LAMPIRAN

Lampiran 1 Surat Kesediaan Membimbing TA Pembimbing 1

SURAT KESEDIAAN MEMBIMBING TA

Yang bertanda tangan di bawah ini:

Nama	: Rais, S.Pd., M.Kom.
NIDN	: 0614108501
NIPY	: 07.011.083
Jabatan Struktural	: Ka. UPT Pelatihan dan Sertifikasi
Jabatan Fungsional	: Lektor

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

Nama	: Rifki Rizki Pratama
NIM	: 22040085
Program Studi	: DIII Teknik Komputer

Judul TA : PENGEMBANGAN PROTOTYPE ROBOT SAR HEXAPOD UNTUK PENANGANAN PASCA GEMPA

Demikian pernyataan dibuat agar dapat dilaksanakan sebagaimana mestinya.

Tegal, 14 April 2025

Mengetahui,
Ka. Prodi DIII Teknik Komputer,

Ida Afriliana, ST., M.Kom
NIPY. 12.013.168

Dosen Pembimbing I,

Rais, S.Pd., M.Kom.
NIPY. 07.011.083

Lampiran 2 Surat Kesediaan Membimbing TA Pembimbing 2

SURAT KESEDIAAN MEMBIMBING TA

Yang bertanda tangan di bawah ini:

Nama	: Nurohim, S.ST, M.Kom
NIDN	: 0625067701
NIPY	: 09.017.342
Jabatan Struktural	: Dosen Tetap
Jabatan Fungsional	: Asisten Ahli

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

Nama	: Rifki Rizki Pratama
NIM	: 22040085
Program Studi	: DIII Teknik Komputer

Judul TA : PENGEMBANGAN PROTOTYPE ROBOT SAR HEXAPOD UNTUK PENANGANAN PASCA GEMPA

Demikian pernyataan dibuat agar dapat dilaksanakan sebagaimana mestinya.

Tegal, 15 April 2025

Mengetahui,
Ka. Prodi DIII Teknik Komputer,

Ida Afriliana, ST., M.Kom
NIPY. 12.013.168

Dosen Pembimbing II,

Nurohim, S.ST, M.Kom
NIPY. 09.017.342

Lampiran 3 Dokumentasi

Lampiran 4 Source Code

```
//main.ino
#include <Servo.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include "vl53l0x_manager.h"
#include "ultrasonic.h"
#include "LegMovement.h"
#include <Pixy2.h>

Servo servoupdown;
Servo servogrip;
LiquidCrystal_I2C lcd(0x27, 16, 2);
Pixy2 pixy;

enum MoveType {
  NONE,
  FORWARD,
  BACKWARD,
  TURN_LEFT,
  TURN_RIGHT,
  SLIDE_LEFT,
  SLIDE_RIGHT
};

const int buttonPin = 8;
bool standBy = true;
unsigned long lastDebounceTime = 0;
unsigned long debounceDelay = 50;
bool lastButtonState = HIGH;
bool buttonPressed = false;
bool targetDetected = false;
bool targetDiambil = false;
bool modeAmbilTarget = false;
bool modeMenujuTempatTaruh = false;
bool tempatTaruhDetected = false;
bool habisTaruh = false;
bool sudahSelesai = false;

void setup() {
  Serial.begin(115200);
  Wire.begin();

  pixy.init();
  setupVL53();
  ultrasonicSetup();
  setupLegs();

  servoupdown.attach(40);
  servogrip.attach(42);
  servoupdown.write(90); // Angkat lengan
  servogrip.write(0); // Tutup gripper

  lcd.begin();
  lcd.backlight();
```

```
  pinMode(buttonPin, INPUT_PULLUP);
}

void loop() {
  updateVL53();
  handleButton();
  bacajarak();
  bacaPixy();
  readDistance();

  if (!standBy) {
    start();
  } else {
    Hstandby();
  }
}

void handleButton() {
  int reading = digitalRead(buttonPin);
  if (reading != lastButtonState) lastDebounceTime = millis();
  if ((millis() - lastDebounceTime) > debounceDelay) {
    if (reading == LOW && !buttonPressed) {
      standBy = !standBy;
      buttonPressed = true;
    } else if (reading == HIGH) {
      buttonPressed = false;
    }
  }
  lastButtonState = reading;
}

void bacajarak() {
  if (modeAmbilTarget) {
    lcd.clear();
    lcd.setCursor(5, 1);
    lcd.print("        ");  // Bersihkan baris bawah
    lcd.print(readDistance());
    return;
  }

  int jarakDepan = getDistance(2);
  int jarakKananDepan = getDistance(3);
  int jarakKiriDepan = getDistance(4);
  int jarakKananBelakang = getDistance(1);
  int jarakKiriBelakang = getDistance(0);
  int jarakBelakang = getDistance(5);

  lcd.setCursor(12, 0); lcd.print("    "); lcd.setCursor(12, 0);
lcd.print(jarakKananDepan);
  lcd.setCursor(0, 0);  lcd.print("    "); lcd.setCursor(0,
0);  lcd.print(jarakKiriDepan);
  lcd.setCursor(7, 0);  lcd.print("    "); lcd.setCursor(7,
0);  lcd.print(jarakDepan);
  lcd.setCursor(7, 1);  lcd.print("    "); lcd.setCursor(7, 1);
lcd.print(jarakBelakang);
```

```
  lcd.setCursor(12, 1); lcd.print("    "); lcd.setCursor(12, 1);
lcd.print(jarakKananBelakang);
  lcd.setCursor(0, 1);  lcd.print("    "); lcd.setCursor(0, 1);
lcd.print(jarakKiriBelakang);
}


void bacaPixy() {
  pixy.ccc.getBlocks();
  targetDetected = false;
  tempatTaruhDetected = false;

  for (int i = 0; i < pixy.ccc.numBlocks; i++) {
    uint8_t sig = pixy.ccc.blocks[i].m_signature;
    int xCenter = pixy.ccc.blocks[i].m_x;

    if (sig == 1) { // Target (objek)
      if (xCenter < 120) HTurnLeft();
      else if (xCenter > 140) HTurnRight();
      else targetDetected = true;
    }

    if (sig == 2) { // Tempat taruh
      tempatTaruhDetected = true;

      if (xCenter < 120) HTurnLeft();
      else if (xCenter > 140) HTurnRight();
    }
  }
}

void ambilTarget() {
  if (!targetDetected) return; // Tidak ambil jika tidak ada
target

  modeAmbilTarget = true;

  Hstandby();
  delay(1000);
  servogrip.write(75); // Buka gripper
  delay(500);
  servoupdown.write(10); // Turunkan lengan
  delay(500);

  // Mulai dekati objek jika belum cukup dekat
  unsigned long startTime = millis();
  const unsigned long timeout = 5000;

  while (readDistance() > 5 && millis() - startTime < timeout) {
    HForward();      // Maju pelan sambil mendekat
    delay(250);      // Gerakan langkah smooth atau delay sesuai
gait
  }

  Hstandby(); // Berhenti setelah cukup dekat
  delay(1000);
```

```
  // Cek lagi, pastikan benar-benar dekat
  if (readDistance() < 5) {
    Lstandby();
    delay(1000);
    forwardpick();
    delay(1000);
    servogrip.write(0); // Tutup gripper
    delay(500);
    servoupdown.write(90); // Angkat lengan
    delay(500);
    Hstandby();
    delay(500);
  } else {
    Lstandby();
    delay(1000);
    forwardpick();
    delay(1000);
    servogrip.write(0); // Tutup gripper
    delay(500);
    servoupdown.write(90); // Angkat lengan
    delay(500);
    Hstandby();
    delay(500);
  }

  modeAmbilTarget = false;
}

void taruhTarget() {
  Hstandby();
  delay(1000);
  forwardpick();
  delay(1000);
  servoupdown.write(0); // Turunkan
  delay(500);
  servogrip.write(90); // Buka gripper (lepas)
  delay(500);
  servoupdown.write(90); // Angkat
  delay(500);
  servogrip.write(0);
  delay(500);
  targetDiambil = false; // Reset status
  modeMenujuTempatTaruh = false;
  Hstandby();
  delay(500);
}

void start() {
  static MoveType lastMove = NONE;
  static bool delayStandby = false;
  static unsigned long delayStartTime = 0;
  const unsigned long standbyDelay = 100;
  MoveType nextMove = NONE;

  if (sudahSelesai) {
    Hstandby();
```

```
  return;
}

if (delayStandby) {
  if (millis() - delayStartTime < standbyDelay) return;
  delayStandby = false;
  Hstandby();
  return;
}

// Baca sensor
int jarakDepan = getDistance(2);
int jarakKananDepan = getDistance(3);
int jarakKiriDepan = getDistance(4);
int jarakKananBelakang = getDistance(1);
int jarakKiriBelakang = getDistance(0);
int jarakBelakang = getDistance(5);

// Threshold
const int tSangatDekat = 60;
const int tDekat = 80;
const int tSedang = 100;
const int tJauh = 150;
const int tSangatJauh = 200;

// Target logic
if (targetDetected && !targetDiambil) {
  if (jarakDepan < 250) {
    ambilTarget();
    targetDiambil = true;
    modeMenujuTempatTaruh = true;
    targetDetected = false;
    return;
  }
}

if (modeMenujuTempatTaruh && targetDiambil) {
  if (tempatTaruhDetected) {
    if (jarakDepan < 250) {
      taruhTarget();
      habisTaruh = true;
      targetDiambil = false;
      modeMenujuTempatTaruh = false;
      targetDetected = false;
      return;
    }
  }
}

if (habisTaruh) {
  Hstandby();
  habisTaruh = false;
  sudahSelesai = true;
  return;
}
```

```
  // === Abaikan jarak depan untuk penghindaran kalau mode target
===
  bool abaikanJarakDepan = (targetDetected && !targetDiambil) ||
(modeMenujuTempatTaruh && targetDiambil);

  // Deteksi bahaya gerakan sebelumnya
  bool hindarDepan = !abaikanJarakDepan && jarakDepan <
tSangatJauh;
  bool hindarKanan = jarakKananDepan < tDekat ||
jarakKananBelakang < tDekat;
  bool hindarKiri = jarakKiriDepan < tDekat || jarakKiriBelakang <
tDekat;

  // Lanjut gerakan sebelumnya jika masih bahaya
  if (lastMove == BACKWARD && hindarDepan) {
    HBackward(); return;
  }
  if (lastMove == SLIDE_LEFT && hindarKanan) {
    HSlideLeft(); return;
  }
  if (lastMove == SLIDE_RIGHT && hindarKiri) {
    HSlideRight(); return;
  }
  if (lastMove == TURN_LEFT && hindarKiri) {
    HTurnLeft(); return;
  }
  if (lastMove == TURN_RIGHT && hindarKanan) {
    HTurnRight(); return;
  }

  // === Logika penghindaran utama ===
  int rataKanan = (jarakKananDepan + jarakKananBelakang) / 2;
  int rataKiri = (jarakKiriDepan + jarakKiriBelakang) / 2;

  if (!abaikanJarakDepan && jarakDepan < tSangatJauh) {
    if (jarakBelakang > tDekat) {
      nextMove = BACKWARD;
    } else {
      if (rataKanan > rataKiri && rataKanan > tDekat) {
        nextMove = TURN_RIGHT;
      } else if (rataKiri > tDekat) {
        nextMove = TURN_LEFT;
      } else {
        nextMove = BACKWARD;
      }
    }

  } else if (jarakKananDepan < tDekat && jarakKananBelakang <
tDekat &&
             jarakKiriDepan < tDekat && jarakKiriBelakang <
tDekat) {
    if (jarakBelakang > jarakDepan) {
      nextMove = BACKWARD;
    } else {
      nextMove = FORWARD;
    }
```

```
  } else if (jarakKananDepan < tDekat && jarakKananBelakang <
tDekat) {
    nextMove = SLIDE_LEFT;

  } else if (jarakKiriDepan < tDekat && jarakKiriBelakang <
tDekat) {
    nextMove = SLIDE_RIGHT;

  } else if (jarakKananDepan < tSangatDekat || jarakKananBelakang
< tSangatDekat) {
    nextMove = SLIDE_LEFT;

  } else if (jarakKiriDepan < tSangatDekat || jarakKiriBelakang <
tSangatDekat) {
    nextMove = SLIDE_RIGHT;

  } else if (jarakKananDepan < tSedang || jarakKananBelakang <
tSedang) {
    nextMove = TURN_LEFT;

  } else if (jarakKiriDepan < tSedang || jarakKiriBelakang <
tSedang) {
    nextMove = TURN_RIGHT;

  } else {
    nextMove = FORWARD;
  }

  // Delay setelah slide
  bool isSlideNow = lastMove == SLIDE_LEFT || lastMove ==
SLIDE_RIGHT;
  bool isSlideNext = nextMove == SLIDE_LEFT || nextMove ==
SLIDE_RIGHT;

  if (isSlideNow && !isSlideNext && nextMove != NONE) {
    delayStandby = true;
    delayStartTime = millis();
    lastMove = NONE;
    return;
  }

  // Jalankan perintah gerakan
  switch (nextMove) {
    case SLIDE_LEFT: HSlideLeft(); break;
    case SLIDE_RIGHT: HSlideRight(); break;
    case TURN_LEFT: HTurnLeft(); break;
    case TURN_RIGHT: HTurnRight(); break;
    case BACKWARD: HBackward(); break;
    case FORWARD: HForward(); break;
    default: Hstandby(); break;
  }

  if (nextMove != NONE) lastMove = nextMove;
}
```

Lampiran 5 Manual Book

The Manual Book

PROTOTYPE ROBOT SAR HEXAPOD

"MAKERSPOD"

Parameter:

Item	: Prototype Robot SAR Hexapod
Version	: Makerspod 2.0
Processor	: Arduino Mega 2560
Sensor	: Pixy2 CMUcam 5, VL53L0X, HC-SR04
Display	: LCD I2C 16x2
Add tools	: Step-down
Output	: Information for display LCD
Baterai	: LiPo 3s 11V 5200 mAh 25C
Waktu Penggunaan	: 1-2 jam (pengisian penuh)
Waktu pengisian	: 1-1,5 jam

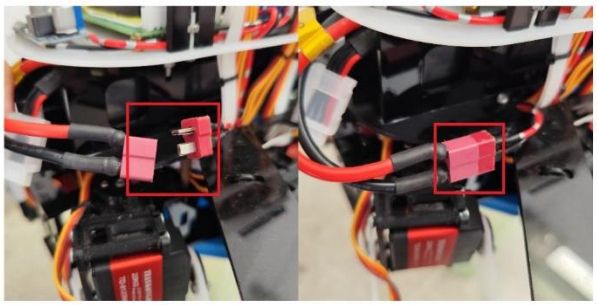
Fitur

Pada alat:

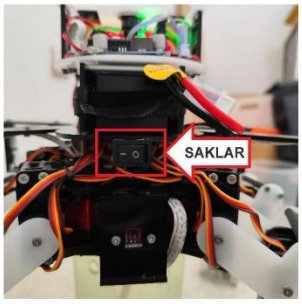
1. Pendeteksi jarak pada badan robot untuk mendeteksi rintangan di sekitar.
2. Pendeteksi jarak pada lengan gripper untuk mendeteksi jarak calon korban.
3. Pendeteksi objek berdasarkan warna untuk mendeteksi calon korban.
4. Dapat melihat nilai sensor jarak pada display LCD I2C 16x2.

Cara penggunaan:

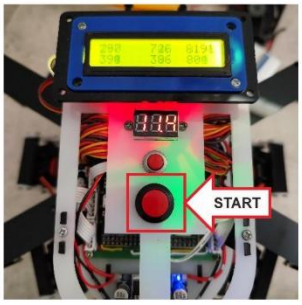
1. Siapkan Prototype robot SAR hexapod dan sambungkan dengan baterai.



2. Tekan saklar di bagian belakang untuk menyalakan robot. Tunggu hingga tampilan LCD menyala dan menampilkan nilai sensor sebagai tanda robot siap digunakan..



3. Jalankan robot dengan menekan tombol start.



4. Hentikan robot dengan menekan tombol start sekali lagi.
5. Jika terjadi error tekan push button reset.

Cara pengisian baterai:

1. Isi baterai jika pada volt display menampilkan voltase kurang dari 7V.

2. Lepaskan socket baterai dari robot dan keluarkan baterai dari tempat baterai.
3. Sambungkan baterai ke socket charger LiPo 3s.
4. Jika sudah penuh maka charger akan memberikan notifikasi.

Saran penyimpanan dan pemeliharaan:

Penyimpanan:

1. Simpan robot di tempat kering dan sejuk, hindari kelembapan dan panas berlebih.
2. Gunakan kotak pelindung atau koper berlapis busa untuk mencegah kerusakan fisik.
3. Lepaskan baterai dari robot jika tidak digunakan dalam jangka waktu lama.
4. Simpan baterai dalam LiPo bag atau wadah tahan api untuk keamanan.
5. Jauhkan dari jangkauan anak-anak dan hewan peliharaan.

Pemeliharaan:

1. Periksa konektor dan kabel secara berkala untuk memastikan tidak ada yang longgar atau rusak.
2. Bersihkan sensor (Pixy2 CMUcam 5, VL53L0X, HC-SR04) dengan kain halus dan kering secara rutin.
3. Kalibrasi ulang sensor jika pembacaan tidak akurat.
4. Update firmware atau program Arduino jika ada pembaruan atau perbaikan kode.
5. Isi baterai sesuai prosedur dan jangan biarkan overcharge atau kosong terlalu lama.
6. Periksa bagian mekanik seperti kaki dan gripper untuk memastikan tidak macet atau longgar.