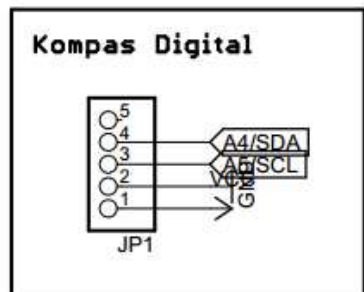
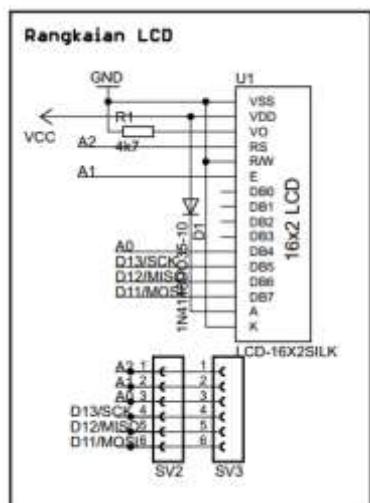
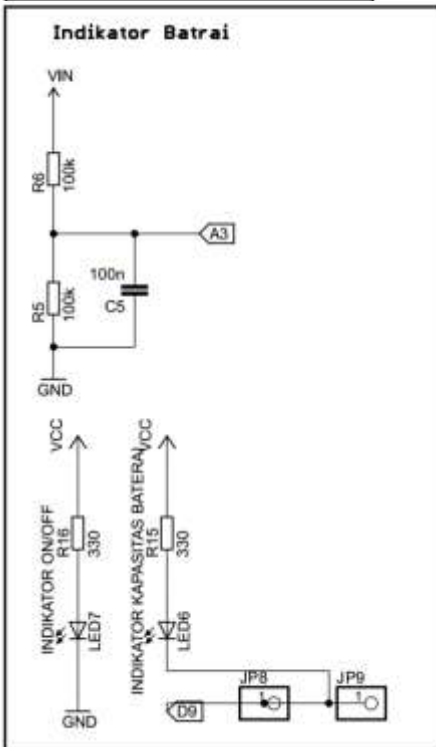
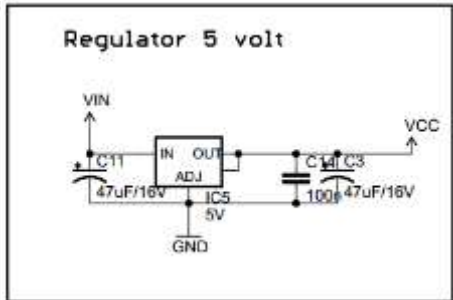
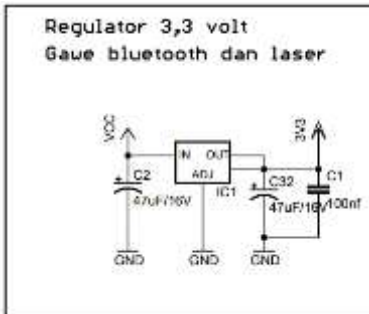
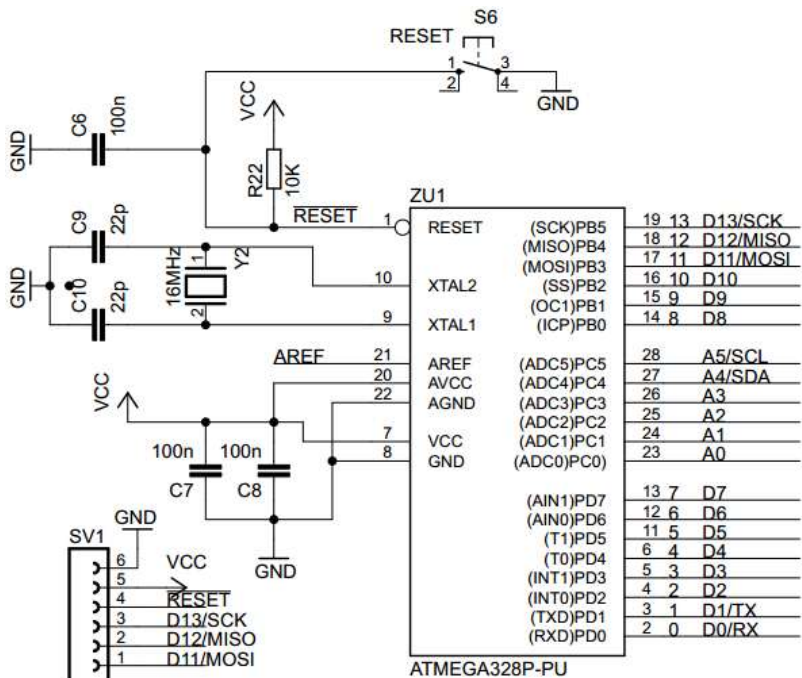


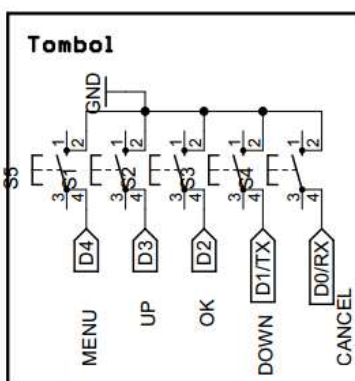
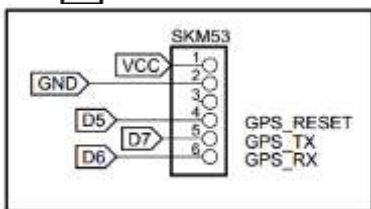
LAMPIRAN 1



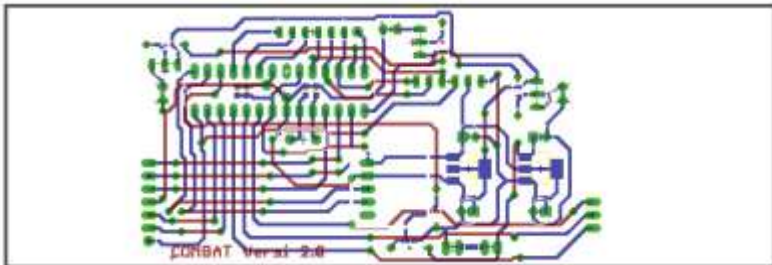
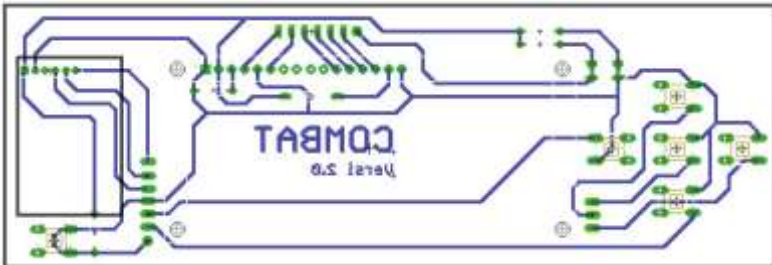
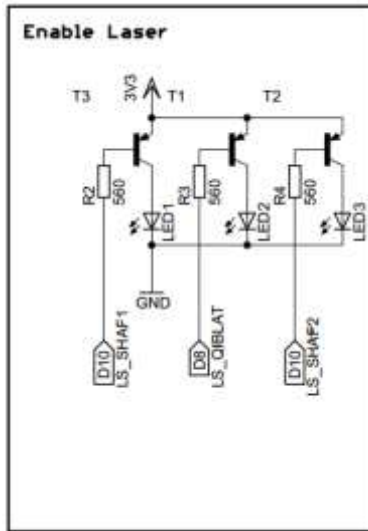
LAMPIRAN 2



ATMEGA328P-PU



LAMPIRAN 3



LAMPIRAN 4

Program Uji Coba GPS

```
#include <TinyGPS.h>

#include <SoftwareSerial.h>

#include <LiquidCrystal.h>

LiquidCrystal lcd(A2,A1,A0,13,12,11);

unsigned long fix_age;

SoftwareSerial GPS(7,6);

TinyGPS gps;

void gpsdump(TinyGPS &gps);

bool feedgps();

void getGPS();

long lat, lon;

float LAT, LON;

void setup()

{

  GPS.begin(9600);

  lcd.begin(16, 2);

}
```

```
void loop()
{
    long lat, lon;

    unsigned long fix_age, time, date, speed, course;

    unsigned long chars;

    unsigned short sentences, failed_checksum;

    // retrieves +/- lat/long in 100000ths of a degree
    gps.get_position(&lat, &lon, &fix_age);

    getGPS();

    lcd.setCursor(0,0);

    lcd.print("Lt: ");

    lcd.print(LAT/100000,7);

    lcd.setCursor(0,1);

    lcd.print("Ln: ");

    lcd.print(LON/100000,7);

    delay(1000);
}
```

```
void getGPS()
{
    bool newdata = false;

    unsigned long start = millis();

    // Every 1 seconds we print an update
    while (millis() - start < 1000)
```

```

{
    if (feedgps ()){newdata = true; }
}

if (newdata)
{gpsdump(gps);}
}

bool feedgps()
{
    while (GPS.available())
    {
        if (gps.encode(GPS.read()))
            return true;
    }return 0;
}

void gpsdump(TinyGPS &gps)
{
    //byte month, day, hour, minute, second, hundredths;
    gps.get_position(&lat, &lon);
    LAT = lat;
    LON = lon;
    { feedgps(); }
}

```

LAMPIRAN 5

Program Uji Coba Kompas

```
#include <Wire.h>

#include <HMC5883L.h>

#include <LiquidCrystal.h>

LiquidCrystal lcd(A2, A1, A0, 13, 12, 11);

float LX = -7.36791666667, BX = 112.701888889 , LK = 21.422412, BK = 39.826176, K,
K1, KK ;

HMC5883L compass;

int error = 0;

float kmps;

void setup()
{
  lcd.begin(16, 2);

  Serial.begin(9600);

  Serial.println("Starting the I2C interface.");

  Wire.begin(); // Start the I2C interface.

  Serial.println("Constructing new HMC5883L");

  compass = HMC5883L(); // Construct a new HMC5883 compass.

  Serial.println("Setting scale to +/- 1.3 Ga");

  error = compass.SetScale(1.3); // Set the scale of the compass.

  if(error != 0) // If there is an error, print it out.
```

```

Serial.println(compass.GetErrorText(error));

Serial.println("Setting measurement mode to continous.");

error = compass.SetMeasurementMode(Measurement_Continuous);

if(error != 0) // If there is an error, print it out.

Serial.println(compass.GetErrorText(error));

}

void loop()

{

// put your main code here, to run repeatedly:

tampil_kompas();

// muncul_kompas();

hitung();

muncul_kiblat();

arah();

}

void hitung()

{


$$K = \frac{\sin((BX-BK)*PI/180)}{\cos((LX)*PI/180)*\tan((LK)*PI/180) - \sin((LX)*PI/180)*\cos((BX-BK)*PI/180)}$$


K1 = atan(K)*180/PI;

KK = (360-K1);

}

```



```

void tampil_kompas()
{
    MagnetometerRaw raw = compass.ReadRawAxis();

    MagnetometerScaled scaled = compass.ReadScaledAxis();

    int MilliGauss_OnThe_XAxis = scaled.XAxis;// (or Y Axis, or ZAxis)

    float heading = atan2(scaled.YAxis, scaled.XAxis);

    float declinationAngle = 0.01890773356;

    heading -= declinationAngle;

    // Correct for when signs are reversed.

    if(heading < 0)
        heading += TWO_PI;

    // Check for wrap due to addition of declination.

    if(heading > TWO_PI)
        heading -= TWO_PI;

    // Convert radians to degrees for readability.

    float headingDegrees = heading * RAD_TO_DEG;

    kmps = headingDegrees;

    //delay(500);}

void muncul_kompas()
{
    lcd.setCursor(0,0);

    lcd.print("Kompas Digital");

    lcd.setCursor(0,1);

```

```
lcd.print(kmps,4);  
lcd.print((char)223);  
}
```

```
void muncul_kiblat()  
{  
  lcd.setCursor(0,0);  
  lcd.print("Q= ");  
  lcd.print(KK,6);  
  lcd.print((char)223);  
  lcd.setCursor(0,1);  
  lcd.print("K= ");  
  lcd.print(kmps,6);  
  lcd.print((char)223);  
}
```

```
void arah()  
{  
  if(kmps>=KK-0.5 && kmps<=KK+0.5)  
  {  
    lcd.setCursor(0,1);  
    lcd.print("Arah Kiblat  ");  
  }  
}
```