



Prototype Kacamata Pemandu bagi Tunanetra dengan Keterbatasan Penglihatan

Rometdo Muzawi
Jurusan Manajemen
Informatika, STMIK Amik
Riau
rometdomuzawi@stmik-amik-
riau.ac.id

Syahrul Imardi
Jurusan Teknik Informatika,
STMIK Amik Riau
syahrulimardi@stmik-amik-
riau.ac.id

Yoyon Efendi
Jurusan Teknologi
Informasi, STMIK Amik
Riau
yoyonefendi@stmik-amik-
riau.ac.id

Abstract

Microcontroller is a family of microprocessors with a chip that can perform digital data processing in accordance with given language commands. By utilizing a microcontroller an object detection device for the blind is created. Researchers want to make the latest innovations, one of which is a low-cost glasses detection system that can be used to help blind people. With the presence of a glasses detection system, sufferers no longer use a stick as a guide. This project uses several ultrasonic proximity sensors, an arduino mini pro, an mp3 player module, and several vibration motors. The circuit board used in this project was designed in the form of a prototype, which can be worn for people with visual impairments. The installed Arduino will detect obstructions with the help of sensors and notify users through headphones and vibration motors. The sensor works to detect objects and holes that are in front or at the bottom, so the speaker gives a warning to the user of these glasses.

Keywords: *Microcontroller, Glasses, Sensors, Limited Vision*

Abstrak

Mikrokontroler merupakan keluarga mikroprosesor dengan sebuah chip yang dapat melakukan pemrosesan data secara digital sesuai dengan perintah bahasa yang diberikan. Dengan memanfaatkan mikrokontroler diciptakan suatu alat pendeteksi benda bagi tunanetra. Peneliti ingin membuat inovasi terbaru salah satunya sistem pendeteksi kacamata benda berbiaya rendah yang dapat digunakan untuk membantu penderita tunanetra.

Dengan adanya sistem pendeteksi kacamata benda, penderita tidak lagi menggunakan tongkat sebagai penuntun. Proyek ini menggunakan beberapa sensor jarak ultrasonik, mini pro arduino, modul mp3 player, dan beberapa motor getaran. Papan sirkuit yang digunakan dalam proyek ini dirancang dalam bentuk prototype, yang dapat dikenakan bagi penderita tunanetra. Arduino yang dipasang akan mendeteksi penghalang dengan bantuan sensor dan memberi tahu pengguna melalui headphone dan motor getaran. Sensor bekerja mendeteksi benda dan lubang yang berada didepan atau dibagian bawah, sehingga speaker memberikan peringatan kepada pengguna kacamata tersebut.

Kata kunci : *Mikrokontroler, Kacamata, Sensor, Keterbatasan Penglihatan.*

1. Pendahuluan

Kemajuan ilmu pengetahuan dan teknologi telah mendorong manusia untuk berusaha mengatasi segala permasalahan yang timbul di sekitarnya serta meringankan pekerjaan yang ada. Salah satu teknologi yang berkembang saat ini adalah kemajuan teknologi dalam dunia mikrokontroler. Mikrokontroler merupakan keluarga mikroprosesor dengan sebuah chip yang dapat melakukan pemrosesan data secara digital sesuai dengan perintah bahasa yang diberikan (Anwar, 2012). Tidak dapat dipungkiri kemajuan teknologi yang semakin cepat harus bisa dimanfaatkan, dipelajari serta diterapkan dalam kehidupan sehari-hari. Dengan memanfaatkan

mikrokontroler diciptakan suatu alat pendeteksi yang dapat membantu tunanetra.

Kajian penelitian terdahulu yang dilakukan oleh (Agung, 2016), “K-Netra, Kacamata Cerdas Untuk Tunanetra”. Adapun maksud dan tujuan K-Netra berfungsi untuk melakukan deteksi benda yang berada dalam jalur gerak penyandang tunanetra. Ia mampu mengidentifikasi benda-benda sebagai benda mati atau makhluk hidup. Dengan K-Netra, ini diharapkan dapat mengurangi risiko terjadinya kecelakaan pada penyandang tunanetra.

Merujuk pada penelitian yang telah dilakukan (Nada, Fakhr, & Seddik, 2015) “Assistive Infrared Sensor-Based Smart Stick For Blind People” membuat alat bantu bagi tuna netra seperti alat bantu tongkat pintar yang memungkinkan tuna netra untuk bernavigasi secara bebas. Tongkat pintar ini memandu pengguna melalui suara ke earphone dengan menggunakan jaringan Bluetooth. Namun, biaya yang diperlukan dalam membuat *prototype* masih relatif mahal. Karena itu peneliti ingin membuat inovasi terbaru salah satunya sistem pendeteksi pada kacamata pendeteksi benda berbiaya rendah yang dapat digunakan untuk membantu penderita tunanetra. Bagi para penderita tuna netra yang aktifitas kesehariannya kurang normal pada khususnya berjalan, biasanya dibantu dengan tongkat. Dengan adanya sistem pendeteksi benda pada kacamata, penderita tidak lagi menggunakan tongkat sebagai penuntun.

Proyek ini menggunakan beberapa sensor jarak ultrasonik, mini pro arduino, modul mp3 player, dan beberapa motor getaran. Papan sirkuit yang digunakan dalam proyek ini dirancang dalam bentuk *prototype*, yang dapat dikenakan bagi penderita tunanetra. Arduino yang dipasang akan mendeteksi penghalang dengan bantuan sensor dan memberi tahu pengguna melalui *headphone* dan motor getaran.

2. Kajian Pustaka

Pengertian dari kebutaan dapat ditinjau dari dua sisi, menurut definisi legal dan menurut definisi edukasional/fungsional (Setiawan et al., 2014). Definisi legal adalah definisi menurut undang-undang yang digunakan untuk menentukan fasilitas kesehatan sesuai undang-undang. Dua hal yang dinilai dari pandangan ini adalah ketajaman penglihatan (visual acuity) dan medan pandang (visual field) (Lausegger, Spitzer, & Ebner, 2017). Snellen Chart merupakan alat paling umum untuk menentukan tingkat visus seseorang. Berdasarkan hasil dari tes Snellen Chart, WHO mendefinisikan kebutaan berdasarkan ketajaman penglihatan dengan rentang sebagai berikut. a. visus

6/6 hingga 6/18 : penglihatan normal (normal vision)
b. visus <6/18 hingga >=3/60 : kurang awas (low vision)
c. visus <3/60 : buta (blindness).

Pada tahun 2017 Bai J, Lian S, dan teman-teman diciptakan Kacamata Pemandu Cerdas untuk Orang-Orang Tunanetra di Lingkungan Dalam Ruangan (Yulianto, 2019). Dalam studi ini mereka menerapkan kacamata dengan algoritma multi-sensor fusion based obstacle avoiding yang diusulkan, yang memanfaatkan baik sensor kedalaman dan sensor ultrasonik untuk memecahkan masalah mendeteksi hambatan kecil, dan hambatan transparan.

Dikutip dari (Muzawi & Fauzan, 2018). *Prototype* adalah tahap analisis dari siklus pengembangan sistem yang merupakan pendefinisian dari kebutuhan-kebutuhan fungsional, serta menggambarkan bagaimana suatu sistem dibentuk yang dapat berupa penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi, termasuk menyangkut mengkonfigurasi dari komponen-komponen perangkat keras dan perangkat lunak dari suatu sistem. *Prototyping* model dapat diklasifikasikan menjadi beberapa tipe seperti terlihat pada gambar 1.



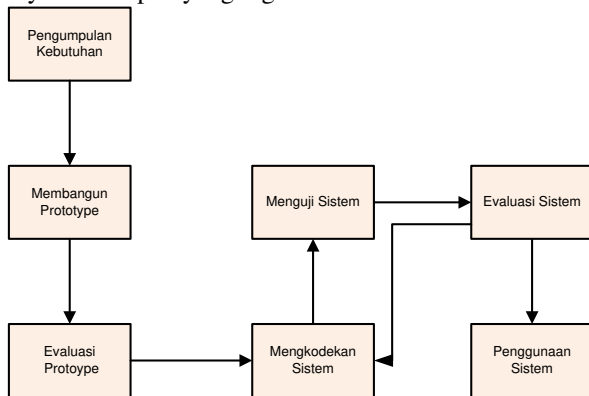
Gambar 1. Klasifikasi Prototyping Model

- Reusable prototype* :
Prototype yang akan ditransformasikan menjadi produk final.
- Throwaway prototype* :
Prototype yang akan dibuang begitu selesai menjalankan maksudnya.
- Input/Output prototype* :
Prototype yang terbatas pada antar muka pengguna (user interface).
- Processing prototype* :
Prototype yang meliputi perawatan file dasar dan proses-proses transaksi.
- System prototype* :
Prototype yang berupa model lengkap dari perangkat lunak.

Tahap-tahap dalam *prototyping* boleh dikata merupakan tahap-tahap yang dipercepat. Strategi utama dalam *prototyping* adalah kerjakan yang mudah terlebih dahulu dan sampaikan hasil kepada pengguna sesegera mungkin.

3. Metodologi Penelitian

Adapun kerangka kerja dalam penelitian ini sebanyak 7 tahapan yang digambarkan berikut ini:



Gambar 2. Metodologi Penelitian

Tahapan diatas dapat dijelaskan sebagai berikut : Berdasarkan kerangka kerja diatas, maka masing – masing tahapan tersebut dapat dijelaskan sebagai berikut :

1. Pengumpulan Kebutuhan

Dalam pengumpulan data, metode yang digunakan dalam penulisan ini yaitu Studi Pustaka, dengan mempelajari dan membaca teori tentang *Smart Glass* yang menunjang penulisan ini serta referensi dari jurnal lain yang akan menunjang pembahasan.

2. Membangun *Prototyping*

Membangun *prototyping* dengan membuat perancangan kacamata pintar (*smart glass*) sesuai data yang ada berdasarkan tahapan yang ditetapkan pada tapen pengumpululan data (Muzawi & Fauzan, 2018).

3. Evaluasi *Prototyping*

Evaluasi ini dilakukan oleh peneliti apakah *prototyping* yang sudah dibangun sudah sesuai dengan keinginan. Jika sudah sesuai maka langkah 4 akan diambil. Jika tidak *prototyping* akan direvisi dengan mengulang langkah 1, 2 , dan 3.

4. Mengkodekan Sistem

Dalam tahap ini *prototyping* yang sudah di sepakati diterjemahkan ke dalam bahasa pemrograman yang sesuai.

5. Menguji Sistem

Setelah sistem kacamata pintar (*smart glass*) sudah jadi yang siap pakai, maka harus dites dahulu sebelum digunakan.

6. Evaluasi

Menguji seluruh spesifikasi yang terstruktur, baik dari segi hardware maupun software secara keseluruhan. Pada tahap ini dilakukan uji coba sistem yang telah selesai dibuat. Proses uji coba ini diperlukan untuk memastikan bahwa sistem yang telah dibuat sudah benar, sesuai dengan karakteristik yang ditetapkan dan tidak ada kesalahan yang terkandung didalamnya.

7. Penggunaan Sistem

Perangkat yang telah diuji dan siap untuk digunakan

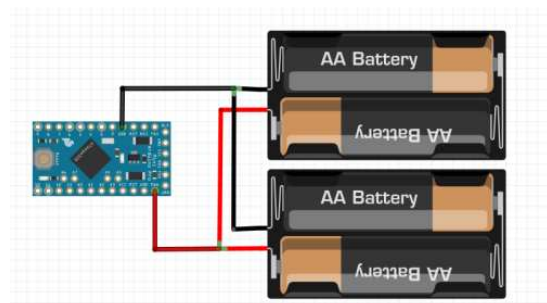
4. Pembahasan

4.1 Perancangan Perangkat Keras

Rangkaian sistem terdiri dari beberapa komponen perangkat keras. Semua perangkat tersebut akan dirangkai pada Arduino.

4.1.1 Rangkaian Instalasi Catu Daya ke Arduino.

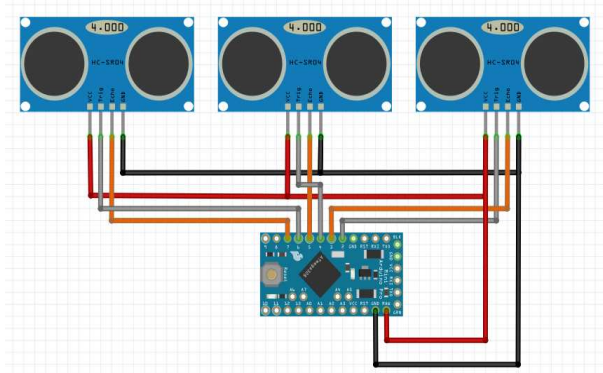
Rangkaian installasi catu daya ke Arduino. Rangkaian ini menggambarkan hubungan Baterai dengan Arduino dimana Baterai terhubung melalui pin Arduino, desain rangkaian dapat dilihat pada gambar 3 dibawah ini :



Gambar 3. Rangkaian Instalasi Catu Daya ke Arduino

4.1.2 Rangkaian Sensor Ping ke Arduino

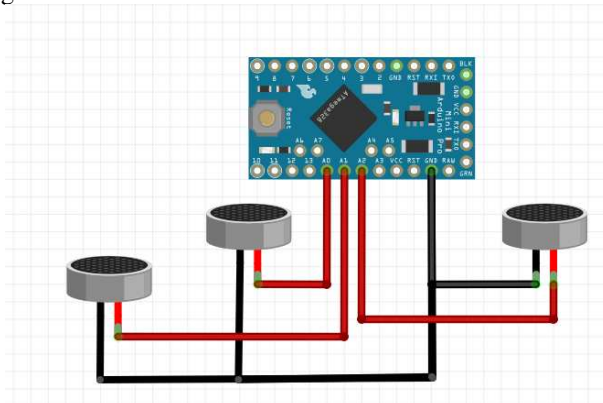
Rangkaian *Sensor Ping* ke Arduino. Rangkaian ini berfungsi sebagai pendeteksi benda yang ada didepan dan dibawahnya. Sensor Ping ini dikontrol oleh Arduino melalui pin 2,4,6 sebagai trig, pin 3,5,7 echo, Vcc pin Raw dan Ground pada pin Gnd. rangkaian ini membutuhkan tegangan 5V Desain perancangan dapat dilihat pada gambar 4.



Gambar 4. Rangkaian Sensor Ping ke Arduino

4.1.3 Rangkaian Microphone ke Arduino

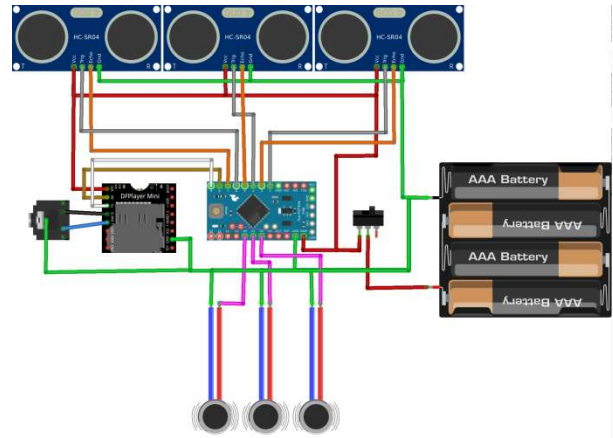
Rangkaian *Microphone* ke Arduino. Rangkaian ini berfungsi sebagai indikator benda yang ada didepan dan dibawahnya. *Microphone* ini dikontrol oleh Arduino melalui pin A0-A2 Vcc pin dan Raw pada pin Ground. rangkaian ini membutuhkan tegangan 5V Desain perancangan dapat dilihat pada gambar 5.



Gambar 5. Rangkaian Micphone ke Arduino

4.1.4 Rangkaian Sistematika

Rangkaian sistematika perangkat merupakan rangkaian keseluruhan dari alat *monitoring* dan pengamanan otomatis sepeda motor dalam penelitian ini. Rangkaian dapat dilihat pada gambar 6:

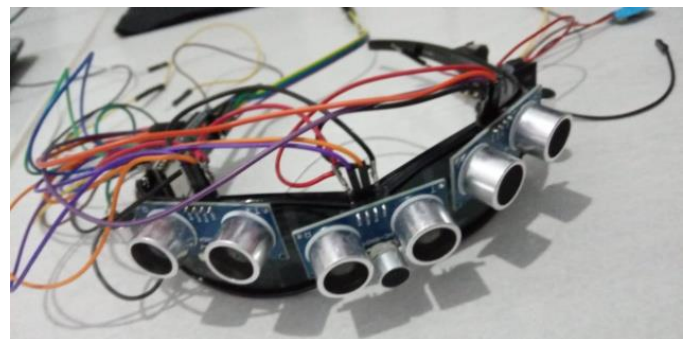


Gambar 6. Rangkaian Keseluruhan

4.2 Desain Perangkat Keras

Pada desain tampilan dimaksudkan untuk menampilkan *prototype* alat kacamata pemandu untuk tunanetra yang digunakan untuk mendeteksi benda yang ada didepan dan dibawah serta memberikan peringatan melalui microphone dan earphone sebagai adanya benda atau lubang yang berada didepan.

1. Cara Kerja perangkat keras ini, dimana alat ini bekerja dengan mendeteksi rintangan nya ada didepanya.
2. Sensor akan bekerja jika ada nya hambatan (benda) yang berada didepan sensor dan akan mengirimkan perintah bib atau suara kespeaker yang telah diposisikan pada kacamata tersebut.
3. Sensor ini bekerja dengan tegangan daya sebesar 5 volt.



Gambar 7. Design Perangkat Keras

4.3 Perancangan Algoritma Program

Perancangan algoritma program bertujuan untuk menentukan alur program agar mempermudah saat penulisan program dan membuat penulisan

program lebih terarah. Algoritma program ini menggambarkan proses yang akan dilakukan oleh mikrokontroler pada saat menerima sinyal masukan dari ultrasonik dan memberikan sinyal keluaran kepada modul mp3 player untuk memberikan suara peringatan.

```
#include <Arduino.h>
#include <NewPing.h>
#include <SoftwareSerial.h>
#include <DFRobotDFPlayerMini.h>

#define MAX_DISTANCE 400

const int trigPin1 = 2;
const int echoPin1 = 3;
const int trigPin2 = 4;
const int echoPin2 = 5;
const int trigPin3 = 6;
const int echoPin3 = 7;

const int motorPinRight = A0;
const int motorPinCenter = A1;
const int motorPinLeft = A2;

int minLeftDistance = 100;
int minCenterDistance = 100;
int minRightDistance = 100;

static unsigned long timer = millis();

int waitTime = 3500; NewPing
sonarLeft(trigPin1, echoPin1
,MAX_DISTANCE);
NewPing sonarCenter(trigPin2, echoPin2
,MAX_DISTANCE);
NewPing sonarRight(trigPin3, echoPin3
,MAX_DISTANCE);

SoftwareSerial mp3SoftwareSerial(8, 9); //
RX, TX
DFRobotDFPlayerMini DFPlayer;

void printDetail(uint8_t type, int value){
switch (type) {
case TimeOut:
Serial.println(F("Time Out!"));
break;
case WrongStack:
Serial.println(F("Stack Wrong!"));
break; case DFPlayerCardInserted:
Serial.println(F("Card Inserted!"));
break;
}
```

```
case DFPlayerCardRemoved:
Serial.println(F("Card Removed!"));
break;
case DFPlayerCardOnline:
Serial.println(F("Card Online!"));
break;
case DFPlayerPlayFinished:
Serial.print(F("Number:"));
Serial.print(value);
Serial.println(F(" Play Finished!"));
break;
case DFPlayerError:
Serial.print(F("DFPlayerError:"));
switch (value) {
case Busy:
Serial.println(F("Card not found"));
break;
case Sleeping:
Serial.println(F("Sleeping"));
break;
case SerialWrongStack:
Serial.println(F("Get Wrong Stack"));
break;
case CheckSumNotMatch:
Serial.println(F("Check Sum Not Match"));
break;
case FileIndexOut:
Serial.println(F("File Index Out of
Bound"));
break;
case FileMismatch:
Serial.println(F("Cannot Find File"));
break;
case Advertise:
Serial.println(F("In Advertise"));
break;
default:
break;
}
break;
default:
break;
}
}
void leftAlert(int distance) {
if (millis() - timer > waitTime) {
timer = millis();
DFPlayer.playLargeFolder(01, distance+1);
}
if (DFPlayer.available()) {
printDetail(DFPlayer.readType(),
DFPlayer.read()); //Print the detail message
from DFPlayer
}
}
```

```
void centerAlert(int distance) {

  if (millis() - timer > waitTime) {
    timer = millis();
    DFPlayer.playLargeFolder(02, distance+1);
  }
  if (DFPlayer.available()) {
    printDetail(DFPlayer.readType(),
DFPlayer.read()); //Print the detail message
from DFPlayer
  }
}

void rightAlert(int distance) {

  if (millis() - timer > waitTime) {
    timer = millis();
    DFPlayer.playLargeFolder(03, distance+1);
  }
  if (DFPlayer.available()) {
    printDetail(DFPlayer.readType(),
DFPlayer.read()); //Print the detail message
from DFPlayer
  }
}

void setupDFPlayer() {

  mp3SoftwareSerial.begin(9600);
  Serial.println();
  Serial.println(F("Blind assistances smart
glass..."));
  Serial.println(F("Initializing DFPlayer ...
(May take 3~5 seconds)"));

  if (!DFPlayer.begin(mp3SoftwareSerial)){
    Serial.println(F("Unable to begin:"));
    Serial.println(F("1.Please recheck the
connection!"));
    Serial.println(F("2.Please insert the SD
card!"));
    while(true);
  }
  Serial.println(F("DFPlayer Mini online.));
DFPlayer.setTimeout(500); //Set serial
communicaion time out 500ms
DFPlayer.volume(29); //Set volume value
(0~30)
DFPlayer.EQ(DFPLAYER_EQ_NORMAL);
DFPlayer.outputDevice(DFPLAYER_DEVIC
E_SD);
}

void setup() {
  Serial.begin(9600);
  setupDFPlayer();
}
```

```
void loop() {
  delay(29);
  int leftDistance = sonarLeft.ping_cm();
  int centerDistance =
sonarCenter.ping_cm();
  int rightDistance =
sonarRight.ping_cm();

  // For debugging
  // Serial.print("Left: ");
  // Serial.print(leftDistance);
  // Serial.print(", Center: ");
  // Serial.print(centerDistance);
  // Serial.print(", Right: ");
  // Serial.println(rightDistance);

  if(leftDistance < minLeftDistance &&
leftDistance > 1) {

Serial.print("Obstacle ");
Serial.print(leftDistance);
Serial.println(" CM to the left");
leftAlert(leftDistance);
analogWrite(motorPinLeft, 512);
}
else {
  analogWrite(motorPinLeft, 0);
}

  if(centerDistance < minCenterDistance
&& centerDistance > 1) {

Serial.print("Obstacle ");
Serial.print(centerDistance);
Serial.println(" CM ahead");

centerAlert(centerDistance);
analogWrite(motorPinCenter, 512);
}
else {
  analogWrite(motorPinCenter, 0);
}

  if(rightDistance < minRightDistance &&
rightDistance > 1) {

Serial.print("Obstacle ");
Serial.print(rightDistance);
Serial.println(" CM to the right");

rightAlert(rightDistance);
analogWrite(motorPinRight, 512);
}
else {
  analogWrite(motorPinRight, 0);
}
}
```

4.4 Hasil

Pengujian sistem dilakukan untuk mengetahui apakah sistem dapat berfungsi secara efektif dan dapat menghasilkan keluaran sebagaimana yang diharapkan. Pengujian diawali dengan menguji komponen atau modul secara terpisah. Setelah itu barulah dilakukan pengujian terhadap sistem secara keseluruhan.

1. Pengujian Sensor Ping

Untuk pengujian sensor ping dalam mendeteksi benda atau penghalangan depan dilakukan dengan menempatkan objek benda atau penghalang pada jarak 1 meter, 2 meter dan 3 meter dan dilakukan pengujian sebanyak 5 kali percobaan. Hasil pengujian sensor dapat dilihat pada tabel berikut:

Tabel 1. Pengujian Sensor Ping

Jarak Benda (Meter)	Percobaan (dalam cm)					Rata-rata	Standar Deviasi	Efektifitas
	I	II	III	IV	V			
1 m	99,1	96,4	98,9	99,9	97,5	98,36	1,244	98,756%
2 m	188	185	180	183	191	199,4	9,42	90,1548%
3 m	297	290	289	291	291	293,2	14,588	85,412%

2. Respon Time Alat

Pengujian respon time alat, dilakukan untuk mengetahui selang waktu antara saat sensor mendeteksi benda atau halangan hingga munculnya respon output berupa suara peringatan. Adapun hasil pengujian respon time alat dapat dilihat dari tabel berikut:

Tabel 2. Pengujian respon time alat

Sensor	Posisi Obyek	Respon Time
Sensor 1	Kiri	3 dtk
	Depan	1 dtk
Sensor 2	Bawah	2 dtk
	Kanan	3 dtk

Setelah melihat ke dua tabel diatas, dapat diketahui bahwa pada saat tuna netra berjalan dalam

jarak 1 meter memerlukan waktu tempuh rata-rata 9 detik. Adapun waktu yang dibutuhkan sensor untuk memberikan perintah kepada speaker untuk mengeluarkan output hanya memerlukan waktu 1 detik, jadi pada saat tuna netra berjalan pada jarak 1 meter masih mempunyai waktu sebanyak 9 detik untuk menghindari halangan yang berada di depannya.

5. Simpulan

Mikrokontroler Atmega 328p dapat digunakan sebagai kacamata pendeteksi benda untuk penderita tunanetra. Alat ini bekerja dengan bantuan sensor yang dengan cara mendeteksi benda dan lubang yang berada dibagian depan dan dibawah. Saat sensor bekerja, speaker memberikan peringatan kepada pengguna kacamata tersebut

6. Ucapan Terimakasih

Terimakasih kepada Yayasan Komputasi Riau yang telah mendukung dan mendanai penelitian internal STMIK Amik Riau dan terimakasih juga kepada LPPM STMIK Amik Riau yang telah memfasilitasi sehingga penelitian ini dapat terlaksana dengan baik.

7. Referensi

- Agung. (2016). K-Netra , Kacamata Cerdas Untuk Tunanetra. 2016, Pp. 1–3.
- Anwar, A. (2012). Kacamata Pendeteksi Benda Untuk Tuna Netra Berbasis Mikrokontroler Atmega8. *Universitas Islam Negeri Alauddin Makassar*, 1–79.
- Due, B. L. (2014). The Future Of Smart Glasses : An Essay About Challenges And Possibilities With Smart Glasses. *Working Papers On Interaction And Communication*, 1(2), 1–21.
- Lausegger, G., Spitzer, M., & Ebner, M. (2017). Omnicolor - A Smart Glasses App To Support Colorblind People. *International Journal Of Interactive Mobile Technologies*, 11(5), 161–177. <https://doi.org/10.3991/ijim.v11i5.6922>
- Muzawi, Rometdo, Yoyon Efendi, W. A. (2018). Satin – Sains Dan Teknologi Informasi Sistem Pengendalian Lampu Berbasis Web Dan Mobile. *Satin – Sains Dan Teknologi Informasi*, 4(1).
- Muzawi, R., & Fauzan, A. (2018). Prototype Pengontrolan Titik Fokus Panel Surya Terhadap Energi Matahari Secara Otomatis. *Inovtek Polbeng*, 3, 1–9.
- Nada, A. A., Fakhri, M. A., & Seddik, A. F. (2015). Assistive Infrared Sensor Based Smart Stick For Blind People. *Science And Information Conference*, 1149–1154.
- Nugroho, A. B. (2011). Perancangan Tongkat Tuna Netra Menggunakan Teknologi Sensor Ultrasonik Untuk Membantu Kewaspadaan Dan Mobilitas.
- Setiawan, W., Kusnendar, J., Indonesia, U. P., Ultrasonik, S.,

- Pendahuluan, I., & Belakang, A. L. (2014). Pengembangan Kacamata Pemandu Tuna Netra Menggunakan Sensor Ultrasonik Dan Mikrokontroler.
- Yulianto, A. (2019). Design Prototype Of Audio Guidance System For Blind By Using Raspberry Pi And Fuzzy Logic Controller Design Prototype Of Audio Guidance System For Blind By Using Raspberry Pi And Fuzzy Logic Controller. *Journal Of Physics: Conference Series Paper*, 1–11. <https://doi.org/10.1088/1742-6596/1230/1/012024>