LAMPIRAN

```
#include <MsTimer2.h>
#include <LiquidCrystal.h>
#include <EEPROM.h>
#include <SoftwareSerial.h>

SoftwareSerial Serial4(A15, 51); // RX, TX
SoftwareSerial Serial5(50, 52); // RX, TX

LiquidCrystal lcd(34, 36, 38, 40, 42, 44);

#define sw6 41
#define sw5 33
#define sw4 35
#define sw3 37
#define sw2 39
#define sw1 31

#define pwmL1 11
#define pwmL2 10
#define pwmR1 5
#define pwmR2 4
#define pwmB1 7
#define pwmB2 6
#define pwmD1 13
#define pwmD2 12

#define kick 23
#define kick1 25
#define intPinL 3
#define intPinR 20
#define intPinB 21
#define digPinL 30
#define digPinR 26
#define digPinB 28
#define addTeam 4
float const maxRPM = 530;
float const rRobot = 0.185; //18.5 cm
float const dRobot = 0.37; //37 cm
float const perRotasi = 270; //270 pulsa
float const dRoda = 10; //10 cm
float const kelRoda = (dRoda * 22 / 7);
float const jarPerPulsa = kelRoda / perRotasi;

float   rpmL, rpmR, rpmB, pwmL, pwmR, pwmB, jarL, jarR, jarB, jarL1, jarR1, jarB1,
kecL, kecR, kecB, errorL, errorR, errorB, Theta, rTheta, T, rotpsL, rotpsR, dataGyro,
        rpsL, rpsR, rpsB, error, error1, lastError, PID, mmpsL, mmpsR, mmpsB,
errorL1, errorR1, errorB1, errorIL, errorIR, errorIB, errorIB1, radpsL, radpsR,
        vL, vR, vB, vX, vY, vT, iX, iY, iT, derajat, errorX, errorY, errorT, xTarget,
yTarget, tTarget;
float jarX = 0, jarY = 0;

String inString = "";
int xBola = -1, yBola = -1, xBola1, yBola1, xBolaH, yBolaH, m, mm, titikTuju,
thetaNow, tengahX, tengahY, countPos, kir, kan, bel;
float dBola, rBola;
char data;
long count, L, R, B, intL, intR, intB, frekL, frekR, frekB;
int spX = 220, spY = 240; //spX=400, spY=430;
int posisiBola, detekBola = 0, countRun, countTendang, countManual, detikManual,
detikRun, countDetek, countGaDetek;
byte  pilihTendang = 1, enablePosisioning = 'N', enablePosisioning1 = 'N', enableMain
= 'N', enableGyro = 1, strategi = 1, strategiKick = 1, dapatBola = 'N', intruksi =
'S',
      intruksiFull  =  'S',  intruksiR  =  'S',  intruksiPlay  =  'K',  team,
enableReadThetaNow, runIntruksi = 'Y', enableDetek = 'N', bolehMain = 'N', lihatBola
= 'N', lihatBolaRoso = 'N',
      koreksiBola = 'N', sudahMundur = 'N', enableKickOff = 'N', enablePenalty =
'N', enableInisial = 'N';
unsigned char timeSerang, timeTengah, timeBlkang;

const char *tampilStart[10] = {
```

```
  "Auto Play C ",
  "Auto Play M ",
  "Full Auto C ",
  "Full Auto M ",
  "Mnual Serang",
  "Mnual Tengah",
  "Mnual Blkang",
};

const char *tampilTest[11] = {
  "PID Bola  ",
  "PID BolaXY",
  "Kinematik ",
  "Kick1     ",
  "Kick2     ",
  "keDepan   ",
  "keDepan1  ",
  "keBelakang",
  "Manual    ",
  "Thesis    ",
};

void setup() {
  pinMode(pwmL1, OUTPUT);
  pinMode(pwmL2, OUTPUT);
  pinMode(pwmR1, OUTPUT);
  pinMode(pwmR2, OUTPUT);
  pinMode(pwmB1, OUTPUT);
  pinMode(pwmB2, OUTPUT);
  pinMode(pwmD1, OUTPUT);
  pinMode(pwmD2, OUTPUT);
  pinMode(sel1, OUTPUT);
  pinMode(sel2, OUTPUT);
  pinMode(sel3, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(8, OUTPUT);
  digitalWrite(9, 1);
  digitalWrite(10, 1);

  MsTimer2::set(100, interrupt0);
  MsTimer2::start();
  attachInterrupt(digitalPinToInterrupt(intPinL), interruptL, CHANGE);
  attachInterrupt(digitalPinToInterrupt(intPinR), interruptR, CHANGE);
  attachInterrupt(digitalPinToInterrupt(intPinB), interruptB, CHANGE);

  lcd.begin(20, 4);
  Serial.begin(9600);
  Serial1.begin(9600);
  Serial2.begin(9600);
  Serial3.begin(38400);
  Serial4.begin(9600);
  Serial5.begin(9600);

  pinMode(sw1, INPUT_PULLUP);
  pinMode(sw2, INPUT_PULLUP);
  pinMode(sw3, INPUT_PULLUP);
  pinMode(sw4, INPUT_PULLUP);
  pinMode(sw5, INPUT_PULLUP);
  pinMode(sw6, INPUT_PULLUP);

  pinMode(intPinL, INPUT_PULLUP);
  pinMode(intPinR, INPUT_PULLUP);
  pinMode(intPinB, INPUT_PULLUP);
  pinMode(digPinL, INPUT_PULLUP);
  pinMode(digPinR, INPUT_PULLUP);
  pinMode(digPinB, INPUT_PULLUP);

  motor(0, 0, 0);
  team = EEPROM.read(addTeam);
}
```

```
byte baruReset = 'Y';

void loop() {
  if (baruReset == 'Y') {
    jarX = 0; jarY = 0;

    Serial2.print(String("*?,") + (int)jarX + String(",") + (int)jarY + String(",")
+ (int)Theta + String(",") + (int)lihatBola + String(",\n"));
    Serial2.print(String("*!,") + (int)xTarget + String(",") + (int)yTarget +
String(",") + (int)tTarget + String(",\n"));
    if (team == 'C' || team == 'M') {runAutoPlay();}
    else if (team == 'c' || team == 'm') {runFullAuto();}

    motor(0, 0, 0);
  }
  baruReset = 'N';

  if(Theta < 0){rTheta = 360 + Theta;}
  else{rTheta = Theta;}

  parseBola();

  lcd.setCursor(0, m); lcd.print(">");

  lcd.setCursor(1, 0); lcd.print("Start:"); lcd.print(tampilStart[mm]);
  lcd.setCursor(1,   1);   lcd.print("Gyro:");   lcd.print(Theta);   lcd.print("
");lcd.print((int)rTheta); lcd.print("  ");
  lcd.setCursor(1, 2); lcd.print("Setting ");
  lcd.setCursor(1, 3); lcd.print("Sensor x"); lcd.print(xBola); lcd.print(" y");
lcd.print(yBola); lcd.print("  ");

  if (U()) {lcd.setCursor(0, m); lcd.print(" "); m--; delay(150); }
  if (D()) { lcd.setCursor(0, m); lcd.print(" "); m++; delay(150); }
  if (P() && m == 0) { mm--; delay(150); }
  if (M() && m == 0) { mm++; delay(150); }

  if (m < 0) { m = 3; }
  if (m > 3) { m = 0; }
  if (mm < 2) { mm = 3; }
  if (mm > 3) { mm = 2; }

  if (C() && m == 1) { Serial3.println("0#"); jarX = jarY = jarL = jarR = jarB = 0;
}
  else if (C()) { dribble(-255); delay(200);  motor(0, 0, 0); }

  if (O() && m == 2) { setting(); }
  if (O() && m == 3) { cekSensor(); }

  if (O() && m == 0 && mm == 0) { team = 'C'; EEPROM.write(addTeam, team);
runAutoPlay(); }
  if (O() && m == 0 && mm == 1) { team = 'M'; EEPROM.write(addTeam, team);
runAutoPlay(); }
  if (O() && m == 0 && mm == 2) { team = 'c'; EEPROM.write(addTeam, team);
runFullAuto(); }
  if (O() && m == 0 && mm == 3) { team = 'm'; EEPROM.write(addTeam, team);
runFullAuto(); }
  if (O() && m == 0 && mm == 4) { strategi = 0; runManualPlay(); }
  if (O() && m == 0 && mm == 5) { strategi = 1; runManualPlay(); }
  if (O() && m == 0 && mm == 6) { strategi = 2; runManualPlay(); }

  count++;
  if (count > 20) {
    Serial2.print(String("*?,") + (int)jarX + String(",") + (int)jarY + String(",")
+ (int)Theta + String(",") + (int)lihatBola + String(",\n"));
    Serial2.print(String("*!,") + (int)xTarget + String(",") + (int)yTarget +
String(",") + (int)tTarget + String(",\n"));
    count = 0;
  }
  listenCoach();
}
```

```
void interrupt0() {
  jarL = jarPerPulsa * frekL;
  jarR = jarPerPulsa * frekR;
  jarB = jarPerPulsa * frekB;

  rpmL = frekL * 600 / perRotasi;
  rpmR = frekR * 600 / perRotasi;
  rpmB = frekB * 600 / perRotasi;

  rpsL=rpmL*0.104719755;
  rpsR=rpmR*0.104719755;
  rpsB=rpmB*0.104719755;

  updateJarak();

  frekL = frekR = frekB = 0;

  countRun++;
  if (countRun > 10) {
    detikRun++;
    countRun = 0;
  }

  countManual++;
  countDetek++;
  countGaDetek++;
  countTendang++;
}
void interruptL() {
  if (digitalRead(intPinL) != digitalRead(digPinL)) { intL++; frekL++;}
  else {intL--; frekL--;}
}
void interruptR() {
  if (digitalRead(intPinR) != digitalRead(digPinR)) { intR++; frekR++;}
  else { intR--; frekR--; }
}
void interruptB() {
  if (digitalRead(intPinB) != digitalRead(digPinB)) { intB++; frekB++;
  }
  else { intB--; frekB--; }
}

bool O() {
  if (digitalRead(sw1) == 0) { return true;}
  else {return false;}
}
bool D() {
  if (digitalRead(sw2) == 0) { return true; }
  else { return false; }
}
bool U() {
  if (digitalRead(sw3) == 0) { return true; }
  else { return false; }
}
bool P() {
  if (digitalRead(sw4) == 0) { return true; }
  else { return false; }
}
bool M() {
  if (digitalRead(sw5) == 0) { return true; }
  else { return false; }
}
bool C() {
  if (digitalRead(sw6) == 0) { return true; }
  else { return false; }
}

void motor(int L, int R, int B) {
  if (L > 0) {
    analogWrite(pwmL1, L);
    digitalWrite(pwmL2, 0);
```

```
  } else if (L < 0) {
    analogWrite(pwmL1, 255 + L);
    digitalWrite(pwmL2, 1);
  } else {
    digitalWrite(pwmL1, 0);
    digitalWrite(pwmL2, 0);
  }

  if (R > 0) {
    analogWrite(pwmR1, R);
    digitalWrite(pwmR2, 0);
  } else if (R < 0) {
    analogWrite(pwmR1, 255 + R);
    digitalWrite(pwmR2, 1);
  } else {
    digitalWrite(pwmR1, 0);
    digitalWrite(pwmR2, 0);
  }

  if (B > 0) {
    analogWrite(pwmB1, B);
    digitalWrite(pwmB2, 0);
  } else if (B < 0) {
    analogWrite(pwmB1, 255 + B);
    digitalWrite(pwmB2, 1);
  } else {
    digitalWrite(pwmB1, 0);
    digitalWrite(pwmB2, 0);
  }
  pwmL = L;
  pwmR = R;
  pwmB = B;
}

void majuGyro(int L, int R, int yaw) {
  float kp = 2, ki = 0, kd = 3 , PID;

  int error = yaw - Theta;

  PID = (kp * error) + (kd * (error - error1));

  error1 = error;
}

void parseGyro() {
  int dataG;
  while (Serial3.available() > 0) {
    int inChar = Serial3.read();

    if (inChar != '\n') {
      inString += (char)inChar;
    }
    else {
      dataGyro = inString.toFloat();
      inString = "";
    }
  }
}

void runXYPwm(int X, int Y, int W, int maxx) {

  vL = -(sin(radians(150)) * X) + (cos(radians(150)) * Y) + (W);
  vR = -(sin(radians(30)) * X) + (cos(radians(30)) * Y) + (W);
  vB = -(sin(radians(270)) * X) + (cos(radians(270)) * Y) + (W);

  if (vL > maxx) {
    vL = maxx;
  }
  if (vR > maxx) {
    vR = maxx;
  }
```

```
  if (vB > maxx) {
    vB = maxx;
  }
  if (vL < -maxx) {
    vL = -maxx;
  }
  if (vR < -maxx) {
    vR = -maxx;
  }
  if (vB < -maxx) {
    vB = -maxx;
  }

  motor(vL, vR, vB);
}

//peke 3 rotary
void updateJarak() {
  float vX, vY, w;
  float offsetX = 0.98, offsetY = 1.12;

  vX = ((2*rpsB - rpsL - rpsR)/6.85);
  vY = ((sqrt(3)*rpsR - sqrt(3)*rpsL)/6.85);

  jarX += (cos(radians(Theta)) * vX + sin(radians(Theta)) * vY) * offsetX;
  jarY += (-sin(radians(Theta)) * vX + cos(radians(Theta)) * vY) * offsetY;

  parseGyro();
  if (dataGyro < 0) {
    Theta = 360 + dataGyro;
  } else {
    Theta = dataGyro;
  }
  if (Theta > 180) {
    Theta = Theta - 360;
  }
}

void displayJarak() {
  lcd.setCursor(0, 1);
  lcd.print("X"); lcd.print((int)jarX);
  lcd.print(" Y"); lcd.print((int)jarY);
  lcd.print(" T"); lcd.print((int)Theta); lcd.print("   ");
}

void runKinematik(int x, int y, int t, int maxPWM) {
  float kpXY = 3.3, kiXY = 0.095, kdXY = 0;
  float kpT = 2.8, kiT = 0.07, kdT = 0, maxRPM = maxPWM;
  float eTheta, pTheta,mTheta, batasPlus, batasMin;

  errorX = x - jarX;
  errorY = y - jarY;

  if(Theta < 0){pTheta = 360 + Theta;}
  else{pTheta = Theta;}
  mTheta = Theta;

  if(t<0){ eTheta = 360 + t; }
  else{ eTheta = t; }

  if(pTheta > eTheta){errorT = pTheta - eTheta;}
  else {errorT = Theta - t;}

  if(errorT>180){errorT =(int) (errorT - 360);}

  long r = sqrt(((errorX / 10) * (errorX / 10)) + ((errorY / 10) * (errorY / 10)));
  r *= 10;

  if (r < 15) {
    iX += errorX;
    iY += errorY;
```

```
    if (errorT < 15 && errorT > -15) { iT += errorT; }
  }else{
    iX=iY=iT=0;
  }

  vX = errorX * kpXY + (iX * kiXY);
  vY = errorY * kpXY + (iY * kiXY);
  vT = errorT * kpT + (iT * kiT);

  if(vX>0 && vY>0){ //Kuadran I
    if(vX>maxPWM && vX>vY){
      vY=maxPWM*vY/vX;
      vX=maxPWM;
    }else if(vY>maxPWM && vY>vX){
      vX=maxPWM*vX/vY;
      vY=maxPWM;
    }
  }else if(vX<0 && vY>0){ //Kuadran II
    if(-vX>maxPWM && -vX>vY){
      vY=maxPWM*vY/-vX;
      vX=-maxPWM;
    }else if(vY>maxPWM && vY>-vX){
      vX=maxPWM*vX/vY;
      vY=maxPWM;
    }
  }else if(vX>0 && vY<0){ //Kuadran III
    if(vX>maxPWM && vX>-vY){
      vY=maxPWM*vY/vX;
      vX=maxPWM;
    }else if(-vY>maxPWM && -vY>vX){
      vX=maxPWM*vX/-vY;
      vY=-maxPWM;
    }
  }else if(vX<0 && vY<0){ //Kuadran IV
    if(-vX>maxPWM && -vX>-vY){
      vY=maxPWM*vY/-vX;
      vX=-maxPWM;
    }else if(-vX>maxPWM && -vY>-vX){
      vX=maxPWM*vX/-vY;
      vY=-maxPWM;
    }
  }

  vB = ((vY) * (-sin(radians(Theta)))) + ((vX) * (cos(radians(Theta)))) + ((vT /
3));
  vL = ((vY) * (-sin(radians(60 - Theta)))) + ((vX) * (-cos(radians(60 - Theta))))
+ ((vT / 3));
  vR = ((vY) * ( sin(radians(60 + Theta)))) + ((vX) * (-cos(radians(60 + Theta))))
+ ((vT / 3));

  if (vL > maxRPM) {
    vL = maxRPM;
  }
  if (vR > maxRPM) {
    vR = maxRPM;
  }
  if (vB > maxRPM) {
    vB = maxRPM;
  }
  if (vL < -maxRPM) {
    vL = -maxRPM;
  }
  if (vR < -maxRPM) {
    vR = -maxRPM;
  }
  if (vB < -maxRPM) {
    vB = -maxRPM;
  }

  if ((errorX < 4 && errorX > -4) && (errorY < 4 && errorY > -4) && (errorT < 2 &&
errorT > -2)) {
```

```
    motor(0, 0, 0);
    iX = iY = iT = 0;
    titikTuju = 1;
  }
  else {
    motor(vL, vR, vB);
    titikTuju = 0;
  }
lcd.print((int)eTheta);  lcd.print(" p"); lcd.print((int)pTheta);  lcd.print("   ");
}

void testRunKinematik() {
  while (!C()) {
    lcd.clear();
    while (!C() && titikTuju != 1) {
      runKinematik(0, 200, 0, 130);
      displayJarak();
    } titikTuju = 0;
    while (!C() && titikTuju != 1) {
      runKinematik(0, 200, 180, 130);
      displayJarak();
    } titikTuju = 0;
    while (!C() && titikTuju != 1) {
      runKinematik(150, 0, 180, 130);
      displayJarak();
    } titikTuju = 0;
    while (!C() && titikTuju != 1) {
      runKinematik(150, 200, 180, 130);
      displayJarak();
    } titikTuju = 0;
    while (!C() && titikTuju != 1) {
      runKinematik(150, 200, 0, 130);
      displayJarak();
    } titikTuju = 0;
    while (!C() && titikTuju != 1) {
      runKinematik(0, 0, 0, 130);
      displayJarak();
    } titikTuju = 0;
  }
  motor(0, 0, 0); delay(300); lcd.clear();
}

void setting() {
  delay(300); lcd.clear();
  int m = 0, mm = 0, p = 0;
  char ok;

  while (!C()) {
    errorIL = 0;
    errorIR = 0;
    errorIB = 0;
    countDetek = 0;
    countGaDetek = 0;

    lcd.setCursor(0, m); lcd.print(">");

    lcd.setCursor(1, 0); lcd.print("Set PID Bola");
    lcd.setCursor(1, 1); lcd.print("Set Jarak");
    lcd.setCursor(1, 2); lcd.print("Time "); lcd.print(tampilPlayer[p]);
    lcd.setCursor(1, 3); lcd.print("Test : "); lcd.print(tampilTest[mm]);

    lcd.setCursor(13, 2);
    if (ok == 1) {
      if (p == 0) {
        lcd.print(":[");
        lcd.print(timeSerang);
      }
      else if (p == 1) {
        lcd.print(":[");
        lcd.print(timeTengah);
      }
```

```
  else if (p == 2) {
    lcd.print(":[");
    lcd.print(timeBlkang);
  }
  lcd.print("] ");

  if (P() && p == 0) {
    timeSerang++;
    delay(120);
  }
  if (M() && p == 0) {
    timeSerang--;
    delay(120);
  }
  if (P() && p == 1) {
    timeTengah++;
    delay(120);
  }
  if (M() && p == 1) {
    timeTengah--;
    delay(120);
  }
  if (P() && p == 2) {
    timeBlkang++;
    delay(120);
  }
  if (M() && p == 2) {
    timeBlkang--;
    delay(120);
  }

  EEPROM.write(addTimeSerang, timeSerang);
  EEPROM.write(addTimeTengah, timeTengah);
  EEPROM.write(addTimeBlkang, timeBlkang);
}
else {
  lcd.print(": ");
  if (p == 0) {
    lcd.print(timeSerang);
  }
  else if (p == 1) {
    lcd.print(timeTengah);
  }
  else if (p == 2) {
    lcd.print(timeBlkang);
  }
  lcd.print("  ");
}


if (U() && ok == 0) {
  lcd.setCursor(0, m);
  lcd.print(" ");
  m--;
  delay(150);
}
if (D() && ok == 0) {
  lcd.setCursor(0, m);
  lcd.print(" ");
  m++;
  delay(150);
}
if (P() && m == 2 && ok == 0) {
  p--;
  delay(150);
}
if (M() && m == 2 && ok == 0) {
  p++;
  delay(150);
}
if (P() && m == 3) {
```

```
      mm--;
      delay(150);
    }
    if (M() && m == 3) {
      mm++;
      delay(150);
    }
    if (m > 3) {
      m = 0;
    } if (m < 0) {
      m = 3;
    }
    if (p > 2) {
      p = 0;
    } if (p < 0) {
      p = 2;
    }
    if (mm > 9) {
      mm = 0;
    } if (mm < 0) {
      mm = 9;
    }

    if (O() && m == 2) {
      ok++;
      delay(150);
    }
    if (ok > 1) {
      ok = 0;
    }

    if (O() && m == 3 && mm == 0) {
      testPidBola();
    }
    if (O() && m == 3 && mm == 1) {
      testPidBolaXY();
    }
    if (O() && m == 3 && mm == 2) {
      testRunKinematik();
    }
    if (O() && m == 3 && mm == 3) {
      tendang1();
    }
    if (O() && m == 3 && mm == 4) {
      tendang2();
    }
    if (O() && m == 3 && mm == 5) {
      keTengahLapangan();
    }
    if (O() && m == 3 && mm == 6) {
      testKeDepan();
    }
    if (O() && m == 3 && mm == 7) {
      testKeBelakang();
    }
    if (O() && m == 3 && mm == 8) {
      manualKontrol();
    }
    if (O() && m == 3 && mm == 9) {
      dataThesis();
    }

  }
  delay(300); lcd.clear();
}

void dataThesis(){
  unsigned char spd=0;
  int tesMotor=0, enableRun=0;
  delay(300); lcd.clear();
  while (!C()) {
```

```
    kameraHp();
    parseGyro();
    parseBola();
    lcd.setCursor(0, 0);
    lcd.print("L"); lcd.print(rpmL);
    lcd.print(" R"); lcd.print(rpmR);
    lcd.print(" B"); lcd.print(rpmB);  lcd.print("  ");
    lcd.setCursor(0, 1);
    lcd.print("T");lcd.print(Theta);  lcd.print("  ");

    lcd.setCursor(0, 2);
    if(tesMotor==1 && enableRun==1){
      lcd.print("Kiri :");
      motor(spd,0,0);
    }
    else if(tesMotor==2 && enableRun==1){
      lcd.print("Kanan :");
      motor(0,spd,0);
    }
    else if(tesMotor==3 && enableRun==1){
      lcd.print("Belakang:");
      motor(0,0,spd);
    }
    else {
      motor(0,0,0);
      lcd.print("Berhenti ");
    }
    lcd.print(spd); lcd.print("  ");


    if(U()){tesMotor++; delay(300);}
    if(D()){tesMotor--; delay(300);}
    if(P()){spd++; delay(100);}
    if(M()){spd--; delay(100);}
    if(tesMotor>3){tesMotor=0;}
    if(tesMotor<0){tesMotor=3;}

    if(O()){enableRun++; delay(300);}
    if(enableRun>1){enableRun=0;}

  }
  delay(300); lcd.clear();
}

void cekSensor() {
  delay(300); lcd.clear();
  while (!C()) {
    kameraHp();
    parseGyro();
    parseBola();
    lcd.setCursor(0, 0);
    lcd.print("D"); lcd.print(jarDeL1()); lcd.print(" ");  lcd.print(jarDribble());
lcd.print(" "); lcd.print(jarDeR1()); lcd.print("  ");

    lcd.setCursor(0, 1);
    lcd.print("D");  lcd.print(jarDeL());  lcd.print("   ");  lcd.print(jarDeR());
lcd.print(" "); lcd.print(jarde()); lcd.print(" ");

    lcd.setCursor(0, 2);
    lcd.print("L");  lcd.print(jarki());  lcd.print("   B");  lcd.print(jarBeL());
lcd.print("   ");   lcd.print(jarBeR());   lcd.print("   R");   lcd.print(jarka());
lcd.print("  ");

    lcd.setCursor(0, 3);
        lcd.print("L");lcd.print(intL);
        lcd.print(" R");lcd.print(intR);
        lcd.print(" B");lcd.print(intB);  lcd.print("  ");

  }
  delay(300); lcd.clear();
}
```

```
void runFullAuto() {
  delay(300); lcd.clear();
  int count, countKoreksi;
  char pernahStart = 'N';
  enableKickOff = 'N';

  xTarget = 410; yTarget = 450; tTarget = 0;

  while (!C()) {

    if (intruksiPlay == 'R' && titikTuju == 0) {
      if (enableReadThetaNow == 1) {
        thetaNow = Theta;
        enableReadThetaNow = 0;
      }

      runKinematik(xTarget, yTarget, tTarget, 100);
    }

    if (team == 'c') {
      lcd.setCursor(0, 0);
      lcd.print("AUTO C");
    }
    if (team == 'm') {
      lcd.setCursor(0, 0);
      lcd.print("AUTO M");
    }
    lcd.print(" S:"); lcd.print(strategi); lcd.print(" ");

    lcd.setCursor(0, 1);
    lcd.print("X"); lcd.print((int)jarX);
    lcd.print(" Y"); lcd.print((int)jarY);
    lcd.print(" T"); lcd.print((int)Theta); lcd.print("  ");

    if (bolehMain == 'N') {
      lcd.setCursor(0, 2);
      lcd.print("X"); lcd.print(xBola);
      lcd.print(" Y"); lcd.print(yBola); lcd.print("  ");

      lcd.setCursor(0, 3);
      lcd.print("Play Now: ");
    }

    listenCoach();

    if (intruksiPlay == 'S') {
      lcd.setCursor(12, 2); lcd.print("       ");
      motor(0, 0, 0);
      errorIL = 0;
      errorIR = 0;
      errorIB = 0;

      titikTuju = 0;
      enableReadThetaNow = 1;
      runIntruksi = 'N';
      bolehMain = 'N';
      enableMain = 'N';
      countDetek = 0;
      countGaDetek = 0;
    }

    count++;
    if (count > 10) {

      //ini untuk kirim data rosi ke nodeMcu
      Serial2.print(String("*?,")  +  (int)jarX  +  String(",")  +  (int)jarY  +
String(",") + (int)Theta + String(",") + (int)lihatBola + String(",\n"));
      Serial2.print(String("*!,")  +  (int)xTarget  +  String(",")  +  (int)yTarget  +
String(",") + (int)tTarget + String(",\n"));
      count = 0;
    }
```

```
  }

  delay(300); lcd.clear();
}

void listenCoach() {
  if (Serial2.available() > 0) {
    char data = Serial2.read();

    if (data == 'S') {
      intruksiPlay = 'S';
      intruksiR = 'S';
      lcd.setCursor(10, 3);
      lcd.print("Stop      ");
      enableMain = 'N';
      enableInisial = 'N';
    }
    else if (data == 's') {
      intruksiPlay = 's';
      intruksiR = 'S';
      lcd.setCursor(10, 3);
      lcd.print("Start     ");
      enableMain = 'N';
      enableInisial = 'N';
    }
    else if (data == 'R') {
      intruksiPlay = 'R';
      intruksiR = 'S';
      lcd.setCursor(10, 3);
      lcd.print("Reset Pos.");
      enableMain = 'N';
      enableInisial = 'N';
    }
    else if (data == 'r') {
      intruksiPlay = 'r';
      intruksiR = 'S';
      lcd.setCursor(10, 3);
      lcd.print("Krksi Bola");
      enableMain = 'N';
      enableInisial = 'N';
    }

    else if (data == 'N') {
      intruksi = 'N';
      intruksiFull = 'N';
      intruksiR = 'S';
      lcd.setCursor(10, 3);
      lcd.print("DropBall  ");
      enableMain = 'Y';
      enableInisial = 'Y';
    }

    else if (data == 'K') {
      intruksi = 'K';
      intruksiFull = 'K';
      intruksiR = 'S';
      lcd.setCursor(10, 3);
      lcd.print("KickOff C ");
      enableMain = 'Y';
      enableInisial = 'Y';
    }
    else if (data == 'F') {
      intruksi = 'F';
      intruksiFull = 'F';
      intruksiR = 'S';
      lcd.setCursor(10, 3);
      lcd.print("FreeKick C");
      enableMain = 'Y';
      enableInisial = 'Y';
    }
    else if (data == 'G') {
```

```
  intruksi = 'G';
  intruksiFull = 'G';
  intruksiR = 'S';
  lcd.setCursor(10, 3);
  lcd.print("GoalKick C");
  enableMain = 'Y';
  enableInisial = 'Y';
}
else if (data == 'C') {
  intruksi = 'C';
  intruksiFull = 'C';
  intruksiR = 'S';
  lcd.setCursor(10, 3);
  lcd.print("Corner C  ");
  enableMain = 'Y';
  enableInisial = 'Y';
}
else if (data == 'P') {
  intruksi = 'P';
  intruksiFull = 'P';
  intruksiR = 'S';
  lcd.setCursor(10, 3);
  lcd.print("Penalty C ");
  enableMain = 'Y';
  enableInisial = 'Y';
}

else if (data == 'k') {
  intruksi = 'k';
  intruksiFull = 'k';
  intruksiR = 'S';
  lcd.setCursor(10, 3);
  lcd.print("KickOff M ");
  enableMain = 'Y';
  enableInisial = 'Y';
}
else if (data == 'f') {
  intruksi = 'f';
  intruksiFull = 'f';
  intruksiR = 'S';
  lcd.setCursor(10, 3);
  lcd.print("FreeKick M");
  enableMain = 'Y';
  enableInisial = 'Y';
}
else if (data == 'g') {
  intruksi = 'g';
  intruksiFull = 'g';
  intruksiR = 'S';
  lcd.setCursor(10, 3);
  lcd.print("GoalKick M");
  enableMain = 'Y';
  enableInisial = 'Y';
}
else if (data == 'c') {
  intruksi = 'c';
  intruksiFull = 'c';
  intruksiR = 'S';
  lcd.setCursor(10, 3);
  lcd.print("Corner M  ");
  enableMain = 'Y';
  enableInisial = 'Y';
}
else if (data == 'p') {
  intruksi = 'p';
  intruksiFull = 'p';
  intruksiR = 'S';
  lcd.setCursor(10, 3);
  lcd.print("Penalty M ");
  enableMain = 'Y';
  enableInisial = 'Y';
```

```
}
else if (data == 'Q') {
  intruksiR = 'L';
  intruksiFull = 'S';
  lcd.setCursor(10, 3);
  lcd.print("Rotasi L  ");
  enableMain = 'N';
}
else if (data == 'E') {
  intruksiR = 'R';
  intruksiFull = 'S';
  lcd.setCursor(10, 3);
  lcd.print("Rotasi R  ");
  enableMain = 'N';
}
else if (data == 'D') {
  lihatBolaRoso = 'Y';
}
else if (data == 'd') {
  lihatBolaRoso = 'N';
}

// aktifin ini untuk terima data Rosi

else if (data == '?') {
  data = Serial2.read();
  if (data == 'X') {
    int dataX = Serial2.parseInt();
    if (Serial2.read() == 'Y') {
      int dataY = Serial2.parseInt();
      if (Serial2.read() == '#') {
        jarX = (int)dataX;
        jarY = (int)dataY;
        jarL = jarR = jarB = 0;
      }
    }
  }
  else if (data == 'W') {
    int dataW = Serial2.parseInt();
    if (Serial2.read() == '#') {
      Serial3.println(dataW + String("#"));
    }
  }
  enableInisial = 'N';
}
else if (data == '!') {
  data = Serial2.read();
  if (data == 'X') {
    int dataX = Serial2.parseInt();
    if (Serial2.read() == 'Y') {
      int dataY = Serial2.parseInt();
      if (Serial2.read() == '#') {
        xTarget = (int)dataX;
        yTarget = (int)dataY;
      }
    }
  }
  else if (data == 'W') {
    int dataW = Serial2.parseInt();
    if (Serial2.read() == '#') {
      tTarget = dataW;
      titikTuju = 0;
    }
  }
  enableInisial = 'N';
}
else if (data == '@') {
  int dataIn = Serial2.parseInt();
  if (Serial2.read() == '#') {
    if (dataIn < 3) {
      strategi = dataIn;
```

```
        }
      }
      enableInisial = 'N';
    }
  }
}
```

Desain dan Implementasi Sistem Gerak Robot Sepak Bola Beroda Menggunakan Forward dan Inverse Kinematics

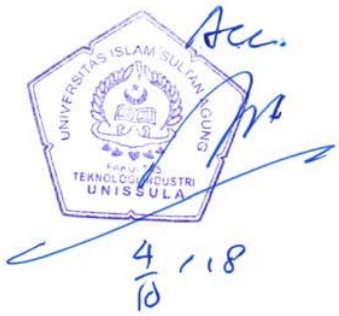
ORIGINALITY REPORT

2%	2%	0%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	media.neliti.com Internet Source	2%

Exclude quotes	Off	Exclude matches	< 2%
Exclude bibliography	On		



4/10, 18