LAMPIRAN

LAMPIRAN

1. *Datasheet* Komponen
   a. Arduino Uno

Parameter	Spesifikasi
Mikrokontroler	ATmega328P
Tegangan Operasi (Logic)	5V
Tegangan Input (rekomendasi)	7 – 12V
Tegangan Input (batas)	6 – 20V
Pin Digital I/O	14 (6 di antaranya mendukung PWM)
Pin PWM	6 (pin 3, 5, 6, 9, 10, 11)
Pin Analog Input	6 (A0–A5)
Arsitektur	AVR 8-bit
Clock Speed	16 MHz
Memori Flash	32 KB (ATmega328P) — 0.5 KB digunakan oleh bootloader
SRAM	2 KB
EEPROM	1 KB
Port USB	USB Type-B
Regulator Tegangan	5V dan 3.3V (3.3V terbatas hingga 50 mA)
Arus Maksimum per Pin I/O	40 mA (rekomendasi ≤ 20 mA)
Arus Total untuk 3.3V	Maks. 50 mA
Komunikasi	UART, SPI, I2C
Dimensi Fisik	68.6 mm x 53.4 mm
Berat	~25 g

b. Arduino Nano

Parameter	Spesifikasi
Mikrokontroler	ATmega328P
Tegangan Operasi (Logic)	5V (beberapa varian 3.3V)
Tegangan Input (melalui pin VIN)	6 – 12V
Tegangan Input (USB)	5V (melalui konektor Mini-USB)
Pin Digital I/O	14 (6 di antaranya mendukung PWM)
Pin PWM	6 (D3, D5, D6, D9, D10, D11)
Pin Analog Input	8 (A0–A7)
Clock Speed	16 MHz
Memori Flash	32 KB (0.5 KB digunakan oleh bootloader)
SRAM	2 KB
EEPROM	1 KB
Arus Maksimum per Pin I/O	40 mA (disarankan ≤ 20 mA)
Port USB	Mini-USB
Konverter USB-Serial	CH340, FT232, atau ATmega16U2 (tergantung versi board)
Komunikasi	UART, SPI, I2C
Dimensi	45 mm x 18 mm
Berat	~7 gram

c. Modul GPS Neo-6M

Parameter	**Spesifikasi**
Model Chip	u-blox NEO-6M
Antena	Patch ceramic + konektor u.FL untuk antena eksternal
Tegangan Operasi	3.3V – 5V (dengan regulator onboard)
Level Logika I/O	3.3V TTL (kompatibel dengan 5V mikrokontroler)
Protokol Komunikasi	UART (default), juga mendukung SPI dan I2C (jika dikonfigurasi)
Kecepatan Baudrate	Default 9600 bps
Waktu Akuisisi (TTFF)	Cold Start: < 38 detikWarm Start: < 35 detikHot Start: < 1 detik
Akurasi Posisi	< 2.5 meter (CEP 50%)
Sensitivitas	Tracking: –161 dBmAcquisition: –147 dBm
Jumlah Satelit Maksimal	22 tracking, 66 acquisition channels
Update Rate	1 Hz (default), hingga 5 Hz
Backup Battery (RTC)	Tersedia (biasanya CR1220 atau solder pad)
Dimensi Modul	~25 mm x 35 mm
Konektor Antena Eksternal	u.FL atau IPX
LED Indikator	Status fix GPS (berkedip saat belum fix, stabil saat fix)
Software Konfigurasi	u-center dari u-blox (untuk konfigurasi lanjutan via PC)

d. MPU6050

Parameter	Spesifikasi
Sensor	3-axis Accelerometer + 3-axis Gyroscope
Antarmuka Komunikasi	I²C (default), SPI (pada beberapa varian chip)
Alamat I²C Default	0x68 (atau 0x69 jika pin AD0 dihubungkan ke VCC)
Tegangan Operasi	2.3V – 3.4V (biasanya 3.3V)
Level Logika I/O	3.3V (kompatibel dengan 5V via modul breakout dengan regulator/level shifter)
Accelerometer Range	±2g, ±4g, ±8g, ±16g (dapat dipilih via register)
Gyroscope Range	±250, ±500, ±1000, ±2000 °/s (dapat dipilih via register)
Resolusi Akselerometer	16-bit
Resolusi Gyroscope	16-bit
Output Data Rate	Hingga 1 kHz
Buffer FIFO	1024 byte
Sensor Fusion Support	Ya, via Digital Motion Processor (DMP) internal
DMP (Motion Processor)	Mendukung pemrosesan gerakan onboard (perlu firmware khusus)
Dimensi Modul (umum)	Sekitar 20 x 15 mm (modul breakout)
Temperatur Operasional	–40 °C hingga +85 °C
Konsumsi Daya	Sekitar 3.9 mA (aktif), <10 µA (sleep)

e. LCD TFT 2.4 Inci

Parameter	Spesifikasi Umum
Ukuran Layar	2.4 inci TFT LCD
Resolusi	240 x 320 piksel (QVGA)
Driver IC Umum	ILI9341 / ILI9325 / SPFD5408 / HX8347
Antarmuka	8-bit paralel (kadang SPI untuk versi lain)
Kedalaman Warna	65K warna (16-bit RGB565)
Touchscreen	Ya (resistif, opsional)
Controller Touchscreen	XPT2046 atau ADS7843 (jika ada touchscreen)
Tegangan Operasi	3.3V (tetapi modul umumnya dilengkapi level shifter agar bisa 5V)
Tegangan Backlight LED	Biasanya 3.3V – 5V (diatur dengan resistor onboard)
Konektor	18 pin header (untuk langsung ditancap ke Arduino Uno)
Kompatibilitas	Arduino Uno (langsung), Arduino Mega (dengan wiring manual)
Konsumsi Daya	±40–80 mA tergantung backlight dan penggunaan
Ukuran Fisik Modul	Sekitar 70 mm x 50 mm

f. Push Button Toogle

Parameter	Spesifikasi Umum
Tipe	Push Button Toggle / Latching (bukan momentary)
Mode Operasi	**Latching** (tekan sekali ON, tekan lagi OFF)
Konfigurasi Kontak	SPST (Single Pole Single Throw) atau DPDT (varian)
Jumlah Kaki	2 kaki (SPST) atau 3–6 kaki (jika ada LED/polaritas/DPDT)
Tegangan Maksimum	12V – 250V (tergantung jenisnya, AC atau DC)
Arus Maksimum	0.5A – 5A (tergantung model dan tegangan)
Bahan Kontak	Tembaga berlapis perak atau emas
Durabilitas Mekanik	> 100.000 siklus tekan
Tahanan Kontak	< 50 mΩ
Tipe Pemasangan	Panel mount / PCB mount
Ukuran Umum	Bervariasi: 10 mm – 20 mm (diameter tombol)
Indikator LED (opsional)	Beberapa model dilengkapi LED (perlu wiring khusus)
Suhu Operasional	–25°C hingga +85°C

g. Push Button

Parameter	Spesifikasi
Nama Umum	Latching Push Button / Toggle Push Button / Push Lock Switch
Tipe Mekanik	Latching (mengunci posisi hingga ditekan ulang)
Mode Operasi	Tekan pertama: ON, tekan kedua: OFF
Konfigurasi Kontak	SPST (Single Pole Single Throw) paling umum
Jumlah Pin	2 (untuk switch), kadang 4 (jika ada LED internal)
Tegangan Maksimal	12V – 250V tergantung model (AC/DC)
Arus Maksimum	0.5A – 3A (tergantung model)
Ukuran Tombol	Bervariasi: 8 mm – 19 mm diameter
Tipe Pemasangan	Panel-mount atau PCB-mount
Fitur Tambahan	Bisa ada LED indikator (butuh wiring terpisah untuk LED)
Tahanan Kontak	< 50 mΩ
Umur Mekanis	≥ 100.000 siklus tekan
Contoh Produk	PBS-110B, PBS-33B, PBS-24, dll.

2. Program
   a. Arduino Nano

```
//-------------------------------------------------------
sudut
#include <Wire.h> // Library untuk komunikasi I2C
#include <MPU6050.h> // Library untuk MPU6050
MPU6050 mpu; // Membuat objek MPU6050
//-----------------------------------------------GPS
#include <TinyGPSPlus.h> // Library untuk GPS
#include <SoftwareSerial.h> // Library untuk komunikasi
serial software
static const int RXPin = 4, TXPin = 3;  // Pin RX dan TX
untuk GPS
static const uint32_t GPSBaud = 9600; // Baud rate GPS
TinyGPSPlus gps; // Membuat objek GPS
SoftwareSerial ss(RXPin, TXPin); // Membuat objek
SoftwareSerial

// Variabel global
float pitch = 0.0; // Variabel untuk sudut pitch
float lat = 0.00; // Variabel latitude GPS
float lng = 0.00; // Variabel longitude GPS
String data = ""; // String untuk data output
int pinSta1 = 5; // Pin tombol pertama
int pinSta2 = 6; // Pin tombol kedua

unsigned long lastSudutMillis = 0; // Waktu terakhir membaca
sudut
const unsigned long intervalSudut = 1000; // Interval baca
sudut (1 detik)

void setup()
{
  Serial.begin(9600); // Memulai komunikasi serial dengan
baud 9600
pinMode(pinSta1, INPUT_PULLUP); // Set pin tombol pertama
sebagai input dengan pull-up
pinMode(pinSta2, INPUT_PULLUP); // Set pin tombol kedua
sebagai input dengan pull-up

  //-----------------------------------------sudut
Wire.begin(); // Memulai komunikasi I2C
mpu.initialize(); // Inisialisasi MPU6050
if (!mpu.testConnection()) { // Cek koneksi MPU6050
Serial.println("MPU tidak terhubung!"); // Tampilkan error
jika gagal koneksi
while (1);  // Hentikan program jika gagal koneksi
}


  //-----------------------------------------GPS
  ss.begin(GPSBaud); // Memulai komunikasi serial software
GPS
}

void bacaSudut() {
```

```
  int16_t ax, ay, az, gx, gy, gz; // Variabel untuk nilai akselerasi dan gyro
  mpu.getAcceleration(&ax, &ay, &az); // Baca nilai akselerasi MPU6050
  mpu.getRotation(&gx, &gy, &gz); // Baca nilai gyro MPU6050
  pitch = atan2(ay, az) * (180.0 / PI); // Hitung sudut pitch dalam derajat
}

void bacaGPS()
{
while (ss.available() > 0) // Selama ada data di serial GPS
if (gps.encode(ss.read())) // Decode data GPS
{
if (gps.location.isValid()) // Jika lokasi valid
{
lat = gps.location.lat(); // Ambil latitude
lng = gps.location.lng();  // Ambil longitude
      }
      else
      {
lat = 0.00;  // Jika tidak valid, set latitude 0
lng = 0.00; // Jika tidak valid, set longitude 0
      }
    }

if (millis() > 5000 && gps.charsProcessed() < 10) // Jika 5 detik lebih dan data GPS kurang dari 10 karakter
  {
    Serial.println("GPS tidak terdeteksi."); // Tampilkan pesan error GPS
    while (true);  // Hentikan program
  }
}

void loop()
{
  bacaGPS(); // Panggil fungsi baca GPS

  unsigned long currentMillis = millis(); // Ambil waktu sekarang


 // Cek jika sudah waktunya membaca sudut
  if (currentMillis - lastSudutMillis >= intervalSudut) {
    lastSudutMillis = currentMillis; // Update waktu terakhir baca sudut
    bacaSudut() ;// Panggil fungsi baca sudut

    // Ambil data tombol
int sta1 = digitalRead(pinSta1); // Baca status tombol 1
    int sta2 = digitalRead(pinSta2); // Baca status tombol 2

    // Kirim data
```

```
    data = String(sta1) + '#' + String(sta2) + '#' +
    String(pitch) + '#' + String(lat, 6) + '#' + String(lng,
    6); // Gabung semua data jadi satu string dengan '#'
    sebagai pemisah
    Serial.println(data); // Kirim data ke serial
  }
```

b. Arduino Uno

```
#include <Adafruit_GFX.h>//Library grafis untuk tampilan LCD TFT
#include <MCUFRIEND_kbv.h> // Library khusus untuk driver TFT 2.4"
shield
MCUFRIEND_kbv tft; // Objek TFT untuk mengakses fungsi LCD
#include <Fonts/FreeSans9pt7b.h> // Font ukuran 9pt dari Adafruit
#include <Fonts/FreeSans12pt7b.h> // Font ukuran 12pt dari Adafruit
#include <Fonts/FreeSerif12pt7b.h> // Font Serif 12pt dari Adafruit
#include <FreeDefaultFonts.h> // Font default tambahan
// Deklarasi variabel global bertipe String untuk menampung data
dari sensor
String dataSensor;
String sudut;
String lat;
String lng;
String jarak;
String kemiringan;
String sta;
// Warna-warna dalam format 16-bit RGB565
#define BLACK   0x0000
#define BLUE    0x001F
#define RED     0xF800
#define GREEN   0x07E0
#define CYAN    0x07FF
#define MAGENTA 0xF81F
#define YELLOW  0xFFE0
#define WHITE   0xFFFF
#define GREY    0x8410
#define ORANGE  0xE880
// Variabel untuk menyimpan lokasi sebelumnya (GPS)
float prevLat = 0.0;
float prevLng = 0.0;
bool firstPoint = true; // Penanda titik pertama
// Variabel untuk menghitung rata-rata sudut
float averageSudut = 0.0;
int sudutCount = 0;
String prevSta = "0"; // Menyimpan status sebelumnya
bool buttonPressed = false; // Flag tombol (tidak digunakan di
sini)
   // Fungsi untuk menghitung jarak antara dua titik GPS
   menggunakan rumus Haversine
   float haversine(float lat1, float lon1, float lat2, float lon2)
{
     const float R = 6371000; // Radius bumi dalam meter
     float dLat = radians(lat2 - lat1); // Konversi delta latitude
ke radian
```

```
    float dLon = radians(lon2 - lon1); // Konversi delta longitude
ke radian
    lat1 = radians(lat1); // Konversi latitude awal
    lat2 = radians(lat2); // Konversi latitude akhir

    float a = sin(dLat / 2) * sin(dLat / 2) +
              cos(lat1) * cos(lat2) *
              sin(dLon / 2) * sin(dLon / 2); // Rumus Haversine
    float c = 2 * atan2(sqrt(a), sqrt(1 - a));
    return R * c;  // Jarak dalam meter
  }

  void setup(void)
  {
      Serial.begin(9600); // Inisialisasi komunikasi serial
      uint16_t ID = tft.readID(); // Baca ID driver dari TFT
      if (ID == 0xD3D3) ID = 0x9481; // Jika ID tidak valid, pakai
  default (umum  pada clone)
      tft.begin(ID); // Mulai komunikasi dengan TFT
      tft.setRotation(1);  // Atur orientasi layar horizontal
      tft.fillScreen(BLACK); // Bersihkan layar jadi hitam
      tampilanAwal(); // Tampilkan tampilan awal
  }

  void tampilanAwal()
  {
    // Menampilkan semua label dan judul pada layar awal
    tampil(25, 20, 1, &FreeSans12pt7b, GREEN, "Detektor Kemiringan
Jalan");
    tft.drawFastHLine(10, 30, tft.width()-10, GREY);
    tampil(10, 60, 1, &FreeSans9pt7b, CYAN, "Koord. Lokasi");
    tampil(20, 83, 1, &FreeSans9pt7b, CYAN, "-  Longitude");
    tampil(135, 83, 1, &FreeSans9pt7b, CYAN, ":");
    tampil(20, 106, 1, &FreeSans9pt7b, CYAN, "-  Latitude");
    tampil(135, 106, 1, &FreeSans9pt7b, CYAN, ":");
    tampil(10, 133, 1, &FreeSans9pt7b, CYAN, "Sudut Terbaca");
    tampil(135,133, 1, &FreeSans9pt7b, CYAN, ":");
    tft.drawFastHLine(10, 150, tft.width()-120, GREY);
    tampil(220, 153, 1, &FreeSans9pt7b, MAGENTA, "Hasil Akhir");
    tampil(10, 180, 1, &FreeSans12pt7b, YELLOW, "Jarak (m)");
    tampil(160, 180, 1, &FreeSans12pt7b, YELLOW, "Kemiringan");
    tampil(277, 200, 1, &FreeSans12pt7b, WHITE, "o"); // Derajat
  }

  // Fungsi untuk membaca data dari serial
  void bacaData()
  {
    while(Serial.available()>0) // Jika ada data masuk
    {
      delay(10); // Delay sebentar agar data lengkap
      char c = Serial.read();  // Baca karakter
      dataSensor += c; // Tambahkan ke string
    }

    // Format: status#sudut#lat#lng
    if (dataSensor.length() > 0)
```

```
      {
        clearDisplay(); // Bersihkan tampilan lama
        sta  = pisahData(dataSensor, '#', 0); // Pisah data status
        sudut = pisahData(dataSensor, '#', 1); // Pisah data sudut
        lat = pisahData(dataSensor, '#', 2); // Pisah data latitude
        lng = pisahData(dataSensor, '#', 3); // Pisah data longitude
        Serial.println(sta); // Cetak ke serial monitor
        Serial.println(sudut);
        Serial.println(lat);
        Serial.println(lng);
        dataSensor = ""; // Reset string setelah dibaca
      }
    }

    // Fungsi untuk memisahkan string berdasarkan separator
    String pisahData(String data, char separator, int index)
    {
      int found = 0;
      int strIndex[] = {0, -1};
      int maxIndex = data.length() - 1;
      for (int i = 0; i <= maxIndex && found <= index; i++) {
        if (data.charAt(i) == separator || i == maxIndex) {
            found++;
            strIndex[0] = strIndex[1] + 1;
            strIndex[1] = (i == maxIndex) ? i + 1 : i;
        }
      }
      return   found   >   index   ?   data.substring(strIndex[0],
strIndex[1]) : "";
    }
    // Fungsi untuk menghapus tampilan nilai lama (dengan menimpa
warna hitam)
    void clearDisplay()
    {
      tampil(150, 83, 1, &FreeSans9pt7b, BLACK, lat);
      tampil(150, 106, 1, &FreeSans9pt7b, BLACK, lng);
      tampil(150, 135, 1, &FreeSans12pt7b, BLACK, sudut);
    }

    // Fungsi untuk menampilkan data real-time ke layar
    void display()
    {
      tampil(150, 83, 1, &FreeSans9pt7b, BLACK, lat);
      tampil(150, 83, 1, &FreeSans9pt7b, YELLOW, lat); // latitude
      tampil(150, 106, 1, &FreeSans9pt7b, BLACK, lng);
      tampil(150, 106, 1, &FreeSans9pt7b, YELLOW, lng); // longitude
      tampil(150, 135, 1, &FreeSans12pt7b, BLACK, sudut);
      tampil(150, 135, 1, &FreeSans12pt7b, YELLOW, sudut); // sudut
terbaca
    }
    // Fungsi untuk menghitung jarak dan rata-rata kemiringan
    void hitung()
    {
      tampil(10, 225, 2, &FreeSans12pt7b, BLACK, jarak); // Hapus
jarak lama
```

```
    tampil(160, 225, 2, &FreeSans12pt7b, BLACK, kemiringan); // Hapus kemiringan lama


    float currentLat = lat.toFloat(); // Konversi latitude ke float
    float currentLng = lng.toFloat(); // Konversi longitude ke float
    float distance = 0.0;

    if (!firstPoint) {
      distance = haversine(prevLat, prevLng, currentLat, currentLng); // Hitung jarak
    } else {
      firstPoint = false; // Menandai bahwa titik pertama sudah di-set
    }

   prevLat = currentLat; // Simpan titik saat ini sebagai titik sebelumnya
    prevLng = currentLng;

    jarak = String(distance, 2); // Konversi hasil jarak jadi string
    kemiringan = (sudutCount > 0) ? String(averageSudut / sudutCount, 2) : "0.00"; // Rata-rata sudut

    tampil(10, 225, 2, &FreeSans12pt7b, WHITE, jarak); // Tampilkan jarak
    tampil(160, 225, 2, &FreeSans12pt7b, WHITE, kemiringan); // Tampilkan kemiringan

    Serial.print("Jarak (m): "); Serial.println(jarak); // Kirim ke serial monitor
    Serial.print("Kemiringan (o): "); Serial.println(kemiringan);
  }

  // Fungsi utama yang berjalan terus-menerus
  void loop()
  {
    bacaData(); // Ambil data dari serial

    // Reset variabel jika status berubah dari 0 ke 1
    if (sta == "1" && prevSta == "0") {
      averageSudut = 0.0;
      sudutCount = 0;
      firstPoint = true;
    }

    if (sta == "1") {
      display(); // Tampilkan data real-time
      float sudutFloat = sudut.toFloat();
      averageSudut += sudutFloat; // Tambahkan ke total sudut
      sudutCount++; // Tambah jumlah data sudut
    }
    else if (sta == "0") {
```

```
        hitung(); // Jika tombol dilepas, hitung hasil
      }

      prevSta = sta; // Simpan status sekarang sebagai status sebelumnya
      delay(100); // Delay pendek untuk mengurangi frekuensi update
    }

    // Fungsi untuk mencetak teks ke layar dengan font, warna, dan posisi tertentu
    void tampil(int x, int y, int sz, const GFXfont *f, int warna, String nilai)
    {
        tft.setFont(f); // Atur font
        tft.setCursor(x, y); // Atur posisi kursor
        tft.setTextColor(warna); // Atur warna teks
        tft.setTextSize(sz); // Atur ukuran teks
        tft.print(nilai); // Tampilkan nilai ke layar
    }
```

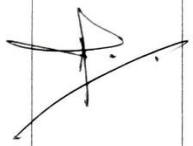
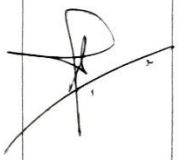
LOGBOOK CAPSTONE DESIGN
PROGRAM STUDI TEKNIK ELEKTRO
UNIVERSITAS MUHAMMADIYAH PAREPARE

GAMBARAN UMUM CAPSTONE DESIGN PROJECT

Judul Capstone Design Project	**Rancangan Sistem Pemetaan Kemiringan Jalan Menggunakan Data Elevasi dari GPS dan Gyroscope**		
Permasalahan:		Solusi:	
Lokasi :			
Mahasiswa 1:	**Reza Andika**	NIM:	221180007
Mahasiswa 2:	**Apriyana Munir**	NIM:	221180016
Dosen Pembimbing 1	**Muh. Basri, ST., MT**	NBM / NIDN	959 773
Dosen Pembimbing 2	**Dr. H. Hakzah, ST., MT**	NBM / NIDN	938 317
Spesifikasi Desain (*Design Specification*):			
Durasi Proyek	Tanggal mulai: 04 Mei 2024	Tanggal rencana penyelesaian: 5 Februari 2025 Tanggal realisasi penyelesaian: 18 Mei 2025	

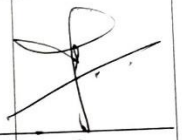
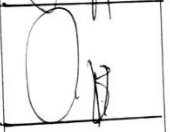
LOGBOOK CAPSTONE DESIGN
PROGRAM STUDI TEKNIK ELEKTRO
UNIVERSITAS MUHAMMADIYAH PAREPARE

URAIAN PROGRES KEGIATAN

Waktu Pelaksanaan	Item Kegiatan dan Hasil	Bukti Kegiatan	Keterangan	Tanda Tangan Pembimbing
Senin, 4 Nov 2024	kami berdua melakukan Study literatur dimana kami membuat 5 resume jurnal yg telah di dapat dan ~~menghasilkan~~ membuat tabel berisi parameter, mengetahui cara menghitung jarak dari perubahan GPS dan standar kemiringan dalan			
Selasa, 4 Nov 2024	Kami melakukan bimbingan dengan dosen pembimbing ~~satu~~ 1 dan dosen pembimbing 2 dan mengetahui cara kerja, alat dan bahan dari alat yg akan dibuat			
Senin, 23 Des 2024	kami melakukan bimbingan dan merevisi format penulisan dari skripsi yg telah disusun + sistematika penulisan			

LOGBOOK CAPSTONE DESIGN
PROGRAM STUDI TEKNIK ELEKTRO
UNIVERSITAS MUHAMMADIYAH PAREPARE

URAIAN PROGRES KEGIATAN

Waktu Pelaksanaan	Item Kegiatan dan Hasil	Bukti Kegiatan	Keterangan	Tanda Tangan Pembimbing
Selasa, 24 Des 2024	Kami melakukan bimbingan dan merevisi format penulisan dari skripsi yang telah disusun.			
Senin, 6 Januari 2025	Kami melakukan bimbingan dan membahas alat & bahan yang akan digunakan			
Senin, 17 Februari 2025	~~Pengambilan~~ Pengujian modul GPS Neo-6m dan memperoleh hasil keakuratan kinerja modul GPS			
Kamis, 27 Februari 2025	Pengujian sensor MPU6050 dan Push buttun, memperoleh hasil bahwa MPU6050 dan push button berfungsi dengan baik			
Minggu, 18 Mei 2025	Pengujian alat yg telah dirancang dan memperoleh hasil ~~ke~~ jarak dan kemiringan jalan			

4. Lembar Kontrol Capstone Design

LEMBAR KONTROL
CAPSTONE DESIGN PROJECT

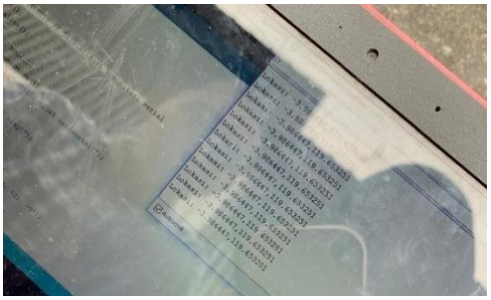
Tim Pelaksana	**Reza Andika** NIM. 221180007	**Apriyana Munir** NIM. 221180016	PROGRAM STUDI **TEKNIK ELEKTRO** UNIVERSITAS MUHAMMADIYAH PAREPARE
Dosen Pembimbing	**Muh. Basri, ST., MT.** Pembimbing I	**Dr. H. Hakzah, ST., MT.** Pembimbing II	
Judul Proyek	Rancang Sistem Pemetaan Kemiringan Jalan Menggunakan Data Elevasi dari GPS dan Gyroscope		

No	Tahapan	Kegiatan	Waktu Pelaksanaan	Tanda Tangan Pembimbing 1	Tanda Tangan Pembimbing 2	Tanda Tangan Instruktur
1	Identifikasi dan Perumusan Masalah	**Pengumpulan Data** - Data statistik tingkat kecelakaan yang disebabkan oleh kemiringan jalan. Sumber Data : Dinas Perhubungan/Jasa Marga/Kepolisian. **Studi Literatur** - Standar kemiringan jalan, Metode pengukuran kemiringan jalan Sumber Data : Buku referensi, Jurnal terkait. **Rumusan Masalah** - Keterbatasan metode pengukuran manual, kebutuhan data real time, efisiensi pengukuran. **Identifikasi Variabel** - Koordinat Lokasi - Koordinat dan ketinggian titik awal dan titik akhir pengambilan sample sudut. - Sudut kemiringan	10 hari			
2	Teknik Analisis Data (teoritis)	**Koordinat Lokasi** Mengetahui cara menggunakan GPS untuk menentukan lokasi jalan yang miring. **Koordinat dan ketinggian titik awal dan titik akhir pengambilan sample sudut.**	7 hari			

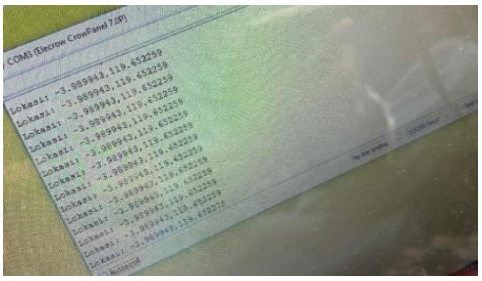
No	Tahapan	Kegiatan	Waktu Pelaksanaan	Tanda Tangan Pembimbing 1	Tanda Tangan Pembimbing 2	Tanda Tangan Instruktur
		Mengetahui cara mengukur koordinat dan ketinggian titik awal dan titik akhir pengambilan sample sudut menggunakan GPS. **Sudut Kemiringan** Mengetahui cara menggunakan regresi linier, phytagoras, fungsi trigonometri.				
3	Perancangan dan Realisasi Sistem	**Perancangan Perangkat Keras** - Membuat blok diagram system - Pemilihan dan Penyediaan komponen - Pengujian dan Kalibrasi komponen - Analisis data hasil Pengujian Komponen **Perancangan Perangkat Lunak** - Pembuatan Flowchart Sistem **Realisasi Sistem** - Perakitan Perangkat Keras - Pembuatan Perangkat Lunak	20 hari 2 hari 30 hari			
4	Pengujian Sistem	- Menetapkan Metode Pengujian - Membuat Prosedur Pengujian - Pelaksanaan Pengujian Lapangan/lokasi	10 hari			
5	Analisis Data Hasil Pengujian	**Melakukan analisis data dari hasil pengujian Lapangan :** - Koordinat Lokasi. - Koordinat dan ketinggian titik awal dan titik akhir pengambilan sample sudut. - Menghitung Sudut kemiringan Jalan mengunakan Regresi Linier, Phytagoras, dan persamaan fungsi trigonometri.	7 hari			
6	Laporan dan Dokumentasi	- Dokumentasi Tiap Pelaksanaan Kegiatan - Penyusunan Laporan Hasil Rancangan Perangkat Keras, Perangkat Lunak, Pengujian, dan analisis data sesuai format yang telah ditetapkan	14 hari			

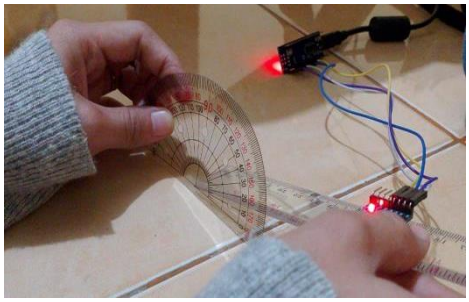
5. Data Pendukung Lainnya

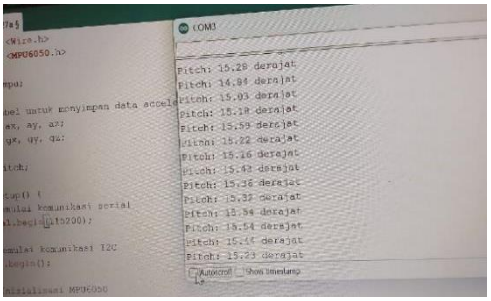






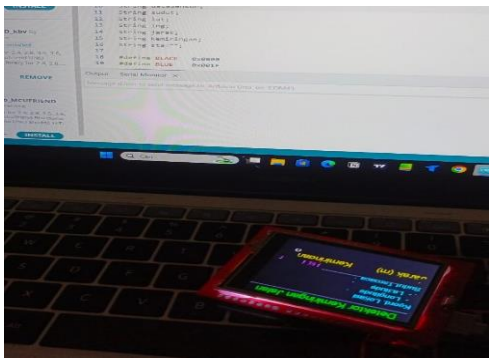


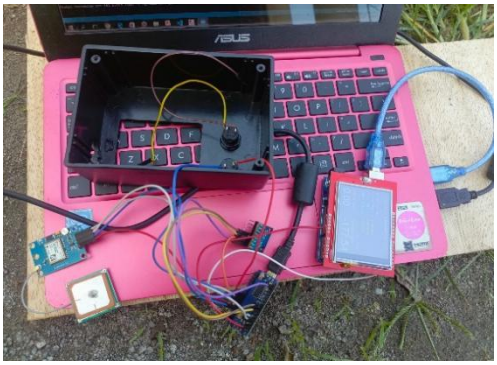















Jarak (m)
120.36
8.23