan pelatihan membutuhkan waktu sebanyak 71,3 detik untuk mengendalikan robot membuka pintu hingga masuk ke dalam ruangan. Perbedaan waktu yang dihasilkan oleh operator A, B dan operator yang sudah terlatih dapat terjadi dapat disebabkan oleh banyak faktor seperti pengalaman operator, cara berpikir operator dan lainnya, namun masalah ini dapat diatasi dengan memberikan pelatihan pada operator sehingga operator dapat lebih terbiasa untuk mengendalikan robot asisten medis, hal ini tergambar dari grafik pada Gambar 4-18 di mana waktu yang dihasilkan operator A dan B semakin lama semakin sedikit.

Hasil dari pengujian robot menutup pintu yang dikendalikan oleh operator A dan B dapat dilihat pada Tabel 4-13 atau Gambar 4-19, dari data yang diperoleh didapati baik operator A maupun operator B berhasil mengendalikan robot menutup pintu 100% dari 5 kali percobaan, dengan catatan operator A berhasil dengan lancar sebesar 40% dan kurang lancar 60% sedangkan operator B berhasil dengan lancar sebesar 60% dan kurang lancar 40%.

Data waktu yang dibutuhkan operator A dan operator B untuk menutup pintu dapat dilihat pada Tabel 4-14 dan lebih jelasnya dapat dilihat grafik pada Gambar 4-20, dari data yang diperoleh operator A membutuhkan waktu selama 51 detik untuk mengendalikan robot menutup pintu sedangkan operator B membutuhkan waktu selama 34,6 detik. Berdasarkan grafik pada Gambar 4-20 terlihat bahwa pada awalnya operator A mempunyai catatan waktu yang cukup baik, dan pada percobaan keempat dan kelima mengalami kesulitan sehingga membutuhkan waktu yang jauh lebih lama, sedangkan operator B memiliki catatan waktu yang lebih stabil dibanding dengan operator A, di mana waktu yang dibutuhkan operator B untuk mengendalikan robot menutup pintu yang berkisar di antara 26 hingga 42 detik, hal ini disebabkan oleh operator yang memang belum terbiasa dengan sistem kendali robot asisten medis, sehingga memang perlu diadakan pelatihan untuk operator baru agar lebih terbiasa dan menguasai sistem kendali robot seperti yang terjadi pada pengujian menutup pintu dengan operator yang sudah melakukan pelatihan selama kurang lebih 1 Minggu sehingga hanya membutuhkan waktu rata-rata 23,3 detik untuk mengendalikan robot menutup pintu.

4.3.4 Perbandingan robot asisten medis UII dengan robot asisten medis lainnya

Adanya perancangan mekanisme pembuka pintu pada robot asisten medis yang telah dilakukan menambah keunggulan robot asisten medis yang dikembangkan oleh UII, untuk perbandingan lebih jelas antara robot asisten medis UII dan robot asisten medis lainnya, dapat dilihat pada Tabel 4-15.

Tabel 4-15 Perbandingan robot asisten medis yang sudah ada

	UII	RAISA (ITS & Unair)	AMY	KECE (Unesa)
Biaya Produksi	40 juta	100 juta	210 juta	75 juta
Kapasitas/kekuatan struktur	100 kg	50 kg	5 kg	100 kg
Pengendali	Android	joystick	autonomous	Tidak dicantumkan
Kapasitas baterai	0.48 kWh	0.85 kWh	20 Ah	Tidak dicantumkan
Konfigurasi rak	Bisa dilepas	Tidak bisa dilepas	Tidak bisa dilepas	Tidak bisa dilepas
Fitur membuka pintu	Terdapat mekanisme pembuka pintu	Tidak ada	Tidak ada	Tidak ada

Berdasarkan Tabel 4-15 terlihat bahwa robot yang telah dirancang UII memiliki beberapa fitur yang tidak dimiliki oleh robot asisten medis lainnya, seperti kapasitas beban yang dapat ditahan yang mencapai 100 kg, fitur konfigurasi rak yang dapat dilepas dan fitur mekanisme pembuka pintu. Selain itu, biaya yang dibutuhkan untuk memproduksi 1 unit robot asisten medis rancangan UII lebih rendah dibandingkan dengan ketiga robot asisten medis lainnya.

BAB 5

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, dapat diambil kesimpulan sebagai berikut :

1. Telah dirancang dan dibuat aplikasi remot kontrol pada *Android* untuk mengendalikan mekanisme pembuka pintu pada robot asisten medis.
2. Telah dirancang dan dibuat sistem kendali untuk mekanisme lengan robot asisten medis menggunakan *ESP32* sebagai pusat kendali lengan yang dapat dikendalikan dengan remot berbasis *Android* dan memanfaatkan konektifitas *Wi-Fi* sebagai jalur komunikasi.
3. Robot sudah dapat dikendalikan untuk membuka pintu dengan persentase keberhasilan 100% baik yang dikendalikan oleh operator yang sudah latihan maupun yang belum, namun dari pengujian robot membuka pintu yang dilakukan dengan operator yang terlatih dan melihat robot secara langsung, persentase keberhasilan robot membuka pintu dengan lancar mencapai 60%, kurang lancar 40% dan waktu total rata-rata yang dibutuhkan robot untuk membuka pintu hingga masuk ke dalam ruangan adalah 63,2 detik, jika operator terlatih melakukan pengujian mengendalikan robot dari jarak jauh persentase keberhasilan turun menjadi 50% berjalan dengan lancar dan 50% kurang lancar dengan waktu total rata-rata 71,3 detik, pada pengujian dengan operator yang belum terlatih, persentase keberhasilan robot membuka pintu menjadi 20% berjalan dengan lancar, 80% kurang lancar dan waktu total rata-rata yang dibutuhkan adalah 159 detik untuk operator A, serta 60% berjalan dengan lancar, 40% kurang lancar dan membutuhkan waktu selama 89,6 detik untuk operator B. Pengujian menutup pintu yang dilakukan baik dengan operator terlatih atau belum, menghasilkan tingkat keberhasilan 100% dengan catatan pada operator yang terlatih jika melihat secara langsung persentase keberhasilan menutup pintu dengan lancar mencapai 90%, kurang lancar 10% dan waktu yang dibutuhkan robot untuk menutup pintu adalah 22,8 detik

sedangkan jika dilakukan dari jarak jauh persentase berhasil dengan lancar menjadi 60%, kurang lancar 40% dan waktu yang dibutuhkan menjadi 23,3 detik, untuk hasil pengujian dengan operator belum terlatih operator A berhasil dengan lancar sebesar 40%, kurang lancar 60% dan waktu yang dibutuhkan adalah 51 detik, serta operator B berhasil dengan lancar 60% dan kurang lancar 40% dan robot membutuhkan waktu selama 34,6 detik untuk dapat membuka pintu.

5.2 Saran atau Penelitian Selanjutnya

Dari hasil penelitian yang telah dilakukan masih banyak kekurangan dan sangat dimungkinkan untuk dilakukan pengembangan lebih lanjut, di antaranya adalah :

1. Merancang Aplikasi kontrol pada *Android* yang dapat mengendalikan robot beserta mekanisme pembuka pintu agar operator lebih mudah dalam mengendalikan robot asisten medis
2. Merancang lengan robot yang lebih fleksibel sehingga dapat digunakan untuk sesuatu yang lain, tidak hanya khusus untuk membuka pintu
3. Merancang rangka robot yang baru agar keseimbangan robot dapat tersebar dengan rata.
4. Merancang sistem kamera robot dapat melihat kondisi sekeliling robot untuk memudahkan operator dalam mengendalikan robot

DAFTAR PUSTAKA

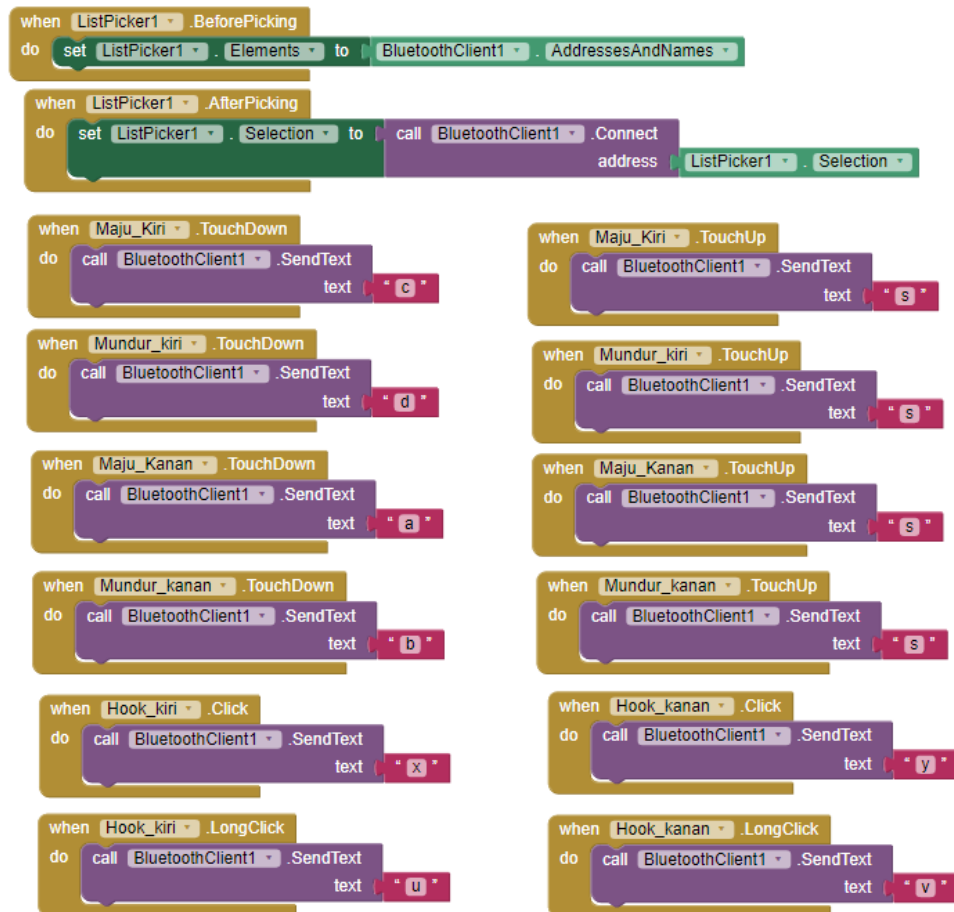
1. COVID-19, G.T.P.P. *Situasi Virus Corona*. 2021; Available from: <https://www.covid19.go.id/situasi-virus-corona/> (Diakses pada 7 April 2021).
2. Prawira, A.E. *237 Dokter di Indonesia Meninggal karena Virus Corona COVID-19*. 2021; Available from: <https://www.liputan6.com/health/read/4447499/237-dokter-di-indonesia-meninggal-karena-virus-corona-covid-19> (Diakses pada 7 April 2021).
3. Khuzzai, R., *PROTOTYPE SISTEM KENDALI KRAN ELEKTRIK PADA METERAN AIR PDAM BERBASIS APLIKASI ANDROID*. 2018.
4. Aminullah, T., *TA: Rancang Bangun Drone Pembersih Sampah Menggunakan Arduino Uno sebagai Pengendali Utama*. 2020, Universitas Dinamika.
5. Utama, S.N. and O.V. Putra, *RANCANG BANGUN ROBOT PEMOTONG RUMPUT OTOMATIS MENGGUNAKAN WIRELESS KONTROLER MODUL ESP32-CAM BERBASIS INTERNET of THINGS (IoT)*. Jurnal Teknoinfo, 2021. **15**(1): p. 45-55.
6. Latifa, U. and J.S. Saputro, *Perancangan Robot Arm Gripper Berbasis Arduino Uno Menggunakan Antarmuka Labview*. Barometer, 2018. **3**(2): p. 138-141.
7. Suryawan, D. and R.S. Adinandra, *RANCANG BANGUN ROBOT PELAYAN MEDIS UNTUK PASIEN KARANTINA COVID-19 DENGAN KENDALI BERBASIS ANDROID*. JTT (Jurnal Teknologi Terapan), 2021. **7**(1): p. 68-76. doi: <https://doi.org/10.31884/jtt.v7i1.312>
8. Kosasi, S., *PERANCANGAN SISTEM INFORMASI PENGENDALI PERANGKAT ELEKTRONIK MENGGUNAKAN PEMROGRAMAN JAVA DAN C++*. TECHSI-Jurnal Teknik Informatika, 2014. **6**(1).
9. Djuandi, F., *Pengenalan arduino*. E-book. www. tobuku, 2011. **24**.
10. Kurniawan, A., *Internet of Things Projects with ESP32: Build exciting and powerful IoT projects using the all-new Espressif ESP32*. 2019: Packt Publishing Ltd.
11. Sadewo, A.D.B., E.R. Widasari, and A. Muttaqin, *Perancangan Pengendali Rumah menggunakan Smartphone Android dengan Konektivitas Bluetooth*. Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer e-ISSN, 2017. **2548**: p. 964X.
12. Zulfauzi, Z., *RANCANG BANGUN PINTU GESER OTOMATIS BERBASIS ANDROID MENGGUNAKAN JARINGAN WIFI*. Jurnal TIPS: Jurnal Teknologi Informasi dan Komputer Politeknik Sekayu, 2016. **5**(2): p. 34-40.
13. Sari, R.P., *PEMANFAATAN SMARTPHONE SEBAGAI PENGONTROL ROTATOR ANTENA MENGGUNAKAN WI-FI*. 2017, Politeknik Negeri Padang.
14. Brooks, A., *A robust layered control system for a mobile robot*, Massachusetts Institute of Technology. Artificial Intelligence Laboratory, 1985.

15. *User Guide BTS7960 High Current 43A H-Bridge Motor Driver*, H. Technology, Editor.: www.handsontec.com.
16. Aryaviocholda, F., M.H.H. Ichsan, and A.S. Budi, *Rancangan Sistem Pendeteksi Pencurian Helm Menggunakan Protokol MQTT Dan Bluetooth HC-05 Berbasis Arduino*. Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer e-ISSN, 2020. **2548**: p. 964X.
17. Wolber, D., et al., *App Inventor*. 2011: " O'Reilly Media, Inc."



LAMPIRAN 1

PROGRAM PADA APLIKASI ANDROID UNTUK PENGGUNAAN *BLUETOOTH*



المعهد الإسلامي للدراسات والبحوث

LAMPIRAN 2

PROGRAM PADA ARDUINO



```
lengan_bluetooth $

#include <Servo.h>

Servo myservo;
Servo myservo2;
int ls_kama = 2;
int ls_kamu = 3;
int ls_kima = 4;
int ls_kimu = 5;
int mkap = 6;
int mkam = 7;
int mkip = 8;
int mkim = 9;

void setup() {
  Serial.begin(9600);
  myservo.attach(10);
  myservo2.attach(11);
  pinMode(ls_kima, INPUT_PULLUP);
  pinMode(ls_kimu, INPUT_PULLUP);
  pinMode(ls_kamu, INPUT_PULLUP);
  pinMode(ls_kama, INPUT_PULLUP);
  pinMode(mkap, OUTPUT);
  pinMode(mkam, OUTPUT);
  pinMode(mkip, OUTPUT);
  pinMode(mkim, OUTPUT);
  myservo.write(0);
  myservo2.write(150);
}

void loop() {
  int kond1 = digitalRead(ls_kamu);
  int kond2 = digitalRead(ls_kama);
  int kond3 = digitalRead(ls_kima);
  int kond4 = digitalRead(ls_kimu);|
```



```
lengan_bluetooth $

void loop() {
  int kond1 = digitalRead(ls_kamu);
  int kond2 = digitalRead(ls_kama);
  int kond3 = digitalRead(ls_kima);
  int kond4 = digitalRead(ls_kimu);

  if(Serial.available()>0){
    char baca = Serial.read();
    if (baca == 'x'){
      myservo2.write(70);
    }
    else if (baca == 'u'){
      myservo2.write(180);
    }
    else if (baca == 'y'){
      myservo.write(180);
    }
    else if (baca == 'v'){
      myservo.write(0);
    }
    else if(baca == 'a' ){
      if(kond1 == 0){
        digitalWrite(mkap, HIGH);
        digitalWrite(mkam, LOW);
        // Serial.println("kanan maju");
      }
      if(kond1 == 1){
        digitalWrite(mkap, LOW);
        digitalWrite(mkam, LOW);
      }
    }
    else if(baca == 'b' ){
      if(kond2 == 0){
        digitalWrite(mkap, LOW); //kanan mundur
        digitalWrite(mkam, HIGH);
        //Serial.println("kanan mundur");
      }
    }
  }
}
```

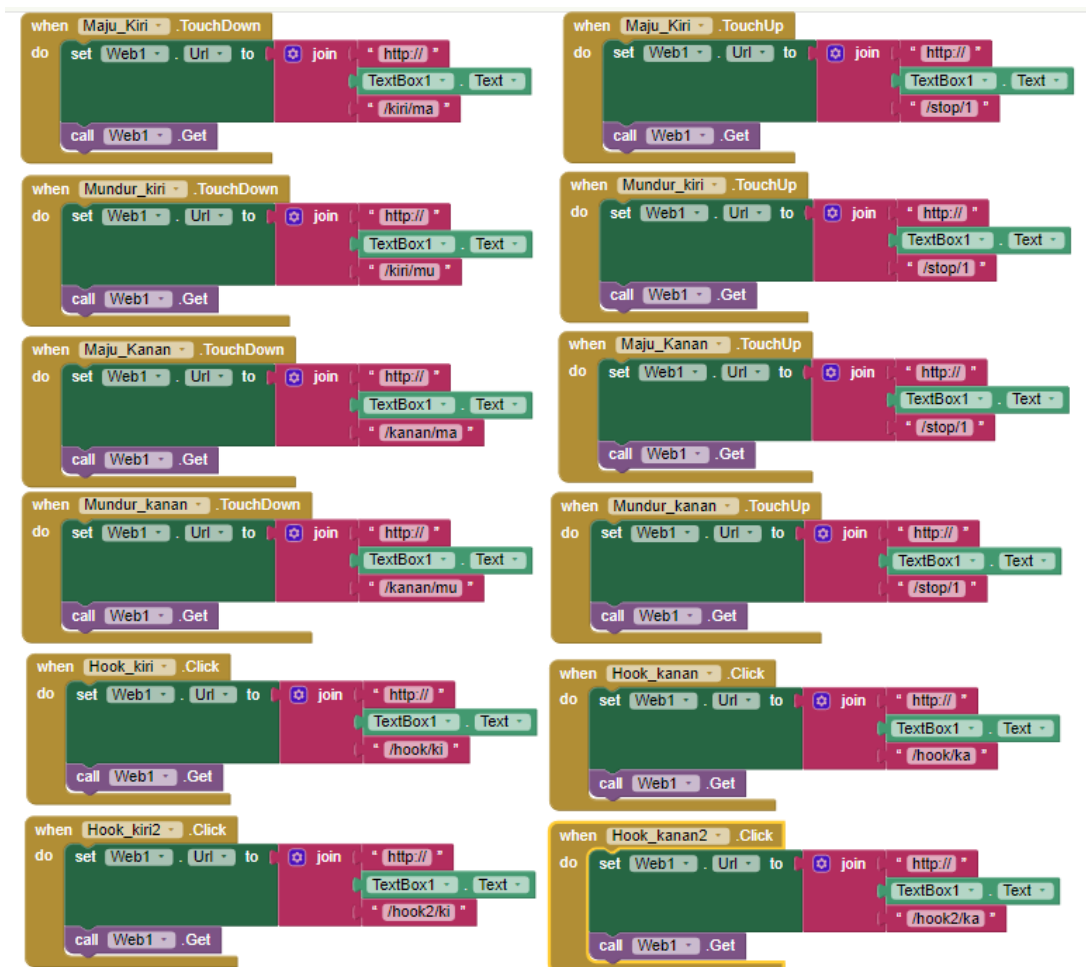


```
lengan_bluetooth$

else if(baca == 'b' ){
  if(kond2 == 0){
    digitalWrite(mkap, LOW); //kanan mundur
    digitalWrite(mkam, HIGH);
    //Serial.println("kanan mundur");
  }
  if(kond2 == 1){
    digitalWrite(mkap, LOW);
    digitalWrite(mkam, LOW);
  }
}
else if(baca == 'c' ){
  if(kond3 == 0){
    digitalWrite(mkip, HIGH); // kiri maju
    digitalWrite(mkim, LOW);
    Serial.println("kimaju");
  }
  if(kond3 == 1){
    digitalWrite(mkip, LOW);
    digitalWrite(mkim, LOW);
  }
}
else if(baca == 'd' ){
  if(kond4 == 0){
    digitalWrite(mkip, LOW); // kiri mundur
    digitalWrite(mkim, HIGH);
    Serial.println("kimundur");
  }
  if(kond4 == 1){
    digitalWrite(mkip, LOW);
    digitalWrite(mkim, LOW);
  }
}
else if (baca == 's'){
  digitalWrite(mkap, LOW);
  digitalWrite(mkam, LOW);
  digitalWrite(mkip, LOW);
  digitalWrite(mkim, LOW);
}
}
```

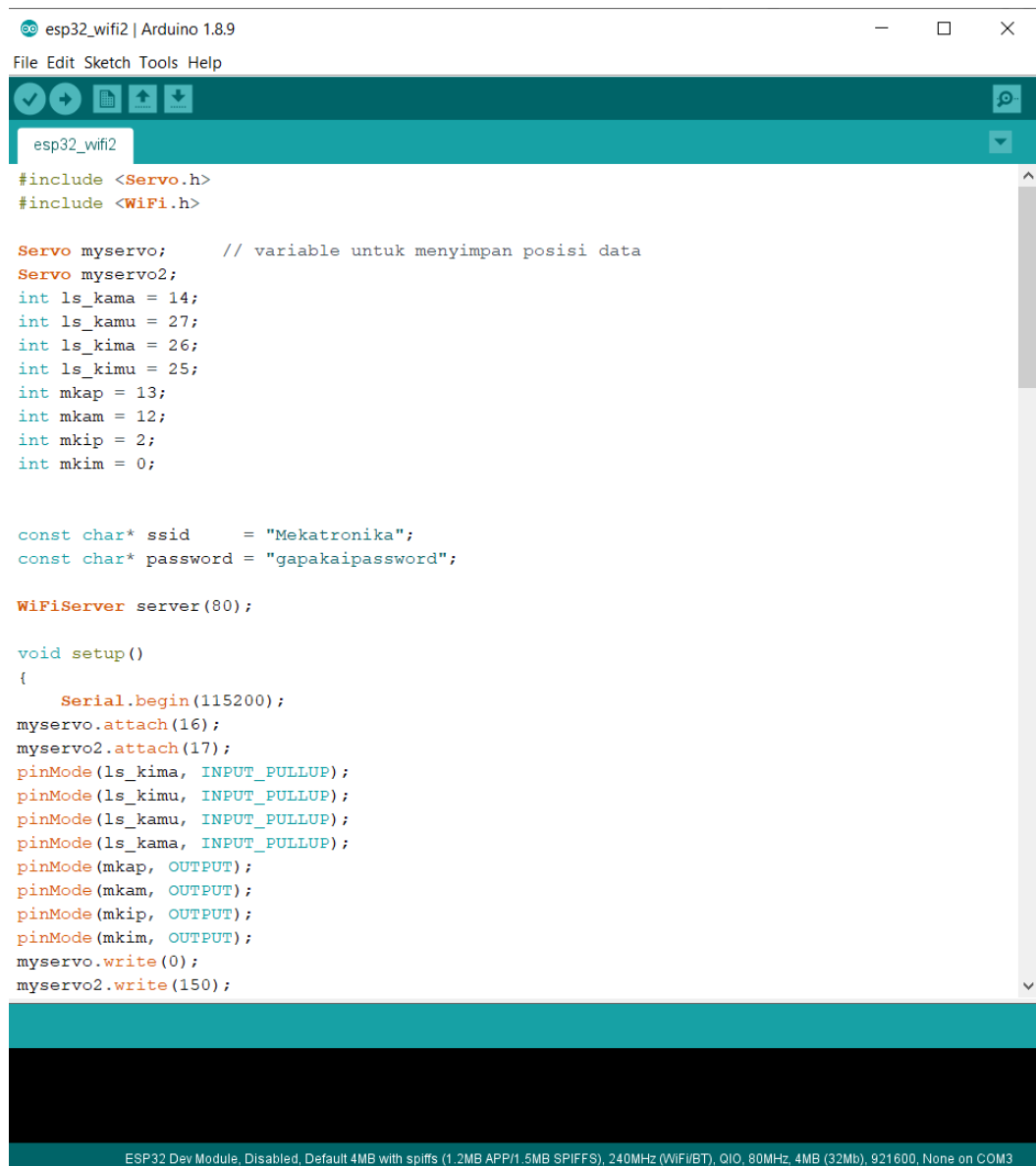
LAMPIRAN 3

PROGRAM PADA APLIKASI ANDROID UNTUK PENGGUNAAN *BLUETOOTH*



LAMPIRAN 4

PROGRAM PADA ESP32



```
esp32_wifi2 | Arduino 1.8.9
File Edit Sketch Tools Help

esp32_wifi2

#include <Servo.h>
#include <WiFi.h>

Servo myservo;      // variable untuk menyimpan posisi data
Servo myservo2;
int ls_kama = 14;
int ls_kamu = 27;
int ls_kima = 26;
int ls_kimu = 25;
int mkap = 13;
int mkam = 12;
int mkip = 2;
int mkim = 0;

const char* ssid     = "Mekatronika";
const char* password = "gapakaipassword";

WiFiServer server(80);

void setup()
{
    Serial.begin(115200);
    myservo.attach(16);
    myservo2.attach(17);
    pinMode(ls_kima, INPUT_PULLUP);
    pinMode(ls_kimu, INPUT_PULLUP);
    pinMode(ls_kamu, INPUT_PULLUP);
    pinMode(ls_kama, INPUT_PULLUP);
    pinMode(mkap, OUTPUT);
    pinMode(mkam, OUTPUT);
    pinMode(mkip, OUTPUT);
    pinMode(mkim, OUTPUT);
    myservo.write(0);
    myservo2.write(150);
}

ESP32 Dev Module, Disabled, Default 4MB with spiiffs (1.2MB APP/1.5MB SPIFFS), 240MHz (WiFi/BT), QIO, 80MHz, 4MB (32Mb), 921600, None on COM3
```



```
esp32_wifi2 | Arduino 1.8.9
File Edit Sketch Tools Help

esp32_wifi2
myservo.write(0);
myservo2.write(150);
  delay(10);

  // We start by connecting to a WiFi network

  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  Serial.println("");
  Serial.println("WiFi connected.");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());

  server.begin();
}

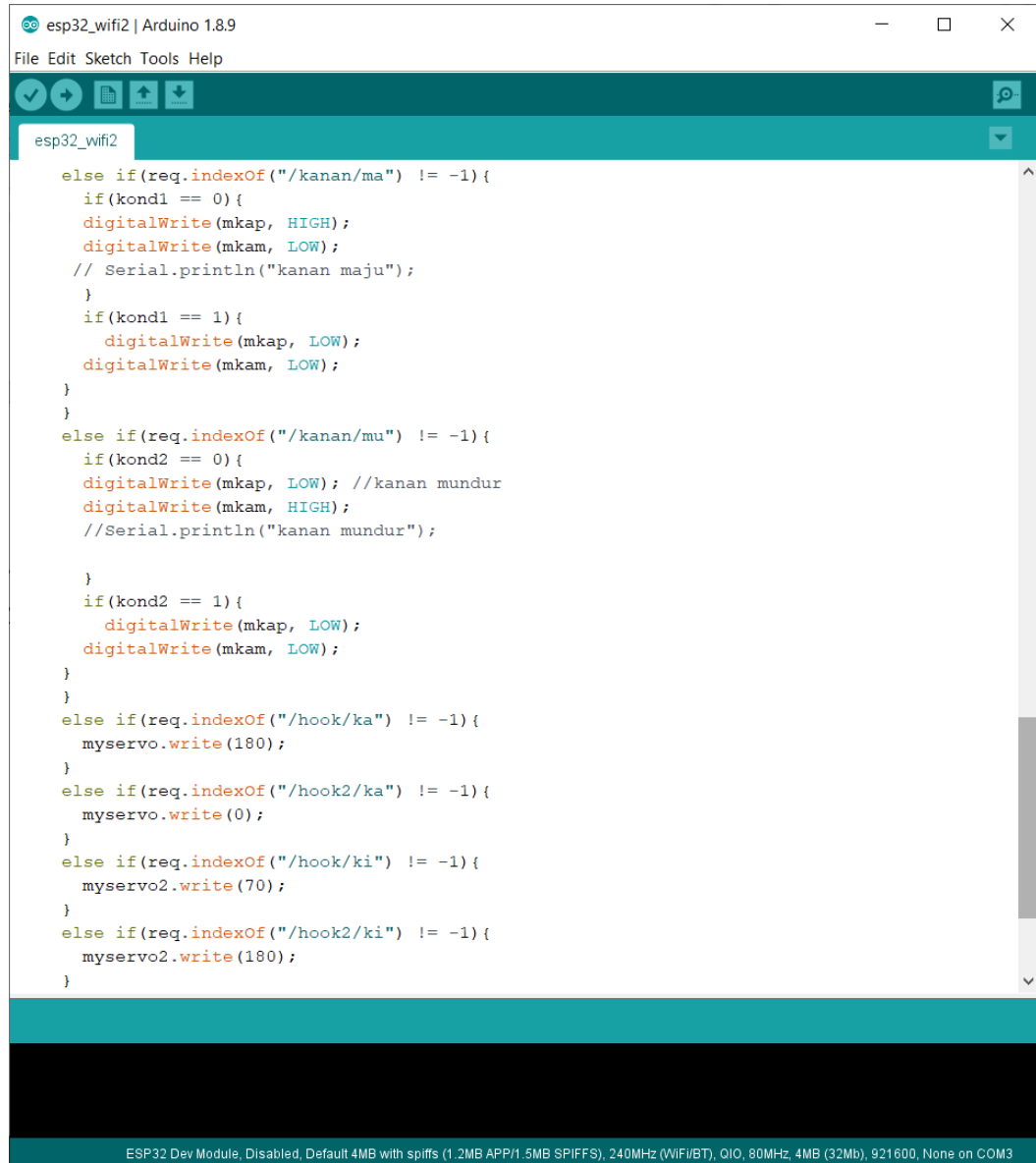
int value = 0;

void loop() {
  WiFiClient client = server.available(); // listen for incoming clients
  int kond1 = digitalRead(ls_kamu);
  int kond2 = digitalRead(ls_kama);
  int kond3 = digitalRead(ls_kima);
  int kond4 = digitalRead(ls_kimu);
```

ESP32 Dev Module, Disabled, Default 4MB with spiiffs (1.2MB APP/1.5MB SPIFFS), 240MHz (WiFi/BT), QIO, 80MHz, 4MB (32Mb), 921600, None on COM3



```
void loop() {
  WiFiClient client = server.available(); // listen for incoming clients
  int kond1 = digitalRead(1s_kamu);
  int kond2 = digitalRead(1s_kama);
  int kond3 = digitalRead(1s_kima);
  int kond4 = digitalRead(1s_kimu);
  if (client) { // if you get a client,
    String req = client.readStringUntil('\r');
    client.flush();
    if (req.indexOf("/kiri/ma") != -1) {
      if (kond3 == 0) {
        digitalWrite(mkip, HIGH); // kiri maju
        digitalWrite(mkim, LOW);
        Serial.println("kimaju");
      }
      if (kond3 == 1) {
        digitalWrite(mkip, LOW);
        digitalWrite(mkim, LOW);
      }
    }
    if (req.indexOf("/kiri/mu") != -1) {
      if (kond4 == 0) {
        digitalWrite(mkip, LOW); // kiri mundur
        digitalWrite(mkim, HIGH);
        Serial.println("kimundur");
      }
      if (kond4 == 1) {
        digitalWrite(mkip, LOW);
        digitalWrite(mkim, LOW);
      }
    }
  }
  else if (req.indexOf("/kanan/ma") != -1) {
    if (kond1 == 0) {
      digitalWrite(mkap, HIGH);
      digitalWrite(mkam, LOW);
    }
  }
}
```



```
esp32_wifi2

else if(req.indexOf("/kanan/ma") != -1){
  if(kond1 == 0){
    digitalWrite(mkap, HIGH);
    digitalWrite(mkam, LOW);
    // Serial.println("kanan maju");
  }
  if(kond1 == 1){
    digitalWrite(mkap, LOW);
    digitalWrite(mkam, LOW);
  }
}
else if(req.indexOf("/kanan/mu") != -1){
  if(kond2 == 0){
    digitalWrite(mkap, LOW); //kanan mundur
    digitalWrite(mkam, HIGH);
    //Serial.println("kanan mundur");

  }
  if(kond2 == 1){
    digitalWrite(mkap, LOW);
    digitalWrite(mkam, LOW);
  }
}
else if(req.indexOf("/hook/ka") != -1){
  myservo.write(180);
}
else if(req.indexOf("/hook2/ka") != -1){
  myservo.write(0);
}
else if(req.indexOf("/hook/ki") != -1){
  myservo2.write(70);
}
else if(req.indexOf("/hook2/ki") != -1){
  myservo2.write(180);
}
```

ESP32 Dev Module, Disabled, Default 4MB with spiffs (1.2MB APP/1.5MB SPIFFS), 240MHz (WiFi/BT), QIO, 80MHz, 4MB (32Mb), 921600, None on COM3

esp32_wifi2 | Arduino 1.8.9

File Edit Sketch Tools Help

esp32_wifi2

```
}
}
else if(req.indexOf("/kanan/mu") != -1){
  if(kond2 == 0){
    digitalWrite(mkap, LOW); //kanan mundur
    digitalWrite(mkam, HIGH);
    //Serial.println("kanan mundur");

  }
  if(kond2 == 1){
    digitalWrite(mkap, LOW);
    digitalWrite(mkam, LOW);
  }
}
else if(req.indexOf("/hook/ka") != -1){
  myservo.write(180);
}
else if(req.indexOf("/hook2/ka") != -1){
  myservo.write(0);
}
else if(req.indexOf("/hook/ki") != -1){
  myservo2.write(70);
}
else if(req.indexOf("/hook2/ki") != -1){
  myservo2.write(180);
}
else if(req.indexOf("/stop/1") != -1){
  digitalWrite(mkap, LOW);
  digitalWrite(mkam, LOW);
  digitalWrite(mkip, LOW);
  digitalWrite(mkim, LOW);
}
}
```

Updates available for some of your [boards](#) and [libraries](#)

MB SPIFFS), 240MHz (WiFi/BT), QIO, 80MHz, 4MB (32Mb), 921600, None on COM3

المعهد الإسلامي للدراسات والبحوث