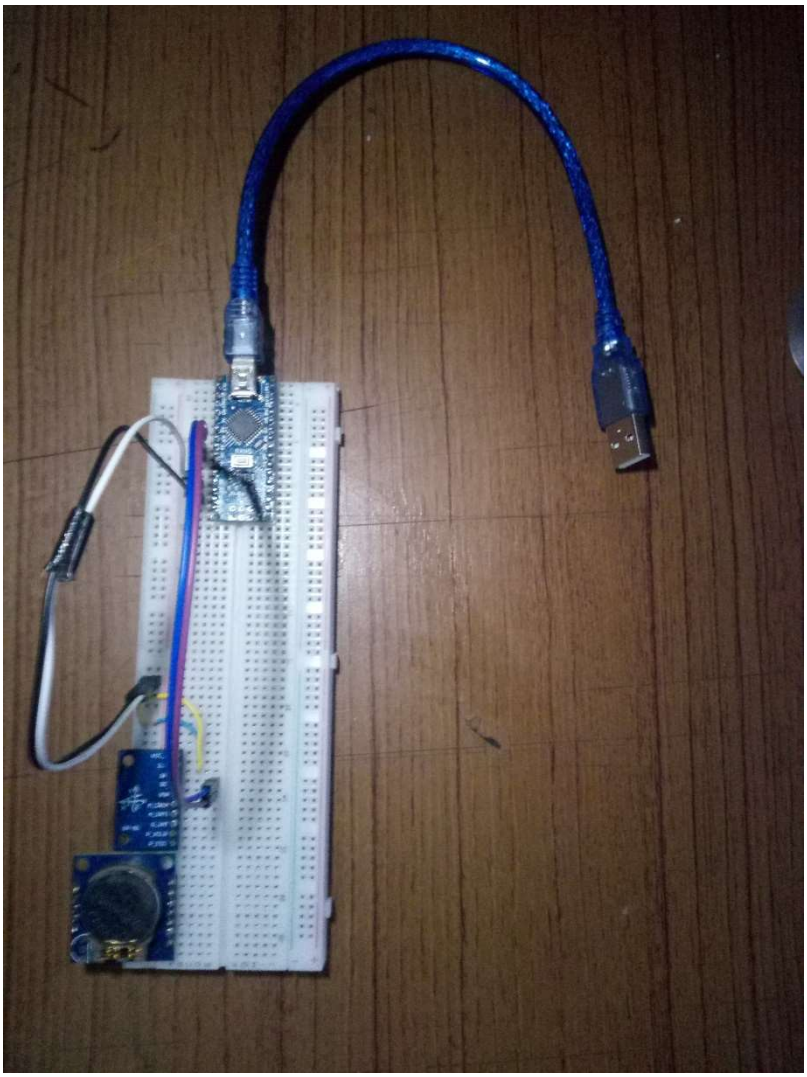
3.3 Realisasi

Pada tahap ini mencakup pembuatan sistem dan alat serta mengintegrasikan komponen. Berikut bagian-bagian yang direalisasikan.

3.3.1 Realisasi perangkat keras



Gambar 3.5 Perangkat keras Sensor 10DOOF, arduino nano dan RTC DS1307

3.3.2 Realisasi PCB

3.3.3 Realisasi Perakitan

3.3.4 Realisasi Pengkabelan

3.3.5 Relisasi Perangkat Lunak

Gambar 3.6 Perangkat lunak untuk menampilkan monitor setiap sensor

3.3.6 Realisasi Program

Dibawah merupakan kode program pada arduino untuk mengambil data dari setiap sensor yang ada di sensor 10DOF untuk dikirimkan melalui komunikasi serial.

```
#include <Wire.h>

//---------Accelerometer BMA180--------------
#define BMA180 0x40 //alamat register accelerometer BMA180
#define RESET 0x10
#define PWR 0x0D
#define BW 0X20
#define RANGE 0X35
#define DATA 0x02

int offx_Accelero = 31; //offset data x
int offy_Accelero = 47; //offset data y
int offz_Accelero = -23; //offset data z

//---------Magnetometer HMC5883L------------
#define Magnetometer 0x1E //alamat register magnetometer HMC5883
#define MagMode 0x02
#define Magnetometer_mX0 0x03
#define Magnetometer_mX1 0x04
#define Magnetometer_mZ0 0x05
#define Magnetometer_mZ1 0x06
#define Magnetometer_mY0 0x07
#define Magnetometer_mY1 0x08
 int mX0, mX1, mX_out;
 int mY0, mY1, mY_out;
 int mZ0, mZ1, mZ_out;
 float heading, headingDegrees, headingFiltered, declination;
 float Xm,Ym,Zm;

//-------------Gyroscope ITG3200------------
#define GYRO 0x68 //  when AD0 is connected to GND ,gyro address is 0x68.
//#define GYRO 0x69   when AD0 is connected to VCC ,gyro address is 0x69
#define G_SMPLRT_DIV 0x15
#define G_DLPF_FS 0x16
#define G_INT_CFG 0x17
#define G_PWR_MGM 0x3E
#define G_TO_READ 8 // 2 bytes for each axis x, y, z
// offsets are chip specific.
int g_offx = 120;
int g_offy = 18;
int g_offz = 87;
int gX_out, gY_out, gZ_out, gTemp_out;

//------------Barometric Pressure Sensor BMP085------------
#define BMP085_ADDRESS 0x77  // I2C address of BMP085
const unsigned char OSS = 0;  // Oversampling Setting
// Calibration values
int ac1;
int ac2;
int ac3;
unsigned int ac4;
unsigned int ac5;
unsigned int ac6;
int b1;
int b2;
int mb;
int mc;
int md;
```

```
// b5 is calculated in bmp085GetTemperature(...), this variable is also used in
bmp085GetPressure(...)
// so ...Temperature(...) must be called before ...Pressure(...).
long b5;
short temperature;
float pressure;

//-----Data Output-----
String data_Accel, data_Magnetometer, data_Gyro, data_Barometric;

//-------------Setup----------------
void setup()
{
  Serial.begin(9600);
  Wire.begin();
  //Serial.println("Menginisialisasi Sensor Accelerometer.....");
  AccelerometerInit();
 // Serial.println("Menginisialisasi Sensor Magnetometer.....");
  MagnetometerInit();
  //Serial.println("Menginisialisasi Sensor Gyroscope.....");
  GyroscopeInit();
  //Serial.println("Menginisialisasi Sensor Barometric Pressure.....");
  BarometricInit();
  //Serial.println("Sensor telah terinisialisasi");
  }

//------------Accelerometer Inisialisasi-------------
void AccelerometerInit()
{
 byte temp[1];
 byte temp1;
  //Inisialisasi
  tulisKe(BMA180,RESET,0xB6);
  //wake up mode 0xB6
  tulisKe(BMA180,PWR,0x10);
  // low pass filter,
  bacaDari(BMA180, BW,1,temp);
  temp1=temp[0]&0x0F;
  tulisKe(BMA180, BW, temp1);
  // range +/- 2g
  bacaDari(BMA180, RANGE, 1 ,temp);
  temp1=(temp[0]&0xF1) | 0x04;
  tulisKe(BMA180,RANGE,temp1);
}

//-------------Magnetometer Inisialisasi--------
void MagnetometerInit()
{
  //pilih mode register ke pengukuran terus-menerus
  tulisKe(0x3C, 0x00, 0x70);
  tulisKe(0x3C, 0x01, 0xA0); //default 15Hz pengukuran
  tulisKe(Magnetometer, MagMode, 0x00);
  }

//--------------Gyroscope Inisialisasi-------------
void GyroscopeInit()
{
 //ITG 3200
 //power management set to:
 //clock select = internal oscillator
 //no reset, no sleep mode
 //no standby mode
 //sample rate to = 125Hz
 //parameter to +/- 2000 degrees/sec
 //low pass filter = 5Hz
 //no interrupt
 //***************************************
```

```
 tulisKe(GYRO, G_PWR_MGM, 0x00);
 tulisKe(GYRO, G_SMPLRT_DIV, 0x07); // EB, 50, 80, 7F, DE, 23, 20, FF
 tulisKe(GYRO, G_DLPF_FS, 0x1E); // +/- 2000 dgrs/sec, 1KHz, 1E, 19
 tulisKe(GYRO, G_INT_CFG, 0x00);
}

//------------Barometric Inisialisasi----------------
void BarometricInit()
{
ac1 = bmp085ReadInt(0xAA);
ac2 = bmp085ReadInt(0xAC);
ac3 = bmp085ReadInt(0xAE);
ac4 = bmp085ReadInt(0xB0);
ac5 = bmp085ReadInt(0xB2);
ac6 = bmp085ReadInt(0xB4);
b1 = bmp085ReadInt(0xB6);
b2 = bmp085ReadInt(0xB8);
mb = bmp085ReadInt(0xBA);
mc = bmp085ReadInt(0xBC);
md = bmp085ReadInt(0xBE);
}

//----------------Accelerometer Baca Data-----------------
String AccelerometerRead()
{
 // baca 3 axis data, 2 byte untuk setiap data
 // cetak data ke terminal
 int n=6;
 byte hasil_Acc[5];
 bacaDari(BMA180, DATA, n , hasil_Acc);

 int x= (( hasil_Acc[0] | hasil_Acc[1]<<8)>>2)+offx_Accelero ;
 float x1=x/4096.0; //faktor pembagi sensitivitas 4096 untuk sensitivitas +/- 2g
 //Serial.print("Accelero: ");
 //Serial.print("x=");
 //Serial.print(x1*9.8); //dikalikan dengan percepatan gravitasi standar 9.8 m/s2
 //
 int y= (( hasil_Acc[2] | hasil_Acc[3]<<8 )>>2)+offy_Accelero;
 float y1=y/4096.0;
 //Serial.print(",y=");
 //Serial.print(y1*9.8);
 //
 int z= (( hasil_Acc[4] | hasil_Acc[5]<<8 )>>2)+offz_Accelero;
 float z1=z/4096.0;
 //Serial.print(",z=");
 //Serial.print(z1*9.8);

 data_Accel = String(x1*9.8)+";"+String(y1*9.8)+";"+String(z1*9.8)+";";
 return data_Accel;
 //delay(50);
}

//--------------Magnetometer Baca Data------------
String MagnetometerRead()
{
  //---- X-Axis
  Wire.beginTransmission(Magnetometer); // transmit to device
  Wire.write(Magnetometer_mX1);
  Wire.endTransmission();
  Wire.requestFrom(Magnetometer,1);
  if(Wire.available()<=1)
  {
    mX0 = Wire.read();
  }
  Wire.beginTransmission(Magnetometer); // transmit to device
  Wire.write(Magnetometer_mX0);
  Wire.endTransmission();
```

```
  Wire.requestFrom(Magnetometer,1);
  if(Wire.available()<=1)
  {
    mX1 = Wire.read();
  }

  //---- Y-Axis
  Wire.beginTransmission(Magnetometer); // transmit to device
  Wire.write(Magnetometer_mY1);
  Wire.endTransmission();
  Wire.requestFrom(Magnetometer,1);
  if(Wire.available()<=1)
  {
    mY0 = Wire.read();
  }
  Wire.beginTransmission(Magnetometer); // transmit to device
  Wire.write(Magnetometer_mY0);
  Wire.endTransmission();
  Wire.requestFrom(Magnetometer,1);
  if(Wire.available()<=1)
  {
    mY1 = Wire.read();
  }

  //---- Z-Axis
  Wire.beginTransmission(Magnetometer); // transmit to device
  Wire.write(Magnetometer_mZ1);
  Wire.endTransmission();
  Wire.requestFrom(Magnetometer,1);
  if(Wire.available()<=1)
  {
    mZ0 = Wire.read();
  }
  Wire.beginTransmission(Magnetometer); // transmit to device
  Wire.write(Magnetometer_mZ0);
  Wire.endTransmission();
  Wire.requestFrom(Magnetometer,1);
  if(Wire.available()<=1)
  {
    mZ1 = Wire.read();
  }
   //---- X-Axis
  mX1=mX1<<8;
  mX_out =mX0+mX1; // Raw data
  // From the datasheet: 0.92 mG/digit
  Xm = mX_out*0.00092; // Gauss unit
  //* Earth magnetic field ranges from 0.25 to 0.65 Gauss, so these are the values that
we need to get approximately.
  //Serial.print("\tMagnetometer : ");
  //Serial.print("x=");
  //Serial.print(mX_out);

  //---- Y-Axis
  mY1=mY1<<8;
  mY_out =mY0+mY1;
  Ym = mY_out*0.00092;
  //Serial.print(",y=");
  //Serial.print(mY_out);

  //---- Z-Axis
  mZ1=mZ1<<8;
  mZ_out =mZ0+mZ1;
  Zm = mZ_out*0.00092;
  //Serial.print(",z=");
  //Serial.print(mZ_out);
  //delay(50);
  data_Magnetometer = String(mX_out)+";"+String(mY_out)+";"+String(mZ_out)+";";
```

```
  return data_Magnetometer;
  }

//----------------Gyroscope Baca Data-----------------
String GyroscopeRead()
{
  /************************************
 Gyro ITG-3200 I2C
 registers:
 temp MSB = 1B, temp LSB = 1C
 x axis MSB = 1D, x axis LSB = 1E
 y axis MSB = 1F, y axis LSB = 20
 z axis MSB = 21, z axis LSB = 22
 ************************************/
 int regAddress = 0x1B;
 byte buff[G_TO_READ];
 bacaDari(GYRO, regAddress, G_TO_READ, buff); //read the gyro data from the ITG3200
 int gX = ((buff[2] << 8) | buff[3]) + g_offx;
 gX_out = gX / 14.375;
 //Serial.print("\tGyro : ");
 //Serial.print("x=");
// Serial.print(gX_out);

 int gY = ((buff[4] << 8) | buff[5]) + g_offy;
 gY_out = gY / 14.375;
 //Serial.print(",y=");
 //Serial.print(gY_out);

 int gZ = ((buff[6] << 8) | buff[7]) + g_offz;
 gZ_out = gZ / 14.375;
 //Serial.print(",z=");
 //Serial.print(gZ_out);

 int gTemp = (buff[0] << 8) | buff[1]; // temperature
 gTemp_out = 35+ ((double) (gTemp + 13200)) / 280;
  //Serial.print(",Temp=");
  //Serial.print(gTemp_out);
  //delay(50);
  data_Gyro = String(gX_out)+";"+String(gY_out)+";"+String(gZ_out)+";";
  return data_Gyro;
  }

//-------------Sensor Tekanan Baca Data-----------
String BarometricRead()
{
temperature = bmp085GetTemperature(bmp085ReadUT());
pressure = bmp085GetPressure(bmp085ReadUP());
//Serial.print("\tBarometric: ");
//Serial.print("Temp= ");
//Serial.print(temperature*0.1);
//Serial.print("C");
//Serial.print(",Tekanan= ");
//Serial.print(pressure);
//Serial.print("Pa");
//delay(50);
data_Barometric = String(temperature*0.1)+";"+String(pressure/10000)+";";
return data_Barometric;
}
short bmp085GetTemperature(unsigned int ut)
{
long x1, x2;
x1 = (((long)ut - (long)ac6)*(long)ac5) >> 15;
x2 = ((long)mc << 11)/(x1 + md);
b5 = x1 + x2;
return ((b5 + 8)>>4);
}
// Calculate pressure given up
```

```
// calibration values must be known
// b5 is also required so bmp085GetTemperature(...) must be called first.
// Value returned will be pressure in units of Pa.
long bmp085GetPressure(unsigned long up)
{
long x1, x2, x3, b3, b6, p;
unsigned long b4, b7;
b6 = b5 - 4000;
// Calculate B3
x1 = (b2 * (b6 * b6)>>12)>>11;
x2 = (ac2 * b6)>>11;
x3 = x1 + x2;
b3 = (((((long)ac1)*4 + x3)<<OSS) + 2)>>2;
// Calculate B4
x1 = (ac3 * b6)>>13;
x2 = (b1 * ((b6 * b6)>>12))>>16;
x3 = ((x1 + x2) + 2)>>2;
b4 = (ac4 * (unsigned long)(x3 + 32768))>>15;
b7 = ((unsigned long)(up - b3) * (50000>>OSS));
if (b7 < 0x80000000)
  p = (b7<<1)/b4;
else
  p = (b7/b4)<<1;
x1 = (p>>8) * (p>>8);
x1 = (x1 * 3038)>>16;
x2 = (-7357 * p)>>16;
p += (x1 + x2 + 3791)>>4;
return p;
}
// Read 1 byte from the BMP085 at 'address'
char bmp085Read(unsigned char address)
{
unsigned char data;
Wire.beginTransmission(BMP085_ADDRESS);
Wire.write(address);
Wire.endTransmission();
Wire.requestFrom(BMP085_ADDRESS, 1);
while(!Wire.available());
return Wire.read();
}
// Read 2 bytes from the BMP085
// First byte will be from 'address'
// Second byte will be from 'address'+1
int bmp085ReadInt(unsigned char address)
{
unsigned char msb, lsb;
Wire.beginTransmission(BMP085_ADDRESS);
Wire.write(address);
Wire.endTransmission();
Wire.requestFrom(BMP085_ADDRESS, 2);
while(Wire.available()<2);
msb = Wire.read();
lsb = Wire.read();
return (int) msb<<8 | lsb;
}
// Read the uncompensated temperature value
unsigned int bmp085ReadUT()
{
unsigned int ut;
// Write 0x2E into Register 0xF4
// This requests a temperature reading
Wire.beginTransmission(BMP085_ADDRESS);
Wire.write(0xF4);
Wire.write(0x2E);
Wire.endTransmission();
// Wait at least 4.5ms
delay(5);
```

```
// Read two bytes from registers 0xF6 and 0xF7
ut = bmp085ReadInt(0xF6);
return ut;
}
// Read the uncompensated pressure value
unsigned long bmp085ReadUP()
{
unsigned char msb, lsb, xlsb;
unsigned long up = 0;
// Write 0x34+(OSS<<6) into register 0xF4
// Request a pressure reading w/ oversampling setting
Wire.beginTransmission(BMP085_ADDRESS);
Wire.write(0xF4);
Wire.write(0x34 + (OSS<<6));
Wire.endTransmission();
// Wait for conversion, delay time dependent on OSS
delay(2 + (3<<OSS));
// Read register 0xF6 (MSB), 0xF7 (LSB), and 0xF8 (XLSB)
Wire.beginTransmission(BMP085_ADDRESS);
Wire.write(0xF6);
Wire.endTransmission();
Wire.requestFrom(BMP085_ADDRESS, 3);
// Wait for data to become available
while(Wire.available() < 3);
msb = Wire.read();
lsb = Wire.read();
xlsb = Wire.read();
up = (((unsigned long) msb << 16) | ((unsigned long) lsb << 8) | (unsigned long) xlsb)
>> (8-OSS);
return up;
}

//----------------Program Loop-----------------
void loop()
{
 AccelerometerRead();
 MagnetometerRead();
 GyroscopeRead();
 BarometricRead();
 String data_Sensor = data_Accel+data_Magnetometer+data_Gyro+data_Barometric;
 Serial.print(data_Sensor);
 Serial.println();
 delay(100);
}

//---------------- Functions--------------------
//Tulis nilai ke alamat register di Sensor
void tulisKe(int SENSOR, byte alamat, byte nilai)
{
  Wire.beginTransmission(SENSOR);   //mulai transmsi ke sensor
  Wire.write(alamat);               //tulis alamat register sensor
  Wire.write(nilai);                   //kirim nilai ke write
  Wire.endTransmission();           //akhiri transmisi dari sensor
}
//membaca num bytes mulai dari alamat register ke buffer array
 void bacaDari(int SENSOR, byte alamat , int num ,byte buff[])
 {
 Wire.beginTransmission(SENSOR); //mulai transmsi ke sensor
 Wire.write(alamat);            //kirim alamat register
 Wire.endTransmission();        //akhiri transmisi dari sensor

 Wire.beginTransmission(SENSOR); //mulai transmsi ke sensor
 Wire.requestFrom(SENSOR,num);  //meminta 6 bit dari sensor

 int i=0;
 while(Wire.available())        //Sensor may abnormal
 {
```

```
  buff[i] =Wire.read();        //receive a byte
  i++;
  }
  Wire.endTransmission();         //end transmission
  }
```