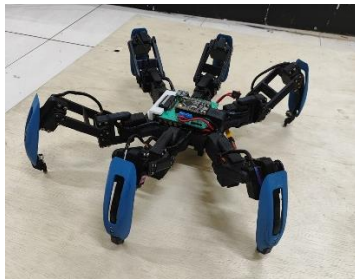


## LAMPIRAN A

### HASIL DESAIN KESELURUHAN ROBOT



Lampiran 1 PCB Robot



Lampiran 2 Mekanik Hexapod



Lampiran 3 Bentuk Keseluruhan Robot

Date	Time	Speed
20.02.2020	10:40	10.10
20.02.2020	10:41	10.10
20.02.2020	10:42	10.10
20.02.2020	10:43	10.10
20.02.2020	10:44	10.10
20.02.2020	10:45	10.10
20.02.2020	10:46	10.10
20.02.2020	10:47	10.10
20.02.2020	10:48	10.10
20.02.2020	10:49	10.10
20.02.2020	10:50	10.10
20.02.2020	10:51	10.10
20.02.2020	10:52	10.10
20.02.2020	10:53	10.10
20.02.2020	10:54	10.10
20.02.2020	10:55	10.10
20.02.2020	10:56	10.10
20.02.2020	10:57	10.10
20.02.2020	10:58	10.10
20.02.2020	10:59	10.10
20.02.2020	11:00	10.10

Lampiran 4 Data Pengujian Trayektorri

Date	Time	Speed
20.02.2020	10:40	10.10
20.02.2020	10:41	10.10
20.02.2020	10:42	10.10
20.02.2020	10:43	10.10
20.02.2020	10:44	10.10
20.02.2020	10:45	10.10
20.02.2020	10:46	10.10
20.02.2020	10:47	10.10
20.02.2020	10:48	10.10
20.02.2020	10:49	10.10
20.02.2020	10:50	10.10
20.02.2020	10:51	10.10
20.02.2020	10:52	10.10
20.02.2020	10:53	10.10
20.02.2020	10:54	10.10
20.02.2020	10:55	10.10
20.02.2020	10:56	10.10
20.02.2020	10:57	10.10
20.02.2020	10:58	10.10
20.02.2020	10:59	10.10
20.02.2020	11:00	10.10

Lampiran 5 Data Pengujian Limit Switch

## LAMPIRAN B

### PROGRAM SENSOR PADA ROBOT

```
#include "I2Cdev.h"
#include "MPU6050_6Axis_MotionApps20.h"
#if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
#include "Wire.h"
#endif
MPU6050 mpu;

#define OUTPUT_READABLE_YAWPITCHROLL
#define INTERRUPT_PIN 2
bool blinkState = false;
bool dmpReady = false;
uint8_t mpuIntStatus;
uint8_t devStatus;
uint16_t packetSize;
uint16_t fifoCount;
uint8_t fifoBuffer[64];
Quaternion q;
VectorInt16 aa;
VectorInt16 aaReal;
VectorInt16 aaWorld;
VectorFloat gravity;
float euler[3];
float ypr[3];

uint8_t teapotPacket[14] = { '$', 0x02, 0, 0, 0, 0, 0, 0, 0, 0, 0x00, 0x00,
'\r', '\n' };
volatile bool mpuInterrupt = false; // indicates whether MPU interrupt
pin has gone high

void dmpDataReady() {
  mpuInterrupt = true;
}
```

```

void MPU_setup() {
#if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
  Wire.begin();
  Wire.setClock(400000); // 400kHz I2C clock. Comment this line if
having compilation difficulties
#elif I2CDEV_IMPLEMENTATION ==
I2CDEV_BUILTIN_FASTWIRE
  Fastwire::setup(400, true);
#endif

  mpu.initialize();
  devStatus = mpu.dmpInitialize();
  mpu.setXGyroOffset(220);
  mpu.setYGyroOffset(76);
  mpu.setZGyroOffset(-85);
  mpu.setZAccelOffset(1788);

  if (devStatus == 0) {
    mpu.CalibrateAccel(6);
    mpu.CalibrateGyro(6);
    mpu.PrintActiveOffsets();
    Serial.println(F("Enabling DMP..."));
    mpu.setDMPEntered(true);
    mpu.IntStatus = mpu.getIntStatus();
    dmpReady = true;
    packetSize = mpu.dmpGetFIFOpacketSize();
  }
  else {
    Serial.print(F("DMP Initialization failed (code ");
    Serial.print(devStatus);
    Serial.println(F(")"));
  }
}
void MPU() { //rumus penentuan kanan kiri -180 dan +180

```

```

    if (mpu.dmpGetCurrentFIFOPacket(fifoBuffer)) { // Get the Latest
packet
    mpu.dmpGetQuaternion(&q, fifoBuffer);
    mpu.dmpGetGravity(&gravity, &q);
    mpu.dmpGetYawPitchRoll(ypr, &q, &gravity);
    yaw = ypr[0] * 180 / M_PI;
    pitch = (ypr[1] * 180 / M_PI);
    roll = (ypr[2] * 180 / M_PI);
    }
}
void bacaSwitch() {
    LS1 = digitalRead(Sw1);
    LS2 = digitalRead(Sw2);
    LS3 = digitalRead(Sw3);
    LS4 = digitalRead(Sw4);
    LS5 = digitalRead(Sw5);
    LS6 = digitalRead(Sw6);
}

#include <Pixy2.h>
Pixy2 pixy;
uint16_t x, y;
uint16_t deteksi;
uint16_t blocks;
int dataX, dataY;
pixy.init();
void bacapixy() {
    pixy.ccc.getBlocks();
    if (pixy.ccc.numBlocks)
    {
        deteksi = 1;
        x = pixy.ccc.blocks[0].m_x;
        y = pixy.ccc.blocks[0].m_y;
    }
    else {
        deteksi = 0;
    }
}

```

```
x = 0;
y = 0;
}
dataX = map(x, 155, 0, 0, -155);
dataY = map(y, 0, 102, -155, 0);
if (x == 0) dataX = 404;
if (y == 0) dataY = 404;
}
}
```

## LAMPIRAN C

### PROGRAM KESELURUHAN

```
#include <Servo.h>
#include <SoftwareSerial.h>
SoftwareSerial myserial(10, 12);
//Servo-----
int Sgama[6] = { 46, 36, 47, 25, 41, 26 };
int Salpha[6] = { 44, 38, 49, 30, 45, 27 };
int Sbeta[6] = { 42, 40, 48, 23, 43, 28 };
Servo servoG[7],
  servoA[7],
  servoB[7],
  servoARM1, servoARM2, servoARM3, servoARM4, servoARM5,
  servoGrip;

//ParameterRobot-----
float coxa = 4.3;
float femur = 8;
float tibia = 13.5;

float a = 11, b = 11, c = 0,
  A = 0, B = 15;

int langkah = 2;
float p = 0;
float k = 0;
//parameter ARM-----
double carm = 17;
double darm = 16.5;
double farm = sqrt(sq(15) + sq(0.5));
double tarm = sqrt(sq(11) + sq(2.5));
float Xlastarm = 22, Ylastarm = 0, Zlastarm = 30;

//IMU-----
```

```

String myString = " ";
double yaw, pitch, roll;

struct gy25 {
    char buffer[50];
    int counter;
    float heading;
} cmpr;
//PIXY CAM-----
#include <Pixy2.h>
Pixy2 pixy;
uint16_t x, y;
uint16_t deteksi;
uint16_t blocks;
int dataX, dataY;

//PID-----
struct pid {
    double kp;
    double ki;
    double kd;
    double setPoint;
    double error;
    double lastError;
    double ts;
    double outputPID;
    double elapsedTime;
    double integral, derivatif;
};
pid rotasiPitch;
pid rotasiRoll;
//millis-----
struct waktu {
    unsigned long waktuSekarang;
    unsigned long waktuSebelum = 0;
};

```



```

waktu millis1;
waktu millis2;

//inverskinematik-----
struct inverskinematik {
    double finalAlpha;
    double finalGamma;
    double finalBeta;
};
inverskinematik kaki1, kaki2, kaki3, kaki4, kaki5, kaki6;
//Limit_Switch-----
const int Sw1 = 37,
        Sw2 = 39,
        Sw3 = 35,
        Sw4 = 31,
        Sw5 = 33,
        Sw6 = 22;
int LS1, LS2, LS3, LS4, LS5, LS6;

//Bodykinematik-----
float offsetX = 8,
      offsetY = 6.25,
      offsetZ = 4,
      offsetXt = 0,
      offsetYt = 8.2;

float Xbodi = offsetX + a,
      Ybodi = offsetY + b,
      Xtbodi = offsetXt + A,
      Ytbodi = offsetYt + B,
      Zbodi = offsetZ + c;

struct bodiKinematik {
    double XbodiAakhir;
    double YbodiAakhir;
    double ZbodiAakhir;
};

```

```

double mby;
double mbx;
double sudutCy;
double sudutCx;
double sudutAkhir;
};
bodiKinematik kakiKananDepan;
bodiKinematik kakiKiriDepan;
bodiKinematik kakiKananTengah;
bodiKinematik kakiKiriTengah;
bodiKinematik kakiKananBelakang;
bodiKinematik kakiKiriBelakang;
struct trajektoriStabilisasi {
    double Xlast, Ylast, Zlast;
    double acuanX1, acuanX2;
    double acuanY1, acuanY2;
    double acuanZ1, acuanZ2;
    double PX, PY, PZ;
    double rentangx, rentangy, rentangz;
    double nilaiX, nilaiY, nilaiZ;
    bool kondisiKaki = true;
};
trajektoriStabilisasi kakiD1;
trajektoriStabilisasi kakiD2;
trajektoriStabilisasi kakiD3;
trajektoriStabilisasi kakiD4;
trajektoriStabilisasi kakiD5;
trajektoriStabilisasi kakiD6;

float count = 0;
double outputPID, pitchPID, rollPID;
bool kondisiCount = true;
int posisi = 1;
int tunggu = 40;
bool keseimbangan = false,
    arah = false;

```

```

float putarRobot;
int putar, xpixy;
// String dataTerima;
int posisiuji = 0;
int counteruji;
int counMAK = 0;
int br;
String dataTerima;
int penyelamatan = 0;
void setup() {
  Serial.begin(9600);
  myserial.begin(9600);
  Serial.println("----- SetUP -----");
  for (int j = 0; j <= 5; j++) {
    servoG[j + 1].attach(Sgamma[j]);
    servoA[j + 1].attach(Salpha[j]);
    servoB[j + 1].attach(Sbeta[j]);
  }
  servoARM1.attach(15);
  servoARM2.attach(13);
  servoARM3.attach(7);
  servoARM4.attach(5);
  servoARM5.attach(9);
  servoGrip.attach(11);

  pinMode(Sw1, INPUT);
  pinMode(Sw2, INPUT);
  pinMode(Sw3, INPUT);
  pinMode(Sw4, INPUT);
  pinMode(Sw5, INPUT);
  pinMode(Sw6, INPUT);
  //posisi standby-----
  standby();
  standbyARM();
  pixy.init();
  MPU_setup();

```

```

void loop()
bacapixy();
if (myserial.available() > 0) {
  dataTerima = myserial.readStringUntil('\n');
  Serial.println(dataTerima);
}
if (dataTerima == "q") {
  // Bodikinematik(0, 0, 12, 0, 0, 0); //-----Bodikinematik-----
  // majutrajektoriV2(4, 0, 12, 0.02, 10);
  MPU();
  if (roll < -3) {
    br -= 3;
    if (br < -14) br = -14;
    Bodikinematik(0, 0, 12, br, 0, 0);
  }
  if (roll > 3) {
    br += 3;
    if (br > 14) br = 14;
    Bodikinematik(0, 0, 12, br, 0, 0);
  }
  majutrajektoriV2(4, 0, 12, 0.02, 15);
} else if (dataTerima == "w") {
  // Bodikinematik(0, 0, 12, 0, 0, 0); //-----Bodikinematik-----
  // majutrajektoriV2(-4, 0, 12, 0.02, 10);
  MPU();
  if (roll < -3) {
    br -= 3;
    if (br < -14) br = -14;
    Bodikinematik(0, 0, 12, br, 0, 0);
  }
  if (roll > 3) {
    br += 3;
    if (br > 14) br = 14;
    Bodikinematik(0, 0, 12, br, 0, 0);
  }
  majutrajektoriV2(-4, 0, 12, 0.02, 15);
}

```

```

} else if (dataTerima == "e") {
    BodikinematikPutar(0, 0, 12, 0, 0, 0); //-----Bodikinemati Putar-----
    putarBodykinematik(10, 12, 0.02, 15);
} else if (dataTerima == "r") {
    BodikinematikPutar(0, 0, 12, 0, 0, 0); //-----Bodikinemati Putar-----
    putarBodykinematik(-10, 12, 0.02, 15);
} else if (dataTerima == "s") {
    Bodikinematik(0, 0, 12, 0, 0, 0); //-----Bodikinematik-----
    majutrajektoriV2(0, 0, 12, 0.02, 20);
    penyelamatan = 1;
} else if (dataTerima == "o") {
    if (counMAK == 0) {
        inverskinematikARM(22, 0, 30, 0, 0, 1);
        counMAK = 1;
    }
    if (counMAK == 1) {
        ambilkorban();
        penyelamatan = 0;
    }
} else if (dataTerima == "p") {
    if (penyelamatan == 0) {
        if (counMAK == 1) {
            posisiuji = 0;
            inverskinematikARM(22, 0, 30, 0, 0, 0);
            Bodikinematik(0, 0, 12, 0, 0, 0); //-----Bodikinematik-----
            majutrajektoriV2(0, 0, 12, 0.02, 15);
            counMAK = 2;
        }
        if (counMAK == 2) {
            delay(2000);
            standby();
            standbyARMpegang();
            counMAK = 3;
        }
    } else if (penyelamatan == 1) {

```

```

if (counMAK == 3) {
    inverskinematikARM(22, 0, 29, 0, 0, 0);
    delay(2000);
    inverskinematikARM(36, 0, 29, 0, 0, 0);
    delay(2000);
    inverskinematikARM(36, 0, 29, 0, 0, 1);
    delay(3000);
    inverskinematikARM(22, 0, 29, 0, 0, 1);
    delay(2000);
    counMAK = 4;
}
if (counMAK == 4) {
    standby();
    standbyARM();
    counMAK = 0;
}
}
} else if (dataTerima == "t") {
    majuRobot(4, 0, 12, 0.02, 15); //-----maju stabilisasi medan-----
} else if (dataTerima == "y") {
    majuRobot(-4, 0, 12, 0.02, 15); //-----maju stabilisasi medan-----
-
} else if (dataTerima == "u") {
    Bodikinematik(0, 0, 12, 0, 0, 0); //-----Bodikinematik-----
    majutrajektoriV2(0, 2, 12, 0.02, 15);
} else if (dataTerima == "i") {
    Bodikinematik(0, 0, 12, 0, 0, 0); //-----Bodikinematik-----
    majutrajektoriV2(0, -2, 12, 0.02, 15);
}

```

```

float offsetGamma1 = 0, offsetAlpha1 = 0, offsetBeta1 = 5,
    offsetGamma2 = 0, offsetAlpha2 = 0, offsetBeta2 = 0,
    offsetGamma3 = 0, offsetAlpha3 = 0, offsetBeta3 = 10,
    offsetGamma4 = 0, offsetAlpha4 = 0, offsetBeta4 = 10,
    offsetGamma5 = 0, offsetAlpha5 = -5, offsetBeta5 = 0,

```

```

offsetGamma6 = 0, offsetAlpha6 = 0, offsetBeta6 = 0;

void leg1move(float x1, float y1, float z1) {
    float L1 = sqrt(sq(x1) + sq(y1));
    float gamma = atan(x1 / y1) / PI * 180;
    float L2 = sqrt(sq(L1 - coxa) + sq(z1));
    float beta = acos((sq(tibia) + sq(femur) - sq(L2)) / (2 * tibia * femur)) /
    PI * 180;
    // alpha1 = atan((L1 - coxa) / z1) / PI * 180;
    float alpha1 = acos((sq(z1) + sq(L2) - sq(L1-coxa)) / (2 * z1 * L2)) / PI
    * 180;
    float alpha2 = acos((sq(femur) + sq(L2) - sq(tibia)) / (2 * femur * L2)) /
    PI * 180;
    float alpha = alpha1 + alpha2;
    kaki1.finalGamma = (90 - gamma) + offsetGamma1;
    kaki1.finalAlpha = alpha + offsetAlpha1;
    kaki1.finalBeta = (180 - beta) + offsetBeta1;
    servoG[1].write(kaki1.finalGamma);//keluar
    servoA[1].write(kaki1.finalAlpha);
    servoB[1].write(kaki1.finalBeta);//kedalam
}

struct Zkaki {
    double ZX;
    double ZY;
    double R;
};

Zkaki kakiB1;
Zkaki kakiB2;
Zkaki kakiB3;
Zkaki kakiB4;
Zkaki kakiB5;
Zkaki kakiB6;

void Bodikinematik(float tx, float ty, double tz, double rx, double ry,
double rz) {
    kakiKananDepan.XbodiAkhir = Xbodi - tx;

```

kakiKananDepan.YbodiAkhir = Ybodi - ty;

kakiKiriDepan.XbodiAkhir = Xbodi - tx;

kakiKiriDepan.YbodiAkhir = Ybodi + ty;

kakiKananTengah.XbodiAkhir = Xtbodi + tx;

kakiKananTengah.YbodiAkhir = Ytbodi - ty;

kakiKiriTengah.XbodiAkhir = Xtbodi + tx;

kakiKiriTengah.YbodiAkhir = Ytbodi + ty;

kakiKananBelakang.XbodiAkhir = Xbodi + tx;

kakiKananBelakang.YbodiAkhir = Ybodi - ty;

kakiKiriBelakang.XbodiAkhir = Xbodi + tx;

kakiKiriBelakang.YbodiAkhir = Ybodi + ty;

```
kakiB1.R      =      sqrt(sq(kakiKananDepan.XbodiAkhir)      +
sq(kakiKananDepan.YbodiAkhir));
kakiB2.R      =      sqrt(sq(kakiKiriDepan.XbodiAkhir)      +
sq(kakiKiriDepan.YbodiAkhir));
kakiB3.R      =      sqrt(sq(kakiKananTengah.XbodiAkhir)    +
sq(kakiKananTengah.YbodiAkhir));
kakiB4.R      =      sqrt(sq(kakiKiriTengah.XbodiAkhir)     +
sq(kakiKiriTengah.YbodiAkhir));
kakiB5.R      =      sqrt(sq(kakiKananBelakang.XbodiAkhir)  +
sq(kakiKananBelakang.YbodiAkhir));
kakiB6.R      =      sqrt(sq(kakiKiriBelakang.XbodiAkhir)   +
sq(kakiKiriBelakang.YbodiAkhir));
```

if (rz != 0) {

kakiKananDepan.XbodiAkhir = kakiB1.R \* sin(rz / 57.2958);

kakiKananDepan.YbodiAkhir = kakiB1.R \* cos(rz / 57.2958);

kakiKiriDepan.XbodiAkhir = kakiB2.R \* sin(-rz / 57.2958);

kakiKiriDepan.YbodiAkhir = kakiB2.R \* cos(-rz / 57.2958);

kakiKananTengah.XbodiAkhir = kakiB3.R \* sin(rz / 57.2958);



```

kakiKananTengah.YbodiAkhir = kakiB3.R * cos(rz / 57.2958);
kakiKiriTengah.XbodiAkhir = kakiB4.R * sin(-rz / 57.2958);
kakiKiriTengah.YbodiAkhir = kakiB4.R * cos(-rz / 57.2958);
kakiKananBelakang.XbodiAkhir = kakiB5.R * sin(rz / 57.2958);
kakiKananBelakang.YbodiAkhir = kakiB5.R * cos(rz / 57.2958);
kakiKiriBelakang.XbodiAkhir = kakiB6.R * sin(-rz / 57.2958);
kakiKiriBelakang.YbodiAkhir = kakiB6.R * cos(-rz / 57.2958);
}

```

```

kakiB1.ZX = kakiKananDepan.XbodiAkhir * tan(rx / 57.2958);
kakiB1.ZY = kakiKananDepan.YbodiAkhir * tan(ry / 57.2958);

```

```

kakiB2.ZX = kakiKiriDepan.XbodiAkhir * tan(rx / 57.2958);
kakiB2.ZY = kakiKiriDepan.YbodiAkhir * tan(-ry / 57.2958);

```

```

kakiB3.ZX = kakiKananTengah.XbodiAkhir * tan(rx / 57.2958);
kakiB3.ZY = kakiKananTengah.YbodiAkhir * tan(ry / 57.2958);

```

```

kakiB4.ZX = kakiKiriTengah.XbodiAkhir * tan(rx / 57.2958);
kakiB4.ZY = kakiKiriTengah.YbodiAkhir * tan(-ry / 57.2958);

```

```

kakiB5.ZX = kakiKananBelakang.XbodiAkhir * tan(-rx / 57.2958);
kakiB5.ZY = kakiKananBelakang.YbodiAkhir * tan(ry / 57.2958);

```

```

kakiB6.ZX = kakiKiriBelakang.XbodiAkhir * tan(-rx / 57.2958);
kakiB6.ZY = kakiKiriBelakang.YbodiAkhir * tan(-ry / 57.2958);

```

```

kakiKananDepan.ZbodiAkhir = kakiB1.ZX + kakiB1.ZY + tz + Zbodi;
kakiKiriDepan.ZbodiAkhir = kakiB2.ZX + kakiB2.ZY + tz + Zbodi;
kakiKananTengah.ZbodiAkhir = kakiB3.ZX + kakiB3.ZY + tz + Zbodi;
kakiKiriTengah.ZbodiAkhir = kakiB4.ZX + kakiB4.ZY + tz + Zbodi;
kakiKananBelakang.ZbodiAkhir = kakiB5.ZX + kakiB5.ZY + tz +
Zbodi;
kakiKiriBelakang.ZbodiAkhir = kakiB6.ZX + kakiB6.ZY + tz + Zbodi;

```

```

kakiKananDepan.XbodiAkhir -= offsetX;

```

```
kakiKananDepan.YbodiAkhir -= offsetY;  
kakiKananDepan.ZbodiAkhir -= offsetZ;
```

```
kakiKiriDepan.XbodiAkhir -= offsetX;  
kakiKiriDepan.YbodiAkhir -= offsetY;  
kakiKiriDepan.ZbodiAkhir -= offsetZ;
```

```
kakiKananTengah.XbodiAkhir -= offsetXt;  
kakiKananTengah.YbodiAkhir -= offsetYt;  
kakiKananTengah.ZbodiAkhir -= offsetZt;
```

```
kakiKiriTengah.XbodiAkhir -= offsetXt;  
kakiKiriTengah.YbodiAkhir -= offsetYt;  
kakiKiriTengah.ZbodiAkhir -= offsetZt;
```

```
kakiKananBelakang.XbodiAkhir -= offsetX;  
kakiKananBelakang.YbodiAkhir -= offsetY;  
kakiKananBelakang.ZbodiAkhir -= offsetZ;
```

```
kakiKiriBelakang.XbodiAkhir -= offsetX;  
kakiKiriBelakang.YbodiAkhir -= offsetY;  
kakiKiriBelakang.ZbodiAkhir -= offsetZ;
```

```
}
```

```
void majutrajektoriV2(float X, float Y, float Z, float i, float v) {  
    float XFRD, XRRD, XFRT, XRRT, XFRB, XRRB,  
          XFLD, XRLD, XFLT, XRLT, XFLB, XRLB,  
          YFRD, YRRD, YFRT, YRRT, YFRB, YRRB,  
          YFLD, YRLD, YFLT, YRLT, YFLB, YRLB,  
          ZARD, ZLRD, ZALD, ZLLD, ZART, ZLRT,  
          ZALT, ZLLT, ZARB, ZLRB, ZALB, ZLLB,  
          zd, zdl;  
    float at, bt, ct, dt;  
    float acuanX1RD = kakiKananDepan.XbodiAkhir - X / 2;  
    float acuanX2RD = kakiKananDepan.XbodiAkhir + X / 2;  
    float acuanY1RD = kakiKananDepan.YbodiAkhir - Y / 2;  
    float acuanY2RD = kakiKananDepan.YbodiAkhir + Y / 2;
```

float acuanX1LD = kakiKiriDepan.XbodiAkhir - X / 2;  
float acuanX2LD = kakiKiriDepan.XbodiAkhir + X / 2;  
float acuanY1LD = kakiKiriDepan.YbodiAkhir - Y / 2;  
float acuanY2LD = kakiKiriDepan.YbodiAkhir + Y / 2;

float acuanX1RT = kakiKananTengah.XbodiAkhir - X / 2;  
float acuanX2RT = kakiKananTengah.XbodiAkhir + X / 2;  
float acuanY1RT = kakiKananTengah.YbodiAkhir - Y / 2;  
float acuanY2RT = kakiKananTengah.YbodiAkhir + Y / 2;

float acuanX1LT = kakiKiriTengah.XbodiAkhir - X / 2;  
float acuanX2LT = kakiKiriTengah.XbodiAkhir + X / 2;  
float acuanY1LT = kakiKiriTengah.YbodiAkhir - Y / 2;  
float acuanY2LT = kakiKiriTengah.YbodiAkhir + Y / 2;

float acuanX1RB = kakiKananBelakang.XbodiAkhir - X / 2;  
float acuanX2RB = kakiKananBelakang.XbodiAkhir + X / 2;  
float acuanY1RB = kakiKananBelakang.YbodiAkhir - Y / 2;  
float acuanY2RB = kakiKananBelakang.YbodiAkhir + Y / 2;

float acuanX1LB = kakiKiriBelakang.XbodiAkhir - X / 2;  
float acuanX2LB = kakiKiriBelakang.XbodiAkhir + X / 2;  
float acuanY1LB = kakiKiriBelakang.YbodiAkhir - Y / 2;  
float acuanY2LB = kakiKiriBelakang.YbodiAkhir + Y / 2;

float acuanZARD = kakiKananDepan.ZbodiAkhir - Z;  
float acuanZLRD = kakiKananDepan.ZbodiAkhir;

float acuanZALD = kakiKiriDepan.ZbodiAkhir - Z;  
float acuanZLLD = kakiKiriDepan.ZbodiAkhir;

float acuanZART = kakiKananTengah.ZbodiAkhir - Z;  
float acuanZLRT = kakiKananTengah.ZbodiAkhir;

float acuanZALT = kakiKiriTengah.ZbodiAkhir - Z;

```
float acuanZLLT = kakiKiriTengah.ZbodiAkhir;
```

```
float acuanZARB = kakiKananBelakang.ZbodiAkhir - Z;
```

```
float acuanZLRB = kakiKananBelakang.ZbodiAkhir;
```

```
float acuanZALB = kakiKiriBelakang.ZbodiAkhir - Z;
```

```
float acuanZLLB = kakiKiriBelakang.ZbodiAkhir;
```

```
int j = 1 / i;
```

```
int counterJ = j/2;
```

```
// Serial.print(kakiKananDepan.ZbodiAkhir);
```

```
// Serial.print("");
```

```
// Serial.println(acuanZARD);
```

```
k = 0;
```

```
for(k = 0 ;k < j + 1; k++) {
```

```
    if (p > 1) {
```

```
        p = 0;
```

```
    }
```

```
    at = pow((1 - p), 3);
```

```
    bt = 3 * p * pow((1 - p), 2);
```

```
    ct = 3 * pow(p, 2) * (1 - p);
```

```
    dt = pow(p, 3);
```

```
    XFRD = (kakiKananDepan.XbodiAkhir - X) * at + acuanX1RD * bt  
+ acuanX2RD * ct + (kakiKananDepan.XbodiAkhir + X) * dt;
```

```
    XRRD = (kakiKananDepan.XbodiAkhir + X) * at + acuanX2RD * bt  
+ acuanX1RD * ct + (kakiKananDepan.XbodiAkhir - X) * dt;
```

```
    YFRD = (kakiKananDepan.YbodiAkhir - Y) * at + acuanY1RD * bt  
+ acuanY2RD * ct + (kakiKananDepan.YbodiAkhir + Y) * dt;
```

```
    YRRD = (kakiKananDepan.YbodiAkhir + Y) * at + acuanY2RD * bt  
+ acuanY1RD * ct + (kakiKananDepan.YbodiAkhir - Y) * dt;
```

```
    XFLD = (kakiKiriDepan.XbodiAkhir - X) * at + acuanX1LD * bt +  
acuanX2LD * ct + (kakiKiriDepan.XbodiAkhir + X) * dt;
```

```
    XRLD = (kakiKiriDepan.XbodiAkhir + X) * at + acuanX2LD * bt +  
acuanX1LD * ct + (kakiKiriDepan.XbodiAkhir - X) * dt;
```

YFLD = (kakiKiriDepan.YbodiAkhir - Y) \* at + acuanY1LD \* bt + acuanY2LD \* ct + (kakiKiriDepan.YbodiAkhir + Y) \* dt;

YRLD = (kakiKiriDepan.YbodiAkhir + Y) \* at + acuanY2LD \* bt + acuanY1LD \* ct + (kakiKiriDepan.YbodiAkhir - Y) \* dt;

XFRT = (kakiKananTengah.XbodiAkhir - X) \* at + acuanX1RT \* bt + acuanX2RT \* ct + (kakiKananTengah.XbodiAkhir + X) \* dt;

XRRT = (kakiKananTengah.XbodiAkhir + X) \* at + acuanX2RT \* bt + acuanX1RT \* ct + (kakiKananTengah.XbodiAkhir - X) \* dt;

YFRT = (kakiKananTengah.YbodiAkhir - Y) \* at + acuanY1RT \* bt + acuanY2RT \* ct + (kakiKananTengah.YbodiAkhir + Y) \* dt;

YRRT = (kakiKananTengah.YbodiAkhir + Y) \* at + acuanY2RT \* bt + acuanY1RT \* ct + (kakiKananTengah.YbodiAkhir - Y) \* dt;

XFLT = (kakiKiriTengah.XbodiAkhir - X) \* at + acuanX1LT \* bt + acuanX2LT \* ct + (kakiKiriTengah.XbodiAkhir + X) \* dt;

XRLT = (kakiKiriTengah.XbodiAkhir + X) \* at + acuanX2LT \* bt + acuanX1LT \* ct + (kakiKiriTengah.XbodiAkhir - X) \* dt;

YFLT = (kakiKiriTengah.YbodiAkhir - Y) \* at + acuanY1LT \* bt + acuanY2LT \* ct + (kakiKiriTengah.YbodiAkhir + Y) \* dt;

YRLT = (kakiKiriTengah.YbodiAkhir + Y) \* at + acuanY2LT \* bt + acuanY1LT \* ct + (kakiKiriTengah.YbodiAkhir - Y) \* dt;

XFRB = (kakiKananBelakang.XbodiAkhir - X) \* at + acuanX1RB \* bt + acuanX2RB \* ct + (kakiKananBelakang.XbodiAkhir + X) \* dt;

XR RB = (kakiKananBelakang.XbodiAkhir + X) \* at + acuanX2RB \* bt + acuanX1RB \* ct + (kakiKananBelakang.XbodiAkhir - X) \* dt;

YFRB = (kakiKananBelakang.YbodiAkhir - Y) \* at + acuanY1RB \* bt + acuanY2RB \* ct + (kakiKananBelakang.YbodiAkhir + Y) \* dt;

YR RB = (kakiKananBelakang.YbodiAkhir + Y) \* at + acuanY2RB \* bt + acuanY1RB \* ct + (kakiKananBelakang.YbodiAkhir - Y) \* dt;

XFLB = (kakiKiriBelakang.XbodiAkhir - X) \* at + acuanX1LB \* bt + acuanX2LB \* ct + (kakiKiriBelakang.XbodiAkhir + X) \* dt;

XRLB = (kakiKiriBelakang.XbodiAkhir + X) \* at + acuanX2LB \* bt + acuanX1LB \* ct + (kakiKiriBelakang.XbodiAkhir - X) \* dt;

YFLB = (kakiKiriBelakang.YbodiAkhir - Y) \* at + acuanY1LB \* bt  
+ acuanY2LB \* ct + (kakiKiriBelakang.YbodiAkhir + Y) \* dt;

YRLB = (kakiKiriBelakang.YbodiAkhir + Y) \* at + acuanY2LB \* bt  
+ acuanY1LB \* ct + (kakiKiriBelakang.YbodiAkhir - Y) \* dt;

ZARD = kakiKananDepan.ZbodiAkhir \* at + acuanZARD \* bt +  
acuanZARD \* ct + kakiKananDepan.ZbodiAkhir \* dt;

ZLRD = kakiKananDepan.ZbodiAkhir \* at + acuanZLRD \* bt +  
acuanZLRD \* ct + kakiKananDepan.ZbodiAkhir \* dt;

ZALD = kakiKiriDepan.ZbodiAkhir \* at + acuanZALD \* bt +  
acuanZALD \* ct + kakiKiriDepan.ZbodiAkhir \* dt;

ZLLD = kakiKiriDepan.ZbodiAkhir \* at + acuanZLLD \* bt +  
acuanZLLD \* ct + kakiKiriDepan.ZbodiAkhir \* dt;

ZART = kakiKananTengah.ZbodiAkhir \* at + acuanZART \* bt +  
acuanZART \* ct + kakiKananTengah.ZbodiAkhir \* dt;

ZLRT = kakiKananTengah.ZbodiAkhir \* at + acuanZLRT \* bt +  
acuanZLRT \* ct + kakiKananTengah.ZbodiAkhir \* dt;

ZALT = kakiKiriTengah.ZbodiAkhir \* at + acuanZALT \* bt +  
acuanZALT \* ct + kakiKiriTengah.ZbodiAkhir \* dt;

ZLLT = kakiKiriTengah.ZbodiAkhir \* at + acuanZLLT \* bt +  
acuanZLLT \* ct + kakiKiriTengah.ZbodiAkhir \* dt;

ZARB = kakiKananBelakang.ZbodiAkhir \* at + acuanZARB \* bt +  
acuanZARB \* ct + kakiKananBelakang.ZbodiAkhir \* dt;

ZLRB = kakiKananBelakang.ZbodiAkhir \* at + acuanZLRB \* bt +  
acuanZLRB \* ct + kakiKananBelakang.ZbodiAkhir \* dt;

ZALB = kakiKiriBelakang.ZbodiAkhir \* at + acuanZALB \* bt +  
acuanZALB \* ct + kakiKiriBelakang.ZbodiAkhir \* dt;

ZLLB = kakiKiriBelakang.ZbodiAkhir \* at + acuanZLLB \* bt +  
acuanZLLB \* ct + kakiKiriBelakang.ZbodiAkhir \* dt;

//Serial.println(langkah);

bacaSwitch();

```

switch (langkah) {
  case 2:
    leg1move(XFRD, YFRD, ZARD);
    leg2move(XRLD, YFLD, ZLLD);
    leg3move(XRRT, YRRT, ZLRT);
    leg4move(XFLT, YRLT, ZALT);
    leg5move(XRRB, YFRB, ZARB);
    leg6move(XFLB, YFLD, ZLLB);
    break;
  case 3:
    leg1move(XRRD, YRRD, ZLRD);
    leg2move(XFLD, YRLD, ZALD);
    leg3move(XFRT, YFRT, ZART);
    leg4move(XRLT, YFLT, ZLLT);
    leg5move(XFRB, YRRB, ZLRB);
    leg6move(XRLB, YRLB, ZALB);
    break;
}
p += i;

delay(v);
}
langkah++;
if (langkah > 3) langkah = 2;
}

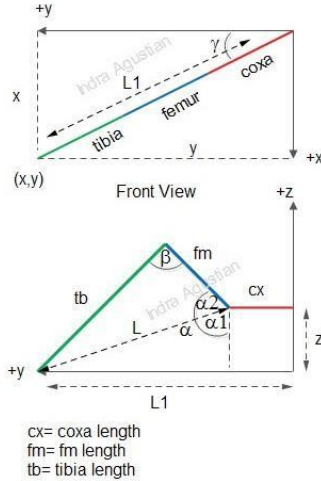
```





## LAMPIRAN D

### CONTOH PERHITUNGAN *INVERS KINEMATIC*



Gambar diatas merupakan ilustrasi *invers kinematic* sebagai parameter acuan perhitungan *invers kinematic*. Pada perhitungan *invers kinematic* digunakan untuk mencari sudut gamma, alpha, dan beta berdasarkan inputan koordinat (x, y, z) dan konstanta coxa, femur dan tibia. Pada contoh perhitungan ini mengikuti panjang coxa, femur, dan tibia pada kaki robot yang sebenarnya untuk membuktikan keakuratan perhitungan *invers kinematic*. Pada robot yang sebenarnya panjang coxa adalah 4.3 cm, panjang femur 8 cm, dan panjang tibia adalah 13.5 cm. Pada perhitungan *invers kinematic* ini menggunakan inputan koordinat x sebesar 10 cm, y sebesar 10 cm dan z sebesar 10 cm. Berikut contoh perhitungan dari *invers kinematic*.

$$\begin{aligned}
 L1 &= \sqrt{x^2 + y^2} \\
 L1 &= \sqrt{10^2 + 10^2} \\
 &= 10\sqrt{2} \\
 \gamma &= \tan^{-1} \frac{x}{y}
 \end{aligned}$$

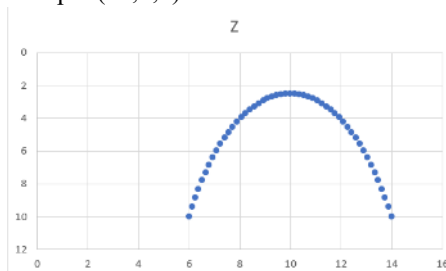
$$\begin{aligned}
\gamma &= \tan^{-1} \frac{10}{10} \\
&= 45^0 \\
L &= \sqrt{(L1 - cx)^2 + z^2} \\
&= \sqrt{(10\sqrt{2} - 4.3)^2 + 10^2} \\
&= 14.03 \\
\beta &= \cos^{-1}((tb^2 + fm^2 - L^2)/(2 \times tb \times fm)) \\
&= \cos^{-1}((13.5^2 + 8^2 - 14.03^2)/(2 \times 13.5 \times 8)) \\
&= 76.78^0 \\
\alpha 1 &= \arccos \left( \frac{z}{L} \right) \\
&= \arccos \left( \frac{10}{14.03} \right) \\
&= 44.54^0 \\
\alpha 2 &= \cos^{-1}((fm^2 + L^2 - tb^2)/(2 \times fm \times L)) \\
&= \cos^{-1}((8^2 + 14.03^2 - 13.5^2)/(2 \times 8 \times 14.03)) \\
&= 69.50^0 \\
a &= \alpha 1 + \alpha 2 \\
&= 44.54 + 69.50 \\
&= 114.04^0
\end{aligned}$$

Berdasarkan perhitungan *invers kinematic* dengan inputan koordinat (10,10,10) didapatkan sudut gamma sebesar 45°, alpha sebesar 114.04° dan beta sebesar 76.78°. berdasarkan contoh perhitungan tersebut terbukti akurat setelah diterapkan pada robot seperti pada pengujian *invers kinematic*.

### **CONTOH PERHITUNGAN TRAYEKTORI**

Perhitungan trayektori digunakan untuk menghasilkan lintasan parabola berdasarkan inputan 4 titik koordinat dan banyaknya jumlah titik. Cara kerja dari trayektori adalah dengan memasukan 4 titik koordinat inputan dan banyaknya jumlah titik yang akan membentuk lintasan ke dalam persamaan polinomial sehingga menghasilkan titik-titik koordinat yang membentuk lintasan dan inputan banyaknya jumlah titik menentukan banyaknya jumlah titik koordinat yang membentuk lintasan. Berikut merupakan gambaran lintasan trayektori berdasarkan inputan 4

titik koordinat, titik pertama (6,0,8), titik kedua (8,0,2), titik ketiga (12,0,2), dan titik keempat (14,0,8).



berdasarkan empat titik koordinat yang membentuk lintasan seperti pada gambar di atas akan dibuat contoh perhitungan untuk membuktikan perhitungan trayektori. Berikut merupakan contoh perhitungan dari trayektori.

$t$  = merupakan faktor literasi persamaan polinomial yang nilainya dari 0 hingga 1 sehingga untuk menentukan banyaknya titik yang membentuk lintasan perlu membagi angka 1 dengan banyaknya jumlah titik pada lintasan. Pada robot jumlah titik yang diinputkan adalah 50 titik.

$$\begin{aligned} t &= 1/\text{banyaknya titik} \\ &= 1/50 \\ &= 0.02 \end{aligned}$$

$$P(t)_{x,y,z} = (1-t)^3 P1_{x,y,z} + 3t(1-t)^2 P2_{x,y,z} + 3t^2(1-t) P3_{x,y,z} + t^3 P4_{x,y,z}$$

$$\begin{aligned} P(0)_x &= (1-0)^3 6 + 3(0)(1-0)^2 8 + 3(0)^2(1-0.02) 12 + 0^3 14 \\ &= 6 \end{aligned}$$

$$\begin{aligned} P(0)_y &= (1-0)^3 0 + 3(0)(1-0)^2 0 + 3(0)^2(1-0.02) 0 + 0^3 0 \\ &= 0 \end{aligned}$$

$$\begin{aligned} P(0)_z &= (1-0)^3 8 + 3(0)(1-0)^2 2 + 3(0)^2(1-0.02) 2 + 0^3 8 \\ &= 8 \end{aligned}$$

Berdasarkan perhitungan diatas pada titik lintasan pertama didapat koordinat x,y, dan z sebesar (6, 0, 8). Berikutnya perhitungan untuk mencari koordinat pada titik kedua pada lintasan dengan nilai t bertambah 0.02.

$$P(0.02)_x = (1 - 0.02)^3 6 + 3(0.02)(1 - 0.02)^2 8 + 3(0.02)^2 (1 - 0.02) 12 + 0.02^3 14$$

$$= 6.12$$

$$P(0.02)_y = (1 - 0.02)^3 0 + 3(0.02)(1 - 0.02)^2 0 + 3(0.02)^2 (1 - 0.02) 0 + 0.02^3 0$$

$$= 0$$

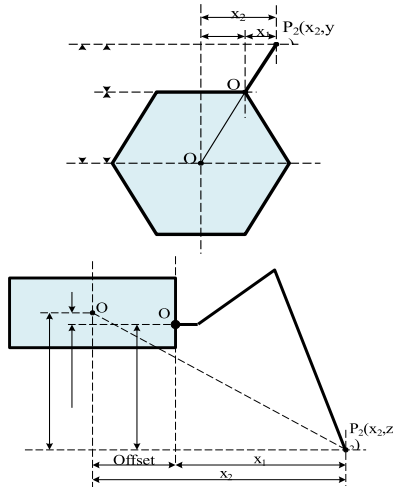
$$P(0.02)_z = (1 - 0.02)^3 8 + 3(0.02)(1 - 0.02)^2 2 + 3(0.02)^2 (1 - 0.02) 2 + 0.02^3 8$$

$$= 7.65$$

Berdasarkan perhitungan diatas pada titik lintasan kedua didapat koordinat x,y, dan z sebesar (6.12, 0, 7.65).

### **CONTOH PERHITUNGAN *BODY KINEMATIC***

Pada perhitungan body kinematic digunakan untuk mengintegrasikan koordinat kaki robot terhadap koordinat bodi robot. Pada siklus langkah robot terdapat koordinat standby yang digunakan sebagai koordinat awal sebelum robot melangkah, pada penerapan body kinematic koordinat standby tersebut diubah berdasarkan koorniat bodi robot dengan menambahkan nilai offset bodi robot pada koordinat standby. Berikut merupakan gambar ilustrasi dari integrasi koordinat standby dengan koordinat bodi robot.



Pada robot yang sebenarnya nilai offset pada bodi robot terbagi menjadi 2 yaitu offset pada kaki tengah dan offset pada kaki depan atau belakang. Pada robot offset x pada kaki tengah sebesar 0 cm, offset y kaki tengah sebesar 8.2 cm, dan offset z sebesar 4 cm, pada offset x pada kaki depan sebesar 8 cm dan offset y kaki depan 6.25 cm. Berikut perhitungan untuk mengintegrasikan koordinat standby dengan koordinat bodi robot.

$$x_{\text{standby\_bodiRobot}} = x_{\text{standby\_kaki}} + \text{offset } x$$

$$\begin{aligned} x_{\text{standby\_bodiRobot\_kakiDepan}} &= 10 + 8 \\ &= 18 \end{aligned}$$

$$\begin{aligned} x_{\text{standby\_bodiRobot\_kakiTengah}} &= 0 + 0 \\ &= 0 \end{aligned}$$

$$y_{\text{standby\_bodiRobot}} = y_{\text{standby\_kaki}} + \text{offset } y$$

$$\begin{aligned} y_{\text{standby\_bodiRobot\_kakiDepan}} &= 10 + 6.25 \\ &= 16.25 \end{aligned}$$

$$\begin{aligned} y_{\text{standby\_bodiRobot\_kakiTengah}} &= 15 + 8.2 \\ &= 23.2 \end{aligned}$$

$$Z_{\text{standby\_bodiRobot}} = Z_{\text{standby\_kaki}} + \textit{offset z}$$

$$\begin{aligned} Z_{\text{standby\_bodiRobot\_kakiDepan}} &= 8 + 4 \\ &= 12 \end{aligned}$$

$$\begin{aligned} Z_{\text{standby\_bodiRobot\_kakiTengah}} &= 8 + 4 \\ &= 12 \end{aligned}$$