



**RANCANG BANGUN IKAT PINGGANG PINTAR SEBAGAI ALAT
BANTU TUNANETRA**



TUGAS AKHIR

Program Studi

S1 Sistem Komputer

**INSTITUT BISNIS
DAN INFORMATIKA**

stikom
SURABAYA

Oleh :

HANS MULIONO

13.41020.0013

FAKULTAS TEKNOLOGI DAN INFORMATIKA

INSTITUT BISNIS DAN INFORMATIKA STIKOM SURABAYA

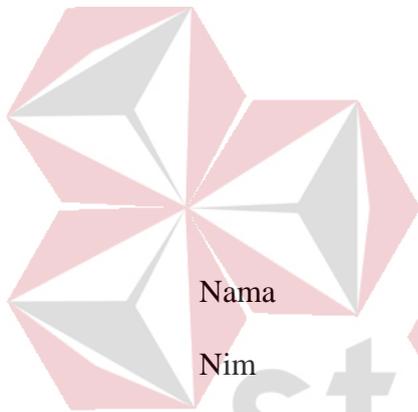
2018

**RANCANG BANGUN IKAT PINGGANG PINTAR SEBAGAI ALAT
BANTU TUNANETRA**

TUGAS AKHIR

Diajukan sebagai salah satu syarat untuk menyelesaikan

Program Sarjana Komputer



Disusun Oleh : INSTITUT BISNIS
DAN INFORMATIKA

Nama : Hans Muliono
Nim : 13.41020.0013
Program : S1 (Strata Satu)
Jurusan : Sistem Komputer

**FAKULTAS TEKNOLOGI DAN INFORMATIKA
INSTITUT BISNIS DAN INFORMATIKA STIKOM SURABAYA**

2018



Tugas Akhir ini saya persembahkan kepada

Tuhan Yesus Kristus, Orang Tua, saudara dan teman yang selalu memberi dukungan dan mendoakan saya.

Terimakasih kepada Dosen Pembimbing

Serta para dosen - dosen yang memberikan ilmu dan memberikan motivasi kepada saya. Untuk Seluruh rekan - rekan di S1 Sistem Komputer dan dikampus Institut Bisnis dan Informatika Stikom Surabaya yang membantu dan memberikan motivasi kepada saya.

Beserta semua orang yang telah membantu.

**RANCANG BANGUN IKAT PINGGANG PINTAR SEBAGAI ALAT
BANTU TUNANETRA**

Dipersiapkan dan disusun oleh

Hans Muliono

Nim : 13.41020.0013

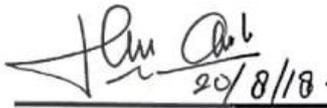
Telah diperiksa, diuji dan disetujui oleh Dewan Penguji

Pada : Agustus 2018

Susunan Dewan Penguji

Pembimbing

**I. Harianto, S.Kom., M.Eng.
NIDN. 0722087701**


20/8/18

**II. Yosefine Triwidayastuti, M.T.
NIDN. 0729038504**

INSTITUT BISNIS
DAN INFORMATIKA


Pembahas

**I. Heri Pratikno, M.T., MTCNA., MTCRE
NIDN. 0716117302**

STIKOM SURABAYA


Tugas Akhir ini telah diterima sebagai salah satu persyaratan
Untuk memperoleh gelar Sarjana



FAKULTAS TEKNOLOGI
DAN INFORMATIKA

stikom
SURABAYA

Dr. Jusak

Dekan Fakultas Teknologi dan Informatika

23/8/18

**INSTITUT BISNIS DAN INFORMATIKA STIKOM
SURABAYA**

**SURAT PERNYATAAN
PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA
ILMIAH**

Sebagai mahasiswa Institut Bisnis dan Informatika Stikom Surabaya, saya :

Nama : Hans Muliono
NIM : 13.4102.0013
Program Studi : S1 Sistem Komputer
Fakultas : Fakultas Teknologi dan Informatika
Jenis Karya : Tugas Akhir
Judul Karya : **RANCANG BANGUN IKAT PINGGANG PINTAR
SEBAGAI ALAT BANTU TUNANETRA**

Menyatakan dengan sesungguhnya bahwa :

1. Demi pengembangan ilmu pengetahuan, Teknologi dan Seni, saya menyetujui memberikan kepada Institut Bisnis dan Informatika Stikom Surabaya Hak Bebas Royalti Non-Eksklusif (*Non-Exclusive Royalti Free Right*) atas seluruh isi/ sebagian karya ilmiah saya tersebut di atas untuk disimpan, dialih mediakan dan dikelola dalam bentuk pangkalan data (*database*) untuk selanjutnya didistribusikan atau dipublikasikan demi kepentingan akademis dengan tetap mencantumkan nama saya sebagai penulis atau pencipta dan sebagai pemilik Hak Cipta.
2. Karya tersebut di atas adalah karya asli saya, bukan plagiat baik sebagian maupun keseluruhan. Kutipan, karya atau pendapat orang lain yang ada dalam karya ilmiah ini adalah semata hanya rujukan yang dicantumkan dalam Daftar Pustaka saya.
3. Apabila dikemudian hari ditemukan dan terbukti terdapat tindakan plagiat pada karya ilmiah ini, maka saya bersedia untuk menerima pencabutan terhadap gelar kesarjanaan yang telah diberikan kepada saya.

Demikian surat pernyataan ini saya buat dengan sebenarnya

Surabaya, Agustus 2018

Surabaya, Agustus 2018
METERAI
MPEL
20
2AFF128733991
000
RIBU RUPIAH
muliono
NIM : 13.41020.0013

ABSTRAK

Tunanetra adalah penyakit yang diderita oleh manusia. Tetapi sangat jarang ada teknologi yang masuk untuk membantu para penderita penyakit Tunanetra. Dengan alasan itulah penulis membuat alat bantu untuk tunanetra yang dibentuk dalam ikat pinggang. Dimana ikat pinggang sendiri adalah kebutuhan sehari-hari manusia. Ikat pinggang tersebut dapat digunakan setiap hari dan tunanetra tidak perlu memakai tongkat ataupun pemandu jalan yang biasanya berbentuk binatang. Alat ini bisa mendeteksi halangan di bawah, di depan dan di atas. Alat ini terdiri dari sensor *ultrasonic* yang terdiri dari 3 jumlah. Alat ini mempunyai program yang mampu mendeteksi halangan sejauh dua meter ke depan. Alat ini juga dilengkapi dengan DFPlayer untuk memberikan informasi kepada tunanetra. Hasil percobaan sensor terdapat selisih koma dengan rata-rata error 2.28% untuk sensor atas dan depan, serta 2.31% untuk sensor bawah. Data error tersebut didapatkan dalam 17 percobaan.

Kata kunci: Tunanetra, Ikat Pinggang, Sensor Ultrasonik, Pendeteksi Halangan.

KATA PENGANTAR

Pertama-tama penulis panjatkan puji dan syukur kepada Tuhan Yang Maha Esa atas berkar dan Rahmat penulis dapat menyelesaikan laporan penelitian ini dengan judul “RANCANG BANGUN IKAT PINGGANG PINTAR SEBAGAI ALAT BANTU TUNANETRA” yang merupakan salah satu syarat menempuh Tugas Akhir pada Program Studi S1 Sistem Komputer di Institut Bisnis Stikom Surabaya.

Di dalam buku Laporan Tugas Akhir ini dilakukan pembahasan mengenai pembuatan sistem otomasi Ikat pinggang pintar. Harapan penulis semoga laporan ini dapat berguna dan bermanfaat untuk menambah wawasan bagi pembacanya. Penulis juga menyadari dalam penulisan Laporan Tugas Akhir ini masih terdapat banyak kekurangan.

Dalam usaha menyelesaikan Laporan Tugas Akhir ini penulis banyak mendapatkan bantuan dari berbagai pihak baik moral maupun materi. Oleh karena itu penulis mengucapkan terima kasih dan penghargaan setinggi-tingginya kepada:

1. Ayah, Ibu dan saudara - saudara saya tercinta yang telah memberikan dukungan dan bantuan baik moral maupun materi sehingga penulis dapat menempuh dan menyelesaikan Tugas Akhir maupun laporan ini.
2. Bapak Dr. Jusak selaku Dekan Fakultas Teknologi dan Informatika (FTI) Institut Bisnis dan Informatika Stikom Surabaya telah membantu proses penyelesaian Tugas Akhir yang dibuat oleh penulis dengan baik.

3. Kepada Bapak Pauladie Susanto, S.Kom., M.T., selaku Ketua Program Studi Sistem Komputer Stikom Surabaya atas ijin yang diberikan untuk mengerjakan Tugas Akhir ini.
4. Kepada Bapak Harianto, S.Kom, M.Eng., dan Ibu Yosefine Triwidyastuti, M.T., selaku Dosen Pembimbing. Terima kasih atas bimbingan yang diberikan sehingga penulis dapat menyelesaikan Tugas Akhir dengan baik.
5. Kepada Bapak Heri Pratikno, M.T., MTCNA., MTCRE. selaku Dosen Penguji memberi masukan dalam menyusun buku Tugas Akhir.
6. Semua staf dosen yang telah mengajar dan memberikan ilmunya.
7. Terima kasih terhadap seluruh rekan - rekan S1 Sistem Komputer khususnya rekan - rekan seperjuangan angkatan 2013 yang selalu memberikan semangat dan bantuannya.
8. Serta semua pihak lain yang tidak dapat disebutkan secara satu per satu, yang telah membantu dalam menyelesaikan Tugas Akhir ini baik secara langsung maupun tidak langsung.

Surabaya, Agustus 2018

Penulis

DAFTAR ISI

HALAMAN JUDUL.....	1
HALAMAN SYARAT	<u>ii</u>
MOTTO	ii
HALAMAN PERSEMBAHAN	iii
HALAMAN PENGESAHAN.....	v
HALAMAN PERNYATAAN	<u>vi</u>
ABSTRAK.....	<u>vii</u>
KATA PENGANTAR	<u>viii</u>
DAFTAR ISI.....	<u>xi</u>
DAFTAR GAMBAR	x
DAFTAR TABEL.....	<u>ix</u>
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang Masalah.....	<u>2</u>
1.2 Rumusan Masalah	2
1.3 Tujuan.....	<u>3</u>
1.4 Manfaat.....	<u>3</u>
1.5 Batasan Masalah.....	<u>3</u>
1.6 Sistematika Penulisan Laporan.....	<u>4</u>
BAB II LANDASAN TEORI.....	7
2.1 Tinjauan Pustaka	<u>7</u>
2.2 Arduino Nano	<u>8</u>
2.3 Sensor Ultrasonik	<u>10</u>

2.4	DfPlayer.....	<u>12</u>
2.5	Load Speaker	<u>14</u>
2.6	Koneksi Arduino nano.....	<u>14</u>
2.7	Ikat Pinggang.....	<u>16</u>
2.8	Power Bank	<u>17</u>
BAB III METODE PENELITIAN.....		<u>18</u>
3.1	Perancangan Sistem.....	<u>19</u>
3.2	<i>Flowchart</i> Sistem Pengering Pakaian.....	<u>21</u>
3.3	Rangkaian Pada Otomasi Sistem.....	<u>22</u>
3.4	Model Perancangan	<u>22</u>
3.5	Ukuran Dimensi.....	<u>23</u>
3.6	Struktur Material	<u>24</u>
3.7	Pemrograman.....	<u>25</u>
BAB IV HASIL DAN PEMBAHASAN		<u>26</u>
4.1	Hasil Pengujian Arduino Uno	<u>25</u>
4.2	Hasil Pengujian Sensor <i>Ultrasonic</i>	<u>26</u>
4.3	Hasil Pengujian <i>Ultrasonic</i> Atas	<u>29</u>
4.4	Hasil Pengujian <i>Ultrasonic</i> bawah	<u>30</u>
4.5	Hasil Pengujian Us 100	<u>30</u>
4.6	Hasil Pengujian DFplyer	<u>36</u>
4.7	Hasil Pengujian seluruh rangkaian	<u>36</u>
BAB V PENUTUP.....		<u>37</u>
5.1	Kesimpulan.....	<u>38</u>
5.2	Saran	<u>38</u>

DAFTAR PUSTAKA	<u>39</u>
LAMPIRAN	<u>40</u>
BIODATA PENULIS	<u>41</u>



DAFTAR GAMBAR

Gambar 2.1 Arduino nano.....	10
Gambar 2.2 Koneksi Arduino nano ke DFplayer.....	10
Gambar 2.3 Ikat Pinggang.....	13
Gambar 3.1 Blok Diagram	13
Gambar 3.2 Flowchart Sistem.....	14
Gambar 3.3 Rangkaian otomasi sistem.....	15
Gambar 3.4 Model rancangan	15
Gambar 3.5 Foto Hardware rancangan	16
Gambar 3.6 library program.....	17
Gambar 3.7 Variable Program	17
Gambar 3.8 Setup Program.....	19
Gambar 3.9 Looping Program.....	21
Gambar 4.1 Upload Berhasil pada Arduino IDE	23
Gambar 4.2 Hasil dari serial monitor.....	27
Gambar 4.4 Program ultraviolet.....	30
Gambar 4.5 Program ultraviolet atas	32
Gambar 4.6 Program ultraviolet bawah	36
Gambar 4.7 Program Us-100	37
Gambar 4.8 Program Dfplayer	39

DAFTAR TABEL

Tabel 4.1 Hasil Uji coba <i>Ultrasonic</i>	36
Tabel 4.2 Hasil Uji coba <i>Ultrasonic</i> bawah	37
Tabel 4.3 Hasil Uji coba DFplayer	38
Tabel 4.4 Hasil Uji coba seluruh perangkat	39



BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Indera penglihatan adalah salah satu sumber informasi vital bagi manusia. Tidak berlebihan apabila dikemukakan bahwa sebagian besar informasi yang diperoleh manusia berasal dari indera penglihatan, sedangkan selebihnya berasal dari panca indera yang lain. Dengan demikian, dapat dipahami bila seseorang mengalami gangguan pada indera penglihatan, maka kemampuan aktifitasnya akan jadi sangat terbatas, karena informasi yang diperoleh akan jauh berkurang dibandingkan mereka yang berpenglihatan normal. Saat ini penyandang Tunanetra umumnya menggunakan alat bantu jalan berupa tongkat putih atau anjing terlatih untuk membantu pergerakan dan meningkatkan keamanan dan kemandirian pada saat berjalan. *System* ini memang banyak ditemukan, tetapi kesempurnaan dalam pembuatan *system* ini masih kurang. Salah satu *system* yang sudah ada adalah mengacu pada (Azhari, 2015). Dimana *system* tersebut menggunakan sepatu sebagai media dan *buzzer* sebagai pusat suara. Contoh yang lain yaitu mengacu pada (Bimantoro, 2015) dimana sebagai media menggunakan tongkat dan pusat informasi menggunakan *buzzer*.

Perbedaan pada *system* yang penulis buat adalah pada media beserta informasi sebagai pusat suara. Untuk media, penulis menggunakan ikat pinggang, sedangkan pusat suara menggunakan DFPlayer. Dimana DFPlayer dapat mengeluarkan suara dan suara tersebut yakni adalah jarak dari manusia ke halangan tersebut. Informasi tersebut berupa data yang didapatkan dari sensor *ultrasonic*. Selain itu perbedaan pada *system* ini dengan *system* yang sudah ada adalah terdapat

pada pendeteksi halangan, dimana *system* penulis bisa mendeteksi pada halangan yang berada diatas.

Dengan mempunyai informasi yang cukup terhadap jalur perjalanan yang akan dilewati penyandang Tunanetra dapat lebih nyaman untuk bernavigasi pada lingkungan yang belum dikenal. Tujuan dari penelitian ini adalah merencanakan dan merealisasikan sebuah alat bantu jalan yang dapat digunakan oleh penyandang Tunanetra dalam melakukan perjalanan secara mandiri. Media yang diambil penulis untuk mewujudkan alat ini adalah dengan menggunakan Ikat pinggang sebagai pusat informasi bagi Tunanetra. Ikat pinggang tersebut terdapat kombinasi antara mikrokontrol sebagai pusat kontrol dimana mikrokontrol. Tersebut mendapatkan *input* berupa sensor *ultrasonic* dan *output* berupa suara dengan indikasi jarak. Sensor *ultrasonic* tersebut berjumlah dua dimana sensor tersebut dapat mendeteksi lubang, batu, sungai dan halangan didepan pengguna. Penulis berharap pada alat ini dapat digunakan untuk membantu Tunanetra dalam berjalan.

1.2 Rumusan Masalah

Adapun rumusan masalah yang terdapat dalam pengerjaan tugas akhir ini diantaranya adalah :

1. Bagaimana sensor *ultrasonic* dapat mengetahui jarak didepan, diatas dan dibawah?
2. Bagaimana sistem dapat memberitahukan orang buta dalam bentuk suara yang menyebutkan jarak halangan, keberadaan lubang, dan keberadaan halangan diatas?

1.3 Batasan Masalah

Dalam perancangan dan pembuatan alat ini, terdapat beberapa batasan masalah diantara lain :

1. Pusat kontrol adalah Arduino Nano.
2. Sensor yang digunakan adalah sensor *ultrasonic*.
3. Informasi yang digunakan berupa DFPlayer.
4. Alat ini hanya mendeteksi halangan didepan, diatas dan dibawah.
5. Posisi sensor diatur secara manual sesuai dengan ketinggian pemakai alat tersebut.

1.4 Tujuan

Tujuan dari penelitian yang akan dilakukan adalah sebagai berikut :

1. Dapat merancang sensor *ultrasonic* yang dapat mengetahui jarak didepan, diatas dan dibawah.
2. Merancang *system* yang dapat memberitahukan orang buta dalam bentuk suara yang menyebutkan jarak halangan, keberadaan lubang, dan keberadaan halangan diatas.

1.5 Manfaat

Manfaat pada penelitian yang akan dilakukan adalah sebagai berikut :

1. Digunakan untuk membantu Tunanetra ketika berjalan.
2. Untuk meningkatkan pemanfaatan teknologi pada masa ini.
3. Untuk menghilangkan alat bantu konvensional, seperti anjing penunjuk arah dan tongkat.

1.6 Sistematika Penulisan Laporan

Tugas akhir secara garis besar terseusun dari 5 (lima) bab, yaitu diuraikan sebagai berikut:

BAB I PENDAHULUAN

Pada Bab ini akan dibahas mengenai latar belakang masalah, batasan masalah, tujuan penulisan, dan sistematika penulisan.

BAB II LANDASAN TEORI

Pada Bab ini akan dibahas teori penunjang dari permasalahan, yaitu membahas mengenai Mikrokontroler Arduino, Sensor *Ultrasonic*, Servo, DFPlayer, dan lain-lain.

BAB III METODE PENELITIAN DAN PERANCANGAN SISTEM

Pada Bab ini akan dibahas tentang blok diagram *system* serta metode yang dilakukan dalam mewujudkan pembuatan alat. Pada bab ini juga akan dibahas tentang program dan perancangan hardware.

BAB IV HASIL DAN PEMBAHASAN

Pada Bab ini akan dibahas mengenai hasil dari pengujian masing-masing komponen pendukung dalam pembuatan alat, dimana nantinya hasil dari pengujian masing-masing komponen akan bekerja dengan baik atau tidak. Kemudian juga membahas tentang uji coba perangkat secara keseluruhan.

BAB V PENUTUP

Berisi kesimpulan yang dapat dari hasil penelitian berdasarkan rumusan masalah serta saran untuk perkembangan penelitian selanjutnya.



BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Menurut (Azhari, 2015), dengan penelitiannya yang berjudul rancang bangun alat pembantu tunanetra berbasis ATmega 16. Penelitian ini bertujuan mengembangkan sebuah alat bantu yang dapat digunakan untuk mendeteksi halangan bagi penyandang tunanetra. Penelitian tersebut menggunakan media sepatu. Dibandingkan dengan alat sejenis keunggulan dari alat ini yaitu mampu mendeteksi halangan didepan dan dibawah. Alat ini dikembangkan dengan ATmega 16 karena kecepatan dalam eksekusi program yang lebih cepat dan konsumsi daya terhadap kecepatan eksekusi perintah rendah. Alat ini juga dikembangkan dengan menggunakan sensor SRF04 karena performa yang stabil dan akurasi yang tinggi.

Metode yang digunakan dalam pembuatan rancang bangun *tongkat ultrasonic* pendeteksi halangan untuk penyandang tunanetra berbasis ATmega 16 ini adalah Metode rancang bangun, dengan Metode ini didapatkan teknik perancangan yang terdiri dari beberapa tahap yaitu : (1) Identifikasi kebutuhan; (2) Analisis kebutuhan; (3) Perancangan perangkat keras dan perangkat lunak; (4) Pembuatan alat; (5) Pengujian alat dan; (6) Pengoperasian alat. Perangkat keras terdiri dari : (1) *System* minimum ATmega 16 sebagai pendendali utama; (2) Sensor SRF04 sebagai mendeteksi obyek dan lubang jalan; (3) *Buzzer* memberikan peringatan bila terdapat obyek yang terdeteksi oleh sensor; (4) Roda menjaga jarak antara sensor bawah dengan permukaan tanah; (5) Tombol *ON/OFF* menghidupkan

atau mematikan *system* Sedangkan menurut (Bimantoro, 2015), penelitian yang diciptakan berupa *tongkat* untuk membantu tunanetra. Sebenarnya *tongkat* itu merupakan *tongkat* biasa. Yang membedakan, *tongkat* tersebut dilengkapi dua sensor *ultrasonic*. Sensor itu diletakkan dibagian atas dan bawah *tongkat*. Sensor bagian atas digunakan untuk mendeteksi benda seperti tembok rumah, *pintu*, pagar, tiang, atau benda lain. Sementara itu, sensor bagian bawah diharapkan memudahkan mendeteksi permukaan jalan yang tidak rata. Misalnya, ada batu ataupun lubang disekitar pengguna. Sensor tersebut akan menghasilkan bunyi jika terhalang sesuatu.

Penulis tersebut membuat *tongkat* itu dengan menggunakan pipa paralon. Dibagian bawah, diberi roda untuk memudahkan pengguna. Termasuk memudahkan gerak *tongkat* ketika mendeteksi benda dihadapannya. Suara sensor bagian atas dan bawah juga dibedakan agar bisa memudahkan pengguna.

2.2 Arduino Nano

Menurut (Owen, 2004), Arduino Nano adalah salah satu varian dari produk *board* mikrokontroller keluaran Arduino. Arduino Nano adalah *board* Arduino terkecil, menggunakan mikrokontroller Atmega 328 untuk Arduino Nano 3.x dan ATmega168 untuk Arduino Nano 2.x. varian ini mempunyai rangkaian yang sama dengan jenis Arduino Duemilanove, tetapi dengan ukuran dan desain PCB yang berbeda. Arduino Nano tidak dilengkapi dengan soket catu daya, tetapi terdapat *pin* untuk catu daya luar atau dapat menggunakan catu daya dari mini USB *port*. Arduino Nano didesain dan diproduksi oleh Gravitech. Arduino Nano mempunyai 14 *pin* digital yang dapat digunakan

sebagai *pin input* atau *output*. *Pin* ini akan mengeluarkan tegangan 5V untuk mode *HIGH* (logika 1) dan 0V untuk mode *LOW* (logika 0) jika dikonfigurasi sebagai *pin output*. Jika dikonfigurasi sebagai *pin input*, maka ke 14 *pin* ini dapat menerima tegangan 5V untuk mode *HIGH* (logika1) dan 0V untuk mode *LOW* (logika 0). Besar arus listrik yang diijinkan untuk melewati *pin* digital I/O adalah 40 mA. *Pin* digital I/O ini juga sudah dilengkapi dengan resistor *pull-up* sebesar 20-50 k Ω . Ke 14 *pin* digital I/O ini selain berfungsi sebagai *pin* I/O juga mempunyai fungsi khusus yaitu :

Pin D₀ dan *pin* D₁ juga berfungsi sebagai *pin* TX dan RX untuk komunikasi data serial. Kedua *pin* ini terhubung langsung ke *pin* IC FTDI USB_TTL. *Pin* D₂ dan *pin* D₃ juga berfungsi sebagai *pin* untuk interupsi eksternal. Kedua *pin* ini dapat dikonfigurasi untuk pemacu interupsi dari sumber eksternal. Interupsi dapat terjadi ketika timbul kenaikan atau penurunan tegangan pada *pin* D₂ atau *pin* D₃. *Pin* D₄, *pin* D₅, *pin* D₆, *pin* D₉, *pin* D₁₀ dan *pin* D₁₁ dapat digunakan sebagai *pin* PWM (*pulse width modulator*). *Pin* D₁₀, *pin* D₁₁, *pin* D₁₂ dan *pin* D₁₃, ke empat *pin* ini dapat digunakan untuk komunikasi mode SPI. *Pin* D₁₃ terhubung ke sebuah LED.

Arduino Nano juga dilengkapi dengan 8 buah *pin* analog, yaitu *pin* A₀, A₁, A₂, A₃, A₄, A₅, A₆ dan A₇. *Pin* analog ini terhubung ke ADC (*analog to digital converter*) internal yang terdapat didalam mikrokontroler. Pada kondisi awal, *pin* analog ini dapat mengukur variasi tegangan dari 0V sampai 5V pada arus searah dengan besar arus maksimum 40 mA. Lebar *range* ini dapat diubah dengan memberikan sebuah tegangan referensi dari luar melalui *pin* V_{ref}. *Pin* analog selain dapat digunakan untuk *input* data analog, juga dapat digunakan sebagai *pin* digital I/O, kecuali *pin* A₆ dan A₇ yang hanya dapat digunakan untuk *input* data analog saja.

Fungsi khusus untuk *pin* analog antara lain : *pin* A₄ untuk *pin* SDA, *pin* A₅ untuk *pin* SCL, *pin* ini digunakan untuk komunikasi I2C. *Pin* A_{ref} digunakan sebagai *pin* tegangan referensi dari luar untuk mengubah *range* ADC. *Pin* reset, *pin* ini digunakan untuk *reset board* Arduino Nano, yaitu dengan menghubungkan *pin* ini ke *ground* selama beberapa milidetik. Gambar pada Arduino Nano dapat dilihat pada Gambar 2.1



Gambar 2.1. Arduino Nano

2.3 Sensor *Ultrasonic*

Sensor *ultrasonic* adalah sebuah sensor yang berfungsi untuk mengubah besaran fisis (bunyi) menjadi besaran listrik dan sebaliknya. Cara kerja sensor ini didasarkan pada prinsip dari pantulan suatu gelombang suara sehingga dapat dipakai untuk menafsirkan eksistensi (jarak) suatu benda dengan frekuensi tertentu. Disebut sebagai sensor *ultrasonic* karena sensor ini menggunakan gelombang *ultrasonic* (bunyi *ultrasonic*). Gelombang *ultrasonic* adalah gelombang bunyi yang mempunyai frekuensi sangat tinggi yaitu 20.000 Hz. Bunyi *ultrasonic* tidak dapat didengar oleh telinga manusia. Bunyi *ultrasonic* dapat didengar oleh anjing, kucing, kelelawar, dan lumba-lumba. Bunyi *ultrasonic* bisa merambat melalui zat padat,

cair dan gas. Reflektivitas bunyi *ultrasonic* di permukaan zat padat hampir sama dengan reflektivitas bunyi *ultrasonic* di permukaan zat cair. Akan tetapi, gelombang bunyi *ultrasonic* akan diserap tekstil dan busa.

Pada sensor *ultrasonic*, gelombang *ultrasonic* dibangkitkan melalui sebuah alat yang disebut dengan Piezoelektrik (umumnya berfrekuensi 40kHz) ketika sebuah osilator diterapkan pada benda tersebut. Secara umum, alat ini akan menembakkan gelombang *ultrasonic* menuju suatu area atau suatu target. Setelah gelombang menyentuh permukaan target, maka target akan memantulkan kembali gelombang tersebut. Gelombang pantulan dari target akan ditangkap oleh sensor, kemudian sensor menghitung selisih antara waktu pengiriman gelombang dan waktu gelombang pantul diterima. Komunikasi pada Arduino terdapat VCC, GND, *Trigger* dan *Echo*. Komunikasi tersebut *trigger* dan *echo* sudah disetting dalam Arduino. Dan Arduino tersebut juga menggunakan *library* khusus untuk sensor tersebut (Bimantoro, 2015). Gambar sensor *ultrasonic* dapat dilihat pada Gambar 2.2 dibawah ini.

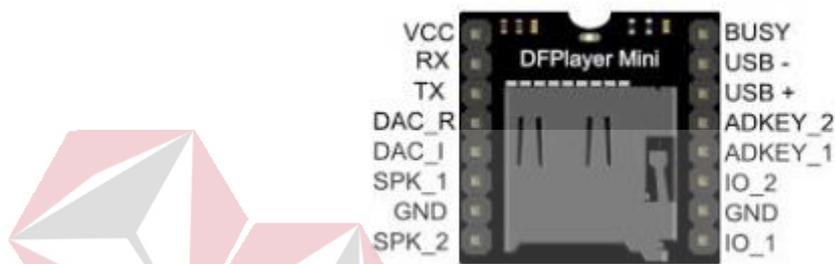


Gambar 2.2 Sensor *Ultrasonic*

2.4 DFPlayer

Menurut (Hurisantri, 2016), DFPlayer Mini adalah modul MP3 yang kecil dan murah dengan sambungan *Output* yang sangat sederhana yang bisa langsung

disambungkan ke *Speaker* atau *Headset*. Modul ini sangat mudah diakses hanya dengan menggunakan perintah serial melalui *PIN TX RX*. Selain itu modul ini juga *support SD card* dengan format *Fat32* yang berkapasitas hingga 32GB. Penggunaannya bermacam-macam seperti pemutar music sederhana, mesin pemanggil antrian hingga suara navigasi untuk *system GPS*. Gambar DFPlayer dapat dilihat pada Gambar 2.3 dibawah ini.



Gambar 2.3 DFPlayer

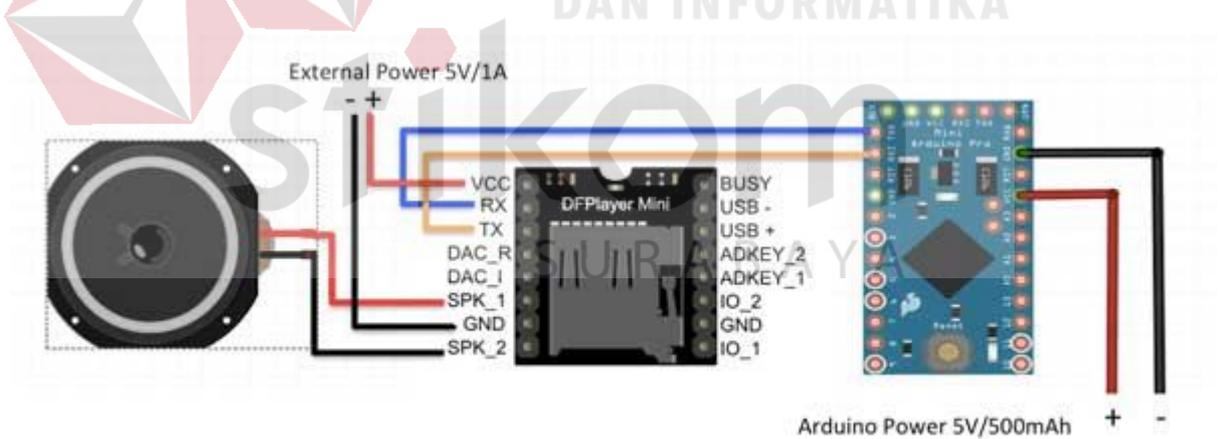
2.5 *Speaker*

Menurut, (Hurisantri, 2016) *speaker* adalah sebuah komponen elektronika yang berfungsi untuk mengubah getaran listrik menjadi getaran suara. Pada dasarnya prinsip kerja *laudspeaker* hampir sama dengan *buzzer*, jadi *laudspeaker* juga terdiri dari kumparan yang terpasang pada diafragma dan kemudian kumparan tersebut dialiri arus sehingga menjadi elektromagnetik, kumparan tadi akan tertarik kedalam atau keluar, tergantung dari arah arus dan polaritas magnetnya, karena kumparan dipasang pada diafragma maka setiap gerakan kumparan akan menggerakkan diafragma secara bolak-balik sehingga membuat udara bergetar yang akan menghasilkan suara. *Laudspeaker* bisaa digunakan sebagai indikator bahwa proses telah selesai atau terjadi suatu kesalahan pada sebuah alat (*alarm*).Speaker dapat dilihat pada Gambar 2.4 dibawah ini.

Gambar 2.4 *Laudspeaker*

2.6 Koneksi Arduino Nano dan DFPlayer

Komunikasi *DFPlayer* dan Arduino Nano adalah dengan memanfaatkan RX dan TX pada masing-masing perangkat. RX dan TX berfungsi untuk mengirim dan menerima data pada perangkat. Untuk penguat suara akan ditambahkan dengan amplifier dan *port* untuk *headset*. Tegangan pada perangkat adalah 5V/1A. komunikasi juga bisa menggunakan kabel serial sebagai pengiriman data. Contoh dapat dilihat pada Gambar 2.5.



Gambar 2.5 Koneksi Arduino Nano ke DFPlayer

2.7 Ikat Pinggang

Kaum pria masih sering menganggap ikat *pinggang* sebagai aksesoris yang sepele, padahal cara mengenakan ikat *pinggang* merupakan salah satu tolak ukur

menilai kemampuan *fashion* pria. Jika anda mampu mengenakan ikat *pinggang* yang tepat, artinya anda sudah tahu cara berpenampilan dengan baik. Sebaliknya jika ikat *pinggang* anda menimbulkan kejanggalaan atau merusak keserasian penampilan maka anda masih perlu belajar bagaimana cara berpakaian yang baik. Namun anda tidak perlu khawatir berlebihan karena dasar-dasar pemilihan ikat *pinggang* yang tepat tidak begitu sulit untuk dipelajari dan kebanyakan berhubungan dengan logika sehari-hari. Sedangkan sisanya tergantung selera pribadi masing-masing dan ikat *pinggang* memberikan ruang ekspresi yang luas untuk itu. Contoh bisa dilihat Gambar 2.6 dibawah ini.



Gambar 2.6. Ikat *Pinggang*

2.8 *Power bank*

Pengisi baterai *portable* adalah sebuah perangkat yang digunakan untuk memasukkan energi listrik kedalam baterai yang bisa diisi ulang tanpa harus menghubungkan perangkat tersebut pada *outlet* listrik. Pengisi baterai ini disebut *portable* karena berbeda dengan pengisi baterai yang harus dihubungkan pada *outlet* listrik, pengisi *portable* dapat digunakan tanpa harus menghubungkan pada perangkat listrik. Namun pengisi baterai *portable* ini memiliki daya tampung energi listrik sehingga ketika daya tersebut telah habis terpakai, energi listrik harus

kembali diisi kembali dengan cara menghubungkan kabel dengan *outlet* listrik. Pengisi baterai *portable* ini tidak hanya bisa untuk mengisi ulang baterai *handphone*, tetapi juga dapat mengisi ulang baterai pada perangkat lain seperti iPod, iPad, mp3, *player*, tablet, dan perangkat lainnya. Kabel konektor yang menghubungkan perangkat dengan pengisi baterai *portable* pada satu ujung kabel pengisi baterai penghubung USB yang dicolokkan pada pengisi baterai *portable* dan ujung yang lain berbentuk sesuai dengan tempat pengisi perangkat yang disesuaikan. *Power bank* dapat dilihat pada Gambar 2.7 dibawah ini.



Gambar 2.7 *Power bank*

2.9 Kabel *Jumper*

Dalam merancang sebuah desain peralatan elektronik tentunya sangat dibutuhkan sebuah kabel untuk menghubungkan komponen elektronik yang satu dengan komponen elektronik yang lainnya. Maka dari itu **Kabel *Jumper Breadboard Male to male*** merupakan salah satu jenis kabel *jumper* untuk *breadboard* yang dapat Anda gunakan untuk menghubungkan komponen-komponen elektronik yang satu dengan yang lainnya. **Kabel *Jumper*** ini dapat digunakan untuk menyambungkan komponen elektronik yang satu dengan yang

lainnya pada saat membuat penelitian prototipe dengan menggunakan *breadboard*. **Kabel Jumper untuk Breadboard** berfungsi untuk menghubungkan beberapa *breadboard*, menghubungkan antartitik pada pcb *single slide* dan juga dapat digunakan untuk menghubungkan jalur rangkaian yang terputus dengan cara menjumpernya. Kabel *Jumper* dapat dilihat pada Gambar 2.8 dibawah ini.



Gambar 2.8 Kabel *Jumper Breadboard*

2.10 *Push button*

Push button switch (saklar tombol tekan) adalah perangkat / saklar sederhana yang berfungsi untuk menghubungkan atau memutuskan aliran arus listrik dengan *system* kerja tekan *unlock* (tidak mengunci). *System* kerja *unlock* disini berarti saklar akan bekerja sebagai *device* penghubung atau pemutus aliran arus listrik saat tombol ditekan, dan saat tombol tidak ditekan (dilepas), maka saklar akan kembali pada kondisi normal. Sebagai *device* penghubung atau pemutus, *push button switch* hanya memiliki 2 kondisi, yaitu *On* dan *Off* (1 dan 0). Istilah *On* dan *Off* ini menjadi sangat penting karena semua perangkat listrik yang memerlukan sumber energi listrik pasti membutuhkan kondisi *On* dan *Off*. Karena *system* kerjanya yang *unlock* dan langsung berhubungan dengan operator, *push button switch* menjadi *device* paling utama yang bisaa digunakan untuk memulai dan mengakhiri kerja mesin di industri. Secanggih apapun sebuah mesin bisa dipastikan *system* kerjanya tidak

terlepas dari keberadaan sebuah saklar seperti *push button switch* atau perangkat lain yang sejenis yang bekerja mengatur pengkondisian *On* dan *Off*. Berdasarkan fungsi kerjanya yang menghubungkan dan memutuskan, *push button switch* mempunyai 2 tipe kontak yaitu NC (*Normally close*) dan NO (*Normally open*).

- NO (*Normally open*), merupakan kontak terminal dimana kondisi normalnya terbuka (aliran arus listrik tidak mengalir). Dan ketika tombol saklar ditekan, kontak yang NO ini akan menjadi menutup (*Close*) dan mengalirkan atau menghubungkan arus listrik. Kontak NO digunakan sebagai penghubung atau menyalakan *system circuit* (*Push button ON*).
- NC (*Normally close*), merupakan kontak terminal dimana kondisi normalnya tertutup (mengalirkan arus listrik). Dan ketika tombol saklar *push button* ditekan, kontak NC ini akan menjadi membuka (*Open*), sehingga memutus aliran arus listrik. Kontak NC digunakan sebagai pemutus atau mematikan *system circuit* (*Push button Off*).

Gambar *Push button* dapat dilihat pada Gambar 2.8 dibawah ini.



Gambar 2.9 *Push button*

2.11 Rumus Segitiga Siku- Siku

Segitiga siku-siku adalah segitiga yang satu sudutnya tepat sebesar 90° (tegak lurus atau siku-siku). Dalam bahasa Inggris disebut *right triangle* atau *right-angled triangle*, dan dulu disebut *rectangled triangle*. Sisi yang berhadapan dengan sudut tegak lurus tersebut disebut hipotenusa, merupakan sisi terpanjang pada segitiga siku-siku. Sisi-sisi lainnya disebut kaki dari segitiga tersebut (dalam bahasa Inggris *catheti* (jamak: *cathetus*)).

Segitiga siku-siku mematuhi teorema Pythagoras: jumlah kuadrat dari panjang kedua kaki sama dengan kuadrat panjang hipotenusa: $a^2 + b^2 = c^2$, di mana a dan b adalah panjang masing - masing kaki dan c adalah panjang hipotenusa. Demikian pula sebaliknya, jika panjang masing - masing kaki memenuhi persamaan tersebut, maka segitiga tersebut pasti memiliki sudut siku-siku di seberang sisi terpanjangnya.

Segitiga siku-siku istimewa adalah segitiga siku-siku yang memiliki sifat yang membuat perhitungannya menjadi lebih mudah. Salah satunya adalah segitiga siku-siku 3–4–5, di mana $3^2 + 4^2 = 5^2$. Dalam situasi ini 3, 4, dan 5 adalah tripel Pythagorean. Yang lainnya adalah segitiga sama kaki yang memiliki dua sudut lain sebesar 45° .

Beberapa fakta mengenai segitiga siku-siku:

- Jumlah sudut-sudut yang bukan siku-siku pasti 90° .
- Jika kaki-kaki segitiga siku-siku sama panjang, maka sudut yang berseberangan dengan kaki-kaki tersebut akan sama besarnya. Dengan demikian karena sifat

pertama di atas, maka sudutnya pasti 45° . Dengan teorema Pythagoras, panjang hipotenusa adalah panjang kaki dikali $\sqrt{2}$.

- Di segitiga siku-siku jika besar sudut lainnya 30° dan 60° , maka panjang hipotenusa dua kali panjang kaki terpendek dan panjang kaki yang lebih panjang sama dengan panjang kaki yang lebih pendek dikali $\sqrt{3}$.

2.12 SD Card

SD Card merupakan storage yg dulu bisaa digunakan pada HP, kamera digital, namun sekarang mulai digunakan untuk menyimpan data pada komputer, beriringan dengan *flashdisk*. Pada tahun 2001 SanDisk Corporation, Matsushita (Panasonic) dan Toshiba memperkenalkan SD Memory card atau Secure Digital. Satu-satunya perbedaan adalah bahwa *memory card* SD sedikit lebih tebal dan memiliki *write protection switch*. Multi Media Card, MMC, *memory card*, kartu standar SD dan memiliki faktor bentuk hampir sama seukuran perangkat. Karena kartu MMC yang lebih tipis dari SD Memory card dapat digunakan di semua slot SD Memory card namun tidak sebaliknya. Kartu Secure Digital SD adalah *memory card flash* ultra kecil yang dirancang untuk menyediakan memori berkapasitas tinggi dalam ukuran yang kecil. *Portable device* seperti kamera digital, camcorder video digital, notebook, audio *player* dan ponsel semuanya membutuhkan SD Card. Umumnya ukuran SD Card ukuran 32 x 24 x 2,1 mm dan berat sekitar 2gram. Tersedia dalam beragam kapasitas mulai dari 16 Megabyte sampai 1 Gigabyte. Saat ini *memory card* yang paling sering digunakan adalah SD Card, digunakan pada perangkat elektronik seperti kamera digital, PDA, dan lain lain. Kini sebagian besar perangkat elektronik memiliki kartu memory yang dapat digunakan untuk lebih dari *memory card flash*. Secure Digital In Out disingkat SDIO, adalah nama umum yang

diberikan kepada berbagai modul ekspansi yang dapat ditemukan dalam faktor bentuk Memory SD Card itu. Hal ini juga dapat digunakan untuk fungsi-fungsi lainnya seperti *adapter* Bluetooth, penerima GPS, kamera digital, TV tuner, dan lain lain hanya dengan menyisipkan SD ke dalam *slot*. Dalam perkembangannya, memori jenis SD ini telah memiliki 3 kelompok varian yaitu :

- **SD memory card.** Jenis memori yang paling populer ini memiliki kapasitas mulai dari 8 MB hingga yang tertinggi 4 GB, dengan format FAT16. Memori berukuran 24 x 34 mm dan ketebalan 2 mm ini dinamai *secure* (aman) karena telah memiliki *Content Protection for Recordable Media* (CPRM) untuk mencegah pembajakan media dan adanya *Write-Protect tab* yang mencegah penghapusan isi memori secara tidak sengaja.
- **MiniSD dan MicroSD memory card.** Dengan ukuran memori miniSD yang kecilnya hanya 20 x 21,5 mm dan microSD dengan ukuran 11 x 15 mm, memori jenis ini memang cocok untuk dipasang di ponsel atau MP3 *player*. Bagi yang ingin memakai memori miniSD / microSD pada perangkat yang memiliki *slot* SD, maka diperlukan sebuah *adapter* (bisaanya disertakan saat membeli miniSD / microSD).
- **SDHC memory card.** SDHC (SD *High Capacity*) adalah pengembangan selanjutnya dari SD *card* yang meski tidak merubah bentuk dan desain, namun kini kecepatan dan kinerjanya telah ditingkatkan dengan memakai format FAT32. MicroSD adalah salah satu jenis format memory *flash* yang diluncurkan tahun 2005 , sedangkan Micro SDHC diluncurkan tahun 2007. SDHC *card* memiliki kapasitas mulai dari 4 GB hingga 32 GB. Untuk urusan kompatibilitas, satu hal yang perlu diingat bahwa bila perangkat kita telah mampu mendukung memori jenis SDHC, maka ia akan tetap kompatibel dengan memori SD biasa. Namun sebaliknya, jangan

gunakan memori SDHC pada perangkat lama yang hanya mampu mengenali memori SD card.

SDCard dapat dilihat pada Gambar 2.10 dibawah ini.



Gambar 2.10 SD Card

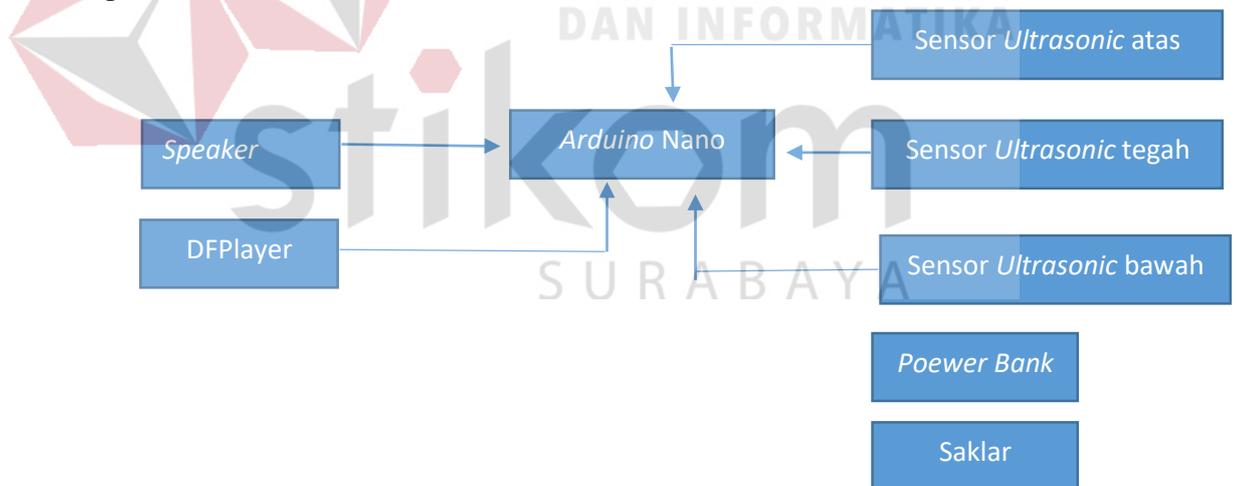
BAB III

METODE PENELITIAN

Metode penelitian yang digunakan dalam pengerjaan Tugas Akhir ini adalah studi literatur berupa data- data dari masing-masing komponen, perancangan perangkat keras dan pembuatan program untuk membuat ikat pinggang pintar. Pada perancangan *system* terdapat sensor *ultrasonic* yang digunakan untuk mengetahui jarak pada masing-masing halangan. Juga terdapat DFPlayer untuk *output* pada Arduino Nano. DFPlayer digunakan untuk mengatur suara yang keluar dari micro SD yang digunakan.

3.1 Perancangan System

Secara umum gambar Blok Diagram pada rancangan perangkat keras dapat dilihat pada Gambar 3.1 dibawah ini

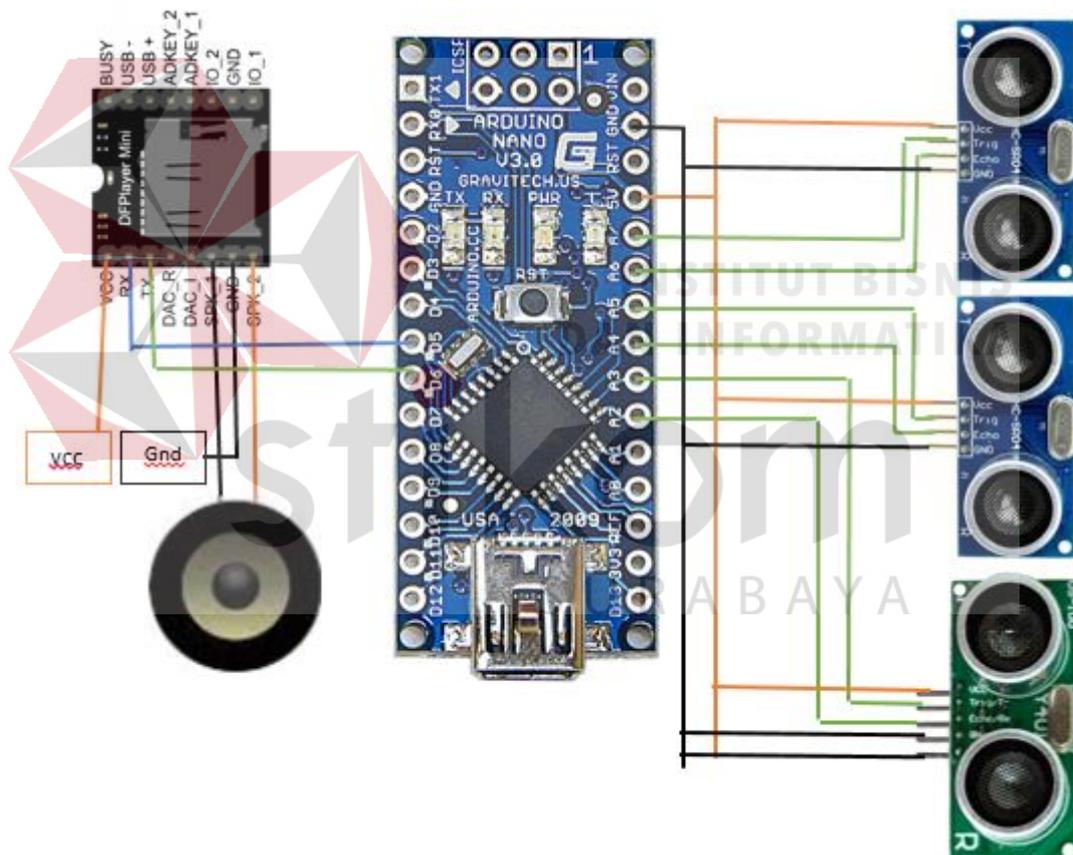


Gambar 3.1 Blok Diagram

Pada Gambar 3.1 terdapat Blok diagram dengan pusat kontrol yaitu Arduino Nano. Arduino Nano tersebut memiliki *input* dan *output* dimana *input* berupa sensor *ultrasonic* atas, sensor *ultrasonic* tengah, sensor *ultrasonic* bawah dan saklar. Untuk *output* terdapat DFPlayer dan *speaker*. *Input* sensor digunakan untuk mengambil

data jarak pada masing masing sensor. Saklar tersebut digunakan untuk tombol *reset*. Untuk *output* DFPlayer dan *speaker* digunakan untuk memberikan informasi berupa suara. Suara tersebut terdapat 3 macam suara dimana suara yang pertama adalah “ada halangan diatas”, yang kedua adalah “ada halangan dibawah” ,yang ketiga adalah “ada halangan didepan”. Masing masing suara tersebut disimpan pada Memory *SD card* yang terdapat pada DFPlayer.

3.2 Perancangan Perangkat Keras

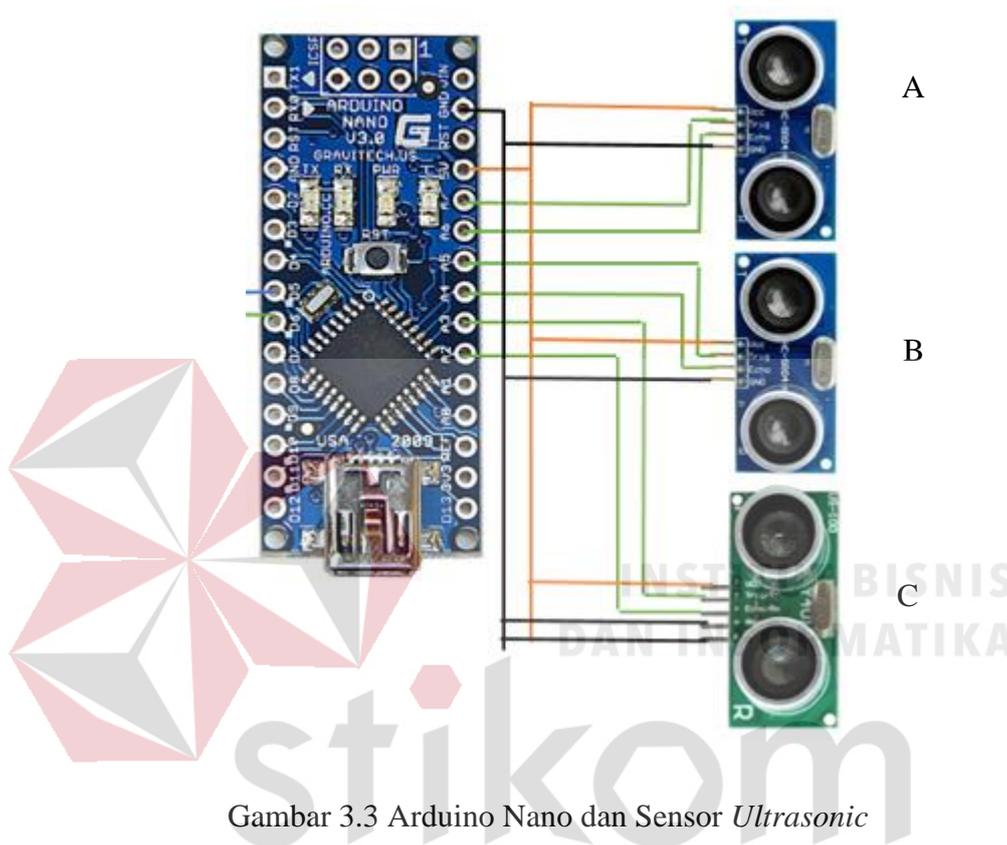


Gambar 3.2 Rangkaian Otomatisasi System

Pada Gambar 3.2 adalah rangkaian *system* yang mana akan dipasang pada alat. Yang terdiri dari DFPlayer, Arduino Nano, *Ultrasonic* US100, *Ultrasonic* Hc-SR04 dan *speaker*. System tersebut dapat digunakan untuk system otomatisasi alat

bantu tunanetra pada penulis. Untuk penjelasan secara jelas dapat dilihat pada sub bab berikut ini.

3.2.1 Komunikasi Arduino Nano dan Sensor *Ultrasonic*



Gambar 3.3 Arduino Nano dan Sensor *Ultrasonic*

Pada Gambar 3.3 terdapat sensor *ultrasonic* A, B dan C dimana sensor *ultrasonic* A adalah sensor *ultrasonic* atas, *ultrasonic* B adalah sensor depan dan *ultrasonic* C adalah sensor *ultrasonic* bawah. Sensor *ultrasonic* A dan B adalah sensor *ultrasonic* HCSR-04, sedangkan sensor *ultrasonic* C adalah sensor *ultrasonic* US100. Sensor A dihubungkan dengan cara menggabungkan VCC dan *Ground* pada Arduino Nano dimana VCC dilambangkan dengan warna orange dan *Ground* dilambangkan dengan warna hitam pada Gambar 3.4 diatas. Setelah itu terdapat *echo* dan *trigger* yang dimana masing-masing dihubungkan pada *port* A7 dan A6 pada Arduino Nano

Sedangkan untuk sensor B prosesnya sama dengan proses pada sensor A yaitu menggabungkan VCC dan *Ground* kemudian menggabungkan *port A5* pada *trigger* dan *port A4* pada *echo*. Selanjutnya adalah sensor C dimana sensor tersebut adalah US100 dimana memiliki 2 *Ground* dan 1 VCC. *Ground* dan VCC pada sensor dihubungkan dengan Arduino Nano sedangkan *echo* dan *trigger* dihubungkan pada *port A2* dan *A3* pada Arduino Nano. Perangkat yang dibutuhkan untuk mengakses sensor tersebut adalah :

1. Laptop
2. Arduino Nano
3. Sensor *Ultrasonic*
4. Kabel *Jumper*
5. Software Arduino IDE
6. Kabel USB *Downloader*

Perangkat tersebut adalah software Arduino IDE yang digunakan untuk memprogram sensor. Dan perangkat pendukung adalah laptop. Untuk elektronika adalah arduino Nano, sensor *ultrasonic* dan kabel *jumper*. Untuk program mengakses sensor A dan B dapat dilihat pada Gambar 3.4 dibawah ini.

```

#include <NewPing.h>

#define TRIGGER_PIN  A7 // Arduino pin tied to trigger pin on the ultraso
#define ECHO_PIN     A6 // Arduino pin tied to echo pin on the ultrasonic
#define MAX_DISTANCE 200 // Maximum distance we want to ping for (in centi

NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE); // NewPing setup of pi

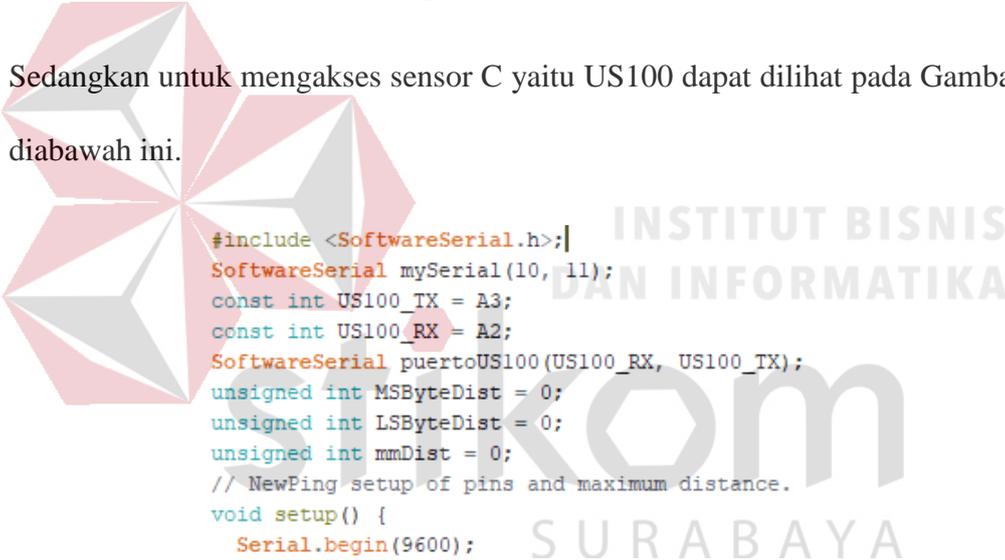
void setup() {
  Serial.begin(115200); // Open serial monitor at 115200 baud to see ping
}

void loop() {
  delay(50); // Wait 50ms between pings (about 20 pin
  unsigned int uS = sonar.ping(); // Send ping, get ping time in microseco
  Serial.print("Ping: ");
  Serial.print(uS / US_ROUNDTRIP_CM); // Convert ping time to distance in
  Serial.println("cm");
}

```

Gambar 3.4 Program Sensor *Ultrasonic* HC-SR04

Sedangkan untuk mengakses sensor C yaitu US100 dapat dilihat pada Gambar 3.5 di bawah ini.



```

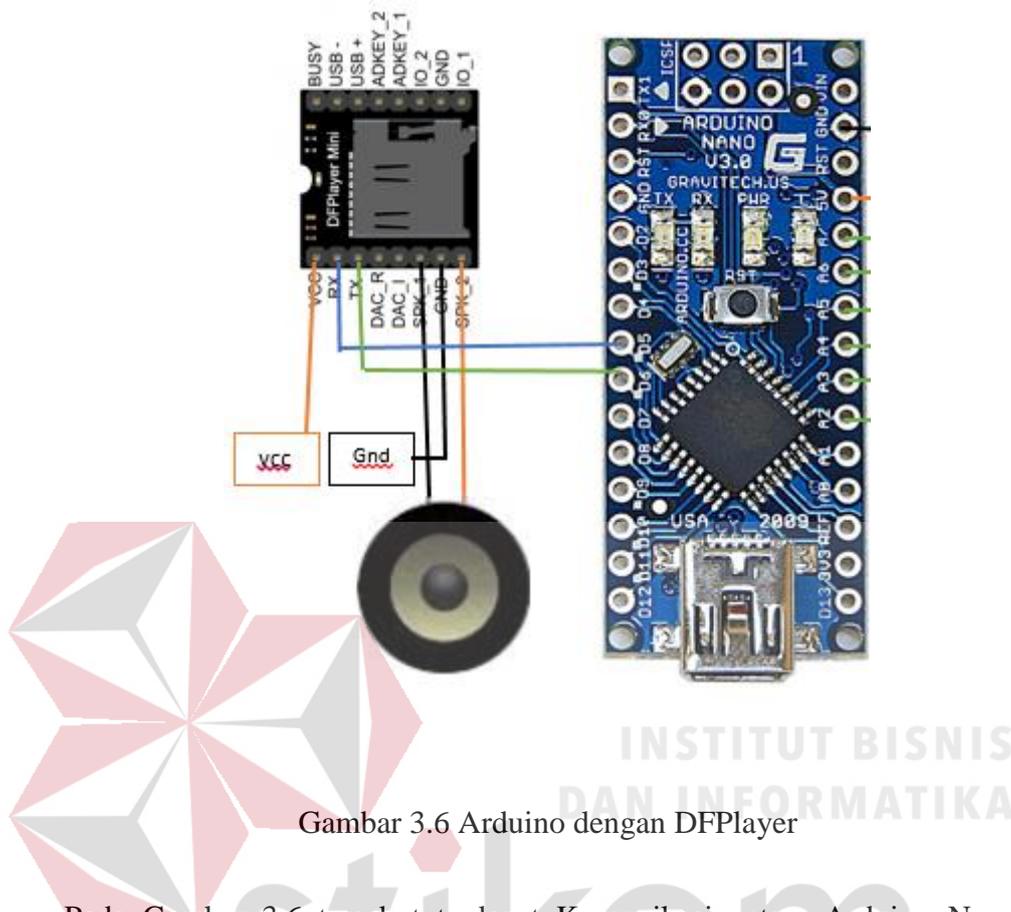
#include <SoftwareSerial.h>;
SoftwareSerial mySerial(10, 11);
const int US100_TX = A3;
const int US100_RX = A2;
SoftwareSerial puertoUS100(US100_RX, US100_TX);
unsigned int MSByteDist = 0;
unsigned int LSByteDist = 0;
unsigned int mmDist = 0;
// NewPing setup of pins and maximum distance.
void setup() {
  Serial.begin(9600);
  delay(10);
}

void loop() {
  puertoUS100.begin(9600);
  delay(50); // Wait 50ms between
  puertoUS100.flush(); // limpia el buffer del puerto
  puertoUS100.write(0x55); // orden de medición de di:
  if(puertoUS100.available() >= 2) // comprueba la :
  {
    MSByteDist = puertoUS100.read(); // lectura d
    delay(200);
    LSByteDist = puertoUS100.read();
    delay(200);
    mmDist = MSByteDist * 256 + LSByteDist; // d:
    if((mmDist > 1) && (mmDist < 10000)) // compr
    {

```

Gambar 3.5 Program US100

3.2.1 Komunikasi Arduino Nano dan DFPlayer



Gambar 3.6 Arduino dengan DFPlayer

Pada Gambar 3.6 tersebut terdapat Komunikasi antara Arduino Nano dan DFPlayer dimana dalam DFPlayer, *port* yang digunakan adalah *port* RX, TX, SPK1, SPK2, VCC dan *Ground*. Untuk *Ground* tetap terhubung dengan Arduino Nano sedangkan VCC tidak dihubungkan pada Arduino Nano tetapi dihubungkan pada sumber yang lain seperti *power bank*. Untuk RX dihubungkan pada D5 yang dilambangkan pada kabel biru. Untuk TX dihubungkan pada D6 yang dilambangkan pada kabel hijau. Untuk *Speaker* dihubungkan pada *port* SPK1 dan SPK2. Sedangkan untuk mengisi suara pada DFPlayer dibutuhkan MicroSD yang terisi *file* mp3. Untuk menjalankan program diperlukan perangkat dibawah ini.

1. Laptop
2. MicroSD

3. DFPlayer
4. *Speaker*
5. Arduino Nano
6. Kabel Jumper
7. Kabel USB *Downloader*
8. Software Arduino IDE

Untuk Program dapat dilihat pada Gambar 3.8 dibawah ini.

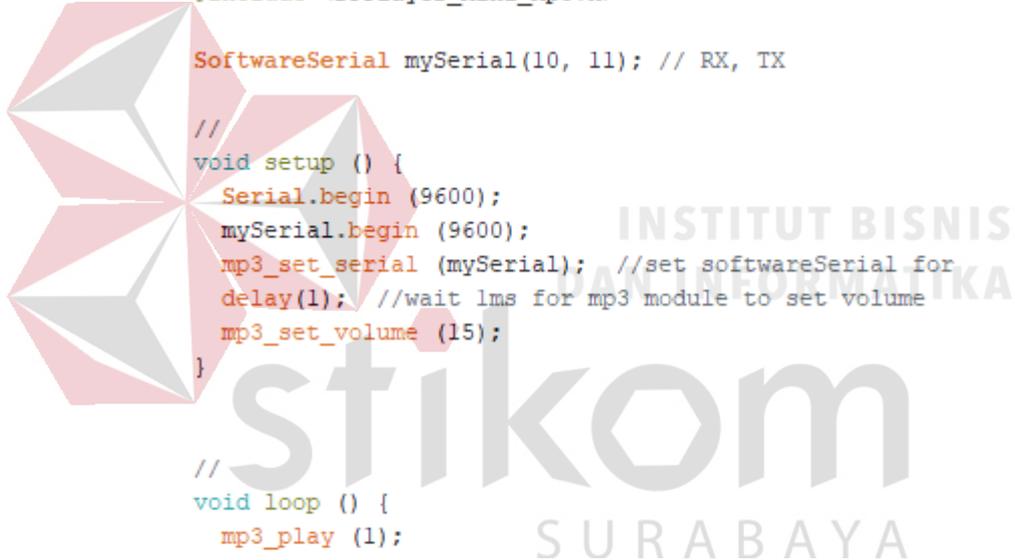
```
#include <SoftwareSerial.h>
#include <DFPlayer_Mini_Mp3.h>

SoftwareSerial mySerial(10, 11); // RX, TX

//
void setup () {
  Serial.begin (9600);
  mySerial.begin (9600);
  mp3_set_serial (mySerial); //set softwareSerial for
  delay(1); //wait lms for mp3 module to set volume
  mp3_set_volume (15);
}

//
void loop () {
  mp3_play (1);
  delay (6000);
  mp3_next ();
  delay (6000);
  mp3_prev ();
  delay (6000);
  mp3_play (4);
  delay (6000);
}

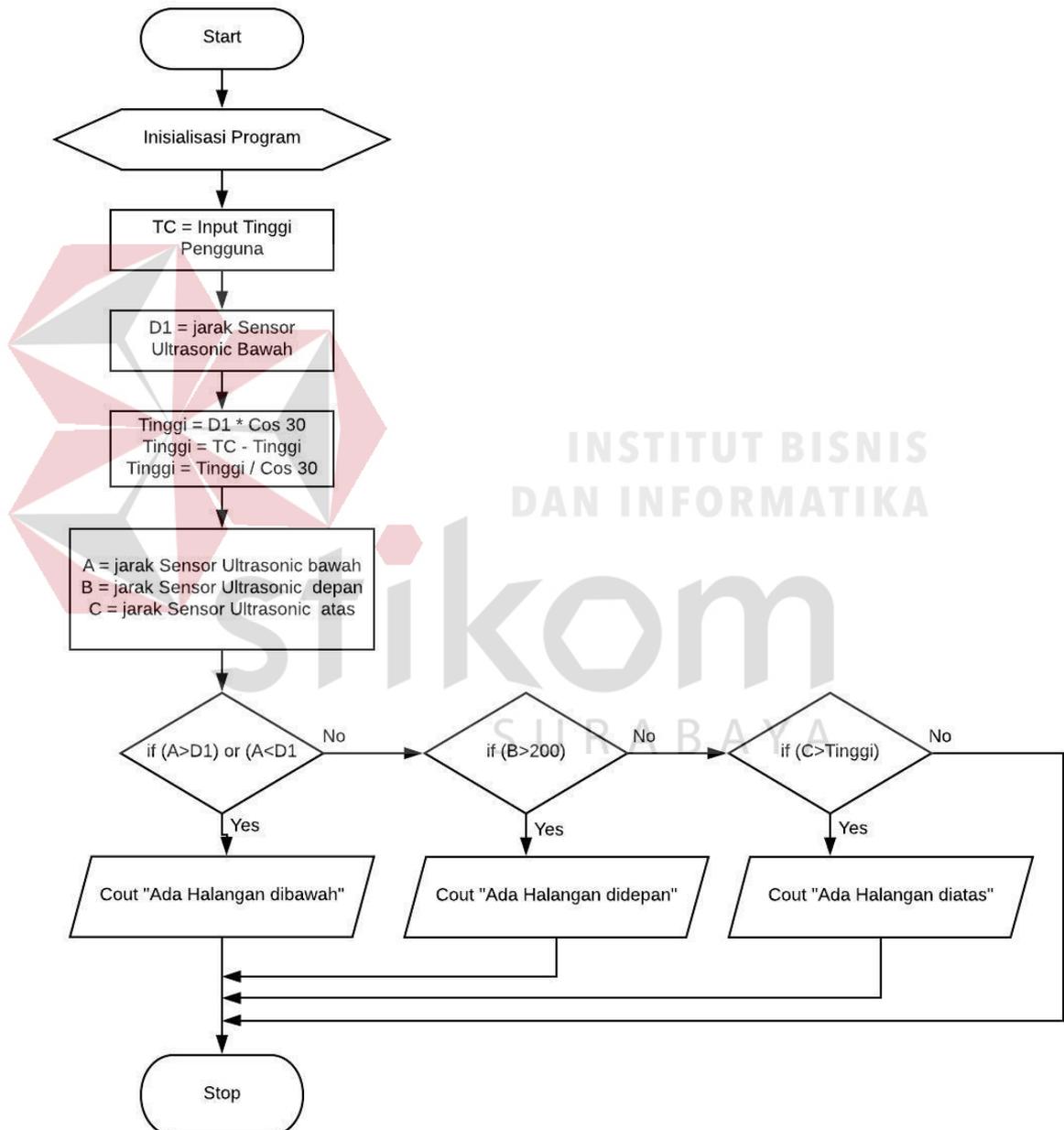

```



Gambar 3.7 Program DFPlayer

3.3 Perancangan Perangkat Lunak

Untuk dapat menuju pada *system* ikat pinggan pintar diperlukan beberapa tahapan seperti yang terlihat pada Gambar 3.8 dibawah ini.



Gambar 3.8 *Flowchart System*

Proses *Flowchart* pada Gambar 3.8 adalah sebagai berikut :

- a. *Start* digunakan untuk memulai program
- b. Inisialisasi Program digunakan untuk membuat *library* dan *variable* yang digunakan.
- c. D1 adalah pengambilan jarak pada sensor *ultrasonic* bawah untuk batasan pada sensor bawah.
- d. Selanjutnya adalah melalui proses perhitungan rumus tersebut akan didapatkan ketinggian aman pada sensor atas. Dengan ketentuan ketinggian orang harus di tulis pada program terlebih dahulu.
- e. Cara perhitungannya adalah D1 (Jarak sensor bawah) dikali dengan $\cos 30$. Selanjutnya hasil dari perkalian tersebut akan dikurangkan dengan tinggi pengguna. Hasilnya akan dibagi dengan $\cos 30$. Ketinggian aman didapatkan setelah melalui perhitungan tersebut.
- f. A = Sensor *Ultrasonic* atas digunakan untuk mengambil data jarak sensor bawah yang nantinya disalin ke *variable* A.
- g. B = Sensor *Ultrasonic* depan digunakan untuk mengambil data jarak sensor depan yang nantinya disalin ke *variable* B.
- h. C = Sensor *Ultrasonic* belakang digunakan untuk mengambil data jarak sensor belakang yang nantinya disalin ke *variable* C.
- i. Selanjutnya adalah masuk pada proses percabangan untuk Hasil jarak sensor dibawah dimana sensor tersebut akan dibandingkan dengan hasil pengambilan pertama pada sensor yang disimpan pada D1. Jika $A > D1$ or $A < D1$ maka akan aktif suara “ ada halangan dibawah”.

- j. Selanjutnya adalah masuk pada proses percabangan untuk Hasil sensor tengah, dimana nilai tersebut jika bernilai lebih kecil dari 200 maka akan mengeluarkan suara “Ada halangan didepan”. Apabila halangan tersebut lebih besar dari 200 maka masuk pada percabangan berikutnya.
- k. Selanjutnya adalah masuk pada proses percabangan untuk Hasil sensor atas dimana nilai tersebut jika bernilai lebih kecil dari variable tinggi maka akan mengeluarkan suara “Ada halangan diatas”. Apabila halangan tersebut lebih besar dari 130 maka masuk pada percabangan berikutnya.
- l. *Stop* adalah proses untuk menghentikan program.

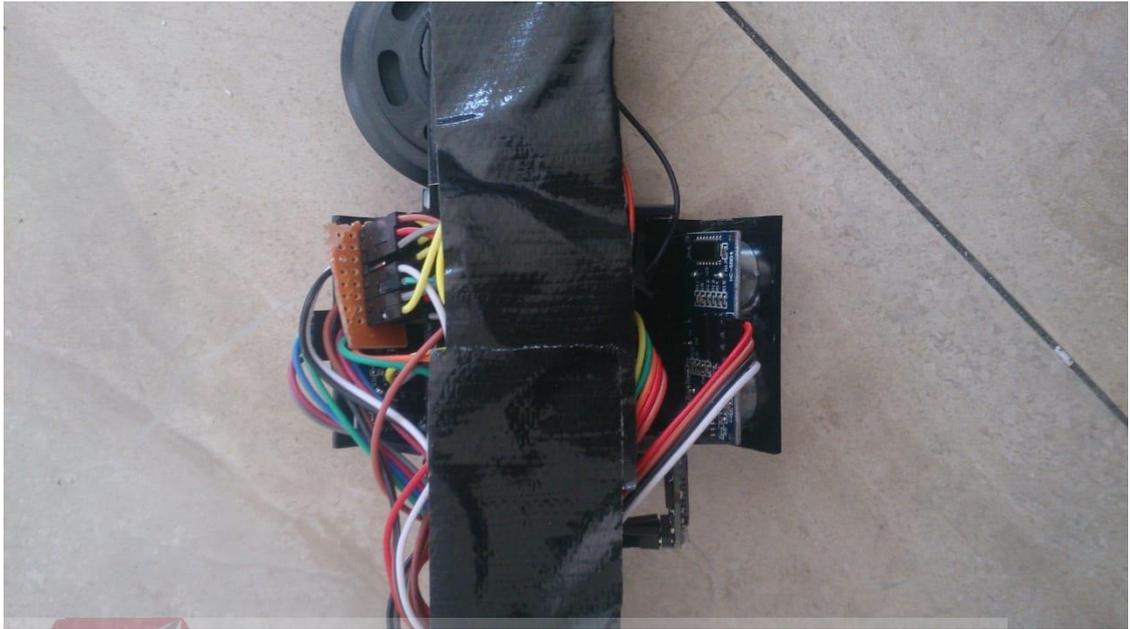
3.4 Model Rancangan

Berikut ini Foto rancangan otomatisasi system pada alat bantu tunanetra.

Foto tersebut dapat dilihat pada Gambar dibawah ini.



Gambar 3.9 Foto Desain *Hardware*



Gambar 3.10 Foto Desain *Hardware* Rangkaian

3.5. Struktur Material

Dalam penelitian ini bahan material yang digunakan sebagai berikut :

Bagian rancang bangun :

1. Akrilik
2. Kabel *Jumper*
3. Ikat Pinggang
4. PCB lubang

Bagian Elektro :

1. Arduino Nano
2. Sensor *Ultrasonic* HC-SR04
3. Sensor *Ultrasonic* Us100
4. *Poewer Bank*

5. DFPlayer

6. *Speaker*

3.6. Pemrograman

Pemrograman dilakukan setelah *hardware* selesai dibuat. Seluruh perangkat keras diuji apakah sudah dapat berjalan dan sesuai yang diinginkan atau tidak. Kemudian memasukkan program bahasa C ke dalam Arduino Nano. Didalam program Arduino Nano terdapat *library*, *Setup* dan *Loop*. Langkah-langkah memasukkan adalah sebagai berikut:

1. Langkah pertama adalah memasukkan *Library* pada *Arduino* Nano, *library* yang dicantumkan adalah *library software* serial, newping, dan DFPlayer. *Library* Serial digunakan untuk memasukkan sensor Us100, sedangkan newping digunakan untuk menggunakan sensor HC-SR04. Program bisa dilihat pada Gambar 3.11 dibawah ini.



```

Hans_ikat_pinggang
// -----
// Example NewPing library sketch that does a ping about 20 tim
// -----
#include <SoftwareSerial.h>;
#include <NewPing.h>
#include <DFPlayer_Mini_Mp3.h>
SoftwareSerial mySerial(10, 11);
const int US100_TX = 5;
const int US100_RX = 4;

SoftwareSerial puertoUS100(US100_RX, US100_TX);
unsigned int MSByteDist = 0;
unsigned int LSByteDist = 0;
unsigned int mmDist = 0;

```

Gambar 3.11 Inisialisasi program

2. Selanjutnya adalah memasukkan *variable* pada program, dimana *variable* dibutuhkan untuk membantu menyimpan data pada program. Program bisa dilihat pada Gambar 3.12 dibawah ini.

```
#define TRIGGER_PIN 8 // Arduino pin tied to trigger pin on
#define ECHO_PIN 9 // Arduino pin tied to echo pin on the
#define MAX_DISTANCE 200 // Maximum distance we want to ping f
/////
#define TRIGGER_PIN1 6
#define ECHO_PIN1 7
#define MAX_DISTANCE1 200
NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE);
NewPing sonar1(TRIGGER_PIN1, ECHO_PIN1, MAX_DISTANCE1);
// NewPing setup of pins and maximum distance.
```

Gambar 3.12 *Trigger* dan *Echo* pada Sensor *ultrasonic*

3. Selanjutnya adalah masuk pada menu *Setup* dimana menu tersebut digunakan untuk jalan awal pada program. Menu ini juga digunakan untuk memasukkan nilai *variable local* yang digunakan. Program bisa dilihat pada Gambar 3.13 dibawah ini.

```
void setup() {
  Serial.begin(9600); // Open serial monitor at 115200 baud
  // puertoUS100.begin(9600);
  // mySerial.begin (9600);
  mp3_set_serial (mySerial);
  delay(10);

  mp3_set_volume (50);
}
```

Gambar 3.13 Deklarasi Serial

4. Selanjutnya adalah masuk pada menu *Looping* atau perulangan, dimana menu ini adalah jalannya program terus menerus ketika program sudah *start*. Didalam program ini terdapat perintah untuk mengambil nilai data

pada sensor dan memasukkan hasil pada *output*. Program bisa dilihat pada

Gambar 3.14 dibawah ini.

```

void loop() {
  puertoUS100.begin(9600);
  delay(50); // Wait 50ms between pings (about 20 pings/sec). 29ms should be the shortest delay bet
  unsigned int uS = sonar.ping();
  unsigned int uS1 = sonar1.ping();
  uS = uS / US_ROUNDTRIP_CM;
  uS1 = uS1 / US_ROUNDTRIP_CM;
  puertoUS100.flush(); // limpia el buffer del puerto serie
  puertoUS100.write(0x55); // orden de medición de distancia

  if(puertoUS100.available() >= 2) // comprueba la recepción de 2 bytes
  {
    MSByteDist = puertoUS100.read(); // lectura de ambos bytes
    LSByteDist = puertoUS100.read();
    mmDist = MSByteDist * 256 + LSByteDist; // distancia
    if((mmDist > 1) && (mmDist < 10000)) // comprobación de la distancia dentro de rango
    {
      puertoUS100.end();
      mySerial.begin (9600);
      Serial.print("Distancia: ");
      Serial.print(mmDist);
      Serial.print(" mm , ");
      Serial.print("Ping: ");
      Serial.print(uS); // Convert ping time to distance in cm and print result (0 = outside set distance range)
      Serial.print("cm ");
      Serial.print("Ping1: ");
      Serial.print(uS1); // Convert ping time to distance in cm and print result (0 = outside set distance range)
      Serial.println("cm");

      if (mmDist < 2000)
      {
        mp3_play (1);
        delay (1000);
      }
      if (uS < 130 && uS > 0)
      {
        mp3_play (2);
        delay (1000);
      }
      if (uS1 < 130 && uS1 > 0)
      {
        mp3_play (3);
        delay (1000);
      }
      mySerial.end();
    }
  }
  delay(500);
  // Send ping, get ping time in microseconds (uS).

```

Gambar 3.14 Program Percabangan untuk informasi DFPlayer

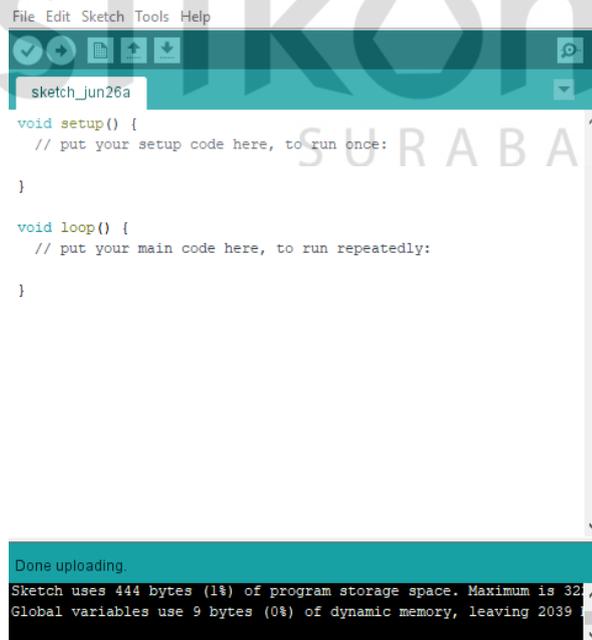
BAB IV

HASIL DAN PEMBAHASAN

Dalam bab ini penulis akan menguraikan dan menjelaskan hasil analisis pengujian dari hasil penelitian tugas akhir ini yang telah dilakukan, pengujian dilakukan dalam beberapa bagian yang disusun dalam urutan dari yang sederhana menuju *system* yang lengkap. Pengujian dilakukan meliputi pengujian masing-masing perangkat keras yang kemudian pengujian keseluruhan.

4.1 Hasil Pengujian Arduino Nano

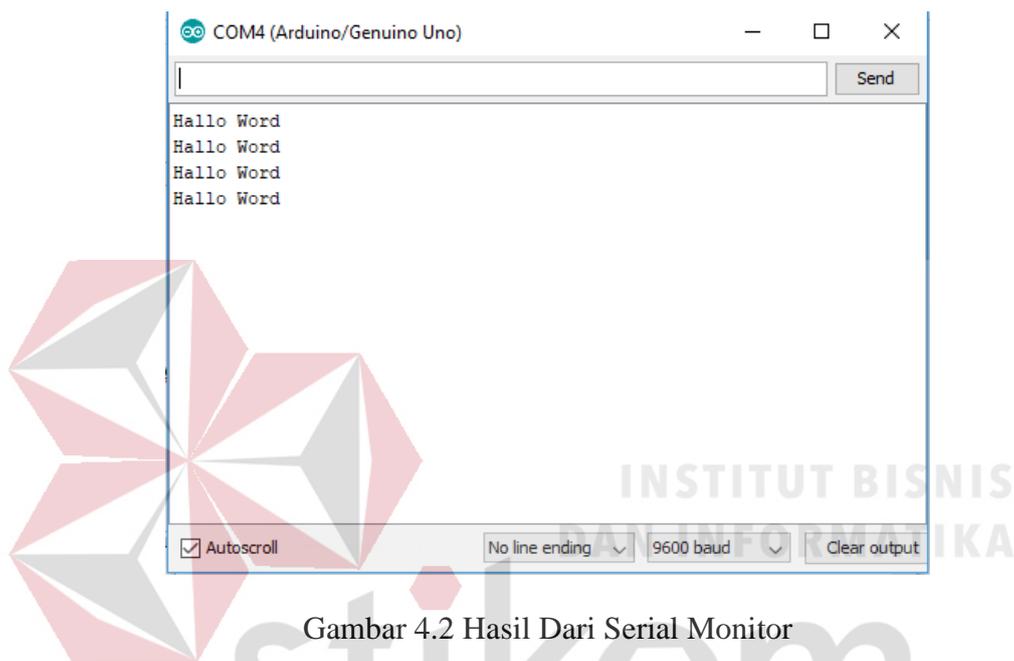
Pengujian program pada Arduino Nano dengan *Software* Arduino IDE dapat dilihat pada Gambar 4.1 bertuliskan “Done Uploading”, yang menekan bahwa program yang ditulis telah benar dan berhasil di-*upload* pada Arduino Nano.



```
File Edit Sketch Tools Help
sketch_jun26a
void setup() {
  // put your setup code here, to run once:
}
void loop() {
  // put your main code here, to run repeatedly:
}
Done uploading.
Sketch uses 444 bytes (1%) of program storage space. Maximum is 32256 bytes.
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes free.
```

Gambar 4.1 *Upload* Berhasil Pada Arduino IDE

Program yang dimasukkan kedalam Arduino Nano merupakan program untuk mengirimkan data menggunakan *port* serial. Proses pengiriman pada Arduino Nano harus terhubung dahulu dengan USB PC agar dapat menerima data yang dikirim melalui menu serial monitor pada *software* Arduino IDE. Hasil dari serial monitor dapat dilihat pada Gambar 4.2.



Gambar 4.2 Hasil Dari Serial Monitor

Pada Gambar 4.2, menunjukkan bahwa data yang dikirim pada serial monitor sesuai dengan program perintah yang dibuat dan di *upload* pada Arduino Nano. Dengan begitu Arduino Nano ini dapat bekerja dengan baik dan dapat digunakan dalam pembuatan *system*.

4.2 Hasil Pengujian Sensor *Ultrasonic* HC-SR04 atas

Pengujian sensor *ultrasonic* dilakukan pada Arduino Nano dengan cara memasukkan program yang telah dibuat dan akan mengeluarkan hasil pada serial.. Program dapat dilihat pada Gambar 4.3 di bawah ini.

```

File Edit Sketch Tools Help
NewPingExample
// -----
// Example NewPing library sketch that does a ping about 20 times per second.
// -----

#include <NewPing.h>

#define TRIGGER_PIN 12 // Arduino pin tied to trigger pin on the ultrasonic sensor.
#define ECHO_PIN 11 // Arduino pin tied to echo pin on the ultrasonic sensor.
#define MAX_DISTANCE 200 // Maximum distance we want to ping for (in centimeters). Maximum sensor distance is rated at 400-500cm.

NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE); // NewPing setup of pins and maximum distance.

void setup() {
  Serial.begin(115200); // Open serial monitor at 115200 baud to see ping results.
}

void loop() {
  delay(50); // Wait 50ms between pings (about 20 pings/sec). 29ms should be the shortest delay between pings.
  unsigned int uS = sonar.ping(); // Send ping, get ping time in microseconds (uS).
  Serial.print("Ping: ");
  Serial.print(uS / US_ROUNDTRIP_CM); // Convert ping time to distance in cm and print result (0 = outside set distance range)
  Serial.println("cm");
}

```

Gambar 4.3 Program sensor *ultrasonic* atas

Hasil sensor dapat dilihat pada Tabel 4.1 dibawah ini.

Tabel 4.1 Hasil sensor atas

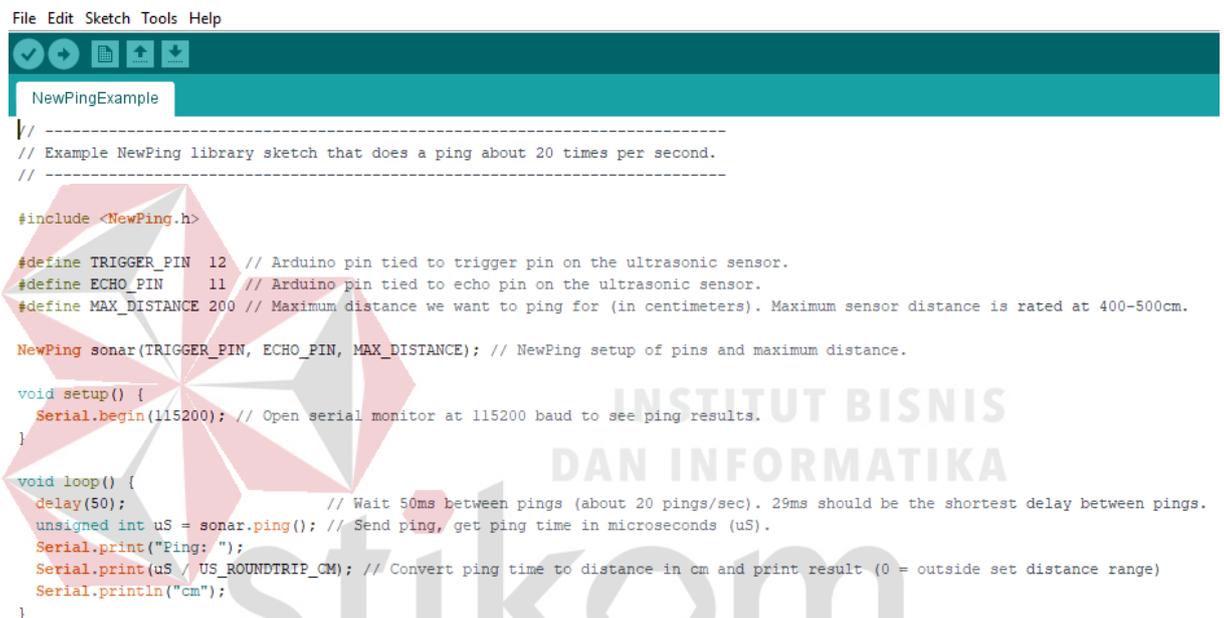
Detik	Jarak Asli (cm)	Hasil Sensor (cm)	Error (%)
1	10	11	10
2	20	21	5
3	30	31	3.3
4	40	41	2.5
5	50	51	2
6	60	61	1.7
7	70	71	1.4
8	80	81	1.3
9	90	91	1.1
10	100	101	1
11	110	111	0.9
13	130	132	1.4
14	140	142	1.3
15	150	152	1.3
16	160	162	1.2
17	170	172	1.1

Dari table tersebut dapat disimpulkan bahwa rata-rata error pada sensor atas adalah 2.28%

4.3 Hasil Pengujian Sensor *Ultrasonic* HC-SR04 Depan

Pengujian sensor *ultrasonic* dilakukan pada Arduino Nano dengan cara memasukkan program yang telah dibuat dan akan mengeluarkan hasil pada serial.

Program dapat dilihat pada Gambar 4.4 di bawah ini.



```

File Edit Sketch Tools Help
NewPingExample
// -----
// Example NewPing library sketch that does a ping about 20 times per second.
// -----

#include <NewPing.h>

#define TRIGGER_PIN 12 // Arduino pin tied to trigger pin on the ultrasonic sensor.
#define ECHO_PIN 11 // Arduino pin tied to echo pin on the ultrasonic sensor.
#define MAX_DISTANCE 200 // Maximum distance we want to ping for (in centimeters). Maximum sensor distance is rated at 400-500cm.

NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE); // NewPing setup of pins and maximum distance.

void setup() {
  Serial.begin(115200); // Open serial monitor at 115200 baud to see ping results.
}

void loop() {
  delay(50); // Wait 50ms between pings (about 20 pings/sec). 29ms should be the shortest delay between pings.
  unsigned int uS = sonar.ping(); // Send ping, get ping time in microseconds (uS).
  Serial.print("Ping: ");
  Serial.print(uS / US_ROUNDTRIP_CM); // Convert ping time to distance in cm and print result (0 = outside set distance range)
  Serial.println("cm");
}

```

Gambar 4.4 Program *Ultrasonic* depan

Hasil sensor dapat dilihat pada Tabel 4.2 di bawah ini.

Tabel 4.2 Hasil Sensor Depan

Detik	Jarak Asli (cm)	Hasil Sensor (cm)	Error (%)
1	10	11	10
2	20	21	5
3	30	31	3.3
4	40	41	2.5
5	50	51	2
6	60	61	1.7
7	70	71	1.4

Detik	Jarak Asli (cm)	Hasil Sensor (cm)	Error (%)
8	80	81	1.3
9	90	91	1.1
10	100	101	1
11	110	111	0.9
13	130	132	1.4
14	140	142	1.3
15	150	152	1.3
16	160	162	1.2
17	170	172	1.1

Dari table tersebut dapat disimpulkan bahwa rata-rata error pada sensor Depan adalah 2.28%.

4.4 Hasil Pengujian Sensor *Ultrasonic* Bawah

Pengujian sensor *ultrasonic* dilakukan pada Arduino Nano dengan cara memasukkan program yang telah dibuat dan akan mengeluarkan hasil pada serial. Program dapat dilihat pada Gambar 4.5 di bawah ini.

```

puertoUS100.flush(); // limpia el buffer del puerto serie
puertoUS100.write(0x55); // orden de medición de distancia

if(puertoUS100.available() >= 2) // comprueba la recepción de 2 bytes
{
  MSByteDist = puertoUS100.read(); // lectura de ambos bytes
  LSByteDist = puertoUS100.read();
  mmDist = MSByteDist * 256 + LSByteDist; // distancia
  if((mmDist > 1) && (mmDist < 10000)) // comprobación de la distancia dentro de rango
  {
    puertoUS100.end();
    mySerial.begin (9600);
    Serial.print("Distancia: ");
    Serial.print(mmDist);
    Serial.print(" mm , ");
    Serial.print("Ping: ");
    Serial.print(uS); // Convert ping time to distance in cm and print result (0 = outside set distance range)
    Serial.print("cm ");
    Serial.print("Pingl: ");
    Serial.print(uS1); // Convert ping time to distance in cm and print result (0 = outside set distance range)
    Serial.println("cm");
  }
}

```

Gambar 4.5 Program *Ultrasonic* depan

Hasil sensor dapat dilihat pada Tabel 4.3 dibawah ini.

Tabel 4.3 Hasil Sensor Bawah

Detik	Jarak Asli (cm)	Hasil Sensor (cm)	Error (%)
1	10	11	10
2	20	21	5
3	30	31	3.3
4	40	41	2.5
5	50	51	2
6	60	61	1.7
7	70	71	1.4
8	80	81	1.3
9	90	91	1.1
10	100	101	1
11	110	111	0.9
12	120	121	1.4
13	130	132	1.4
14	140	142	1.3
15	150	152	1.3
16	160	162	1.2
17	170	172	1.1

Dari table tersebut dapat disimpulkan bahwa rata-rata error pada sensor Bawah adalah 2.31%

4.5 Hasil Pengujian DFPlayer

Pengujian DFPlayer dilakukan pada Arduino Nano dengan cara memasukkan program yang telah dibuat dan akan mengeluarkan hasil pada serial.

Program dapat dilihat pada Gambar 4.6 dibawah ini.

```

File Edit Sketch Tools Help
dfplayer
#include <SoftwareSerial.h>
#include <DFPlayer_Mini_Mp3.h>

SoftwareSerial mySerial(10, 11);

void setup () {
  pinMode(8, OUTPUT);
  digitalWrite(8, HIGH);
  Serial.begin (9600);
  mySerial.begin (9600);
  mp3_set_serial (mySerial);
  delay(10);

  mp3_set_volume (50);
}

void loop () {
  mp3_play (1);
  delay (6000);
}

```

Gambar 4.6 Pengujian DFPlayer

Hasil DFPlayer dapat dilihat pada Tabel 4.4 dibawah ini.

Tabel 4.4 Uji DFPlayer

Input Tombol	Suara yang dihasilkan	Keterangan
1	Ada halangan di atas	Sesuai
2	Ada halangan di bawah	Sesuai
3	Ada halangan di depan	Sesuai

4.6 Hasil Pengujian Rumus

Pada pengujian rumus terdapat perhitungan manual dan perhitungan didalam program. Pada hasil perhitungan manual dan perhitungan program memiliki perbedaan yaitu selisih koma. Dimana perhitungan manual memiliki selisih koma

karena tidak dibulatkan, sedangkan perhitungan hasil program tidak memiliki hasil koma dikarenakan bertipe data integer. Hasil dapat dilihat pada Tabel dibawah ini.

Tabel 4.5 Hasil Perhitungan Tinggi Ikat Pinggang ke Kaki

Percobaan	Tinggi (cm)	Sensor Bawah (cm)	Sudut (°)	Hasil Manual (cm)	Hasil Program (cm)	Error (%)
1	150	62	30	53.94	53	0.94
2	165	67	30	58.29	58	0.29
3	170	72	30	62.64	62	0.64
4	175	77	30	66.99	66	0.99
5	180	82	30	71.34	71	0.34
6	182	87	30	75.69	75	0.69
7	190	92	30	80.04	80	0.04

Tabel 4.6 Hasil Perhitungan Tinggi Ikat Pinggang ke Kepala

Percobaan	Tinggi (cm)	Sensor Bawah (cm)	Sudut (°)	Hasil Manual (cm)	Hasil Program (cm)	Error (%)
1	150	62	30	96.06	96	0.06
2	165	67	30	106.71	106	0.71
3	170	72	30	107.36	107	0.36
4	175	77	30	108.01	108	0.01
5	180	82	30	108.66	108	0.66
6	182	87	30	106.31	106	0.31
7	190	92	30	109.96	109	0.96

Tabel 4.7 Hasil Perhitungan batas Tinggi sensor atas

Percobaan	Tinggi (cm)	Sensor Bawah (cm)	Sudut (°)	Hasil Manual (cm)	Hasil Program (cm)	Error (%)
1	150	62	30	110.4137931	110	0.413793
2	165	67	30	122.6551724	122	0.655172
3	170	72	30	123.4022989	123	0.402299
4	175	77	30	124.1494253	124	0.149425
5	180	82	30	124.8965517	124	0.896552
6	182	87	30	122.1954023	122	0.195402
7	190	92	30	126.3908046	126	0.390805

4.7 Hasil Pengujian Keseluruhan Rangkaian

Pada Program dimasukkan contoh ketinggian pengguna yaitu 170 cm. Dan hasil sensor bawah adalah 83 cm. Setelah melalui perhitungan rumus diperoleh maksimal ketinggian sensor atas adalah 115 cm. Sedangkan batas maksimal sensor depan adalah 150 cm. Sudut pada masing-masing sensor adalah 30 derajat. Hasil data pengujian berdasarkan data tersebut dapat dilihat pada Tabel 4.6 dibawah ini.

Tabel 4.8 Hasil Pengujian Sensor *Ultrasonic* bawah

Percoobaan	Jarak (cm)	Speaker	Keterangan
1	10	“Ada halangan dibawah”	Sesuai
2	20	“Ada halangan dibawah”	Sesuai
3	30	“Ada halangan dibawah”	Sesuai
4	40	“Ada halangan dibawah”	Sesuai
5	50	“Ada halangan dibawah”	Sesuai
6	60	“Ada halangan dibawah”	Sesuai
7	70	“Ada halangan dibawah”	Sesuai
8	80		Sesuai
9	90	“Ada halangan dibawah”	Sesuai
10	100	“Ada halangan dibawah”	Sesuai
11	110	“Ada halangan dibawah”	Sesuai
12	120	“Ada halangan dibawah”	Sesuai
13	130	“Ada halangan dibawah”	Sesuai
14	140	“Ada halangan dibawah”	Sesuai
15	150	“Ada halangan dibawah”	Sesuai
16	160	“Ada halangan dibawah”	Sesuai
18	180	“Ada halangan dibawah”	Sesuai
19	190	“Ada halangan dibawah”	Sesuai
20	200	“Ada halangan dibawah”	Sesuai

Tabel 4.9 Hasil Pengujian Sensor *Ultrasonic* Depan

Percoobaan	Jarak (cm)	Speaker	Keterangan
1	10	“Ada halangan didepan”	Sesuai
2	20	“Ada halangan didepan”	Sesuai
3	30	“Ada halangan didepan”	Sesuai
4	40	“Ada halangan didepan”	Sesuai
5	50	“Ada halangan didepan”	Sesuai
6	60	“Ada halangan didepan”	Sesuai
7	70	“Ada halangan didepan”	Sesuai
8	80	“Ada halangan didepan”	Sesuai
9	90	“Ada halangan didepan”	Sesuai
10	100	“Ada halangan didepan”	Sesuai
11	110	“Ada halangan didepan”	Sesuai
12	120	“Ada halangan didepan”	Sesuai
13	130	“Ada halangan didepan”	Sesuai
14	140	“Ada halangan didepan”	Sesuai
15	150	“ ”	Sesuai
16	160	“ ”	Sesuai
17	170	“ ”	Sesuai
18	180	“ ”	Sesuai
19	190	“ ”	Sesuai
20	200	“ ”	Sesuai

Tabel 4.10 *Ultrasonic Atas*

Percoobaan	Jarak (cm)	Speaker	Keterangan
1	10	“Ada halangan diatas”	Sesuai
2	20	“Ada halangan diatas”	Sesuai
3	30	“Ada halangan diatas”	Sesuai
4	40	“Ada halangan diatas”	Sesuai
5	50	“Ada halangan diatas”	Sesuai
6	60	“Ada halangan diatas”	Sesuai
7	70	“Ada halangan diatas”	Sesuai
8	80	“Ada halangan diatas”	Sesuai
9	90	“Ada halangan diatas”	Sesuai
10	100	“Ada halangan diatas”	Sesuai
11	110	“ ”	Sesuai
12	120	“ ”	Sesuai
13	130	“ ”	Sesuai
14	140	“ ”	Sesuai
15	150	“ ”	Sesuai
16	160	“ ”	Sesuai
17	170	“ ”	Sesuai
18	180	“ ”	Sesuai
19	190	“ ”	Sesuai
20	200	“ ”	Sesuai

Dari pengambilan data diatas telah didapatkan hasil yang sesuai. Alat tersebut harus terdapat input tinggi orang pada program. Batas sensor depan adalah 200 cm, sedangkan batas sensor bawah adalah sesuai dengan ketinggian orang tersebut ketika kondisi awal memakai alat. Untuk batas sensor atas didapatkan dari perhitungan rumus segitiga siku-siku dimana sudut lancip pada segitiga tersebut memiliki nilai 30 derajat. Hasil dari perhitungan tersebut dapat dijadikan batas sensor atas pada sensor. Apabila terdapat halangan atau sensor melewati batas yang sudah ditentukan maka DFPlayer akan aktif untuk memberikan informasi pada pengguna. Informasi tersebut berupa suara yang didapatkan dari *speaker* aktif.

BAB V

PENUTUP

5.1 Kesimpulan

Pada kesimpulan kali ini dapat dilihat pada data dibawah ini. Data tersebut didasari dengan uji coba perangkat :

1. Alat ini harus memasukkan ketinggian pengguna pada program terlebih dahulu.
2. Sudut pada alat adalah 30 derajat dikarenakan jika lebih besar dari 30 derajat dapat membuat sensor tidak stabil. Tidak stabilnya sensor dikarenakan adanya permukaan yang miring.
3. Informasi yang didapatkan oleh pengguna adalah informasi berupa suara yang berasal dari DFPlayer. DFPlayer dihubungkan dengan Speaker dan Arduino Nano sebagai pusat kontrol.
4. Sensor ultrasonic dapat mengetahui jarak di depan, di atas, dan di bawah dengan memiliki rata-rata error 2.8% untuk sensor atas dan depan, serta 2.31% untuk sensor bawah. Data *error* tersebut didapatkan dalam 17 percobaan.

5.2 Saran

Saran yang didapatkan dari penulis berdasarkan hasil penelitian adalah sebagai berikut:

1. Sensor lebih baik menggunakan US100 Semua.
2. Desain lebih baik menggunakan 3D *printing* karena lebih mudah dibentuk.
3. Lebih baik *speaker* ditambahkan dengan resistor untuk memperhalus suara

DAFTAR PUSTAKA

B, S. (2013). *Under/Over Voltage Relay Berbasis Mikrokontroller ATMEGA 328*.

Batam: Politeknik Negri Batam.

D, M. (2010). *Pembuatan Prototipe Alat Elektronika berbasis Arduino Uno* .

Surakarta: Universitas Sebelas Maret.

Handoko, A. P. (2017). *Penjualan Obat Otomatis Berbasis Arduino Uno*.

Yogyakarta : Universitas Sanata Dharma .

Jatmiko A, D. (2012). *Pintar Arduino dalam 30 menit*. Surabaya: Universitas

Wijaya Putra.

Noviriyadi. (2013). *Pengukur Suhu dan Kelembaban (DHT11) Dengan Penampil*

LCD Menggunakan Mikrokontroller ATmega 8535. Jakarta: Universitas

Mercu Buana.

Owen. (2004). *Dasar-dasar Elektronika*. Jakarta: Erlangga.

