

**Penerapan *Analog to Digital Conversion* dan *Digital to Analog Conversion* Pada *Microprocessor* Menggunakan *Arduino* Untuk *Body Awareness* Pada Anak Balita**

**SKRIPSI**

**RHEZA YUDHISTIRA RAMADHANA**

**201401061**



**PROGRAM STUDI S-1 ILMU KOMPUTER  
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI  
UNIVERSITAS SUMATERA UTARA  
2024**

**UNIVERSITAS SUMATERA UTARA**

***Penerapan Analog to Digital Conversion dan Digital to Analog  
Conversion Pada Microprocessor Menggunakan Arduino Untuk Body  
Awareness Pada Anak Balita***

**SKRIPSI**

**Diajukan untuk melengkapi tugas dan memenuhi syarat memperoleh ijazah  
Sarjana Ilmu Komputer**

**RHEZA YUDHISTIRA RAMADHANA**

**201401061**



**PROGRAM STUDI S-1 ILMU KOMPUTER  
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI  
UNIVERSITAS SUMATERA UTARA**

**2024**

**PERSETUJUAN**

Judul : PENERAPAN *ANALOG TO DIGITAL CONVERSION* DAN *DIGITAL TO ANALOG CONVERSION* PADA *MICROPROCESSOR* MENGGUNAKAN ARDUINO UNTUK *BODY AWARENESS* PADA ANAK BALITA

Kategori : SKRIPSI

Nama : RHEZA YUDHISTIRA RAMADHANA

Nomor Induk Mahasiswa : 201401061

Program Studi : SARJANA (S-1) ILMU KOMPUTER

Fakultas : ILMU KOMPUTER DAN TEKNOLOGI INFORMASI UNIVERSITAS SUMATERA UTARA

Komisi Pembimbing :

Dosen Pembimbing I



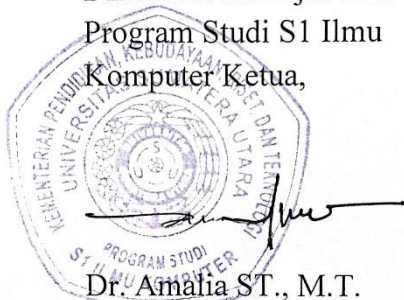
Handrizal S.Si., M.Comp.Sc.  
NIP. 197706132017061001

Dosen Pembimbing II



Hayatunnufus S.Kom, M.Cs.  
NIP. 199207192020012001

Diketahui/disetujui oleh  
Program Studi S1 Ilmu  
Komputer Ketua,



Dr. Amalia ST., M.T.  
NIP. 197812212014042001

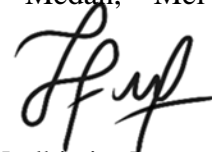
**PERNYATAAN**

**Penerapan *Analog to Digital Conversion* dan *Digital to Analog Conversion* Pada *Microprocessor* Menggunakan Arduino Untuk *Body Awareness* Pada Anak Balita**

**SKRIPSI**

Saya mengakui bahwa skripsi ini adalah hasil penelitian saya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing telah dicantumkan sumbernya.

Medan, Mei 2024



Rheza Yudhistira Ramadhana  
NIM. 201401061

## KATA PENGANTAR

Segala puji dan syukur kami haturkan kepada Allah SWT yang telah melimpahkan berkah, rahmat, dan petunjuk-Nya kepada penulis. Dengan anugerah tersebut, penulis berhasil menyelesaikan penyusunan skripsi ini sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer di Program Studi S1 Ilmu Komputer, Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara. Shalawat dan salam semoga selalu tercurah kepada Rasulullah, Nabi Muhammad Shallallahu 'alaihi Wasallam, yang telah menjadi perantara dalam memperoleh ilmu pengetahuan yang diridhoi oleh Allah.

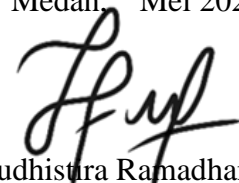
Penulis ingin menyampaikan rasa hormat dan terima kasih yang sebesar-besarnya kepada:

1. Rektor Universitas Sumatera Utara Bapak Prof. Dr. Muryanto Amin S.Sos., M.Si.
2. Dekan Fasilkom-TI USU Ibu Dr. Maya Silvi Lydia, B.Sc., M.Sc.
3. Ketua Prodi S-1 Ilmu Komputer USU Ibu Dr. Amalia, S.T., M.T., yang telah mendukung dan mengawasi kegiatan akademik yang dilakukan penulis.
4. Pembimbing 1 Skripsi Bapak Handrizal S.Si., M.Comp.Sc, atas bimbingan kepada penulis selama masa pembuatan skripsi.
5. Pembimbing 2 Ibu Hayatunnufus S.Kom, M.Cs atas bimbingan kepada penulis selama masa pembuatan skripsi.
6. Pembimbing Akademik Bapak Dr. Mohammad Andri Budiman, S.T., M.Comp.Sc., M.E.M., atas bimbingan dan arahannya ketika melaksanakan pendidikan akademik.
7. Pak Seniman S.Kom, M.Kom atas bantuan dan saran dalam penelitian pengerjaan proyek prototype skripsi.
8. Bapak Ramdhana Feri Martin dan Ibu Nurbaity br. Tarigan selaku orang tua penulis yang selalu mendukung penulis dalam melakukan seluruh kegiatan akademik di dalam kampus maupun diluar kampus. Serta adik-adik penulis yang membantu penulis dalam memenuhi dan membantu penulis selama masa perkuliahan ini.

9. Ira Nurhasanah Pasaribu selaku orang yang selalu mendukung penulis dalam pelaksanaan berbagai kegiatan dan selalu menemani penulis ketika membutuhkan saran.
10. Muhammad Yusuf Daulay selaku teman penulis semenjak kecil hingga saat ini dan selalu membantu penulis ketika dalam kesulitan.
11. Muhammad Syah Al-Fikri selaku teman yang membantu penulis dalam pembuatan proyek skripsi.
12. Nico selaku teman yang selalu hadir ketika penulis membutuhkan tempat untuk tidur ketika penulis membutuhkan tempat tidur.
13. Erick Yudha Pratama Sukku selaku teman yang memberikan pemahaman logis kepada penulis.
14. Ariyan Satya Sikoko selaku teman penulis yang hadir sebagai tempat keluh kesah dan selalu memberikan respon yang logis.
15. Gideon, Galileo, Farhan, dan Kamal dalam menemani penulis sebagai para penjaga lab pak seniman.
16. Nasha Jamil dan Annisa Muflihah selaku teman penulis yang menemani penulis serta sahabat penulis ketika mengalami kondisi gabut.
17. Bang Adhitya Bagaskara selaku senior yang selalu membantu penulis dalam pembuatan proyek dan meminjamkan alat untuk pengetesan proyek.
18. Sastra Harapan Gulo sebagai penolong kepada penulis ketika melakukan pembuatan skripsi.
19. Nurul Sofia Dewi yang telah banyak membantu penulis ketika mempelajari tentang arduino.

Skripsi ini belum sempurna sehingga saran dan kritik sangat diharapkan oleh penulis.

Medan, Mei 2024



Rheza Yudhistira Ramadhana  
NIM. 201401061

## **Penerapan *Analog to Digital Conversion* dan *Digital to Analog Conversion* Pada *Microprocessor* Menggunakan Arduino Untuk *Body Awareness* Pada Anak**

### **Balita**

#### **ABSTRAK**

Peningkatan kasus pelecehan seksual terhadap balita, seperti yang tercatat oleh KemenPPPA, menunjukkan eskalasi yang signifikan dari tahun ke tahun. Dalam mengatasi masalah ini, pentingnya pendidikan seks pada anak balita menjadi sorotan, khususnya dalam mengembangkan kesadaran tubuh. Namun, kendala muncul karena tidak semua orang tua dapat memberikan waktu atau keterampilan untuk melibatkan anak dalam aktivitas yang mendukung pengenalan tubuh. Masa balita, sebagai periode rentan, memerlukan perhatian khusus dalam pengembangan kesadaran tubuh. Kurangnya pengawasan dapat membuka peluang terjadinya tindakan kriminal, terutama pelecehan seksual, dan kurangnya pemahaman anak terhadap bagian tubuhnya sendiri dapat mempermudah pelaku kejahatan. Penelitian ini bertujuan untuk menciptakan media pembelajaran *body awareness* menggunakan boneka cerdas yang mengintegrasikan algoritma ADC dan DAC. Dengan demikian, anak dapat belajar mengenali bagian tubuh yang tidak boleh disentuh oleh orang lain melalui respon suara berbasis kondisi. ADC dan DAC merupakan algoritma konversi dari sinyal analog menjadi sinyal digital. Pemanfaatan FSR (*Force Sensitive Resistor*) dan INMP441 *Microphone* dalam penerapan ADC dan DAC menghasilkan output yang sesuai dengan keinginan. FSR menginputkan sinyal analog yang didapat dari intensitas tekanan pada sensor. INMP441 menghasilkan sinyal analog yang langsung dikonversi menjadi bit-bit sinyal digital. Bit-bit yang didapat dari gelombang suara diproses menggunakan API Google *Speech-to-Text* dan dikembalikan dalam bentuk *string*. Sinyal digital dikonversi kembali menjadi sinyal analog yang berupa suara tersimpan di *SD Card* dan dipanggil tergantung dengan kondisi. Tingkat keakuratan respon terhadap nilai analog yang berasal dari sensor FSR sebesar 100% dan untuk *voice recognition* mencapai 96%.

**Kata Kunci :** *Body Awareness, Voice Recognition, ADC, DAC, FSR.*

## **Application of Analogue to Digital Conversion and Digital to Analogue Conversion on Microprocessor Using Arduino for Body Awareness in Toddler**

### **ABSTRACT**

The increase in sexual abuse cases against toddlers, as recorded by KemenPPPA, shows a significant escalation from year to year. In addressing this issue, the importance of sex education in children under five is highlighted, particularly in developing body awareness. However, obstacles arise because not all parents can provide the time or skills to engage children in activities that support body recognition. Toddlerhood, as a vulnerable period, requires special attention in developing body awareness. Lack of supervision can open up opportunities for criminal acts, especially sexual abuse, and children's lack of understanding of their own body parts can make it easier for criminals. This research aims to create body awareness learning media using intelligent dolls that integrate ADC and DAC algorithms. Thus, children can learn to recognise body parts that should not be touched by others through condition-based voice responses. ADC and DAC are conversion algorithms from analogue signals to digital signals. The utilisation of FSR (Force Sensitive Resistor) and INMP441 Microphone in the application of ADC and DAC produces the desired output. FSR inputs an analogue signal obtained from the intensity of the pressure on the sensor. INMP441 produces an analogue signal that is directly converted into digital signal bits. The bits obtained from the sound waves are processed using the Google Speech-to-Text API and returned in the form of a string. The digital signal is converted back into an analogue signal in the form of a voice stored on the SD Card and called depending on the conditions. The accuracy of the response to the analogue value coming from the FSR sensor is 100% and for voice recognition it reaches 96%.

**Keywords :** *Body Awareness, Voice Recognition, ADC, DAC, FSR.*



## DAFTAR ISI

<b>PERSETUJUAN.....</b>	<b>ii</b>
<b>PERNYATAAN.....</b>	<b>iii</b>
<b>KATA PENGANTAR.....</b>	<b>iv</b>
<b>ABSTRAK.....</b>	<b>vi</b>
<b>ABSTRACT.....</b>	<b>vii</b>
<b>DAFTAR ISI.....</b>	<b>viii</b>
<b>DAFTAR TABEL.....</b>	<b>x</b>
<b>DAFTAR GAMBAR.....</b>	<b>xi</b>
<b>BAB I.....</b>	<b>1</b>
<b>PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	3
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian.....	4
1.5 Manfaat Penelitian.....	4
1.6 Metodologi Penelitian .....	5
1.7 Sistematika Penulisan.....	7
<b>BAB II.....</b>	<b>8</b>
<b>LANDASAN TEORI.....</b>	<b>8</b>
2.1 <i>Analog to Digital Conversion (ADC)</i> .....	8
2.2 <i>Digital to Analog Conversion (DAC)</i> .....	9
2.3 Arduino.....	9
2.4 <i>Application Programming Interface (API)</i> .....	10
2.5 <i>Microprocessor</i> .....	11
2.6 Body Awareness.....	12
2.7 ESP-32.....	12
2.8 Penelitian Relevan .....	12
<b>BAB III.....</b>	<b>15</b>
<b>ANALISIS DAN PERANCANGAN SISTEM.....</b>	<b>15</b>
3.1 Analisis Sistem.....	15

3.1.1 Analisis Masalah .....	15
3.1.2 Analisis Kebutuhan .....	17
3.1.2.1 Kebutuhan Fungsional.....	17
3.1.2.2 Kebutuhan Non-Fungsional .....	17
3.1.3 Diagram Umum Sistem .....	18
3.1.4 Diagram Ishikawa.....	19
3.2 Pemodelan Sistem .....	19
3.2.1 <i>Use Case Diagram</i> .....	20
3.2.2 <i>Activity Diagram</i> .....	21
3.2.3 <i>Sequence Diagram</i> .....	21
3.3 Flowchart.....	22
3.3.1 <i>Flowchart Pressure Sensor</i> .....	23
3.3.2 <i>Flowchart Voice Recognition</i> .....	25
<b>BAB IV .....</b>	<b>26</b>
<b>IMPLEMENTASI DAN PENGUJIAN SISTEM .....</b>	<b>26</b>
4.1 Implementasi dan Hasil .....	26
4.1.1 Penggunaan Alat.....	26
4.1.2 Bentuk Desain Sistem .....	27
4.2 Pengujian Sistem .....	28
4.2.1 Pengujian Program Pada FSR ( <i>Force Sensitive Resistor</i> ) dan <i>Output</i> Suara.....	28
4.2.2 Pengujian Pada <i>Voice Recognition</i> .....	32
4.3 Tampilan Akhir Prototype .....	40
<b>BAB V .....</b>	<b>41</b>
<b>KESIMPULAN DAN SARAN .....</b>	<b>41</b>
5.1 Kesimpulan.....	41
5.2 Saran .....	41
<b>DAFTAR PUSTAKA .....</b>	<b>43</b>
<b>LAMPIRAN.....</b>	<b>46</b>
LISTING PROGRAM.....	46
CURRICULUM VITAE .....	60

**DAFTAR TABEL**

Tabel 4.1 Tabel Alat yang Digunakan.....	26
Tabel 4.2 Tabel <i>Output</i> Berdasarkan <i>Input</i> Nilai Analog .....	31
Tabel 4.3 Tabel Input Untuk <i>Trigger</i> Output Suara.....	39

## DAFTAR GAMBAR

<i>Gambar 2.1</i> Sinyal Analog dan Proses Perubahan dari Sinyal Analog ke Sinyal Digital	8
<i>Gambar 2.2</i> Sinyal Digital (Sumber : Syam, 2014)	9
<i>Gambar 2.3</i> Ilustrasi API	11
<i>Gambar 3.1</i> Diagram Umum Sistem	18
<i>Gambar 3.2</i> Diagram Ishikawa	19
<i>Gambar 3.3</i> Use Case Diagram	20
<i>Gambar 3.4</i> Activity Diagram	21
<i>Gambar 3.5</i> Sequence Diagram	22
<i>Gambar 3.6</i> Flowchart Pressure Sensor	23
<i>Gambar 3.7</i> Flowchart Voice Recognition	25
<i>Gambar 4.1</i> Tampilan Rangkaian <i>Prototype</i> Untuk Penggunaan Sensor FSR	27
<i>Gambar 4.2</i> Tampilan Rangkaian <i>Prototype</i> Untuk Penggunaan Voice Recognition	27
<i>Gambar 4.3</i> Tampilan Hasil Dari Pengecekan Isi <i>SD Card</i>	29
<i>Gambar 4.4</i> Uji Coba Menjalankan Musik Dari Dalam <i>SD Card</i>	29
<i>Gambar 4.5</i> Nilai Analog yang Keluar Ketika 2 Sensor FSR yang Berbeda Ditekan	30
<i>Gambar 4.6</i> Grafik Ketika Sensor Dalam Proses Ditekan	30
<i>Gambar 4.7</i> Pengecekan Sinyal <i>Microphone</i> INMP441 Berdasarkan <i>Datasheet</i>	32
<i>Gambar 4.8</i> Tampilan Sinyal Analog Yang Masuk Dari Mikrofon INMP441	33
<i>Gambar 4.9</i> Uji Coba N9200 MP3 Player Module Menggunakan Komunikasi Serial	34
<i>Gambar 4.10</i> Tampilan Ketika Memilih <i>Credentials</i>	35
<i>Gambar 4.11</i> Tampilan Pengisian <i>Service Account Details</i>	35
<i>Gambar 4.12</i> Tampilan Pengisian <i>Service Account Access</i>	36
<i>Gambar 4.13</i> Tampilan Pengisian Akses <i>User</i> ke <i>Service Account</i>	36
<i>Gambar 4.14</i> Pengambilan Token API	37
<i>Gambar 4.15</i> Tampilan Hasil dari Uji Coba Program Voice Recognition	38
<i>Gambar 4.16</i> Bentuk Akhir <i>Prototype</i>	40

## **BAB I**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

Kasus pelecehan seksual terhadap balita mengalami peningkatan. Menurut data KemenPPPA, kasus kekerasan seksual terhadap anak mencapai 9.588 kasus pada tahun 2022. Angka ini menunjukkan peningkatan dari tahun sebelumnya yang tercatat sebanyak 4.162 kasus. (Sumber Berita CNN). Menurut data KemenPPPA anak balita laki-laki mengambil bagian dari 14% dari 812 kasus yang terdata. Sedangkan anak balita perempuan mengambil bagian 5,7% dari 4.193 kasus yang terdata (Sumber Website KemenPPPA).

Pada masa perkembangan anak balita, *body awareness* atau kesadaran tubuh merupakan keterampilan penting yang harus dikuasai. *Body awareness* meliputi kemampuan untuk memahami dan mengendalikan respon tubuh, serta kemampuan untuk memahami dan mengenali bagian-bagian tubuh, oleh karena itu, sangat penting untuk meningkatkan pemahaman anak-anak tentang bagian-bagian tubuh mereka dan memberikan edukasi mengenai tindakan yang tidak boleh dilakukan oleh orang lain terhadap bagian-bagian tubuh tersebut (Russel, 2020).

Untuk membantu anak balita mengembangkan *body awareness*, dapat dilakukan berbagai aktivitas yang melibatkan respon dan pengenalan bagian-bagian tubuh. Aktivitas ini biasanya didampingi oleh orang tua atau pengasuh anak yang sudah dipercayakan. Namun, tidak semua orang tua atau pengasuh memiliki waktu atau keterampilan untuk melakukan aktivitas tersebut. Pendidikan seks pada anak usia dini sebaiknya diawali dengan pengenalan anggota-anggota tubuh atau bagian-bagian tubuh pada anak, kemudian dilanjutkan dengan materi-materi lain yang relevan (Suhsmi, 2021).

Dengan perkembangan dan penggunaan teknologi *microprocessor* dapat membantu meningkatkan kesadaran anak tentang bagian-bagian tubuh dan memberikan pendidikan tentang apa yang tidak boleh dilakukan oleh orang lain pada bagian-bagian tubuh mereka. Salah satu teknologi yang dapat digunakan

adalah boneka yang dilengkapi dengan multi sensor dan algoritma ADC (*Analog to Digital Conversion*) dan DAC (*Digital to Analog Conversion*) yang diimplementasikan pada *microprocessor* Arduino (Barret, 2022).

ADC (*Analog to Digital Conversion*) merupakan proses perubahan sinyal analog ke digital yang dilalui proses sampling dan quantizing hingga ke coding (Priyono, 2022). Algoritma ADC berada di komponen yang bertugas untuk mengkonversi sinyal analog ke digital atau biasa disebut dengan ADCs (*Analog to Digital Converters*).

Sedangkan DAC (*Digital to Analog Conversion*) merupakan algoritma yang merubah sinyal digital yang diterima dan mengubahnya kembali menjadi sinyal analog (Britannica, 2023). Algoritma DAC ini digunakan untuk memproses hasil sinyal digital yang telah dikonversi oleh ADC pada komponen yang berada pada Arduino MEGA.

Boneka yang dilengkapi dengan beberapa sensor dan *microphone* yang mampu mengukur tingkat kekuatan tekanan dan memproses suara yang dilakukan oleh anak saat anak memberikan inputan berupa tekanan serta suara. Sensor yang digunakan merupakan Force Sensitive Resistor (FSR). Sensor-sensor tersebut menghasilkan sinyal analog yang kemudian diubah menjadi data digital menggunakan algoritma ADC (*Analog to Digital Conversion*) (Syam, 2014). Data digital tersebut kemudian diproses menggunakan algoritma DAC (*Digital to Analog Conversion*) untuk menghasilkan suara yang memberikan respon atau instruksi kepada anak (Arar, 2021).

Sebagai contoh, jika anak diminta untuk menyentuh bagian-bagian tubuh tertentu pada boneka dan juga mengucapkan beberapa kata pada boneka. Sensor pada boneka akan mengukur tekanan tangan anak dan mengolah kata-kata yang diucapkan oleh anak lalu mengirimkan sinyal analog ke *microprocessor*. Sinyal tersebut kemudian diubah menjadi data digital menggunakan algoritma ADC (*Analog to Digital Conversion*) (Arar, 2021). Data digital tersebut kemudian diproses menggunakan algoritma DAC (*Digital to Analog Conversion*) untuk menghasilkan suara yang memberikan petunjuk kepada anak untuk tidak menyentuh bagian-bagian tubuh tertentu (Arar, 2021).

Dengan menggunakan boneka yang dilengkapi dengan algoritma ADC (*Analog to Digital Conversion*) dan DAC (*Digital to Analog Conversion*), anak balita dapat belajar mengenali dan mempelajari bagian tubuh yang tidak boleh disentuh oleh orang lain.

## 1.2 Rumusan Masalah

Masa balita merupakan masa rentan yang butuh perhatian khusus dari orang dewasa khususnya orang tua dalam mengedukasi *body awareness* pada anak. Kurangnya perhatian dan pengawasan pada anak balita menyebabkan sering terjadinya tindakan kriminal salah satunya pelecehan seksual dan minimnya pemahaman dari anak mengenai bagian penting pada tubuhnya sendiri membuat para pelaku pelecehan seksual semakin leluasa untuk melakukan tindakannya. Akibatnya anak balita yang menjadi korban akan mengalami trauma hingga dewasa. Oleh karena itu penelitian ini bertujuan membuat media pembelajaran *body awareness* untuk anak balita menggunakan sarana boneka yang menerapkan algoritma ADC (*Analog to Digital Conversion*) dan DAC (*Digital to Analog Conversion*) yang berguna untuk mengolah sinyal sensor dan sinyal suara hingga merubahnya menjadi respon suara berdasarkan kondisi. Penggunaan boneka tersebut akan ditujukan kepada tenaga pendidik anak balita sebagai bahan edukasi.

## 1.3 Batasan Masalah

Dalam melakukan penelitian ini, penulis membatasi ruang masalah yang akan diteliti. Batasan masalah yang digunakan antara lain :

1. *Microprocessor* yang dibangun hanya bertujuan untuk menstimulasi anak agar dapat memahami respon ketika tempat yang seharusnya tidak disentuh tetapi malah disentuh serta memahami respon yang tepat terhadap ajakan orang asing.
2. Input yang dibutuhkan oleh *microprocessor* yang telah dibuat agar dapat memberi respon dan dapat mengajarkan *body awareness* adalah sentuhan

dan tekanan tangan ke tempat-tempat yang diatur dan juga mengajarkan tindakan yang harus dilakukan terhadap ajakan orang asing.

3. Metode yang digunakan dalam membuat *prototype* ini adalah dengan metode algoritma ADC (*Analog to Digital Conversion*) dan DAC (*Digital to Analog Conversion*).
4. *Microprocessor* ini akan menghasilkan output yang berupa output suara tergantung dengan tingkat tekanan ataupun area yang ditekan dan juga memberikan output suara berdasarkan suara yang diucapkan.
5. Tidak membahas penggunaan NLP dalam penerapan *prototype*.
6. *Microprocessor* Arduino berperan sebagai modul pengolahan sensor FSR sedangkan ESP-32 sebagai modul *voice recognition*.
7. *Microprocessor* ini akan dibuat dalam bentuk *prototype*.
8. Tingkat nilai analog sensor FSR untuk merespon ada 5. Nilai < 20 (Tidak Ada Sentuhan), nilai < 50 (Tekanan Sangat Lembut), nilai < 200 (Tekanan Normal), nilai < 500 (Tekanan Kuat), nilai < 800 (Tekanan Sangat Kuat).

#### 1.4 Tujuan Penelitian

Berdasarkan rumusan masalah di atas, maka tujuan dari penelitian ini adalah membangun sebuah *prototype* dengan metode ADC (*Analog to Digital Conversion*) dan DAC (*Digital to Analog Conversion*) yang mampu memberikan respon tergantung dengan tingkat tekanan dan area tekanan yang dilakukan oleh balita serta dapat memberikan respon suara berdasarkan suara ajakan yang diberikan. Sehingga balita dapat memahami pemahaman *body awareness* dari *prototype* yang telah dibuat.

#### 1.5 Manfaat Penelitian

Manfaat yang diperoleh dari penelitian ini antara lain :

1. Balita akan menjadi memahami respon apa yang harus dilakukan ketika terjadi tindakan seperti menyentuh area vital maupun menekan dengan kuat serta dapat memahami respon yang tepat terhadap ajakan orang asing.



2. Dengan *prototype* ini, guru-guru paud atau tenaga yang mendidik anak-anak dapat mengajarkan kepada balita secara kreatif dan responsif.

## 1.6 Metodologi Penelitian

Metode yang digunakan dalam penelitian ini adalah:

1. Studi Pustaka

Pada fase permulaan penelitian ini, langkah pertama yang diambil adalah menginisiasi proses dengan menghimpun berbagai referensi dari berbagai sumber, termasuk buku, artikel ilmiah, jurnal, dan skripsi terdahulu yang memiliki relevansi dengan topik Penerapan *Analog to Digital Conversion dan Digital to Analog Conversion* Pada *Microprocessor* Menggunakan *Arduino* Untuk *Body Awareness* Pada Anak Balita. Tujuan dari tindakan ini adalah untuk memperoleh informasi yang terkait dan mendalam terkait dengan penelitian yang sedang dilaksanakan.

2. Pengumpulan Data

Pada tahap pengumpulan data merupakan tahap dimana proses mengumpulkan data pengetahuan yang dibutuhkan. Data yang dimaksud seperti tingkat respon sensor terhadap tekanan kekuatan anak balita, nilai kekuatan cengkraman anak-anak berdasarkan umur, dan respon terhadap beberapa kalimat ajakan.

3. Pemodelan Sistem

Pada tahap pemodelan sistem akan dilakukan proses desain rancangan dasar dari arduino yang dibuat pada Proteus 8. Selain itu juga dibuat desain dari Use case diagram, Activity diagram, Sequence diagram, dan diagram alir (flowchart).

4. Implementasi Sistem

Tahap ini adalah proses pembuatan sistem (coding) berdasarkan flowchart yang dirancang dengan menggunakan bahasa pemrograman.

#### 5. Pengujian Sistem

Pada tahapan ini akan dilakukan pengujian sistem yang akan menimbang apakah sistem yang telah dibuat sesuai dengan spesifikasi yang diinginkan dalam penelitian ini.

#### 6. Dokumentasi

Pada tahap dokumentasi, akan dilakukan dokumentasi dari sistem yang telah dibuat mulai dari tahap analisis sampai dengan pengujian dalam bentuk skripsi.

## 1.7 Sistematika Penulisan

Sistematika penulisan dari skripsi ini terdiri dari lima bab, yakni:

### **BAB I            PENDAHULUAN**

Semua yang berkaitan mengenai latar belakang dari permasalahan yang akan diteliti yakni rumusan masalah, tujuan penelitian, batasan masalah, tujuan penelitian, manfaat penelitian, metode penelitian, dan sistematika penulisan dijabarkan pada bab ini.

### **BAB II           LANDASAN TEORI**

Tinjauan teoritis yang berkaitan dengan *Analog to Digital Conversion (ADC)* dan *Digital to Analog Conversion (DAC)* akan dibahas pada bab ini.

### **BAB III          ANALISIS PERANCANGAN**

Melakukan analisis terhadap masalah dan sistem yang dibangun akan dibahas pada bab ini. Kemudian akan dilanjutkan dengan perancangan sistem dengan menggunakan *Analog to Digital Conversion (ADC)* dan *Digital to Analog Conversion (DAC)*.

### **BAB IV          IMPLEMENTASI DAN PENGUJIAN SISTEM**

Pada bab ini akan dilakukan implementasi dan pengujian sistem berdasarkan tahapan analisis dan perancangan yang telah disebutkan diatas.

### **BAB V           PENUTUP**

Pada bab ini akan berisi kesimpulan dan saran dari penelitian yang telah dilakukan untuk pengembangan penelitian selanjutnya.

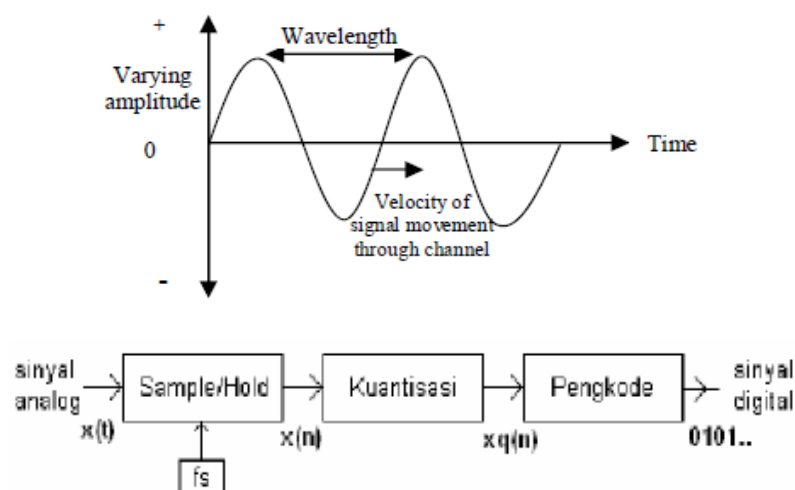
## BAB II

### LANDASAN TEORI

#### 2.1 *Analog to Digital Conversion (ADC)*

Arsitektur *Analog to Digital Conversion (ADC)* digunakan untuk mengonversi output analog dari sensor nonlinier, hasil konversi akan membutuhkan tambahan siklus pemrosesan tambahan untuk melinearisasi, sehingga membuat sistem yang lengkap membutuhkan lebih banyak daya. Mungkin juga memerlukan penggunaan ADC dengan resolusi yang lebih tinggi daripada yang lain diperlukan untuk memperoleh rentang sinyal penuh dengan presisi yang diinginkan. Ini adalah kasusnya ketika, untuk setiap zona yang diminati, sensor menunjukkan variasi kecil dalam output ketika rangsangan input sangat bervariasi (Santos, 2019).

Secara ringkas, *Analog to Digital Converter (ADC)* merupakan perangkat atau modul yang mengubah sinyal analog menjadi bentuk digital seperti pada Gambar 2.1. ADC banyak digunakan dalam berbagai aplikasi, berfungsi sebagai jembatan antara sensor analog dan sistem digital. Kinerja ADC ditentukan oleh tingkat sampling, yang menunjukkan seberapa sering konversi dilakukan, dan resolusi, yang menentukan tingkat detail dalam representasi digital dari masukan analog.

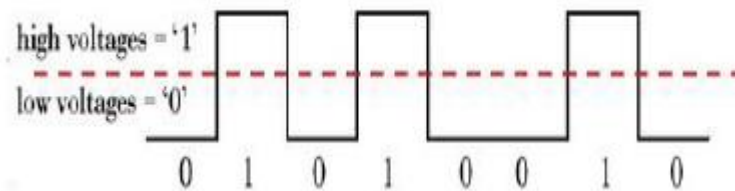


**Gambar 2.1** Sinyal Analog dan Proses Perubahan dari Sinyal Analog ke Sinyal Digital (Sumber: Syam, 2014)

## 2.2 Digital to Analog Conversion (DAC)

*Digital to Analog Conversion* (DAC) atau Konversi Digital ke Analog merupakan proses pengubahan sinyal digital yang diterima hingga diubah menjadi sinyal analog. Sebagai contoh sebuah sinyal digital dari modem diubah menjadi sinyal analog frekuensi suara yang digunakan pada sambungan telepon (Britannica, 2023). DAC menggunakan inputan sinyal digital. Ini bisa berupa urutan biner, di mana setiap bit mewakili nilai biner (0 atau 1) seperti pada Gambar 2.2.

DACs atau biasa disebut dengan *Digital to Analog Converters* merupakan komponen yang menggunakan algoritma DAC. Penggunaan DACs pada Jurnal “Methods of Digital-To-Analog Conversion for Reproduction of Sound Waves” (Danylenko, dkk, 2022) menggunakan DACs sebagai alat yang berguna untuk menciptakan gelombang suara.



**Gambar 2.2** Sinyal Digital (Sumber : Syam, 2014)

## 2.3 Arduino

Diambil dari “*Arduino Documentation*” (Arduino, 2015) Arduino merupakan entitas perangkat keras dan perangkat lunak komputer yang bersifat *open source*. Komunitas Arduino merujuk pada kumpulan proyek dan pengguna yang merancang serta menggunakan papan pengembangan berbasis mikrokontroler. Papan pengembangan tersebut, yang dikenal sebagai Modul Arduino, merupakan *platform prototyping* yang bersifat *open source*. Berbagai jenis paket papan pengembangan yang menggunakan mikrokontroler yang disederhanakan ini tersedia untuk digunakan.

Arduino menyediakan berbagai varian dan model papan pengembangan, seperti Arduino Uno, Arduino Mega, dan Arduino Nano. Setiap model memiliki

keunggulan dan kelebihan tersendiri sesuai dengan kebutuhan pengguna. Selain itu, tersedia juga modul tambahan yang dapat digunakan untuk memperluas fungsionalitas papan Arduino, seperti modul komunikasi nirkabel, sensor-sensor tambahan, dan modul pengendali motor.

Selain perkembangan perangkat keras, Arduino juga terus mengembangkan perangkat lunaknya. Arduino IDE, singkatan dari Integrated Development Environment, merupakan software yang digunakan untuk memprogram *Arduino board*. dengan bahasa pemrograman yang sederhana dan mudah dipahami. Selain itu, telah ada pengembangan dan dukungan untuk berbagai bahasa pemrograman lainnya, seperti Python dan JavaScript, yang memperluas fleksibilitas dalam pengembangan proyek.

#### **2.4 Application Programming Interface (API)**

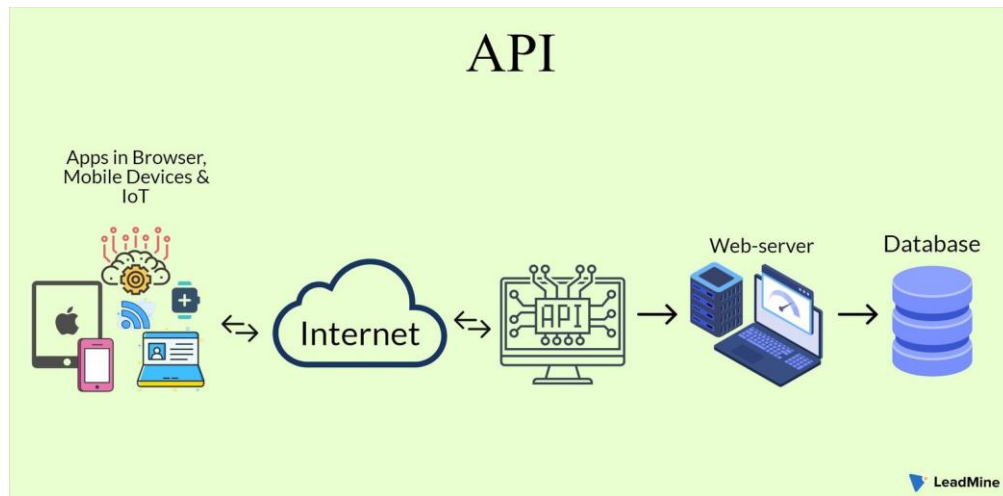
Antarmuka Pemrograman Aplikasi (API) adalah sekumpulan protokol dan perintah yang memfasilitasi interaksi antara perangkat lunak. API berfungsi sebagai jembatan yang memungkinkan pertukaran data efisien antara dua atau lebih aplikasi.

Keberadaan API menguntungkan pengembang dengan menghilangkan kebutuhan untuk memahami secara mendalam detail aplikasi lain. Mereka dapat dengan mudah mengintegrasikan fungsionalitas dari satu aplikasi ke dalam yang lain tanpa kesulitan yang berlebihan.

API dapat mencakup berbagai elemen seperti instruksi, protokol komunikasi, atau bahkan fungsi dari perpustakaan perangkat lunak. Penggunaannya melibatkan akses ke halaman web atau integrasi dengan platform lain.

Selain itu, API juga dapat berperan sebagai sarana untuk mendistribusikan Data Terbuka (*Open Data*) seperti pada Gambar 2.3. Sebagai contoh, dalam manipulasi data, API memungkinkan pengembang untuk mengambil sebagian kecil dari situs web tanpa mengubah struktur keseluruhan. Misalnya, konversi suhu dari Fahrenheit ke Celcius dapat dilakukan dengan meminta data melalui API tanpa memanipulasi tampilan halaman web. Ini memberikan fleksibilitas yang besar dalam mengakses dan menggunakan informasi dari berbagai sumber

tanpa harus terkungung oleh rancangan halaman web yang spesifik. (Sugimoto, G. 2019).



**Gambar 2.3** Ilustrasi API (Sumber : leadmine.net)

## 2.5 Microprocessor

Mikroprosesor, yang juga dikenal sebagai CPU atau Central Processing Unit, adalah sebuah rangkaian terintegrasi (IC) yang berfungsi sebagai unit pengolah dalam sebuah mesin. Software membutuhkan sistem operasi dan kernel untuk berjalan. Namun, apabila mikroprosesor mengalami keusangan akan mempengaruhi kinerja tersebut (Naharisetti, 2023)

Revolusi mikroprosesor, yang menandai ulang tahunnya yang ke-50 dalam edisi ini, telah mendorong terciptanya inovasi luar biasa dalam arsitektur set instruksi, mikro-arsitektur, dan desain sistem. Beberapa fitur tersebut merupakan konsep yang sepenuhnya baru, sementara yang lainnya memiliki akar dalam arsitektur prosesor sebelum revolusi ini. Bahkan, di antara teknologi kuno, ada satu yang terus berkembang di era teknologi informasi (TI) modern: mainframe IBM. Diluncurkan sebagai S/360 pada tahun 1964, mainframe ini, sekarang dikenal sebagai IBM Z, telah menjadi cikal bakal beberapa fitur "modern" (Webb, 2021).

## 2.6 Body Awareness

Menurut Jurnal “*The Impact of Body Awareness on Women’s Sexual Health: A Comprehensive Review*” (Seal, dkk, 2020) Peningkatan kesadaran tubuh tampaknya memiliki dampak positif terhadap kesejahteraan seksual sejumlah wanita. Temuan ini diperkuat oleh hasil penelitian di laboratorium yang melibatkan partisipan wanita dengan dan tanpa kesulitan seksual. Begitu pula, penelitian intervensi yang berfokus pada indra dan kesadaran juga memberikan dukungan untuk wanita yang mengalami berbagai masalah seksual. Namun, dalam studi intervensi, sejauh mana peningkatan kesadaran tubuh memberikan kontribusi terhadap hasil seringkali kurang jelas karena adanya fitur lain dari intervensi dan/atau desain studi. Ringkasan ini menggambarkan intervensi dan studi laboratorium mengenai kesadaran tubuh, menunjukkan bahwa kesadaran tubuh memegang peranan penting dalam penanganan disfungsi seksual.

## 2.7 ESP-32

Mikrokontroler ESP32 menonjol sebagai sistem-on-chip yang menyatukan berbagai fungsi dengan kebutuhan daya yang rendah dan biaya yang terjangkau. Dalam penelitian ini, ESP32 berperan sebagai otak utama dan penghubung untuk berbagai sensor yang akan diintegrasikan dalam rangkaian sistem peringatan dini kebakaran. Keunggulan ESP32 meliputi kemampuan Wi-Fi, memori flash berkapasitas besar, mode Bluetooth ganda, dan sejumlah fitur canggih lainnya. Sebagai pengembangan dari mikrokontroler 8266, ESP32 menawarkan frekuensi radio terbaik (RF) dan kinerja optimal dalam berbagai skenario aplikasi. Keberhasilan penggunaan modul ESP-32 telah banyak terbukti, khususnya dalam pengembangan sistem IoT dengan biaya yang efisien (Anggarawan, 2022).

## 2.8 Penelitian Relevan

Berikut ini beberapa penelitian yang berkaitan dengan body awareness dan microprocessor menggunakan ADC (*Analog to Digital Conversion*) atau DAC (*Digital to Analog Conversion*):



1. Penelitian yang dilakukan oleh Mirna Ari Mulyani *et al.*, (2016) dengan judul Interpretasi dan Strategi Kesadaran Tubuh sebagai Konsep Memahami (Individu Interpretation and Body Awareness Strategies) diperoleh bahwa pentingnya kesadaran tubuh dalam mengenali, memahami, dan merespons sensasi dan proses tubuh secara mendalam. Dan juga strategi yang digunakan individu dalam mengembangkan kesadaran tubuh seperti menggunakan teknik relaksasi, meditasi, pengaturan pernapasan, dan latihan fisik tertentu.
2. Penelitian yang dilakukan oleh Elgamar Syam, (2014) dengan judul Analisa dan Implementasi Transformasi Analog to Digital Converter (ADC) untuk Mengkonversi Suara ke Bentuk Teks diperoleh bahwa teknologi algoritma ADC (*Analog to Digital Converter*) memiliki keunggulan dalam mengubah sinyal suara menjadi bentuk teks yang dapat diproses diproses lebih lanjut oleh komputer.
3. Penelitian yang dilakukan oleh Noprita Elisabeth *et al.*, (2018) dengan judul Desain Komunikasi Visual Iklan Layanan Masyarakat Tentang Pelecehan Seksual Pada Anak di Kota Medan diperoleh bahwa pentingnya penggunaan media iklan layanan masyarakat sebagai sarana edukasi dan pencegahan pelecehan seksual pada anak. Karena rendahnya tingkat kesadaran dan pemahaman masyarakat dibutuhkan media komunikasi visual yang efektif dan menarik sehingga meningkatkan kesadaran dan pemahaman masyarakat.
4. Penelitian yang dilakukan oleh Hermaini Siswati, (2018) dengan judul Lindungi Anak Dari Korban dan Pelaku Pelecehan Seksual diperoleh bahwa pentingnya melindungi anak dari bahaya pelecehan seksual yang diderita baik sebagai korban maupun sebagai pelaku yang dimana dapat menggunakan beberapa metode untuk mencegahnya seperti pendidikan seksual yang sehat, penguatan pola asuh yang positif dan penegakan hukum yang tegas bagi pelaku. Dan juga peran masyarakat dan keluarga yang penting dalam membangun lingkungan yang aman dan mendukung perkembangan anak yang sehat.

5. Penelitian yang dilakukan oleh Berry Richard Ornos Sinambela *et al.*, (2020) dengan judul Penerapan Algoritma Start-Step-Stop Code Pada Kompresi File Audio diperoleh hasil pengujian bahwa algoritma SSSC (*Start-Step-Stop Code*) mampu mencapai tingkat kompresi yang lebih baik daripada beberapa metode kompresi audio lainnya. Oleh karena itu, penerapan algoritma SSSC pada kompresi file audio dapat membantu mengurangi ukuran file audio, meningkatkan efisiensi penyimpanan data, dan mempercepat waktu transfer data melalui jaringan komputer.
6. Penelitian yang dilakukan oleh Danylenko *et al.*, (2022) dengan judul *Methods of Digital-To-Analog Conversion for Reproduction of Sound Waves* ini membahas metode konversi *digital-to-analog* yang inovatif menggunakan aliran udara yang dihasilkan oleh kipas untuk mentransfer informasi suara. Kompleks perangkat lunak-perangkat keras, algoritma, dan hasil eksperimen yang dilakukan untuk memverifikasi efektivitas metode ini dijelaskan secara rinci. Eksperimen melibatkan partisipan yang harus mengidentifikasi berbagai kelas suara yang direproduksi oleh kompleks yang dibuat. Hasil eksperimen menunjukkan bahwa orang dapat mengidentifikasi aliran udara dari kompleks dengan tingkat keberhasilan sekitar 83%. Selain itu, penelitian juga mencoba menggunakan sistem klasifikasi perangkat lunak untuk mengidentifikasi suara dengan hasil yang kurang berhasil. Penelitian ini menyarankan bahwa kompleks semacam itu dapat digunakan di pusat-pusat dengan orang-orang yang memiliki masalah pendengaran atau di tempat-tempat perlindungan untuk memberikan informasi penting secara cepat. Langkah penelitian selanjutnya akan mencakup eksperimen dengan konfigurasi pengaturan kipas yang berbeda untuk meningkatkan hasil eksperimen.

## **BAB III**

### **ANALISIS DAN PERANCANGAN SISTEM**

#### **3.1 Analisis Sistem**

Proses analisis sistem adalah tahap kritis dalam penelitian yang memiliki tujuan mendalam untuk merinci unsur-unsur yang diperlukan agar suatu sistem dapat beroperasi secara optimal. Analisis sistem terbagi menjadi dua aspek utama, yaitu analisis masalah yang bertujuan mengidentifikasi akar penyebab dan konsekuensi dari suatu permasalahan, serta analisis kebutuhan yang ditujukan untuk menyelidiki informasi dan langkah-langkah yang diperlukan dalam merancang suatu sistem.

##### **3.1.1 Analisis Masalah**

Pada penelitian, proses analisis masalah memiliki tujuan untuk dapat mengetahui suatu penyebab dari permasalahan yang ada lalu dilakukan pengkajian lebih dalam terhadap masalah yang ingin diteliti. Dalam proses penelitian ini dilakukan analisis dalam bagaimana *Analog to Digital Conversion* dan *Digital to Analog Conversion* dapat menyelesaikan masalah pada anak balita yaitu kurangnya pemahaman tentang *body awareness*.

Dalam melakukan analisis masalah, metode yang sangat bermanfaat adalah metode *5-Whys*. Metode ini merupakan suatu pendekatan yang memanfaatkan serangkaian pertanyaan "Mengapa?" secara berulang untuk merinci akar penyebab suatu masalah. Fungsinya adalah untuk menemukan penyebab mendasar dari suatu permasalahan dengan mengidentifikasi serta menyelidiki berbagai faktor yang berkontribusi terhadap timbulnya masalah tersebut. Pendekatan ini membantu memperoleh wawasan yang lebih mendalam dan menyeluruh terkait dengan permasalahan yang dihadapi.

Berikut adalah beberapa pertanyaan yang digunakan untuk mengeksplorasi lebih lanjut masalah penelitian diatas:

1. Mengapa *body awareness* atau kesadaran tubuh menjadi keterampilan penting pada masa perkembangan anak balita?

Kesadaran tubuh penting karena pada masa perkembangan anak balita, mereka sedang belajar untuk memahami dan mengendalikan respon tubuh serta mengenali bagian-bagian tubuh. Kesadaran tubuh membantu dalam pengembangan pemahaman diri dan juga memberikan dasar untuk pendidikan seks yang sehat.

2. Mengapa ADC (*Analog to Digital Conversion*) diperlukan dalam sistem ini?

ADC diperlukan untuk mengubah sinyal analog yang dihasilkan oleh sensor-sensor, seperti *Force Sensitive Resistor* (FSR), pada boneka menjadi format yang dapat diproses oleh komputer atau mikrokontroler. ADC mengonversi sinyal fisik yang terukur menjadi data digital yang dapat diinterpretasikan oleh perangkat elektronik.

3. Mengapa ADC dan DAC digunakan pada boneka untuk memberikan respon kepada anak?

ADC digunakan untuk mengubah sinyal analog dari sensor-sensor, misalnya tekanan tangan anak pada boneka, menjadi data digital. Data digital ini kemudian diolah oleh sistem, dan DAC digunakan untuk mengubah hasil olahan tersebut kembali menjadi sinyal analog, seperti suara. Proses ini memungkinkan boneka memberikan respon atau instruksi kepada anak berdasarkan interaksi mereka.

4. Mengapa penting mengintegrasikan teknologi ini untuk membantu anak mengenali bagian tubuh yang tidak boleh disentuh?

Integrasi teknologi ini penting karena memberikan solusi inovatif untuk membantu anak balita mengenali dan memahami bagian tubuh yang tidak boleh disentuh. Teknologi ini memberikan pengalaman belajar yang interaktif dan responsif, memastikan pendidikan seks yang tepat pada usia dini, terutama dalam mengenali dan melindungi diri dari potensi pelecehan seksual.

### 3.1.2 Analisis Kebutuhan

Analisis kebutuhan merupakan langkah kunci dalam mengidentifikasi data dan prosedur yang esensial dalam perancangan sistem. Dalam proses perancangan, analisis kebutuhan melibatkan penentuan persyaratan fungsional dan non-fungsional yang harus dipenuhi agar sistem yang dikembangkan mampu mencapai tujuan yang telah ditetapkan dengan optimal.

#### 3.1.2.1 Kebutuhan Fungsional

Kebutuhan Fungsional merupakan segala proses yang akan dilakukan sistem dalam memenuhi tujuannya. Kebutuhan Fungsional yang digunakan pada penelitian dan *prototype* ini sebagai berikut:

1. *Prototype* dapat mendeteksi tekanan.
2. *Prototype* dapat mengenali inputan suara dari mikrofon
3. *Prototype* dapat mengeluarkan respon output suara berdasarkan tekanan ataupun suara dari mikrofon.

#### 3.1.2.2 Kebutuhan Non-Fungsional

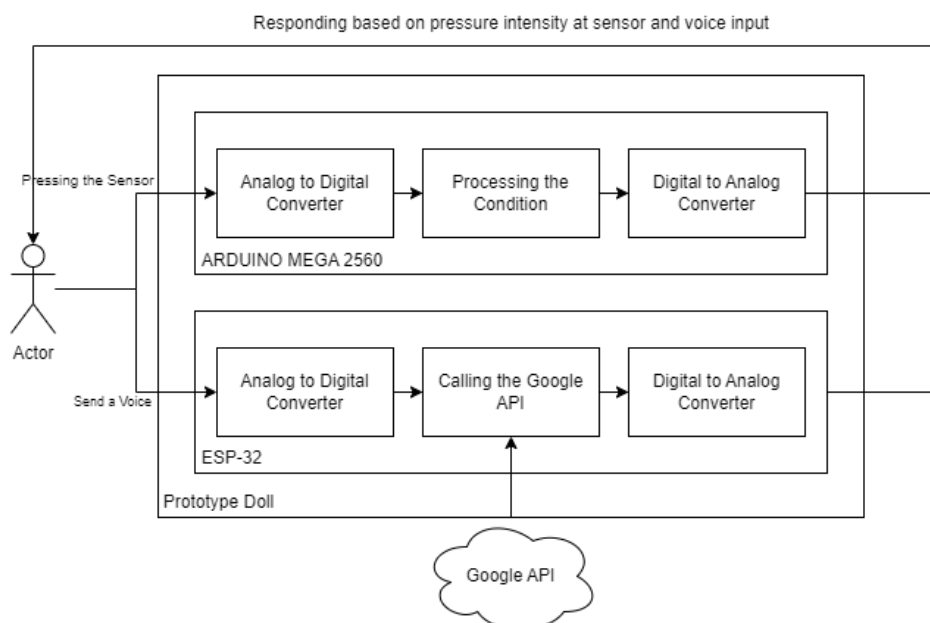
Kebutuhan non-fungsional mencakup fitur, atribut, pembatasan layanan, atau aspek-aspek lain dari sistem, seperti batasan waktu, kendala proses pengembangan, dan standar sistem yang berperan sebagai pelengkap. Di bawah ini terdapat daftar kebutuhan non-fungsional yang diperlukan dalam *prototype* ini:

1. Hanya dapat mendeteksi tekanan dan suara.
2. Inputan suara yang dapat di trigger hanya “ikut” & “om”.
3. Respon suara berdasarkan titik dan tekanan pada sensor. Apabila tekanan lemah maka respon adalah senang. Apabila kuat maka respon adalah marah. Namun apabila titik yang disentuh pada bagian vital, maka respon akan langsung marah.
4. Respon suara dari *voice recognition* merupakan respon dari *text* yang dihasilkan *speech-to-text* oleh API Google.

5. Internet diperlukan untuk terhubung dengan sistem.

### 3.1.3 Diagram Umum Sistem

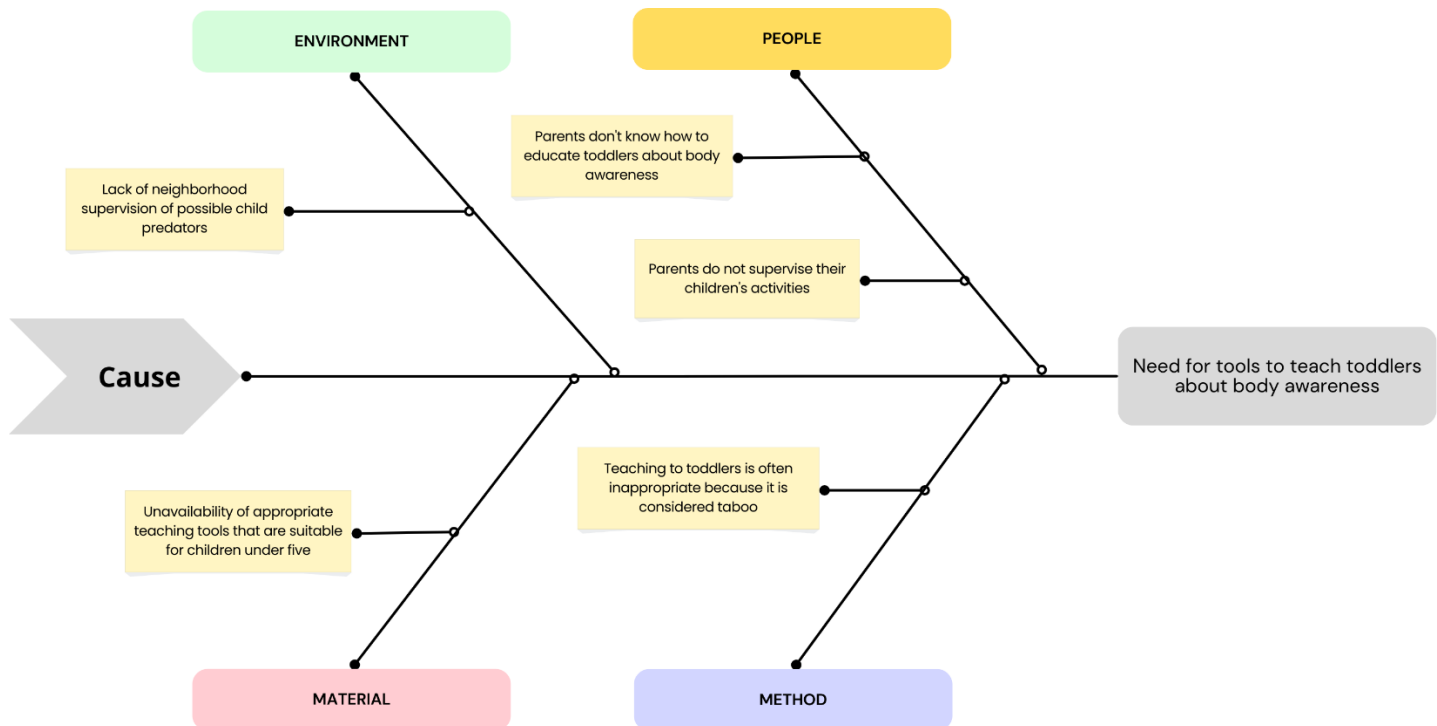
Diagram sistem umum merupakan ilustrasi umum tentang bagaimana sistem dapat berjalan dengan memperlihatkan bagaimana proses, aliran, dan interaksi antara komponen-komponen dalam sistem berlangsung. Seluruh desain aplikasi ini akan dijelaskan secara rinci dalam ilustrasi pada Gambar 3.1 ini:



**Gambar 3.1** Diagram Umum Sistem

Pada Gambar 3.1 terlihat dalam satu *prototype* terdapat dua *microprocessor* yaitu Arduino Mega 2560 dan ESP-32. Arduino menjadi modul yang menjalankan respon terhadap tekanan pada sensor FSR yang dijalankan menggunakan ADC dan DAC. ESP-32 menjadi modul yang mengolah data analog gelombang suara lalu memanggil API Google lalu merespon berdasarkan dengan suara yang dimuat.

### 3.1.4 Diagram Ishikawa



**Gambar 3.2** Diagram Ishikawa

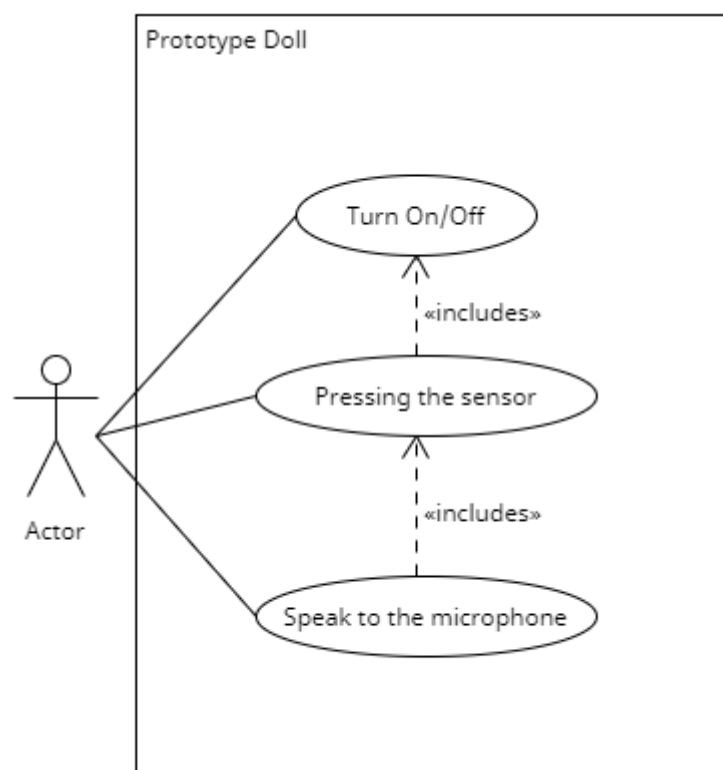
Pada Gambar 3.2 terlihat bahwa faktor-faktor apa saja yang menjadi penyebab kurangnya pemahaman anak balita terhadap *body awareness*. Faktor penyebab merupakan hal hal disekitar mereka mulai dari keluarga, metode edukasi, alat edukasi, lingkungan, dan banyak lagi.

### 3.2 Pemodelan Sistem

Pemodelan sistem merupakan suatu proses yang merinci interaksi antara pengguna dan aplikasi yang dibangun, sehingga memastikan operasional sistem secara efisien. Secara umum, pemodelan sistem diwujudkan dalam bentuk *Unified Modeling Language* (UML). *Unified Modeling Language* (UML) adalah bahasa pemodelan yang umum digunakan untuk menggambarkan hubungan antara komponen dalam sebuah sistem, memungkinkan interaksi melalui pengguna. Dalam penelitian ini, model UML yang digunakan meliputi *Use Case Diagram*, *Activity Diagram*, dan *Sequence Diagram*.

### 3.2.1 Use Case Diagram

*Use Case Diagram* adalah tipe diagram dalam *Unified Modeling Language* (UML) yang bertujuan untuk menggambarkan interaksi antara sistem atau aplikasi dengan aktor-aktor yang terlibat dalam proses tersebut. Aktor, dalam konteks ini, merujuk pada entitas luar yang memiliki peran dalam berinteraksi dengan sistem, seperti pengguna manusia, perangkat keras, atau sistem lainnya. Berikut merupakan representasi diagram use case yang diterapkan dalam sistem:



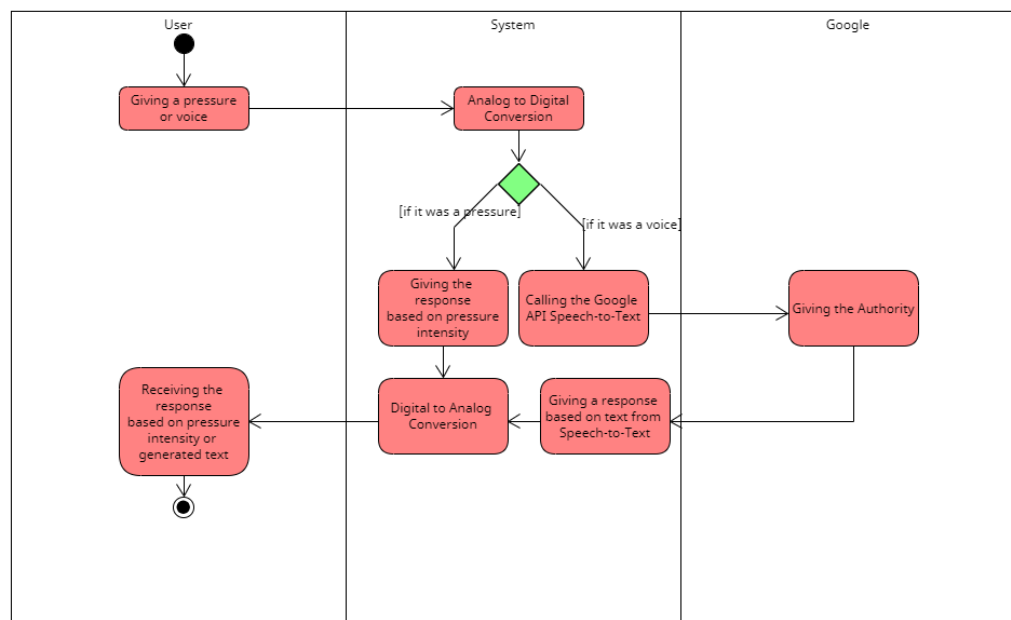
**Gambar 3.3** Use Case Diagram

Gambar 3.3 menunjukkan bahwa aktor melakukan interaksi langsung dengan sistem yang berdasarkan tekanan pada sensor dan berbicara kepada mikrofon yang bergantung pada *Turn On/Off* sistem.



### 3.2.2 Activity Diagram

*Activity Diagram* atau Diagram Aktivitas adalah gambaran visual dari urutan alur kerja atau proses kegiatan di dalam sistem, dimulai dari tahap awal hingga mencapai tahap akhir. Selain itu, *Activity Diagram* juga berperan dalam mengilustrasikan komponen-komponen yang terdapat dalam Diagram Use Case. Gambar 3.4 menunjukkan *Activity Diagram* yang diterapkan dalam sistem:



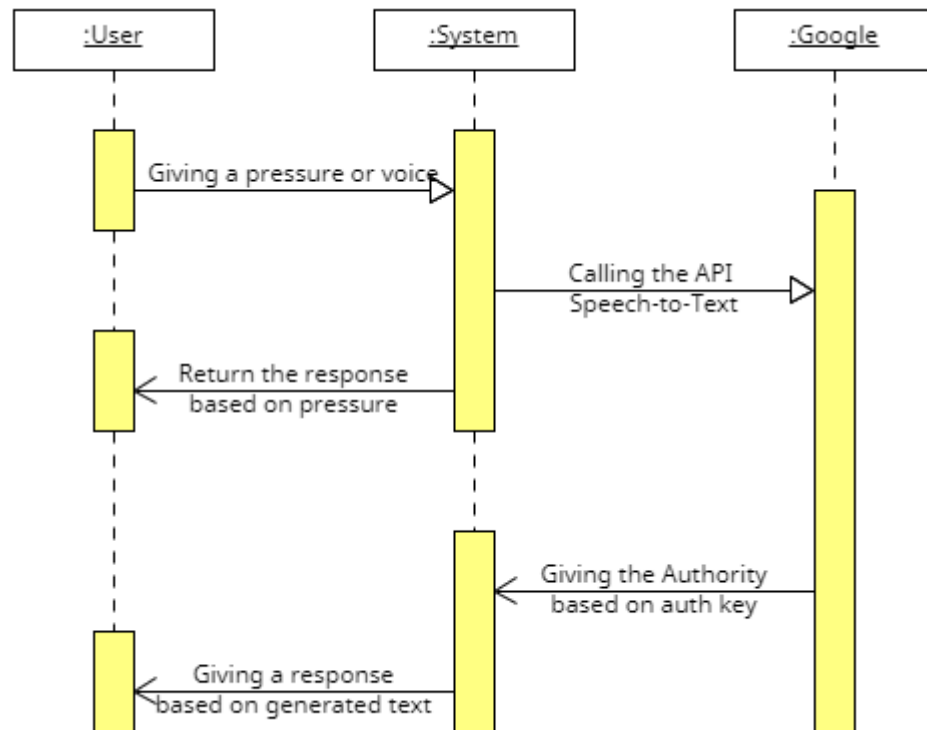
**Gambar 3.4** Activity Diagram

Gambar 3.4 menjelaskan dimana user memulai dengan memberikan tekanan maupun suara yang akan langsung diproses di *Analog to Digital Conversion* dan akan bergerak tergantung *input* yang diberikan *user* seperti suara maupun tekanan. Lalu setelah melewati kondisi yang ada maka akan diolah oleh *Digital to Analog Conversion*.

### 3.2.3 Sequence Diagram

*Sequence Diagram* atau Diagram Urutan merupakan salah satu jenis diagram dalam *Unified Modeling Language* (UML) yang berfungsi untuk memodelkan interaksi antara objek atau komponen dalam sebuah sistem sesuai dengan urutan waktu. Diagram ini sangat berguna untuk

memperlihatkan cara objek-objek saling berkomunikasi dalam konteks skenario atau proses tertentu. Pada Gambar 3.5 adalah *Sequence Diagram* yang diterapkan pada sistem:



**Gambar 3.5** *Sequence Diagram*

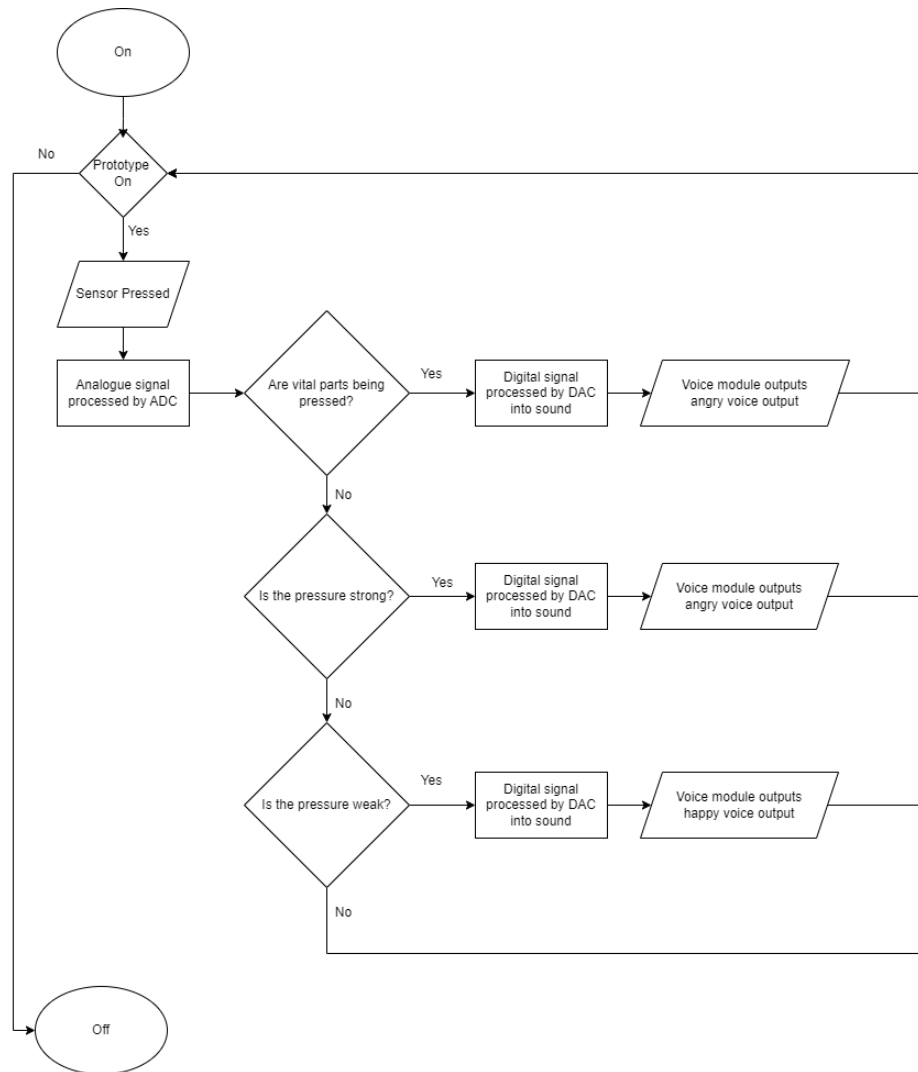
Pada *Sequence Diagram* Gambar 3.5, terdapat 3 penyusun yang saling berkomunikasi satu sama lain. Dari mulai menekan atau memberikan suara hingga mendapat respon yang sesuai.

### 3.3 Flowchart

Diagram Alir (*Flowchart*) adalah gambaran visual dari serangkaian langkah-langkah yang dilakukan oleh suatu algoritma secara sistematis. Tiap langkah dari proses ini diwakili oleh simbol khusus dan diberikan penjelasan untuk masing-masing tahapannya.

### 3.3.1 Flowchart Pressure Sensor

Pada Gambar 3.6 adalah *flowchart* dari *pressure sensor* yang digunakan:



**Gambar 3.6** Flowchart Pressure Sensor

Penjelasan Konsep Pengolahan Data ADC dan DAC :

Berdasarkan Gambar 3.6 Sinyal analog yang dikirim oleh sensor dikarenakan respon dari pengguna akan masuk dan dikirim ke ADC untuk diubah menjadi sinyal digital lalu akan diproses oleh DSP (*Digital Signal Processing*) untuk dikirim lagi ke DAC agar sinyal digital tersebut diubah kembali menjadi sinyal analog yang berbentuk sebagai suara.

Penjelasan *flowchart* berdasarkan Gambar 3.6:

1. *On* : Berfungsi sebagai start untuk menghidupkan *prototype*.
2. *Prototype on* : Kondisi Loop While yang dimana akan terus berjalan selama kondisi *prototype* hidup.
3. *Sensor Pressed* : Merupakan inputan sinyal analog dari sensor ke *microprocessor* arduino.
4. *Analogue signal processed by ADC* : Sinyal analog yang masuk diolah menjadi digital oleh arduino untuk diproses.
5. Kondisi (*Are vital parts being pressed?*) : Kondisi yang dimana apabila di titik tersebut ditekan akan mengeluarkan respon marah. Disini maksud dari titik tertentu merupakan titik vital yang ada di boneka.

Kondisi *YES* :

- a. *Digital signal processed by DAC into sound* : Sinyal digital yang masuk setelah diolah ADC akan diubah kembali menjadi sinyal analog yang berbentuk suara.
  - b. *Voice module outputs angry voice output* : Modul suara mengeluarkan hasil olahan dari DAC yang merupakan sinyal analog berbentuk suara.
6. Kondisi (*Is the pressure strong?*) : Kondisi dimana apabila titik sensor tersebut ditekan dengan kekuatan yang melebihi batas maka akan mengeluarkan respon marah.

Kondisi *YES* :

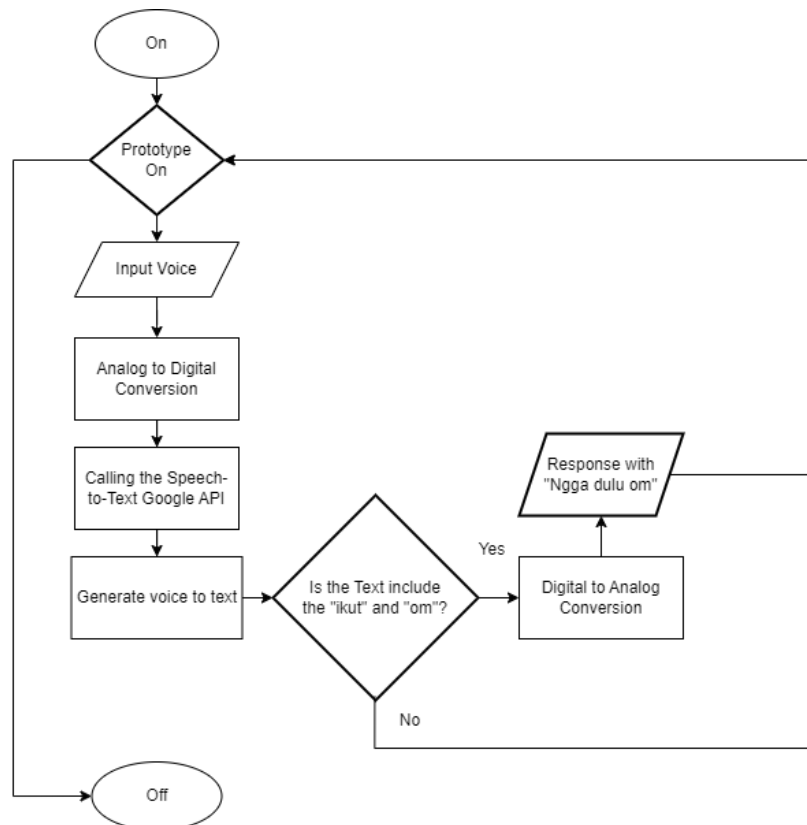
- a. *Digital signal processed by DAC into sound* : Sinyal digital yang masuk setelah diolah ADC akan diubah kembali menjadi sinyal analog yang berbentuk suara.
  - b. *Voice module outputs angry voice output* : Modul suara mengeluarkan hasil olahan dari DAC yang merupakan sinyal analog berbentuk suara.
7. Kondisi (*Is the pressure weak?*) : Kondisi dimana apabila titik sensor tersebut ditekan dengan kekuatan yang dibawah batas maka akan mengeluarkan respon senang.

Kondisi *YES* :

- a. *Digital signal processed by DAC into sound* : Sinyal digital yang masuk setelah diolah ADC akan diubah kembali menjadi sinyal analog yang berbentuk suara.
  - b. *Voice module outputs happy voice output* : Modul suara mengeluarkan hasil olahan dari DAC yang merupakan sinyal analog berbentuk suara.
8. *Off* : Berfungsi sebagai end untuk mematikan prototype.

### 3.3.2 Flowchart Voice Recognition

Berikut adalah *flowchart* sederhana dari voice recognition:



**Gambar 3.7** Flowchart Voice Recognition

Pada Gambar 3.7 dapat dilihat bahwa *voice recognition* memanggil *Speech-to-text* Google API untuk men-generate suara menjadi text lalu dilakukan proses agar merespon kata “ikut” dan “om” agar mengeluarkan suara “Ngga dulu om”.

## **BAB IV**

### **IMPLEMENTASI DAN PENGUJIAN SISTEM**

#### **4.1 Implementasi dan Hasil**

Pada penelitian ini, *prototype* yang akan dibangun memanfaatkan *Analog to Digital Conversion* dan *Digital to Analog Conversion*. *Prototype* yang dibangun berjalan menggunakan ESP-32 dan Arduino MEGA 2560. Penggunaan *voice recognition* menggunakan API Google *Speech-to-Text*. Penerapan tersebut menghasilkan respon yang sesuai dengan kondisi yang dibutuhkan selain dari sentuhan tekanan dan suara agar dapat menjadi dipresentasikan untuk edukasi.

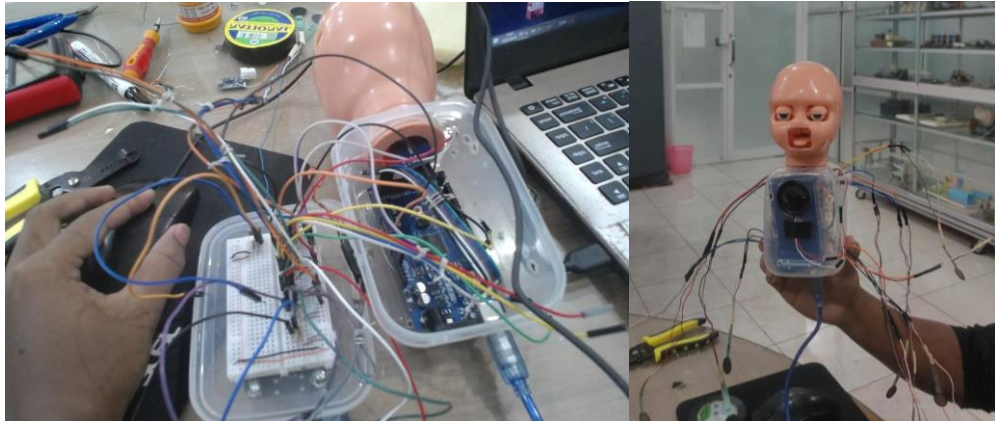
##### **4.1.1 Penggunaan Alat**

**Tabel 4.1** Tabel Alat yang Digunakan

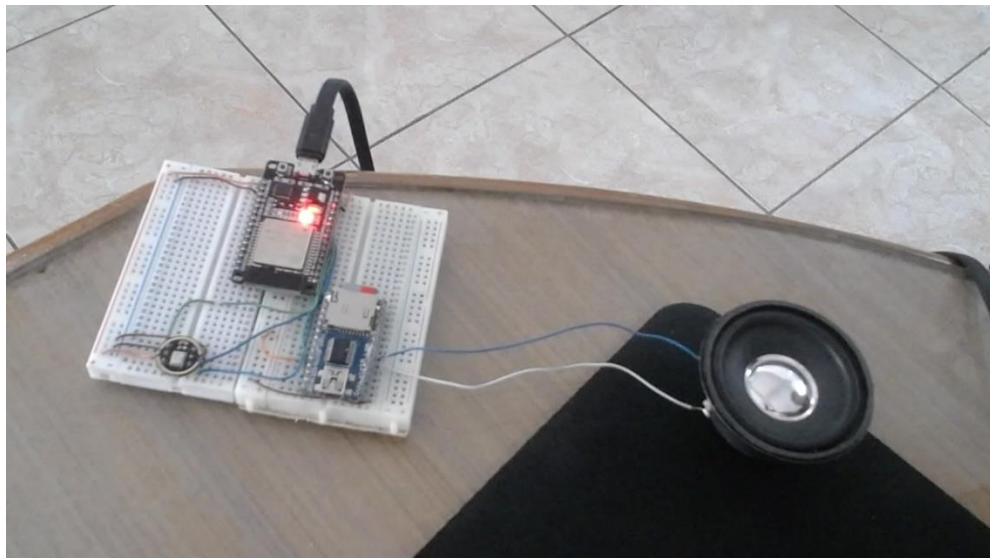
No	Nama Alat
1	Arduino MEGA 2560
2	ESP-WROOM-32
3	SD Card Module
4	Microphone (INMP441)
5	FSR ( <i>Force Sensitive Resistor</i> )
6	Speaker Module
7	Breadboard
8	N9200 MP3 Module

Pada tabel diatas merupakan alat alat yang digunakan ketika dilakukannya penelitian.

#### 4.1.2 Bentuk Desain Sistem



**Gambar 4.1** Tampilan Rangkaian *Prototype* Untuk Penggunaan Sensor FSR



**Gambar 4.2** Tampilan Rangkaian *Prototype* Untuk Penggunaan *Voice Recognition*

Pada Gambar 4.1 dan Gambar 4.2, dapat dilihat bahwa terdapat susunan dan sambungan dari beberapa perangkat yaitu ESP-32, Arduino UNO, dll. Susunan tersebut merupakan rangkaian dalam menyelesaikan penelitian **“Penerapan *Analog to Digital Conversion* dan *Digital to Analog Conversion* Pada *Microprocessor* Menggunakan Arduino Untuk *Body Awareness* Pada Anak Balita”**.

## 4.2 Pengujian Sistem

Tahap pengujian sistem dilakukan setelah berhasilnya tahap implementasi dengan tujuan untuk memverifikasi bahwa sistem yang telah dikembangkan mampu beroperasi dengan akurat dalam memproses dan memprediksi tingkat tekanan serta input suara. Pada tahap pengujian dilakukan pada 12 sensor dan 1 mikrofon. Untuk listing program berada pada bab lampiran.

### 4.2.1 Pengujian Program Pada FSR (*Force Sensitive Resistor*) dan *Output Suara*

Tahap awal dalam membangun prototype yaitu dengan melakukan proses pengecekan memori agar dapat digunakan untuk mengakses suara dalam format *.wav* di dalam *sd card* memori.

```

Initializing SD card...Wiring is correct and a card is present.

Card type:           SDHC
Clusters:            243680
Blocks x Cluster:    64
Total Blocks:        15595520

Volume type is:      FAT32
Volume size (Kb):    7797760
Volume size (Mb):    7615
Volume size (Gb):    7.44

Files found on the card (name, date and size in bytes):
SYSTEM~1/           2023-12-17 19:07:32
  INDEXE~1          2023-12-17 19:07:32 76
  WPSETT~1.DAT      2023-12-19 03:13:18 12
ANIMAL.WAV          2022-07-16 18:08:46 2434390
CAPSULE.WAV         2022-07-27 14:06:06 2797710
DONA.WAV            2022-07-16 17:36:02 3754446
SAKIT1A.WAV         2022-08-31 12:02:22 51814
SAKIT1B.WAV         2022-08-31 12:07:48 51758
SAKIT1C.WAV         2022-08-31 21:39:32 19598
SAKIT1N.WAV         2022-08-31 03:09:46 19620
SAKIT2A.WAV         2022-08-31 12:03:04 48294
SAKIT2B.WAV         2022-08-31 12:08:20 48294
SAKIT2C.WAV         2022-08-31 21:39:32 29998
SAKIT2N.WAV         2022-08-31 03:10:28 29998
SENANG1A.WAV        2022-08-31 12:02:22 75814
SENANG1B.WAV        2022-08-31 12:07:48 75758
SENANG1C.WAV        2022-08-31 21:39:32 59890
SENANG1N.WAV        2022-08-31 03:09:46 59890
SENANG2A.WAV        2022-08-31 12:02:22 61734
SENANG2B.WAV        2022-08-31 12:07:48 61678

```



SENANG2C.WAV	2022-08-31	21:40:00	34122
SENANG2N.WAV	2022-08-31	03:09:46	34122
SENANG3A.WAV	2022-08-31	12:03:04	69094
SENANG3B.WAV	2022-08-31	12:08:20	69038
SENANG3C.WAV	2022-08-31	21:40:00	35118
SENANG3N.WAV	2022-08-31	03:10:28	35156
VAM.WAV	2022-07-16	15:16:34	2881324
VITAL1A.WAV	2022-08-31	12:03:04	62374
VITAL1B.WAV	2022-08-31	12:08:20	62318
VITAL1C.WAV	2022-08-31	21:40:00	29678
VITAL1N.WAV	2022-08-31	03:10:28	29754

**Gambar 4.3** Tampilan Hasil Dari Pengecekan Isi *SD Card*

Pada Gambar 4.3 terlihat hasil dari program *example* yang tersedia dalam Arduino IDE dan terlihat bahwa kartu memori telah bisa diakses dan dapat dibaca keseluruhan datanya. Tahap berikutnya adalah menggabungkan program membaca *SD Card* dengan program untuk menjalankan musik.

```

1  #include "SD.h"
2  #define SD_ChipSelectPin 53
3  #include "TMRpcm.h"
4  #include "SPI.h"
5
6  TMRpcm tmrpcm;
7
8
9  void setup(void) {
10   tmrpcm.speakerPin = 11;
11   Serial.begin(9600);
12   if (!SD.begin(SD_ChipSelectPin)) {
13     Serial.println("SD fail");
14     return;
15   }
16   tmrpcm.setVolume(5);
17   Serial.println("Play Music");
18   tmrpcm.play("animal.wav");
19   Serial.println("Loop Music");
20 }

```

Output Serial Monitor x

Message (Enter to send message to 'Arduino Mega or Mega 2560' on 'COM5')

Play Music

Loop Music

**Gambar 4.4** Uji Coba Menjalankan Musik Dari Dalam *SD Card*

Berdasarkan Gambar 4.4 didapati bahwa program tersebut berjalan dengan baik sehingga suara berputar. Tahap berikutnya merupakan tahap mencoba menerapkan fsr sensor.

```

1
A1 No Touch
1
A2 No Touch
2
A3 No Touch
187
A4 Little bit Hard Touch
1
A5 No Touch
1
A6 No Touch
1
A7 No Touch
1
1
2
A10 No Touch
263
Vital Area Being Touched
1
A0 No Touch
0
A1 No Touch
0
A2 No Touch
0

```

**Gambar 4.5** Nilai Analog yang Keluar Ketika 2 Sensor FSR yang Berbeda Ditekan



**Gambar 4.6** Grafik Ketika Sensor Dalam Proses Ditekan

Pada Gambar 4.5 dan Gambar 4.6 diatas terlihat bahwa grafik ketika sensor FSR ditekan. Ketika sensor tersebut ditekan dia akan menyesuaikan dengan hambatan listrik pada analog tersebut.

**Tabel 4.2** Tabel *Output* Berdasarkan *Input* Nilai Analog

No	PIN	Nilai Analog	Output	No	PIN	Nilai Analog	Output	No	PIN	Nilai Analog	Output
1	A7	574	sakit2N.wav	31	A1	630	sakit2N.wav	61	A3	715	sakit2N.wav
2	A5	163	senang2N.wav	32	A2	319	sakit1N.wav	62	A0	689	sakit2N.wav
3	A3	408	sakit1N.wav	33	A5	430	sakit1N.wav	63	A4	392	sakit1N.wav
4	A0	72	senang2N.wav	34	A0	471	sakit1N.wav	64	A5	244	sakit1N.wav
5	A4	681	sakit2N.wav	35	A10	35	senang1N.wav	65	A10	105	senang2N.wav
6	A10	28	senang1N.wav	36	A3	722	sakit2N.wav	66	A6	756	sakit2N.wav
7	A2	729	sakit2N.wav	37	A4	614	sakit2N.wav	67	A1	311	sakit1N.wav
8	A6	350	sakit1N.wav	38	A6	28	senang1N.wav	68	A4	670	sakit2N.wav
9	A1	411	sakit1N.wav	39	A5	199	senang2N.wav	69	A7	123	senang2N.wav
10	A0	245	sakit1N.wav	40	A7	288	sakit1N.wav	70	A2	712	sakit2N.wav
11	A5	639	sakit2N.wav	41	A0	771	sakit2N.wav	71	A3	209	sakit1N.wav
12	A3	445	sakit1N.wav	42	A4	541	sakit2N.wav	72	A1	400	sakit1N.wav
13	A2	132	senang2N.wav	43	A1	750	sakit2N.wav	73	A0	87	senang2N.wav
14	A4	55	senang2N.wav	44	A10	63	senang2N.wav	74	A5	599	sakit2N.wav
15	A6	7	No Touch	45	A6	277	sakit1N.wav	75	A4	354	sakit1N.wav
16	A1	65	senang2N.wav	46	A3	86	senang2N.wav	76	A7	420	sakit1N.wav
17	A7	490	sakit1N.wav	47	A2	218	sakit1N.wav	77	A2	785	sakit2N.wav
18	A10	13	No Touch	48	A7	313	sakit1N.wav	78	A6	200	sakit1N.wav
19	A3	95	senang2N.wav	49	A5	389	sakit1N.wav	79	A1	701	sakit2N.wav
20	A4	120	senang2N.wav	50	A4	79	senang2N.wav	80	A4	800	sakit2N.wav
21	A5	561	sakit2N.wav	51	A1	403	sakit1N.wav	81	A3	15	No Touch
22	A6	370	sakit1N.wav	52	A2	476	sakit1N.wav	82	A5	737	sakit2N.wav
23	A1	49	senang1N.wav	53	A6	142	senang2N.wav	83	A0	182	senang2N.wav
24	A2	296	sakit1N.wav	54	A0	299	sakit1N.wav	84	A2	625	sakit2N.wav
25	A7	390	sakit1N.wav	55	A5	470	sakit1N.wav	85	A8	30	vital1N.wav
26	A10	312	sakit1N.wav	56	A4	552	sakit2N.wav	86	A9	345	vital1N.wav
27	A3	107	senang2N.wav	57	A1	216	sakit1N.wav	87	A11	59	vital1N.wav
28	A0	180	senang2N.wav	58	A6	677	sakit2N.wav				
29	A4	212	sakit1N.wav	59	A2	25	senang1N.wav				
30	A6	256	sakit1N.wav	60	A7	18	No Touch				

Pada Tabel 4.2 dapat dilihat hasil dari percobaan pada tiap pin dan mengeluarkan output sesuai dengan program yang telah dibuat. Terlihat juga bahwa persentase respon terhadap sinyal analog yang masuk mencapai 100% akurat berdasarkan logika yang sudah diterapkan.

#### 4.2.2 Pengujian Pada *Voice Recognition*

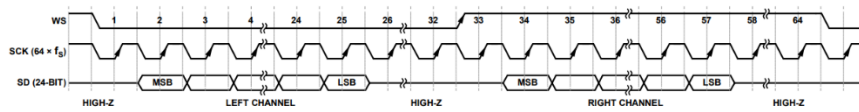


Figure 8. Stereo-Output I²S Format

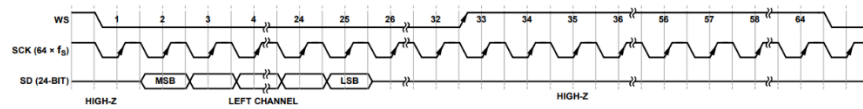


Figure 9. Mono-Output I²S Format Left Channel (L/R = 0)

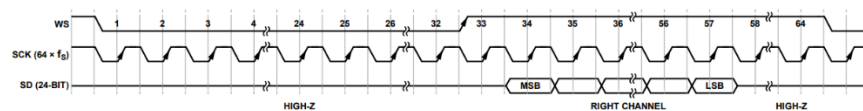
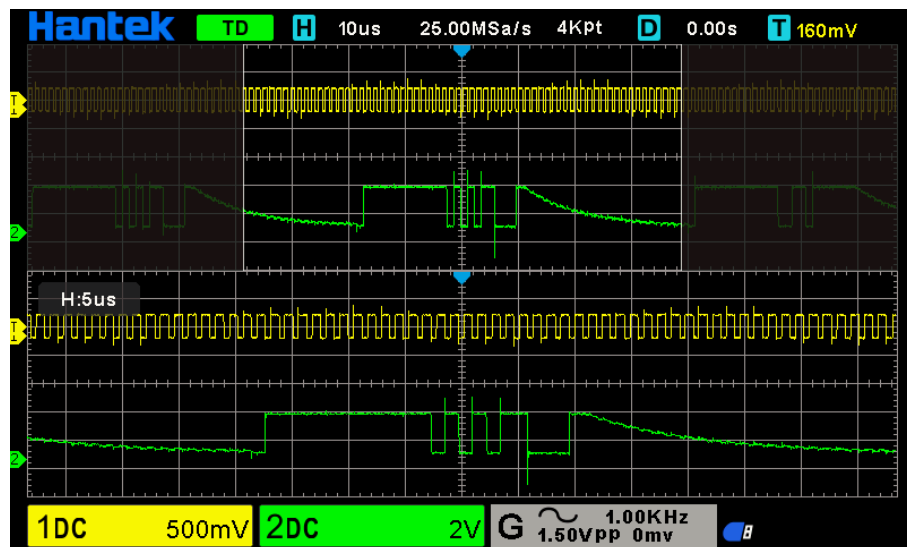


Figure 10. Mono-Output I²S Format Right Channel (L/R = 1)



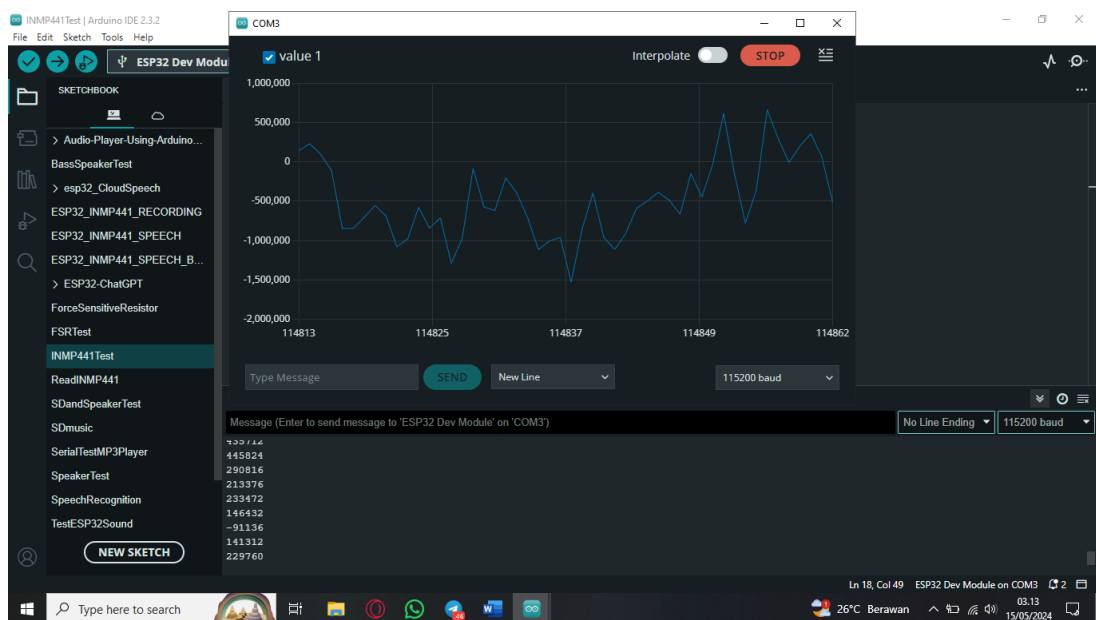
**Gambar 4.7** Pengecekan Sinyal *Microphone* INMP441 (Bawah)

Berdasarkan *Datasheet* (Atas)

Pada Gambar 4.7 dapat terlihat bahwa sinyal yang dihasilkan INMP441 dicek menggunakan osiloskop untuk dapat diketahui pada bit berapa sinyal analog yang masuk dengan tepat. Terlihat pada datasheet INMP441 bahwa alat tersebut mengolah sinyal analog suara dalam 64 bit dibagi menjadi 2 bagian. Untuk bagian dalam 32 bit pertama bisa dilihat untuk pembagian *left-channel* sedangkan 32 bit berikutnya adalah *right-channel*. Namun

penggunaan di tiap channel tidak menghabiskan 32 bit namun hanya menggunakan 24 bit saja.

Setelah didapati pada osiloskop dimana bahwa hanya *right-channel* yang berfungsi dikarenakan fungsi INMP441 yang dipakai adalah *mono* bukan *stereo*. Maka selanjutnya akan dilanjutkan dengan proses pengetesan INMP441.



**Gambar 4.8** Tampilan Sinyal Analog Yang Masuk Dari Mikrofon INMP441

Terlihat pada Gambar 4.8 hasil dari pengetesan menggunakan program uji coba perangkat mikrofon INMP441. Hasil coba menunjukkan bahwa mikrofon berjalan pada *right-channel* dari INMP441 itu sendiri. Tahapan selanjutnya adalah dengan uji coba perangkat N9200 MP3 Player Module.

The image shows a screenshot of an Arduino IDE. The top part displays the code for `SerialTestMP3Player.ino`. The code includes a `setup()` function that initializes the serial port at 9600 baud, prints "start", and writes two arrays of data to the serial port. The `loop()` function is empty. The bottom part shows the serial monitor output, which displays the words "start" and "end" on separate lines.

```

SerialTestMP3Player.ino
4 void setup()
5 {
6   //SerialController.begin( 9600 );
7   //SerialTFMini.begin( 9600, SERIAL_8N1, 27, 26 );
8
9   Serial.begin(115200);
10  Serial1.begin(9600, SERIAL_8N1,27,26); // open serial over TX and RX pins
11
12  Serial.println("start");
13  uint8_t vol20[6] = {0x06,0xF9,0x02,0x01,0x14,0x16};
14  uint8_t play[5] = {0x04,0xFB,0x01,0x01,0x01};
15
16  Serial1.write(vol20,6);
17  delay(1000);
18  Serial1.write(play,5);
19  Serial.println("end");
20 }
21
22 void loop()
23 {

```

Output Serial Monitor X

Message (Enter to send message to 'ESP32 Dev Module' on 'COM3')

```

load:0x3fff0030,len:1344
load:0x40078000,len:13964
load:0x40080400,len:3600
entry 0x400805f0
start
end

```

**Gambar 4.9** Uji Coba N9200 MP3 Player Module Menggunakan Komunikasi Serial

Berdasarkan Gambar 4.9 dapat dilihat bahwa *module* tersebut berjalan dengan menggunakan komunikasi serial yang dibuat berdasarkan dokumentasi dari modul. Tahapan berikutnya adalah pembuatan Token API Google *Speech-to-Text*.

### + CREATE CREDENTIALS

#### API key

Identifies your project using a simple API key to check quota and access

#### OAuth client ID

Requests user consent so your app can access the user's data

#### Service account

Enables server-to-server, app-level authentication using robot accounts

#### Help me choose

Asks a few questions to help you decide which type of credential to use


**Gambar 4.10** Tampilan Ketika Memilih *Credentials*

Pada gambar diatas merupakan tampilan dari halaman *credentials* di *dashboard* Google API. Untuk pemilihan *credentials*, silahkan pilih *Service Account*. Lalu bisa ke tahap selanjutnya.

**1 Service account details**

Service account name   
Display name for this service account

Service account ID \*  X ↺  
Email address: speech-to-text-969@dazzling-card-408207.iam.gserviceaccount.com

 Service account description   
Describe what this service account will do

[CREATE AND CONTINUE](#)

**Gambar 4.11** Tampilan Pengisian *Service Account Details*

Kemudian isilah nama akun dan deskripsi ketika mengisi *Service Account Details*. Lalu tekan *Create and Continue*.

**2 Grant this service account access to project (optional)**

Grant this service account access to My First Project so that it has permission to complete specific actions on the resources in your project. [Learn more](#)

Role  
Owner

IAM condition (optional) ?  
+ ADD IAM CONDITION

Full access to most Google Cloud resources. See the list of included permissions.  
  
+ ADD ANOTHER ROLE

CONTINUE

*Gambar 4.12 Tampilan Pengisian Service Account Access*

Terlihat pada gambar proses pengisian role untuk akses *Service Account*. Pengisian role yang diberikan merupakan owner. Lalu bisa ke tahap selanjutnya.

**3 Grant users access to this service account (optional)**

Grant access to users or groups that need to perform actions as this service account. [Learn more](#)

Service account users role ?  
Grant users the permissions to deploy jobs and VMs with this service account

Service account admins role ?  
Grant users the permission to administer this service account

DONE CANCEL

*Gambar 4.13 Tampilan Pengisian Akses User ke Service Account*



Tahapan terakhir merupakan pengisian akses user namun apabila tidak diperlukan bisa langsung klik *Done*.

API Keys

<input type="checkbox"/>	Name	Creation date ↓	Restrictions	Actions
<input type="checkbox"/>	✓ API key 1	Mar 26, 2024	Cloud Speech-to-Text API ...	<a href="#">SHOW KEY</a> ⋮

**Gambar 4.14** Pengambilan Token API

Pada Gambar 4.14 merupakan kolom untuk pengambilan token API Google yang telah dibuat. Tekan tulisan “*SHOW KEY*” untuk menunjukkan token yang telah dibuat lalu simpan untuk digunakan pada program *voice recognition*. Masuk ke tahapan selanjutnya yaitu uji coba program *voice recognition*.

```

47
304444
init i2s
Connecting to vivo Y17s
....
Connected! IP address: 192.168.167.189
64000
85393
2154
245456
***Allocate Buffer***
.....
2190
141008
***Start Recording***
.....
....
4200
141008
***Recording Finish***
126 284564 64000 270 -1056468 951827
5692
96320
Send Request

6367
83988
Wait for Reply

141
{

```

```

"results": [
  {
    "alternatives": [
      {
        "transcript": "ayo sini sama Om",
        "confidence": 0.9672866
      }
    ],
    "resultEndTime": "1.950s",
    "languageCode": "id-id"
  }
],
"totalBilledTime": "2s",
"requestId": "27813229855339720",
"usingLegacyModels": true
}
0
7114
95464
start
end

```

**Gambar 4.15** Tampilan Hasil dari Uji Coba Program *Voice Recognition*

Pada Gambar 4.15 terlihat hasil uji coba pada program *voice recognition*. Didapatkan hasil bahwa *speech-to-text* berjalan dengan normal. Terlihat pada akhir output terdapat tulisan “start” dan “end” yang dimana itu batas yang memproses bejalannya program output suara. Namun, respon dari *voice recognition* terhitung lambat dikarenakan faktor eksternal yang berasal dari pemanfaatan API Google. Terlihat juga pada Gambar 4.15 bahwa tingkat persentase keberhasilan *voice recognition* yang tertulis pada baris bertuliskan “*confidence*” sebesar 96%.

**Tabel 4.3** Tabel Input Untuk *Trigger* Output Suara

Input	Kondisi	Input	Kondisi
Ayo ikut Om pergi ke sana.	Aktif	Bisakah kita semua ikut Om?	Aktif
Om, mari ikut!	Aktif	Sini dan ikutlah dengan Om.	Aktif
Ayo kita ikut!	Tidak Aktif	Ikut Om atau tidak?	Aktif
Mari kita ikut Om bersama-sama.	Aktif	Om bilang ikut ke sini.	Aktif
Om sedang menunggu di sini, ayo!	Aktif	Saya tidak bisa ikut kali ini, maaf.	Tidak Aktif
Ayo kita ikut Om nanti.	Aktif	Om, saya akan di sini.	Tidak Aktif
Bolehkah saya ikut Om?	Aktif	Om, ikut saya ke sini.	Aktif
Om, ayo kita ikut bersama!	Aktif	Di sini kita bisa ikut bersama.	Aktif
Di sini ada Om yang ingin ikut.	Aktif	Om, ikut atau tidak?	Aktif
Saya akan ikut besok, ayo!	Tidak Aktif	Saya ikut saja.	Tidak Aktif
Om dan saya akan ke sini nanti.	Aktif	Om mengajak ikut serta, ayo!	Aktif
Ayo ikut ke sini sekarang.	Aktif	Mari ikut ke sini dan temui Om.	Aktif
Ikut Om adalah hal yang menyenangkan, ayo!	Aktif		

Pada Tabel 4.3 terlihat inputan apa saja yang dapat memancing output suara. Kalimat acak yang tak sesuai dengan tujuan sebagai ajakan dari orang dewasa ke anak masih dapat memancing alat agar mengeluarkan output suara. Kondisi aktif atau tidak aktif merupakan kondisi dimana alat mengeluarkan suara apabila aktif dan tidak mengeluarkan apapun ketika tidak aktif.

### 4.3 Tampilan Akhir Prototype



**Gambar 4.16** Bentuk Akhir Prototype

Pada Gambar 4.16 ditunjukkan bentuk akhir dari prototype yang dibangun yang terintegrasi sensor FSR pada bawah baju yang dikenakan boneka. Modul *voice recognition* menggunakan ESP-32 tidak bisa langsung diimplementasikan dikarenakan besarnya modul tersebut dan akan diimplementasikan apabila sudah dalam bentuk akhir yaitu PCB.

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Berikut adalah kesimpulan berdasarkan analisis, perancangan, dan pengujian pada penelitian “Penerapan *Analog to Digital Conversion* dan *Digital to Analog Conversion* Pada Microprocessor Menggunakan Arduino Untuk *Body Awareness* Pada Anak Balita”:

1. Penerapan *Analog to Digital Conversion* dan *Digital to Analog Conversion* pada Arduino dan ESP32 sangat baik sehingga *prototype* dapat terealisasi untuk mengajarkan balita tentang *body awareness*.
2. Penerapan ketika menggunakan nilai analog sebagai acuan lebih efektif untuk menentukan nilai yang akan disesuaikan dengan tingkat kekuatan pada anak balita. Nilai analog yang dikonversi dengan tepat juga membuat potongan data dari suara tidak hilang.
3. Penggunaan sensor FSR kecil kurang efisien dalam efektivitas penerapan pada boneka.
4. Penerapan Google API pada C++ untuk dijalankan langsung pada ESP-32 tanpa perangkat yang kompatibel akan memiliki banyak masalah dan juga respon *voice recognition* memiliki delay dikarenakan hambatan eksternal.
5. Tingkat keakuratan respon terhadap sensor FSR berdasarkan logika yang diterapkan sebesar 100% dan untuk keakuratan *voice recognition* sebesar 96%

#### **5.2 Saran**

Berikut merupakan saran yang dapat dijadikan sebagai bahan dalam pertimbangan untuk penelitian-penelitian berikutnya:

1. Menggunakan modul khusus untuk *voice recognition* apabila ingin menggunakan *voice recognition*.
2. Menyempurnakan perhitungan pada FSR dengan menggunakan sensor yang lebih besar.

3. Menambahkan AI agar lebih efisien ketika memasukkan *voice recognition* di dalamnya.
4. Lebih memperhatikan kompatibilitas perangkat ketika merangkai agar penggunaan sumber daya jadi tidak lebih boros.

## DAFTAR PUSTAKA

- Anggarawan, A., & Hadi, S., & Satria C., (2022). IoT-Based Garbage Container System Using NodeMCU ESP32 Microcontroller. *Journal of Advances in Information Technology* Vol. 13, No. 6.
- Arduino, S. A. (2015). *Arduino*. Arduino LLC, 372.
- Arar, S. (2021). An Introduction to Digital Signal Processing. *TECHNICAL ARTICLE*. Diakses 28 Juli 2023, dari <https://www.allaboutcircuits.com/technical-articles/an-introduction-to-digital-signal-processing/>.
- Barrett, S. (2022). *Arduino Microcontroller Processing for Everyone! Part II*. Springer Nature.
- Britannica, T. Editors of Encyclopaedia (2023, March 20). digital-to-analog conversion. *Encyclopedia Britannica*. <https://www.britannica.com/technology/digital-to-analog-conversion>.
- Dalmia, H., & Sinha, S. K. (2020). Analog to digital converters (ADC): A literature review. In *E3S Web of Conferences* (Vol. 184, p. 01025). EDP Sciences.
- Danylenko, S., Smelyakov, K., & Chupryna, A. (2022, October). Methods of Digital-To-Analog Conversion for Reproduction of Sound Waves. In *2022 IEEE 9th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T)* (pp. 43-48). IEEE.
- Elisabeth, N., Yulika, F., & Waspada, A. E. B. (2018). Desain komunikasi visual iklan layanan masyarakat tentang pelecehan seksual pada anak di kota medan. *ANDHARUPA: Jurnal Desain Komunikasi Visual & Multimedia*, 4(02), 188-195.
- Mulyani, M. A. (2016). Interpretasi dan Strategi Kesadaran Tubuh sebagai Konsep Memahami (Individu Interpretation and Body Awareness Strategies). *Wardah: Jurnal Dakwah dan Kemasyarakatan*, 17(1), 51-70.

- Mehling, W. E., Gopisetty, V., Daubenmier, J., Price, C. J., Hecht, F. M., & Stewart, A. (2009). Body awareness: construct and self-report measures. *PloS one*, 4(5), e5614.
- Naraharisetti, D., Karne, R., Weymouth, J., & Wijesinha, A. (2023, May). Obsolescence in Operating Systems and Microprocessors. In 2023 IEEE/ACIS 21st International Conference on Software Engineering Research, Management and Applications (SERA) (pp. 110-115). IEEE.
- Prijono, W. A. (2022). RANCANG BANGUN DIGITAL BANDPASS FILTER SEBAGAI APLIKASI SOUND ACTIVATION.
- Russell, D., Higgins, D., & Posso, A. (2020). Preventing child sexual abuse: A systematic review of interventions and their efficacy in developing countries. *Child abuse & neglect*, 102, 104395.
- Santos, M., Guilherme, J., Horta, N., Santos, M., Guilherme, J., & Horta, N. (2019). Logarithmic ADC. Logarithmic Voltage-to-Time Converter for Analog-to-Digital Signal Conversion, 37-58.
- Seal, B. N., & Meston, C. M. (2020). The impact of body awareness on women's sexual health: a comprehensive review. *Sexual medicine reviews*, 8(2), 242-255.
- Sinambela, B. R. O., Syahrizal, M., & Siregar, K. (2020). Penerapan Algoritma Start-Step-Stop Code Pada Kompresi File Audio. *KOMIK (Konferensi Nasional Teknologi Informasi dan Komputer)*, 4(1).
- Siswati, H. (2018). Lindungi Anak Dari Korban dan Pelaku Pelecehan Seksual. *Quantum: Jurnal Ilmiah Kesejahteraan Sosial*, 14(2), 121-133.
- Steiglitz, K. (2020). *Digital Signal Processing Primer*. Courier Dover Publications.
- Sugimoto, G. (2019). Introduction to Populating a Website with API Data. DOI:10.46430/phen0086
- Suhsmi, N. C., & Ismet, S. (2021). Materi Pendidikan Seks Bagi Anak Usia Dini. *Jurnal Golden Age*, 5(01), 164-174.



- Syam, E. (2014). Analisa dan Implementasi Transformasi Analog to Digital Converter (ADC) untuk Mengkonversi Suara Kebentuk Teks. Jurnal SATIN-Sains dan Teknologi Informasi, 3(2).
- Wang, L., Chen, W., Chen, K., He, R., & Zhou, W. (2022). The Research on the Signal Generation Method and Digital Pre-Processing Based on Time-Interleaved Digital-to-Analog Converter for Analog-to-Digital Converter Testing. Applied Sciences, 12(3), 1704.
- Webb, C. (2021). Microprocessor advances and the mainframe legacy. IEEE Micro, 41(6), 68-70.
- Yang, S., Wang, J., Yang, B., Chi, H., Ou, J., Zhai, Y., & Li, Q. (2022). A serial digital-to-analog conversion based on photonic time-stretch technology. Optics Communications, 510, 127949.

## LAMPIRAN

## LISTING PROGRAM

## 1. ForceSensitiveResistor.io

```
#include "SD.h"
#define SD_ChipSelectPin 53
#include "TMRpcm.h"
#include "SPI.h"

TMRpcm tmrpcm;

int fsrPin0 = A0;
int fsrPin1 = A1;
int fsrPin2 = A2;
int fsrPin3 = A3;
int fsrPin4 = A4;
int fsrPin5 = A5;
int fsrPin6 = A6;
int fsrPin7 = A7;
int fsrPin8 = A8;
int fsrPin9 = A9;
int fsrPin10 = A10;
int fsrPin11 = A11;
// FSR TERKONEKSI DENGAN RESISTOR 10K
int fsrReading0,
    fsrReading1,
    fsrReading2,
    fsrReading3,
    fsrReading4,
    fsrReading5,
    fsrReading6,
    fsrReading7,
    fsrReading8,
    fsrReading9,
    fsrReading10,
    fsrReading11;

void setup(void) {
    tmrpcm.speakerPin = 11;
    Serial.begin(9600);
    if (!SD.begin(SD_ChipSelectPin)) {
        Serial.println("SD fail");
        return;
    }
    tmrpcm.setVolume(6);
}

void loop(void) {
```

```

fsrReading0 = analogRead(fsrPin0)-90;
fsrReading1 = analogRead(fsrPin1)-90;
fsrReading2 = analogRead(fsrPin2)-90;
fsrReading3 = analogRead(fsrPin3)-90;
fsrReading4 = analogRead(fsrPin4)-90;
fsrReading5 = analogRead(fsrPin5)-90;
fsrReading6 = analogRead(fsrPin6)-90;
fsrReading7 = analogRead(fsrPin7)-90;
fsrReading8 = analogRead(fsrPin8)-90;
fsrReading9 = analogRead(fsrPin9)-90;
fsrReading10 = analogRead(fsrPin10)-90;
fsrReading11 = analogRead(fsrPin11)-90;

Serial.println(fsrReading0);
if (fsrReading0 < 20) {
    Serial.println("A0 No Touch");
} else if (fsrReading0 < 50) {
    tmrpcm.play("senang1N.wav");
    Serial.println("A0 Gentle Touch");
} else if (fsrReading0 < 200) {
    tmrpcm.play("senang2N.wav");
    Serial.println("A0 Little bit Hard Touch");
} else if (fsrReading0 < 500) {
    tmrpcm.play("sakit1N.wav");
    Serial.println("A0 Medium Touch");
} else if (fsrReading0 < 800) {
    tmrpcm.play("sakit2N.wav");
    Serial.println("A0 Hard Touch");
} else {
    Serial.println("There is a bug");
}

Serial.println(fsrReading1);
if (fsrReading1 < 20) {
    Serial.println("A1 No Touch");
} else if (fsrReading1 < 50) {
    tmrpcm.play("senang1N.wav");
    Serial.println("A1 Gentle Touch");
} else if (fsrReading1 < 200) {
    tmrpcm.play("senang2N.wav");
    Serial.println("A1 Little bit Hard Touch");
} else if (fsrReading1 < 500) {
    tmrpcm.play("sakit1N.wav");
    Serial.println("A1 Medium Touch");
} else if (fsrReading1 < 800) {
    tmrpcm.play("sakit2N.wav");
    Serial.println("A1 Hard Touch");
} else {
    Serial.println("There is a bug");
}

```

```

}

Serial.println(fsrReading2);
if (fsrReading2 < 20) {
    Serial.println("A2 No Touch");
} else if (fsrReading2 < 50) {
    tmrpcm.play("senang1N.wav");
    Serial.println("A2 Gentle Touch");
} else if (fsrReading2 < 200) {
    tmrpcm.play("senang2N.wav");
    Serial.println("A2 Little bit Hard Touch");
} else if (fsrReading2 < 500) {
    tmrpcm.play("sakit1N.wav");
    Serial.println("A2 Medium Touch");
} else if (fsrReading2 < 800) {
    tmrpcm.play("sakit2N.wav");
    Serial.println("A2 Hard Touch");
} else {
    Serial.println("There is a bug");
}

Serial.println(fsrReading3);
if (fsrReading3 < 20) {
    Serial.println("A3 No Touch");
} else if (fsrReading3 < 50) {
    tmrpcm.play("senang1N.wav");
    Serial.println("A3 Gentle Touch");
} else if (fsrReading3 < 200) {
    tmrpcm.play("senang2N.wav");
    Serial.println("A3 Little bit Hard Touch");
} else if (fsrReading3 < 500) {
    tmrpcm.play("sakit1N.wav");
    Serial.println("A3 Medium Touch");
} else if (fsrReading3 < 800) {
    tmrpcm.play("sakit2N.wav");
    Serial.println("A3 Hard Touch");
} else {
    Serial.println("There is a bug");
}

Serial.println(fsrReading4);
if (fsrReading4 < 20) {
    Serial.println("A4 No Touch");
} else if (fsrReading4 < 50) {
    tmrpcm.play("senang1N.wav");
    Serial.println("A4 Gentle Touch");
} else if (fsrReading4 < 200) {
    tmrpcm.play("senang2N.wav");
    Serial.println("A4 Little bit Hard Touch");
}

```

```

} else if (fsrReading4 < 500) {
    tmrpcm.play("sakit1N.wav");
    Serial.println("A4 Medium Touch");
} else if (fsrReading4 < 800) {
    tmrpcm.play("sakit2N.wav");
    Serial.println("A4 Hard Touch");
} else {
    Serial.println("There is a bug");
}

Serial.println(fsrReading5);
if (fsrReading5 < 20) {
    Serial.println("A5 No Touch");
} else if (fsrReading5 < 50) {
    tmrpcm.play("senang1N.wav");
    Serial.println("A5 Gentle Touch");
} else if (fsrReading5 < 200) {
    tmrpcm.play("senang2N.wav");
    Serial.println("A5 Little bit Hard Touch");
} else if (fsrReading5 < 500) {
    tmrpcm.play("sakit1N.wav");
    Serial.println("A5 Medium Touch");
} else if (fsrReading5 < 800) {
    tmrpcm.play("sakit2N.wav");
    Serial.println("A5 Hard Touch");
} else {
    Serial.println("There is a bug");
}

Serial.println(fsrReading6);
if (fsrReading6 < 20) {
    Serial.println("A6 No Touch");
} else if (fsrReading6 < 50) {
    tmrpcm.play("senang1N.wav");
    Serial.println("A6 Gentle Touch");
} else if (fsrReading6 < 200) {
    tmrpcm.play("senang2N.wav");
    Serial.println("A6 Little bit Hard Touch");
} else if (fsrReading6 < 500) {
    tmrpcm.play("sakit1N.wav");
    Serial.println("A6 Medium Touch");
} else if (fsrReading6 < 800) {
    tmrpcm.play("sakit2N.wav");
    Serial.println("A6 Hard Touch");
} else {
    Serial.println("There is a bug");
}

Serial.println(fsrReading7);

```

```

if (fsrReading7 < 20) {
    Serial.println("A7 No Touch");
} else if (fsrReading7 < 50) {
    tmrpcm.play("senang1N.wav");
    Serial.println("A7 Gentle Touch");
} else if (fsrReading7 < 200) {
    tmrpcm.play("senang2N.wav");
    Serial.println("A7 Little bit Hard Touch");
} else if (fsrReading7 < 500) {
    tmrpcm.play("sakit1N.wav");
    Serial.println("A7 Medium Touch");
} else if (fsrReading7 < 800) {
    tmrpcm.play("sakit2N.wav");
    Serial.println("A7 Hard Touch");
} else {
    Serial.println("There is a bug");
}

Serial.println(fsrReading8);
if (fsrReading8 > 30) {
    tmrpcm.play("vital1N.wav");
    Serial.println("Vital Area Being Touched");
}

Serial.println(fsrReading9);
if (fsrReading9 > 30) {
    tmrpcm.play("vital1N.wav");
    Serial.println("Vital Area Being Touched");
}

Serial.println(fsrReading10);
if (fsrReading10 < 20) {
    Serial.println("A10 No Touch");
} else if (fsrReading10 < 50) {
    tmrpcm.play("senang1N.wav");
    Serial.println("A10 Gentle Touch");
} else if (fsrReading10 < 200) {
    tmrpcm.play("senang2N.wav");
    Serial.println("A10 Little bit Hard Touch");
} else if (fsrReading10 < 500) {
    tmrpcm.play("sakit1N.wav");
    Serial.println("A10 Medium Touch");
} else if (fsrReading10 < 800) {
    tmrpcm.play("sakit2N.wav");
    Serial.println("A10 Hard Touch");
} else {
    Serial.println("There is a bug");
}

```

```
Serial.println(fsrReading11);  
if (fsrReading11 > 30) {  
    tmrpcm.play("vital1N.wav");  
    Serial.println("Vital Area Being Touched");  
}  
delay(1000);  
}
```

## 2. ESP32\_INMP441\_SPEECH.ino

```
#include <driver/i2s.h>
#include <WiFiClientSecure.h>
#include "network_param.h"

#define I2S_WS 25
#define I2S_SD 33
#define I2S_SCK 32
#define I2S_PORT I2S_NUM_0
#define I2S_SAMPLE_RATE (16000)
#define I2S_SAMPLE_BITS (16) // 24
#define I2S_CHANNEL_BITS (32)
#define RECORD_TIME (2) //Seconds
#define I2S_CHANNEL_NUM (1)
#define FLASH_RECORD_SIZE (I2S_CHANNEL_NUM * I2S_SAMPLE_RATE *
I2S_SAMPLE_BITS / 8 * RECORD_TIME) // 1 * 16000 * 2 * 2 = 64.000 bytes
#define HEADER_SIZE 44
#define HEADER_SIZE_2BYTES 48
#define HEADER_SIZE_BASE64 60
#define RECORD_STORAGE_SIZE ((FLASH_RECORD_SIZE * 4 / 3) +
HEADER_SIZE_BASE64)

WiFiClientSecure client;

void i2sInit() {
    i2s_config_t i2s_config = {
        .mode = (i2s_mode_t)(I2S_MODE_MASTER | I2S_MODE_RX),
        .sample_rate = I2S_SAMPLE_RATE,
        .bits_per_sample =
i2s_bits_per_sample_t(I2S_CHANNEL_BITS), //I2S_BITS_PER_SAMPLE_24BIT,//I
2S_BITS_PER_SAMPLE_16BIT,
        .channel_format =
I2S_CHANNEL_FMT_RIGHT_LEFT, //I2S_CHANNEL_FMT_ONLY_LEFT,
        .communication_format = I2S_COMM_FORMAT_STAND_I2S,
        .intr_alloc_flags = 0,
        .dma_buf_count = 2,
        .dma_buf_len = 1024,
        .use_apll = 1,
        //.bits_per_chan = i2s_bits_per_chan_t(I2S_CHANNEL_BITS)
    };
    i2s_driver_install(I2S_PORT, &i2s_config, 0, NULL);

    const i2s_pin_config_t pin_config = {
        .bck_io_num = I2S_SCK,
        .ws_io_num = I2S_WS,
        .data_out_num = -1,
        .data_in_num = I2S_SD
    };
    i2s_set_pin(I2S_PORT, &pin_config);
```



```

}

void wavHeader(byte* header, int wavSize) {
    // header plain binary

    // 16 bit format
    header[0] = 'R';
    header[1] = 'I';
    header[2] = 'F';
    header[3] = 'F';
    unsigned int fileSize = wavSize + HEADER_SIZE - 8; // 2 detik ==>
    16.000 * 2 bytes * 2 detik = 64.036 ==> 00 00 FA 00
    header[4] = (byte)(fileSize & 0xFF); // 24
    header[5] = (byte)((fileSize >> 8) & 0xFF); // FA
    header[6] = (byte)((fileSize >> 16) & 0xFF); // 00
    header[7] = (byte)((fileSize >> 24) & 0xFF); // 00
    header[8] = 'W';
    header[9] = 'A';
    header[10] = 'V';
    header[11] = 'E';
    header[12] = 'f';
    header[13] = 'm';
    header[14] = 't';
    header[15] = ' ';
    header[16] = 0x10; // Length of format data as listed above
    header[17] = 0x00;
    header[18] = 0x00;
    header[19] = 0x00;
    header[20] = 0x01; // Type of format (1 is PCM) - 2 byte integer
    header[21] = 0x00;
    header[22] = 0x01; // Number of Channels => 1
    header[23] = 0x00;
    header[24] = 0x80; // Sample Rate - 32 byte integer 0x3E80 = 16000
    header[25] = 0x3E;
    header[26] = 0x00;
    header[27] = 0x00;
    header[28] = 0x00; //0x00; // (Sample Rate * BitsPerSample * Channels)
    / 8 ==> 0x7D00 = 32000 <= 16.000 * 16 * 1 / 8
    header[29] = 0x7D; //0x7D;
    header[30] = 0x00;
    header[31] = 0x00;
    header[32] = 0x02; //0x02; // (BitsPerSample * Channels) / 8) => 16 *
    1 / 8 = 2 ==> 16 bit mono or 8 bit stereo
    header[33] = 0x00;
    header[34] = 0x10; //0x10; // Bits per sample ==> 0x10 = 16
    header[35] = 0x00;
    header[36] = 'd';
    header[37] = 'a';
    header[38] = 't';
}

```

```

header[39] = 'a';
header[40] = (byte)(wavSize & 0xFF);           // 00
header[41] = (byte)((wavSize >> 8) & 0xFF);    // FA
header[42] = (byte)((wavSize >> 16) & 0xFF);   // 00
header[43] = (byte)((wavSize >> 24) & 0xFF);   // 00
}

static const char b64_table[] = {
'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H',
'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P',
'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X',
'Y', 'Z', 'a', 'b', 'c', 'd', 'e', 'f',
'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n',
'o', 'p', 'q', 'r', 's', 't', 'u', 'v',
'w', 'x', 'y', 'z', '0', '1', '2', '3',
'4', '5', '6', '7', '8', '9', '+', '/'
};

// base64 encode from 3 byte src buffer to 4 byte dst buffer
void bs64_enc34(unsigned char* src, unsigned char* dst, size_t len) {
    unsigned char tmp[3];
    uint8_t i = 0;
    uint16_t spos=0, dpos = 0; // size = 0, num = 0;
    while(len)
    {
        // read up to 3 bytes at a time into `tmp`
        tmp[i] = src[spos+i];
        i++;
        // if 3 bytes read then encode into destination
        if (3 == i) {
            dst[dpos] = b64_table[tmp[0] >> 2];
            dst[dpos+1] = b64_table[((tmp[0] & 0x03) << 4) + (tmp[1] >> 4)];
            dst[dpos+2] = b64_table[((tmp[1] & 0x0f) << 2) + (tmp[2] >> 6)];
            dst[dpos+3] = b64_table[tmp[2] & 0x3f];
            spos += 3;
            dpos += 4;
            i = 0;
        }
        len--;
    }
    //Serial.printf("%d-%d ",i,len);
    if (i > 0) {
        dst[dpos] = b64_table[tmp[0] >> 2];
        if (i == 1) {
            dst[dpos+1] = b64_table[(tmp[0] & 0x03) << 4];
            dst[dpos+2] = '=';
        } else { // if counts == 2
            dst[dpos+1] = b64_table[((tmp[0] & 0x03) << 4) + (tmp[1] >> 4)];
            dst[dpos+2] = b64_table[(tmp[1] & 0x0f) << 2];
        }
    }
}

```

```

    }
    dst[dpos+3] = '=';
}
}

bool connstate = 0;
void setup() {
    // put your setup code here, to run once:
    const char* ssid = "PUT YOUR OWN WIFI NAME"; const char* password =
"PUT YOUR OWN PASSWORD";
    Serial.begin(115200);
    Serial.println(millis());
    Serial.println(ESP.getFreeHeap());

    Serial.println(F("init i2s"));
    i2sInit();
    //for (int i = 0; i < 5; i++) {
        //i2s_read(I2S_NUM_0, test_data, 2048, &i2slen, portMAX_DELAY);
    //}

    Serial.printf("Connecting to %s\n", ssid);
    if (String(WiFi.SSID()) != String(ssid)) {
        WiFi.mode(WIFI_STA);
        WiFi.begin(ssid, password);
    }
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }

    Serial.println();
    Serial.print(F("Connected! IP address: "));
    Serial.println(WiFi.localIP());
}

// transport len 960 become 1280 base64 data
#define TRANSPORT_LENGTH 960 // 768 // 960 // 1056 // 1152
#define TRANSPORT_LENGTH_B64 (TRANSPORT_LENGTH * 4 / 3)
#define SEGMENT_LENGTH 2160 // 2048
#define BUFFER_COUNT 48

void print_http_gspeech(uint8_t ** audio, uint32_t conlen) {
    String HttpBody1 =
"{"config":{"encoding":"LINEAR16","sampleRateHertz":16000,"languageCode":{"id-ID"},"audio":{"content":""";
    String HttpBody3 = "}}\r\n\r\n";
    //int httpBody2Length = (audio->wavDataSize + sizeof(audio-
>paddedHeader)) * 4 / 3; // 4/3 is from base64 encoding
    String ContentLength = String(HttpBody1.length() + conlen*4/3 +
HttpBody3.length());

```

```

String HttpHeader;
// if (authentication == USE_APIKEY)
HttpHeader = String("POST /v1/speech:recognize?key=") + ApiKey
              + String(" HTTP/1.1\r\nHost:
speech.googleapis.com\r\nContent-Type: application/json\r\nContent-
Length: ") + ContentLength + String("\r\n\r\n");

int idx = 0, pos = 0, lap = 0, len, iter = 0;
client.print(HttpHeader);
client.print(HttpBody1);
//Serial.println(conlen);
uint8_t httpdata[TRANSPORT_LENGTH_B64];
while(conlen){
    if(conlen < TRANSPORT_LENGTH){
        len = conlen;
    } else {
        len = TRANSPORT_LENGTH;
    }

    if(pos + len <= SEGMENT_LENGTH){
        bs64_enc34(audio[idx]+pos, httpdata, len);
        if(pos + len == SEGMENT_LENGTH){
            pos = 0;
            idx++;
        } else {
            pos += len;
        }
    } else {
        lap = 0;
        bs64_enc34(audio[idx]+pos, httpdata, SEGMENT_LENGTH - pos); // 2048
- 1536 = 512
        idx++;
        lap += (SEGMENT_LENGTH - pos) * 4 / 3;
        bs64_enc34(audio[idx], httpdata+lap, len + pos - SEGMENT_LENGTH);
// 768 + 1536 - 2048 = 256
        pos += len - SEGMENT_LENGTH;
    }

    client.write(httpdata, len*4/3);
    conlen -= len;
}
//client.write(audio,conlen);
client.print(HttpBody3);
}

void loop() {
    Serial.println(FLASH_RECORD_SIZE);
    Serial.println(RECORD_STORAGE_SIZE);
    Serial.println(millis());
}

```

```

Serial.println(ESP.getFreeHeap());
uint8_t i2s_data[SEGMENT_LENGTH];
uint32_t i2slen, record_size, inavail = 0, noavail = 0;
const char* server = "speech.googleapis.com";
uint16_t i, curpos, actlen;
uint8_t arrpos; //byte0, byte1, byte2, byte3, tmprev[2];
int32_t maxd = 0, mind = 0;
//WiFiClientSecure client;
for (i = 0; i < 5; i++) {
    i2s_read(I2S_NUM_0, i2s_data, SEGMENT_LENGTH, &i2slen,
portMAX_DELAY);
}

client.setCACert(root_ca);

Serial.println(F("***Allocate Buffer***"));
uint8_t *recdata[BUFFER_COUNT];
for(i=0;i<BUFFER_COUNT;i++){
    recdata[i] = (uint8_t *)calloc(SEGMENT_LENGTH,1);
    if(recdata[i] == NULL) {
        Serial.print(F("Failed Buffer ")); Serial.println(i);
        while(1);
    }
    Serial.print('.');
}
Serial.println();
Serial.println(millis());
Serial.println(ESP.getFreeHeap());

wavHeader(recdata[0], FLASH_RECORD_SIZE);

while (1) {
    record_size = 0;
    arrpos = 0;
    curpos = 44; //48; // 56; // 60;
    //flash_wrsz = HEADER_SIZE;
    if (true) { // button press conditional if any
        Serial.println(F("***Start Recording***"));
        while (record_size < FLASH_RECORD_SIZE) {
            //read data from I2S bus of INMP441
            i2slen = 0;
            i2s_read(I2S_NUM_0, i2s_data, SEGMENT_LENGTH, &i2slen, 0);
            if (i2slen) {
                if ((i2slen/8)*2 + record_size > FLASH_RECORD_SIZE) {
                    i2slen = ((FLASH_RECORD_SIZE - record_size)/2) * 8;
                }
                actlen = 0;
                for (i = 0; i < i2slen; i = i + 8) {

```

```

        int32_t tmpd = (((int32_t)(int8_t)i2s_data[i+3])) << 17) |
        (((uint32_t)i2s_data[i+2]) << 9) | (((uint32_t)i2s_data[i+1]) << 1) |
        (((uint32_t)i2s_data[i]) >> 7);
        if(tmpd > maxd){
            maxd = tmpd;
        } else if (tmpd < mind) {
            mind = tmpd;
        }
        tmpd = tmpd / 64;
        recdata[arrpos][curpos] = ((uint8_t) (tmpd & 0xFF));
        recdata[arrpos][curpos+1] = (uint8_t) (tmpd >> 8);
        curpos += 2;
        if(curpos == SEGMENT_LENGTH){
            curpos = 0;
            arrpos++;
        }
        actlen += 2; // 3 6
    }
    Serial.print('.');
    record_size += actlen;
    //Serial.printf("%d %d %d %d
%d\n",i2slen,actlen,record_size,arrpos,curpos);
    inavail++;
} else {
    noavail++;
}
}

Serial.println();
Serial.println(millis());
Serial.println(ESP.getFreeHeap());

Serial.println(F("***Recording Finish***"));
Serial.printf("%d %d %d %d %d %d\n", inavail, noavail, record_size,
actlen, mind, maxd);
if (client.connect(server, 443)) {
    connstate = 1;
} else {
    connstate = 1;
    Serial.println("Connection failed!");
    while(1);
}
Serial.println(millis());
Serial.println(ESP.getFreeHeap());
Serial.println(F("Send Request"));
print_http_gspeech(recdata,FLASH_RECORD_SIZE + HEADER_SIZE);
Serial.println();
String text_return = "";
Serial.println(millis());

```

```

Serial.println(ESP.getFreeHeap());
Serial.println(F("Wait for Reply"));
while (!client.available());

while (client.available())
{
    char temp = client.read();
    text_return += temp;
}
Serial.println();
Serial.println(text_return);
Serial.println(millis());
Serial.println(ESP.getFreeHeap());
if ((text_return.indexOf("ikut") != -1 && text_return.indexOf("Om") != -
1) || (text_return.indexOf("sini") != -1 && text_return.indexOf("ikut") != -1
|| text_return.indexOf("Om") != -1)) {
    Serial1.begin(9600, SERIAL_8N1,27,26); // open serial over TX and RX
pins
    Serial.println("start");
    uint8_t vol20[6] = {0x06,0xF9,0x02,0x01,0x14,0x16};
    //uint8_t play[5] = {0x04,0xFB,0x01,0x01,0x01};
    //uint8_t play[7] = {0x04,0xFB,0x03,0x06,0x00,0x01,0x09};
    uint8_t play[7] = {0x04,0xFB,0x03,0x06,0x00,0x02,0x0A};
    Serial1.write(vol20,6);
    delay(1000);
    Serial1.write(play,7);
    Serial.println("end");
}
else if ((text_return.indexOf("om") != -1 &&
text_return.indexOf("makanan") != -1) || (text_return.indexOf("sini") != -1 &&
text_return.indexOf("permen") != -1)) {
    Serial1.begin(9600, SERIAL_8N1,27,26); // open serial over TX and RX
pins
    Serial.println("start");
    uint8_t vol20[6] = {0x06,0xF9,0x02,0x01,0x14,0x16};
    //uint8_t play[5] = {0x04,0xFB,0x01,0x01,0x01};
    uint8_t play[7] = {0x04,0xFB,0x03,0x06,0x00,0x01,0x09};
    //uint8_t play[7] = {0x04,0xFB,0x03,0x06,0x00,0x02,0x0A};
    Serial1.write(vol20,6);
    delay(1000);
    Serial1.write(play,7);
    Serial.println("end");
}
while(1);
} // if (true)
} // while (1)
}

```

## CURRICULUM VITAE

### Rheza Yudhistira Ramadhana

Medan, 16 November 2002

Jl. Asoka 1 Gang Mekar No. 10, Kelurahan Asam Kumbang, Kecamatan Medan Selayang

+62 822 1398 3749 | rhezaramadhana@gmail.com

Linkedin : Rheza Yudhistira Ramadhana



#### PENDIDIKAN :

##### Perguruan Tinggi

Universitas Sumatera Utara

Fakultas Ilmu Komputer dan Teknologi Informasi

Jurusan Ilmu Komputer

2020 - Sekarang

IPK 3,82 / 4,00.

##### Sekolah Menengah Kejuruan

SMK Telkom Sandhy Putra Medan

Jurusan Teknik Komputer dan Jaringan

2017- 2020

#### KEMAMPUAN KOMPUTER

Pemrograman: C/C++, JavaScript, Pascal, HTML, CSS.

Database : MySQL.

IDE : Visual Studio Code, DevC++.

OS : Debian, Ubuntu, Windows.

Software : Microsoft Office, Adobe Premiere

Pro, Adobe Photoshop, CorelDraw, Autodesk, Winbox, VMWare.

#### PROYEK

##### Data Kebun Binatang

2020

- Membuat program menggunakan Pascal yang merupakan tugas dari Lab 3 Konsep Dasar Pemrograman .

##### Kiddo Web

2020-2021

- Membangun website menggunakan bahasa pemrograman PHP, HTML, JavaScript, dan *database* MySQL untuk proyek akhir semester Lab 3 Pemrograman Web 1.

##### Brenda : Body Awareness Education Smart Doll

21 Maret - 1 Oktober 2022

- Membuat alat peraga yang mengajarkan anak balita tentang pentingnya pemahaman tubuh sejak dini yang dibangun menggunakan Arduino pada kegiatan PKM Nasional di bidang PKM-KC Tahun 2022.

#### PENGALAMAN KERJA

##### Praktik Kerja Lapangan di PT. 2K123 | Divisi Game dan Pengawas Server

Juni 2019 - Agustus 2019

- Mengerjakan pembuatan hotspot menggunakan mikrotik.
- Mengerjakan pembuatan server menggunakan CentOS 5.
- Bergerak ke lapangan untuk melakukan *troubleshooting* di rumah pelanggan

##### Guru Ekstrakurikuler di SMPIT Al-Fityan | Guru Ekstrakurikuler Pemrograman

24 Juli 2021 – 14 Desember 2021

- Mengajari algoritma dasar dari pemrograman.
- Mengajari bahasa pemrograman pascal.
- Mengajari bahasa pemrograman pascal yang dipersiapkan untuk olimpiade.

##### Praktik Kerja Lapangan di PT. Lintasarta | Divisi Operasional

Juni 2023 – Juli 2023

- Mengerjakan konfigurasi router bernama Fortinet.
- Mengerjakan konfigurasi M2M bernama Bivocom.
- Mengumpulkan data-data pada kegiatan lapangan yang sedang berlangsung.

##### Guru Ekstrakurikuler SMA Al-Fityan | Guru Ekstrakurikuler Pemrograman

24 Juli 2023 – Sekarang

- Mengajari algoritma dasar dari pemrograman.
- Mengajari bahasa pemrograman pascal.
- Mengajari pemrograman web.
- Mengajari bagaimana dasar *men-develop* web.