DOC. EU-100

Identifikasi Masalah

1.1. Latar Belakang Masalah

Jalan adalah sarana transportasi darat yang mencakup semua bagian jalan, termasuk bangunan penghubung, bangunan pelengkap, dan perlengkapannya yang ditujukan untuk lalu lintas. Jalan dapat berada di permukaan tanah, di atas atau di bawah permukaan tanah, di atas permukaan air, kecuali jalan rel, jalan lori, dan jalan kabel (UU No. 2 Tahun 2022)

Kemiringan jalan merupakan kondisi dimana jalan memiliki perbedaan tinggi pada titik awal dan akhir jalan. Perbedaan tinggi tersebut kemudian dinyatakan dalam persentase kemiringan atau derajat kemiringan. Sehingga, kemiringan jalan sendiri dapat diartikan sebagai perbandingan perbedaan tinggi jalan dengan panjang jalan. Perhitungan kemiringan jalan biasanya digunakan dalam perencanaan pembangunan jalan raya, jembatan, dan infrastruktur transportasi lainnya. Hal ini bertujuan untuk menentukan keamanan serta kenyamanan bagi para pengendara dan pengguna jalan. Oleh sebab itu, pemerintah melalui dinas terkait selalu memperhatikan kemiringan jalan agar sesuai dengan standar nasional yang telah ditentukan (Rahmat Ardiansyah, 2020)

Selain itu, kemiringan jalan juga berpengaruh pada kecepatan kendaraan dan pengelolaan laju air hujan yang mengalir ke permukaan jalan. Jika kemiringan jalan terlalu curam, maka kendaraan akan sulit untuk melewati jalan tersebut, terlebih jika jalan basah akibat curah hujan. Selanjutnya, kemiringan jalan yang tepat akan

membantu air hujan mengalir dengan baik sehingga tidak menimbulkan genangan air yang bisa membahayakan pengendara. Dalam perhitungannya, kemiringan jalan biasanya dinyatakan dalam persentase atau derajat kemiringan. Pada persentase kemiringan, hasil perhitungan dinyatakan dalam satu persen atau pecahan satuan persen. Sedangkan pada derajat kemiringan, hasil perhitungan dinyatakan dalam satuan derajat seperti umumnya pengukuran sudut. Dalam perencanaan pembangunan jalan raya, terdapat standar nilai kemiringan jalan yang harus dipenuhi. Standar tersebut untuk memastikan bahwa jalan yang dibangun aman dan nyaman bagi para pengguna. Menurut Kriteria Teknis Jalan (KTJ, 2019).

Penelitian terdahulu telah menunjukkan berbagai pendekatan dalam mendeteksi kemiringan jalan menggunakan pengolahan citra digital. Misalnya, penelitian oleh Chen et al. (2021) menggunakan kombinasi deteksi tepi Canny dan jaringan segmentasi semantik untuk mendeteksi retakan dan garis pada citra jalan, dan berhasil mendeteksi kemiringan dengan tingkat akurasi yang tinggi. Selain itu, studi oleh Li et al. (2021) mengembangkan metode estimasi kemiringan jalan berbasis visi dengan menganalisis garis jalan dan fitur lokal yang diekstraksi dari gambar, memberikan pendekatan yang akurat dan robust untuk estimasi kemiringan. Adapula penelitian oleh Bagus Nurhannuddin (2024) berfokus pada pengembangan sistem sederhana untuk mendeteksi kemiringan jalan menggunakan metode Otsu Thresholding. Tujuan utamanya adalah menciptakan sistem yang efektif dan efisien yang mampu mengidentifikasi berbagai tingkat kemiringan jalan dengan akurat. Sistem ini memanfaatkan teknik pemrosesan citra, di mana metode

Otsu Thresholding diterapkan untuk membedakan antara permukaan jalan dan lingkungan sekitarnya.

Berdasarkan penelitian di atas yang masih bergantung pada pengolahan gambar menjadi citra biner dan tidak dapat mengukur kemiringan jalan pada titik tertentu. Maka penelitian ini berfokus pada perancangan alat sistem pemetaan kemiringan jalan menggunakan data elevasi dari GPS dan gyroscope. Alat ini mampu diatur untuk mengukur kemiringan jalan sesuai dengan titik lokasi yang akan di ukur. Alat ini menggunakan GPS dan gyroscope sebagai perangkat utama untuk menentukan kemiringan jalan. Selain itu alat ini mudah digunakan, portable dan ekonomis.

1.2. Rumusan Masalah

1. Bagaimana cara mengetahui pengukuran kemiringan jalan dengan menggunakan metode pengukuran manual?
2. Bagaimana merancang sistem untuk melakukan pengukuran secara real time terhadap kemiringan jalan?
3. Bagaimana membandingkan efisiensi pengukuran antara metode pengukuran manual dan pengukuran secara real time?

1.3. Karakteristik Solusi

Rancangan sistem pemetaan kemiringan jalan menggunakan data elevasi dari GPS dan gyroscope memiliki beberapa fitur utama yang inovatif. Pertama, sistem ini dilengkapi dengan kemampuan akuisisi data elevasi secara real-time melalui modul GPS yang dapat menangkap data titik koordinat lokas awal dan akhir

pengukuran. Selain itu, sistem juga memanfaatkan sensor gyroscope untuk mengukur kemiringan kendaraan terhadap bidang horizontal, yang memberikan informasi tentang kemiringan jalan. Data elevasi dari GPS dan data kemiringan dari gyroscope kemudian digabungkan melalui algoritma fusi data yang canggih. Teknik filtering diterapkan untuk mengurangi noise dan outlier, menghasilkan data kemiringan jalan yang lebih akurat dan andal. Hasil pengukuran kemiringan jalan selanjutnya akan ditampilkan pada layar sistem.

Sistem ini juga menyediakan fitur start/stop yang membuat pengguna dapat mengukur kemiringan jalan hanya pada titik yang diinginkan. Informasi kemiringan jalan dapat dimanfaatkan untuk optimalisasi perencanaan rute, perhitungan konsumsi bahan bakar, atau pemberian peringatan keselamatan saat mengemudi di jalan dengan kemiringan yang signifikan. Identifikasi area dengan kemiringan ekstrim, atau perbandingan kemiringan antar wilayah. Dengan kombinasi fitur-fitur canggih tersebut, solusi pemetaan kemiringan jalan menggunakan data elevasi dari GPS dan gyroscope menawarkan pendekatan yang komprehensif dan real-time untuk memahami kondisi kemiringan jalan. Sistem ini berpotensi memberikan manfaat signifikan dalam berbagai aplikasi, mulai dari peningkatan keselamatan jalan, optimalisasi perencanaan rute, hingga pengelolaan infrastruktur jalan yang lebih efisien. Dengan informasi kemiringan jalan yang akurat dan mudah diakses, pengambilan keputusan yang lebih baik dapat dilakukan untuk menciptakan sistem transportasi yang lebih aman, nyaman, dan berkelanjutan.

Mengukur keberhasilan solusi dari rancangan sistem pemetaan kemiringan jalan menggunakan data elevasi dari GPS dan gyroscope, terdapat beberapa kriteria

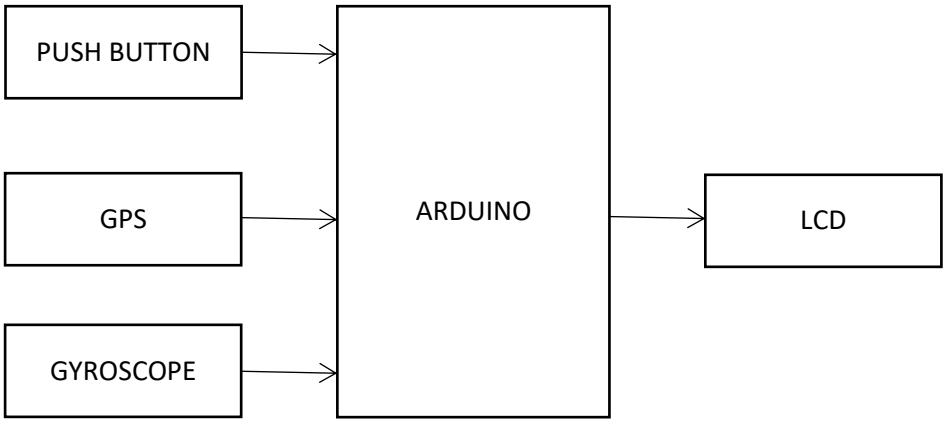
yang harus dipenuhi. Pertama, sistem harus mampu mengakuisisi data elevasi dari GPS. Kedua, gyroscope yang digunakan harus memiliki sensitivitas yang tinggi. Algoritma pengolahan data gyroscope harus dapat mengkompensasi drift dan meminimalkan gangguan getaran kendaraan. Hasil perhitungan kemiringan jalan harus memiliki akurasi minimal 0,5 derajat dibandingkan dengan pengukuran manual menggunakan alat ukur kemiringan. Selain akurasi data, resolusi spasial juga menjadi pertimbangan penting. Sistem harus mampu memetakan kemiringan jalan dengan resolusi spasial minimal 10 meter, artinya setiap segmen jalan sepanjang 10 meter memiliki nilai kemiringan yang terukur. Semakin tinggi resolusi spasial, semakin detail informasi kemiringan jalan yang dapat dipetakan. Cakupan area pemetaan juga harus mencakup minimal 90% dari total panjang jalan yang ada di wilayah yang ditargetkan.

Efisiensi dalam pengumpulan data juga menjadi kriteria keberhasilan. Proses pengumpulan data harus efisien dan dapat dilakukan dengan menggunakan kendaraan survei yang dilengkapi dengan GPS dan gyroscope. Sistem harus dapat mengumpulkan data kemiringan jalan secara otomatis tanpa memerlukan intervensi manual yang signifikan. Dan hasil pemetaan kemiringan jalan harus divisualisasikan secara jelas dan mudah dipahami. Keandalan dan ketahanan sistem juga menjadi faktor penting. Perangkat keras dan perangkat lunak yang digunakan dalam sistem harus andal dan tahan terhadap kondisi lapangan yang bervariasi. Sistem harus dapat beroperasi secara kontinu dalam jangka waktu yang lama tanpa mengalami kegagalan atau penurunan performa yang signifikan.

Dengan memenuhi kriteria-kriteria tersebut, sistem pemetaan kemiringan jalan berbasis data elevasi dari GPS dan gyroscope diharapkan dapat menghasilkan peta kemiringan jalan yang akurat, detail, dan bermanfaat bagi berbagai aplikasi seperti perencanaan rute, manajemen infrastruktur jalan, dan analisis keselamatan lalu lintas.

1.4. Skenario Produk

Rancangan Sistem Pemetaan Kemiringan jalan Menggunakan Data Elevasi Dari GPS dan Gyroscope. Sistem ini bertujuan untuk mengukur tingkat kemiringan jalan. Produk ini nantinya memiliki tombol start/stop sehingga tidak semua panjang jalan akan diukur dengan kata lain hanya titik lokasi yang ingin diukur saja yang akan diproses oleh produk ini. Diagram blok produk dapat dilihat pada gambar 1.1.



Gambar 1. 1 Diagram Blok

Alur kerja dari sistem pemetaan kemiringan jalan menggunakan Data elevasi dari GPS dan gyroscope sesuai Gambar 1.1 adalah pertama push button ditekan untuk mrmulai pengambilan data koordinat awal GPS dan gyroscope akan memulai

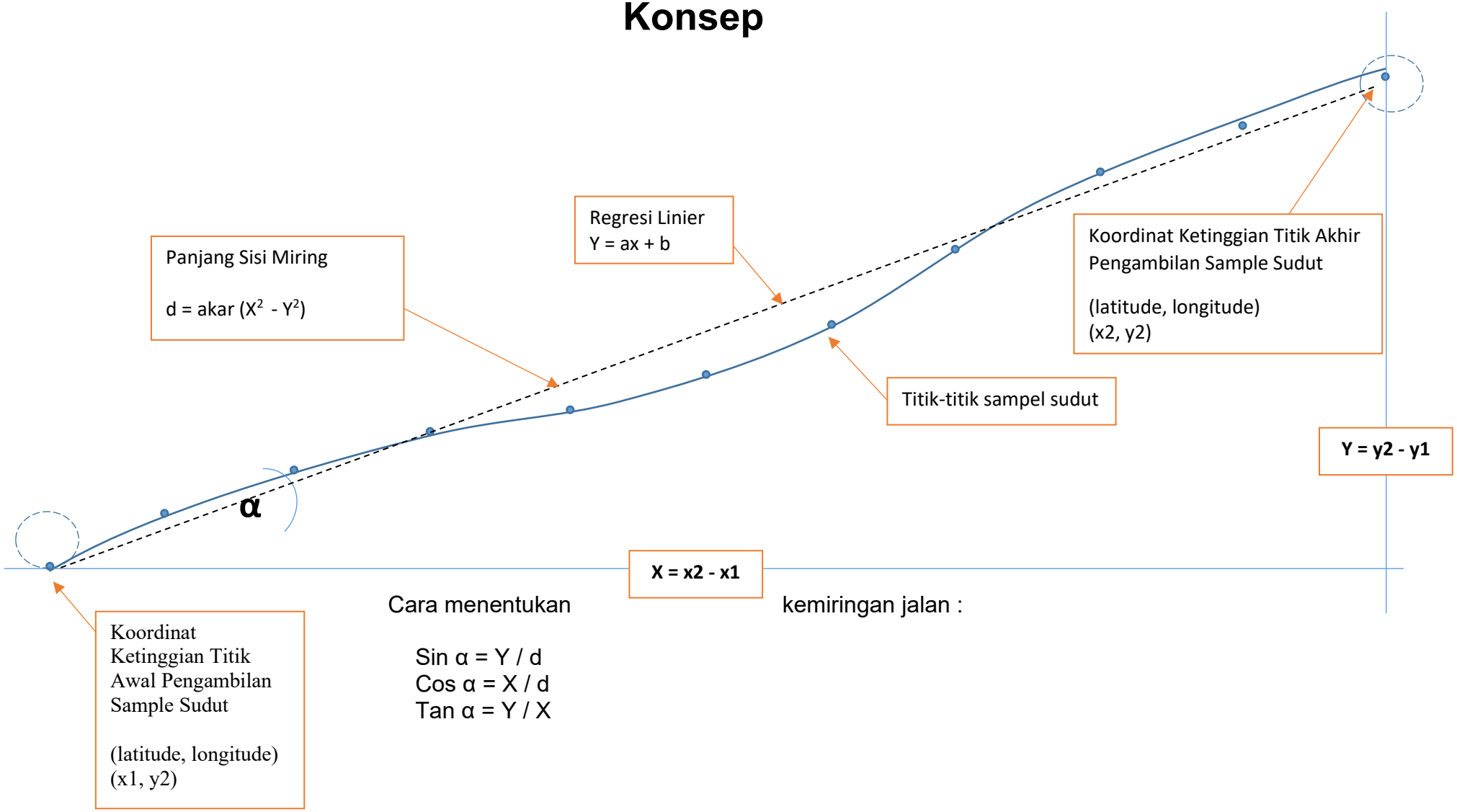
mengambil data sudut kemiringan setiap 1 detik. Tekan kembali push button untuk mengambil koordinat akhir GPS dan berhenti mengambil data sudut kemiringan. Data yang didapatkan oleh GPS dan gyroscope akan diproses oleh Arduino dan hasilnya akan ditampilkan dilayar LCD TFT. Desain produk dapat dilihat pada gambar 1.2.



Gambar 1. 2 Desain Produk

Pada Gambar 1.2, bentuk perancangan sistem memiliki dimensi 10x8 cm. Bahan yang digunakan adalah plastik, dimana bahan tersebut memiliki beban yang ringan dan cukup kuat untuk menopang seluruh sistem didalamnya. Sistem tersebut memiliki 2 tombol yaitu tombol merah dan tombol kuning. Tombol merah berfungsi untuk start/stop pengambilan data. Tombol kuning berfungsi untuk menampilkan hasil dari data yang telah diambil. Pada layar monitor menggunakan Display TFT 2,4”. Pada layar akan ditampilkan koordinat awal, koordinat akhir dan jarak serta sudut kemiringan akhir hasil pengukuran.

Konsep



Cara menentukan kemiringan jalan :

Sin α = Y / d
Cos α = X / d
Tan α = Y / X

1.5. Keuntungan Produk

A. Manfaat Langsung

1. Pemetaan Kemiringan yang Akurat dan Real-time.

Alat ini bisa mendapatkan peta kemiringan jalan dengan akurasi tinggi tanpa perlu survei manual yang memakan waktu. Pemetaan kemiringan jalan memungkinkan pembuatan peta kontur yang lebih detil, mengidentifikasi titik-titik dengan kemiringan ekstrem atau potensi masalah seperti turunan tajam atau tanjakan curam.

2. Keamanan Pengguna Jalan

Teknologi ini membantu mengidentifikasi kemiringan yang tidak sesuai standar keselamatan, seperti tanjakan atau turunan yang terlalu curam, sehingga pihak berwenang dapat menindaklanjuti dengan perubahan desain atau memasang rambu peringatan. Dengan mengukur dan menyesuaikan kemiringan secara tepat, kecelakaan yang disebabkan oleh jalan yang terlalu miring bisa diminimalisir.

3. Pemeliharaan Jalan dan Deteksi Kerusakan

Data elevasi dapat digunakan untuk memantau perubahan pada permukaan jalan, misalnya adanya penurunan elevasi atau pergeseran tanah yang bisa menyebabkan kerusakan atau kecelakaan. Jika ada perubahan kemiringan yang signifikan, hal ini bisa mengindikasikan masalah struktural seperti pergeseran tanah atau kerusakan pada jalan. Dengan data real-time, pemeliharaan jalan bisa dilakukan lebih dini dan preventif, mengurangi biaya perbaikan yang besar di masa depan.

B. Manfaat Tidak Langsung

1. Efisiensi Biaya dan Sumber Daya

Dengan pemetaan digital menggunakan GPS dan gyroscope, survei lapangan yang membutuhkan tenaga kerja manual dan waktu lebih lama dapat dikurangi. Ini mengurangi biaya tenaga kerja dan perjalanan. Mengurangi kesalahan manusia dalam pengukuran kemiringan dan perencanaan yang dapat mengarah pada perubahan desain yang mahal atau bahkan perbaikan infrastruktur setelah konstruksi selesai.

2. Peningkatan Pengalaman Pengendara

Data kemiringan yang lebih presisi menghasilkan desain jalan yang lebih nyaman, mengurangi kejutan atau rasa tidak nyaman bagi pengendara yang melintasi tanjakan atau turunan yang terlalu tajam. Jalan yang dirancang dengan kemiringan yang lebih optimal membantu kendaraan bergerak lebih efisien, mengurangi konsumsi bahan bakar dan keausan kendaraan, yang pada gilirannya meningkatkan pengalaman berkendara secara keseluruhan.

3. Perencanaan Transportasi yang Lebih Baik

Data kemiringan jalan dapat digunakan untuk merencanakan rute transportasi yang lebih efisien, menghindari tanjakan ekstrem atau area rawan longsor yang bisa menambah beban kendaraan, serta memastikan pengiriman barang lebih cepat dan aman. Dengan merancang jalan yang lebih efisien dan sesuai dengan kemiringan yang optimal, pemilik kendaraan, terutama kendaraan berat, dapat mengurangi keausan dan kerusakan pada mesin serta komponen kendaraan lainnya.

4. Dukungan untuk Pengambilan Keputusan Pemerintah dan Pembangunan Infrastruktur

Pemerintah dan perencana infrastruktur dapat menggunakan data pemetaan kemiringan ini untuk mendukung pengambilan keputusan yang lebih baik terkait perencanaan dan pembangunan jalan. Ini termasuk memutuskan apakah perlu dilakukan perbaikan, ekspansi, atau renovasi jalan. Pemetaan kemiringan yang tepat memberikan informasi yang sangat berharga untuk perencanaan jangka panjang, termasuk memperkirakan dampak dari pertumbuhan populasi atau perubahan lalu lintas terhadap kondisi jalan.

1.6. Kendala dan Parameter Penting

A. Mengidentifikasi batasan teknis, ekonomi, dan regulasi

Dari segi teknis, akurasi GPS menjadi salah satu tantangan utama. GPS memiliki keterbatasan akurasi posisi dan elevasi, terutama di daerah perkotaan dengan bangunan tinggi atau di daerah dengan sinyal GPS yang terbatas. Hal ini dapat mempengaruhi akurasi data elevasi yang diperoleh. Selain itu, kualitas gyroscope yang digunakan juga harus memiliki sensitivitas dan stabilitas yang tinggi untuk mendeteksi perubahan kemiringan jalan dengan akurat. Gyroscope dengan kualitas rendah dapat menghasilkan data yang tidak akurat dan memerlukan algoritma kompensasi yang lebih kompleks. Integrasi data dari GPS dan gyroscope juga memerlukan algoritma sensor fusion yang tepat untuk menggabungkan data elevasi dan kemiringan. Kesalahan dalam integrasi sensor dapat menyebabkan kesalahan dalam perhitungan kemiringan jalan. Kondisi lingkungan seperti cuaca

buruk, getaran kendaraan, dan interferensi elektromagnetik juga dapat mempengaruhi performa dan akurasi sensor GPS dan gyroscope, sehingga diperlukan metode pengolahan sinyal yang robust untuk mengatasi gangguan ini.

B. Menentukan parameter kunci yang mempengaruhi kinerja produk

Dalam merancang sistem pemetaan kemiringan jalan menggunakan data elevasi dari GPS dan gyroscope, terdapat dua parameter kunci yang mempengaruhi kinerja produk. Pertama adalah titik koordinat dan yang kedua adalah sudut kemiringan.

1. Global Positioning System (GPS)

Sistem navigasi satelit yang menggunakan jaringan satelit di orbit Bumi untuk menentukan lokasi, kecepatan, dan waktu dengan akurasi tinggi di permukaan Bumi atau di sekitarnya.



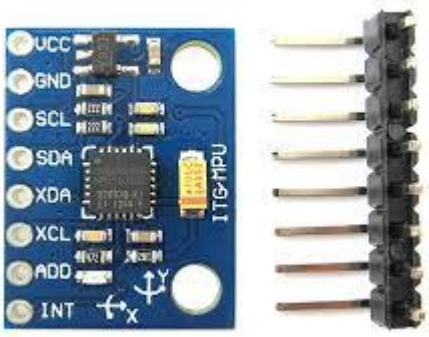
Gambar 1. 3 Modul GPS NEO 6

Table 1.1 Datasheet 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)
Parameter	Spesifikasi
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)

2. Gyroscope

Gyroscope adalah sebuah perangkat sensor yang berfungsi untuk mengukur atau mempertahankan orientasi berdasarkan prinsip kekekalan momentum sudut. Secara lebih spesifik, gyroscope dapat mendeteksi dan mengukur pergerakan rotasi atau perubahan orientasi suatu objek terhadap kerangka referensi inersia.



Gambar 1. 4 Sensor MPU 6050

Tabel 1. 1 Datasheet 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)

3. LCD TFT 2,4 inci

LCD TFT 2,4 inci adalah sebuah modul layar kecil berukuran diagonal 2,4 inci yang menggunakan teknologi Thin Film Transistor (TFT) sebagai matriks aktif untuk menampilkan gambar dengan kualitas dan kontras yang lebih baik dibandingkan LCD matriks pasif, serta dapat dikontrol oleh mikrokonroller seperti Arduino dan Raspberry Pi untuk berbagai aplikasi interaktif.



Gambar 1.5 LCD TFT 2,4 inci

Tabel 1.3 Datasheet LCD TFT 2,4 inci

Parameter	**Spesifikasi**
Jenis Layar	TFT LCD (Thin Film Transistor), warna penuh (65K / 262K warna)
Ukuran Layar	2,4 inci (diagonal)
Resolusi	240 × 320 piksel (QVGA)
Antarmuka Komunikasi	8-bit / 16-bit parallel, beberapa modul mendukung SPI
Controller IC	ILI9341 (umum, tergantung varian modul)
Tegangan Operasi	2.8V – 3.3V (kompatibel hingga 5V dengan regulator onboard)
Level Logika I/O	3.3V (kompatibel 5V via modul breakout)
Sudut Pandang	±160° (H/V)

Rasio Aspek	4:3
Luminansi (Kecerahan)	±250 cd/m²
Dimensi Modul	±42 mm × 60 mm × 2.5 mm
Area Aktif Layar	±36 mm × 48 mm
Touchscreen	Ada pada varian tertentu (resistif)
Temperatur Operasional	–20 °C hingga +70 °C
Konsumsi Daya	±50 mA (backlight aktif), <1 mA (tanpa backlight)

4. Push Button

Push button adalah sebuah komponen elektronik sederhana yang berfungsi sebagai saklar mekanis sementara (momentary switch). Komponen ini bekerja dengan cara menghubungkan (close) dan memutuskan (open) rangkaian listrik ketika ditekan dan dilepas. Push button umumnya digunakan sebagai input pada sistem mikrokontroler, seperti tombol reset, tombol navigasi menu, maupun kontrol manual pada perangkat elektronik.



Gambar 1.6 Push Button

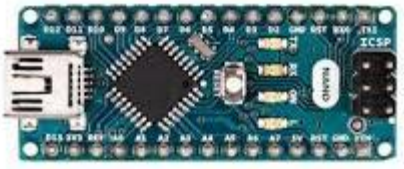
Tabel 1.4 Datasheet Push Button

Parameter	Spesifikasi
Jenis Komponen	Push Button / Tactile Switch
Mekanisme Kerja	Normally Open (NO), menutup saat ditekan dan terbuka kembali saat dilepas
Jumlah Pin	2 pin atau 4 pin (tergantung varian)
Dimensi Fisik	Umumnya 6 × 6 mm (standar), tersedia varian lain (3×6 mm, 12×12 mm, dll.)
Tegangan Maksimal	12V DC – 50V DC (tergantung tipe)
Arus Maksimal	50 mA – 100 mA
Tahanan Kontak	< 100 mΩ
Tahanan Isolasi	> 100 MΩ (pada 100V DC)
Gaya Tekan	100 gf – 260 gf (tergantung tipe)
Umur Mekanis	±100.000 hingga 1.000.000 kali tekan
Suhu Operasional	–20 °C hingga +70 °C
Material Kontak	Paduan logam konduktif (umumnya Ag atau Au plating)

5. Arduino Nano

Arduino Nano merupakan papan mikrokontroler berbasis ATmega328P yang digunakan secara luas dalam prototyping elektronik dan sistem embedded. Papan ini memiliki ukuran yang kecil sehingga cocok untuk aplikasi dengan keterbatasan ruang, namun tetap menyediakan sejumlah pin digital dan analog untuk input-output.

Dalam penelitian ini, Arduino Nano digunakan sebagai pengolah data dari sensor, termasuk sensor GPS dan MPU6050, sebelum data dikirim ke Arduino Uno untuk ditampilkan pada LCD TFT. Arduino Nano dipilih karena kemudahan pemrograman melalui Arduino IDE, kompatibilitas dengan berbagai modul sensor, serta ukuran yang kompak sehingga mudah diintegrasikan ke dalam perangkat eksperimen.



Gambar 1.7 Arduino Nano

Tabel 1.5 Datasheet Arduino Nano

Parameter	Spesifikasi
Mikrokontroler	ATmega328P
Tegangan Operasi	5V
Tegangan Input (disarankan)	7–12V
Tegangan Input (limit)	6–20V
Pin Digital I/O	14 pin (6 pin dapat digunakan sebagai PWM)
Pin Analog Input	8
Arus Pin I/O	Maksimal 40 mA per pin
Memori Flash	32 KB (0.5 KB digunakan oleh bootloader)
SRAM	2 KB
EEPROM	1 KB
Kecepatan Clock	16 MHz
Antarmuka Komunikasi	UART, I2C, SPI
Dimensi	45 × 18 mm
Berat	±7 gram

6. Arduino Uno

Arduino Uno adalah papan mikrokontroler berbasis ATmega328P yang banyak digunakan dalam penelitian dan proyek elektronika. Papan ini memiliki ukuran sedang dan menyediakan sejumlah pin digital dan analog yang cukup untuk aplikasi kontrol dan monitoring.

Dalam penelitian ini, Arduino Uno digunakan sebagai unit pemrosesan dan tampilan data, yang menerima informasi dari Arduino Nano melalui komunikasi serial dan menampilkan hasilnya pada LCD TFT 2,4 inci. Arduino Uno dipilih karena kemampuannya dalam mengolah data secara real-time, kompatibilitas dengan berbagai modul input-output, serta kemudahan pemrograman melalui Arduino IDE.



Gambar 1.8 Arduino Uno

Tabel 1.6 Datasheet Arduino Uno

Parameter	Spesifikasi
Mikrokontroler	ATmega328P
Tegangan Operasi	5V
Tegangan Input (disarankan)	7–12V
Tegangan Input (limit)	6–20V

Parameter	Spesifikasi
Pin Digital I/O	14 pin (6 pin dapat digunakan sebagai PWM)
Pin Analog Input	6
Arus Pin I/O	Maksimal 40 mA per pin
Memori Flash	32 KB (0.5 KB digunakan oleh bootloader)
SRAM	2 KB
EEPROM	1 KB
Kecepatan Clock	16 MHz
Antarmuka Komunikasi	UART, I2C, SPI
Dimensi	68.6 × 53.4 mm
Berat	±25 gram

DOC. EU-200

SPESIFIKASI SISTEM/PRODUK

2.1. Fungsi Produk

Fungsi produk dari penelitian "Rancangan Sistem Pemetaan Kemiringan Jalan Menggunakan Data Elevasi Dari GPS dan Gyroscope" adalah menyediakan sistem yang dapat mengukur tingkat kemiringan jalan secara akurat dengan memanfaatkan data elevasi yang diperoleh dari GPS dan data orientasi yang diperoleh dari gyroscope. Sistem ini memungkinkan pemantauan kemiringan jalan di berbagai titik, yang sangat berguna dalam perencanaan transportasi dan perawatan infrastruktur. Data elevasi dari GPS digunakan untuk mengambil titik koordinat posisi geografis jalan, sementara gyroscope berfungsi untuk mendeteksi perubahan sudut dan kemiringan dinamis, terutama saat kendaraan bergerak. Dengan mengintegrasikan kedua jenis data tersebut, sistem ini dapat menghasilkan peta kemiringan jalan yang akurat, yang dapat digunakan untuk meningkatkan keselamatan dan kenyamanan berkendara, serta membantu pihak berwenang dalam merencanakan perbaikan atau pemeliharaan jalan yang memiliki kemiringan curam.

2.2. Kinerja Produk

Pada komponen GPS, target pengukuran posisi adalah dengan akurasi 3 hingga 5meter untuk latitude dan longitude, serta akurasi elevasi sekitar 5 hingga 10meter, yang memungkinkan pengukuran lokasi secara akurat. GPS juga diharapkan dapat memperbarui data secara kontinu dengan frekuensi minimal 1

detik untuk memastikan informasi yang diperoleh selalu relevan. Meskipun dapat memberikan informasi posisi geografis dan elevasi, GPS memiliki batasan dalam hal akurasi. Posisi horizontal dapat diukur dengan akurasi sekitar 5 hingga 10meter dalam kondisi normal, tetapi akurasi pengukuran elevasi sering kali lebih rendah, sekitar 10meter atau lebih. GPS juga bergantung pada sinyal satelit, yang dapat terhalang oleh gedung tinggi, terowongan, atau area dengan banyak vegetasi, sehingga menurunkan kualitas sinyal dan akurasi pengukuran. Selain itu, kecepatan pembaruan data GPS terbatas pada interval beberapa detik, yang dapat menjadi kendala untuk aplikasi yang memerlukan data real-time yang lebih cepat.

Gyroscope memiliki target pengukuran sudut kemiringan dengan akurasi 0,1 hingga 0,5 derajat, serta kemampuan untuk mengukur sudut dalam rentang 360 derajat. Gyroscope juga harus mampu memberikan respons cepat dalam pengukuran dengan waktu respons kurang dari 100 milidetik untuk memastikan kemiringan dapat terdeteksi secara real-time, bahkan saat kendaraan bergerak. Gyroscope berfungsi untuk mengukur perubahan sudut kemiringan, namun memiliki batasan berupa akumulasi kesalahan atau *drift*. Seiring waktu, kesalahan kecil dalam pengukuran sudut dapat menumpuk, mengurangi akurasi pembacaan jika tidak dikalibrasi ulang. Gyroscope juga memiliki rentang pengukuran sudut tertentu, sehingga tidak dapat mengukur kemiringan ekstrem, seperti posisi vertikal yang lebih dari 90 derajat. Selain itu, sensor ini sensitif terhadap getaran eksternal, yang dapat mengganggu akurasi pengukuran, terutama dalam kondisi kendaraan yang bergerak atau jalan yang tidak rata. Gyroscope juga hanya mengukur perubahan sudut relatif dan tidak dapat memberikan informasi posisi absolut,

sehingga perlu digabungkan dengan GPS untuk memperoleh informasi posisi yang lengkap.

2.3. Karakteristik Produk

Produk yang dirancang dalam "Rancangan Sistem Pemetaan Kemiringan Jalan Menggunakan Data Elevasi Dari GPS dan Gyroscope" memiliki ukuran yang kompak dan efisien agar mudah dipasang pada kendaraan tanpa mengganggu kenyamanan pengemudi. Unit GPS dan gyroscope yang digunakan memiliki dimensi sekitar 10 cm x 10 cm x 3 cm dan berat sekitar 100 hingga 150 gram, terbuat dari bahan plastik ABS atau polikarbonat untuk memberikan ketahanan dan ringan. Unit pengolahan data, seperti mikrokontroler atau komputer embedded, memiliki dimensi sekitar 12 cm x 8 cm x 2 cm dengan berat 200 hingga 300 gram, dilindungi oleh casing plastik atau logam agar komponen internal tetap aman dari guncangan atau getaran. Jika sistem dilengkapi dengan tampilan visual, layar memiliki ukuran sekitar 15 cm x 10 cm dan berat sekitar 200 gram, terbuat dari kaca atau plastik lapisan anti-gores untuk ketahanan yang optimal. Sistem daya, berupa baterai lithium-ion, memiliki ukuran sekitar 8 cm x 5 cm x 2 cm dengan berat sekitar 150gram dan dilindungi oleh casing plastik atau logam. Secara keseluruhan, dimensi produk ini dapat mencapai sekitar 20 cm x 15 cm x 5 cm dengan berat total antara 500gram hingga 800 gram, bergantung pada konfigurasi komponen yang digunakan. Bahan utama yang digunakan untuk produk ini adalah plastik ABS, polikarbonat, dan logam ringan seperti aluminium, yang memberikan kombinasi kekuatan, ketahanan, dan ringan, memastikan sistem dapat dipasang dan

digunakan dengan mudah pada kendaraan untuk aplikasi pemetaan kemiringan jalan.

2.4. Rencana dan Jadwal (timeline) Pelaksanaan Proyek

Tabel 2. 1 Rencana dan Jadwal (timeline) Pelaksanaan Proyek

No	Uraian Kegiatan	Bulan			
		1	2	3	4
1	Studi Literatur				
2	Penyusunan EU-100				
3	Penyusunan EU-200				
4	Perancangan Sistem Perangkat Keras				
5	Perancangan Sistem Perangkat Lunak				
6	Desain Sistem Secara Keseluruhan				
7	Desain Konstruksi/Mekanik				

2.5. Rencana Anggaran Belanja

Tabel 2. 2 Rencana Anggaran Biaya

No	**Jenis Pengeluaran**	**Volume**	**Harga Satuan (Rp)**	**Total (Rp)**
1	Modul GPS	2 buah	48.300	96.600
2	Gyroscope	2 buah	32.650	65.300
3	Arduino Nano	1 buah	39.900	39.900
4	Arduino Uno	1 buah	100.000	100.000

No	Jenis Pengeluaran	Volume	Harga Satuan (Rp)	Total (Rp)
6	Display TFT 2,4”	1 buah	89.250	89.250
7	Kabel Jumper	40 buah	400	10.000
9	Box	1 buah	16.335	16.335
10	Push Button	2 buah	7.000	14.000
TOTAL				430.785

2.6. Rencana Pengujian

Pengujian pertama akan dilakukan untuk menguji akurasi data elevasi yang diperoleh dari GPS dengan membandingkannya dengan data elevasi yang sudah terverifikasi dari peta referensi. Selanjutnya, sistem akan diuji untuk mengukur akurasi data kemiringan yang diperoleh dari gyroscope, dengan memverifikasi sudut kemiringan yang terdeteksi oleh gyroscope terhadap metode pengukuran lain yang lebih konvensional. Setelah itu, pengujian kombinasi data GPS dan gyroscope akan dilakukan untuk menghitung kemiringan jalan dan memastikan bahwa hasil yang diperoleh dari gabungan kedua sumber data ini akurat dan konsisten. Pengujian visualisasi peta kemiringan akan mengevaluasi kemampuan sistem dalam menampilkan peta atau grafik yang mudah dipahami dan dapat digunakan oleh pengguna. Sistem juga akan diuji untuk menilai kemampuannya dalam pemantauan dan analisis real-time, dengan menguji apakah sistem dapat memberikan data kemiringan jalan yang akurat dan langsung, terutama dalam

aplikasi navigasi atau peringatan dini. Selain itu, pengujian ketahanan dan stabilitas sistem akan dilakukan untuk memastikan bahwa perangkat keras seperti GPS dan gyroscope dapat berfungsi dengan baik dalam berbagai kondisi lingkungan dan medan jalan. Terakhir, pengujian perbandingan dengan sistem pemetaan kemiringan jalan lainnya dapat dilakukan untuk menilai apakah sistem yang dikembangkan lebih efisien dan akurat.

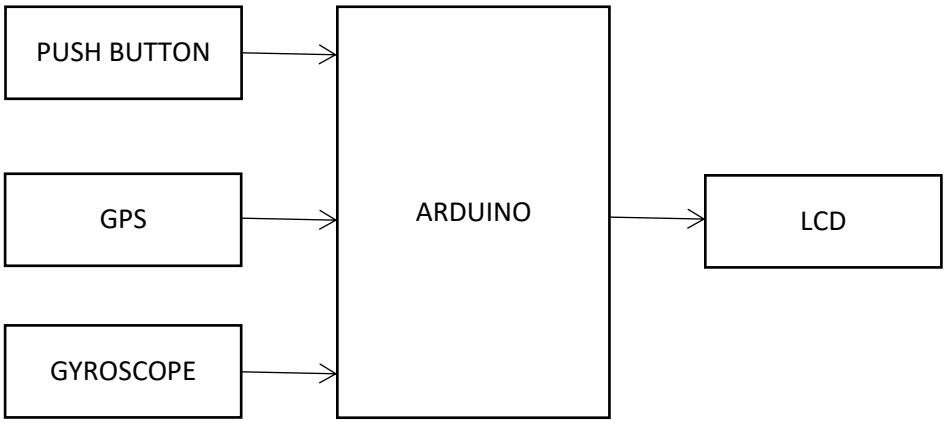
DOC. EU-300

PERANCANGAN SISTEM/PRODUK

3.1. Perancangan Sistem Perangkat Keras

A. Diagram Blok

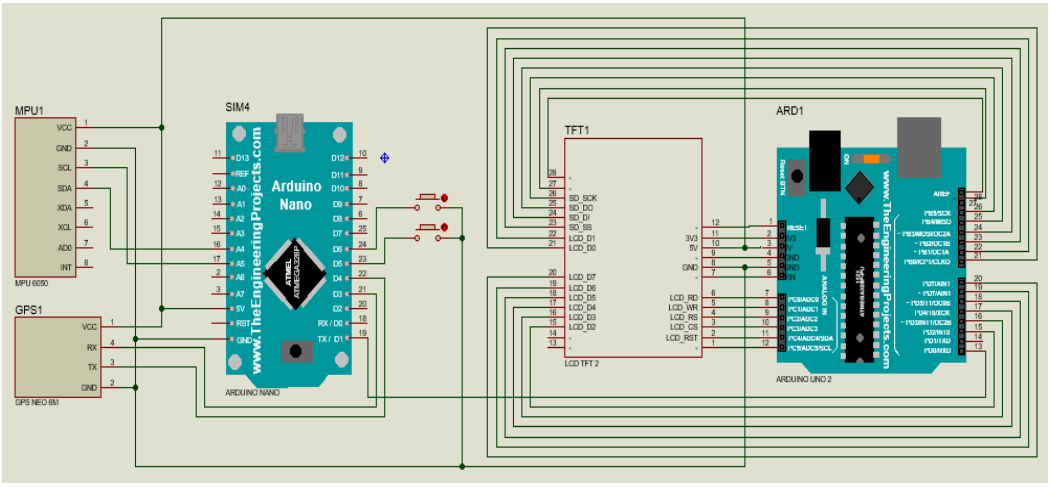
Alur kerja dari sistem pemetaan kemiringan jalan menggunakan Data elevasi dari GPS dan gyroscope sesuai Gambar 3.1 adalah pertama push button ditekan untuk mrmulai pengambilan data koordinat awal GPS dan gyroscope akan memulai mengambil data sudut kemiringan setiap 1 detik. Tekan kembali push button untuk mengambil koordinat akhir GPS dan berhenti mengambil data sudut kemiringan. Data yang didapatkan oleh GPS dan gyroscope akan diproses oleh Arduino dan hasilnya akan ditampilkan dilayar LCD TFT.



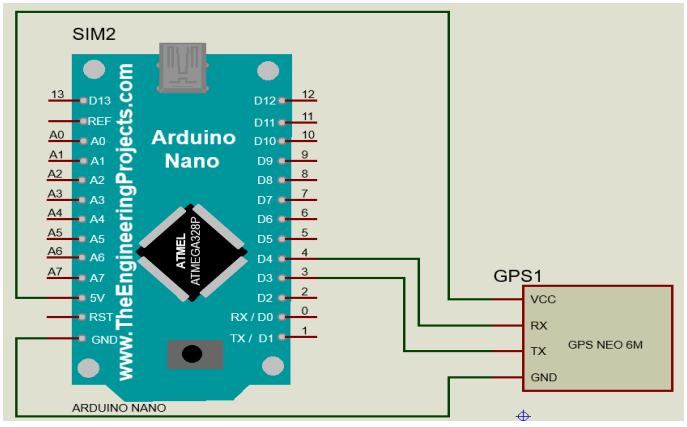
Gambar 3. 1 Diagram Blok

B. Desain Schematic Sistem

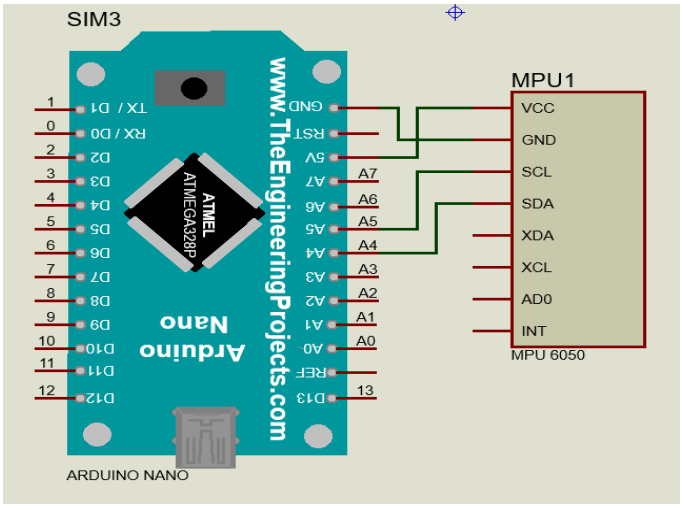
Desain Schematic sistem secara keseluruhan dapat dilihat pada gambar 3.2, desain skematik GPS dapat dilihat pada gambar 3.3, desain skematik gyroscope dapat dilihat pada gambar 3.4, desain skematik LCD dapat dilihat pada gambar 3.5, dan desain skematik Push Button dapat dilihat pada gambar 3.6.



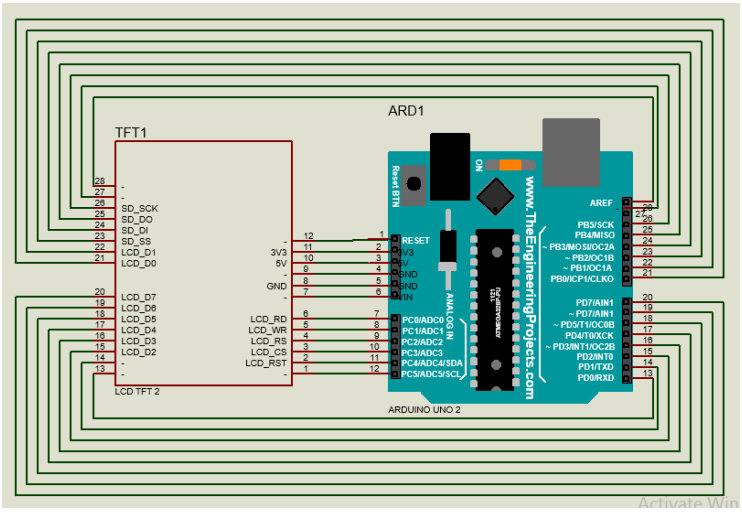
Gambar 3. 2 Desain Schematic Sistem



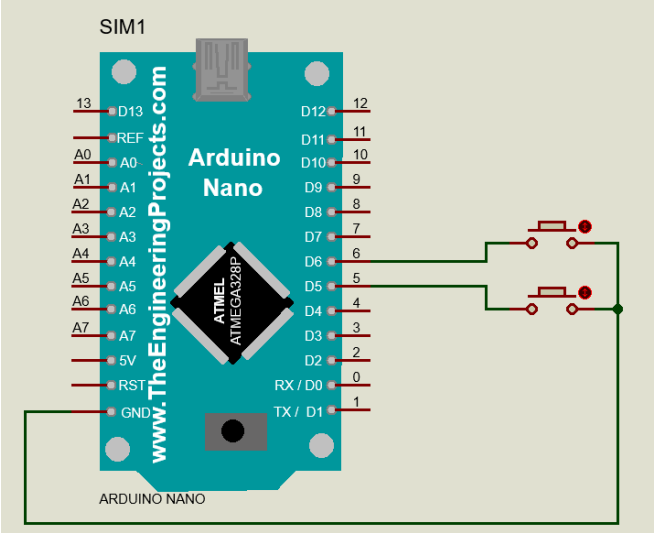
Gambar 3. 3 Desain Schematic GPS



Gambar 3. 4 Desain Schematic Gyroscope



Gambar 3. 5 Desain Schematic LCD



Gambar 3. 6 Desain Schematic Push Button

3.2. Perancangan Sistem Perangkat Lunak

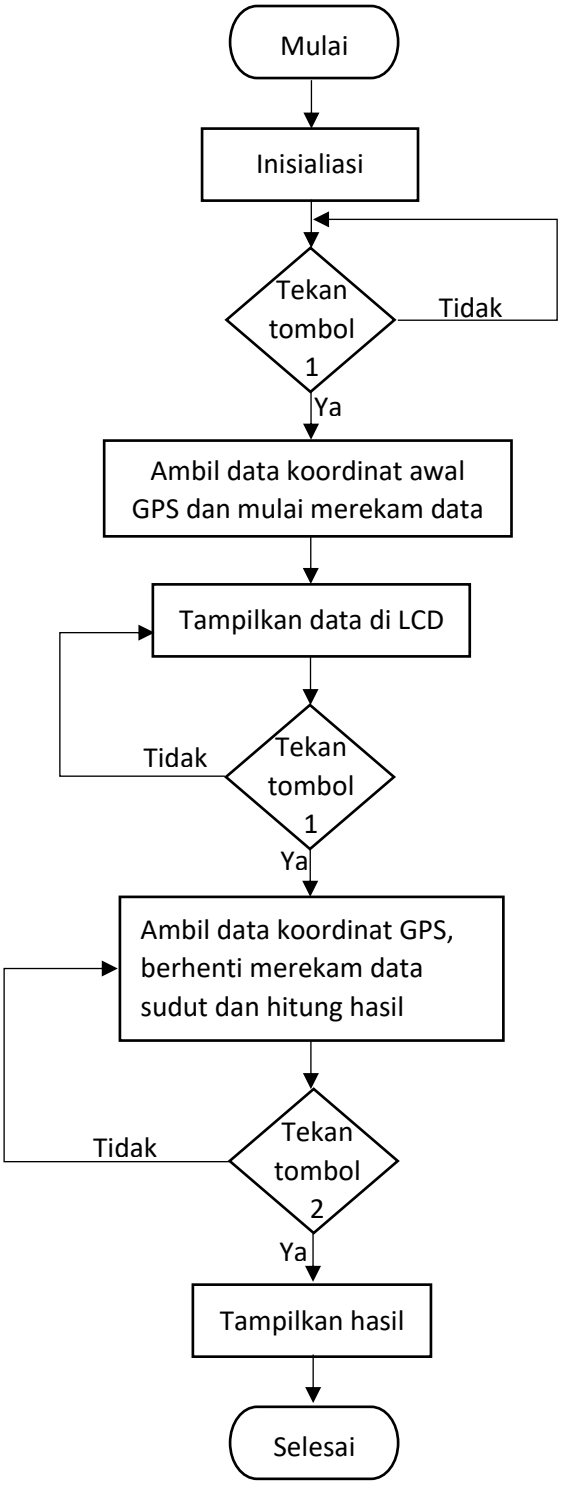
A. Platform:

Pemrograman perangkat lunak akan dilakukan menggunakan platform Arduino IDE karena kemudahan dalam penggunaan perangkat lunak.

B. Bahasa Pemrograman

Bahasa C/C++: Bahasa ini digunakan untuk pemrograman Arduino IDE karena fleksibilitas dan kontrol yang diberikan dalam pengolahan data dan komunikasi antar perangkat.

C. Flowchart



Gambar 3. 7 Flowchart

D. Algoritma Sistem

1. Mulai; Proses sistem dimulai. Mikrokontroler aktif, sistem siap bekerja.
2. Inisialisasi; Sistem melakukan setup awal sensor GPS, sensor gyroscope (MPU6050), LCD, serta membaca status tombol.
3. Tekan Tombol 1 (Pertama Kalinya); Sistem menunggu sampai tombol biru (tombol 1) ditekan oleh pengguna. Jika belum ditekan, sistem akan tetap menunggu. Jika ditekan, lanjut ke proses berikutnya.
4. Ambil Data; GPS mencatat koordinat awal. MPU6050 mulai membaca sudut kemiringan setiap 1 detik.
5. Tampilkan Data di LCD; Data yang sedang dibaca (koordinat dan sudut) ditampilkan secara real-time di LCD TFT
6. Tekan Tombol 1 (Kedua Kalinya); Sistem terus menampilkan data sambil menunggu tombol 1 ditekan kembali. Jika belum ditekan, proses berulang (loop) menampilkan data. Jika ditekan lagi, lanjut ke proses akhir pengambilan data.
7. Ambil Data Kedua GPS, Berhenti Ambil Data Sudut, dan Hitung Hasil; GPS mencatat koordinat akhir. MPU6050 berhenti merekam data. Sistem menghitung Jarak berdasarkan dua titik koordinat (pakai rumus Haversine) dan rata-rata sudut kemiringan berdasarkan data sudut yang terekam.
8. Tekan Tombol 2; Menunggu pengguna menekan tombol kuning untuk melihat hasil akhir.
9. Tampilkan Hasil; LCD menampilkan Jarak antara dua titik dalam meter dan sudut kemiringan rata-rata.

10. Selesai; Proses pengukuran selesai. Sistem siap kembali ke awal bila diperlukan.

3.3. Desain Sistem Secara Keseluruhan

Berikut adalah uraian fungsi dan interaksi antar subsistem dalam produk pemetaan kemiringan jalan menggunakan GPS dan gyroscope:

A. Subsistem Input

1. Fungsi: Mengukur parameter kunci kemiringan jalan
2. Komponen:
    a. GPS: Mengukur posisi geografis latitude dan longitude jalan secara akurat.
    b. Gyroscope: Mengukur perubahan sudut kemiringan jalan secara dinamis dengan akurasi tinggi setiap 1 detik.
3. Interaksi: Data yang diperoleh dari setiap sensor akan dikirin ke arduino UNO untuk diproses.

B. Subsistem Processor

1. Fungsi: Sebagai pusat pemrosesan data dari sensor dan pengambilan keputusan
2. Interaksi:
    a. Menerima input data dari GPS dan gyroscope.
    b. Mengolah data untuk menghasilkan informasi yang dapat ditampilkan pada layar atau disimpan dalam basis data.
    c. Mengontrol komunikasi antar subsistem untuk memastikan sinkronisasi data.

C. Subsistem Output

1. Fungsi: Menampilkan data hasil pemrosesan secara langsung kepada pengguna.
2. Komponen:
    a. LCD TFT 2,4”: Menampilkan hasil dari sudut kemiringan jalan yang telah diukur
3. Interaksi: Menerima output dari unit pengolahan data untuk ditampilkan dalam bentuk angka.

D. Sistem Daya

1. Fungsi: Menyediakan sumber energi untuk semua komponen elektronik
2. Komponen:
    a. Power Bank
3. Interaksi:
    a. Memberikan daya langsung ke semua subsistem melalui distribusi yang efisien.
    b. Baterai dapat diisi ulang dan memiliki perlindungan terhadap overcharging untuk menjaga keandalan sistem.

E. Interaksi Antar Subsistem Secara Keseluruhan

1. Data Input: GPS dan gyroscope masing-masing mengumpulkan data posisi geografis dan kemiringan jalan.
2. Pemrosesan Data: Arduino UNO menerima input dari GPS dan gyroscope, memprosesnya untuk menghasilkan peta kemiringan jalan.

3. Output: Hasil pemrosesan ditampilkan melalui LCD TFT dalam bentuk angka.
4. Daya Operasional: Sistem daya memastikan kelancaran operasi semua subsistem, bahkan dalam kondisi lingkungan yang tidak stabil.

DOC. EU-400

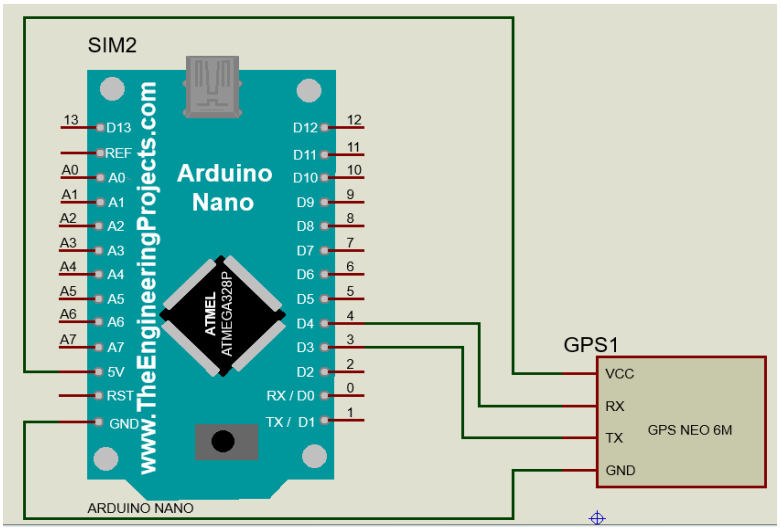
IMPLEMENTASI

4.1. Hasil Perancangan Sistem Perangkat Keras

Perancangan sistem perangkat keras ini menggunakan GPS Neo 6m, MPU 6050, Arduino Nano, Arduino Uno, LCD TFT 2,4 inci dan dua push button serta kabel jumper sebagai penghubung antar komponen.

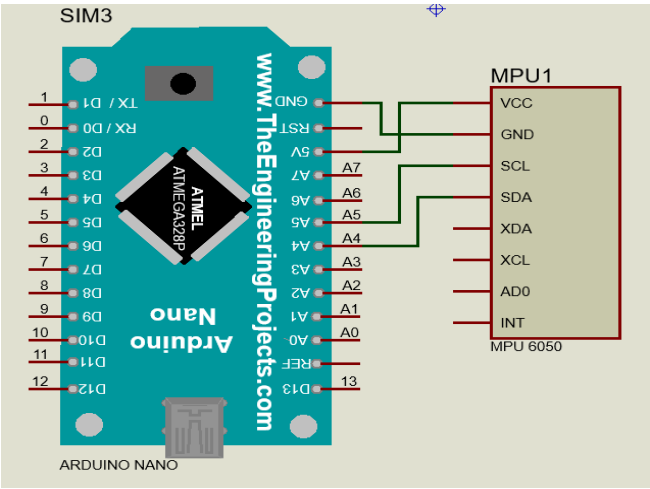
GPS berfungsi untuk mengambil data *latitude* dan *longitude*, MPU 6050 berfungsi untuk mengambil data sudut kemiringan (*pitch*), arduino nano dan uno berfungsi sebagai pemproses data, LCD untuk menampilkan data serta push button 1 berfungsi sebagai memulai dan mengakhiri pengambilan data dan menghitung data yang telah diambil dan push button 2 berfungsi untuk menampilkan data yang telah di proses.

Modul GPS Neo 6M dihubungkan ke Arduino Nano dengan konfigurasi sebagai berikut: pin VCC GPS dihubungkan ke pin 5V Arduino Nano, pin GND ke pin GND, pin TX GPS dihubungkan ke pin digital D4 Arduino Nano, dan pin RX GPS dihubungkan ke pin digital D3. Desain Skematik GPS dapat dilihat pada gambar 4.1.



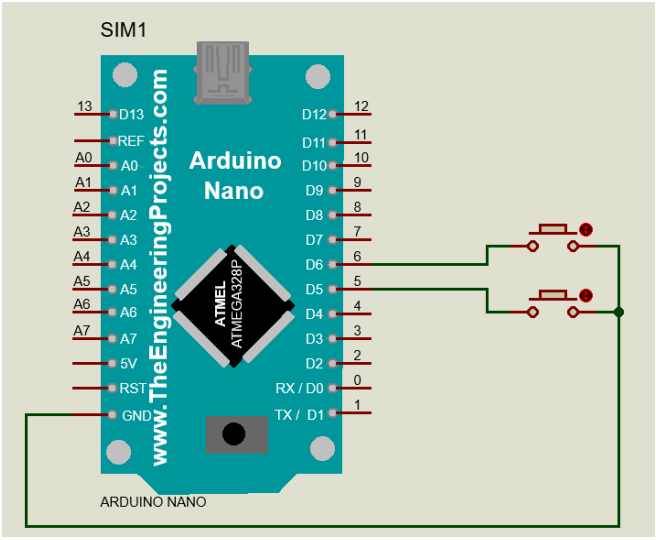
Gambar 4. 1 Desain Schematic GPS

Sensor MPU6050 dihubungkan ke Arduino Nano dengan konfigurasi sebagai berikut: pin VCC sensor dihubungkan ke pin 5V Arduino Nano, pin GND ke pin GND, pin SDA dihubungkan ke pin A4 Arduino Nano, dan pin SCL dihubungkan ke pin A5. Desain skematik MPU6050 dapat dilihat pada gambar 4.2



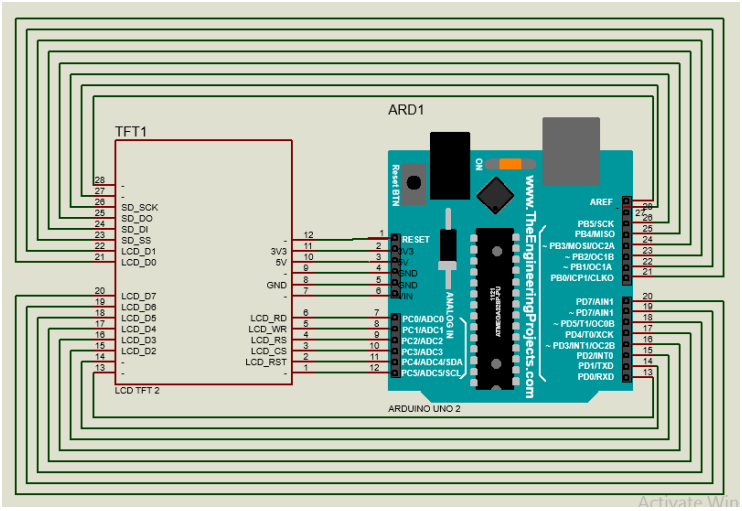
Gambar 4. 2 Desain Schematic MPU6050

Pin push button 1 dihubungkan ke D5 dan GND Arduino Nano dan pin push buton 2 dihubungkan ke D6 dan GND Arduino Nano. Desain skematik push button dapat dilihat pada gambar 4.3



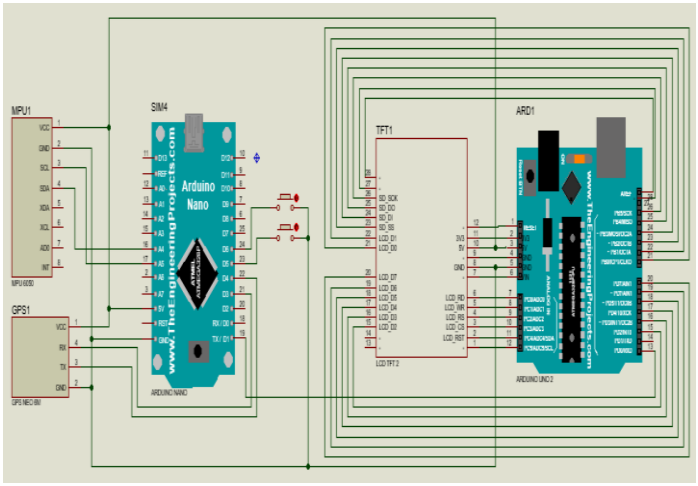
Gambar 4. 3 Desain Schematic Push Button

LCD TFT 2.4 inci dihubungkan ke Arduino Uno. LCD TFT 2,4 inci yang digunakan memiliki 28 pin dan dihubungkan langsung ke Arduino Uno dengan sistem pemasangan shield, sehingga seluruh pin digital (D0–D13) dan analog (A0–A5) pada Arduino Uno digunakan. Desain skematik LCD dapat dilihat pada gambar 4.4.

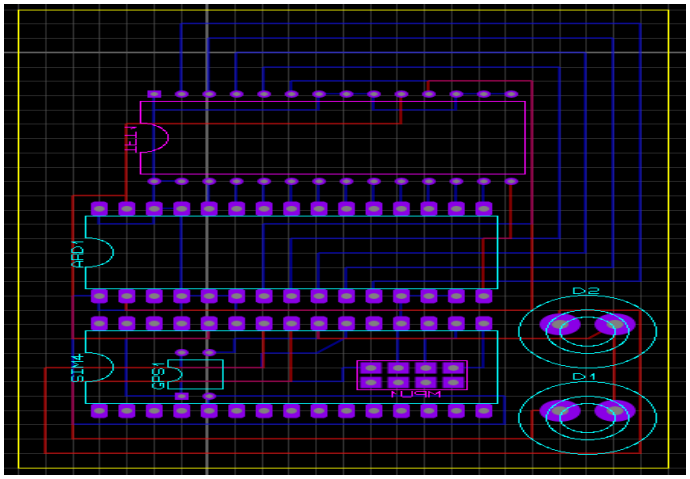


Gambar 4. 4 Desain Schematic LCD

Dalam sistem ini, Arduino Uno digunakan sebagai sumber daya (*power supply*) bagi Arduino Nano dan sekaligus sebagai penerima data melalui komunikasi serial. Konfigurasi yang digunakan adalah sebagai berikut: pin 5V pada Arduino Uno dihubungkan ke pin 5V Arduino Nano untuk mensuplai tegangan, pin GND Uno ke GND Nano, serta pin RX pada Arduino Uno dihubungkan ke pin TX pada Arduino Nano untuk menerima data serial yang dikirimkan. Desain skematik keseluruhan sistem dapat dilihat pada gambar 4.5 dan desain PCB board sistem dapat dilihat pada gambar 4.6.



Gambar 4. 5 Desain Schematic Keseluruhan Sistem



Gambar 4. 6 Desain PCB Board Sistem

4.2. Hasil Perancangan Sistem Perangkat Lunak

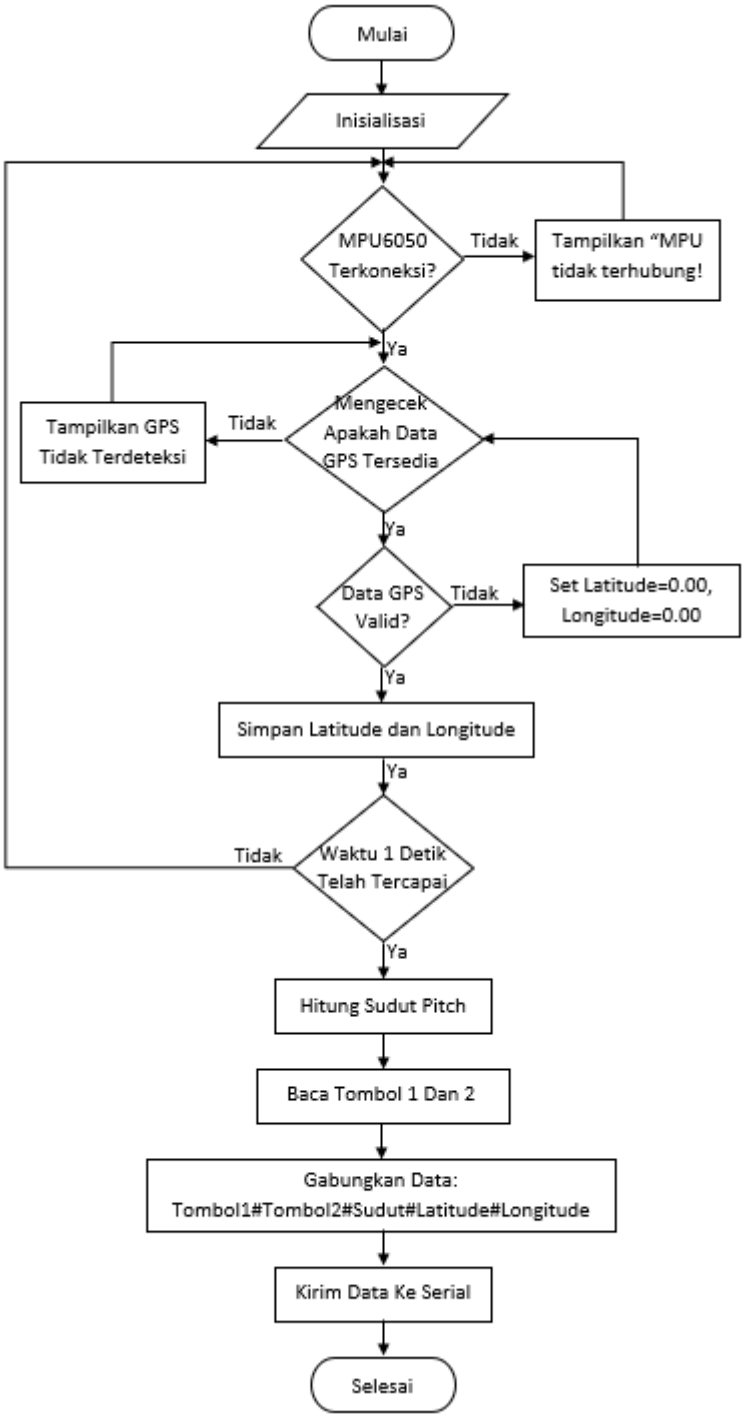
Setelah seluruh perangkat keras berhasil dirancang dan dirakit, tahap selanjutnya adalah perancangan perangkat lunak sebagai pengendali utama sistem. Perangkat lunak dikembangkan menggunakan Arduino IDE dengan bahasa pemrograman C/C++, yang mencakup pembacaan data dari sensor GPS Neo-6M dan MPU6050, pengolahan data, komunikasi serial, serta penampilan data pada layar LCD TFT 2,4 inci. Push button 1 digunakan untuk memulai dan menghentikan proses pengambilan data sekaligus menghitung jarak dan sudut kemiringan, sedangkan push button 2 digunakan untuk menampilkan hasil pengukuran. Sistem ini menggunakan dua mikrokontroler, yaitu Arduino Nano dan Arduino Uno, yang masing-masing memiliki fungsi berbeda. Arduino Nano bertugas mengakuisisi data dari sensor dan tombol, lalu mengirimkannya ke Arduino Uno melalui komunikasi serial. Arduino Uno kemudian memproses data dan menampilkannya ke layar. Pemisahan fungsi ini bertujuan untuk mengatasi

keterbatasan jumlah pin, mengurangi beban kerja mikrokontroler, dan meningkatkan kestabilan sistem saat dijalankan secara real-time.

Penggunaan dua mikrokontroler, yaitu Arduino Nano dan Arduino Uno, dipilih sebagai alternatif dari satu Arduino Mega. Pemisahan ini bertujuan agar proses akuisisi data sensor dan pengolahan tampilan tidak saling membebani, sehingga sistem lebih stabil saat bekerja secara real-time. Selain itu, pendekatan ini menjadikan sistem lebih modular, memudahkan proses pengembangan maupun perbaikan, serta memastikan performa sesuai kebutuhan penelitian.

a. Arduino Nano

Arduino Nano berperan sebagai unit akuisisi data. Mikrokontroler ini membaca input dari dua push button, sensor GPS, dan sensor MPU6050. Data yang diperoleh kemudian dikirimkan ke Arduino Uno melalui komunikasi serial. Alur kerja Program Arduino Nano dapat dilihat pada gambar 4.7.



Gambar 4. 7 Flowchart Arduino Nano

Penjelasan Flowchart Arduino Nano

1. Mulai: Proses dimulai.
2. Inisialisasi: Sistem melakukan inisialisasi (setup awal) terhadap GPS, MPU6050, dan 2 tombol.
3. Periksa koneksi MPU6050: Jika MPU6050 tidak terkoneksi, tampilkan pesan “MPU tidak terhubung!”. Jika MPU6050 terkoneksi, lanjut ke langkah berikutnya.
4. Pengecekan Ketersediaan Data GPS: Jika data GPS tidak tersedia, tampilkan pesan "GPS Tidak Terdeteksi" dan kembali ke proses inisialisasi. Jika data GPS tersedia, lanjut ke validasi data.
5. Validasi data GPS: Jika data GPS tidak valid, set koordinat *latitude* dan *longitude* menjadi 0.00. Jika data GPS valid, simpan data *latitude* dan *longitude*.
6. Sistem menunggu hingga 1 detik berlaku sejak pembacaan sebelumnya, jika belum kembali keloop awal. Jika sudah lanjutkan keperhitungan pitch.
7. Perhitungan *pitch*: Membaca data akselerasi dan gyro dari MPU6050. Hitung sudut *pitch* dari data akselerasi (sumbu Y dan Z).
8. Membaca status dua tombol push button.
9. Gabungkan data: Menggabungkan data yang diperoleh dari tombol, sudut *pitch*, *latitude* dan *longitude*.
10. Mengirim data gabungan ke serial
11. Selesai: Proses 1 siklus selesai.

Berikut kode program dari Arduino Nano yang telah dibua

```
   //-------------------------------------------------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

#include <Wire.h> → Mengimpor library I2C untuk komunikasi
dengan sensor MPU6050.

#include <MPU6050.h> → Mengimpor library khusus untuk sensor
MPU6050.

MPU6050 mpu; → Membuat objek mpu agar bisa mengakses fungsi
dari library MPU6050.

#include <TinyGPSPlus.h> → Mengimpor library untuk mengolah
data dari modul GPS.

#include <SoftwareSerial.h> → Mengimpor library untuk membuat
port serial tambahan (selain Serial utama).

   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

RXPin = 4, TXPin = 3 → Menentukan pin 4 sebagai RX (menerima
data) dan pin 3 sebagai TX (mengirim data) untuk GPS.

GPSBaud = 9600 → Baud rate komunikasi dengan GPS (umumnya
9600 bps).

TinyGPSPlus gps; → Membuat objek gps agar bisa memproses data
NMEA dari GPS.

SoftwareSerial ss(RXPin, TXPin); → Membuat port serial
software bernama ss di pin 4 dan 3.


   // 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)
```

pitch → Menyimpan sudut pitch (kemiringan dari MPU6050).

lat, lng → Menyimpan koordinat GPS (latitude & longitude).

data → String untuk menampung gabungan semua data sebelum dikirim ke Serial.

pinSta1, pinSta2 → Pin untuk tombol input (push button).

lastSudutMillis → Menyimpan waktu terakhir sensor dibaca (pakai fungsi millis()).

intervalSudut → Interval pembacaan sensor pitch = 1000 ms (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
```

Serial.begin(9600); → Mengaktifkan komunikasi serial dengan PC.

pinMode(..., INPUT_PULLUP); → Mengatur pin tombol sebagai input dengan resistor internal 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
    }
```

Wire.begin(); → Mengaktifkan I2C.

mpu.initialize(); → Inisialisasi sensor MPU6050.

if (!mpu.testConnection()) → Mengecek apakah sensor terhubung. Jika tidak → tampilkan error dan hentikan program.

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

```
   }
```

ss.begin(GPSBaud); → Menyalakan komunikasi serial ke modul 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
   }
```

getAcceleration() → Baca data akselerometer (sumbu X, Y, Z).

getRotation() → Baca data gyroscope (sumbu X, Y, Z).

atan2(ay, az) → Menghitung sudut kemiringan berdasarkan sumbu Y & Z.

* (180.0 / PI) → Konversi radian → 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
         }
       }
```

ss.available() → Mengecek apakah ada data dari GPS.

gps.encode(ss.read()) → Membaca data mentah (NMEA) dari GPS lalu menguraikannya.

gps.location.isValid() → Mengecek apakah lokasi valid.

lat, lng → Diisi nilai koordinat, jika tidak valid → diisi 0.00.

```
   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
      }
   }

Mengecek apakah GPS benar-benar aktif. Jika tidak ada cukup
data setelah 5 detik → dianggap tidak ada GPS.

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

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

millis() → waktu sejak Arduino dinyalakan.


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

Mengecek apakah sudah waktunya membaca sudut lagi (tiap 1
detik).

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

digitalRead() → Membaca status tombol (HIGH = tidak ditekan,
LOW = ditekan karena pakai pull-up).


       // 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
   }
String(...) → Mengubah data ke string.

Format data: tombol1#tombol2#pitch#latitude#longitude.

Serial.println(data) → Mengirim data ke komputer/Arduino lain
untuk ditampilkan/diolah.
```

Program menggunakan bahasa pemrograman Arduino dengan bantuan beberapa library. Library Wire.h dan MPU6050.h digunakan untuk membaca

data akselerometer serta gyroscope dari sensor MPU6050, sedangkan TinyGPSPlus.h dan SoftwareSerial.h digunakan untuk mengolah data GPS dan menyediakan port komunikasi serial tambahan.

Variabel global yang didefinisikan berfungsi untuk menyimpan sudut kemiringan (pitch), koordinat GPS (latitude dan longitude), serta status tombol input. Selain itu, mekanisme waktu menggunakan fungsi millis() diterapkan agar pembacaan sudut dilakukan secara periodik setiap satu detik.

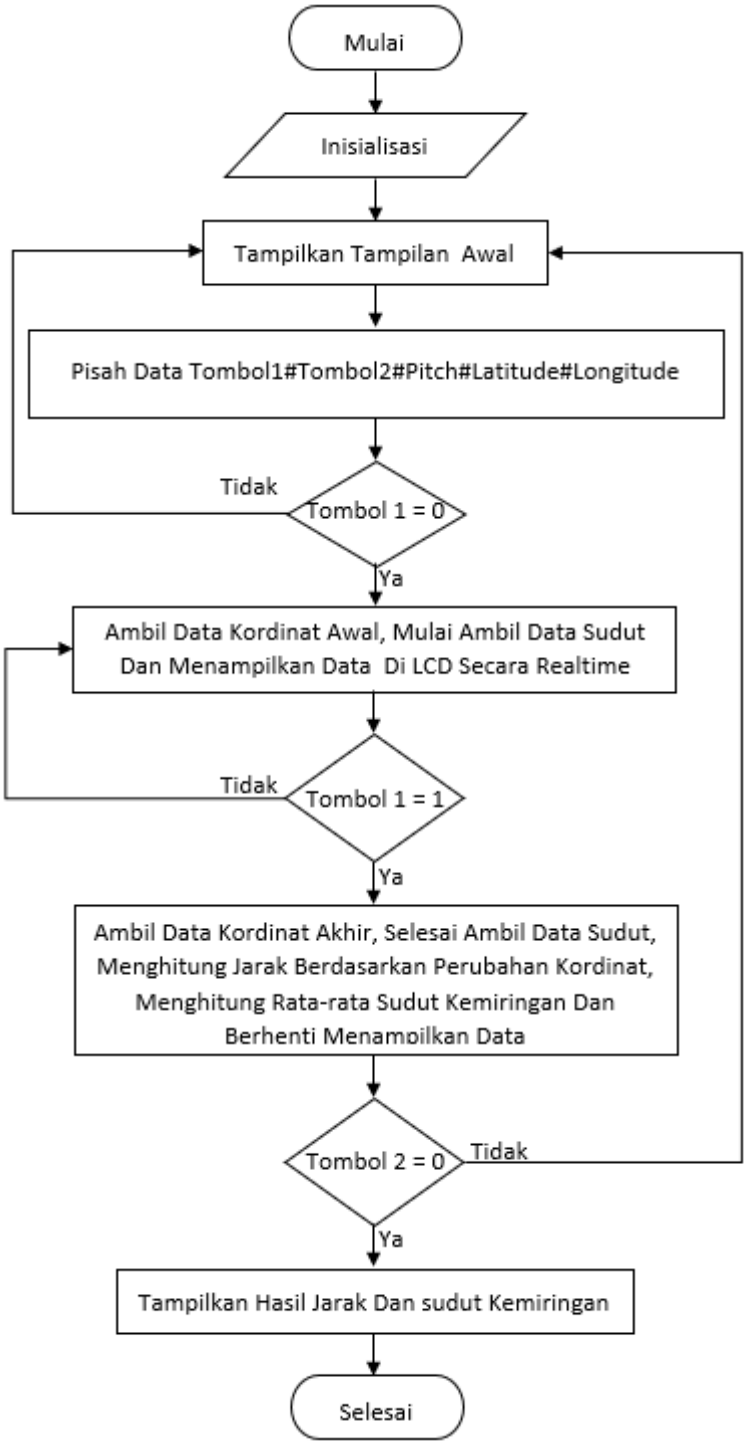
Fungsi setup() bertugas melakukan inisialisasi perangkat, yaitu mengaktifkan komunikasi serial, mengatur pin tombol, menginisialisasi sensor MPU6050, serta memulai komunikasi dengan modul GPS. Jika sensor tidak terdeteksi, program akan menampilkan pesan kesalahan dan berhenti.

Fungsi bacaSudut() menghitung sudut pitch berdasarkan data akselerometer dari sensor MPU6050 menggunakan persamaan trigonometri atan2. Sementara itu, fungsi bacaGPS() digunakan untuk memperoleh data latitude dan longitude dari GPS, sekaligus mengecek validitas data lokasi yang diterima.

Fungsi loop() menjalankan proses utama, yaitu membaca data GPS, menghitung sudut pitch setiap satu detik, membaca status tombol input, serta menggabungkan seluruh informasi ke dalam format string dengan pemisah tanda pagar (#). Data yang terbentuk kemudian dikirimkan melalui komunikasi serial agar dapat diproses lebih lanjut.

a. Arduino Uno

Arduino Uno berfungsi sebagai unit pengolah dan penampil data. Mikrokontroler ini menerima data dari Arduino Nano, ketika tombol 1 ditekan maka GPS akan mengambil data koordinat pertama dan MPU6050 akan mengambil data sudut kemiringan jalan. Apabila tombol 1 ditekan untuk kedua kalinya maka akan mengambil data koordinat akhir dan MPU6050 berhenti mengambil data sudut dan kemudian data yang diperoleh akan diproses untuk mendapatkan jarak dari perubahan titik kordinat dan sudut kemiringan berdasarkan rata-rata sudut dan push button 2 berfungsi untuk menampilkan data yang telah diproses kemudian ditampilkan pada LCD TFT 2,4 inci. Alur kerja Program Arduino Nano dapat dilihat pada gambar 4.8.



Gambar 4. 8 Flowchart Arduino Uno

Penjelasan Flowchart Arduino Uno

1. Mulai: Proses dimulai.
2. Inisialisasi: Sistem melakukan inisialisasi (setup awal) LCD
3. Menampilkan tampilan awal pada LCD.
4. Menerima dan memisahkan data yang dikirim dari Arduino Uno dan menyimpannya dalam variabel masing-masing (data tombol, sudut, dan koordinat).
5. Cek Tombol 1: Sistem menunggu tombol 1 ditekan. Jika tidak ditekan, sistem tetap menunggu. Jika ditekan, lanjut ke langkah berikutnya.
6. Ambil data koordinat GPS dan data sudut : GPS mengambil data koordinat awal dan mulai merekam data sudut. Data yang sedang direkam ditampilkan di layar LCD secara *real-time*.
7. Cek Tombol 1 (lagi): Sistem menunggu tombol 1 ditekan kembali untuk menghentikan proses pengambilan data. Jika tidak ditekan, proses pengambilan data sudut dan penampilan di LCD terus berjalan. Jika ditekan, ambil data koordinat akhir GPS, berhenti mengambil data sudut, dan menghitung jarak berdasarkan perubahan titik koordinat, menghitung rata-rata sudut kemiringan, dan berhenti menampilkan data di LCD.
8. Cek Tombol 2: Sistem menunggu tombol 2 ditekan. Jika tidak ditekan, sistem tetap menunggu. Jika ditekan, tampilkan hasil.
9. Tampilkan Hasil: Menampilkan hasil perhitungan di LCD berupa jarak dan sudut kemiringan
10. Selesai: Proses selesai.

b. Berikut kode program dari Arduino UNO yang telah dibuat

```
#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
Adafruit_GFX.h → Library grafis dasar untuk menggambar di layar
(teks, garis, dll).
MCUFRIEND_kbv.h → Library driver khusus TFT 2.4” tipe shield.
MCUFRIEND_kbv tft; → Membuat objek tft agar bisa dipakai
mengontrol layar.
FreeSans..., FreeSerif..., FreeDefaultFonts.h → Kumpulan font
tambahan dari Adafruit.

String dataSensor;
String sudut;
String lat;
String lng;
String jarak;
String kemiringan;
String sta;

Variabel String untuk menampung data yang diterima dari Arduino
pengirim (via Serial).
sudut, lat, lng → Data sensor MPU6050 dan GPS.
jarak, kemiringan → Hasil perhitungan.
sta → Status tombol.

// 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

Definisi warna dalam format RGB565 (16-bit).
Mempermudah pemanggilan warna (contoh tft.fillScreen(BLACK)

// 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)

prevLat, prevLng → Menyimpan posisi GPS sebelumnya.
firstPoint → Penanda titik GPS pertama.
averageSudut, sudutCount → Untuk menghitung rata-rata
      kemiringan.
prevSta → Menyimpan status tombol sebelumnya (0/1).
buttonPressed → Flag tombol (tidak dipakai di kode ini).

// 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
   }

Menghitung jarak antar 2 titik koordinat GPS dengan rumus
Haversine.
Input: lat1, lon1, lat2, lon2 (dalam derajat).
Output: float 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
   }

Serial.begin(9600) → Mulai komunikasi serial.
tft.readID() → Baca ID chip dari TFT. Jika tidak valid → pakai
default 0x9481.
tft.begin(ID) → Inisialisasi TFT.
tft.setRotation(1) → Atur orientasi horizontal.
tft.fillScreen(BLACK) → Bersihkan layar jadi hitam.
```

```
tampilanAwal() → Tampilkan template 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
   }

Menampilkan judul, label, garis, dan keterangan pada layar saat
pertama kali menyala.
tampil() → fungsi khusus cetak teks (dibuat di bawah).

// 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
     }
   }

Membaca data string dari serial.
Format data: status#sudut#lat#lng.
pisahData() → Memecah string jadi variabel.
```

```
Setelah dibaca → tampilan lama dihapus (clearDisplay()) lalu
update baru.

// 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);
   }

Menghapus nilai lama dengan cara menimpa teks menggunakan warna
hitam.

// 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
   }

Menampilkan latitude, longitude, sudut di layar secara real-
time.
Warna teks → kuning.

// 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);
    }

Menghitung jarak GPS antar titik dengan Haversine.
Menghitung rata-rata sudut selama tombol ditekan.
Menampilkan hasil pada layar dan serial monitor.


// 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
    }

sta == "1" → tombol ditekan → tampilkan data real-time +
kumpulkan sudut.
sta == "0" → tombol dilepas → hitung jarak & rata-rata
kemiringan.
prevSta → menyimpan status lama agar bisa deteksi perubahan.
delay(100) → menstabilkan update tampilan.

// 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
    }

Membuat fungsi cetak teks dengan parameter: posisi, ukuran,
font, warna, dan isi teks.
Mempermudah pemanggilan di fungsi lain.
```

Program ini menggunakan library Adafruit_GFX.h sebagai dasar pengolahan grafis pada layar, serta MCUFRIEND_kbv.h sebagai driver untuk modul LCD TFT 2,4 inci tipe shield. Beberapa font tambahan dari library Adafruit juga diimpor untuk mempercantik tampilan teks. Sebuah objek tft dibuat agar mikrokontroler dapat mengakses fungsi-fungsi dari layar TFT.

Variabel global bertipe String digunakan untuk menyimpan data yang diterima dari Arduino pengirim, yaitu sudut kemiringan, koordinat GPS (latitude dan longitude), jarak tempuh, rata-rata kemiringan, serta status tombol. Selain itu, terdapat variabel

numerik untuk menyimpan titik koordinat GPS sebelumnya, menghitung rata-rata sudut, serta mendeteksi status tombol yang ditekan atau dilepas.

Fungsi setup() berfungsi untuk menginisialisasi komunikasi serial, membaca ID driver TFT, mengatur orientasi layar, serta menampilkan tampilan awal berupa judul, label, dan kerangka layout. Fungsi tampilanAwal() memuat elemen-elemen visual utama seperti judul aplikasi, label untuk latitude, longitude, sudut terbaca, jarak, dan kemiringan.

Fungsi bacaData() membaca string data yang dikirim melalui komunikasi serial dengan format tertentu, lalu memisahkannya menjadi variabel-variabel terpisah menggunakan fungsi pisahData(). Fungsi clearDisplay() digunakan untuk menghapus nilai lama dengan menimpa teks berwarna hitam, sedangkan fungsi display() menampilkan data terbaru (latitude, longitude, dan sudut) ke layar secara real-time.

Untuk perhitungan jarak, program menggunakan fungsi haversine() yang mengimplementasikan rumus Haversine guna menghitung jarak antara dua titik koordinat GPS. Fungsi hitung() kemudian menampilkan hasil perhitungan jarak serta rata-rata sudut kemiringan pada layar, sekaligus mengirimkan nilai tersebut ke serial monitor.

Fungsi utama loop() mengatur alur logika berdasarkan status tombol. Ketika tombol ditekan (sta == "1"), data real-time ditampilkan dan nilai sudut terus dikumpulkan. Ketika tombol dilepas (sta == "0"), program memanggil fungsi hitung() untuk menampilkan hasil akhir berupa jarak tempuh dan rata-rata kemiringan jalan.

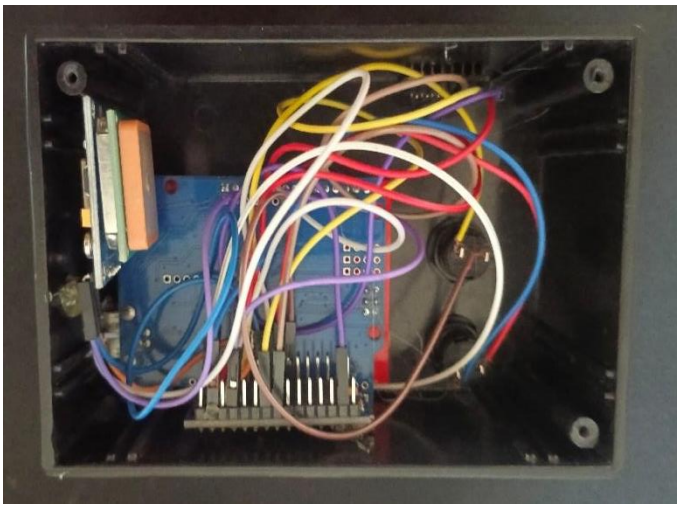
Fungsi tampil() digunakan untuk mempermudah pencetakan teks ke layar dengan parameter posisi, ukuran font, warna, dan isi teks.

4.3. Konstruksi/Mekanik

Produk ini merupakan sebuah alat ukur kemiringan jalan berbasis mikrokontroler yang dirancang untuk membantu mengukur kemiringan jalan secara *real-time*. Alat ini dilengkapi dengan sensor MPU6050 untuk membaca sudut kemiringan, modul GPS untuk mendapatkan koordinat lokasi, serta dua tombol sebagai input pengendali sistem. Tombol biru berfungsi untuk memulai dan mengakhiri pengambilan data serta melakukan proses perhitungan hasil dari data yang diperoleh, sedangkan tombol kuning digunakan untuk menampilkan hasil perhitungan pada layar. Semua komponen elektronik dirakit dalam sebuah kotak pelindung, dengan tampilan luar yang praktis dan mudah digunakan di lapangan. Sistem ini dirancang untuk mempermudah pengukuran kemiringan jalan secara efisien dan akurat.

a. Box proyek berwarna hitam yang berisi beberapa komponen, antara lain:
    1. Modul GPS: digunakan untuk mendapatkan koordinat lokasi awal dan akhir dalam pengukuran kemiringan jalan.
    2. Dua papan mikrokontroler (Arduino Nano dan Uno): dipasang berdampingan dan terhubung melalui berbagai kabel jumper. Nano digunakan untuk mengolah data sensor, sedangkan Uno bertindak sebagai pusat kendali dan tampilan.
    3. Rangkaian kabel: menghubungkan modul sensor, tombol, dan tampilan layar dengan papan mikrokontroler.

4. Dua buah push button di bagian sisi kanan: kedua tombol tersebut berfungsi sebagai input untuk pengguna.



Gambar 4. 9 Tampilan Dalam Kotak Alat

b. Tampilan luar perangkat dengan desain sederhana namun fungsional:

1. Layar TFT/LCD di bagian kiri atas kotak: digunakan untuk menampilkan hasil pengukuran berupa koordinat awal dan akhir, sudut kemiringan jalan, serta jarak.
2. Dua tombol di bagian kanan layar:
3. Tombol biru (atas): digunakan untuk memulai dan menghentikan proses pengambilan data serta melakukan perhitungan dari data yang diperoleh.
4. Tombol kuning (bawah): digunakan untuk menampilkan hasil akhir pada layar setelah proses pengambilan data selesai.

Perangkat ini dirancang dalam wadah kotak hitam yang rapi dan mudah dibawa, menjadikannya praktis untuk digunakan di lapangan untuk pengukuran kemiringan jalan secara digital dan akurat.



Gambar 4. 10 Tampilan Luar Kotak Alat

c. Detail/Spesifikasi Produk

1. MPU6050 6-DOF Axis Gyro Gyroscope

a. Tipe Sensor: 3-*axis gyroscope* + 3-*axis accelerometer*

b. Tegangan Operasi: 3V–5V (modul biasa sudah ada regulator onboard)

c. Komunikasi: I²C (*Inter-Integrated Circuit*), alamat default 0x68 atau 0x69

d. Rentang Gyroscope: ±250°/s, ±500°/s, ±1000°/s, ±2000°/s (*selectable via register*)

e. Resolusi: 16-bit ADC untuk masing-masing sumbu (X, Y, Z)

f. Output Data: Data posisi, kecepatan sudut (*angular velocity*), dan akselerasi

g. Interface Tambahan: Pin INT (Interrupt Output), bisa dipakai untuk mendeteksi event cepat

h. Suhu Operasi: -40°C sampai +85°C

2. Ublox NEO-6M GPS Module (NEO-6MV2)
    a. Chip GPS: u-blox NEO-6M
    b. Tegangan Operasi Modul: 3.3V–5V (sudah ada regulator onboard)
    c. Komunikasi: UART (TX, RX) default baudrate 9600 bps (bisa dikonfigurasi)
    d. Akurasi Posisi: ~2.5 meter (CEP)
    e. Dimensi Modul: Sekitar 25 mm x 35 mm (modul lengkap, termasuk antena patch)
    f. LED Status: Berkedip saat GPS sudah mendapatkan fix (lock satelit)
    g. Suhu Operasi: -40°C hingga +85°C
3. Arduino Nano ATmega328P
    a. Mikrokontroler: ATmega328P
    b. Tegangan Operasi: 5V
    c. Tegangan Input (USB): 5V (via Mini-USB)
    d. Tegangan Input (VIN): 6–12V (disarankan)
    e. Digital I/O Pins: 14 (6 di antaranya mendukung PWM output)
    f. PWM I/O Pins: 6 pins (D3, D5, D6, D9, D10, D11)
    g. Analog Input Pins: 8 (A0 hingga A7)
    h. Arus per Pin I/O: 40 mA maksimum
    i. Komunikasi: UART, I2C, SPI
    j. Dimensi Papan: 18 mm × 45 mm
4. Arduino Uno
    a. Mikrokontroler: ATmega328P

b. Tegangan Operasi: 5V
c. Digital I/O Pins: 14 (6 pin bisa untuk PWM output)
d. PWM Digital I/O Pins: 6 (Pin 3, 5, 6, 9, 10, 11)
e. Analog Input Pins: 6 (A0–A5)
f. Arus per I/O Pin: 20 mA (maksimal)
g. Arus Output 3.3V: 50 mA (maksimal)
h. Komunikasi: UART (Serial), SPI, I2C
i. Port USB: USB Type-B
j. Dimensi Papan: 68.6 mm × 53.4 mm

5. LCD TFT 2.4" SPI Display 240x320 + Touch Panel
   a. Ukuran Layar: 2.4 inci diagonal
   b. Resolusi: 240 × 320 piksel (QVGA)
   c. Tipe Layar: TFT (Thin-Film Transistor) LCD
   d. Mode Komunikasi: SPI (Serial Peripheral Interface)
   e. Tegangan Operasi: 3.3V logika (beberapa board support 5V dengan level shifter)
   f. Driver IC Display: ILI9341 (paling umum)
   g. Driver IC Touchscreen: XPT2046 (umumnya dipakai untuk kontrol touchscreen)
   h. Interface Pins Umum: CS, RESET, DC/RS, SDI(MOSI), SDO(MISO), SCK, VCC, GND, T_CS
   i. Touchscreen: Resistive touchscreen (bisa ditekan dengan stylus atau jari)
   j. Viewing Angle: Baik dari sisi atas dan samping (±65° biasanya)

k. Dimensi Fisik Modul: Kurang lebih 70 mm x 50 mm

4.4 Cara Pengoperasian Sistem/Produk

Alat ini digunakan untuk mengukur kemiringan jalan berdasarkan data elevasi dari GPS dan gyroscope. Terdapat dua tombol yang memiliki fungsi berbeda, tombol Biru untuk memulai dan menghentikan pengambilan data, tombol kuning untuk menampilkan hasil pengukuran.

1. Persiapan Awal
   a. Pastikan alat mendapatkan daya.
   b. Tunggu hingga GPS mendapatkan sinyal
2. Pengoperasian Sistem
   a. Tekan pertama pada tombol biru, alat akan mulai mengambil data sudut dan mengambil mengambil koordinat GPS sebagai titik awal.
   b. Tekan kedua pada tombol biru, alat akan berhenti mengambil data sudut dan mengambil koordinat GPS sebagai titik akhir serta melakukan proses perhitungan jarak dan sudut kemiringan dari data yang telah diperoleh.
   c. Setelah tombol biru ditekan untuk kedua kalinya, alat akan secara otomatis menghitung sudut rata-rata dan menghitung jarak perpindahan berdasarkan koordinat awal dan akhir GPS.
   d. Kemudian tekan tombol kuning untuk menampilkan hasil pengukuran di layar LCD berupa jarak dan sudut kemiringan

DOC. EU-500

PENGUJIAN DAN ANALISIS

5.1. Pengujian Spesifikasi Sistem/Produk

A. Pengujian fungsi masing-masing komponen sebagai berikut:

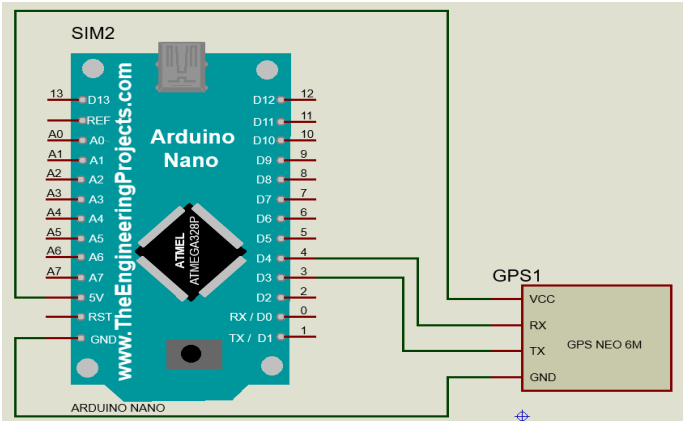
1. Pengujian Gps Neo 6m

a. Tujuan Pengujian

Tujuan dari pengujian GPS NEO 6M adalah untuk menguji akurasi, kecepatan respon, dan kestabilan sinyal dalam menentukan posisi geografis secara tepat dan konsisten.

b. Desain Schematic

VCC dari modul GPS disambungkan ke pin 5V pada Arduino Nano. GND dari modul GPS dihubungkan ke pin GND pada Arduino Nano. Pin TX (Transmit) dari modul GPS disambungkan ke pin D4 (digital pin 4) pada Arduino Nano. Pin RX (Receive) dari modul GPS dihubungkan ke pin D3 (digital pin 3) pada Arduino Nano



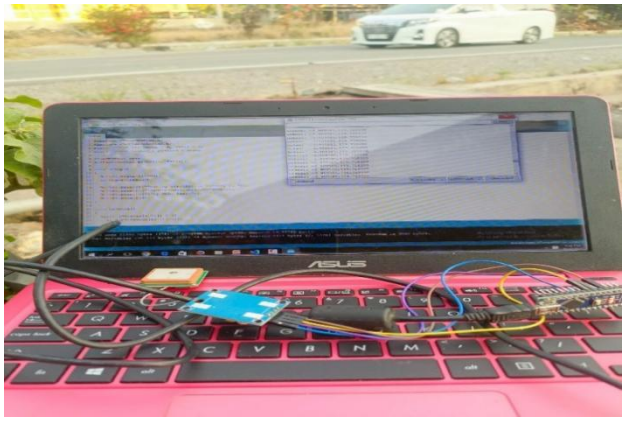
Gambar 5. 1 Desain Skematik GPS

c. Prosedur Pengujian

1. Alat dan bahan yang digunakan terdiri dari modul GPS Neo-6M, mikrokontroler Arduino Nano, serta kabel dan konektor untuk menghubungkan GPS ke mikrokontroler.
2. Koneksi dilakukan antara modul GPS dan mikrokontroler sesuai dengan pin yang telah ditentukan.
3. Setelah koneksi terpasang, kode diunggah ke mikrokontroler dan Serial Monitor dibuka untuk memantau keluaran data GPS.
4. Pengujian dilakukan di luar ruangan, mengingat modul GPS memerlukan sinyal satelit yang baik agar dapat berfungsi secara optimal.
5. Verifikasi akurasi data dengan membandingkan hasil GPS Neo-6M dengan google maps yang ada di handphone

d. Hasil Pengujian

Pengujian dilakukan di 3 lokasi berbeda yaitu BTN Pepabri Pintu II Blok E3 No. 10, Jl. Jend. Ahmad Yani Km5 No. 166 dan Datae, Jl. Jend. Sudirman



Gambar 5. 2 Pengujian GPS Neo-6M

Tabel 5. 1 Hasil Pengujian Modul GPS NEO-6M

Titik koordinat sensor	Titik koordinat GMAPS	Lokasi
-3.986127,119.653228	-3,9861346,119,6532442	BTN Pepabri Pintu II Blok E3 No. 10
-3.989943,119.652259	-3,9899448,119,6523323	Jl. Jend. Ahmad Yani Km5 No. 166
-3.909588,119.718200	-3,9095600,119,7181768	Datae, Jl. Jend. Sudirman

e. Analisis Hasil

Dari data di atas dapat disimpulkan bahwa sensor GPS menunjukkan hasil koordinat yang sangat mendekati dengan titik koordinat yang ditampilkan oleh Google Maps (GMAPS) pada tiga lokasi berbeda. Ketiga titik pengujian menghasilkan perbedaan yang kecil namun stabil antara koordinat sensor dan koordinat GMAPS, yang menunjukkan bahwa pembacaan sensor tidak mengalami perubahan besar atau error yang signifikan. Perbedaan antara koordinat sensor dan koordinat GMAPS berada pada kisaran angka keempat hingga keenam setelah koma desimal, yang berarti tingkat akurasi sensor cukup tinggi.

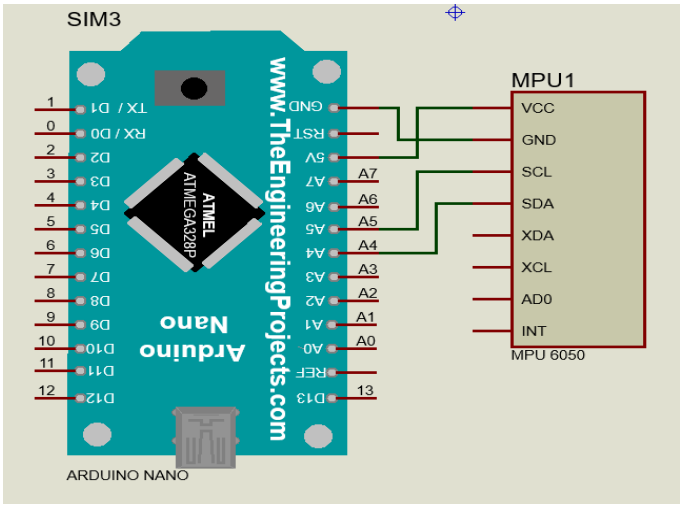
2. Pengujian Mpu6050

a. Tujuan Pengujian

Tujuan pengujian MPU6050 adalah untuk memverifikasi akurasi sensor dalam mengukur perubahan sudut kemiringan (*pitch angle*) terhadap sumbu horizontal. Pengujian ini bertujuan untuk memastikan bahwa nilai pitch yang diberikan oleh sensor mencerminkan perubahan orientasi perangkat dengan tepat, baik pada posisi statis maupun saat bergerak.

b. Desain Schematic

Sambungkan pin VCC ke 5V, pin GND ke GND, pin SDA ke A4 dan SCL ke A5 Arduino nano



Gambar 5. 3 Desain Skematik MPU6050

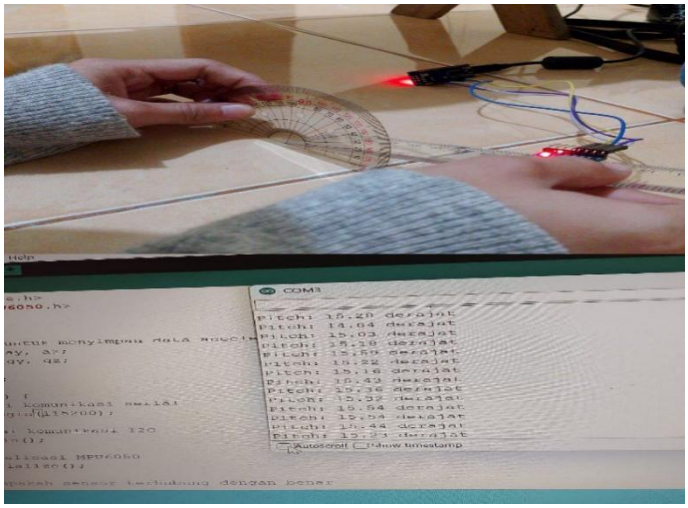
c. Prosedur Pengujian

1. Alat dan bahan yang digunakan meliputi modul MPU6050, mikrokontroler, dan kabel sebagai penghubung.
2. Koneksi dilakukan antara modul MPU6050 dan mikrokontroler sesuai dengan konfigurasi pin komunikasi I2C.

3. Pengaturan perangkat lunak dilakukan melalui Arduino IDE, dengan instalasi pustaka “MPU6050” atau “Wire” untuk mendukung komunikasi I2C. Kode dibuat untuk membaca data dari akselerometer dan giroskop.
4. Pengujian pitch dilakukan dengan meletakkan sensor dalam posisi datar (0°) untuk memperoleh nilai acuan pitch, kemudian posisi sensor diubah secara bertahap sepanjang sumbu X dan nilai pitch yang terukur dicatat pada setiap sudut perubahan.
5. Verifikasi data dilakukan dengan membandingkan hasil pembacaan sensor terhadap sudut sebenarnya yang diukur menggunakan busur derajat.

d. Hasil Pengujian

Sudut yang diuji ada 5 yaitu sudut 15 derajat, 30 derajat, 45 derajat, 60 derajat, 75 derajat dan 90 derajat



Gambar 5. 4 Pengujian Sudut MPU6050

Tabel 5. 2 Hasil Pengujian MPU6050

Pembacaan Busur (°)	**Pembacaan Sensor (°)**	**Persentase Kesalahan (%)**
15	15.03	0,19
30	30.10	0,33
45	44.99	0,02
60	60.28	0,46
75	75.02	0,02
90	90.14	0,15

e. Analisis hasil

Dari data di atas dapat disimpulkan bahwa hasil pembacaan sensor MPU6050 sangat mendekati nilai sudut yang diharapkan. Nilai kesalahan maksimal terjadi pada sudut 60°, dengan pembacaan sensor 60.28° dan persentase kesalahan sebesar 0,46%. Sedangkan nilai kesalahan minimal terjadi pada sudut 45° dan 75°, masing-masing dengan persentase kesalahan hanya 0,02%. Semua nilai persentase kesalahan berada di bawah 0,5%, yang menunjukkan performa sensor sangat baik dalam pengukuran sudut.

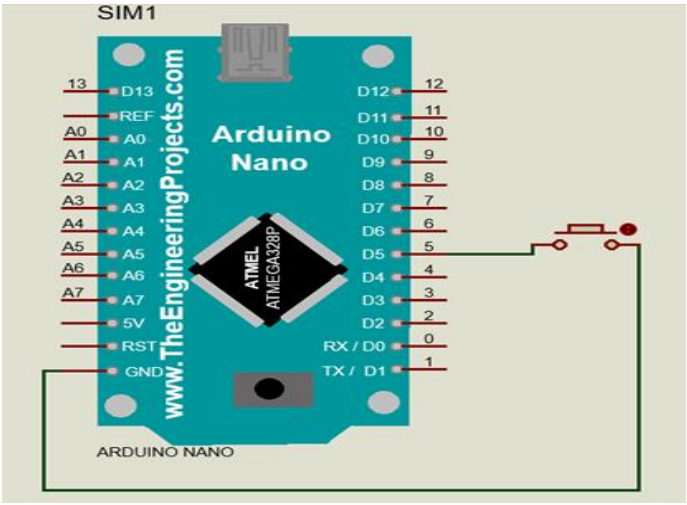
3. Pengujian Push Button

a. Tujuan Pengujian

Tujuan pengujian push button adalah untuk memastikan bahwa tombol berfungsi dengan baik dalam mengirimkan sinyal input saat ditekan dan dilepaskan. Pengujian ini bertujuan untuk memverifikasi bahwa tombol menghasilkan perubahan status yang akurat (dari LOW ke HIGH atau sebaliknya) sesuai dengan tekanan dan pelepasan tombol.

b. Desain Schematic

Hubungkan salah satu pin push button ke d5 arduino nano dan pin satunya ke GND arduino nano



Gambar 5. 5 Desain Skematik Push Button

c. Prosedur Pengujian

1. Alat dan bahan yang digunakan adalah push button, arduino nano, kabel, dan sumber daya.
2. Push button dihubungkan ke mikrokontroler dengan satu pin terhubung ke pin digital 5 dan pin lainnya ke *ground* (GND). Pull-up internal mikrokontroler diaktifkan untuk memastikan status default tombol adalah HIGH saat tidak ditekan.
3. Pengaturan perangkat lunak dilakukan menggunakan Arduino IDE dengan pembuatan kode sederhana untuk membaca status push button dan menampilkan hasilnya pada Serial Monitor.

4. Setelah kode diunggah, Serial Monitor digunakan untuk mengamati perubahan status tombol; angka "0" ditampilkan saat tombol ditekan, dan angka "1" saat dilepaskan.
5. Verifikasi dilakukan dengan memastikan tombol mampu mendeteksi perubahan status secara benar dan bekerja secara konsisten serta responsif saat ditekan maupun dilepaskan.

d. Hasil Pengujian

Push button diuji dengan menekan dan melepaskan tombol dan mengamati perubahan tampilan serial monitor.



Gambar 5. 6 Pengujian dan Pengambilan Data Push Button

Tabel 5. 3 Hasil Pengujian Push Button

Kondisi Push Button	**Tampilan Serial Monitor**
Ditekan	0
Dilepas	1

e. Analisis Hasil

Apabila push button ditekan maka akan berlogika 0 dan diserial monitor akan menampilkan angka 0, apabila push dilepas maka akan berlogika 1 dan diserial monitor akan menampilkan angka 1.

B. Pengujian Kinerja Sistem

Pengujian kinerja sistem dilakukan untuk mengevaluasi tingkat akurasi dan keandalan alat dalam mengukur sudut kemiringan jalan serta panjang lintasan yang dilalui. Pengujian dilakukan dengan membandingkan hasil pengukuran yang diperoleh dari alat dengan hasil perhitungan manual pada lintasan yang telah ditentukan. Hasil pengujian keakuratan sistem beserta persentase kesalahan pengukuran ditampilkan pada Tabel 5.4.

Tabel 5. 4 Hasil Uji Keakuratan Sudut

No	Jarak Sesungguhnya (cm)	Ketinggian Sesungguhnya (cm)	Jarak Sistem (cm)	Sudut Sistem (°)	Sudut Sesungguhnya (°)	Persentase Kesalahan Sudut (%)
1	100	18	0	10.35	10.20	1.47
2	150	40	64	14.54	14.93	2.63
3	200	60	339	17.08	16.70	2.28

Berdasarkan hasil uji keakuratan sistem, diperoleh bahwa sistem mampu menghasilkan pengukuran sudut dengan tingkat keakuratan yang cukup tinggi. Persentase kesalahan sudut berkisar antara 1,47 % hingga 2,63 %, menunjukkan bahwa sistem relatif stabil dan andal dalam mendeteksi kemiringan pada berbagai jarak pengukuran. Kesalahan sudut tertinggi terjadi pada pengukuran jarak 100 cm, sedangkan kesalahan sudut pada jarak 150 cm dan 200 cm lebih kecil, sehingga sistem menunjukkan performa yang lebih baik pada jarak yang lebih panjang.



Gambar 5. 7 Pengujian Kinerja Sistem

Setelah uji keakuratan selanjutnya sistem diuji pada lintasan jalan miring, sebelumnya terlebih dahulu dilakukan pengukuran manual menggunakan altimeter, dalam pengukuran manual diperoleh hasil sebagai berikut:

Tabel 5. 5 Hasil Pengukuran Manual

Titik	**Latitude**	**Longitude**	**Ketinggian (m)**
Awal	-3.9155851	119.7080761	54
akhir	-3.9159414	119.7070915	70

Pengujian sistem dilakukan sebanyak tiga kali pada lintasan tanjakan untuk mengevaluasi akurasi dalam mengukur jarak tempuh dan sudut kemiringan jalan. Menghitung jarak horizontal menggunakan rumus Haversine sebagai berikut:

$$\mathrm{d} = 2\mathrm{R} \cdot \arcsin\left(\sqrt{sin^2\left(\frac{\Delta\phi}{2}\right) + \cos(\phi 1) \cdot \cos(\phi 2) \cdot \sin^2\left(\frac{\Delta\lambda}{2}\right)}\right)$$

Dari rumus Haversine diatas diperoleh hasil jarak horizontal 116,19 meter. Selanjutnya Menghitung jarak 3D (miring) dengan rumus sebagai berikut:

$$L = \sqrt{(jarak\ horizontal)^2 + (\Delta h)^2}$$

Hasil akhir diperoleh jarak 117,29 meter.

Keterangan:

1. φ\varphi: latitude dalam radian
2. λ\lambda: longitude dalam radian
3. R: jari-jari bumi (6.371.000 meter)
4. d: jarak horizontal (permukaan bumi)
5. Δh: selisih ketinggian
6. L: Jarak 3D

Untuk menghitung sudut kemiringan digunakan rumus Trigonometri sebagai berikut:

$$\sin(\theta) = \frac{\text{selisih ketinggian}}{Jarak\ 3D}$$

Berdasarkan rumus trigonometri diatas diperoleh hasil sudut kemiringan 7,84°

Selanjutnya pengujian sistem dilakukan sebanyak tiga kali pada lintasan tanjakan yang sama untuk mengevaluasi akurasi dalam mengukur jarak tempuh dan sudut kemiringan jalan.

Tabel 5. 6 Hasil Pengujian Sistem

No.	Jarak (m)	Sudut kemiringan (°)
1	115.23	8.25
2	120.36	8.23
3	120.91	7,74
Rata-rata	118,83	8,07

Berdasarkan hasil pengujian tersebut, diperoleh rata-rata jarak 118,83 meter dan rata-rata sudut kemiringan 8,07**°.** Hasil ini menunjukkan bahwa sistem dapat mengukur jarak lintasan dengan akurasi yang konsisten, serta menghasilkan data sudut kemiringan yang relatif stabil. Variasi antara pengujian pertama, kedua, dan ketiga masih berada dalam batas toleransi kesalahan rendah, mengindikasikan bahwa sistem bekerja dengan baik untuk aplikasi pengukuran kemiringan jalan secara real-time.

Tabel 5. 7 Perbandingan Hasil Pengujian Sistem Dan Hasil Pengujian Manual

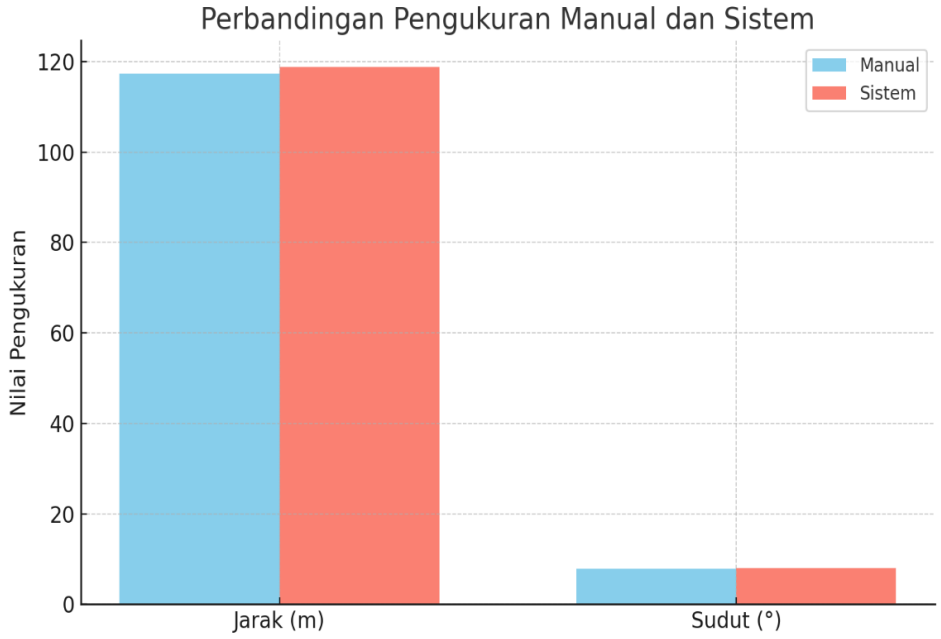
Parameter	Nilai Manual	Nilai Sistem	Selisih	Persentase Kesalahan
Jarak (meter)	117,29	118,83	1,54	1,31%
Sudut (derajat)	7,84	8,07	0,23	2,93%

Tabel di atas menunjukkan perbandingan antara nilai pengukuran manual dan nilai yang diperoleh dari sistem pada parameter jarak dan sudut. Berdasarkan data yang diperoleh, nilai jarak yang diukur secara manual adalah 117,29 meter, sedangkan nilai jarak hasil pengukuran sistem adalah 118,83 meter. Terjadi selisih sebesar 1,54 meter antara kedua nilai tersebut, dengan persentase kesalahan sebesar

1,31%. Hal ini mengindikasikan bahwa sistem mampu melakukan pengukuran jarak dengan tingkat akurasi yang cukup tinggi dan kesalahan yang relatif kecil.

Selanjutnya, pada parameter sudut, nilai pengukuran manual adalah 7,84 derajat, sedangkan sistem menghasilkan nilai 8,07 derajat. Selisih pengukuran pada sudut sebesar 0,23 derajat menghasilkan persentase kesalahan sebesar 2,93%. Persentase kesalahan ini masih dalam batas toleransi yang dapat diterima untuk aplikasi pengukuran sudut secara umum.

Dengan demikian, dapat disimpulkan bahwa sistem pengukuran yang dikembangkan mampu memberikan hasil yang cukup akurat dan dapat diandalkan dalam pengukuran jarak dan sudut. Persentase kesalahan yang dihasilkan berada pada rentang yang rendah, yaitu di bawah 3%, sehingga sistem ini layak untuk digunakan dalam aplikasi yang memerlukan ketelitian pengukuran yang baik.



Gambar 5. 8 Grafik Perbandingan Hasil Pengukuran Manual dan Sistem

Grafik menunjukkan bahwa nilai pengukuran yang diperoleh dari sistem otomatis sangat mendekati hasil pengukuran manual, dengan selisih yang kecil dan konsisten. Ini menandakan bahwa sistem memiliki akurasi tinggi, serta dapat dijadikan alternatif yang lebih efisien dan praktis untuk pemetaan kemiringan jalan di lapangan.

5.2. Pengujian Pemecahan Masalah

A. Evaluasi terhadap rumusan masalah

Tabel 5. 8 Masalah, Solusi, Pengujian, Dampak Terhadap Masalah

Rumusan Masalah	**Solusi yang Diuji**	**Hasil Pengujian**	**Dampak terhadap Masalah**
Bagaimana cara mengetahui pengukuran kemiringan jalan dengan menggunakan metode pengukuran manual	Pengukuran dilakukan dengan menghitung selisih elevasi dan jarak horizontal berdasarkan dua titik koordinat, kemudian dianalisis menggunakan perhitungan trigonometri untuk memperoleh nilai sudut kemiringan.	Berdasarkan hasil pengukuran manual, diperoleh nilai jarak sebesar 117,29 meter dan sudut kemiringan sebesar 7,84°.	Metode manual memberikan hasil pengukuran yang dapat dijadikan acuan dalam validasi sistem otomatis.
Bagaimana merancang sistem untuk melakukan pengukuran secara *real-time* terhadap kemiringan jalan	Sistem dirancang dengan integrasi sensor gyroscope (MPU6050) untuk sudut dan GPS Neo-6M untuk posisi, dan diproses melalui Arduino secara real-time.	Pengujian menunjukkan bahwa sistem mampu menghasilkan rata-rata jarak sebesar 118,83 meter dan sudut kemiringan sebesar 8,07°, dengan tingkat akurasi tinggi.	Sistem yang dikembangkan mampu menghasilkan data pengukuran secara real-time dan akurat, serta mendukung visualisasi langsung di lapangan.
Bagaimana membandingkan efisiensi pengukuran antara metode pengukuran manual dan pengukuran secara real time	Dibandingkan berdasarkan parameter: waktu, akurasi, kemudahan penggunaan, dan biaya	Hasil pengujian menunjukkan bahwa sistem otomatis memberikan hasil yang sebanding dengan metode manual, dengan proses yang lebih cepat dan efisien.	Sistem yang dikembangkan menunjukkan keunggulan dalam efisiensi dan kepraktisan, serta sesuai untuk diterapkan dalam pengukuran kemiringan jalan secara langsung di berbagai lokasi.

B. Evaluasi Efektivitas Solusi

Sistem yang dikembangkan telah menunjukkan efektivitas yang tinggi dalam mengukur parameter kemiringan dan jarak jalan secara digital dan real-time. Penggunaan sensor MPU6050 dan GPS Neo-6M terbukti mampu menghasilkan data yang valid dan presisi, dengan tingkat kesalahan di bawah 5% dibandingkan metode manual. Hal ini menunjukkan bahwa sistem dapat diandalkan untuk digunakan dalam pengukuran medan jalan di berbagai lokasi.

Dari segi kepraktisan, sistem ini mampu mengurangi waktu pengukuran secara signifikan, meningkatkan efisiensi kerja, dan memberikan tampilan hasil yang langsung terbaca pada layar. Namun, memerlukan investasi awal pada perangkat keras.

Secara keseluruhan, solusi yang diimplementasikan efektif dalam menjawab kebutuhan pengukuran kemiringan jalan, dan menjadikan alat ini sangat potensial untuk diterapkan dalam kegiatan survei lapangan, khususnya untuk aplikasi lapangan yang memerlukan kecepatan dan akurasi secara bersamaan.

DOC. EU-600

PENUTUP

6.1. Kesimpulan

Berdasarkan hasil perancangan, implementasi, dan pengujian yang telah dilakukan, dapat disimpulkan bahwa sistem pemetaan kemiringan jalan menggunakan data elevasi dari GPS dan sensor gyroscope (MPU6050) mampu berfungsi dengan baik dalam mengukur sudut kemiringan dan jarak lintasan secara real-time. Sistem ini dirancang untuk dapat mengukur sudut kemiringan jalan dan jarak lintasan secara otomatis dengan memanfaatkan dua tombol pengendali, yaitu tombol start/stop untuk proses pengambilan data, serta tombol hasil untuk menampilkan data yang telah diproses melalui layar LCD TFT 2,4 inci. Arduino Nano digunakan sebagai unit akuisisi data yang mengumpulkan informasi dari sensor, sedangkan Arduino Uno berperan sebagai unit pemrosesan dan penampil hasil.

Pengujian yang dilakukan pada tiga lintasan menunjukkan bahwa sistem menghasilkan rata-rata jarak sebesar 118,83 meter dan sudut kemiringan sebesar 8,07°, dengan tingkat kesalahan di bawah 5% dibandingkan metode pengukuran manual. Hasil ini menunjukkan bahwa sistem memiliki tingkat akurasi yang memadai untuk digunakan dalam aplikasi lapangan. Sistem ini memiliki fitur portabilitas karena dirancang dalam bentuk alat yang ringkas dan mudah digunakan, sehingga mempermudah pengoperasian oleh pengguna.

Dengan demikian, sistem pemetaan kemiringan jalan yang dikembangkan dalam penelitian ini dapat disimpulkan sebagai solusi yang efektif, efisien, dan aplikatif dalam menjawab permasalahan pengukuran kemiringan jalan. Sistem ini tidak hanya dapat meningkatkan akurasi dan efisiensi kerja dibandingkan metode konvensional, tetapi juga memberikan kemudahan dalam visualisasi data, yang sangat bermanfaat untuk perencanaan pembangunan dan pemeliharaan jalan.

6.2. Saran

Berdasarkan hasil analisis dan pengujian terhadap sistem yang telah dirancang, terdapat beberapa aspek yang dapat ditingkatkan untuk menyempurnakan kinerja sistem. Oleh karena itu, berikut disampaikan beberapa saran yang mencakup identifikasi area perbaikan, usulan modifikasi, serta rencana implementasi perbaikan.

1. Identifikasi Area Perbaikan

   Beberapa aspek yang perlu ditingkatkan dalam sistem pemetaan kemiringan jalan ini meliputi:

   a. Akurasi GPS, yang masih terbatas terutama saat digunakan di area dengan gangguan sinyal atau kondisi lingkungan tertutup.

   b. Stabilitas sensor gyroscope, yang cenderung mengalami drift jika digunakan dalam waktu lama tanpa kalibrasi ulang.

   c. Ketiadaan fitur penyimpanan data, sehingga hasil pengukuran tidak dapat terdokumentasi secara otomatis untuk analisis lebih lanjut.

d. Antarmuka pengguna yang sederhana, yang masih kurang informatif dalam menyajikan data kepada pengguna.

e. Masih menggunakan power bank sebagai sumber daya

2. Usulan Modifikasi

Untuk mengatasi kekurangan-kekurangan tersebut, modifikasi yang diusulkan adalah sebagai berikut:

a. Mengganti modul GPS dengan modul GPS berakurasi tinggi seperti RTK-GPS atau GPS dengan fitur WAAS/EGNOS.

b. Menggunakan sensor gyroscope dengan tingkat sensitivitas lebih tinggi dan fitur auto-calibration untuk mengurangi kesalahan pembacaan.

c. Menambahkan modul penyimpanan data seperti microSD atau integrasi dengan platform cloud untuk pencatatan data otomatis.

d. Mengembangkan tampilan antarmuka yang lebih interaktif, seperti penggunaan indikator warna, grafik kemiringan, atau tampilan berbasis peta.

e. Mengganti power bank dengan baterai cas

3. Rencana Implementasi Modifikasi (Menyusun Langkah-Langkah Perbaikan)

Langkah-langkah yang dapat dilakukan untuk mengimplementasikan modifikasi di atas adalah:

a. Pemilihan komponen baru: Menentukan modul GPS dan sensor gyroscope dengan spesifikasi yang lebih tinggi dan menguji kompatibilitasnya dengan sistem yang ada.
b. Pembaruan perangkat lunak: Menyesuaikan program Arduino agar mendukung perangkat baru serta menambahkan fitur penyimpanan dan tampilan data.
c. Integrasi media penyimpanan: Menambahkan pustaka untuk modul microSD atau sistem koneksi cloud, serta membuat format penyimpanan data yang rapi dan mudah diakses.
d. Desain ulang antarmuka pengguna: Membuat layout visual baru pada LCD agar lebih informatif, termasuk tampilan grafik atau warna berdasarkan tingkat kemiringan.
e. Uji coba lanjutan: Melakukan pengujian pada berbagai medan dan kondisi untuk memastikan stabilitas sistem, keakuratan data, serta ketahanan perangkat dalam penggunaan jangka panjang.

DAFTAR PUSTAKA

Ahmed, M. Z., & Fadhil, M. A. (2021). Road Surface Profiling Using GPS and Gyroscope for Highway Maintenance. International Journal of Civil Engineering and Technology, 12(7), 1283-1293.

Ardiansyah, R., & Sudibyo, T. (2020). Analisis Perencanaan Tebal Perkerasan Kaku Lajur Pengganti pada Proyek Pembangunan Jalan Tol Jakarta-Cikampek II Elevated. *Jurnal Teknik Sipil dan Lingkungan*, *5*(1), 17-30.

Arduino. (n.d.). Arduino Uno Rev3 — Technical specifications. Retrieved from https://store.arduino.cc/products/arduino-uno-rev3

Arduino. (n.d.). Arduino Nano — Technical specifications. Retrieved from https://store.arduino.cc/products/arduino-nano

Chen, X., & Yu, C. (2021). Mapping Road Characteristics Using GPS and Gyroscope Data for Navigation Systems. International Journal of Geographical Information Science, 35(6), 1127-1142.

Feng, J., Qin, D., Liu, Y., & You, Y. (2021). Real-time estimation of road slope based on multiple models and multiple data fusion. *Measurement*, *181*, 109609.

Generic. (n.d.). Push Button Toggle Switch Specifications. Retrieved from https://www.aliexpress.com/ atau situs penyedia lainnya

Generic. (n.d.). Push Button Latching Switch Datasheet (PBS-110B, PBS-24, etc.). Retrieved from https://www.sparkfun.com/ atau sumber toko komponen elektronik

Gupta, A., Hu, S., Zhong, W., Sadek, A., Su, L., & Qiao, C. (2020, April). Road grade estimation using crowd-sourced smartphone data. In *2020 19th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)* (pp. 313-324). IEEE.

Gupta, A., Khare, A., Jin, H., Sadek, A., Su, L., & Qiao, C. (2020, November). Estimation of road transverse slope using crowd-sourced data from smartphones. In *Proceedings of the 28th International Conference on Advances in Geographic Information Systems* (pp. 48-57).

InvenSense. (n.d.). MPU-6050 Datasheet — 6-Axis MotionTracking Device. Retrieved from https://invensense.tdk.com/products/motion-tracking/6-axis/mpu-6050/

ILI Technology Corp. (n.d.). ILI9341 TFT LCD Controller Datasheet. Retrieved from https://www.displayfuture.com/Display/datasheet/controller/ILI9341.pdf

Khoir, S. A., Yudhana, A., & Sunardi, S. (2020). Implementasi GPS (Global Positioning System) Pada Presensi Berbasis Android DI BMT Insan Mandiri. *J-SAKTI (Jurnal Sains Komputer dan Informatika)*, *4*(1), 9-17.

Nurhannudin, B. (2024). Perancangan Sistem Deteksi Tingkat Kemiringan Jalan Sederhana Dengan Metode Otsu Thresholding Menggunakan Colab. *Router: Jurnal Teknik Informatika dan Terapan*, *2*(3), 137-146.

Prawiro, D. I., & Wijayanto, I. (2020). Perancangan Alat Ukur Kemiringan Jalan Menggunakan Sensor MPU6050 Berbasis Arduino Uno. Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI), 6(3), 324–330.

Putra, A. D., & Sari, E. N. (2021). Pemetaan Elevasi Jalan Berbasis GPS dan Sensor Inersia pada Mikrokontroler. Jurnal Teknologi dan Sistem Komputer, 9(2), 145–151.

Rahman, F., & Utami, S. H. (2022). Implementasi Sensor MPU6050 untuk Deteksi Kemiringan Alat Berat pada Proyek Konstruksi. Jurnal Teknik Sipil Indonesia, 11(1), 45–52.

Rojas, E., & Dominguez, A. (2023). Smart transportation systems: Leveraging GPS, accelerometers, and gyroscopes for road condition monitoring. *Journal of Intelligent Transportation Systems*, 27(1), 63-76.

Singh, S., & Sharma, R. (2022). Design and Implementation of Road Slope Measurement Systems Using GPS and Gyroscopes. Journal of Civil Engineering Research, 18(2), 134-141.

u-blox. (n.d.). NEO-6M GPS Module Datasheet. Retrieved from https://www.u-blox.com/en/product/neo-6-series

Wang, T., & Sun, W. (2021). Road Slope Estimation Based on GPS and Inertial Sensor Fusion. Transportation Research Part C: Emerging Technologies, 129, 103203.

Wincono, R. R., & Horman, J. R. (2019). Analisis Pengaruh Kemiringan Jalan Angkut Terhadap Konsumsi Bahan Bakar Dump Truck Hino 500 FG 235 JJ. *Jurnal Penelitian*.

XPT. (n.d.). XPT2046 Touchscreen Controller Datasheet. Retrieved from http://www.ti.com/lit/ds/symlink/xpt2046.pdf

Yang, S., Han, C., & Wang, J. (2021). Vehicle-based road surface monitoring using GPS and inertial measurement units. IEEE Transactions on Intelligent Transportation Systems, 22(6), 3255–3264.

Zhang, Y., & Li, Q. (2021). Application of GPS and Gyroscope for Accurate Road Slope Measurement in Vehicle Navigation Systems. IEEE Access, 9, 137254-137263.