



PROYEK AKHIR - VE180626

**PERANCANGAN PERANGKAT PENGENDALI SISTEM HIDROLIK
PADA KENDARAAN GANILA 2.0**

Menik Nurhidayati
NRP 10311600000038

Pembimbing
Imam Arifin, S.T., M.T.
Dr. Berlian Al Kindhi, S.ST., M.T.
Husin Abubakar Alaydrus, S.T.

DEPARTEMEN TEKNIK ELEKTRO OTOMASI
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018



FINAL PROJECT - VE180626

**DESIGN OF HYDRAULIC SYSTEM CONTROL DEVICES ON GANILA
2.0 VEHICLE**

Menik Nurhidayati
NRP 10311600000038

Supervisors
Imam Arifin, S.T., M.T.
Dr. Berlian Al Kindhi, S.ST., M.T.
Husin Abubakar Alaydrus, S.T.

DEPARTEMENT OF ELECTRICAL AUTOMATION ENGINEERING
Vocational Faculty
Institut Teknologi Sepuluh Nopember
Surabaya 2018

PERNYATAAN KEASLIAN PROYEK AKHIR

Dengan ini penulis menyatakan bahwa isi sebagian maupun keseluruhan Proyek Akhir dengan judul :

“PERANCANGAN PERANGKAT PENGENDALI SISTEM HIDROLIK PADA KENDARAAN GANLA 2.0”

Adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang penulis akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, penulis bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 25 Juni 2019



Menik Nurhidayati
NRP 1031160000038

Halaman ini sengaja dikosongkan

**PERANCANGAN PERANGKAT PENGENDALI SISTEM
HIDROLIK PADA KENDARAAN GANILA 2.0**

PROYEK AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan
Memperoleh Gelar Ahli Madya Teknik
Pada
Departemen Teknik Elektro Otomasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember

Menyetujui:

Pembimbing 1

Imam Arifin, S.T., M.T.
NIP. 19730222 200212 1 001

Pembimbing 2

Dr. Berlian Al Kindhi, S.ST., M.T.
NPP. 1985201912087



**SURABAYA
JULI, 2019**

Halaman ini sengaja dikosongkan

**PERANCANGAN PERANGKAT PENGENDALI SISTEM
HIDROLIK PADA KENDARAAN GANILA 2.0
DI
PT BHIMASENA RESEARCH AND DEVELOPMENT
PROYEK AKHIR**

Disusun oleh : NRP. 10311600000038
Menik Nurhidayati

Menyetujui

Kepala *Human Resources Departemen,*

Pembimbing Lapangan,



Muhammad Nur Cahyo Utomo
NIK. 021042017

Husin Abubakar Alaydrus, S.T.
NIK. 020982017

Halaman ini sengaja dikosongkan

PERANCANGAN PERANGKAT PENGENDALI SISTEM HIDROLIK PADA KENDARAAN GANILA 2.0

Menik Nurhidayati
1031160000038

Pembimbing I : Imam Arifin, S.T., M.T.
Pembimbing II : Dr. Berlian Al Kindhi, S.ST., M.T.
Pembimbing III : Husin Abubakar Alaydrus, S.T.

ABSTRAK

Kendaraan Ganila 2.0 merupakan seri terbaru dari Ganila 1.0 yang memiliki dimensi lebih minimalis, sehingga pengguna tidak dapat masuk ke dalam kendaraan tersebut. Oleh karena itu, Ganila 2.0 menggunakan sistem hidrolik untuk menaikkan atap pada kabin belakang, agar pengguna dapat masuk ke dalam kendaraan tersebut. Pengoperasian sistem hidrolik pada Ganila 2.0 yang masih menggunakan *remote* kabel dinilai menyulitkan operator, karena harus mengambil *remote* di dekat panel, sehingga dibuatlah suatu perangkat pengendali sistem hidrolik secara *wireless* untuk mempermudah pengguna dalam mengoperasikan sistem hidrolik secara jarak jauh.

Perangkat pengendali sistem hidrolik terhubung dengan komponen elektrik panel hidrolik berupa *solid state relay*. Komponen hidrolik berupa *solenoid valve* dihubungkan pada *solid state relay*, sehingga ketika relay tersebut aktif maka *solenoid valve* juga aktif. Perangkat pengendali hidrolik dilengkapi dengan fitur berupa pengaktifan silinder hidrolik untuk menaikkan dan menurunkan atap kabin belakang, serta fitur tambahan untuk monitoring ketinggian silinder hidrolik melalui monitor. Komunikasi perangkat dengan panel hidrolik secara *wireless* menggunakan modul *bluetooth* HC-05.

Pengujian dilakukan dengan memastikan fungsi perangkat berjalan dengan baik dengan jarak pengoperasian yang efektif. Hasil pengujian menunjukkan bahwa perangkat mampu mengaktifkan silinder hidrolik pada jarak kurang lebih 15 meter. Durasi pengoperasian hidrolik menggunakan *remote wireless* 48,79% lebih lama dibandingkan *remote* kabel. Selain itu, monitor juga mampu menampilkan spesifikasi Ganila 2.0 dan ketinggian silinder hidrolik.

Kata kunci : Ganila 2.0, *bluetooth* HC-05, sistem hidrolik.

Halaman ini sengaja dikosongkan

DESIGN OF HYDRAULIC SYSTEM CONTROL DEVICES ON GANILA 2.0 VEHICLE

Menik Nurhidayati
1031160000038

Supervisors I : Imam Arifin, S.T., M.T.
Supervisors II : Dr. Berlian Al Kindhi, S.ST., M.T.
Supervisors III : Husin Abubakar Alaydrus, S.T.

ABSTRACT

The Ganila 2.0 vehicle is the latest series from Ganila 1.0 which has minimalist dimensions, so users cannot enter the vehicle. Therefore, Ganila 2.0 uses a hydraulic system to raise the roof of the rear cabin, so that users can enter the vehicle. The operation of the hydraulic system on the Ganila 2.0 which is still using the cable remote is considered to make it difficult for the operator, cause the operator have to take the remote near the panel, so, a wireless system control device is made to make it easier for users to operate a hydraulic system.

The hydraulic system control device is connected to the electric components of a hydraulic panel such as solid state relay. The hydraulic component such as solenoid valve is connected to a solid state relay, so that when the relay is active the solenoid valve is also active. The hydraulic control device is equipped with features such as activating a hydraulic cylinder to raise and lower the roof of the rear cabin, and additional features for monitoring the hydraulic cylinder height through the monitor. Communication of devices with hydraulic panels wirelessly using the HC-05 bluetooth module.

Tests are carried out by ensuring that the device functions properly with effective operating distance. The test results show that the device is able to activate a hydraulic cylinder at a distance of approximately 15 meters. The duration for hydraulic operation using wireless remote is 48.79% longer than cable remote. In addition, the monitor is also able to display the introduction of the Ganila 2.0 vehicle and the height of the hydraulic cylinder.

Keywords : Ganila 2.0, bluetooth HC-05, hydraulic system

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Puji syukur atas kehadiran Allah SWT, atas rahmat dan karunia-Nya penulis dapat menyelesaikan Proyek Akhir yang berjudul **“Perancangan Perangkat Pengendali Sistem Hidrolik pada Kendaraan Ganila 2.0”** untuk memenuhi persyaratan menyelesaikan pendidikan Diploma 3 di Departemen Teknik Elektro Otomasi, Fakultas Vokasi, Institut Teknologi Sepuluh Nopember Surabaya.

Proses penyusunan penelitian ini, penulis telah banyak didukung oleh berbagai pihak. Oleh karena itu, pada kesempatan ini penulis ingin mengucapkan terima kasih kepada seluruh keluarga yang selalu memberikan dukungan dan doa untuk keberhasilan dan kelancaran. Setelah itu, kepada bapak Bapak Imam Arifin, S.T., M.T. dan ibu Dr. Berlian Al Kindhi, S.ST., M.T. selaku dosen pembimbing universitas yang telah memberikan arahan serta nasihat kepada penulis. Selain itu, terima kasih juga kepada bapak Yudi Isvara, S.T. dan Husin Abubakar Alaydrus, S.T. selaku pembimbing lapangan di PT Bhimasena Research and Development. Tidak lupa kepada teman-teman anggota Laboratorium *Cyber Physical*, Otomasi dan Robot Industri angkatan 2016, 2017 dan 2018 atas pemberian semangat dan bantuan dalam proses penyelesaian Proyek Akhir.

Penulis menyadari bahwa buku Proyek Akhir ini belum sempurna, sehingga saran dan masukan sangat diharapkan untuk perbaikan kedepannya. Semoga buku ini dapat bermanfaat bagi pembaca dan masyarakat pada umumnya.

Surabaya, 25 Juni 2019

Menik Nurhidayati
NRP. 1031160000038

Halaman ini sengaja dikosongkan

DAFTAR ISI

LEMBAR PENGESAHAN	Error! Bookmark not defined.
LEMBAR PENGESAHAN PERUSAHAAN .	Error! Bookmark not defined.
ASBTRAK	v
ABSTRACT	vii
KATA PENGANTAR	ix
DAFTAR ISI	xi
DAFTAR GAMBAR	xiii
DAFTAR TABEL	xvii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan	2
1.5 Metodologi Penelitian	2
1.6 Sistematika Penulisan	3
BAB II GANILA 2.0 DAN <i>BLUETOOTH</i> HC-05	5
2.1 Ganila 2.0	5
2.1.1 Dimensi Ganila 2.0	5
2.1.2 Spesifikasi Teknis Kendaraan	6
2.1.3 Spesifikasi Teknis Dapur	7
2.1.4 Fitur Utama Kendaraan	8
2.1.5 Diagram Elektrik Panel Listrik	9
2.1.6 Diagram Elektrik Panel Hidrolik	10
2.1.7 Komponen Panel Hidrolik	11
2.2 Sistem Komunikasi <i>Wireless</i>	15
2.2.1 Modul <i>Bluetooth</i> HC-05	16
2.3 Arduino Mega 2560	17
2.3.1 Spesifikasi	18
2.3.2 Variabel	19
2.3.3 Struktur	19
2.3.4 <i>Library</i>	20
2.4 <i>Graphic User Interface</i> (GUI)	20
2.4.1 Monitor Nextion	21
2.4.2 <i>Software</i>	24
BAB III PERANCANGAN <i>REMOTE WIRELESS</i>	30
3.1 Perancangan <i>Remote Wireless</i>	31

3.1.1	Pemilihan Komponen.....	31
3.1.2	<i>Layout</i> Peletakkan Komponen.....	38
3.1.3	Desain Casing <i>Remote</i>	40
3.2	Pembuatan Desain Tampilan Monitor.....	42
3.3	Perakitan Komponen Penyusun <i>Remote (Assembly)</i>	48
3.4	Konfigurasi sistem	49
BAB IV	PENGUJIAN DAN ANALISA <i>REMOTE WIRELESS</i>	53
4.1	Cara Pengujian <i>Remote Wireless</i>	53
4.2	Cara Pengujian <i>Remote</i> Kabel	55
4.3	Analisa Hasil Pengujian <i>Remote Wireless</i> dengan <i>Remote</i> Kabel	56
BAB V	PENUTUP	61
5.1	Kesimpulan.....	61
5.2	Saran.....	61
DAFTAR PUSTAKA.....		63
LAMPIRAN		
DAFTAR RIWAYAT HIDUP		

DAFTAR GAMBAR

Gambar 2.1 Dimensi Ganila 2.0 tampak depan atap tertutup	5
Gambar 2.2 Dimensi Ganila 2.0 tampak depan atap terbuka.....	5
Gambar 2.3 Tata letak peralatan memasak pada kendaraan Ganila 2.0	7
Gambar 2.4 Digram elektrik panel listrik Ganila 2.0	9
Gambar 2.5 Diagram elektrik panel listrik hidrolis	10
Gambar 2.6 <i>Layout</i> komponen elektrik pada <i>box</i> panel listrik hidrolis	11
Gambar 2.7 <i>Miniature circuit breaker</i> (MCB) 3 fasa [4].....	12
Gambar 2.8 <i>Miniature circuit breaker</i> (MCB) 1 fasa [4].....	12
Gambar 2.9 Konstruksi <i>miniature circuit breaker</i> (MCB)	12
Gambar 2.10 <i>Thermal overload</i> relay.....	14
Gambar 2.11 Relay 8 pin	15
Gambar 2.12 Modul <i>bluetooth</i> HC-05.....	17
Gambar 2.13 Arduino Mega 2560.....	18
Gambar 2.14 <i>Layout board</i> Arduino Mega 2560 [18].....	19
Gambar 2.15 Monitor Nextion tampak depan.....	21
Gambar 2.16 Tampilan <i>software</i> Nextion editor.....	24
Gambar 2.17 Pemilihan jenis monitor yang digunakan.....	25
Gambar 2.18 Bagian-bagian <i>software</i>	25
Gambar 2.19 Sub menu <i>file</i>	26
Gambar 2.20 Bagian-bagian <i>Components pane</i>	27
Gambar 2.21 <i>User event code</i> menu <i>display</i>	28
Gambar 2.22 <i>User event code</i> <i>button</i>	28
Gambar 2.23 <i>User event code</i> <i>button slider</i>	28
Gambar 3.1 Alur kerja pengendali sistem hidrolis	30
Gambar 3.2 Blok diagram pembagian tugas Proyek Akhir	30
Gambar 3.3 Blok diagram <i>remote wireless</i>	31
Gambar 3.4 Desain <i>schematic socket</i> pin Arduino Mega 2560	32
Gambar 3.5 Desain <i>socket</i> pin Arduino Mega 2560 bagian bawah	33
Gambar 3.6 Desain <i>socket</i> pin Arduino Mega 2560 bagian atas.....	33
Gambar 3.7 Hasil cetak <i>board</i> bagian bawah	34
Gambar 3.8 Hasil cetak <i>board</i> bagian atas	34
Gambar 3.9 Hasil <i>board</i> beserta <i>socket</i> jst-xh tampak atas	35

Gambar 3.10 Hasil <i>board</i> beserta <i>socket</i> jst-xh tampak bawah.....	35
Gambar 3.11 <i>Layout</i> komponen bagian bawah	38
Gambar 3.12 <i>Layout</i> komponen bagian atas.....	39
Gambar 3.13 <i>Layout</i> komponen bagian depan (1).....	39
Gambar 3.14 <i>Layout</i> komponen bagian depan (2).....	40
Gambar 3.15 Sketsa desain <i>remote</i>	40
Gambar 3.16 Desain 3D <i>remote</i>	41
Gambar 3.17 Desain 3D <i>remote</i> tampak atas	41
Gambar 3.18 Desain 3D <i>remote</i> tampak bawah	41
Gambar 3.19 Desain 3D <i>remote</i> tampak kiri	41
Gambar 3.20 Gambar desain <i>remote</i> tampak atas.....	42
Gambar 3.21 Gambar desain <i>remote</i> depan.....	42
Gambar 3.22 Tampilan monitor <i>slide</i> 1.....	43
Gambar 3.23 Tampilan monitor <i>slide</i> 2.....	43
Gambar 3.24 Tampilan monitor <i>slide</i> 3.....	44
Gambar 3.25 Tampilan monitor <i>slide</i> 4.....	44
Gambar 3.26 Tampilan monitor <i>slide</i> 5.....	45
Gambar 3.27 Tampilan monitor <i>slide</i> 6.....	45
Gambar 3.28 Tampilan monitor <i>slide</i> 7.....	46
Gambar 3.29 Tampilan monitor <i>slide</i> 8.....	46
Gambar 3.30 Tampilan monitor <i>slide</i> 9.....	47
Gambar 3.31 Tampilan monitor <i>slide</i> 10.....	47
Gambar 3.32 Tampilan monitor <i>slide</i> 11.....	48
Gambar 3.33 Tampilan monitor <i>slide</i> 12.....	48
Gambar 3.34 <i>Charging</i> baterai 3 sel menggunakan <i>balancer</i>	49
Gambar 3.35 Pemasangan komponen dan <i>wiring</i> kabel	49
Gambar 3.36 <i>Listing</i> program konfigurasi <i>bluetooth</i>	50
Gambar 3.37 <i>Listing</i> program <i>master bluetooth</i>	51
Gambar 3.38 <i>Listing</i> program <i>slave bluetooth</i>	51
Gambar 4.1 Pengujian kerja <i>remote</i> menggunakan <i>project board</i>	53
Gambar 4.2 Pengujian kerja <i>remote</i> dengan kendaraan Ganila 2.0	54
Gambar 4.3 Pengujian <i>remote</i> dengan jarak 2 meter	55
Gambar 4.4 Tampilan hasil pengukuran sensor.....	55
Gambar 4.5 Pengujian <i>remote</i> kabel dengan jarak 1 meter dan 3 meter	56

Gambar 4.6 Grafik perbandingan waktu pengoperasian hidrolik antara <i>remote wireless</i> dengan kabel	58
Gambar 4.7 <i>Remote</i> kabel hidrolik	59
Gambar 4.8 <i>Remote wireless</i> hidrolik.....	59

Halaman ini sengaja dikosongkan

DAFTAR TABEL

Tabel 2.1 Spesifikasi kendaraan Ganila 2.0	6
Tabel 2.2 Daftar peralatan memasak pada kendaraan Ganila 2.0	8
Tabel 2.3 Spesifikasi Arduino Mega 2560.....	18
Tabel 2.4 Spesifikasi monitor Nextion	21
Tabel 2.5 Karakteristik elektronik monitor Nextion	22
Tabel 2.6 Performa tampilan monitor Nextion	23
Tabel 2.7 Fitur memori monitor Nextion.....	23
Tabel 3.1 Fungsi tombol pada <i>remote</i>	36
Tabel 4.1 Hasil pengujian <i>remote wireless</i>	56
Tabel 4.2 Hasil pengujian <i>remote</i> kabel	58

Halaman ini sengaja dikosongkan

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kendaraan Ganila 1.0 merupakan truk yang berisi peralatan memasak dan digunakan untuk kegiatan memasak prajurit di lapangan, kendaraan tersebut dibuat untuk memenuhi kebutuhan logistik tentara Indonesia [1].

Seiring dengan kebutuhan kendaraan Ganila 1.0 yang berada di daerah luar pulau atau terpencil, maka dibutuhkan sebuah jalur distribusi yang cepat dan efisien. Pendistribusian melalui jalur udara dinilai lebih cepat dan efisien dengan cara memasukkan kendaraan Ganila 1.0 ke dalam pesawat Hercules C130 H/HS. Cara ini dilakukan oleh TNI Angkatan Udara guna memaksimalkan kinerja pengiriman Ganila 1.0. Namun terdapat kendala dalam memasukkan kendaraan tersebut ke dalam pesawat Hercules yaitu dimensi ukuran yang terlalu besar. Oleh karena itu, dibuatlah jenis Ganila baru dengan desain yang lebih minimalis sesuai permintaan TNI Angkatan Udara.

Ganila 2.0 dapat dimuat ke dalam kargo pesawat Hercules C130 H/HS [2]. Dimensi yang dibuat memiliki tinggi dari lantai ke atap sekitar 1320 mm. Dikarenakan tinggi tersebut, orang tidak dapat masuk ke dalam kendaraan Ganila 2.0. Agar orang dapat masuk ke dalam kendaraan tersebut tanpa mengubah bentuk awal Ganila 2.0, maka digunakanlah sistem hidrolik untuk menaikkan atap pada kabin belakang. Silinder hidrolik ditempatkan pada tiap sudut kabin belakang kendaraan Ganila 2.0 dengan ketinggian silinder maksimal 660 mm. Ketika atap dapat dinaikkan menggunakan silinder hidrolik, maka orang dapat masuk ke dalam kabin belakang kendaraan Ganila 2.0.

Sistem elektrik hidrolik dibuat dalam satu panel dan dioperasikan menggunakan *remote* kabel yang juga terhubung ke panel. Sebelum mengaktifkan hidrolik, operator harus mengaktifkan MCB (*Miniature Circuit Breaker*) pada panel secara manual lalu menaikkan silinder hidrolik dengan cara menekan tombol *up* pada *remote* kabel dengan panjang kabel sebesar 5 meter. Penggunaan *remote* kabel dinilai kurang efisien, karena operator harus mengambil *remote* kabel yang berada di dekat panel kemudian mengoperasikan hidrolik dengan jarak maksimal 5 meter, dan *remote* tidak *portable*. Untuk itu, dibutuhkan sebuah perangkat untuk mengoperasikan hidrolik secara nirkabel atau *wireless*.

1.2 Rumusan Masalah

Pengoperasian sistem hidrolik pada kendaraan Ganila 2.0 masih menggunakan *remote* kabel yang terhubung ke panel hidrolik. Hal ini menyulitkan operator karena harus mengambil *remote* kabel di dekat panel terlebih dahulu. Selain itu, *remote* hanya berfungsi untuk mengaktifkan silinder hidrolik, sedangkan operator tidak bisa memonitor kondisi hidrolik pada kendaraan ketika aktif.

1.3 Batasan Masalah

Perangkat dibuat untuk mengoperasikan sistem hidrolik secara *wireless*. Beberapa fitur pada perangkat yang akan dibuat yaitu tombol untuk menaikkan dan menurunkan silinder hidrolik, tombol *emergency* untuk memutus rangkaian hidrolik sementara, dan monitor untuk menampilkan data ketinggian silinder hidrolik.

1.4 Tujuan

Perangkat pengendali sistem hidrolik secara *wireless* bertujuan untuk mempermudah operator dalam mengoperasikan sistem hidrolik yang terpasang pada kendaraan Ganila 2.0, selain itu operator juga dapat mengetahui ketinggian silinder hidrolik ketika aktif.

1.5 Metodologi Penelitian

Beberapa tahapan yang dilakukan dalam menyelesaikan Proyek Akhir mengenai perancangan perangkat pengendali sistem hidrolik pada kendaraan Ganila 2.0 yaitu survei data atau identifikasi masalah, studi literatur, perencanaan dan pembuatan alat, pengujian, dan penyusunan laporan.

Pada tahap identifikasi masalah dilakukan survei hidrolik pada kendaraan Ganila 2.0 yang sedang dalam proses pembuatan di bengkel PT. Bhimasena Research and Development. Pengoperasian sistem hidrolik pada Ganila 2.0 masih menggunakan *remote* kabel, sehingga diperlukan suatu perangkat pengoperasian secara *wireless* untuk mempermudah operator dalam mengoperasikan sistem hidrolik tersebut.

Selanjutnya untuk menunjang pengerjaan Proyek Akhir diperlukan studi literatur mengenai rangkaian panel hidrolik pada Ganila 2.0, sistem komunikasi *wireless*, komponen elektronika, *graphical user interface* (GUI), dan bahasa pemrograman.

Tahap perencanaan dan pembuatan alat adalah mendesain suatu perangkat pengendalian hidrolik dan menentukan komponen

penyusunnya. Dalam menentukan komponen yang akan digunakan, terlebih dahulu memahami cara kerja perangkat *wireless* yang akan dibuat, selanjutnya membuat alur kerja perangkat *wireless*. Pada kegiatan pembuatan perangkat *wireless* dibutuhkan data spesifikasi komponen penyusun yang digunakan untuk prosedur pengajuan barang. Selain itu juga dilakukan pemrograman menggunakan Arduino Mega 2560 untuk memfungsikan perangkat yang akan dibuat.

Tahap selanjutnya yaitu pengujian perangkat yang dilakukan di bengkel PT. Bhimasena Research and Development. Pengujian dilakukan dengan memfungsikan perangkat *wireless* untuk mengoperasikan hidrolik dan menggunakan parameter jarak untuk mengetahui jangkauan pengoperasian hidrolik. Setelah menyelesaikan tahap pengujian, maka kegiatan selanjutnya adalah penulisan laporan Proyek Akhir agar semua kegiatan dan data-data yang didapatkan dapat tercatat lengkap.

1.6 Sistematika Penulisan

Pembahasan Proyek Akhir ini akan dibagi menjadi lima Bab dengan sistematika sebagai berikut :

BAB I PENDAHULUAN

Bab ini membahas latar belakang, rumusan masalah, batasan masalah, tujuan, metodologi penelitian, dan sistematika penulisan Proyek Akhir.

BAB II DASAR TEORI

Beberapa teori penunjang yang dibutuhkan untuk pengerjaan perangkat ini mengenai rangkaian panel hidrolik pada Ganila 2.0, sistem komunikasi *wireless*, *graphical user interface* (GUI), dan bahasa pemrograman.

BAB III PERANCANGAN DAN PEMBUATAN ALAT

Tahap perancangan dan pembuatan alat berupa perangkat *wireless* dijelaskan dalam bab ini dimulai dari pembahasan cara kerja perangkat, desain, komponen penyusun, dan langkah-langkah pembuatan perangkat *wireless*, serta penulisan program untuk mikrokontroler.

BAB IV PENGUJIAN DAN ANALISIS

Hasil perangkat *wireless* yang telah dibuat kemudian diuji dengan parameter jarak untuk mengetahui fungsi perangkat dapat bekerja dengan baik dan benar. Prosedur

pengujian juga dijelaskan dalam bab ini. Setelah melakukan pengujian, hasil tersebut dianalisis.

BAB V

KESIMPULAN DAN SARAN

Bab ini membahas kesimpulan dari perancangan dan pembuatan perangkat *wireless*, harapan dibuatnya perangkat tersebut, dan saran untuk pengembangan selanjutnya.

BAB II

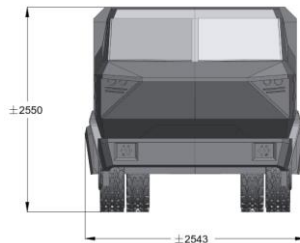
GANIILA 2.0 DAN *BLUETOOTH* HC-05

2.1 Ganila 2.0

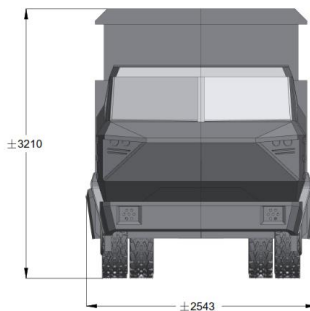
Kendaraan *Kitchen Truck* atau Ganila 2.0 merupakan kendaraan yang didesain, diperlengkap dan berfungsi sebagai dapur lapangan sehingga mampu dengan cepat memberikan dukungan logistik bagi seluruh prajurit [2].

2.1.1 Dimensi Ganila 2.0

Seri terbaru Ganila 2.0 memiliki dimensi yang lebih minimalis. Ketika silinder hidrolik tidak naik, tinggi kendaraan diukur dari roda hingga atap sebesar 2550 mm yang dapat dilihat pada Gambar 2.1. Besar ketinggian tersebut belum dikurangi diameter roda hingga lantai dasar kabin belakang. Sehingga ketinggian ruang kabin belakang ketika silinder hidrolik belum naik sekitar 1320 mm.



Gambar 2.1 Dimensi Ganila 2.0 tampak depan atap tertutup



Gambar 2.2 Dimensi Ganila 2.0 tampak depan atap terbuka

Sedangkan panjang silinder hidrolik dapat diketahui dari ketinggian keseluruhan kendaraan (dapat dilihat pada Gambar 2.2) ketika atap dinaikkan dikurangi dengan ketinggian kendaraan ketika atap belum dinaikkan. Sehingga panjang maksimal silinder ketika naik sekitar 660 mm.

2.1.2 Spesifikasi Teknis Kendaraan

Data informasi kendaraan Ganila 2.0 sangat diperlukan konsumen untuk mengetahui spesifikasi dari produk itu sendiri. Dalam hal ini data spesifikasi yang dicantumkan meliputi ukuran dimensi, material yang digunakan dalam membuat kerangka kendaraan, kemampuan kendaraan, dan lainnya (dapat dilihat pada Tabel 2.1).

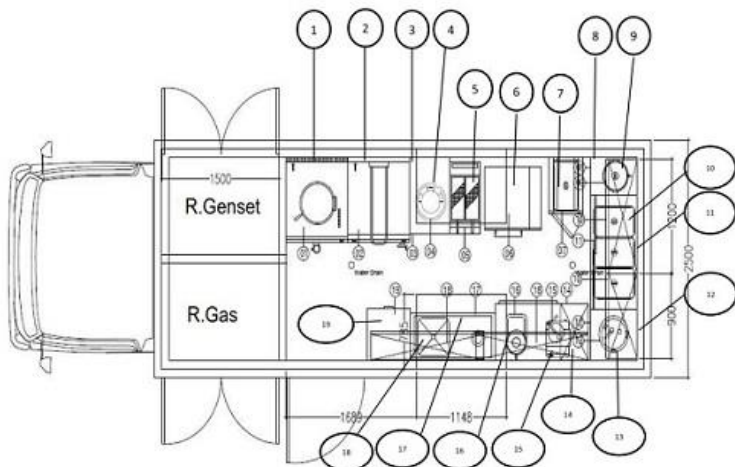
Tabel 2.1 Spesifikasi kendaraan Ganila 2.0

Kecepatan maks. (km/jam)	101
Tenaga maks. (PS/rpm)	235/2500
Torsi maks. (kgm/rpm)	72/1500
<i>Overall length</i>	± 8650 mm
<i>Overall width</i>	± 2600 mm
<i>Overall height (mobile mode)</i>	± 2500 mm
<i>Overall height (cooking mode)</i>	± 3200 mm
<i>Gradeability (tan %)</i>	47
GVWR	15.100 kg
<i>Vehicle weight net</i>	± 10.000 kg
<i>Number of passanger</i>	3
<i>Number of tyre</i>	6+1
<i>Drive system</i>	rear, 4x4 (optional)
Terbuat dari plat baja ketebalan 1.2-1.8 mm dilapisi anti karat.	
Kendaraan memiliki pintu akses personil dan akses <i>loading</i> atau <i>unloading</i> perlengkapan & materil.	
Kendaraan dilengkapi <i>deck</i> di bagian belakang yang dapat digunakan personil untuk mengambil makanan, <i>deck</i> bersifat <i>foldable</i> .	
Kendaraan dilengkapi dengan <i>winch</i> kapasitas 5 ton.	

Kendaraan dilengkapi dengan instrumen radio atau alat komunikasi.
Kendaraan dilengkapi dengan instrumen sirine.
Kendaraan dilengkapi dengan lampu <i>flat bar rotator</i> di beberapa posisi.
Kendaraan dilengkapi dengan lampu ketinggian atau <i>ouliner</i> .
Kendaraan dilengkapi dengan kamera mundur.

2.1.3 Spesifikasi Teknis Dapur

Kendaraan Ganila 2.0 berisi peralatan memasak yang digunakan oleh prajurit di lapangan. Nantinya peralatan memasak ini langsung digunakan dengan menggunakan sumber listrik dari genset. Untuk memudahkan prajurit dalam menggunakan peralatan tersebut, maka perlu disusun denah penempatan perabotan memasak yang dapat dilihat pada Gambar 2.3 beserta keterangan penomorannya yang tertera pada Tabel 2.2.



Gambar 2.3 Tata letak peralatan memasak pada kendaraan Ganila 2.0

Tabel 2.2 Daftar peralatan memasak pada kendaraan Ganila 2.0

No.	Nama Peralatan Dapur
1	<i>Gas Boiling Pan</i>
2	<i>Gas Tilting Pan</i>
3	<i>Counter Top Gas Stove</i>
4	<i>Counter Cabinet</i>
5	<i>Gas Fryer</i>
6	<i>Rice Cooker</i>
7	<i>Upright Cabinet</i>
8	<i>Work Counter</i>
9	<i>Coffee/Tea URN</i>
10	<i>Electric Bain Marie</i>
11	<i>Hanging Cabinet/Storage</i>
12	<i>Storage</i>
13	<i>Rice Warmer</i>
14	<i>UC Chiller</i>
15	<i>Slicer</i>
16	Blender
17	<i>Working Tabel</i>
18	<i>Single Sink</i>
19	<i>Freezer</i>

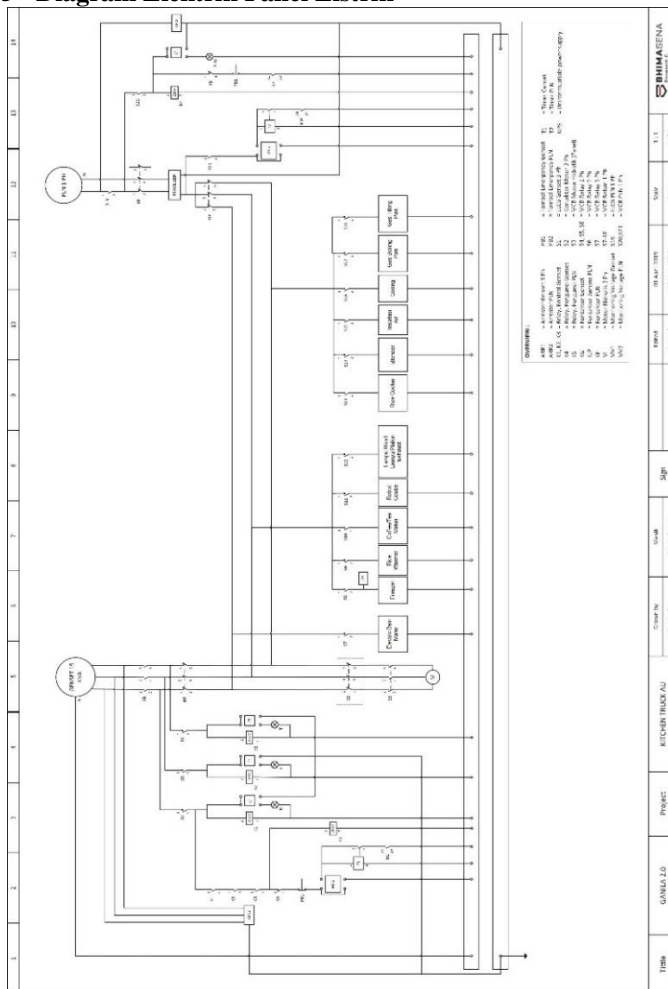
2.1.4 Fitur Utama Kendaraan

Setiap produk harus memiliki keunggulan tersendiri, keunggulan ini dapat dilihat dari aspek spesifikasi maupun fitur suatu produk. Berikut fitur utama kendaraan Ganila 2.0 :

- a. Kendaraan khusus dapur lapangan Ganila 2.0 memiliki kemampuan aspek muat dalam kargo pesawat Hercules C130 H/HS.

- b. *Compact Design* (sumber listrik, LPG, dan dapur terintegrasi dalam satu ruangan).
- c. Kapasitas masak 200-300 porsi dalam 3 jam di luar preparation.

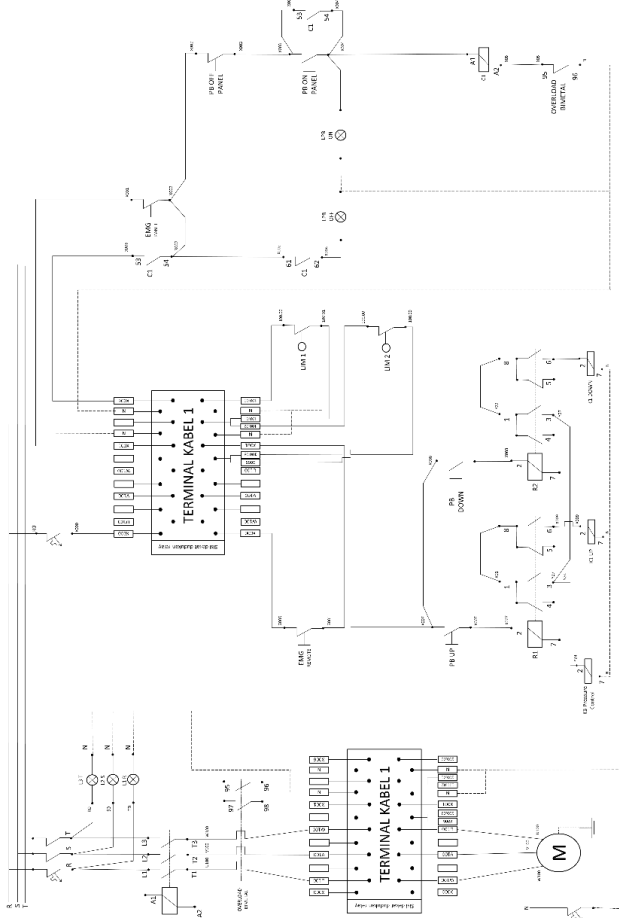
2.1.5 Diagram Elektrik Panel Listrik



Gambar 2.4 Diagram elektrik panel listrik Ganila 2.0

Sistem kelistrikan pada kendaraan Ganila 2.0 dibuat dalam satu panel utama yang terdiri dari beberapa komponen penting seperti *Miniature Circuit Breaker* (MCB), kontaktor, relay, arrester yang saling terhubung dalam rangkaian elektrikal yang ditunjukkan melalui Gambar 2.4.

2.1.6 Diagram Elektrik Panel Hidrolik

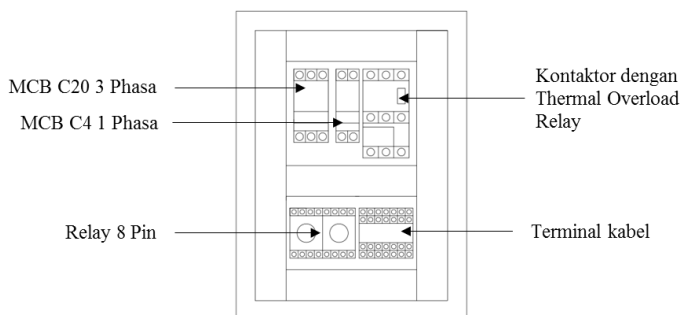


Gambar 2.5 Diagram elektrik panel listrik hidrolik

Sistem kelistrikan hidrolik yang terpasang dalam kendaraan Ganila 2.0 diatur dalam satu panel yang mana berbeda dengan panel utama kendaraan. Untuk wiring diagram elektrik panel hidrolik dapat dilihat pada Gambar 2.5, panel tersebut memiliki komponen elektrikal yang jumlahnya jauh lebih sedikit dibandingkan panel utama. Sumber listrik panel hidrolik berasal dari genset 3 fase.

2.1.7 Komponen Panel Hidrolik

Beberapa komponen elektrik yang digunakan pada sistem hidrolik seperti kontaktor, relay, dan *miniature circuit breaker* (MCB) diletakkan dalam satu panel khusus hidrolik. Panel ini berbeda dengan panel listrik utama untuk kendaraan Ganila 2.0. Dimensi panel hidrolik lebih kecil dibandingkan panel utama listrik Ganila 2.0 karena komponen yang digunakan juga sedikit. Peletakkan komponen elektrik pada panel hidrolik dapat dilihat pada Gambar 2.6.



Gambar 2.6 Layout komponen elektrik pada *box* panel listrik hidrolik

2.1.7.1 MCB 3 Phasa dan 1 Phasa

Miniature Circuit Breaker (MCB) adalah perangkat elektromekanik yang melindungi sebuah rangkaian listrik dari arus berlebih. Arus berlebih dalam sebuah rangkaian listrik biasanya disebabkan karena korsleting, kelebihan, atau kesalahan desain. Sebuah MCB merupakan alternatif yang lebih baik daripada *fuse* karena tidak memerlukan pengganti ketika *overload* terdeteksi. Tidak seperti *fuse*, MCB dapat dengan mudah di *reset* dan memberikan keamanan pengoperasian tanpa memerlukan biaya yang besar [3].

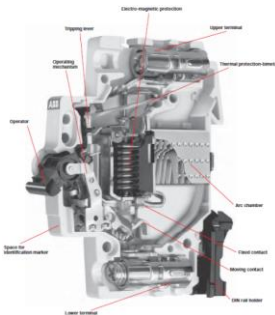


Gambar 2.7 *Miniature circuit breaker* (MCB) 3 phasa [4]



Gambar 2.8 *Miniature circuit breaker* (MCB) 1 phasa [4]

Gambar 2.7 menunjukkan jenis MCB 3 phasa sedangkan Gambar 2.8 adalah MCB 1 phasa. Penggunaan MCB jenis tertentu harus sesuai dengan beban yang digunakan. Selain itu batas arus MCB juga harus sesuai dengan beban. Untuk mengetahui lebih jelas mengenai konstruksi MCB dapat dilihat pada Gambar 2.9.



Gambar 2.9 Konstruksi *miniature circuit breaker* (MCB)
(Sumber : www.electrical-engineering-portal.com)

Dalam kasus *overload*, arus *overload* yang terukur didorong melalui MCB. Saat arus tersebut melewati bimetal strip, bimetal strip akan memanas dan memuai menjadi melengkung, dan melepaskan kait mekanis (*mechanical latch*). Waktu defleksi strip bimetal bergantung

pada jumlah arus yang mengalir melalui strip. Semakin tinggi arus semakin cepat bimetal strip akan melengkung. Selama korsleting, arus transien mengalir melalui *solenoid* memaksa pendorong menuju kait. Hal ini dapat melepaskan kait mekanis (*mechanical latch*) dan membuka *contact* dengan segera. MCB tersedia dalam beberapa jenis yaitu 3 kutub, 4 kutub, 1 kutub, dan 2 kutub [6].

2.1.7.2 Kontakor with *Thermal Overload Relay*

Starter motor magnetic adalah perangkat yang dioperasikan secara elektromagnetik yang mampu mengaktifkan dan menonaktifkan beban motor yang terhubung. Starter magnetik terdiri dari kontakor dan *thermal overload* relay yang memberikan perlindungan ketika terjadi kehilangan daya secara tiba-tiba. Kontakor hampir sama dengan relay, namun dirancang untuk mengalihkan daya listrik yang besar dan menangani beban tegangan tinggi. Tidak seperti relay, kontakor tidak memiliki *common* yang biasanya mengalihkan antara *normally open* dan *normally closed*. Kontakor terdiri dari *coil* yang berfungsi untuk menghubungkan listrik dari sumber menuju beban, sebuah elektromagnet yang biasanya disebut *coil* mampu memberikan gaya untuk menutup *contact* agar arus dapat mengalir, dan penutup terbuat dari bahan isolasi yang mampu untuk menyatukan antar bagian dan memberikan keamanan bagi seseorang yang menyentuh terminal. Umumnya kontakor dibuat *normally open* yang berarti bahwa listrik tidak akan mengalir sampai *coil* diaktifkan. Biasanya *coil* diaktifkan melalui operator, ataupun secara manual, yaitu seseorang dapat mengaktifkan *coil* dengan cara menekan tombol, atau secara otomatis menggunakan sebuah sensor, atau *timer* yang beralih ketika kondisi tertentu tercapai [7].

Thermal overload relay seperti pada Gambar 2.10 dirancang untuk melindungi motor atau beban lainnya dari kerusakan ketika korsleting listrik, atau kelebihan beban dan panas berlebih. *Overload* relay sederhana diaktifkan melalui panas yang ditimbulkan arus tinggi yang mengalir dan melalui jalur bimetal. Strip bimetal adalah dua pita logam yang memiliki koefisien panas yang berbeda. Ketika strip bimetal panas, maka satu bagian strip akan melengkung dibandingkan lainnya, dan menyebabkan perakitan melengkung. Karena bimetal dihubungkan ke rangkaian kontrol kontakor, maka ketika mencapai panas dan mengakibatkan strip melengkung maka secara efektif memutus rangkaian kontakor. Setelah strip bimetal kembali dingin

maka bentuk bimetal kembali ke semula, sehingga rangkaian kontaktor kembali terhubung [7].



Gambar 2.10 Thermal overload relay
(Sumber : www.omnical.co)

2.1.7.3 Relay

Relay dasar 8 pin adalah salah satu perangkat listrik yang paling umum digunakan saat ini. Relay tersedia dalam beberapa jenis yaitu 8 dan 11 pin, namun yang paling umum adalah 8 pin. Umumnya relay terdiri dari 5 bagian, yaitu : *coil*, *amature*, *contacts*, *base* yang terdiri dari *socket pin*, dan *frame (casing relay)*. Relay bekerja berdasarkan prinsip gaya elektromagnetik. Ketika *coil* diberi energi, maka menjadi bermagnet. *Armature* dibuat dari bahan ferromagnetik dan berada dekat *coil*. Pada relay 8 pin (Gambar 2.11), terdapat 2 *contact common*, 2 *normally open*, dan 2 *normally closed*. *Contacts* ini dibuat dari bahan konduktif seperti tembaga [9].

Koneksi pin *out* di dasar relay sebagai berikut :

Pin #1 = Kabel menuju *Common Contact #1*

Pin #2 = Kabel menuju *coil*

Pin #3 = Kabel *Normally Open Contact #1*

Pin #4 = Kabel *Normally Closed Contact #1*

Pin #5 = Kabel *Normally Closed Contact #2*

Pin #6 = Kabel *Normally Open Contact #2*

Pin #7 = Kabel *the Other end of the Relay Coil*

Pin #8 = Kabel menuju *coil*

Common contact membawa suplai tegangan yang dihubungkan ke perangkat listrik. Kondisi awal relay yang digunakan adalah *normally closed*. Ketika relay tidak aktif *contact common 1* terhubung ke *normally closed contact 1*, dan *contact common 2* terhubung ke *normally closed contact 2*. Ketika relay aktif situasinya berubah, *contact common 1* terhubung ke *normally open contact 1*, dan *contact common 2* terhubung ke *normally open contact 2* [9].



Gambar 2.11 Relay 8 pin
(Sumber : www.amazon.com)

2.2 Sistem Komunikasi *Wireless*

Jaringan tanpa kabel (*wireless*) pertama kali dikembangkan pada awal perindustrian. Sistem ini mengirimkan informasi lebih dari jarak garis pandang menggunakan sinyal asap, sinyal obor, cermin berkedip, sinyal suar, dan bendera pramuka.

Tujuan komunikasi nirkabel yang mendukung pertukaran informasi antara orang atau perangkat adalah batas komunikasi beberapa dekade mendatang, dan sebagian besar sudah ada dalam beberapa bentuk. Hal ini akan memungkinkan komunikasi media dari manapun di dunia untuk menggunakan perangkat genggam kecil atau laptop. Jaringan tanpa kabel akan menghubungkan palm laptop, laptop, dan tampilan komputer dimanapun dengan gedung kantor atau kampus, serta dari sudut cafe. Pada jaringan rumahan akan memungkinkan perangkat elektronik yang dapat saling berinteraksi dengan lainnya, dan dengan internet sebagai tambahannya mampu menyediakan konektivitas antara komputer, telepon, dan sistem monitoring atau pengamanan.

Salah satu sistem nirkabel saat ini adalah *bluetooth* dan *Zigbee*. Saat radio mengurangi harga dan konsumsi dayanya, berbagai jenis perangkat elektronik layak untuk menanamkan mereka, yang mana dapat digunakan sebagai bahan untuk membuat rumah pintar, jaringan sensor, dan aplikasi menarik lainnya. Dua jenis radio yang muncul untuk mendukung tren ini adalah *bluetooth* dan *zigbee*.

Bluetooth memberikan jarak koneksi yang pendek antara perangkat nirkabel dengan kemampuan jaringan yang belum sempurna. Standar *bluetooth* didasarkan pada microchip kecil yang menggabungkan sebuah pemancar-penerima dengan perangkat digital. Pemancar-penerima (*transceiver*) menggantikan penghubung kabel untuk perangkat seperti telepon seluler, laptop dan palm laptop, *printer*

portabel dan proyektor, dan jalur akses jaringan. *Bluetooth* terutama digunakan untuk jarak komunikasi yang pendek, seperti laptop ke *printer* sekitarnya, atau telepon seluler ke *headset*. Jarak operasi normal adalah 10 (pada 1 mW daya pancar), dan jarak ini dapat meningkat menjadi 100 m melalui penambahan daya pancar menjadi 100 mW. Operasi sistem dalam pita frekuensi 2.4 GHz tanpa izin, sehingga dapat digunakan di seluruh dunia tanpa masalah lisensi. Standar *bluetooth* menyediakan 1 data asinkron pada 723.2 Kbps. Pada *mode* ini juga dikenal sebagai *asynchronous connection-less*, atau ACL, ada saluran terbalik dengan kecepatan data sebesar 57.6 Kbps. Spesifikasi juga memungkinkan hingga 3 saluran sinkron setiap kecepatan 64 Kbps. Mode ini dikenal sebagai *synchronous connection oriented* atau SCO, ini terutama digunakan untuk pengaplikasian suara seperti *headset*, tetapi juga dapat digunakan untuk data. Perbedaan hasil *mode* ini adalah agregasi kecepatan bit sekitar 1 Mbps. Rute data asinkron dilakukan melalui sebuah paket *switching protocol* berdasarkan perubahan frekuensi (*frequency-hopping*) pada 1600 hops per detik.

Standar *bluetooth* dikembangkan secara bersama oleh 3 Com, Ericsson, Intel, IBM, Lucent, Microsoft, Motorola, Nokia, dan Toshiba. Standar diadopsi oleh lebih dari 1300 manufaktur, dan banyak konsumen produk elektronik yang menggabungkan *bluetooth*, termasuk *headset* nirkabel untuk telepon seluler, usb nirkabel, atau RS232, PCMCIA nirkabel, dan settop *boxes* nirkabel [11].

2.2.1 Modul *Bluetooth* HC-05

Bluetooth mudah digunakan dan dapat menghubungkan hingga tujuh perangkat pada satu waktu. Juga mampu dengan mudah untuk beralih antara perangkat atau yang ditemukan dan perangkat baru. Perangkat ini bekerja pada frekuensi tertentu, yang dinamakan slot waktu, dan berlangsung selama 635 us [12].

Modul komunikasi serial *bluetooth* HC-05 dirancang untuk pengaturan koneksi serial nirkabel (*wireless*) transparan. Memenuhi syarat sebagai *bluetooth* V 2.0 EDR 3 Mbps modulasi dengan 2.4 GHz pemancar dan penerima radio [13].

HC-05 mengadopsi modul *bluetooth slave-master*. Pengguna tidak perlu mempelajari protokol pengiriman modul *bluetooth*, menggunakan *instruction set* untuk menyelesaikan operasi yang sesuai, dan melalui AT *command* untuk mengubah *mode master-slave*.

Modul *bluetooth* HC-05 (Gambar 2.12) merupakan standar teknologi nirkabel. HC-05 mampu bertukar informasi secara nirkabel antar dua sistem sekitar jarak 10 meter. Pertukaran data terjadi menggunakan frekuensi gelombang radio *ultra-high* yang pendek dalam lebar *Industrial Scientific and Medical* (ISM) dari 2.4 ke 2.485 GHz dari perangkat *mobile* yang tetap dan jaringan area pribadi (PAN).



Gambar 2.12 Modul *bluetooth* HC-05
(Sumber : www.ram-e-shop.com)

Modul *slave bluetooth* HC-05 digunakan untuk menghubungkan perangkat *master* seperti *bluetooth* usb dan *handphone* android. Ketika daya diberikan pada modul *master*, maka dilakukan pencarian *bluetooth* HC-05 di area sekitar dan modul *slave* secara langsung terhubung dengan modul *master* [12].

2.3 Arduino Mega 2560

Perangkat *minimum system* berupa Arduino Mega 2560 (Gambar 2.13) adalah sebuah mikrokontroler berdasarkan ATmega 2560. Arduino Mega memiliki 54 pin digital *input* atau *output* yang mana 15 pin diantaranya dapat digunakan sebagai *output* PWM, 16 pin sebagai *input* analog, dan 4 pin sebagai UART (*serial port hardware*). 16 MHz osilator kristal, sebuah koneksi USB, *jack power*, *header* ICSP, dan tombol reset. Itu semua yang dibutuhkan untuk menunjang mikrokontroler. Cukup dengan menghubungkan Arduino Mega 2560 dengan sebuah usb komputer atau kabel daya yang mana terhubung dengan adapter AC-DC atau baterai dapat digunakan untuk mengoperasikannya. Arduino kompatibel dengan pelindung yang dirancang untuk Arduino Duemilanove atau Arduino Diecimila. Arduino Mega 2560 merupakan versi lama dari Arduino Mega [15].

Bahasa pemrograman Arduino dibagi menjadi tiga bagian yaitu, fungsi (*function*), nilai (*variables* dan *constant*), dan stuktur (*structure*) [16].



Gambar 2.13 Arduino mega 2560
(Sumber : store.arduino.cc)

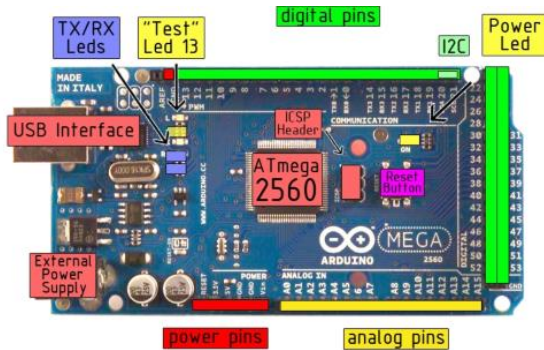
2.3.1 Spesifikasi Perangkat

Mikrokontroler Arduino memiliki berbagai jenis yang dapat dilihat dari jenis IC yang digunakan serta pin I/O. Dalam hal ini jenis Arduino yang digunakan adalah Mega 2560 yang memiliki spesifikasi seperti pada Tabel 2.3.

Tabel 2.3 Spesifikasi Arduino Mega 2560

Microcontroller	Atmega 2560
<i>Operating Voltage</i>	5 v
<i>Input Voltage (Recommended)</i>	7-12 v
<i>Input Voltage (Limits)</i>	6-20 v
Digital I/O Pins	54 (<i>Of Which 14 Provide PWM Output</i>)
Analog <i>Input</i> Pins	16
DC <i>Current</i> Per I/O Pin	40 Ma
Dc <i>Current</i> For 3.3v Pin	50 Ma
<i>Flash Memory</i>	256 KB <i>Of Which 8 KB Used By Bootloader</i>
SRAM	8 kb
EEPROM	4 kb
<i>Clock Speed</i>	16 MHZ

Bagian-bagian komponen Arduino Mega 2560 dapat dilihat pada Gambar 2.14. Pin I/O Arduino terpasang secara berurutan dengan yang lainnya.



Gambar 2.14 *Layout board* Arduino Mega 2560 [18]

2.3.2 Variabel

a. Void

Kata kunci void hanya digunakan dalam deklarasi fungsi. Ini menunjukkan bahwa fungsi tersebut diharapkan tidak mengembalikan informasi ke fungsi dari mana ia dipanggil [19].

b. Char

Tipe data yang digunakan untuk menyimpan nilai karakter. Literal karakter ditulis dalam tanda kutip tunggal, seperti ini : 'A' (untuk banyak karakter - *string* - gunakan tanda kutip ganda: "ABC"). Namun, karakter disimpan sebagai angka. Anda dapat melihat pengkodean spesifik dalam bagan ASCII. Ini berarti bahwa dimungkinkan untuk melakukan aritmatika pada karakter, di mana nilai ASCII karakter digunakan (misalnya 'A' + 1 memiliki nilai 66, karena nilai ASCII dari huruf kapital A adalah 65). Lihat referensi Serial.println untuk informasi lebih lanjut tentang bagaimana karakter diterjemahkan ke angka.

Ukuran tipe data char setidaknya 8 bit. Dianjurkan untuk hanya menggunakan karakter untuk menyimpan karakter. Untuk tipe data yang tidak ditandatangani, satu *byte* (8 bit), gunakan tipe data *byte* [20].

2.3.3 Struktur

a. If

Pernyataan *if* memeriksa suatu kondisi dan mengeksekusi pernyataan proses atau serangkaian pernyataan jika

kondisinya 'benar'. Kondisi boolean yaitu "benar" atau "salah" [21].

b. *#include*

Komponen C++ yang berguna untuk memberikan nama pada nilai yang konstan sebelum program dikompilasi. Konstanta yang ditentukan di Arduino tidak mengambil ruang memori program apa pun pada chip [22].

c. *Setup()*

Digunakan untuk menginisialisasi variabel, *mode* pin, *library*, dan lainnya. Fungsi *setup()* hanya akan berjalan sekali, setelah setiap *power up* atau *reset* Arduino [23].

d. *Return*

Hentikan fungsi dan kembalikan nilai dari fungsi ke fungsi panggilan, jika diinginkan [24].

2.3.4 *Library*

a. *Software serial*

Perangkat keras Arduino memiliki fitur pendukung bawaannya berupa komunikasi *serial* pada pin 0 dan 1 (yang juga masuk ke komputer melalui koneksi USB). Perangkat ini memungkinkan chip Atmega untuk menerima komunikasi serial bahkan saat mengerjakan tugas lain, selama ada ruang dalam buffer serial 64 *byte*.

Software Serial telah dikembangkan untuk memungkinkan komunikasi serial pada pin digital Arduino lainnya, menggunakan *software* untuk mereplikasi fungsionalitas (karenanya disebut "*SoftwareSerial*"). Dimungkinkan untuk memiliki beberapa *port* serial perangkat lunak dengan kecepatan hingga 115200 bps. Tidak semua pin Mega dan Mega 2560 mendukung perubahan interrupts, sehingga hanya beberapa yang bisa digunakan seperti : RX: 10, 11, 12, 13, 14, 15, 50, 51, 52, 53, A8 (62), A9 (63), A10 (64), A11 (65), A12 (66), A13 (67), A14 (68), A15 (69) [25].

2.4 *Graphic User Interface (GUI)*

Tampilan dengan monitor atau yang biasa dikenal dengan *Graphical user interface* adalah jenis antarmuka manusia-komputer pada komputer. Secara konseptual, antarmuka manusia-komputer merupakan sarana yang digunakan orang dan manusia untuk saling

berkomunikasi. Desain GUI yang bagus menghilangkan hambatan komunikasi dengan sistem komputer dan memungkinkan pengguna untuk bekerja secara langsung dengan masalah (objek) yang dihadapi. Dalam istilah ilmu komputer, GUI adalah tampilan operasi visual yang disajikan oleh monitor ke operator komputer. GUI biasanya memiliki windows, ikon, menu, dan tombol. [26].

2.4.1 Monitor Nextion

Nextion (Gambar 2.15) merupakan *Human Machine Interface* (HMI) yang memberikan kontrol antar muka dan visualisasi antara manusia dan proses, mesin, aplikasi, atau alat. Selanjutnya dapat diterapkan pada bidang IOT atau pengguna elektronik. Porduk ini merupakan solusi terbaik untuk mengganti LCD lama dan tabung LED. Dengan perangkat lunak Nextion Editor, pengguna dapat membuat dan mendesain antarmuka untuk tampilan Nextion. Seri Nextion Enhanced lebih kuat dibandingkan dengans seri Nextion Basic. Selain Micro Computer Unit (MCU) yang lebih cepat, fitur pendukung seri Enhanced: RTC bawaan, EEPROM 1K untuk data pengguna, 8 GPIO digital, kapasitas kilat lebih besar, dan lebih banyak RAM (untuk model 3.5 dan atasnya) [27].



Gambar 2.15 Monitor Nextion tampak depan
(Sumber : www.itead.cc)

a. Spesifikasi Bentuk

Data spesifikasi pada Tabel 2.4 digunakan dalam proses pembuatan *casing remote* dikarenakan memerlukan dimensi monitor Nextion.

Tabel 2.4 Spesifikasi monitor Nextion

	Data	Description
Color	64K 65536 colors	16 bit 565, 5R-6G-5B

<i>Layout size</i>	181(L) × 108(W) × 9.3(H)	NX8048K070
	181(L) × 108(W) × 9.3(H)	NX8048K070_011R
	181(L) × 108(W) × 9.3(H)	NX8048K070_011C
<i>Active Area (A.A.)</i>	164.90mm(L) × 100.00mm(W)	-
<i>Visual Area (V.A.)</i>	154.08mm(L) × 85.92mm(W)	-
<i>Resolution</i>	800 × 480 pixel	Also can be set as 480×800
<i>Touch type</i>	<i>Resistive</i>	

b. Karakteristik elektronik

Keterangan mengenai karakteristik elektronik monitor Nextion diperlukan untuk menentukan daya yang diperlukan untuk mengoperasikan monitor tersebut. Beberapa data elektronik yang dibutuhkan terangkum dalam Tabel 2.5.

Tabel 2.5 Karakteristik elektronik monitor Nextion

	<i>Test conditions</i>	Min	<i>Typical</i>	Max	Unit
<i>Operating voltage</i>		4.75	5	7	V
<i>Operating current</i>	VCC=+5V, Brightness is 100%	-	510	-	mA
	<i>SLEEP</i> Mode	-	15	-	mA
<i>Power supply recommend : 5V, 2.0 A, DC</i>					

c. Performa Tampilan

Untuk mengetahui jenis komunikasi data yang digunakan pada monitor Nextion, maka memerlukan data performa tampilan. Pada Tabel 2.6. dapat diketahui bahwa jenis komunikasi yang digunakan untuk mengubah desain tampilan monitor dari laptop dapat menggunakan USB TTL.

Tabel 2.6 Performa tampilan monitor Nextion

	<i>Test conditions</i>	Min	<i>Typical</i>	Max	Unit
<i>Serial port baudrate</i>	<i>Standard</i>	2400	9600	115200	Bps
<i>Output high voltage</i>	IO= -1mA	3.0	3.2		V
<i>Output low voltage</i>	IOL= -1mA		0.1	0.2	V
<i>Input high voltage</i>		2.0	3.3	5.0	V
<i>Input low voltage</i>		-0.7	0.0	1.3	V
<i>Serial port mode</i>	TTL				
<i>Serial port</i>	4 pin _ 2.54 mm				
<i>SD card socket</i>	Yes (FAT32 format), support maximum 32G Micro SD Card * microSD card socket is exclusively used to upgrade Nextion firmware /HMI design				

d. Fitur memori

Informasi mengenai data memori diperlukan untuk mengetahui besar kapasitas monitor yang mampu untuk menyimpan hasil desain. Seperti pada Tabel 2.7 menunjukkan memori yang digunakan sebanyak 32 MB, sehingga besar kapasitas *file* desain monitor tidak boleh lebih dari angka tersebut.

Tabel 2.7 Fitur memori monitor Nextion

<i>Memory type</i>	<i>Test conditions</i>	Min	<i>Typical</i>	Max	Unit
<i>Flash memory</i>	<i>Store fonts and images</i>	-	-	32	MB
<i>User storage</i>	EEPROM	-	-	1024	BYTE
<i>RAM memory</i>	<i>Store variables</i>	-	-	8192	BYTE

<i>Instruction buffer</i>	<i>Instruction buffer</i>	-	-	1024	BYTE
---------------------------	---------------------------	---	---	------	------

2.4.2 Software

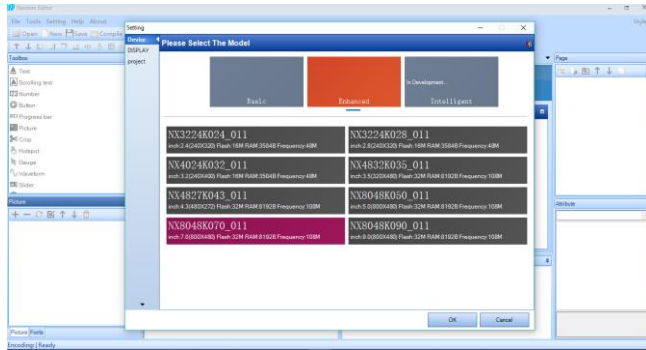
Nextion editor dapat digunakan untuk membuat *Graphical User Interface* (GUI) untuk perangkat *Human Machine Interface* (HMI) Nextion [28]. Tampilan *software* Nextion editor dapat dilihat pada Gambar 2.16.



Gambar 2.16 Tampilan *software* Nextion editor

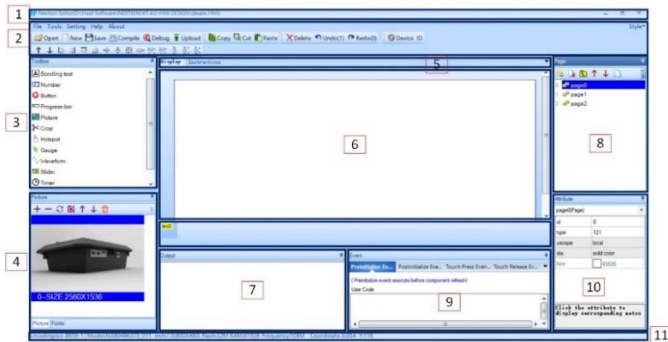
Software ini dapat diperoleh langsung melalui *website* Nextion secara gratis. Penggunaan *software* Nextion Editor, dapat dengan cepat digunakan untuk mengembangkan HMI GUI dengan komponen seperti grafik, teks, tombol, slider, dan lainnya. Dan instruksi berbasis teks ASCII untuk mengkode bagaimana komponen berinteraksi di sisi tampilan [29].

Klik *new*, dan pilih lokasi penyimpanan *file* desain monitor di folder yang diinginkan (seperti Gambar 2.17).



Gambar 2.17 Pemilihan jenis monitor yang digunakan

Memilih jenis monitor yang akan digunakan yaitu Nextion Enhanced NX8048K070_011.

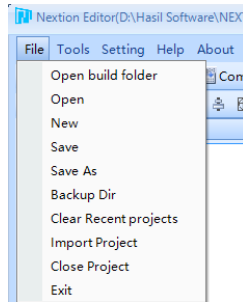


Gambar 2.18 Bagian-bagian *software*

Pada Gambar 2.18 berisi fitur-fitur yang terdapat dalam *software* Nextion Editor. Berikut keterangan fitur pada Gambar 2.18 :

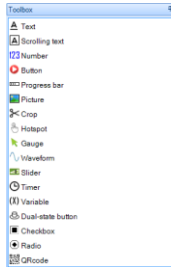
1. *Title bar*
Bagian ini merupakan *tittle bar* (judul) yang terdiri dari nama *file* HMI yang sedang dijalankan.
2. Main menu
 - a. Main menu
 - *File* menu
Bagian sub menu *file* pada Gambar 2.19 terdiri dari sub menu, yaitu : *New, Open, Save, Close, Exit, Import project, Clear*

recent projects, Recent projects, Open build folder, Backup dir.



Gambar 2.19 Sub menu *file*

- *Tool* menu
Bagian ini terdiri dari sub menu, yaitu : *Font generator*. Setiap atribut .txt atau .val yang akan digunakan pada *project* membutuhkan jenis huruf (*font*). Pembuatan ini dapat dilakukan melalui *Font Generator*.
 - *Setting* menu
Pengguna dapat mengubah pengaturan editor melalui menu *setting* atau mengatur tata letak *layout* melalui *Reset Layout*.
 - *Help* menu
 - *About* menu
- b. *Toolbar*
Beberapa sub menu dalam ikon ini, yaitu : *open, new, save, compile, debug, upload, copy, cut, paste, delete, undo, redo, device*, dan ID.
- c. *Component layout toolbar*
3. *Components pane*
Ikon ini terdiri dari beberapa sub menu, yaitu : *Text, scrolling text, number, button, progress bar, picture, crop, hotspot, gauge, waveform, slider, timer, variable, dual-state button, checkbox, radio*, dan QRcode seperti pada Gambar 2.18. Untuk menambahkan komponen tersebut ke dalam desain, cukup klik pada komponen yang dipilih, maka otomatis ditambahkan ke desain halaman.



Gambar 2.20 Bagian-bagian *Components pane*

4. *Picture resource pane*

Gambar yang akan dimasukkan pada HMI terlebih dahulu di *import* melalui *picture resources pane*. Untuk menambahkan gambar ke bagian ini dapat dilakukan dengan menekan simbol plus, untuk menghapus menekan simbol minus, untuk memindahkan urutan gambar dapat menekan simbol panah atas atau bawah.

5. *Display atau instruction tabs*

Bagian *instructon tabs* berfungsi untuk memuat instruksi pada Nextion Editor, sedangkan bagian *display* akan membuat pengguna kembali ke antarmuka utama Nextion Editor.

6. *Design canvas visual components*

Desain utama tampilan monitor dapat dibuat pada bagian ini. Pada tempat *Design canvas visual components* dapat dikreasikan dengan berbagai jenis tampilan.

7. *Output*

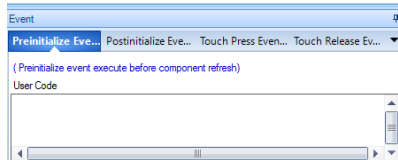
Proses untuk memilih *compile* atau *debug* atau *upload* dipilih dapat dipilih pada opsi ini. *Upload* sendiri digunakan untuk mengirim program ke perangkat tujuan.

8. *Page pane*

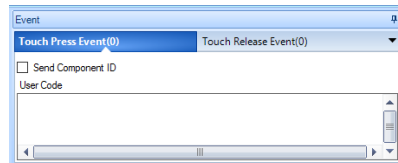
Setiap desain HMI paling sedikit membutuhkan satu *slide*. Untuk menambah beberapa *slide* yang dibutuhkan, dapat digunakan pilihan menu *Add* pada bagian *page pane*. Untuk menghapus *slide* dapat dipilih *delete*. Untuk mengatur urutan *slide* dapat di pilih simbol panah atas atau turun.

9. *User event code*

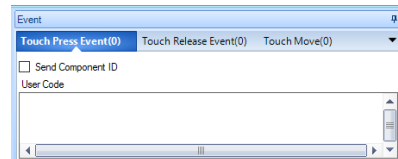
Pengguna dapat memberikan instruksi benar melalui bagian ini. Hampir setiap komponen pada *components pane* memiliki *touch press* dan *touch release*.



Gambar 2.21 *User event code menu display*



Gambar 2.22 *User event code button*



Gambar 2.23 *User event code button slider*

10. *Attributes pane*

Bagian ini berisi daftar pengaturan komponen yang dipilih pada desain HMI. Contoh memilih komponen gambar, maka beberapa atribut untuk mengatur gambar tersebut dapat dilakukan melalui bagian *Attributes pane*. Beberapa hal yang bisa diatur, contohnya letak koordinat gambar pada tampilan HMI, nama gambar, ukuran, dan lainnya.

11. Status bar

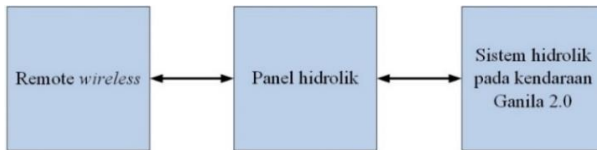
Bagian status bar terdiri dari tiga sub bagian yaitu pengkodean karakter, kecepatan nextion, dan koordinat desain HMI.

Halaman ini sengaja dikosongkan

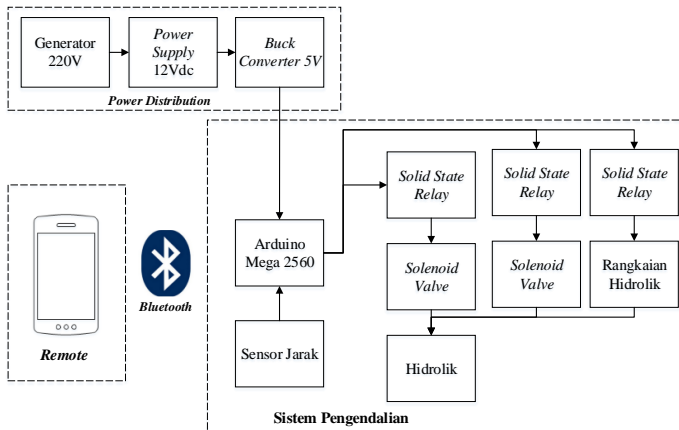
BAB III

PERANCANGAN *REMOTE WIRELESS*

Pengendali sistem hidrolik pada kendaraan Ganila 2.0 dibagi menjadi dua bagian yaitu, panel hidrolik dan perangkat *wireless*. Perangkat pengendali secara *wireless* yang dibuat berupa *remote wireless*. Panel hidrolik mendapat perintah dari *remote wireless* untuk mengaktifkan sistem hidrolik. Selain itu *remote wireless* juga menerima data ketinggian silinder hidrolik dari panel. Hasil pengukuran ketinggian silinder hidrolik diperoleh menggunakan sensor jarak, lalu diolah menjadi data siap kirim. Pengolahan data hasil pengukuran sensor dilakukan oleh bagian panel hidrolik. Setelah data siap, maka dikirimkan menuju *remote wireless* untuk ditampilkan pada monitor. Lebih jelasnya mengenai alur pengoperasian hidrolik dapat dilihat pada Gambar 3.1.



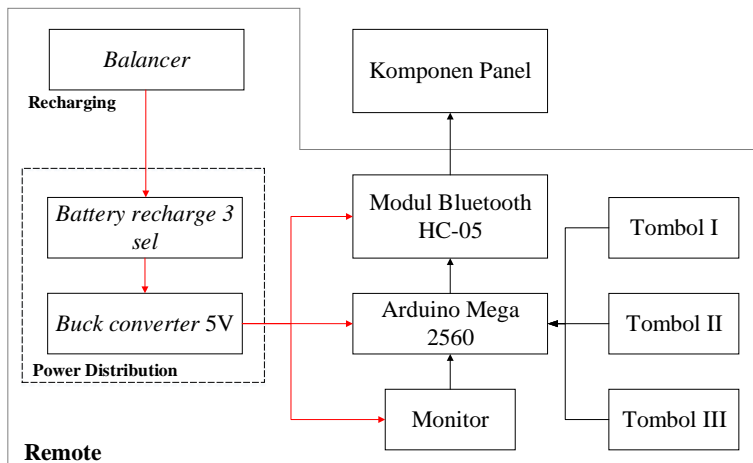
Gambar 3.1 Alur kerja pengendali sistem hidrolik



Gambar 3.2 Blok diagram pembagian tugas Proyek Akhir

3.1 Perancangan *Remote Wireless*

Operator dapat mengaktifkan sistem hidrolik pada kendaraan Ganila 2.0 menggunakan *remote wireless*. Sistem hidrolik dikontrol melalui panel hidrolik, panel tersebut mendapat perintah dari *remote wireless* yang dibawa oleh operator. Beberapa fitur atau fungsi dalam *remote* tersebut adalah tombol untuk menaikkan dan menurunkan silinder hidrolik pada kendaraan Ganila 2.0, tombol *emergency* untuk memutus aliran listrik pada rangkaian hidrolik yang digunakan sebagai *safety operation*, dan monitor untuk menampilkan pengenalan produk kendaraan Ganila 2.0, serta data ketinggian silinder hidrolik.



Keterangan :

- Tegangan
- Aliran data

Gambar 3.3 Blok diagram *remote wireless*

3.1.1 Pemilihan Komponen

Pada tahap perancangan *remote wireless* dilakukan pemilihan komponen penyusun *remote*. Komponen tersebut dipilih berdasarkan fungsi dan kegunaan di *remote*, serta mudah ditemukan dipasaran. Pemilihan komponen ini dibagi menjadi dua bagian yaitu, komponen utama dan komponen pengkabelan.

3.1.1.1 Komponen Utama

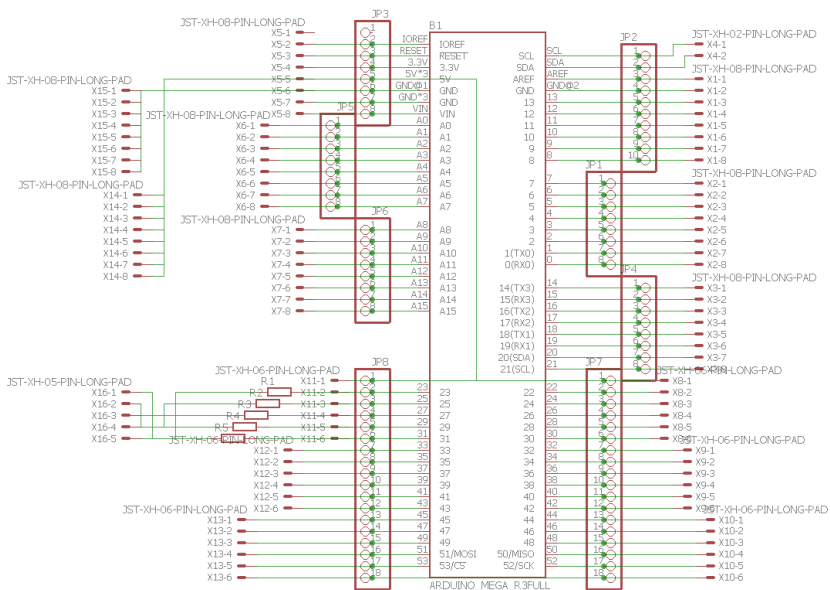
Beberapa komponen yang diperlukan dalam menyusun *remote wireless*, diantaranya :

a. Arduino Mega 2560

Komponen ini digunakan untuk memproses suatu perintah dari tombol *remote* menuju objek melalui program. Program tersebut dapat ditulis menggunakan *software* Arduino IDE.

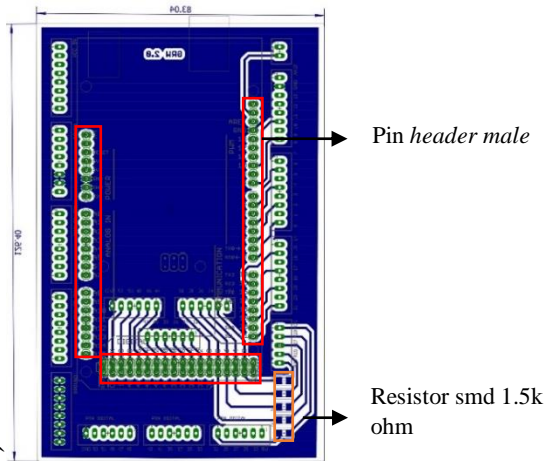
b. Board Socket Pin Arduino Mega 2560

Pada proses pengkabelan komponen penyusun *remote wireless* nantinya tidak langsung dihubungkan dengan pin Arduino Mega 2560, tetapi dihubungkan pada *socket* pengganti pin Arduino Mega 2560. *Socket-socket* tersebut tersusun pada board yang terhubung dengan Arduino Mega 2560.

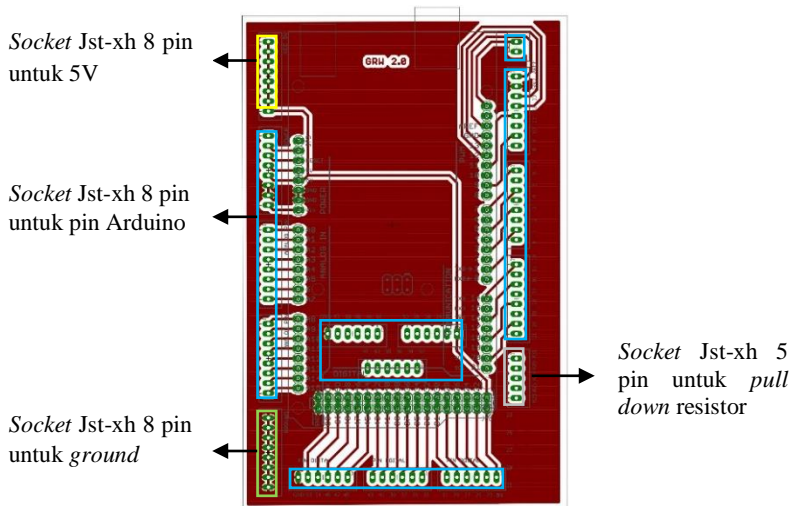


Gambar 3.4 Desain *schematic socket* pin Arduino Mega 2560

Beberapa komponen yang digunakan pada *schematic* di atas yaitu pin *header male*, konektor *jst-xh*, dan resistor *smd*.

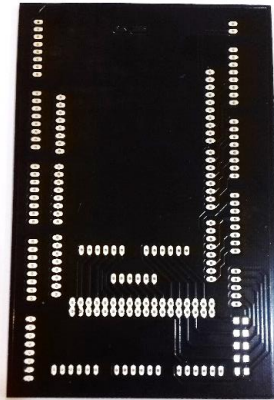


Gambar 3.5 Desain *socket* pin Arduino Mega 2560 bagian bawah



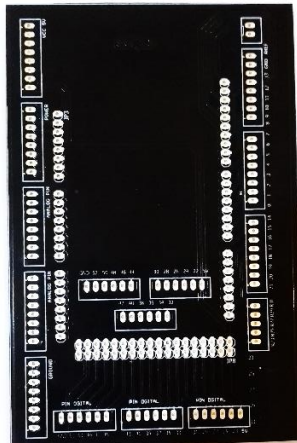
Gambar 3.6 Desain *socket* pin Arduino Mega 2560 bagian atas

Setelah proses desain PCB selesai, selanjutnya pencetakan pada *board*.



Gambar 3.7 Hasil cetak *board* bagian bawah

Desain PCB menggunakan jenis *double layer*, sehingga untuk bagian bawah PCB seperti Gambar 3.7 digunakan untuk penempatan kaki *socket JST-XH*, dan 5 buah resistor smd. Sedangkan untuk bagian atas PCB digunakan untuk penempatan badan *socket JST-XH* dan kaki pin *header male* seperti Gambar 3.8.



Gambar 3.8 Hasil cetak *board* bagian atas



Gambar 3.9 Hasil *board* beserta *socket jst-xh* tampak atas

Hasil PCB bagian atas yang telah diberi komponen dapat dilihat pada Gambar 3.9 yang mana terlihat hanya ada komponen *socket JST-XH*. Sedangkan untuk bagian bawah Gambar 3.10 PCB berisi pin *header male* dan 5 buah resistor smd.



Gambar 3.10 Hasil *board* beserta *socket jst-xh* tampak bawah

c. Modul *Bluetooth* HC-05

Komponen ini digunakan untuk menghubungkan *remote* dengan panel hidrolik agar dapat terhubung. Kaki pin HC-05 dihubungkan ke pin Arduino Mega 2560.

- d. Monitor Nextion 7 inch
Komponen ini digunakan untuk menampilkan data ketinggian silinder hidrolik ketika sedang dioperasikan dan sekilas pengenalan kendaraan Ganila 2.0.
- e. USB TTL CP 2102
Komponen ini digunakan untuk meng-*upload* desain tampilan monitor dari komputer menuju monitor Nextion 7 inch. Selain itu, kaki pin USB TTL CP 2102 juga dihubungkan ke Arduino Mega 2560 untuk mendapatkan data ketinggian silinder.
- f. *Push button*
Beberapa jenis *push button* juga digunakan dalam *remote wireless*, dimana setiap *push button* memiliki fungsi masing-masing dapat dilihat pada Tabel 3.1.

Tabel 3.1 Fungsi tombol pada *remote*

No.	Warna <i>push button</i>	Fungsi dalam <i>remote</i>
1.	Hitam	Tombol <i>power remote</i>
2.	Merah bulat	Tombol <i>up</i> silinder hidrolik
3.	Hijau bulat	Tombol <i>down</i> silinder hidrolik
4.	Merah kotak	Tombol <i>emergency</i> hidrolik

- g. Baterai
Jenis baterai yang digunakan adalah *recharge* dengan kapasitas 3.7 volt dengan arus 3800 mAh. Jumlah baterai yang digunakan sebanyak 3 buah dan dipasang seri, sehingga total tegangan menjadi sekitar 11 volt. Besar tegangan tersebut kemudian diturunkan menggunakan *buck converter* menjadi 5 volt.
- h. *Balancer*
Dalam proses *charging* baterai diperlukan penstabil tegangan yang mengisi baterai, agar setiap sel memiliki tegangan yang sama.

- i. *Buck converter*
Total tegangan baterai sebesar 11 volt dapat diturunkan menjadi 5 volt menggunakan perangkat ini. Tegangan yang telah diturunkan digunakan untuk mensuplai Arduino Mega 2560, monitor Nextion 7 inch, dan tombol.
- j. *Led*
Penggunaan indicator berupa LED bertujuan untuk menunjukkan koneksi *Bluetooth* pada *remote* sudah terhubung atau belum. Warna led yang digunakan adalah merah.
- k. *Heatsink*
Beberapa komponen elektrik tentunya dapat menimbulkan panas ketika sedang beroperasi, untuk mengatasi panas tersebut dapat digunakan *heatsink*. Komponen ini digunakan sebagai transfer panas yang dihasilkan oleh baterai. Dimensi *heatsink* yang digunakan adalah 14x14x6 mm, sehingga pada permukaan baterai seluas 65x53 mm membutuhkan 6 buah *heatsink*.

3.1.1.2Pengkabelan

Untuk menghubungkan antar komponen dibutuhkan beberapa komponen seperti berikut :

- a. Kabel AWG 22
Warna kabel yang digunakan beserta dengan fungsinya :

Merah	: Vcc atau 5V
Hitam	: <i>Ground</i>
Kuning	: rx
Biru	: tx
- b. *Socket Usb Male Tipe 2.0*
Dalam merapikan *wiring* kabel dapat digunakan jenis konektor ini sebagai penghubung baterai menuju konektor usb *female to female* tipe 2.0. dari konektor usb *female to female* tipe 2.0 akan dihubungkan ke charger baterai.
- c. Konektor Usb *Female To Female* Tipe 2.0
Tujuan dari digunakannya konektor jenis ini adalah sebagai penghubung antara baterai dengan charger baterai.

d. Konektor Pin JST-XH

Digunakan sebagai pengganti pin-pin Arduino. *Socket* ini akan di letakkan pada *board socket* Arduino Mega 2560 yang telah di desain.

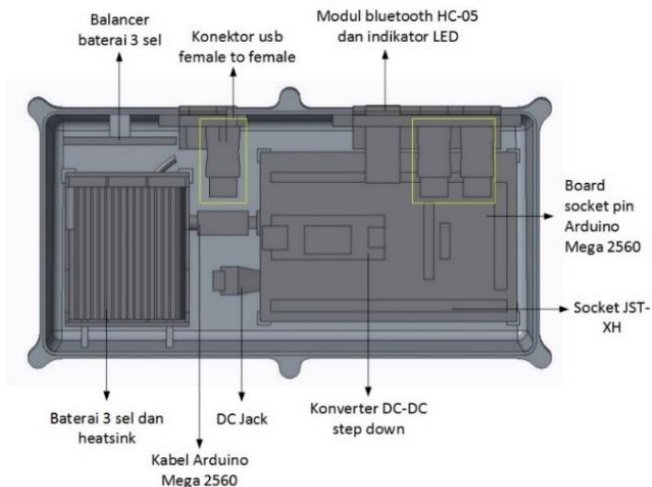
e. *Socket* DC Jack

Jenis perangkat yang digunakan sebagai penghubung antara *converter* dc-dc dengan *socket* suplai daya untuk Arduino Mega 2560.

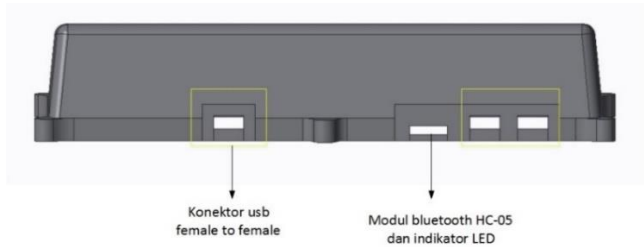
f. Kabel Arduino Mega 2560 *Male To Print*

3.1.2 *Layout* Peletakkan Komponen

Peletakkan komponen penyusun *remote wireless* dibagi menjadi dua bagian besar yaitu bagian bawah dan depan *remote*. Bagian bawah berisi komponen utama seperti Arduino Mega 2560 beserta *board socket*-nya seperti yang terlihat pada Gambar 3.11, dikarenakan dimensi komponen tersebut yang besar dan tebal sekitar 3 cm maka diletakkan dibagian bawah *remote*.

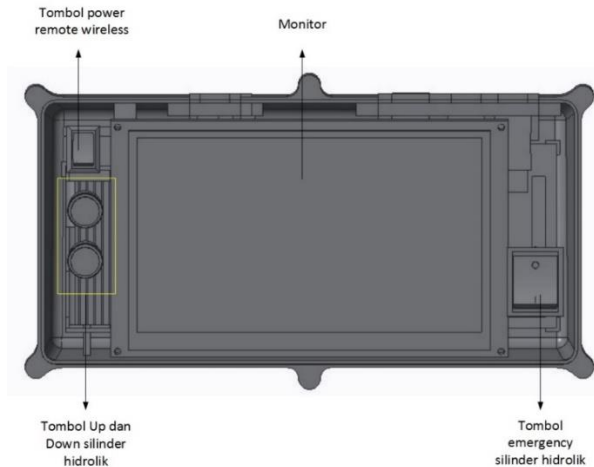


Gambar 3.11 *Layout* komponen bagian bawah



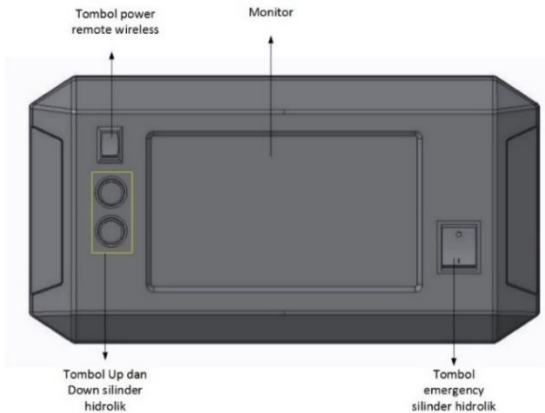
Gambar 3.12 Layout komponen bagian atas

Bagian atas *remote* (Gambar 3.12) digunakan sebagai letak penempatan *socket-socket* seperti *charger* untuk mengisi baterai agar tidak perlu membuka penutup *remote*, *usb* untuk transfer program Arduino digunakan jika ada perubahan dalam program, *usb* untuk mengubah desain tampilan monitor, dan tempat indikator *bluetooth* berupa led yang diletakkan disamping *bluetooth*.



Gambar 3.13 Layout komponen bagian depan (1)

Bagian depan *remote* (Gambar 3.13) digunakan untuk meletakkan komponen *interface* seperti monitor dan beberapa tombol. Penutup bagian atas *remote* dapat dilihat pada Gambar 3.14.



Gambar 3.14 *Layout* komponen bagian depan (2)

3.1.3 Desain Casing *Remote*

a. Sketsa gambar

Pembuatan desain *remote wireless* dilakukan oleh pihak desain produk perusahaan, untuk desain awal yang diberikan berupa sketsa gambar menggunakan pensil. Hasil ini kemudian diajukan kepada pihak kepala divisi untuk mendapat saran dan persetujuan. Hasil sketsa desain *remote wireless* dapat dilihat pada Gambar 3.15.

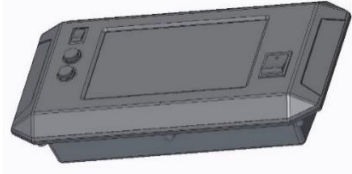


Gambar 3.15 Sketsa desain *remote*

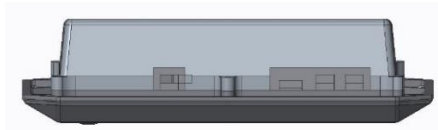
b. Gambar 3D

Proses desain bentuk *remote* selanjutnya adalah tahap desain secara 3D. Setelah mendapat persetujuan mengenai desain sketsa

gambar pensil, selanjutnya mendesain bentuk 3D menggunakan bantuan komputer. Pada tahap ini juga dilakukan pemilihan penempatan komponen penyusun *remote*. Hasil gambar desain 3D *remote wireless* dapat dilihat pada Gambar 3.16 hingga 3.19.



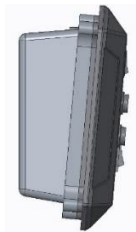
Gambar 3.16 Desain 3D *remote*



Gambar 3.17 Desain 3D *remote* tampak atas



Gambar 3.18 Desain 3D *remote* tampak bawah



Gambar 3.19 Desain 3D *remote* tampak kiri

c. Gambar produk

Tahap selanjutnya adalah mendesain gambar produk dari *remote wireless*. Namun sebelum melakukan tahap ini, perlu fiksasi

penempatan komponen penyusun *remote* terlebih dahulu. Gambar ini nantinya akan ditunjukkan pada konsumen sebagai acuan dalam memperkenalkan produk *remote wireless*. Tampilan atas dapat dilihat pada Gambar 3.20, yang menjadi ikon produk pada gambar tersebut adalah gambar Ganila 2.0.



Gambar 3.20 Gambar desain *remote* tampak atas

Pada bagian sisi depan *remote wireless* terdapat beberapa *socket female* (dapat dilihat pada Gambar 3.21) yang masing-masing memiliki fungsi tersendiri yaitu, *socket charger*, transfer program Arduino, transfer desain monitor, dan led untuk indikator *bluetooth*.



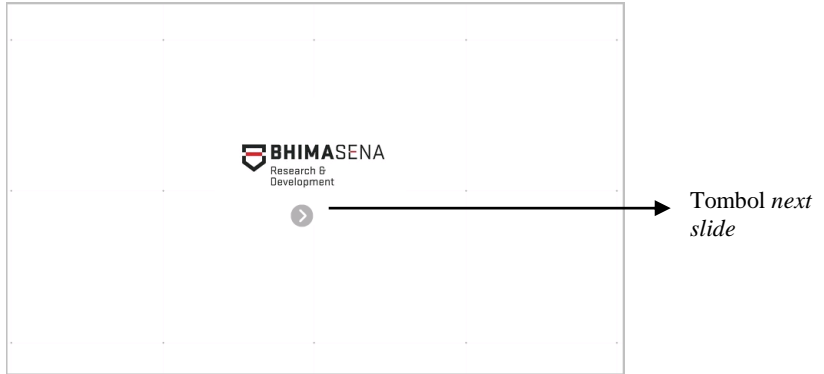
Gambar 3.21 Gambar desain *remote* depan

3.2 Pembuatan Desain Tampilan Monitor

Monitor yang digunakan pada *remote wireless* adalah produk dari Nextion yang memiliki lebar 7 inch. Jenis monitor ini adalah *touchscreen*. Beberapa hal yang akan ditampilkan pada monitor tersebut adalah pengenalan produk Ganila 2.0 dan data ketinggian silinder hidrolik.

Data yang ditampilkan pada monitor *remote wireless* dibagi menjadi dua bagian yaitu, data pengenalan kendaraan Ganila 2.0 dan ketinggian silinder hidrolik. Data pengenalan kendaraan Ganila 2.0

diperoleh dari perusahaan, lalu dikumpulkan dan diseleksi sebagian yang penting. Hasil data yang diseleksi, diubah menjadi tampilan gambar.



Gambar 3.22 Tampilan monitor *slide 1*



Gambar 3.23 Tampilan monitor *slide 2*

Pada Gambar 3.23 terdapat menu utama yaitu *vehicle* dan *hydraulic*. Pada Gambar 3.23 juga menampilkan animasi kendaraan Ganila 2.0 berputar 360 derajat. Menu *vehicle* yang dapat dilihat pada Gambar 3.24 berisi tiga sub menu yaitu, *feature*, *specification*, dan *layout*.

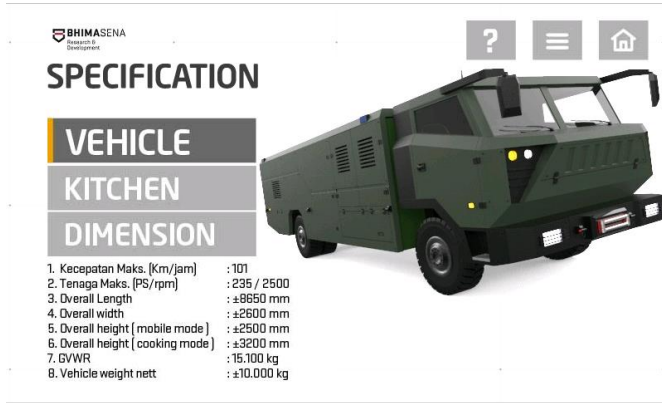


Gambar 3.24 Tampilan monitor *slide 3*



Gambar 3.25 Tampilan monitor *slide 4*

Sub menu *feature* yaitu Gambar 3.25 berisi fungsi dan kemampuan kendaraan. Sub menu *feature* menerangkan bahwa kendaraan Ganila 2.0 memiliki kapasitas masak 200-300 porsi dalam 3 jam diluar preparasi. Informasi ini membantu konsumen untuk memahami kemampuan dari Ganila 2.0 untuk kegiatan memasak, sehingga mereka bisa memperkirakan kegiatan memasak yang akan dilakukan di kendaraan tersebut.

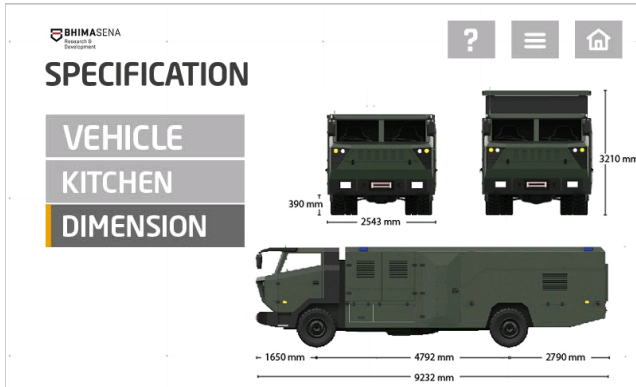


Gambar 3.26 Tampilan monitor *slide 5*

Pada Gambar 3.26 menerangkan bahwa spesifikasi kendaraan dibagi menjadi tiga bagian yaitu, *vehicle*, *kitchen*, dan *dimension*. *Vehicle* pada sub menu *specification* berisi kemampuan badan kendaraan itu sendiri seperti kecepatan, berat kendaraan, dan lainnya. Sub menu *specification* berupa *kitchen* pada Gambar 3.27 berisi mengenai kemampuan kendaraan yang digunakan sebagai kegiatan memasak. *Dimension* berisi ukuran kendaraan Ganila 2.0.

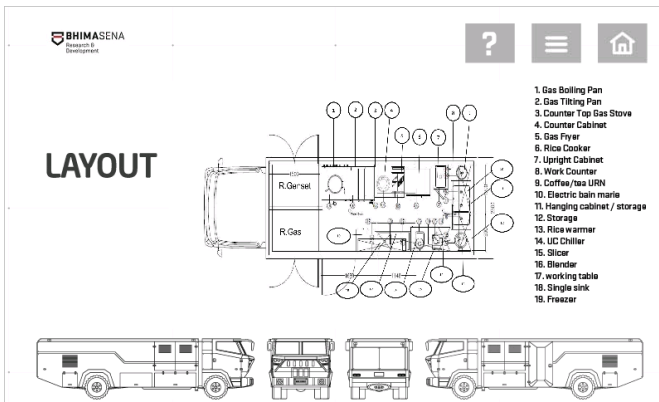


Gambar 3.27 Tampilan monitor *slide 6*



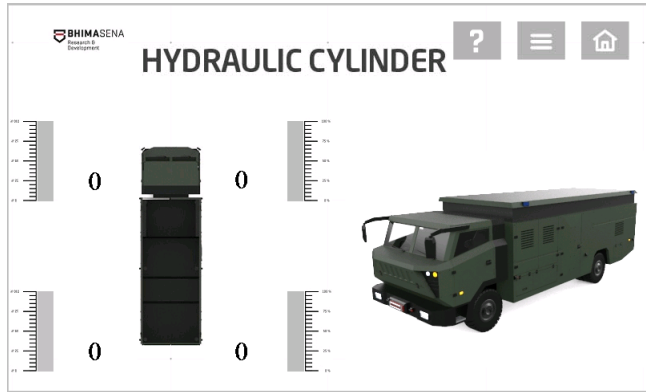
Gambar 3.28 Tampilan monitor *slide 7*

Gambar 3.28 menunjukkan keterangan dimensi kendaraan Ganila 2.0 yang mana lebih minimalis dibandingkan dengan Ganila 1.0. Pada keterangan Gambar 3.28 dapat dilihat bahwa ketinggian Ganila ketika atap dalam kondisi naik sebesar 3210 mm.



Gambar 3.29 Tampilan monitor *slide 8*

Gambar 3.29 berisi penempatan peralatan memasak di dalam kabin kendaraan Ganila 2.0. Informasi pada Gambar 3.29 mempermudah operator untuk mengetahui letak peralatan memasak dengan cepat.



Gambar 3.30 Tampilan monitor *slide 9*

Menu *hydraulic* Gambar 3.30 menampilkan data ketinggian silinder. Data tersebut ditampilkan dalam bentuk angka dan slider. Selain itu untuk memperjelas pemahaman kenaikan silinder hidrolik, maka ditampilkan juga animasi kendaraan Ganila 2.0 ketika atap kabin belakang naik.



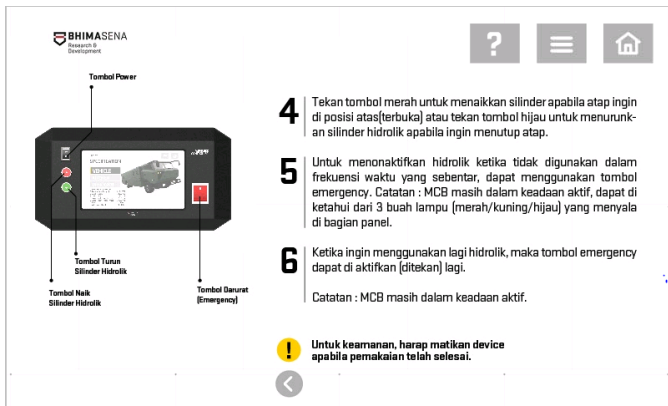
Gambar 3.31 Tampilan monitor *slide 10*

Selain menu *vehicle* dan *hydraulic*, terdapat menu tambahan berupa *operational procedure* pada Gambar 3.31 yang diletakkan pada simbol menu tanda tanya. Menu ini berisi langkah-langkah

mengaktifkan sistem hidrolik pada Ganila 2.0 dan fungsi bagian-bagian *remote wireless*.



Gambar 3.32 Tampilan monitor *slide 11*

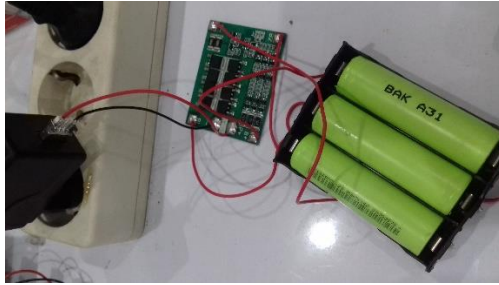


Gambar 3.33 Tampilan monitor *slide 12*

3.3 Perakitan Komponen Penyusun *Remote (Assembly)*

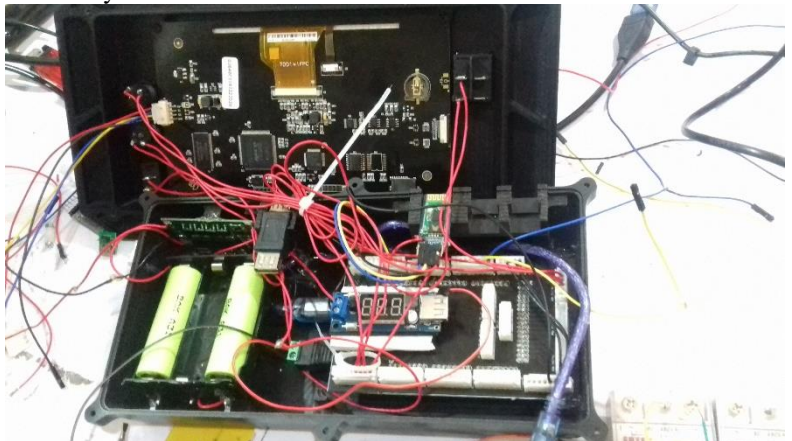
Beberapa langkah yang dilakukan ketika perakitan komponen penyusun *remote* adalah :

1. Memastikan *balancer* baterai dapat digunakan. *Balancer* ini digunakan sebagai penyeimbang pengisian baterai ketika *charging*.



Gambar 3.34 Charging baterai 3 sel menggunakan balancer

2. Merakit komponen penyusun *remote* ke dalam *casing remote* yang telah di cetak, dan mengatur *wiring* kabel komponen di dalamnya.



Gambar 3.35 Pemasangan komponen dan *wiring* kabel

3.4 Konfigurasi sistem

Pengerjaan fungsi *remote wireless* dibagi menjadi dua yaitu, koneksi tombol *remote wireless* dengan panel hidrolik dan sinkronisasi data ketinggian silinder hidrolik dengan hasil pengiriman data dari panel hidrolik yang ditampilkan pada monitor.

Pada tahapan ini membuat rangkaian tiga tombol dengan Arduino Mega 2560. Setiap tombol memiliki kegunaan masing-masing yaitu, tombol 1 (berwarna merah) untuk menaikkan silinder hidrolik, tombol

2 (berwarna hijau) untuk menurunkan silinder hidrolik, dan tombol kotak 3 (kotak berwarna merah) sebagai *emergency*. Rangkaian ini masih dibuat menggunakan *project board* dengan sumber daya dari *power bank*. Pada tahapan ini juga dilakukan konfigurasi *bluetooth remote wireless* dan panel hidrolik, serta dilakukan pengujian singkat. Pengujian singkat meliputi fungsi ketiga tombol dapat bekerja secara optimal.

Konfigurasi *bluetooth* dilakukan dengan cara menghubungkan masing-masing modul *bluetooth* HC-05 dengan Arduino Mega 2560 yang berbeda, selanjutnya memprogram masing-masing *bluetooth* dan menentukan Master-Slave. Penentuan ini dilakukan untuk menentukan *bluetooth* mana yang melakukan komunikasi terlebih dahulu. Singkat program dapat dilihat pada Gambar 3.36, untuk program selengkapnya dapat dilihat di Lampiran A-3.

```

CONFIG §
// by Brandon Plumbo AKA Riftliger
// original version from Martyn Currey http://www.martyncurrey.com/
// HC-05 universal test sketch to enter AT-commands
// Uses hardware serial to talk to the host computer and software serial for communication with the bluetooth module
// Pins
// BT VCC to Arduino 5V out.
// BT GND to GND
// Arduino D11 to BT RX
// Arduino D10 BT TX
//
// When a command is entered in the serial monitor on the computer
// the Arduino will relay it to the bluetooth module and display the result.
//
// The HC-05 modules require both CR and NL

#include <SoftwareSerial.h>
SoftwareSerial BTSerial(10, 11); // TX,RX

char c = ' ';
boolean NL = true;

void setup()
{

```

Gambar 3.36 Listing program konfigurasi *Bluetooth*

Setelah melakukan konfigurasi *Bluetooth*, selanjutnya memprogram masing-masing Arduino untuk mengaktifkan tiga jenis tombol, yang mana setiap tombol memiliki fungsi tersendiri pada sistem hidrolik. Untuk program *master* yang dilakukan oleh *bluetooth remote* dapat dilihat di Lampiran A-1, singkatnya dapat dilihat pada Gambar 3.37. Sedangkan program *slave* dilakukan di bagian panel hidrolik, untuk program selengkapnya dapat dilihat di Lampiran A-2, program singkatnya pada Gambar 3.38.

```
master_menik $
#include <doxygen.h>
#include <NexButton.h>
#include <NexCheckbox.h>
#include <NexConfig.h>
#include <NexCrop.h>
#include <NexDualstateButton.h>
#include <NexGauge.h>
#include <NexGpio.h>
#include <NexHardware.h>
#include <NexHotspot.h>
#include <NexNumber.h>
#include <NexObject.h>
#include <NexPage.h>
#include <NexPicture.h>
#include <NexProgressBar.h>
#include <NexRadio.h>
#include <NexRtc.h>
#include <NexScrolltext.h>
```

Gambar 3.37 Listing program master bluetooth

```
slave_menik
#include <SoftwareSerial.h>
SoftwareSerial BTserial(10, 11);

// Set DEBUG to true to output debug information to the serial monitor
boolean DEBUG = true;

// Variables used for incoming data
const byte maxDataLength = 20;

char receivedChars[21] ;

boolean newData = false;

int jarak1, jarak2, jarak3, jarak4;
double y1 = 0, x1 = 0, y1_1= 0, y2 = 0, x2 = 0, y2_1= 0;
double y3 = 0, x3 = 0, y3_1 = 0, y4 = 0, x4 = 0, y4_1= 0;
const double a = 0.1;

void setup()
{
  if (DEBUG)
  {
```

Gambar 3.38 Listing proram slave bluetooth

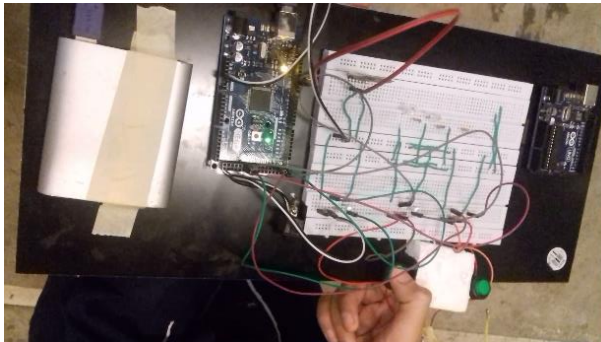
Halaman ini sengaja dikosongkan

BAB IV

PENGUJIAN DAN ANALISIS *REMOTE WIRELESS*

4.1 Cara Pengujian *Remote Wireless*

Sebelum melakukan pengujian *remote wireless*, terlebih dahulu melakukan pengujian fungsi program dasar untuk mengaktifkan 3 tombol yang nantinya digunakan pada *remote* tersebut. Pengujian ini meliputi fungsi tombol dapat bekerja secara benar, namun dalam hal ini belum menggunakan parameter tertentu.



Gambar 4.1 Pengujian kerja *remote* menggunakan *project board*

Seperti pada Gambar 4.1, pengujian fungsi tombol masih menggunakan *remote* yang belum terintegrasi secara *hardware*. Pengujian masih menggunakan papan ACP sebagai alas komponen penyusun. Komponen utama yang digunakan yaitu Arduino Mega 2560, 3 tombol, *project board*, kabel *jumper*, dan *power bank* sebagai penyuplai tegangan untuk Arduino Mega 2560.

Pengujian dilakukan dengan berada disamping kendaraan Ganila 2.0 seperti pada Gambar 4.2. Namun sebelum pengujian harus dipastikan terlebih dahulu tidak ada orang di dalam kendaraan tersebut, karena ditakutkan adanya masalah hidrolis. Pengujian ini hanya digunakan untuk memastikan ketiga fungsi tombol dapat bekerja dengan baik dan belum menggunakan parameter tertentu.



Gambar 4.2 Pengujian kerja *remote* dengan kendaraan Ganila 2.0

Proses pengujian selanjutnya menggunakan *remote wireless* yang telah terintegrasi atau sudah berbentuk *hardware*. Pengujian kali ini menggunakan parameter jarak dan waktu yang dibutuhkan untuk mengoperasikan silinder hidrolik untuk naik. Pengujian dilakukan di bengkel perusahaan didampingi oleh *operator* hidrolik.

Sebelum melakukan pengujian, dipastikan terlebih dahulu tidak ada pegawai bengkel yang berada di dalam kendaraan. Selain itu ketika pengujian, atap kendaraan juga dilepas karena masih mengalami perbaikan. Sehingga untuk mengatasi hal tersebut, keempat bagian silinder ditutup oleh plat, dan sensor mengukur ketinggian plat tersebut ketika silinder dinaikkan.

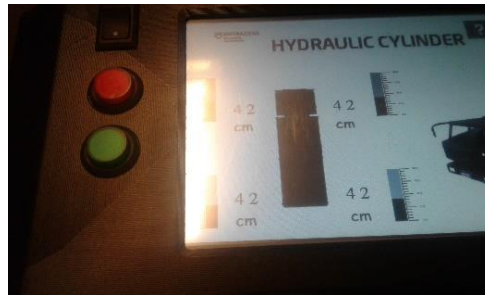
Selanjutnya memasang Arduino beserta komponen elektrik tambahan lainnya ke dalam panel hidrolik. Komponen elektrik tersebut terdiri dari *solid state relay* sebanyak tiga buah, board Arduino Mega 2560, modul *Bluetooth HC-05*, *power supply*, *buck converter*, dan *wiring* kabel. Selanjutnya mengaktifkan *remote wireless* dengan menekan tombol *power* berwarna hitam pada *remote*. Untuk memastikan *remote wireless* telah terhubung dengan panel, dapat dilihat indikator led berwarna merah yang terpasang pada *remote*. Jika led pada *remote* berkedip selama dua detik sekali, maka menunjukkan *remote* telah terhubung.

Hasil pengujian didokumentasikan pada Gambar 4.3. Operator mengoperasikan hidrolik dengan berada didekat kendaraan. Selain itu, ketika proses pengoperasian berlangsung juga dilakukan monitoring

ketinggian silinder hidrolik yang ditampilkan pada monitor *remote wireless* seperti terlihat pada Gambar 4.4.



Gambar 4.3 Pengujian *remote* dengan jarak 2 meter



Gambar 4.4 Tampilan hasil pengukuran sensor

Pada tampilan monitor terdapat gambar berupa animasi kendaraan Ganila 2.0, untuk menunjukkan besar ketinggian silinder hidrolik ditampilkan melalui *counting* angka dan slider.

4.2 Cara Pengujian *Remote* Kabel

Pengujian *remote* kabel dilakukan dengan berada didekat kendaraan Ganila 2.0. Sebelum pengujian, terlebih dahulu dipastikan tidak ada orang di dalam kendaraan Ganila 2.0. Langkah selanjutnya menyalurkan sumber listrik ke panel hidrolik dan mengaktifkan MCB 3 fasa serta 1 fasa. Selanjutnya operator mengambil *remote* kabel untuk mengaktifkan silinder hidrolik agar naik. Cara pengujian *remote* kabel hampir sama dengan *remote wireless* yang berbeda hanya jarak

maksimal untuk mengoperasikan *remote* tersebut, dan kondisi Ganila 2.0 yang masih menggunakan atap seperti terlihat pada Gambar 4.5.



Gambar 4.5 Pengujian *remote* kabel dengan jarak 1 meter dan 3 meter

4.3 Analisa Hasil Pengujian *Remote Wireless* dengan *Remote Kabel*

Parameter pengujian *remote wireless* dan *remote* kabel yaitu jarak *remote* dengan kendaraan Ganila 2.0 dan waktu yang dibutuhkan untuk mengoperasikan sistem hidrolik pada kendaraan tersebut. Hasil pengujian *remote wireless* dirangkum pada Tabel 4.1.

Tabel 4.1 Hasil pengujian *remote wireless*

No.	Jarak <i>remote wireless</i> dengan kendaraan Ganila 2.0	Waktu pengoperasian silinder hidrolik ketika naik	Keterangan
1	2 meter	45 detik	Terhubung
2	5 meter	43 detik	Terhubung
3	10 meter	44 detik	Terhubung
4	15 meter	46 detik	Terhubung

Pada pengujian pertama, *remote wireless* mengalami masalah pada bagian tombol *up* yang digunakan untuk menaikkan silinder hidrolik dimana tombol tidak berfungsi, sehingga dilakukan pengecekan ulang pada *wiring* panel hidrolik dan didapatkan kesalahan dalam pemasangan relay. Untuk pengujian selanjutnya tombol *up* dan *down* dapat berfungsi dengan benar, namun tombol *emergency* tidak berfungsi. Selanjutnya dilakukan pengecekan lagi pada panel hidrolik, dan didapatkan hasil bahwa relay *emergency* tidak terhubung pada rangkaian pemutus sistem hidrolik. Pengujian selanjutnya mencoba ketiga tombol *remote wireless*, dan hasilnya dapat berjalan dengan benar. Namun tombol *emergency* belum bisa memutus sistem hidrolik karena relay tidak dihubungkan pada rangkaian pemutus hidrolik.

Setelah melakukan pengujian fungsi tombol, selanjutnya menguji *remote* dengan jarak tertentu dan menghitung waktu yang dibutuhkan untuk menaikkan silinder hidrolik. Pengujian pertama, *remote wireless* berada pada jarak 2 meter dari kendaraan, dan didapatkan hasil bahwa semua tombol masih berfungsi dengan baik serta waktu yang dibutuhkan untuk mengoperasikan hidrolik adalah 45 detik dapat dilihat pada Tabel 4.1. Pengujian kedua, *remote wireless* berada pada jarak 5 meter dari kendaraan, dan didapatkan hasil bahwa semua tombol masih berfungsi serta waktu yang dibutuhkan sebanyak 43 detik. Pengujian ketiga, *remote wireless* berada pada jarak 10 meter dari kendaraan, dan didapatkan hasil bahwa *remote* masih berfungsi serta waktu pengoperasian sebanyak 44 detik. Pengujian keempat, *remote* berada pada jarak 15 meter, dan didapatkan hasil bahwa *remote* masih berfungsi serta waktu pengoperasian sebanyak 46 detik. Setelah pengujian keempat, data dikumpulkan dan dianalisa.

Berdasarkan hasil pengujian yang ditunjukkan pada Tabel 4.1, *remote wireless* dapat berfungsi dengan baik dalam pengoperasian hidrolik. Penggunaan *remote* dapat mempermudah *operator* dalam mengoperasikan hidrolik dikarenakan bentuknya yang *portable* sehingga dapat dibawa dan diletakkan dimana saja. Selain itu, dapat juga mengoperasikan hidrolik dari sudut manapun seperti dari bagian depan kendaraan, samping, ataupun belakang.

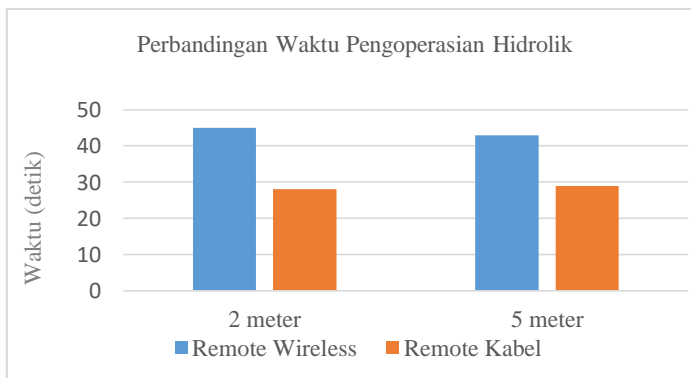
Selanjutnya membandingkan performa antara *remote wireless* dengan *remote* kabel. Hasil pengujian *remote* kabel terangkum pada Tabel 4.2. Pengujian *remote* kabel hanya menggunakan *range* jarak pengujian sejauh 1 hingga 5 meter dikarenakan panjang kabel hanya

sebesar 5 meter. Sedangkan dikarenakan *remote wireless* tidak menggunakan kabel maka *range* jarak pengujian dapat dilakukan antara 1-15 meter. Penentuan batas 15 meter ini karena terkendala dimensi ruangan bengkel kendaraan.

Tabel 4.2 Hasil pengujian *remote* kabel

	Jarak <i>remote kabel</i> dengan kendaraan Ganila 2.0	Waktu pengoperasian silinder hidrolik ketika naik	Keterangan
1	1 meter	27,5 detik	Terhubung
2	2 meter	28,1 detik	Terhubung
3	3 meter	28,5 detik	Terhubung
4	4 meter	28,8 detik	Terhubung
5	5 meter	28,9 detik	Terhubung

Pada Tabel 4.2, selisih waktu yang dibutuhkan untuk mengoperasikan hidrolik antara jarak 1 hingga 5 meter tidak jauh berbeda sekitar 1,4 detik dibandingkan dengan selisih waktu pengoperasian hidrolik menggunakan *remote wireless* dengan jarak antara 2 hingga 15 meter sekitar 3 detik.



Gambar 4.6 Grafik perbandingan waktu pengoperasian hidrolik antara *remote wireless* dengan kabel

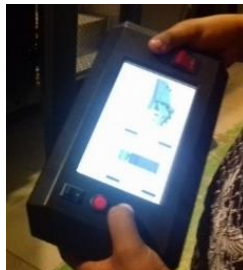
Gambar 4.6 menunjukkan bahwa waktu yang dibutuhkan *remote wireless* untuk mengoperasikan hidrolik jauh lebih lama dibandingkan dengan *remote* kabel. Berdasarkan data pada Gambar 4.6, dapat diketahui presentase selisih durasi pengoperasian hidrolik antara *remote wireless* dengan *remote* kabel pada jarak 5 meter.

$$\begin{aligned} & \% \text{ presentase} \\ &= \frac{\text{Durasi } \textit{remote wireless} - \text{Durasi } \textit{remote} \text{ kabel}}{(\text{Durasi pengoperasian } \textit{remote} \text{ kabel})} \times 100\% \\ &= \frac{43 \text{ s} - 28,9 \text{ s}}{(28,9 \text{ s})} \times 100\% \\ &= 48,79\% \end{aligned}$$

Berdasarkan presentase selisih durasi pengoperasian hidrolik antara *remote wireless* dengan *remote* kabel sebesar 48,79% menunjukkan bahwa *remote wireless* lebih lambat dalam mengoperasikan hidrolik dibandingkan *remote* kabel. Namun dari segi fungsional, *remote wireless* sudah mampu mengoperasikan hidrolik secara jarak jauh.



Gambar 4.7 Remote kabel hidrolik



Gambar 4.8 Remote wireless hidrolik

Dari segi dimensi, dapat dilihat pada Gambar 4.7 *remote* kabel lebih minimalis namun hanya memiliki 3 fungsi, sedangkan pada Gambar 4.8 dapat dilihat bahwa *remote wireless* memiliki fungsi yang kompleks, serta bentuk terlihat lebih modern karena tanpa menggunakan kabel.

BAB V PENUTUP

5.1 Kesimpulan

Berdasarkan pengujian mengenai fungsi dari *remote wireless* yang dibuat, maka dapat disimpulkan bahwa *remote wireless* dapat mengoperasikan hidrolik pada jarak mencapai 15 meter dari kendaraan. Selain itu, beberapa tombol pada *remote* dapat berfungsi dengan benar yaitu menaikkan dan menurunkan atap silinder hidrolik dan tombol *emergency* dapat berfungsi untuk memutus rangkaian hidrolik sementara. Selain itu, fitur lainnya berupa monitor pada *remote* juga mampu menampilkan data ketinggian hidrolik.

Performa *remote wireless* dari segi fungsional sudah mampu dalam mengoperasikan hidrolik secara jarak jauh dibandingkan *remote* kabel. Namun presentase durasi pengoperasian hidrolik menggunakan *remote wireless* yaitu 48,79% lebih lama dibandingkan *remote* kabel.

5.2 Saran

Harapan dalam pembuatan Proyek Akhir ini yaitu masyarakat dapat mengembangkan dan menjalankan *remote* ini dengan benar, serta diharapkan adanya pengembangan lagi terhadap *remote* tersebut, seperti menambahkan data-data kesehatan hidrolik seperti kondisi motor hidrolik, ketersediaan oli, tekanan hidrolik, dan suhu oli hidrolik. Data-data ini nantinya dapat diperoleh dengan menambahkan beberapa sensor.

Halaman ini sengaja dikosongkan

DAFTAR PUSTAKA

- [1] PT Merpati Wahana Raya, “Ganilla (Kitchen Truck) – MWR,” *PT Merpati Wahana Raya*. [Online]. Available: <http://merpatiwahanaraya.com/portfolio/ganilla/>. [Accessed: 13-Jun-2019].
- [2] PT Merpati Wahana Raya, “Ganilla 2.0 (Kitchen Truck) – MWR,” *PT Merpati Wahana Raya*. [Online]. Available: <http://merpatiwahanaraya.com/portfolio/ganilla-2/>. [Accessed: 13-Jun-2019].
- [3] A. Jain, “Insight - How MCB works,” *EngineersGarage*, 2012. [Online]. Available: <https://www.engineersgarage.com/insight/how-mcb-works>.
- [4] Schneider-electric, *Daftar Harga 2017*. Schneider-electric, 2017.
- [5] Edvard Csanyi, “MCB - Miniature Circuit Breaker Construction,” 2012. [Online]. Available: <https://electrical-engineering-portal.com/mcb-miniature-circuit-breaker-construction>. [Accessed: 04-Jul-2019].
- [6] Electrical Classroom, “Miniature Circuit Breaker (MCB) – Principle of operation,” *Electrical Classroom*, 2018. [Online]. Available: <https://www.electricalclassroom.com/miniature-circuit-breaker-mcb-principle-of-operation/>. [Accessed: 31-Mar-2019].
- [7] Springercontrols, “MAGNETIC MOTOR STARTERS: THE BASICS,” *Springer Controls Company*, 2017. [Online]. Available: <https://www.springercontrols.com/news/magnetic-motor-starters-basics/>. [Accessed: 31-Mar-2019].
- [8] Omnical Company, “LRD08 Schneider Electric Thermal overload relay - Omnical,” 2019. [Online]. Available: <https://www.omnical.co/products/schneider-electric-telemecanique/lrd08/276910904>. [Accessed: 04-Jul-2019].
- [9] D. Wade, “Control Relay Tutorial - How a Relay Works,” *Wade Instruments and Service, Inc.*, 2010. [Online]. Available: http://www.wadeinstruments.com/relays/control_relay_tutorial.htm. [Accessed: 31-Mar-2019].
- [10] Amazon.com, “Baomain Power Relay MK2P-I AC 220V Coil DPDT 8 Pin with Plug-in Terminal Socket: Amazon.com: Industrial & Scientific,” 2019. [Online]. Available:

- <https://www.amazon.com/Baomain-Power-MK2P-I-Terminal-Socket/dp/B01N3KO7SY>. [Accessed: 04-Jul-2019].
- [11] A. Goldsmith, (*Wireless Communication by A.Goldsmith*).pdf. Berkeley, California: Cambridge University Press, New York, 2005.
- [12] Y. Li, Q. Xue, J. He, and T. Zhao, “Design of Music Toy Car Based on Smart Phone Via Bluetooth Remote Control,” *Proceedings of 2018 2nd IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference, IMCEC 2018*. pp. 1976–1980, 2018.
- [13] A. Nayyar, “Real Time Smart Home Automation based on PIC Microcontroller, Bluetooth and Android Technology,” *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*. pp. 1478–1484, 2016.
- [14] RAM Electronics, “Bluetooth Module HC-05 , Serial TTL "Most Popular" - RAM Electronics,” *ram-e-shop*, 2019. [Online]. Available: <https://ram-e-shop.com/product/kit-bluetooth-hc05/>. [Accessed: 04-Jul-2019].
- [15] M. Kusriyanto and B. D. Putra, “Smart Home Using Local Area Network (Lan) Based Arduino Mega 2560,” *Proceedings - ICWT 2016: 2nd International Conference on Wireless and Telematics 2016*. pp. 127–131, 2017.
- [16] Arduino, “Language Reference,” 2019. [Online]. Available: <https://www.arduino.cc/reference/en/#page-title>. [Accessed: 28-Apr-2019].
- [17] ARDUINO, “Arduino Mega 2560 Rev3,” *Arduino*, 2019. [Online]. Available: <https://store.arduino.cc/usa/mega-2560-r3>. [Accessed: 04-Jul-2019].
- [18] Arduino, *Arduino Mega 2560*, vol. الح ا العدد. <http://www.mantech.co.za>, 2001.
- [19] Arduino, “Void,” 2019. [Online]. Available: <https://www.arduino.cc/reference/en/language/variables/data-types/void/>. [Accessed: 28-Apr-2019].
- [20] Arduino, “Char,” 2019. [Online]. Available: <https://www.arduino.cc/reference/en/language/variables/data-types/char/>. [Accessed: 28-Apr-2019].
- [21] Arduino, “If,” 2019. [Online]. Available: <https://www.arduino.cc/reference/en/language/structure/control-structure/if/>. [Accessed: 28-Apr-2019].

- [22] Arduino, “#define,” 2019. [Online]. Available: <https://www.arduino.cc/reference/en/language/structure/further-syntax/define/>. [Accessed: 28-Apr-2019].
- [23] Arduino, “setup(),” 2019. [Online]. Available: <https://www.arduino.cc/reference/en/language/structure/sketch/setup/>. [Accessed: 28-Apr-2019].
- [24] Arduino, “Return,” 2019. [Online]. Available: <https://www.arduino.cc/reference/en/language/structure/control-structure/return/>. [Accessed: 28-Apr-2019].
- [25] Arduino, “SoftwareSerial Library,” 2019. [Online]. Available: <https://www.arduino.cc/en/Reference/SoftwareSerial>. [Accessed: 28-Apr-2019].
- [26] B. J. Jansen, “The graphical user interface,” *ACM SIGCHI Bull.*, vol. 30, no. 2, pp. 22–26, 1998.
- [27] ITEAD STUDIO, “NX8048K070_011 - Nextion,” 2011. [Online]. Available: nextion.itead.cc/resources/datasheets/nx8048k070_011/. [Accessed: 28-Apr-2019].
- [28] ITEAD STUDIO, “The Nextion Editor Guide,” 2011. [Online]. Available: https://nextion.itead.cc/editor_guide/. [Accessed: 28-Apr-2019].
- [29] ITEAD STUDIO, “Home - Nextion,” 2011. [Online]. Available: <https://nextion.itead.cc/>. [Accessed: 01-Jul-2019].

Halaman ini sengaja dikosongkan

LAMPIRAN

A-1 Program *Master Bluetooth*

```
master_menik $
#include <doxygen.h>
#include <NexButton.h>
#include <NexCheckbox.h>
#include <NexConfig.h>
#include <NexCrop.h>
#include <NexDualstateButton.h>
#include <NexGauge.h>
#include <NexGpio.h>
#include <NexHardware.h>
#include <NexHotspot.h>
#include <NexNumber.h>
#include <NexObject.h>
#include <NexPage.h>
#include <NexPicture.h>
#include <NexProgressBar.h>
#include <NexRadio.h>
#include <NexRtc.h>
#include <NexScrolltext.h>
#include <NexSlider.h>
#include <NexText.h>
#include <NexTimer.h>
#include <Nexion.h>
#include <NexTouch.h>
#include <NexUpload.h>
#include <NexVariable.h>
#include <NexWaveform.h>
#include <Nexion.h>
//#include "Nexion.h"

#include <SoftwareSerial.h>
SoftwareSerial BTserial(10, 11);

int buttonPin1 = 23; //EMG
int buttonPin2 = 25; //UP
int buttonPin3 = 27; //DO

int ledPin1 = 41; //the pin your button is connected to
int ledPin2 = 43;
int ledPin3 = 45;

int buttonState1 = 0;
int buttonState2 = 0;
int buttonState3 = 0;

// Set DEBUG to true to output debug information to the serial monitor
boolean DEBUG = true;
```

```

// Variables used for incoming data
const byte maxDataLength = 100;
char receivedChars[101] ;
boolean newData = false;

// Variables used for the timer
unsigned long startTime = 0;
unsigned long waitTime = 500;

unsigned long startTime2 = 0;
unsigned long waitTime2 = 500;

void setup()
{
  pinMode(buttonPin1, INPUT);
  pinMode(buttonPin2, INPUT);
  pinMode(buttonPin3, INPUT);

  NexProgressBar j0 = NexProgressBar(0, 5, "j0");
  NexNumber n0= NexNumber(0, 11, "n0");
  NexProgressBar j1 = NexProgressBar(0, 6, "j0");
  NexNumber n1= NexNumber(0, 12, "n1");
  NexProgressBar j2 = NexProgressBar(0, 7, "j2");
  NexNumber n2= NexNumber(0, 13, "n1");
  NexProgressBar j3 = NexProgressBar(0, 8, "j3");
  NexNumber n3= NexNumber(0, 14, "n2");
  uint32_t val;
  nexInit();
  if (DEBUG)
  {
    // open serial communication for debugging
    Serial.begin(9600);
    Serial.println(__FILE__);
    Serial.println(" ");
  }
  BTserial.begin(38400);
  if (DEBUG) { Serial.println("SoftSerial started at 9600"); }

  newData = false;
  startTime = millis();
}

void loop()
{
  if ( millis()-startTime > waitTime )
  {

```

```

    BTserial.print("<sendjarak>");
    if (DEBUG) { Serial.println("Request sent"); }
    startTime = millis();
}
if ( millis()-startTime2 > waitTime2 ) {
{
    BTserial.print("<sendtombol,");
    buttonState1 = digitalRead(buttonPin1);
    if(buttonState1 == HIGH){ //tombol 1
    BTserial.write("a");
    }else{
    BTserial.write("b");
    }
    BTserial.print(",");

    buttonState2 = digitalRead(buttonPin2);
    if (buttonState2 == HIGH) { //tombol 2
    BTserial.write('c'); }
    else {
    BTserial.write('d');
    }
    BTserial.print(",");

    buttonState3 = digitalRead(buttonPin3);
    if (buttonState3 == HIGH) //tombol 3
    {
        BTserial.write('e');
    }
    else {
        BTserial.write('f');
    }
    BTserial.print(">");
    if (DEBUG) { Serial.println("Request sent1"); }
    startTime2 = millis();
}
}

recvWithStartEndMarkers();
if (newData)
{
    if (DEBUG) { Serial.println("Data received"); }

    Serial.print("n0.val=");
    Serial.print(jarak1);
    Serial.write(0xff);
    Serial.write(0xff);
    Serial.write(0xff);
}

```

```

        Serial.print("j0.val=");
        Serial.print(jarak1);
        Serial.write(0xff);
        Serial.write(0xff);
        Serial.write(0xff);

        Serial.print("n1.val=");
        Serial.print(jarak2);
        Serial.write(0xff);
        Serial.write(0xff);
        Serial.write(0xff);
        Serial.print("j1.val=");
        Serial.print(jarak2);
        Serial.write(0xff);
        Serial.write(0xff);
        Serial.write(0xff);

        Serial.print("n2.val=");
        Serial.print(jarak3);
        Serial.write(0xff);
        Serial.write(0xff);
        Serial.write(0xff);
        Serial.print("j2.val=");

        Serial.print(jarak3);
        Serial.write(0xff);
        Serial.write(0xff);
        Serial.write(0xff);

        Serial.print("n3.val=");
        Serial.print(jarak4);
        Serial.write(0xff);
        Serial.write(0xff);
        Serial.write(0xff);
        Serial.print("j3.val=");
        Serial.print(jarak4);
        Serial.write(0xff);
        Serial.write(0xff);
        Serial.write(0xff);
        Serial.print("n4.val=");
        Serial.print(jarak4);
        Serial.write(0xff);
        Serial.write(0xff);
        Serial.write(0xff);
        Serial.print("j4.val=");

        newData = false;
        receivedChars[0]='\0';
    }
}

void recvWithStartEndMarkers()
{
    static boolean recvInProgress = false;
    static byte ndx = 0;
}

```

```
char startMarker = '<';
char endMarker = '>';
char rc;

if (BTserial.available() > 0)
{
    rc = BTserial.read();
    if (recvInProgress == true)
    {
        if (rc != endMarker)
        {
            receivedChars[ndx] = rc;
            ndx++;
            if (ndx > maxDataLength) { ndx = maxDataLength; }
        }
        else
        {
            receivedChars[ndx] = '\\0'; // terminate the string
            recvInProgress = false;
            ndx = 0;
            newData = true;
        }
    }
    else if (rc == startMarker) { recvInProgress = true; }
}
}
```

A-2 Program *Slave Bluetooth*

```
slave_menik
#include <SoftwareSerial.h>
SoftwareSerial BTserial(10, 11);

// Set DEBUG to true to output debug information to the serial monitor
boolean DEBUG = true;

// Variables used for incoming data
const byte maxDataLength = 20;

char receivedChars[21] ;

boolean newData = false;

int jarak1, jarak2, jarak3, jarak4;
double y1 = 0, x1 = 0, y1_1= 0, y2 = 0, x2 = 0, y2_1= 0;
double y3 = 0, x3 = 0, y3_1 = 0, y4 = 0, x4 = 0, y4_1= 0;
const double a = 0.1;

void setup()
{
  if (DEBUG)
  {
    // open serial communication for debugging
    Serial.begin(9600);
    Serial.println(__FILE__);
    Serial.println(" ");
  }

  // open a software serial connection to the Bluetooth module.
  BTserial.begin(38400);
  if (DEBUG) { Serial.println(F("AltSoftSerial started at 9600")); }
  newData = false;
} // void setup()

void loop()
{
  recvWithStartEndMarkers();
  //Serial.print("step 1 selesai");

  if (newData)
  { processCommand(); }
  //Serial.print("step 2 selesai");

  getjarak1();
}
```



```

    //Serial.print("step 3 selesai");
    getjarak2();
    //Serial.print("step 4 selesai");
}
|
void getjarak1()
{
    //Sensor1
    int sen_val1 = analogRead(A0);
    x1 = sen_val1 * 5.0/1023.0;
    y1= a * x1 + (1 - a) * y1_1;

    delay(50);
    y1_1 = y1;

    //Sensor 1
    if(y1>=1.6 && y1<2.84) {
        jarak1 = 66.139-(y1/0.0603); //sensor3 20-40
    }
    else if(y1>=1.27 && y1<1.6) {
        jarak1 = 89.876-(y1/0.0322); //sensor3 40-50
    }
    else if(y1>=0.9 && y1<1.27) {
        jarak1 = 118.925-(y1/0.0186); //sensor3 50-70
    }
    else if(y1>=0.76 && y1<0.9) {
        jarak1 = 134.296-(y1/0.0142); //sensor3 70-80
    }
    else if(y1>=0.64&& y1<0.76) {
        jarak1 = 142.683-(y1/0.0123); //sensor3 80-90
    }
    return jarak1;
}
|
void getjarak2()
{
    //Sensor 2
    int sen_val2 = analogRead(A1);
    x2 = sen_val2* 5.0/1023.0;
    y2= a * x2 + (1 - a) * y2_1;

    delay(50);
    y2_1 = y2;
}

```

```

//Sensor 2
if(y2>=1.6 && y2<=2.7) {
  jarak2 = 69.17-(y2/0.0545); //sensor3 20-40
}
else if(y2>=1.29 && y2<1.6) {
  jarak2 = 90.40-(y2/0.0318); //sensor3 40-50
}
else if(y2>=0.9 && y2<1.29) {
  jarak2 = 116.826-(y2/0.019); //sensor3 50-70
}
else if(y2>=0.78 && y2<0.9) {
  jarak2 = 135.507-(y2/0.0138); //sensor3 70-80
}
else if(y2>=0.65&& y2<0.78) {
  jarak2 = 152.95-(y2/0.0105); //sensor3 80-90
}
}
|
void getjarak3()
{
  int sen_val3 = analogRead(A2);
  x3 = sen_val3 * 5.0/1023.0;
  y3 = a * x3 + (1 - a) * y3_1;
  delay(50);
  y3_1 = y3;

//Sensor 3
if(y3>=1.59 && y3<=2.81) {
  jarak3 = 65.74-(y3/0.0601);; //sensor3 20-40
}
else if(y3>=1.3 && y3<1.59) {
  jarak3 = 94.75-(y3/0.029); //sensor3 40-50
}
else if(y3>=0.87 && y3<1.3) {
  jarak3 = 110.91-(y3/0.0212); //sensor3 50-70
}
else if(y3>=0.75 && y3<0.87) {
  jarak3 = 146.66-(y3/0.0114); //sensor3 70-80
}
else if(y3>=0.62 && y3<0.75) {
  jarak3 = 138.015-(y3/0.0131); //sensor3 80-90
}
}

void getjarak4()
{

```

```

//Sensor 4
int sen_val4 = analogRead(A3);
x4 = sen_val4 * 5.0/1023.0;
y4= a * x4 + (1 - a) * y4_1;

delay(50);
y4_1 = y4;

// Sensor 4
if(y4>=1.63 && y4<2.78) {
jarak4 = 67.102-(y4/0.0578); //sensor3 20-40
}
else if(y4>=1.29 && y4<1.63) {
jarak4 = 89.475-(y4/0.0324); //sensor3 40-50
}
else if(y4>=0.84 && y4<1.29) {
jarak4= 106.048-(y4/0.0227); //sensor3 50-70
}
else if(y4>=0.67 && y4<0.84) {
jarak4 = 120.061-(y4/0.0165); //sensor3 70-80
}
else if(y4>=0.55&& y4<0.67) {
jarak4 = 134.628-(y4/0.0121); //sensor3 80-90
}
}

void processCommand()
{
    Serial.println(receivedChars);

    if (strcmp ("sendjarak",receivedChars) == 0) //membandingkan 2 string
    {
        |
        BTserial.print("<");
        BTserial.print(jarak1);
        BTserial.print(",");
        BTserial.print(jarak2);
        BTserial.print(",");
        BTserial.print(jarak3);
        BTserial.print(",");
        BTserial.print(jarak4);
        BTserial.print(">");
    }

    if (strcmp("sendtombol", receivedChars) == 0 )
    {
        char receivedChars[100]= "sendtombol, buttonState1, buttonState2, buttonState3";
        char *token = strtok(receivedChars, ",");
    }
}

```

```
// DOWN
    if(token[0] == 'a' ) {
        digitalWrite(pin1, HIGH);
    }
    Serial.print(",");
    else (token[0] == 'b' ) {
        digitalWrite(pin1, LOW);
    }
// UP
    token = strtok(NULL, ",");
    if(token[0]== 'c' ) {
        digitalWrite(pin2, HIGH);
    }
    Serial.println(",");
    if(token[0]== 'd' ) {
        digitalWrite(pin2, LOW);
    }
// EMERGENCY
    token = strtok(NULL, ",");
    if(token[0]== 'e' ) {
        digitalWrite(pin3, HIGH);
    }
    Serial.println(",");
    if(token[0]== 'f' ) {
        digitalWrite(pin3, LOW);
    }

    if (DEBUG) {
        Serial.print("jarak1 is ");
        Serial.print(jarak1);
        Serial.println(",");
        Serial.print("jarak2 is ");
        Serial.print(jarak2);
        Serial.println(",");
        Serial.print("jarak3 is ");
        Serial.print(jarak3);
        Serial.println(",");
        Serial.print("jarak4 is ");
        Serial.print(jarak4);
        Serial.println("");
    }
}

newData = false;
receivedChars[0]='\0';
```

```
}

void recvWithStartEndMarkers()
{
    static boolean recvInProgress = false;
    static byte ndx = 0;
    char startMarker = '<';
    char endMarker = '>';
    char rc;

    if (BTserial.available() > 0)
    {
        rc = BTserial.read();
        if (recvInProgress == true)
        {
            if (rc != endMarker)
            {
                receivedChars[ndx] = rc;

                ndx++;
                if (ndx > maxDataLength) { ndx = maxDataLength; }
            }
            else
            {
                receivedChars[ndx] = '\0';

                recvInProgress = false;
                ndx = 0;

                newData = true;
            }
        }
        else if (rc == startMarker) { recvInProgress = true; }
    }
}
```

A-3 Program Konfigurasi *Bluetooth*

```
CONFIG$
// by Brandon Plumbo AKA Riftliger
// original version from Martyn Currey http://www.martyncurrey.com/
// HC-05 universal test sketch to enter AT-commands
// Uses hardware serial to talk to the host computer and software serial for communication with the bluetooth module
// Pins
// BT VCC to Arduino 5V out.
// BT GND to GND
// Arduino D11 to BT RX
// Arduino D10 BT TX
//
// When a command is entered in the serial monitor on the computer
// the Arduino will relay it to the bluetooth module and display the result.
//
// The HC-05 modules require both CR and NL

#include <SoftwareSerial.h>
SoftwareSerial BTSerial(10, 11); // TX,RX

char c = ' ';
boolean NL = true;

void setup()
{
  Serial.begin(9600);
  Serial.println("Sketch HC-05");
  Serial.println("Arduino with HC-05 is ready");
  Serial.println("Make sure Both NL & CR are set");
  Serial.println("");

  //Set to HC-05 default baud rate, found using AT+UART. It is usually 38400.
  BTSerial.begin(38400);
  Serial.println("BTserial started at 38400");
  Serial.println("");
}

void loop()
{
  // Read from the Bluetooth module and send to the Arduino Serial Monitor
  if (BTSerial.available())
  {
    c = BTSerial.read();
    Serial.write(c);
  }

  // Read from the Serial Monitor and send to the Bluetooth module
  if (Serial.available())
  {
    c = Serial.read();
    BTSerial.write(c);

    // Echo the user input to the main window. The ">" character indicates the user entered text.
    if (NL) {
      Serial.print(">");
      NL = false;
    }
    Serial.write(c);
    if (c == 10) {
      NL = true;
    }
  }
}
```

DAFTAR RIWAYAT HIDUP



Menik Nurhidayati lahir di Kota Lamongan pada tanggal 24 Agustus 1997. Penulis adalah anak kedua dari dua bersaudara. Penulis menempuh pendidikan dasar di MI Unnggulan Sabilillah pada tahun 2004-2010, kemudian SMPN 4 Lamongan tahun 2010-2013, dan lulus dari MA Negeri 1 Lamongan pada tahun 2016. Tahun 2016 juga, penulis diterima sebagai mahasiswa jurusan Teknik Elektro Otomasi di Institut Teknologi Sepuluh Nopember (ITS). Semasa kuliah, penulis mengikuti keanggotaan *Laboratorium Cyber Physical*, Otomasi, dan Robot Industri. Penulis telah menyelesaikan magang dan pengerjaan Proyek Akhir di PT Bhimasena Research and Development.