



**APLIKASI SMARTPHONE PENGENDALI ROBOT PEMBERSIH
LANTAI BERBASIS ARDUINO UNO**

TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat Untuk Menyelesaikan Studi
Jenjang Program Diploma Tiga

Oleh:

Nama
Mohamad Idam Kholid

NIM
20041083

**PROGRAM STUDI DIII TEKNIK KOMPUTER
POLITEKNIK HARAPAN BERSAMA TEGAL**

2023

HALAMAN PERNYATAAN KEASLIAN

Yang bertanda tangan dibawah ini:

Nama : Mohamad Idam Kholid
NIM : 20041083
Jurusan/Program Studi : DIII Teknik Komputer
Jenis Karya : Tugas Akhir

Adalah mahasiswa Program Studi DIII Teknik Komputer Harapan Bersama, dengan ini kami menyatakan bahwa laporan Tugas Akhir yang berjudul **“APLIKASI SMARTPHONE PENGENDALI ROBOT PEBERSIH LANTAI BERBASIS ARDUINO UNO“**.

Merupakan hasil pemikiran dan kerjasama kelompok secara orisinil dan kami susun secara mandiri dan tidak melanggar kode etika hak karya cipta. Pada pelaporan Tugas Akhir ini juga bukan merupakan karya yang pernah diajukan untuk memperoleh gelar akademik tertentu disuatu perguruan tinggi, dan sepanjang pengetahuan kami juga tidak terdapat karya atau pendapat yang pernah ditulis atau disebutkan dalam daftar pustaka.

Apabila di kemudian hari ternyata Laporan Tugas Akhir ini terbukti melanggar kode etik karya cipta atau merupakan karya yang dikategorikan mengandung unsur plagiarisme, maka kami bersedia untuk melakukan penelitian baru dan menyusun laporannya sebagai Laporan Tugas Akhir, sesuai dengan ketentuan yang berlaku.

Demikian pernyataan ini kami buat dengan sebenarnya dan sesungguhnya.

Tegal, Juni 2023



(Mohamad Idam Kholid)

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS
AKHIR UNTUK KEPERLUAN AKADEMIS**

Sebagai civitas akademik PoliTeknik Harapan Bersama Tegal, Kami yang bertanda tangan dibawah ini:

Nama : Mohamad Idam Kholid
NIM : 20041083
Jurusan / Program Studi : Teknik Komputer
Jenis Karya : Tugas Akhir

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Politeknik Harapan Bersama Tegal **Hak Bebas Royalti Non eksklusif (Non-exclusive Royalty Free Right)** atas Tugas Akhir kami yang berjudul: **“APLIKASI SMARTPHONE PENGENDALI ROBOT PEMBERSIH LANTAI BERBASIS ARDUINO UNO “**

Beserta perangkat yang ada (jika diperlukan. Dengan Hak Bebas Royalti Non eksklusif ini Politeknik Harapan Bersama Tegal berhak menyimpan, mengalih media/formatkan, mengolah dalam bentuk pangkalan data (*database*), merawat dan mempublikasikan Tugas Akhir kami selama tetap mencantumkan nama kami sebagai penulis/pencipta dan pemilik hak cipta.

Demikian pernyataan ini kami buat dengan sebenarnya.

Dibuat di : Tegal
Pada Tanggal : Juni 2023
Yang menyatakan



Mohamad Idam Kholid
20041083

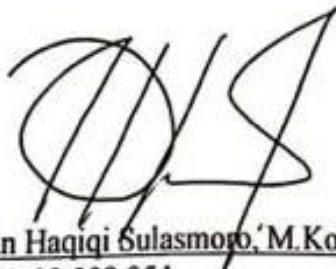
HALAMAN PERSETUJUAN

Tugas Akhir (TA) yang berjudul “**APLIKASI SMARTPHONE PENGENDALI ROBOT PEMBERSIH LANTAI BERBASIS ARDUINO UNO**” yang disusun oleh : Mohamad Idam Kholid, NIM 20041083, telah mendapatkan persetujuan pembimbing dan siap dipertahankan di depan tim penguji Tugas Akhir (TA) program Studi D-III Teknik Komputer Politeknik Harapan Bersama Tegal.

Tegal, Juni 2023

Menyetujui,

Pembimbing I



Arfan Haqiqi Sulasmoro, M.Kom
NIPY. 02.009.054

Pembimbing II



Safar Dwi Kurniawan, M.Kom
NIPY. 03.021.487

HALAMAN PENGESAHAN

Judul : APLIKASI SMARTPHONE PENGENDALI ROBOT PEMBERSIH
LANTAI BERBASIS ARDUINO UNO

Nama : Mohamad Idam Kholid

NIM : 20041083

Program Studi : Teknik Komputer

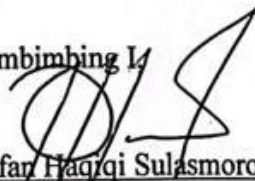
Jenjang : Diploma III

**Dinyatakan LULUS / TIDAK LULUS setelah dipertahankan di depan Tim Penguji
Tugas Akhir Program Studi DIII Teknik Komputer Politeknik Harapan Bersama
Tegal**

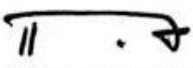
Tegal, 5 Juli 2023

Tim Penguji:


Pembimbing I


Arfar Haqiqi Sulasmoro, M.Kom.
NIPY. 02.099.054


Ketua Penguji,


Rais, Spd., M.Kom
NIPY. 07.011.083

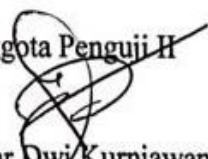
Pembimbing II


Safar Dwi Kurniawan, M.Kom
NIPY. 03.021.487

Anggota Penguji I


M. Teguh Prihandoyo, M.Kom
NIPY. 02.005.012

Anggota Penguji II


Safar Dwi Kurniawan, M.kom
NIPY. 03.021.487

Mengetahui,

Ketua Program Studi DIII Teknik Komputer,
Politeknik Harapan Bersama Tegal



Ida Afriliana, ST, M.Kom
NIPY. 12.013.168

HALAMAN MOTTO

- ❖ Sesungguhnya bersama kesukaran itu ada kemudahan, karena itu bila kau telah selesai mengerjakan yang lain dan kepada Tuhan, berharaplah.(Q.S Al Insyirah : 6-8)
- ❖ Semua orang tidak perlu menjadi malu karena pernah berbuat kesalahan, selamaia menjadi lebih bijaksana daripada sebelumnya. (Alexander Pope)
- ❖ Barangsiapa melepaskan kesusahan seorang mukmin dari kesusahan dunia maka Allah akan melepaskan kesusahannya pada hari kiamat. (HR. Muslim)
- ❖ Banyak kegagalan dalam hidup ini dikarenakan orang-orang tidak menyadari betapa dekatnya mereka dengan keberhasilan saat mereka menyerah. (Thomas Alva Edison)

HALAMAN PERSEMBAHAN

Tugas Akhir ini dipersembahkan untuk:

1. Allah SWT yang telah memberikan kenikmatan dan kemudahan dalam melakukan segala sesuatunnya.
2. Bapak Agung Hendarto, S.E., MA selaku Direktur Politeknik Harapan Bersama Tegal.
3. Ibu Ida Afriliana, ST,M.Kom. Kepala Ketua Program Studi DIIITeknik Komputer Politeknik Harapan Bersama Tegal.
4. Arfan Haqiqi Sulasmoro, M.Kom. selaku Pembimbing I.
5. Safar Dwi Kurniawan, M.Kom. selaku Pembimbing II.
6. Kedua orang tua tercinta yang selalu memberikan doa dan dukungan.
7. Sahabat dan seluruhteman seperjuangan yang telah memberikan semangat dan dukungannya.
8. Almameter kami POLITEKNIK HARAPAN BERSAMA TEGAL.

ABSTRAK

Perkembangan yang terjadi saat ini begitu besar apalagi dibidang teknologi. Dengan kecanggihan yang diciptakan oleh bangsa barat membuat persaingan di seluruh dunia sangatlah ketat. Khususnya di bidang elektronika yang juga telah disiapkan dalam perkembangan teknologi dunia. Mencari permasalahan yang dapat digunakan untuk bahan perancangan sistem. Mencari referensi yang sesuai dengan kebutuhan dalam perancangan sistem yang akan dibuat. Pengumpulan alat dan bahan yang dibutuhkan dalam perancangan sistem. Analisis berdasarkan hasil penelitian terhadap proses bahwa dalam pembuatan aplikasi robot pengepel lantai berbasis android dibutuhkan untuk menggerakkan robot dalam meringankan pekerjaan manusia dalam mengepel lantai. Selanjutnya, tahap perancangan sistem yaitu merancang sistem yang akan digunakan pada robot vacuum dan pel, menyiapkan komponen perangkat keras yaitu Laptop. Tahap berikutnya yaitu menyiapkan komponen perangkat lunak pada Android Studio dan Arduino IDE, dan tahap terakhir pengujian aplikasi yang telah dibuat. Aplikasi ini dapat memerintahakan serta mengontrol robot sesuai dengan yang diperintahkan oleh pengguna. Tombol yang terdapat di aplikasi berfungsi untuk memudahkan pengguna dalam mengambil keputusan untuk mengatur robot supaya bisa mengepel lantai dan vacuum dan tidak membuat robot menabrak halangan yang ada didepannya. program telah diatur robot pengepel dan vacuum sebagai penghubung antara pengguna dengan komponen pada robot pengepel lantai.

Kata kunci : *Arduino, Android, Vacuum, Bluetooth.*

KATA PENGANTAR

Segala puji syukur kehadirat Allah Swt atas segala rahmat serta hidayah- Nya yang telah dilimpahkan dan dikaruniakan kepada kami sehingga dapat menuangkan sebuah karya yang berjudul **“APLIKASI SMARTPHONE PENGENDALI ROBOT PEMBERSIH LANTAI BERBASIS ARDUINO UNO”** ini selesai tepat pada waktunya.

Tugas akhir merupakan suatu kewajiban yang harus dilaksanakan untuk memenuhi salah satu syarat dalam mencapai derajat ahli madya computer pada program studi D III Teknik Komputer Politeknik Harapan Bersama Tegal. Selama melaksanakan penelitian kemudian tersusun dalam laporan tugas akhir ini, banyak pihak yang telah memberikan bantuan, dukungan dan bimbingan.

Pada kesempatan ini kami ingin mengucapkan rasa terima kasih yang sebesar-besarnya kepada:

1. Bapak Agung Hendarto, S.E., MA selaku Direktur Politeknik Harapan Bersama Tegal.
2. Ibu Ida Afriliana, ST, M.Kom. selaku Ketua Program Studi DIII Teknik Komputer Politeknik Harapan Bersama Tegal.
3. Arfan Haqiqi Sulasmoro, M.Kom. selaku pembimbing I.
4. Safar Dwi Kurniawan, M.Kom. selaku pembimbing II.
5. Semua pihak yang telah mendukung, membantu serta mendoakan penyelesaian laporan Tugas Akhir ini.

Laporan Tugas Akhir ini semoga berguna sebagai tambahan ilmu pengetahuan serta dapat memberikan manfaat bagi semua pihak dan dijadikan implikasi selanjutnya bagi mahasiswa.

Tegal, juni 2023

DAFTAR ISI

	Halaman
HALAMAN JUDUL.....	i
HALAMAN PERNYATAAN KEASLIAN.....	ii
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPERLUAN AKADEMIS.....	iii
HALAMAN PERSETUJUAN	iv
HALAMAN PENGESAHAN.....	v
HALAMAN MOTTO	vi
HALAMAN PERSEMBAHAN	vii
ABSTRAK	viii
KATA PENGANTAR.....	ix
DAFTAR ISI.....	x
DAFTAR TABEL	xii
DAFTAR GAMBAR	xiii
DAFTAR LAMPIRAN	xiv
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah.....	2
1.3. Batasan Masalah	3
1.4. Tujuan Dan Manfaat	3
1.4.1. Bagi Mahasiswa	3
1.4.2. Kampus Politeknik Harapan Bersama	4
1.4.3. Masyarakat	4
1.5. Sistematika Penulisan	4
BAB II TINJAUAN PUSTAKA	6
2.1. Teori Terkait	6
2.2. Landasan Teori.....	8
2.2.1. Smartphone.....	8
2.2.2. Android	8
2.2.3. Bluetooth	9
2.2.4. Java.....	10
2.2.5. UML.....	11
BAB III METODOLOGI PENELITIAN.....	18
3.1. Prosedur Penelitian	18
3.1.1. Rencana/ Planning.....	18
3.1.2. Analisis.....	19
3.1.3. Rancangan Atau Desain	19
3.1.4. Implementasi	20
3.2. Metode Pengumpulan Data.....	20
3.3. Waktu Dan Tempat Penelitian	21

3.3.1.	Tempat Penelitian.....	21
3.3.2.	Waktu Penelitian	21
BAB IV ANALISA DAN PERANCANGAN SISTEM		22
4.1.	Analisa Permasalahan	22
4.2.	Analisa Kebutuhan Sistem.....	23
4.2.1.	Kebutuhan Perangkat Keras	23
4.2.2.	Kebutuhan Perangkat Lunak	23
4.3.	Perancangan Sistem	24
4.3.1.	Usecase Diagram.....	24
4.3.2.	Perancangan Coding.....	25
4.4.	Desain Aplikasi Android	27
BAB V IMPLEMENTASI SISTEM.....		29
5.1.	Implementasi Sistem.....	29
5.2.	Implementasi Perangkat Lunak.....	29
5.3.	Rencana Implementasi Sistem.....	30
5.3.1.	Testing Program	30
5.3.2.	Hasil Pengujian	31
BAB VI SIMPULAN DAN SARAN		33
6.1.	Simpulan	33
6.2.	Saran	33
DAFTAR PUSTAKA.....		35

DAFTAR TABEL

	Halaman
Tabel. 2. 1 Simbol Use Case Diagram	12
Tabel. 2. 2 Simbol Activity Diagram	14
Tabel. 2. 3 Simbol Sequence Diagram	15
Tabel. 2. 4 Simbol Class Diagram	17
Tabel. 5. 1 Pengujian Connect Bluetooth	31
Tabel. 5. 2 Pengujian Tombol Pembersih Lantai	31

DAFTAR GAMBAR

	Halaman
Gambar. 2. 1 Smartphone.....	8
Gambar. 2. 2 Logo Android	9
Gambar. 2. 3 Bluetooth	10
Gambar. 2. 4 Logo Java	11
Gambar. 3. 1 Prosedur Penelitian.....	18
Gambar. 4. 1 Flowchart.....	24
Gambar. 4. 2 Perancangan Coding 1.....	25
Gambar. 4. 3 Perancangan Coding 2.....	26
Gambar. 4. 4 Perancangan Coding 3.....	27
Gambar. 4. 5 Desain Aplikasi Android.....	28

DAFTAR LAMPIRAN

	Halaman
Lampiran. 1 Surat Ketersediaan Pembimbing	A-1
Lampiran. 2 Coding Aplikasi Android Android Studio.....	A-3

BAB I

PENDAHULUAN

1.1. Latar Belakang

Lantai adalah suatu tempat didalam ruangan dimana aktifitas manusia berlangsung lebih lama bila dibandingkan diluar ruangan, dengan banyaknya aktifitas maka lantai akan menjadi kotor dan berdebut[1].

Jika tidak dibersihkan maka akan menjadi masalah baru yaitu menjadi sumber penyakit. Lantai harus dibersihkan secara berkala, kegiatan membersihkan lantai di zaman sekarang dibantu dengan robot pembersih lantai [2].

Pada umumnya robot yang digunakan untuk membantu masyarakat dalam membersihkan rumah berjalan otomatis tidak bisa di kontrol, hal ini terkadang membuat masyarakat tidak bisa mengontrol robotnya ketika dalam membersihkan lantai masih ada yang kotor, maka dari itu dibuatkanlah sebuah sistem yang dapat mengatur robot dalam dua mode, yaitu mode otomatis yangt bisa berjalan dengan sendiri dengan mode manual yang bisa di kontrol oleh pengguna [3].

Kemudian pada penelitian Ardi Wijaya dan Doli Juliadi, membuat sebuah aplikasi untuk robot pembersih lantai menggunakan arduino nano dengan sistem pengendali berbasis android, sistem pengendalian robot berupa aplikasi pengendali via Bluetooth berbasis android, Sistem infomasi robot

yaitu berupa notifikasi sms, yang mana robot akan mengirimkan sms setelah berhasil mendeteksi dan memindahkan objek, Mempermudah pekerjaan cleaning service dan dapat menghemat tenaga [4].

Penelitian milik Krisman Kurniawan, membuat aplikasi android pada alat pembersih lantai menggunakan micro controller atmega 16, menghasilkan sistem dengan pengendalian alat penmbersih lantai sepenuhnya dapat dikendalikan oleh user melalui smartphome android dengan cara menekan tombol-tombol yang tersedia pada aplikasi yang telah dirancang di *bluetooth electronics*. Maksimal jarak yang dapat terhubung antara smartphome dan alat pembersih lantai mencapai 12 meter [5].

Berdasarkan latar belakang yang telah diuraikan diatas, maka dalam penelitian ini akan dibuat laporan tugas akhir dengan judul “**APLIKASI SMARTPHONE PENGENDALI ROBOT PEMBERSIH LANTAI BERBASIS ARDUINO UNO**” yang diharapkan dapat membantu pekerjaan rumah terutama dalam mengepel lantai agar lebih ringan.

1.2. Rumusan Masalah

Berdasarkan latar belakang di atas maka adapun permasalahan yang diangkat dalam penelitian ini bagaimana merancang dan menghasilkan aplikasi yang mengatur jalannya robot pengepel lantai dengan menggunakan *smartphone*.

1.3. Batasan Masalah

Batasan Masalah dibuat agar maksud dan tujuan dari penelitian ini terfokus dengan sesuai tujuan dan fungsinya adalah sebagai berikut:

1. Aplikasi yang dibuat hanya untuk prototype robot pengepel lantai berbasis Arduino Uno.
2. Aplikasi hanya bisa dikoneksikan melalui Bluetooth.
3. *MIT APP Inventor* yang digunakan sebagai bahasa pemrograman aplikasi untuk dikoneksikan pada modul *Bluetooth HC-05*.
4. Menggunakan *smartphone* yang akan menjadi kontrol ke 2 pada prototype.

1.4. Tujuan Dan Manfaat

Tujuan dari tugas akhir ini adalah menghasilkan aplikasi yang dapat mengatur jalannya robot pengepel lantai dengan menggunakan *smartphone* yang akan menjadi kontrol ke 2 pada prototype robot pengepel lantai berbasis arduino uno, yang digunakan untuk mengepel lantai dan meringankan pekerjaan masyarakat dalam mengepel lantai.

Manfaat dibuatnya penelitian ini adalah:

1.4.1. Bagi Mahasiswa

1. Menambah wawasan mahasiswa tentang ilmu teknologi.
2. Menyajikan hasil-hasil yang diperoleh.
3. Menggunakan hasil atau data-data untuk dikembangkan menjadi Tugas Akhir.

1.4.2. Kampus *Politeknik Harapan Bersama*

1. Sebagai tolak ukur kemampuan dari mahasiswa dalam menyusun laporan.
2. Memberikan kesempatan pada mahasiswa untuk terjun dan berkomunikasi langsung dengan masyarakat.

1.4.3. Masyarakat

Aplikasi ini dapat memberikan kemudahan masyarakat untuk meminimalisir tenaga manusia dalam mengepel lantai.

1.5. Sistematika Penulisan

Laporan Tugas Akhir ini terdiri dari enam bab, yang masing-masing bab dengan perincian sebagai berikut:

BAB I : PENDAHULUAN

Dalam bab ini menjelaskan tentang latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, dan sistematika penulisan.

BAB II : TINJAUAN PUSTAKA

Dalam bab ini menjelaskan tentang penelitian terkait robot pembersih lantai yang di ambil dari jurnal-jurnal yang di dapatkan dan juga menjelaskan Landasan teori tentang kajian yang di teliti.

BAB III : METODOLOGI PENELITIAN

Bab ini membahas tentang langkah-langkah/tahapan

perencanaan dengan bantuan beberapa metode, teknik, alat (*tools*) yang digunakan seperti metode penelitian dan metode pengumpulan data.

BAB IV : ANALISA DAN PERANCANGAN SISTEM

Dalam bab ini menjelaskan tentang analisa permasalahan, analisa kebutuhan sistem, perancangan sistemnya.

BAB V : HASIL DAN PEMBAHASAN

Dalam bab ini berisikan tentang hasil produk yang telah dibuat. Pada bab ini berisi analisis tentang bagaimana hasil penelitian dapat menjawab pertanyaan pada latar belakang masalah.

BAB VI : SIMPULAN DAN SARAN

Bab ini menguraikan simpulan seluruh isi laporan Tugas Akhir dan saran-saran untuk mengembangkan hasil penelitian ini.

BAB II

TINJAUAN PUSTAKA

2.1. Teori Terkait

Dalam penyusunan penelitian tugas akhir ini, sedikit banyak terinspirasi dan mereferensi dari penelitian-penelitian sebelumnya yang berkaitan dengan latar belakang masalah pada tugas akhir ini.

Penelitian yang dilakukan oleh Ardi Wijaya dan Doli Juliardi, dalam jurnalnya telah membangun sebuah sistem pengendali robot pembersih lantai menggunakan Arduino nano berbasis android. Software yang digunakan dalam membuat sebuah sistem berbasis android yaitu menggunakan MIT App Inverter dengan menggunakan Bluetooth untuk mengkoneksikan pada robot pembersih lantai. Sistem informasi robot yaitu berupa notifikasi sms, yang mana robot akan mengirimkan sms setelah berhasil mendeteksi dan memindahkan objek [6].

Penelitian milik Krisman Kurniawan, membuat aplikasi android pada alat pembersih lantai dengan menggunakan micro controller atmega 16, maksimal jarak yang dapat terhubung antara smartphone dan alat pembersih lantai mencapai 12 meter. Kekurangan dari aplikasi ini, yaitu masih menggunakan kendali dari pengguna tidak terdapat mode otomatis [7].

Robot dan ponsel pintar adalah salah satu alat atau teknologi modern yang sedang berkembang pesat dan sangat membantu manusia. Robot bisa kita buat dan program sesuai dengan kebutuhan kita. Beberapa jenis robot

saat ini adalah robot beroda, robot berkaki, robot humanoid, robot pengintai Selanjutnya penelitian yang dilakukan oleh Ilham Hidayat dan Sumiyati telah mengembangkan prototype robot penyedot debu berbasis mikrokontroller artmega 328 dan fuzzy logic dengan kendali smartphone android di Universitas Serang Raya dimana dijelaskan oleh peneliti disini bahwa pembuatan robot ini menggunakan Fuzzy Logic, yang logikanya memiliki kekaburan antara benar atau salah. Bisa benar dan salah secara bersama. Robot penyedot debu ini memiliki kelebihan yakni menggunakan modul Bluetooth sehingga bisa terhubung dengan modul Android Smartphone, juga memiliki sensor debu sehingga dapat mendeteksi apakah alat ini bekerja dengan baik yakni untuk menghisap debu dengan baik atau tidak. Sedangkan kekurangannya adalah alat ini perlu menggunakan area penyedot debu otomatis agar dapat bekerja dengan baik dan alat ini lebih baik menggunakan Modul Wireless Fidelity sebagai perangkat transmisi robot dengan perangkat controller. Rancang Bangun Robot Pembersih Lantai Otomatis Menggunakan Metode Fuzzy. Selanjutnya ada penelitian robot pengering lantai diprogram membaca halangan menggunakan sensor ultrasonik. Data yang diterima sensor ultrasonik diproses oleh mikrokontroler dan setelah itu robot akan bergerak berdasarkan set poin yang sudah ditentukan dengan menggunakan logika fuzzy[8].

2.2. Landasan Teori

2.2.1. *Smartphone*

Telepon cerdas adalah telepon genggam yang memiliki sistem operasi untuk masyarakat luas, fungsinya tidak hanya untuk SMS dan telepon saja tetapi pengguna dapat dengan bebas menambahkan aplikasi. Telepon cerdas merupakan komputer mini yang mempunyai kapabilitas sebuah telepon. *Smartphone* merupakan alat komunikasi yang dipakai mulai dari kalangan anak-anak, remaja, dewasa, dan orang tua. Pada awalnya *handphone* hanya untuk berkomunikasi saja, seiring perkembangan zaman teknologi hingga bisa mengirim data dan menambah aplikasi. Media komunikasi merupakan kebutuhan pokok bagi individu, kelompok, maupun organisasi. Pada saat ini, *handphone* sudah menjadi kebutuhan primer sehari-hari [9].



Gambar. 2. 1 Smartphone

2.2.2. *Android*

Pengertian Android adalah sistem operasi yang dirancang oleh Google dengan basis kernel Linux untuk mendukung kinerja perangkat elektronik layar sentuh, seperti tablet atau *smartphone*. Jadi, android digunakan dengan sentuhan, gesekan ataupun ketukan

pada layar *gadget* anda.

Android bersifat *open source* atau bebas digunakan, dimodifikasi, diperbaiki dan didistribusikan oleh para pembuat ataupun pengembang perangkat lunak. Dengan sifat *open source* perusahaan teknologi bebas menggunakan OS ini diperangkatnya tanpa lisensi alias *gratis*. Begitupun dengan para pembuat aplikasi, yang bebas membuat aplikasi dengan kode-kode sumber yang dikeluarkan google. Dengan seperti itu android memiliki jutaan *support* aplikasi gratis/berbayar yang dapat diunduh melalui google play [10].



Gambar. 2. 2 Logo Android

2.2.3. *Bluetooth*

Bluetooth adalah suatu peralatan media komunikasi yang dapat digunakan untuk menghubungkan sebuah perangkat komunikasi dengan perangkat komunikasi lainnya, bluetooth umumnya digunakan di *handphone*, komputer atau *pc*, tablet, dan lain-lain. Atau definisi bluetooth yang lainnya adalah sebuah teknologi komunikasi wireless atau tanpa kabel yang beroperasi dalam pita frekuensi 2,4 GHz (antara 2.402 GHz s/d 2.480 GHz) dengan menggunakan sebuah *frequency hopping tranceiver* yang mapu menyediakan layanan komunikasi data dan juga suara secara *real-time* antara *host- host*

bluetooth dengan jarak jangkauan layanan yang terbatas.

Fungsi *bluetooth* yaitu untuk mempermudah berbagi atau sharing file, audio, menggantikan penggunaan kabel dan lain-lain. Saat ini sudah banyak sekali perangkat yang menggunakan *bluetooth*. Pada dasarnya teknologi bluetooth ini diciptakan bukan hanya untuk menggantikan atau menghilangkan penggunaan media kabel dalam melakukan pertukaran data atau informasi, tetapi juga mampu menawarkan fitur yang bagus atau baik untuk teknologi *mobile wireless* atau tanpa kabel, dengan biaya yang relatif rendah, konsumsi daya rendah, *interoperability* yang sangat menjanjikan, mudah dalam pengoperasiannya dan juga mampu menyediakan berbagai macam layanan [11].



Gambar. 2. 3 Bluetooth

2.2.4. Java

Java merupakan bahasa pemrograman tingkat tinggi yang dapat diterapkan pada banyak platform. *Java* mempunyai ciri sebagai bahasa yang sederhana, arsitektur netral berorientasi obyek, mempunyai kinerja yang tinggi, mempunyai multithreaded, kuat, dinamis dan aman. *Java* mempunyai kemampuan dapat berjalan di banyak platform. Sebuah platform adalah perangkat keras atau

perangkat lunak lingkungan dimana program berjalan, seperti : Windows, Linux, Solaris OS dan Mac OS. Platform java punyai dua komponen , yaitu : Java Virtual Machine dan Java Application Programming Interface (API) [12].



Gambar. 2. 4 Logo Java

2.2.5. UML


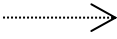
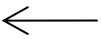
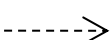
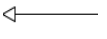
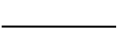
UML (*Unified Modeling Language*) adalah metode pemodelan secara visual sebagai sarana untuk merancang dan atau membuat *software* berorientasi objek. Karena UML ini merupakan bahasa visual untuk pemodelan bahasa berorientasi objek, maka semua elemen dan diagram berbasiskan pada *paradigma object oriented*. UML sendiri juga memberikan standar penulisan sebuah sistem blue print, yang meliputi konsep bisnis proses, penulisan kelas - kelas dalam bahasa program yang spesifik.


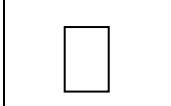
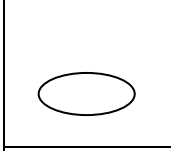
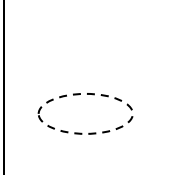
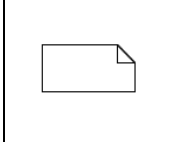
Beberapa diagram yang digunakan di UML (*Unified Modeling Language*) :

1. Use Case Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah use case mempresentasikan sebuah interaksi antara aktor dengan sistem. Use case merupakan sebuah pekerjaan tertentu, misalnya *login* ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya. Seorang atau sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan - pekerjaan tertentu.

Tabel. 2. 1 Simbol Use Case Diagram

Gambar	Nama	Keterangan
	Actor	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
	Dependency	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
	Generalization	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancertor</i>).
	Include	Menspesifikasikan bahwa use case sumber secara eksplisit.
	Extend	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
	Association	Apa yang menghubungkan antara objek satu dengan objek


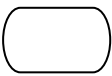




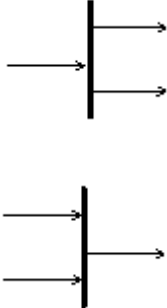

Gambar	Nama	Keterangan
		lainnya.
	System	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
	Use Case	Deskripsi dari urutan aksi – aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
	Collaboration	Interaksi aturan – aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen – elemennya (sinergi).
	Note	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

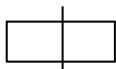

2. Activity Diagram

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing - masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* merupakan *state diagram* khusus, dimana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan *behaviour internal* sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses - proses dan jalur - jalur aktivitas dari level atas secara umum. Sebuah aktivitas dapat direalisasikan oleh satu

use case atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktifitas. *Decision* digunakan untuk menggambarkan *behaviour* pada kondisi tertentu. Untuk mengilustrasikan proses - proses paralel (*fork* dan *join*) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal.

Tabel. 2. 2 Simbol Activity Diagram

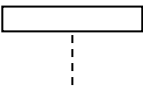
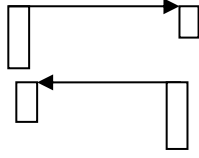

Gambar	Nama	Keterangan
	Activity	Memperlihatkan bagaimana masing - masing kelas antarmuka saling berinteraksi satu sama lain.
	Action	State dari sistem yang mencerminkan eksekusi suatu aksi.
	Initial Node	Bagaimana objek dibentuk atau diawali.
	Final Node	Bagaimana objek dibentuk dan dihancurkan.
	Fork Node	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran.
	Decision	Pilihan untuk mengambil keputusan
	Fork/Join	Digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu
	Rake	Menunjukkan adanya dekomposisi

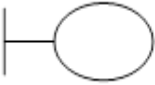



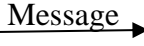
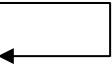
Gambar	Nama	Keterangan
	Time	Tanda waktu
	Send	Tanda pengiriman

3. Sequence Diagram

Sequence diagram menggambarkan interaksi antar di sekitar (pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek - objek yang terkait). *Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah - langkah yang dilakukan sebagai respon dari sebuah event untuk menghasilkan *output* tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan. Masing - masing objek, termasuk aktor, memiliki *lifeline* vertikal.

Tabel. 2. 3 Simbol Sequence Diagram

No	Gambar	Nama	Keterangan
1.		LifeLine	Objek <i>entity</i> , antar muka yang saling berinteraksi.
2.		Message	Spesifikasi dari komunikasi antar objek yang memuat informasi – informasi tentang aktifitas yang terjadi.
3		Actor	Menggambarkan orang yang sedang berinteraksi dengan sistem

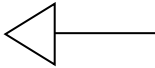
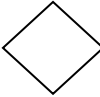
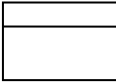

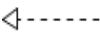
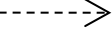

No	Gambar	Nama	Keterangan
4		Boundary Class	Menggambarkan penggambaran dari form
5		Entity Class	Menggambarkan hubungan kegiatan yang akan dilakukan
6.		Control Class	Menggambarkan penghubung antara Boundary dengan tabel
7		Activation	Sebagai sebuah objek yang akan melakukan sebuah aksi
8		Message	Mengindikasikan komunikasi antara objek dengan objek
9		Self Message	Menginndikasikan komunikasi kembali kedalam sebuah objek itu sendiri

4. Class Diagram

Class *adalah* sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* diagram menggambarkan struktur dan deskripsi class, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. *Class* memiliki tiga area pokok : nama, atribut, dan metode yang memiliki salah satu sifat berikut :

- a. *Private*, tidak dapat dipanggil dari luar *class* yang bersangkutan
- b. *Protected*, hanya dapat dipanggil oleh *class* yang bersangkutan dan anak anak yang mewarisinya.

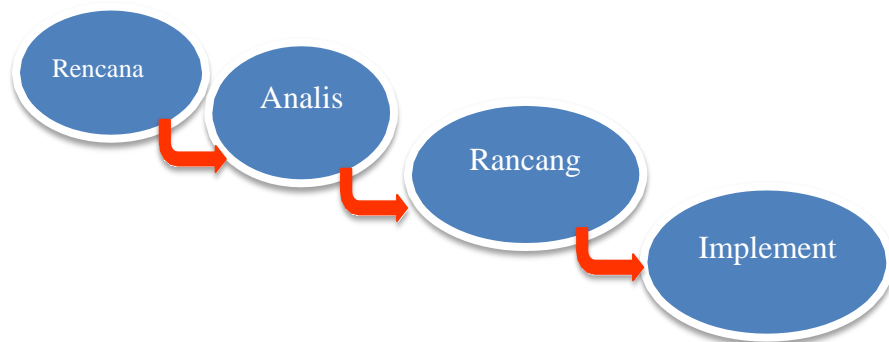
Tabel. 2. 4 Simbol Class Diagram

No	Gambar	Nama	Keterangan
1.		Generalization	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>)
2.		Nary Association	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3.		Class	Himpunan dari objek - objek yang berbagi atribut serta operasi yang sama.
4.		Collaboration	Deskripsi daxri urutan aksi - aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
5.		Dependency	Operasi yang benar - benar dilakukan oleh suatu objek.
6.		Dependency	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
7.		Association	Apa yang menghubungkan antara objek satu dengan objek lainnya.

BAB III

METODOLOGI PENELITIAN

3.1. Prosedur Penelitian



Gambar. 3. 1 Prosedur Penelitian

3.1.1. Rencana/ Planning

Rencana atau planning merupakan langkah awal dalam melakukan penelitian. Rencananya akan di buat aplikasi robot pengepel lantai berbasis android berikut langkah- langkah perancangannya:

- a. Mencari permasalahan yang dapat digunakan untuk bahan perancangan sistem.
- b. Mencari referensi yang sesuai dengan kebutuhan dalam perancangan sistem yang akan dibuat.
- c. Pengumpulan alat dan bahan yang dibutuhkan dalam perancangan sistem.

3.1.2. Analisis

Analisis berdasarkan hasil penelitian terhadap proses bahwa dalam pembuatan aplikasi robot pengepel lantai berbasis android dibutuhkan untuk menggerakkan robot dalam meringankan pekerjaan manusia dalam mengepel lantai.

3.1.3. Rancangan *Atau Desain*

Mengaplikasikan teori yang didapat dari studi literatur. Subyek penelitian ini adalah aplikasi yang mempunyai kemampuan menggerakkan robot menggunakan *Handphone*. Aplikasi didesain memiliki empat buah tombol yaitu tombol untuk menghubungkan *Bluetooth*, tombol kedua untuk mengatur pel, tombol ketiga untuk vacuum, dan tombol keempat adalah menjalankan pel dan vacuum. Control untuk menjalankan robot yang berupa arah atas, kanan, bawah, dan kiri dengan bantuan modul bluetooth HC-05 untuk menghubungkan aplikasi dengan robot. Sistem kerja dalam usaha mengepel lantai, robot ini bisa menghindari halangan dengan berbelok ke kanan, berbelok ke kiri dan juga bisa dikontrol dengan smartphone. Sistem kerja robot ini akan diprogram dengan mikrokontroler Arduino. Perancangan atau desain sistem akan dijelaskan melalui *flowchart* sehingga dapat digambarkan secara rinci sistem kerjanya.

3.1.4. Implementasi

Penelitian ini perlu sebuah rencana yang tersusun dengan baik guna mendapatkan hasil yang objektif. Perencanaan penelitian ini diambil sesuai dengan sistem akan diuji dan dipasang sebagai monitoring guna mengetahui apakah sistem berjalan atau tidak.

3.2. Metode Pengumpulan Data

Untuk menyusun laporan tugas akhir ini, penulis menggunakan metode pengumpulan data sebagai berikut:

1. Literatur

Studi literatur adalah mencari data penelitian berupa data sheet dari sebuah jurnal, Badan Pusat Statistik dan lain-lain. Studi literatur dapat dicantumkan jika dalam pengumpulan datanya mengambil data dari literatur namun jika tidak mengambil data dari literatur tidak usah dicantumkan dalam metode pengumpulan data.

2. Metode Observasi

Metode pengumpulan data melalui pengamatan langsung atau peninjauan secara cermat dan langsung di lapangan atau lokasi penelitian. Dalam hal ini, peneliti dengan berpedoman kepada desain penelitiannya perlu mengunjungi lokasi penelitian untuk mengamati langsung berbagai hal atau kondisi yang ada di lapangan.

3.3. Waktu Dan Tempat Penelitian

3.3.1. Tempat Penelitian

Penelitian dilakukan di rumah penulis, Kelurahan Tunon,
Kecamatan Tegal Selatan

3.3.2. Waktu Penelitian

Waktu Penelitian ini berlangsung selama kurang lebih dua
bulan, dimulai dari bulan Maret sampai dengan bulan Mei 2023.

BAB IV

ANALISA DAN PERANCANGAN SISTEM

4.1. Analisa Permasalahan

Lantai adalah suatu tempat didalam ruangan dimana aktifitas manusia berlangsung lebih lama bila dibandingkan diluar ruangan. Terutama pada lantai rumah yang harus di jaga kebersihannya guna untuk kenyamanan saat istirahat maupun ibadah, lantai dalam rumah harus dibersihkan secara berkala, sehingga diperlukan robot pengepel lantai yang akan membantu seorang masyarakat, agar menghemat tenaga manusia terutama seorang ibu rumah tangga dalam mengepel lantai. Robot pengepel lantai yang berbentuk mobil mainan memudahkan dalam penggunaannya dalam kontrol melalui *smartphone*. Robot pengepel lantai ini cocok dengan media datar, luas, dan minim halangan seperti pada lantai rumah. Apalagi dengan lantai rumah yang terkadang ada halangan tentu memudahkan masyarakat dalam melakukan pekerjaan yaitu mengepel lantai. Dengan robot ini masyarakat bisa menghemat waktu sambil melakukan pekerjaan lainnya. Dan dengan dibandingkan dengan robot pengepel lainnya yang hanya kontrol otomatis dan tidak ada kontrol air, robot ini juga lebih mudah penggunaannya dengan koneksi *bluetooth* yang mudah difahami, dapat menghindari tembok dengan control melalui *smartphone* sehingga tidak akan menabrak, bisa mengatur keluarnya air pel ataupun vacuum dengan kendali melalui *smartphone*.

Feedback dari penggunaan alat ini adalah bentuknya yang seperti mobil mainan dengan menggunakan kendali melalui *bluetooth* dan android membuat alat ini mudah digunakan. Dengan dua mode, *user* dapat mengendalikan sesuai keinginan dengan mode manual, atau dengan mode otomatis tanpa khawatir alat akan menabrak tembok. Alat ini juga cukup efektif dalam membantu pekerjaan manusia dalam mengepel lantai.

4.2. Analisa Kebutuhan Sistem

Analisa kebutuhan *system* dilakukan untuk mengetahui spesifikasi dari kebutuhan robot pengepel lantai yang akan dibuat. Pada tahap ini akan membahas mengenai perangkat keras dan perangkat lunak yang dibutuhkan dalam pembuatan Robot Pengepel Lantai Berbasis Arduino Uno.

4.2.1. Kebutuhan Perangkat Keras

Adapun perangkat keras yang digunakan untuk membangun perangkat ini adalah sebagai berikut:

1. 1 buah Laptop
2. RAM 4 GB
3. SSD 512 GB
4. 1 Handphone

4.2.2. Kebutuhan Perangkat Lunak

Adapun perangkat Lunak yang digunakan untuk membangun perangkat ini adalah sebagai berikut:

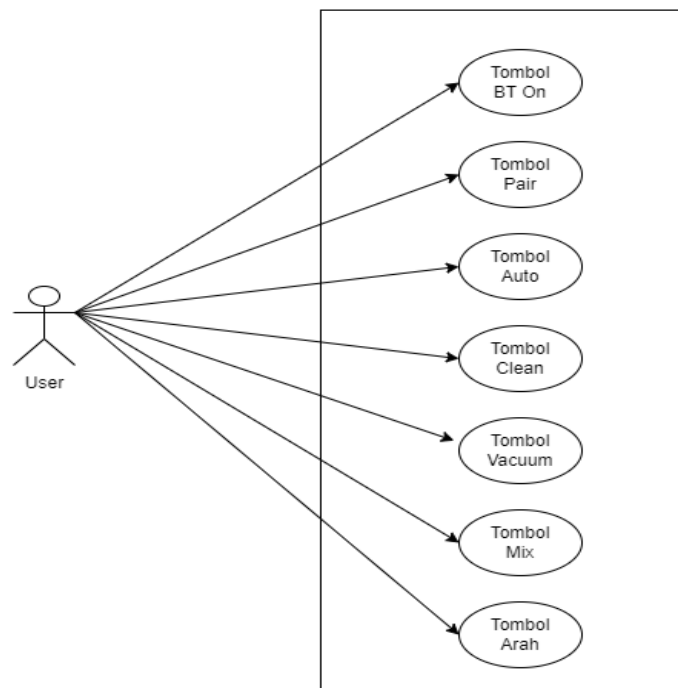
1. *Windows 11*

2. *Arduino IDE*
3. *Android Studio*
4. *Draw.io*

4.3. Perancangan Sistem

Pada perancangan sistem yang dibuat, bisa diketahui alur atau cara kerja robot pengesep lantai yang saling terhubung untuk mendukung sistem yang akan dibangun. Sistem yang akan dibangun dapat digambarkan dengan bentuk alur *flowchart*.

4.3.1. Usecase Diagram



Gambar. 4. 1 Usecase Diagram Aplikasi

Berdasarkan *flowchart* pada Gambar 4.1 bahwa proses awalan ketika menggunakan aplikasi ini adalah buka aplikasi, lalu tekan

tombol BT On untuk mengaktifkan *Bluetooth*, kemudian tekan pair untuk mengkoneksikannya ke aplikasi pengendali, setelah terhubung ada dua mode yang bisa dipilih yaitu manual dan otomatis, jika otomatis maka robot dapat bergerak dengan bantuan sensor, jika pilih manual maka robot bisa dikontrol oleh pengguna dari tombol pengepel, vacuum atau keduanya. Jika ingin mematikan kendali cukup memutuskan koneksi saja pada tombol BT OFF.

4.3.2. Perancangan Coding

```

public void handleMessage(Message msg) {
    if (msg.what == MESSAGE_READ) {
        String readMessage = null;
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.KITKAT) {
            readMessage = new String(byte[] msg.obj, StandardCharsets.UTF_8);
        }
    }
    if (msg.what == CONNECTING_STATUS) {
        checkIfConnected();
        String message = "S";
        if (msg.arg1 == 1) {
            Toast.makeText(getApplicationContext(), "Connected to " + msg.obj,
                Toast.LENGTH_SHORT).show();
            command(message);
        } else {
            Toast.makeText(getApplicationContext(), "Connection Failed",
                Toast.LENGTH_SHORT).show();
        }
    }
};

btn_clean.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
    @Override
    public void onCheckedChanged(CompoundButton compoundButton, boolean b) {
        if (b) {
            compoundButton.setBackgroundColor(Color.BLUE);
            String message = "C";
            command(message);
        } else {
            compoundButton.setBackgroundColor(Color.RED);
            String message = "c";
            command(message);
        }
    }
});

btn_vacuum.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
    @Override
    public void onCheckedChanged(CompoundButton compoundButton, boolean b) {
        if (b) {
            compoundButton.setBackgroundColor(Color.BLUE);
            String message = "V";
            command(message);
        } else {
            compoundButton.setBackgroundColor(Color.RED);
            String message = "v";
            command(message);
        }
    }
});

btn_both.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
    @Override
    public void onCheckedChanged(CompoundButton compoundButton, boolean b) {
        if (b) {
            compoundButton.setBackgroundColor(Color.BLUE);
            String message = "X";
            command(message);
        } else {
            compoundButton.setBackgroundColor(Color.RED);
            String message = "x";
            command(message);
        }
    }
});

btn_forward.setOnTouchListener(new View.OnTouchListener() {
    @Override
    public boolean onTouch(View view, MotionEvent motionEvent) {
        switch (motionEvent.getAction()) {
            case MotionEvent.ACTION_DOWN:
                mHandler.postDelayed(mForward, 100);
                break;
            case MotionEvent.ACTION_UP:
                mHandler.removeCallbacks(mForward);
                String message = "S";
                command(message);
                break;
        }
        return true;
    }
});

Runnable mForward = new Runnable() {
    @Override
    public void run() {
        String message = "F";
        command(message);
        mHandler.postDelayed(this, 100);
    }
};

```

Gambar. 4. 2 Perancangan Coding 1

Pada Gambar 4.2 berisi *function* yang digunakan untuk menggerakkan sebuah motor dinamo penggerak ban. Penggerak roda mobil di program untuk bisa *idle* keadaan berhenti, bahkan setelah

melakukan control juga akan berhenti jika perintah selesai dikerjakan.

```

btn_back.setOnTouchListener(new View.OnTouchListener() {
    @Override
    public boolean onTouch(View view, MotionEvent motionEvent) {
        switch (motionEvent.getAction()) {
            case MotionEvent.ACTION_DOWN:
                mHandler.postDelayed(mBack, 10);
                break;
            case MotionEvent.ACTION_UP:
                mHandler.removeCallbacks(mBack);
                String message="S";
                comand(message);
                break;
        }
        return true;
    }
}
Runnable mBack = new Runnable() {
    @Override
    public void run() {
        String message="B";
        comand(message);
        mHandler.postDelayed(this, 100);
    }
};

btn_left.setOnTouchListener(new View.OnTouchListener() {
    @Override
    public boolean onTouch(View view, MotionEvent motionEvent) {
        switch (motionEvent.getAction()) {
            case MotionEvent.ACTION_DOWN:
                mHandler.postDelayed(mLeft, 10);
                break;
            case MotionEvent.ACTION_UP:
                mHandler.removeCallbacks(mLeft);
                String message="S";
                comand(message);
                break;
        }
        return true;
    }
}
Runnable mLeft = new Runnable() {
    @Override
    public void run() {
        String message="L";
        comand(message);
        mHandler.postDelayed(this, 100);
    }
};
});

```

Gambar. 4. 3 Perancangan Coding 2

Pada Gambar 4.3 dijelaskan bagaimana coding mengatur perintah untuk tombol *on* atau *off* tombol pel dan vacuum.


```

btn_right.setOnTouchListener(new View.OnTouchListener() {
    @Override
    public boolean onTouch(View view, MotionEvent motionEvent) {
        switch (motionEvent.getAction()) {
            case MotionEvent.ACTION_DOWN:
                mHandler.postDelayed(mRight, 10);
                break;
            case MotionEvent.ACTION_UP:
                mHandler.removeCallbacks(mRight);
                String message="S";
                comand(message);
                break;
        }
        return true;
    }
}
Runnable mRight = new Runnable() {
    @Override
    public void run() {
        String message="R";
        comand(message);
        mHandler.postDelayed(this, 100);
    }
};
});
}

private void comand(String Message){
    if (mBTSocket != null) {
        try {
            // Create the command that will be sent to arduino.
            // String must be converted in its bytes to be sent on serial
            // communication
            mBTSocket.getOutputStream().write(Message.getBytes());
        } catch (IOException e) {

        }
    }
}
}
}
}

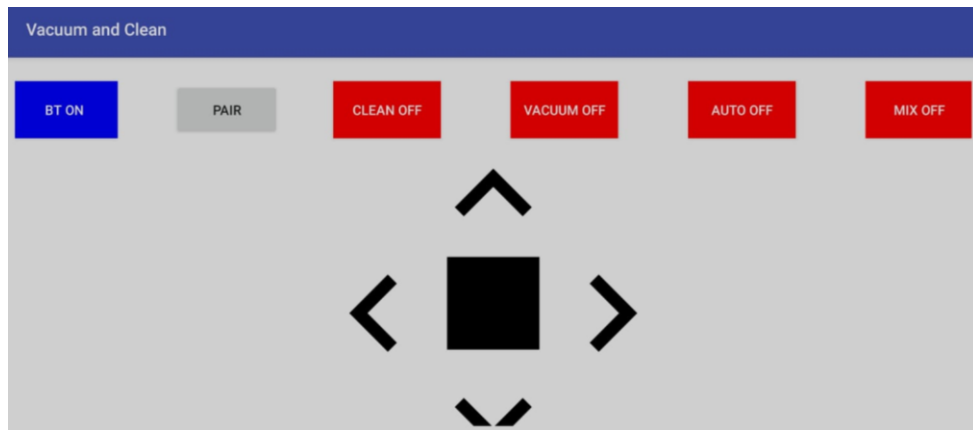
```

Gambar. 4. 4 Perancangan Coding 3

Pada Gambar 4.5 dijelaskan fungsi untuk mengatur tombol yang ketiga, yang dimana fungsi tombol ini adalah melakukan perintah ke robot untuk melakukan mengepel dan vacuum secara bersamaan dan jika ditekan sekali lagi maka menghentikan pel dan vacuum.

4.4. Desain Aplikasi Android

Pada gambar 4.5 adalah desain aplikasi yang terdapat lima tombol yang berfungsi untuk menghubungkan *Bluetooth*, dan mengatur untuk pel dan vacuum, serta mengatur gerak jalan robot.



Gambar. 4. 5 Desain Aplikasi Android.

BAB V

IMPLEMENTASI SISTEM

5.1. Implementasi Sistem

Setelah melakukan metodologi penelitian, maka didapatkan analisa sistem, analisa permasalahan serta analisa kebutuhan perangkat keras dan perangkat lunak guna membangun aplikasi ini. Selanjutnya, tahap perancangan sistem yaitu merancang sistem yang akan digunakan pada robot *vacuum* dan pel, menyiapkan komponen perangkat keras yaitu Laptop. Tahap berikutnya yaitu menyiapkan komponen perangkat lunak pada *Android Studio* dan *Arduino IDE*, dan tahap terakhir pengujian aplikasi yang telah dibuat. Aplikasi ini diimplementasikan di Rumah Tunon Tegal.

5.2. Implementasi Perangkat Lunak

1. *Android Studio*

Pada bagian ini merupakan proses implementasi program *Software* *Android Studio*, yaitu dengan memasukkan *coding* sesuai dengan kebutuhan sistem, lalu *install* ke dalam *Smartphone*.

2. *Smartphone*

Pada bagian ini merupakan aplikasi kendali jalannya robot pengepel lantai yang telah dibuat dengan *android studio*. Implementasi *smartphone* dengan menyalakan *bluetooth* lalu mengkoneksikannya dengan aplikasi yang sudah dibuka, setelah terkoneksi maka robot sudah

bisa dikendalikan otomatis ataupun manual.

3. *Draw.io*

Pada bagian ini merupakan website untuk membuat skema perancangan dari aplikasi guna untuk mempermudah dalam melakukan perancangan dengan *drag* atau *drop* komponen *part* ke dalam lembar projek skematik, lalu menyambungkan atau *wiring* ke masing-masing *pin/port* komponen agar terhubung.

5.3. Rencana Implementasi Sistem

Setelah program dibuat, langkah selanjutnya yaitu mengimplementasikan aplikasi menggunakan *Android Studio*, maka dibuat rencana implementasi Sistem, yaitu dengan *Testing* program.

5.3.1. Testing Program

Setelah program dibuat, maka selanjutnya yaitu Testing program, testing dilakukan untuk mengetahui apakah program tersebut berjalan sesuai dengan yang diharapkan. Testing yang dilakukan yaitu seperti melakukan *connect bluetooth*, pengecakan *control* robot dan pengecakan tombol pel, vacuum, dan tombol untuk mengaktifkan keduanya. Pada saat testing apakah terdapat sebuah kesalahan atau tidak, dan saat testing setiap langkah perlu dicatat sehingga dapat mempermudah perbaikan sistem dan dibuatkan solusi atas kesalahan tersebut.

5.3.2. Hasil Pengujian

Berikut adalah hasil pengujian sistem yang telah dilakukan :

1. Pengujian Connect Bluetooth

Tabel. 5. 1 Pengujian Connect Bluetooth

Yang Diuji	Detail Pengujian	Keterangan
Tombol BT ON	Pengujian dilakukan dengan menekan tombol BT ON untuk menyalakan <i>Bluetooth</i> pada <i>Smartphone</i> yang akan digunakan sebagai <i>control</i> untuk ke robot pengepel	tombol pada aplikasi sudah bisa digunakan dan <i>Bluetooth</i> sudah aktif
Tombol BT OFF	Pengujian dilakukan dengan menekan tombol BT OFF untuk mematikan <i>Bluetooth</i> pada <i>Smartphone</i> .	tombol pada aplikasi sudah bisa digunakan dan <i>Bluetooth</i> berhasil dimatikan
Tombol Pair	Pengujian dilakukan dengan menekan tombol pair, lalu sambungkan sesuai dengan nama <i>Bluetooth</i> pada robot, jika berhasil maka terhubung dengan robot.	tombol pada aplikasi sudah bisa digunakan dan aplikasi sudah bisa terhubung dengan robot.

2. Pengujian Tombol Pembersih Lantai

Tabel. 5. 2 Pengujian Tombol Pembersih Lantai

Yang Diuji	Detail Pengujian	Keterangan
Tombol Clean	Pengujian dilakukan dengan menekan tombol clean untuk melakukan perintah ke robot untuk mengepel lantai.	tombol pada aplikasi sudah bisa digunakan dan robot berhasil mengepel.

Yang Diuji	Detail Pengujian	Keterangan
Tombol Vacum	Pengujian dilakukan dengan menekan tombol vacum untuk melakukan perintah ke robot supaya vacuum dinyalakan.	Tombol pada aplikasi sudah bisa digunakan, dan berhasil memvacuum.
Tombol Both	Pengujian dilakukan dengan menekan tombol Both untuk melakukan perintah ke robot supaya mengaktifkan pengepel dan vacuum.	tombol pada aplikasi sudah bisa digunakan dan robot berhasil mengepel dan vacuum.

BAB VI

SIMPULAN DAN SARAN

6.1. Simpulan

Dari hasil perancangan dan pengujian aplikasi untuk robot penggepel lantai berbasis arduino uno dapat disimpulkan sebagai berikut:

1. Telah dilakukan perancangan, pembuatan dan uji coba sistem secara keseluruhan dan aplikasi ini dapat digunakan untuk mengontrol robot penggepel. Aplikasi ini dapat memerintahkan serta mengontrol robot sesuai dengan yang diperintahkan oleh pengguna. Tombol yang terdapat di aplikasi berfungsi untuk memudahkan pengguna dalam mengambil keputusan untuk mengatur robot supaya bisa menggepel lantai dan vacuum dan tidak membuat robot menabrak halangan yang ada didepannya.
2. Kerja aplikasi sesuai dengan yang telah di program untuk mengatur robot penggepel dan vacuum sebagai penghubung antara pengguna dengan komponen pada robot penggepel lantai.

6.2. Saran

Pembuatan aplikasi harus diperhitungkan dengan baik agar fungsi dari aplikasi dapat berfungsi dengan maksimal. Pengembang selanjutnya agar dapat menambahkan beberapa fitur agar aplikasi untuk robot penggepel lantai dapat dijalankan dengan lebih mudah, diantaranya:

1. Desain dan aplikasi yang lebih efektif supaya pengguna lebih mudah dalam menggunakan aplikasi.
2. Menambahkan notifikasi melalui media sosial yang terhubung seperti *WhatsApp* atau *Telegram* agar mengetahui apakah robotpengepel lantai ini sudah selesai membersihkan lantai atau belum.

DAFTAR PUSTAKA

- [1] M. D. Faraby, M. Akil, A. Fitriati, and I. Isminarti, "Rancang Bangun Robot Pembersih Lantai Berbasis Arduino," *JTT (Jurnal Teknol. Terpadu)*, vol. 5, no. 1, p. 70, 2017, doi: 10.32487/jtt.v5i1.214.
- [2] R. N. Rohmah and N. Kaloka, "Pemanfaatan Smartphone berbasis Android sebagai Alat Pengontrol Pembersih Lantai," *Emit. J. Tek. Elektro*, vol. 19, no. 2, pp. 81–85, 2019, doi: 10.23917/emit.v19i2.8558.
- [3] S. Ardhi, "Pembuatan Alat Pembersih Lantai Yang Dikendalikan Dari Bluetooth Software Android," *Senati*, vol. E.55-8, p. 55.1-55.8, 2016.
- [4] Wijaya Ardi, "Rancang Bangun Robot Pembersih Lantai Menggunakan Arduino Nano dengan Sistem Pengendali Berbasis Android," Vol. 8, pp. 98–107, 2021.
- [5] S. Krisman Kurniawan, "Aplikasi Android Pada Alat Pembersih Lantai Dengan Menggunakan Micro Controller Atmega 16," 2020.
- [6] K. Umam, H. Haryanto, and R. Alfita, "Rancang Bangun Robot Pembersih Kaca Otomatis Berbasis Mikrokontroler ARM STM32 Dengan Sensor Proximity," *Indones. J. Eng. Technol.*, vol. 2, no. 1, pp. 24–29, 2019, doi: 10.26740/inajet.v2n1.p24-29.
- [7] A. B., "Pengertian robot," vol. 3, no. 2, p. 4, 2013, [Online]. Available: https://www.google.com/url?sa=i&rct=j&q=&esrc=s&source=web&cd=&ad=rja&uact=8&ved=0CAQQw7AJahcKEwiQp4ulhpn5AhUAAAAAHQAAAAAQAg&url=http%3A%2F%2Fprints.undip.ac.id%2F41644%2F3%2FBAB_2.pdf&psig=AOvVaw0DPVLGMCM8mOfOiFwczro&ust=1659010180702115
- [8] I. G. Putu, M. Eka, P. Alit, and W. Santiary, "Deteksi Objek Bergerak Menggunakan Cmcum Berbasis Mikrokontroler Atmega 8535 Moving Object Detection Using Cmcum Based Microcontroller Atmega 8535," *J. Log.*, vol. 16, no. 1, pp. 25–29, 2016.
- [9] K. Saleh, "Rancangan Perangkat Lunak Pengendali Robot Pemantau Berbasis Ponsel Pintar Android," *Saintek ITM*, vol. 32, no. 2, pp. 11–15, 2019, doi: 10.37369/si.v32i2.56.
- [10] A. Fitriansyah, G. N. Esmeralda, and D. Setiadi, "Alat Pembersih Lantai Berbasis Arduino Uno dan Android," *J. Teknol. Inform. dan Komput.*, vol. 6, no. 1, pp. 72–84, 2020, doi: 10.37012/jtik.v6i1.163.
- [11] A. Fakhrana, "Pembuatan Prototype Robot Kapal Pemungut Sampah menggunakan Mikrokontroler Arduino Ano dengan Aplikasi Pengendali Berbasis Android," *J. Ilm. Teknol. dan Rekayasa*, vol. 21, no. 3, pp. 185–195, 2016, [Online]. Available: <https://ejournal.gunadarma.ac.id/index.php/tekno/article/view/1597/1356>

- [12] A. Febriandirza, “Perancangan Aplikasi Absensi Online Dengan Menggunakan Bahasa Pemrograman Kotlin,” *Pseudocode*, vol. 7, no. 2, pp. 123–133, 2020, doi: 10.33369/pseudocode.7.2.123-133.

LAMPIRAN

Lampiran. 1 Surat Ketersediaan Pembimbing

SURAT KESEDIAAN MEMBIMBING TA

Yang bertanda tangan dibawah ini:

Nama : Arfan Haqiqi Sulasmoro, M.Kom
NIDN : 0623037704
NIPY : 02.009.054
Jabatan Struktural : -
Jabatan Fungsional : Asisten ahli

Dengan ini menyatakan bersedia menjadi pembimbing I pada Tugas Akhir mahasiswa berikut :

NO	Nama	NIM	Program Studi
1	Mohamad Idam Kholid	20041083	DIII Teknik Komputer

Judul TA : APLIKASI SMARTPHONE PENGENDALI ROBOT PEMBERSIH LANTAI BERBASIS ARDUINO UNO

Demikian pernyataan ini dibuat agar dapat dilaksanakan sebagaimana mestinya.

Tegal, 3 Februari 2023

Mengetahui,
Kepala Program Studi DIII Teknik Komputer



Ida Afriliana, ST, M.Kom
NIPY. 12.013.168

Dosen Pembimbing I,

Arfan Haqiqi Sulasmoro, M.Kom
NIPY.02.009.054

SURAT KESEDIAAN MEMBIMBING TA

Yang bertanda tangan dibawah ini:

Nama : Safar Dwi Kurniawan, M.Kom
NIDN : 0624089101
NIPY : 03.021.487
Jabatan Struktural : -
Jabatan Fungsional : Asisten Ahli

Dengan ini menyatakan bersedia menjadi pembimbing II pada Tugas Akhir mahasiswa berikut :

NO	Nama	NIM	Program Studi
1	Mohamad Idam Kholid	20041083	DIII Teknik Komputer

Judul TA : APLIKASI SMARTPHONE PENGENDALI ROBOT PEMBERSIH LANTAI BERBASIS ARDUINO UNO

Demikian pernyataan ini dibuat agar dapat dilaksanakan sebagaimana mestinya.

Tegal, 3 Februari 2023

Mengetahui,
Kepala Program Studi DIII Teknik Komputer



Ida Afriliana, ST, M.Kom
NIPY: 12.013.168

Dosen Pembimbing II,

Safar Dwi Kurniawan, M.Kom.
NIPY: 03.021.487

Lampiran. 2 Coding Aplikasi Android Android Studio

```
import android.Manifest;
import android.app.AlertDialog;
import android.app.Dialog;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.bluetooth.BluetoothSocket;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.IntentFilter;
import android.content.pm.PackageManager;
import android.graphics.Color;
import android.os.Build;
import android.os.Bundle;
import android.os.Handler;
import android.os.Looper;
import android.os.Message;
import android.provider.CalendarContract;
import android.util.Log;
import android.view.MotionEvent;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.CompoundButton;
import android.widget.ImageView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;
import android.widget.ToggleButton;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;
import java.io.IOException;
import java.io.UnsupportedEncodingException;
import java.lang.reflect.Method;
import java.nio.charset.StandardCharsets;
import java.util.Set;
import java.util.UUID;

public class MainActivity extends AppCompatActivity {
    private final String TAG = MainActivity.class.getSimpleName();

    private static final UUID BT_MODULE_UUID =
        UUID.fromString("00001101-0000-1000-8000-00805F9B34FB"); //
        "random" unique identifier

    // #defines for identifying shared types between calling
    functions
    private final static int REQUEST_ENABLE_BT = 1; // used to
    identify adding bluetooth names
```

```

public final static int MESSAGE_READ = 2; // used in bluetooth
handler to identify message update
    private final static int CONNECTING_STATUS = 3; // used in
bluetooth handler to identify message status

    // GUI Components
    private TextView mBluetoothStatus;
    private TextView mReadBuffer;
    private Button mListPairedDevicesBtn;
    private ToggleButton btn_vacuum;
    private ToggleButton btn_clean;
    private ToggleButton btn_both;
    private ToggleButton btn_toggle;
    private ListView mDevicesListView;
    private ImageView btn_forward;
    private ImageView btn_back;
    private ImageView btn_left;
    private ImageView btn_right;
    private BluetoothAdapter mBTAdapter;
    private Set<BluetoothDevice> mPairedDevices;
    private ArrayAdapter<String> mBTArrayAdapter;
    private Handler mHandler;
    private ConnectedThread mConnectedThread; // bluetooth background
worker thread to send and receive data
    private BluetoothSocket mBTSocket = null; // bi-directional
client-to-client data path

    @Override
    protected void onStart() {
        super.onStart();
        if (ContextCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION) !=
PackageManager.PERMISSION_GRANTED)
            ActivityCompat.requestPermissions(this, new
String[] {Manifest.permission.ACCESS_COARSE_LOCATION}, 1); }

    @Override

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        btn_toggle = findViewById(R.id.btn_toggle);
        btn_forward = findViewById(R.id.up);
        btn_back = findViewById(R.id.down);
        btn_left = findViewById(R.id.left);
        btn_right = findViewById(R.id.right);
        btn_vacuum = findViewById(R.id.vakum);
        btn_clean = findViewById(R.id.pel);
        btn_both = findViewById(R.id.both);
        mListPairedDevicesBtn = (Button) findViewById(R.id.paired_btn);

        mBTArrayAdapter = new ArrayAdapter<>(this,
android.R.layout.simple_list_item_1);
        mBTAdapter = BluetoothAdapter.getDefaultAdapter(); // get a
handle on the bluetooth radio
        mDevicesListView = (ListView)

```

```

findViewById(R.id.devices_list_view);
mDevicesListView.setAdapter(mBTArrayAdapter); // assign model to
view
mDevicesListView.setOnItemClickListener(mDeviceClickListener);
mHandler = new Handler(Looper.getMainLooper()) {
    @Override
    public void handleMessage(Message msg) {
        if (msg.what == MESSAGE_READ) {
            String readMessage = null;
            if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.KITKAT) {
                readMessage = new String((byte[]) msg.obj,
StandardCharsets.UTF_8); } }
            if (msg.what == CONNECTING_STATUS) {
                char[] sConnected;
                String message = "S";
                if (msg.arg1 == 1) {
                    Toast.makeText(getApplicationContext(), "Connected to " +
msg.obj, Toast.LENGTH_SHORT).show();
                    comand(message);
                }else {
                    Toast.makeText(getApplicationContext(), "Connection Failed",
Toast.LENGTH_SHORT).show(); } } } };

        btn_toggle.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
            @Override
            public void onCheckedChanged(CompoundButton compoundButton,
boolean b) { if(b){
                bluetoothOn();
                compoundButton.setBackgroundColor(Color.BLUE);
            }else{
                bluetoothOff();
                compoundButton.setBackgroundColor(Color.RED);
            } } });

        btn_clean.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
            @Override
            public void onCheckedChanged(CompoundButton compoundButton,
boolean b) { if (b){
                compoundButton.setBackgroundColor(Color.BLUE);
                String message = "C";
                comand(message);
            }else{
                compoundButton.setBackgroundColor(Color.RED);
                String message = "c";
                comand(message);
            } } });

        btn_vacuum.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
            @Override
            public void onCheckedChanged(CompoundButton compoundButton,
boolean b) { if (b){
                compoundButton.setBackgroundColor(Color.BLUE);
                String message = "V";

```



```

comand(message);
    }else{
        compoundButton.setBackgroundColor(Color.RED);
        String message = "v";
        comand(message);
    } } });

    btn_both.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
    @Override
    public void onCheckedChanged(CompoundButton compoundButton,
boolean b) { if (b){
        compoundButton.setBackgroundColor(Color.BLUE);
        String message = "X";
        comand(message);
    }else{
        compoundButton.setBackgroundColor(Color.RED);
        String message = "x";
omand(message);
    } } });

    mListPairedDevicesBtn.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View v) {
        listPairedDevices();
    } });

    btn_forward.setOnTouchListener(new View.OnTouchListener() {
    @Override
    public boolean onTouch(View view, MotionEvent motionEvent) {
        switch (motionEvent.getAction()) {
            case MotionEvent.ACTION_DOWN:
                mHandler.postDelayed(mForward, 10);
                break;
            case MotionEvent.ACTION_UP:
                mHandler.removeCallbacks(mForward);
                String message="S";
                comand(message);
                break;
        } return true; }
Runnable mForward = new Runnable() {
    @Override
    public void run() {
        String message="F";
        comand(message);
        mHandler.postDelayed(this, 100);
    } }; });

    btn_back.setOnTouchListener(new View.OnTouchListener() {
    @Override
    public boolean onTouch(View view, MotionEvent motionEvent) {
        switch (motionEvent.getAction()) {
            case MotionEvent.ACTION_DOWN:
                mHandler.postDelayed(mBack, 10);
                break;

```

```

case MotionEvent.ACTION_UP:
    mHandler.removeCallbacks(mBack);
    String message="S";
    comand(message);
    break; }
return true;}
Runnable mBack = new Runnable() {
@Override
public void run() {

    String message="B";
    comand(message);
    mHandler.postDelayed(this, 100);
} }; });

btn_left.setOnTouchListener(new View.OnTouchListener() {
@Override
public boolean onTouch(View view, MotionEvent motionEvent) {
switch (motionEvent.getAction()) {
case MotionEvent.ACTION_DOWN:
    mHandler.postDelayed(mLeft, 10);
    break;
case MotionEvent.ACTION_UP:
    mHandler.removeCallbacks(mLeft);
    String message="S";
    comand(message);
    break; }
return true; }
Runnable mLeft = new Runnable() {
@Override
public void run() {
    String message="L";
    comand(message);
    mHandler.postDelayed(this, 100);
} }; });

btn_right.setOnTouchListener(new View.OnTouchListener() {
@Override
public boolean onTouch(View view, MotionEvent motionEvent) {
switch (motionEvent.getAction()) {
case MotionEvent.ACTION_DOWN:
    mHandler.postDelayed(mRight, 10);
    break;
case MotionEvent.ACTION_UP:
    mHandler.removeCallbacks(mRight);
    String message="S";
    comand(message);
    break;
} return true; }
Runnable mRight = new Runnable() {
@Override
public void run() {
    String message="R";
    comand(message);
    mHandler.postDelayed(this, 100);
} }; }); }

```

```

private void comand(String Message){
    if (mBTSocket != null) {
        try {
            // Create the command that will be sent to arduino.
            // String must be converted in its bytes to be sent on serial
            // communication
            mBTSocket.getOutputStream().write(Message.getBytes());
        } catch (IOException e) {
        } } }

private void bluetoothOn(){
    if (!mBTAdapter.isEnabled()) {
        Intent enableBtIntent = new
Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
        startActivityForResult(enableBtIntent, REQUEST_ENABLE_BT);

Toast.makeText(getApplicationContext(),getString(R.string.sBTturON
),Toast.LENGTH_SHORT).show(); }
        else{

Toast.makeText(getApplicationContext(),getString(R.string.BTison),
Toast.LENGTH_SHORT).show();
        } }

// Enter here after user selects "yes" or "no" to enabling radio

private void bluetoothOff(){
    mBTAdapter.disable(); // turn off
    Toast.makeText(getApplicationContext(),"Bluetooth turned Off",
Toast.LENGTH_SHORT).show();
}

final BroadcastReceiver blReceiver = new BroadcastReceiver() {
    @Override
    public void onReceive(Context context, Intent intent) {
        String action = intent.getAction();
        if(BluetoothDevice.ACTION_FOUND.equals(action)){
            BluetoothDevice device =
intent.getParcelableExtra(BluetoothDevice.EXTRA_DEVICE);
            // add the name to the list
            mBTArrayAdapter.add(device.getName() + "\n" +
device.getAddress());
            mBTArrayAdapter.notifyDataSetChanged();
        } } };

private void listPairedDevices(){
    if(mBTAdapter.isEnabled()) {
        AlertDialog.Builder builderSingle = new
AlertDialog.Builder(MainActivity.this);
        builderSingle.setIcon(R.drawable.ic_android_black_24dp);
        builderSingle.setTitle("Select One Name:-");
        final ArrayAdapter<String> arrayAdapter = new
ArrayAdapter<String>(MainActivity.this,
android.R.layout.select_dialog_singlechoice);
        mBTArrayAdapter.clear();
    }
}

```

```

mPairedDevices = mBTAdapter.getBondedDevices();
    for (BluetoothDevice device : mPairedDevices)
        arrayAdapter.add(device.getName() + "\n" +
device.getAddress());

    builderSingle.setNegativeButton("cancel", new
DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            dialog.dismiss();
        } });

    builderSingle.setAdapter(arrayAdapter, new
DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            if(!mBTAdapter.isEnabled()) {
                Toast.makeText(getBaseContext(),
getString(R.string.BTnotOn), Toast.LENGTH_SHORT).show();
                return; }
            String strName = arrayAdapter.getItem(which);
            final String address = strName.substring(strName.length() -
17);
            final String name = strName.substring(0, strName.length() -
17);
            new Thread(){
                @Override
                public void run() {
                    boolean fail = false;
                    BluetoothDevice device =
mBTAdapter.getRemoteDevice(address);
                    try {
                        mBTSocket = createBluetoothSocket(device);
                    } catch (IOException e) {
                        fail = true;
                        Toast.makeText(getBaseContext(),
getString(R.string.ErrSockCrea), Toast.LENGTH_SHORT).show();

                        } try {
                            mBTSocket.connect();
                        } catch (IOException e) {
                            try {
                                fail = true;
                                mBTSocket.close();
                                mHandler.obtainMessage(CONNECTING_STATUS, -1, -1)
                                    .sendToTarget();
                            } catch (IOException e2) {
                                //insert code to deal with this
                                Toast.makeText(getBaseContext(),
getString(R.string.ErrSockCrea), Toast.LENGTH_SHORT).show();
                            } }
                            if(!fail) {
                                mConnectedThread = new ConnectedThread(mBTSocket,
mHandler);
                                mConnectedThread.start();

```

```

        mHandler.obtainMessage(CONNECTING_STATUS, 1, -1, name)
            .sendToTarget();

        } } }.start(); } });
        builderSingle.show();
    }else{
        Toast.makeText(getBaseContext(), getString(R.string.BTnotOn),
Toast.LENGTH_SHORT).show(); } }

    private AdapterView.OnItemClickListener mDeviceClickListener =
new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int
position, long id) {
        if(!mBTAdapter.isEnabled()) {
            Toast.makeText(getBaseContext(), getString(R.string.BTnotOn),
Toast.LENGTH_SHORT).show();
            return; }
        // Get the device MAC address, which is the last 17 chars in
the View
        String info = ((TextView) view).getText().toString();
        final String address = info.substring(info.length() - 17);
        final String name = info.substring(0,info.length() - 17);

        // Spawn a new thread to avoid blocking the GUI one
        new Thread() {
            @Override

            public void run() {
                boolean fail = false;
                BluetoothDevice device = mBTAdapter.getRemoteDevice(address);
                try {
                    mBTSocket = createBluetoothSocket(device);
                } catch (IOException e) {
                    fail = true;
                    Toast.makeText(getBaseContext(),
getString(R.string.ErrSockCrea), Toast.LENGTH_SHORT).show(); }
                // Establish the Bluetooth socket connection.
                try {
                    mBTSocket.connect();
                } catch (IOException e) {
                    try {
                        fail = true;
                        mBTSocket.close();
                        mHandler.obtainMessage(CONNECTING_STATUS, -1, -1)
                            .sendToTarget();
                    } catch (IOException e2) {
                        //insert code to deal with this
                        Toast.makeText(getBaseContext(),
getString(R.string.ErrSockCrea), Toast.LENGTH_SHORT).show();
                    }
                }
                if(!fail) {
                    mConnectedThread = new ConnectedThread(mBTSocket, mHandler);
                    mConnectedThread.start();

                    mHandler.obtainMessage(CONNECTING_STATUS, 1, -1, name)

```

```

        .sendToTarget();
    } } }.start(); } };

    private BluetoothSocket createBluetoothSocket(BluetoothDevice
device) throws IOException {
    try {
        final Method m =
device.getClass().getMethod("createInsecureRfcommSocketToServiceRe
cord", UUID.class);
        return (BluetoothSocket) m.invoke(device, BT_MODULE_UUID);
    } catch (Exception e) {
        Log.e(TAG, "Could not create Insecure RFComm Connection",e);
    } return
device.createRfcommSocketToServiceRecord(BT_MODULE_UUID);
    } }

```