APPENDIX

CODING SENSOR

```
void sensor(){
  //======================== ultrasonik ============================
  digitalWrite(trig, LOW);
  delayMicroseconds(8);
  digitalWrite(trig,HIGH);
  delayMicroseconds(8);
  digitalWrite(trig,LOW);
  delayMicroseconds(8);

  durasi  = pulseIn(echo, HIGH); //menerima gelombang ultrasonik
  jarak   = (durasi / 2) / 29.1;  // ubah jarak durasi jadi jarak
 // jarak   = 20;
  //========================== TDS ================================
  gravityTds.update();
  nilaiPPM = gravityTds.getTdsValue()  ; //dapatkan nilai PPM
}
```

CODING FUZZUFIKASI

```
 void FuzzyfikasiAir()
 {
  // ================================================ rendah(penuh)
  if (jarak <= 10)
    {
      air[0] = 1;
    }
   else if (jarak > 10 && jarak <= 15)
    {
      air[0] = (15 - jarak)/(15 - 10);
    }
   else if (jarak >= 15)
    {
      air[0] = 0;
    }

  //================================================ cukup
  if (jarak <= 10)
   {
     air[1] = 0;
   }
  else if(jarak > 10 && jarak <= 15)
   {
     air[1] = (jarak - 10) / (15 - 10);
   }
  else if(jarak > 15 && jarak <= 20)
   {
     air[1] = (20 - jarak) / (20 - 15);
   }
  else
   {
     air[1] = 0;
   }

```

```
370.    // ================================================ tinggi(rendah)
371.      if(jarak <= 15)
372.      {
373.        air[2] = 0;
374.      }
375.      else if(jarak > 15 && jarak <= 20)
376.      {
377.        air[2] = (jarak - 15)/(20 - 15);
378.      }
379.      else if (jarak >=20)
380.      {
381.        air[2] = 1;
382.      }
383.
384.    Serial.print(" Penuh  = ");
385.    Serial.print(air[0]);
386.    Serial.println("");
387.    Serial.print(" Cukup  = ");
388.    Serial.print(air[1]);
389.    Serial.println("");
390.    Serial.print(" Rendah = ");
391.    Serial.print(air[2]);
392.    Serial.println("");
393.   }
394.
395.  //==============================                                    TDS
  ====================================
396.   void FuzzyfikasiTDS()
397.   {
398.    // ============================================== sangat kurang
399.    if (nilaiPPM <= 500)
400.      {
401.        nutrisi[0] = 1;
402.      }
403.      else if(nilaiPPM > 500 && nilaiPPM <=750)
404.      {
405.        nutrisi[0]=(750 - nilaiPPM)/(750-500);
406.      }
407.      else if(nilaiPPM >= 750)
408.      {
409.        nutrisi[0]=0;
410.      }
411.
412.    // ============================================== kurang
413.    if (nilaiPPM <= 500)
414.    {
415.      nutrisi[1]= 0;
416.    }
417.    else if(nilaiPPM > 500 && nilaiPPM <= 750)
418.    {
419.     nutrisi[1] = (nilaiPPM - 500)/(750 - 500);
420.    }
421.    else if(nilaiPPM > 750 && nilaiPPM <= 1050)
422.    {
423.      nutrisi[1] = (1050 - nilaiPPM)/(1050 - 750);
424.    }
425.    else
426.    {
427.      nutrisi[1]=0;
```

```
    }

    // ================================================ normal
    if (nilaiPPM <= 750)
    {
      nutrisi[2]=0;
    }
    else if(nilaiPPM > 750 && nilaiPPM <= 1050)
    {
      nutrisi[2] = (nilaiPPM - 750)/(1050 - 750);
    }
    else if(nilaiPPM > 1050 && nilaiPPM <= 1200)
    {
      nutrisi[2] = 1;
    }
    else if(nilaiPPM > 1200 && nilaiPPM <= 1600)
    {
      nutrisi[2] = (1600 - nilaiPPM)/(1600 - 1200) ;
    }

    // ================================================ tinggi
    if(nilaiPPM <= 1200)
    {
      nutrisi[3]= 0;
    }
    else if(nilaiPPM > 1200 && nilaiPPM <= 1600)
    {
      nutrisi[3]=(nilaiPPM - 1200)/(1600 - 1200);
    }
    else if(nilaiPPM >= 1600)
    {
      nutrisi[3]=1;
    }
    Serial.print(" sangat kurang = ");
    Serial.print(nutrisi[0]);
    Serial.println("");
    Serial.print(" kurang        = ");
    Serial.print(nutrisi[1]);
    Serial.println("");
    Serial.print(" normal        = ");
    Serial.print(nutrisi[2]);
    Serial.println("");
    Serial.print(" tinggi        = ");
    Serial.print(nutrisi[3]);
    Serial.println("");
  }
```

CODING RULE

```
  void Rulenutrisi() //========== POMPA NUTRISI =========
  {
    int i, j;
    int no = 1;
    for(i = 0; i <= 3; i = i + 1)
    {
      for(j = 0; j <= 2; j = j + 1)
      {
      rul1 = min(nutrisi[i], air[j]);
      rule [i][j] = rul1;
```

```
      Serial.print("Rules ke ...");
      Serial.print(no++);
      Serial.print(" ->  ");
      Serial.println(rule[i][j]);
      }
    }
  //------------------------------------------------------------
    rule1   = rule[0][0]; // (penuh, sangat kurang = lama)
    rule2   = rule[0][1]; // (cukup, sangat kurang = lama)
    rule3   = rule[0][2]; // (rendah, sangat kurang = lama)
  //------------------------------------------------------------
    rule4   = rule[1][0]; // (penuh, kurang = sedang)
    rule5   = rule[1][1]; // (cukup, kurang = sedang)
    rule6   = rule[1][2]; // (rendah, kurang = sedang)
  //------------------------------------------------------------
    rule7   = rule[2][0]; // (penuh, normal = cepat)
    rule8   = rule[2][1]; // (cukup, normal = cepat)
    rule9   = rule[2][2]; // (rendah, normal = diam)
  //------------------------------------------------------------
    rule10  = rule[3][0]; // (penuh, tinggi = diam)
    rule11  = rule[3][1]; // (cukup, tinggi = diam)
    rule12  = rule[3][2]; // (rendah, tinggi = diam)
  //------------------------------------------------------------
   }

  //======================RuleAir=============================
  void Ruleair()
  {
    int a, b;
    int nom = 1; //mulai dari 1
    for(a = 0; a <= 3; a = a+1)
    {
      for(b = 0; b <= 2; b = b+1)
      {
        rul2 = min(nutrisi[a], air[b]);
        ruleb [a][b] = rul2;
        Serial.print("Rules ke ...");
        Serial.print(nom++);
        Serial.print(" ->  ");
        Serial.println(ruleb[a][b]);
      }
    }
  //------------------------------------------------------------
    ruleb01   = ruleb[0][0]; // (penuh, sangat kurang = diam)
    ruleb02   = ruleb[0][1]; // (cukup, sangat kurang = cepat)
    ruleb03   = ruleb[0][2]; // (rendah, sangat kurang = lama)
  //------------------------------------------------------------
    ruleb04   = ruleb[1][0]; // (penuh, kurang = diam)
    ruleb05   = ruleb[1][1]; // (cukup, kurang = cepat)
    ruleb06   = ruleb[1][2]; // (redah, kurang = lama)
  //------------------------------------------------------------
    ruleb07   = ruleb[2][0]; // (penuh, normal = diam)
    ruleb08   = ruleb[2][1]; // (cukup, normal = cepat)
    ruleb09   = ruleb[2][2]; // (rendah, normal = lama)
  //------------------------------------------------------------
    ruleb010  = ruleb[3][0]; // (penuh, tinggi = cepat)
    ruleb011  = ruleb[3][1]; // (cukup, tinggi = cepat)
    ruleb012  = ruleb[3][2]; // (rendah, tinggi = lama)
  //------------------------------------------------------------
```

```
543.  }
```

CODING DEFUZZIFIKASI

```
   void Defusifikasi_nutrisi ()
544.   {
545.    float Diam = 0;
546.    float Cepat = 5;
547.    float Sedang = 10;
548.    float Lama  = 15;
549.    //Rule();
550.  output1                                                                    =
  (rule1*Lama)+(rule2*Lama)+(rule3*Lama)+(rule4*Sedang)+(rule5*Sedang)+(rul
  e6*Sedang)+(rule7*Diam)+
  (rule8*Diam)+(rule9*Diam)+(rule10*Diam)+(rule11*Diam)+(rule12*Diam);

551.  output2=rule1+rule2+rule3+rule4+rule5+rule6+rule7+rule8+
552.  rule9+rule10+rule11+rule12;
553.
554.    hasil_defu = output1/output2;
555.    delayPompaNutrisi = (hasil_defu * 1000);
556.    Serial.print(output1);
557.    Serial.println("");
558.    Serial.print(output2);
559.    Serial.println("");
560.
561.    if (hasil_defu >= 0.00 && hasil_defu <5.00 ){
562.      digitalWrite(pompa01, LOW);
563.      //delay(delayPompaNutrisi);
564.      digitalWrite(pompa01, LOW);
565.      Serial.print("Result  : ");
566.      Serial.print(hasil_defu);
567.      Serial.println(" -> pompa mati");
568.    }
569.    else if (hasil_defu >= 5.00 && hasil_defu <10.00){
570.      digitalWrite(pompa01,HIGH);
571.      delay(delayPompaNutrisi);
572.      digitalWrite(pompa01, LOW);
573.      Serial.print("Result  : ");
574.      Serial.print(hasil_defu);
575.      Serial.print(" -> pompa cepat");
576.    }
577.    else if (hasil_defu >= 10.00 && hasil_defu < 15.00){
578.      digitalWrite(pompa01,HIGH);
579.      delay(delayPompaNutrisi);
580.      digitalWrite(pompa01, LOW);
581.      Serial.print("Result  : ");
582.      Serial.print(hasil_defu);
583.      Serial.print(" -> pompa sedang");
584.    }
585.    else if (hasil_defu >= 15.00 ){
586.      digitalWrite(pompa01,HIGH);
587.      delay(delayPompaNutrisi);
588.      digitalWrite(pompa01, LOW);
589.      Serial.print("Result  : ");
590.      Serial.print(hasil_defu);
591.      Serial.print(" -> pompa lama");
592.    }
593.
```

```
}

void Defuzifikasi_air()
{
  float Diam1 = 0;
  float Cepat1 = 10;
  float Lama1  = 20;
  //Rule();
output1_air                                                                    =
(ruleb01*Diam1)+(ruleb02*Cepat1)+(ruleb03*Lama1)+(ruleb04*Diam1)+(ruleb05
*Cepat1)+(ruleb06*Lama1)+(ruleb07*Diam1)+
(ruleb08*Cepat1)+(ruleb09*Lama1)+(ruleb010*Cepat1)+(ruleb011*Cepat1)+(rul
eb012*Lama1);
output2_air=ruleb01+ruleb02+ruleb03+ruleb04+ruleb05+ruleb06+ruleb07+r
uleb08+ruleb09+ruleb010+ruleb011+ruleb012;
  hasil_defu_air = output1_air/output2_air;
  delayPompaAir = (hasil_defu_air * 1000);
  Serial.print(output1_air);
  Serial.println("");
  Serial.print(output2_air);
  Serial.println("");

  if (hasil_defu_air >= 0.00 && hasil_defu_air <10 ){
    digitalWrite(pompa02, LOW);
    //delay(delayPompaAir);
    digitalWrite(pompa02, LOW);
    Serial.print("Result  : ");
    Serial.print(hasil_defu_air);
    Serial.println(" -> pompa mati");
  }
  else if (hasil_defu_air >= 10.00 && hasil_defu_air <20.00){
    digitalWrite(pompa02,HIGH);
    delay(delayPompaAir);
    digitalWrite(pompa02, LOW);
    Serial.print("Result  : ");
    Serial.print(hasil_defu_air);
    Serial.print(" -> pompa cepat");
  }
  else if (hasil_defu_air >= 20.00 ){
    digitalWrite(pompa02,HIGH);
    delay(delayPompaAir);
    digitalWrite(pompa02, LOW);
    Serial.print("Result  : ");
    Serial.print(hasil_defu_air);
    Serial.print(" -> pompa lama");
  }

}
```

Link video alat

https://bit.ly/fagg

PLAGIARISM CHECK.ORG

1.21% PLAGIARISM APPROXIMATELY

Report #13962219

INTRODUCTION Background Hydroponics is a farming technique using water media that contains nutrients that plants need to grow and develop, therefore we must be careful in maintaining the nutrient content that is usually stored in a reservoir. This is the main problem if we cannot maintain optimal nutritional conditions in the reservoir, plants that lack nutrients will affect their growth, otherwise, plants with excess nutrients will quickly rot and turn yellow. Therefore, we need a tool that can control the levels of hydroponic nutrients in the reservoir. We need a tool that can control the hydroponic nutrient reservoir based on a microcontroller in which fuzzy logic is embedded. Fuzzy logic itself consists of many rules that can regulate the output of the output pump which consists of a nutrition pump and a water pump, the output is in the form of a delay value which will be used to turn the relay on and off.
As we know hydroponic farming requires patience and accuracy in managing the nutrient reservoir, with this tool we no