

Laporan Praktik Kerja Lapangan

**PENGUJIAN DAN ANALISA PEMBACAAN SENSOR *ACCELEROMETER*,
GYROSCOPE, DAN MODUL GPS MENGGUNAKAN MIKROKONTROLER
STM32F411 SEBAGAI IDE PENGEMBANGAN *FLIGHT CONTROLLER*
DRONE**

**Pusat Riset Teknologi Penerbangan
Badan Riset dan Inovasi Nasional
(PUSTEKBANG-BRIN)**



Disusun Oleh :

Muhammad Maulidan

1900014043

**PROGRAM STUDI FISIKA
FAKULTAS SAINS DAN TEKNOLOGI TERAPAN**

UNIVERSITAS AHMAD DAHLAN

YOGYAKARTA

2023

HALAMAN PENGESAHAN
LAPORAN PRAKTIK KERJA LAPANGAN

Judul : Pengujian dan Analisa Pembacaan Sensor Accelerometer, Gyroscope, dan Modul GPS Menggunakan Mikrokontroler STM32F411 Sebagai Alternatif *Flight Controller* Pada Drone

Tempat PKL : Pusat Riset Teknologi Penerbangan, Badan Riset dan Inovasi Nasional (PUSTEKBANG-BRIN)

Alamat : Jl. Raya Lapan Sukamulya, Kec. Rumpin, Kabupaten Bogor, Jawa Barat 16350

Periode PKL : 08 Agustus – 10 September 2022

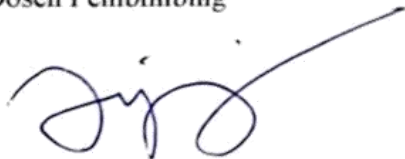
Disusun Oleh :

Nama : Muhammad Maulidan

NIM : 1900014043

Disetujui Oleh:

Dosen Pembimbing



Apik Rusdiarna Indra Praja, S.Si., M.T.

NIY. 60171020

Pembimbing Lapangan



Yanuar Firmansyah, S.ST.

NIP. 199607132018011001

Mengetahui,

Dekan Fakultas Sains dan Teknologi Terapan



Dr. H. Yudi Ari Adi, S.Si., M.Si.

NIY. 60020389

Ketua Program Studi Fisika



Damar Yoga Kusuma, B.Eng., Ph.D.

NIY. 60150785

Laporan Praktik Kerja Lapangan

**Pengujian dan Analisa Pembacaan Sensor Accelerometer, Gyroscope, dan Modul GPS
Menggunakan Mikrokontroler STM32F411 Sebagai Alternatif *Flight Controller* Pada Drone**

Yang Dipersiapkan dan Disusun Oleh:

Muhammad Maulidan

1900014043

Yang Disetujui dan Disahkan Oleh:

Pusat Riset Teknologi Penerbangan

Badan Riset dan Inovasi Nasional (PUSTEKBANG-BRIN)

Rumpin, Bogor

Pada Tanggal 8 September 2022

Dan Dinyatakan Telah Memenuhi Syarat

Disahkan Oleh:

Koordinator Humas BRIN

Kawasan Multi Unit Kerja Rumpin Bogor,



Tri Widodo, S.Sos

NIP. 197003101992031003

Pembimbing Lapangan



Yanuar Firmansyah, S.ST.

NIP. 199607132018011001

KATA PENGANTAR

Alhamdulillah rabbil 'alamiin, puji syukur penulis panjatkan kehadirat Allah SWT. Karena berkat rahmat dan hidayah-Nya penulis dapat menyelesaikan kegiatan Praktik Kerja Lapangan dan laporan yang berjudul “Pengujian dan Analisa Pembacaan Sensor *Accelerometer*, *Gyroscope*, dan Modul GPS Menggunakan Mikrokontroler STM32F411 Sebagai Ide Pengembangan *Flight Controller Drone*” dengan lancar.

Laporan Praktik Kerja Lapangan ini disusun dalam rangka kegiatan Praktik Kerja Lapangan selama lima minggu di Pusat Riset Teknologi Penerbangan, Badan Riset dan Inovasi Nasional (PUSTEKBANG-BRIN) dengan melakukan pengamatan, pengujian serta analisa terhadap hasil pembacaan sensor *accelerometer*, *gyroscope*, dan modul GPS menggunakan mikrokontroler STM32F411. Hal yang lebih detail akan dibahas di dalam laporan ini.

Terselesaikannya laporan ini tidak lepas dari dukungan berbagai pihak yang telah banyak memberikan masukan, suntikan moral, semangat, ilmu, serta bimbingannya. Oleh karena itu, pada kesempatan ini penulis ingin mengucapkan terima kasih kepada:

1. Bapak Damar Yoga Kusuma, B.Eng., Ph.D. selaku Ketua Program Studi Fisika Universitas Ahmad Dahlan.
2. Bapak Bagus Haryadi, S.Si., M.T., Ph.D. selaku Dosen Pembimbing Akademik.
3. Bapak Apik Rusdiarna Indra Praja, S.Si., M.T. selaku Dosen Pembimbing Praktik Kerja Lapangan.
4. Bapak Yanuar Firmansyah, S.ST. selaku Pembimbing Lapangan di PUSTEKBANG-BRIN yang sudah banyak memberikan masukan, saran, bimbingan dan arahan selama pelaksanaan Praktik Kerja Lapangan.
5. Bapak Sanwani yang sudah banyak memberikan bantuan terkait administrasi.
6. Bapak Suyitno selaku petugas dari Mess Putra Lapan yang telah menyiapkan segala fasilitas dan kenyamanan selama tinggal di Mess Putra Lapan Rumpin.

7. Teman-teman pejuang KP dan TA di PUSTEKBANG-BRIN dari ITDA, ITB, UNAND, UIN Syarif Hidayatullah, serta teman-teman kelas Fisika UAD angkatan 2019 atas semangat dan bantuannya.
8. Orang tua yang tidak pernah henti memberikan dukungan baik dukungan moril maupun materiil serta kakak dan adik saya juga berbagai pihak yang tidak bisa saya sebutkan satu persatu.

Penulis menyadari bahwa dalam penulisan laporan Praktik Kerja Lapangan ini masih banyak kesalahan dan jauh dari kata sempurna. Oleh karena itu berbagai kritik, saran, masukan dan komentar yang membangun sangat diharapkan untuk kesempurnaan laporan ini. Semoga laporan Praktik Kerja Lapangan ini dapat memberikan manfaat khususnya bagi penulis maupun pembaca sekalian.

Yogyakarta, September 2023

Muhammad Maulidan

DAFTAR ISI

HALAMAN SAMPUL	i
HALAMAN PENGESAHAN.....	ii
HALAMAN PERSETUJUAN.....	iii
KATA PENGANTAR	iv
DAFTAR ISI.....	vi
DAFTAR GAMBAR	ix
DAFTAR TABEL.....	xi
DAFTAR LAMPIRAN.....	xii
BAB 1. PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Tujuan Praktik Kerja Lapangan	2
1.4. Manfaat Praktik Kerja Lapangan	3
1.5. Lokasi dan Waktu Praktik Kerja Lapangan	3
BAB 2. TINJAUAN INSTANSI	4
2.1. Badan Riset dan Inovasi Nasional (BRIN)	4
2.1.1. Pendahuluan.....	4
2.1.2. Sejarah Singkat.....	4
2.1.3. Visi dan Misi BRIN.	4
2.1.4. Tugas dan Fungsi BRIN.....	5
2.1.5. Struktur Organisasi dan Tata Kerja BRIN.	6
2.1.6. Organisasi Riset BRIN.....	8
2.1.7. Logo BRIN.....	8
2.2. Pusat Riset Teknologi Penerbangan (PUSTEKBANG).....	9
2.2.1. Visi dan Misi.....	9
2.2.2. Tugas dan Fungsi	9

2.2.3. Struktur Organisasi PUSTTEKBANG.....	10
2.2.4. Lokasi PUSTTEKBANG.....	10
BAB 3. TINJAUAN PUSTAKA	11
3.1. Drone.....	11
3.1.1. Pengertian Drone.....	11
3.1.2. Jenis - jenis Drone.....	11
3.1.3. Prinsip Kerja Drone.....	12
3.2. Flight Controller.....	14
3.3. Mikrokontroler STM32F411.....	15
3.4. Sensor Accelerometer	17
3.5. Sensor Gyroscope.....	17
3.6. Modul GPS.....	18
BAB 4. METODOLOGI.....	19
4.1. Tempat dan Waktu Pelaksanaan	19
4.2. Alat dan Bahan.....	19
4.3. Metode Pelaksanaan Praktik Kerja Lapangan.....	19
4.4. Metode Integrasi Sistem.....	21
4.4.1. Integrasi Hardware Rangkaian.	21
4.4.2. Integrasi Software dan Konfigurasi Pin.	23
4.4.3. Codingan Program dan Konfigurasi tools.....	27
4.5. Prosedur Pengambilan Data.....	30
4.5.1. Pengambilan Data Sensor Accelerometer ADXL345.....	30
4.5.2. Pengambilan Data Sensor Gyroscope GY-25.....	31
4.5.3. Pengambilan Data Sensor IMU GY-521(MPU6050).....	32
4.5.4. Pengambilan Data Modul GPS Neo-7m.....	32
BAB 5. HASIL DAN PEMBAHASAN	34
5.1. Hasil	34
5.1.1. Hasil Pembacaan Sensor Accelerometer ADXL345.....	34

5.1.1.1. Analisis Data Hasil Sensor Accelerometer ADXL345.....	37
5.1.2. Hasil Pembacaan Sensor Gyroscope GY-25.....	39
5.1.2.1. Analisis Data Hasil Sensor Gyroscope GY-25	40
5.1.3. Hasil pembacaan Sensor IMU GY-521(MPU6050).....	41
5.1.4. Hasil Pembacaan Modul GPS Neo-7m.....	45
5.1.5. Analisa Kinerja Mikrokontroler STM32F411.....	48
BAB 6. PENUTUP	50
6.1. Kesimpulan	50
6.2. Saran.....	51
DAFTAR PUSTAKA	52
LAMPIRAN.....	53

DAFTAR GAMBAR

Gambar 2.1. Struktur Organisasi BRIN	7
Gambar 2.2. Logo BRIN	9
Gambar 3.1. Drone Untuk Fotografi	11
Gambar 3.2. Drone Fixed Wing	12
Gambar 3.3. Drone Multicopter	12
Gambar 3.4. Prinsip kerja Propeller Multicopter	13
Gambar 3.5. Pixhawk Cube	14
Gambar 3.6. Konfigurasi Pixhawk Cube	15
Gambar 3.7. Konfigurasi Pin STM32F411	16
Gambar 3.8. Prinsip Kerja Gyroscope	17
Gambar 3.9. Modul GPS tipe NEO-7m	18
Gambar 4.1. Diagram Alir Metode Praktik Kerja Lapangan	20
Gambar 4.2. Rangkaian Sensor Accelerometer ADXL345	21
Gambar 4.3. Rangkaian Sensor Gyroscope GY-25.....	21
Gambar 4.4. Rangkaian Sensor IMU GY-521	22
Gambar 4.5. Rangkaian Modul GPS Neo-7m.....	22
Gambar 4.6. Skematik Rangkaian Sensor Accelerometer ADXL345	23
Gambar 4.7. Skematik Rangkaian Sensor Gyroscope GY-25	24
Gambar 4.8. Skematik Rangkaian Sensor IMU GY-521(MPU6050).....	25
Gambar 4.9. Skematik Rangkaian Modul GPS Neo-7m.....	26
Gambar 4.10. Link URL Library STM32 di Software Arduino IDE	27
Gambar 4.11. Boards Manager STM32 di Software Arduino IDE	28
Gambar 4.12. Konfigurasi Tools STM32F411 di Software Arduino IDE	29
Gambar 4.13. Codingan STM32F411 di Software Arduino IDE.....	29
Gambar 4.14. Arah Rotasi Sumbu pada Sensor Accelerometer ADXL345.....	31
Gambar 4.15. Arah Rotasi Sudut Sensor Gyroscope GY-25.....	31
Gambar 4.16. Arah Rotasi Sumbu Sensor IMU GY-521.....	32
Gambar 4.17. Gedung Utama BRIN Lokasi Pengambilan data 1 GPS.....	33
Gambar 4.18. Gedung Avionik Lokasi Pengambilan data 2 GPS.....	33
Gambar 5.1. Respon Output Sensor Accelerometer ADXL345.....	36

Gambar 5.2. Titik Koordinat lokasi 1 Pengambilan data Modul GPS 46
Gambar 5.3. Titik Koordinat lokasi 2 Pengambilan data Modul GPS 47

DAFTAR TABEL

Tabel 4.1. Konfigurasi Pin Sensor Accelerometer ADXL345.....	23
Tabel 4.2. Konfigurasi Pin Sensor Gyroscope GY-25	24
Tabel 4.3. Konfigurasi Pin Sensor IMU GY-521(MPU6050)	25
Tabel 4.4. Konfigurasi Pin Modul GPS Neo-7m	27
Tabel 5.1. Data Hasil Pembacaan Sensor ADXL345 Sumbu x	34
Tabel 5.2. Data Hasil Pembacaan Sensor ADXL345 Sumbu -x.....	34
Tabel 5.3. Data Hasil Pembacaan Sensor ADXL345 Sumbu y	35
Tabel 5.4. Data Hasil Pembacaan Sensor ADXL345 Sumbu -y.....	35
Tabel 5.5. Data Hasil Pembacaan Sensor ADXL345 Sumbu z	35
Tabel 5.6. Data Hasil Pembacaan Sensor ADXL345 Sumbu -z.....	36
Tabel 5.7. Hasil Pengolahan Data Sensor ADXL345 Sumbu x dan -x.....	37
Tabel 5.8. Hasil Pengolahan Data Sensor ADXL345 Sumbu y dan -y.....	38
Tabel 5.9. Hasil Pengolahan Data Sensor ADXL345 Sumbu z dan -z.....	38
Tabel 5.10. Data Hasil Pembacaan Sensor Gyroscope GY-25.....	39
Tabel 5.11. Hasil Pengolahan Data Sensor Gyroscope GY-25.....	40
Tabel 5.12. Data Hasil Pembacaan Sensor IMU GY-521 Sumbu x	41
Tabel 5.13. Data Hasil Pembacaan Sensor IMU GY-521 Sumbu -x	41
Tabel 5.14. Data Hasil Pembacaan Sensor IMU GY-521 Sumbu y	42
Tabel 5.15. Data Hasil Pembacaan Sensor IMU GY-521 Sumbu -y	42
Tabel 5.16. Data Hasil Pembacaan Sensor IMU GY-521 Sumbu z.....	43
Tabel 5.17. Data Hasil Pembacaan Sensor IMU GY-521 Sumbu -z	43
Tabel 5.18. Data Hasil Pembacaan Modul GPS Neo-7m Lokasi 1	45
Tabel 5.19. Data Hasil Pembacaan Modul GPS Neo-7m Lokasi 2	45

DAFTAR LAMPIRAN

Lampiran 1. Listing Program Sensor ADXL345	53
Lampiran 2. Listing Program Sensor GY-25	55
Lampiran 3. Listing Program Sensor IMU GY-521	57
Lampiran 4. Listing Program Modul GPS	58
Lampiran 5. Screenshoot Data Pembacaan Sensor ADXL345	60
Lampiran 6. Screenshoot Data Pembacaan Sensor GY-25	63
Lampiran 7. Screenshoot Data Pembacaan Sensor IMU GY-521	65
Lampiran 8. Screenshoot Data Pembacaan Modul GPS	68
Lampiran 9. Dokumentasi Kegiatan Praktik Kerja Lapangan	69

BAB I PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi dalam kehidupan dimulai dari proses sederhana dalam kehidupan sehari-hari sampai pada tingkat pemenuhan kepuasan sebagai individu dan makhluk sosial. Dari masa ke masa kemajuan teknologi terus berkembang, mulai dari era teknologi pertanian, era teknologi industri, era teknologi informasi, dan era teknologi komunikasi dan informasi. Perkembangan ini membawa berbagai dampak dalam kehidupan bermasyarakat, berbangsa dan bernegara, setiap individu tertarik untuk menggunakan dan memanfaatkan setiap perkembangan ini. Salah satu teknologi yang sedang gencar dikembangkan adalah teknologi penerbangan. Saat ini kita sudah bisa menikmati berbagai bentuk kecanggihan di bidang teknologi penerbangan termasuk teknologi drone.

Drone merupakan tipe pesawat terbang tanpa awak atau biasa dikenal dengan istilah *Unmanned Aerial Vehicle* (UAV) yang beroperasi dengan sendirinya tanpa seorang pilot saat pesawat mengudara. Drone saat ini banyak menjadi bahan penelitian di kalangan mahasiswa dan juga sebagian orang yang menyalurkan hobinya di bidang *aeromodelling* (Pangestu, 2018). Drone terdiri dari beberapa jenis, salah satunya adalah *multidrone* atau *multirotor*. Salah satu *multirotor* yang berkembang saat ini adalah *quadrotor* atau *quadcopter*. Sistem penggerak pada *quadcopter* sendiri terdiri dari 4 buah motor yang tergabung dari sinkronisasi 2 buah motor yang berputar searah jarum jam dan 2 motor yang berputar berlawanan arah jarum jam. Kemampuan *quadcopter* ini didukung oleh komponen utama yang disebut sistem kendali atau *flight controller*.

Flight controller (Fc) adalah kumpulan dari berbagai komponen-komponen dan sensor yang berfungsi untuk menjaga drone tetap seimbang dan dapat terkendali (Setiawan, dkk, 2017). *Flight controller* membutuhkan komponen berupa mikrokontroller sebagai otak pengendali, sensor IMU untuk membaca perubahan *altitude*, sensor *gyroscope* untuk menjaga keseimbangan serta modul GPS untuk mengirimkan data lokasi drone. Pixhawk merupakan salah satu jenis *flight controller* yang sering digunakan, tetapi memiliki batasan pada sistem I/O sehingga sulit untuk dimodifikasi berdasarkan kebutuhan yang diinginkan. Selain itu, harga dari pixhawk sendiri terbilang cukup mahal apalagi untuk pengembangan drone yang bersifat non komersial. Keterbatasan ini memunculkan penelitian untuk merancang dan membuat *flight controller* secara mandiri.

Pusat Teknologi Penerbangan (PUSTEKBANG) BRIN memberikan kesempatan kepada mahasiswa melalui program Praktik Kerja Lapangan (PKL) untuk mengetahui, memahami, dan mempraktikkan ilmu yang didapatkan di dunia perkuliahan secara langsung khususnya di bidang pengembangan teknologi penerbangan. Berdasarkan Peraturan Kepala LAPAN Nomor 8 Tahun 2015 tentang

Organisasi dan Tata Kerja LAPAN, Pasal 86, Pusat Teknologi Penerbangan mempunyai tugas pokok melaksanakan penelitian, pengembangan, perekayasaan dan pemanfaatan serta penyelenggaraan keantariksaan di bidang teknologi aeronautika.

Melalui program Praktik Kerja Lapangan di Pusat Teknologi Penerbangan BRIN penulis melaksanakan penelitian untuk melakukan pengujian dan analisa hasil pembacaan sensor *accelerometer*, *gyroscope* dan modul GPS menggunakan mikrokontroler STM32F411. Penelitian ini merupakan langkah awal untuk menguji seberapa layak mikrokontroler STM32F411 digunakan sebagai *flight controller* pada drone. STM32F411 adalah salah satu tipe dari mikrokontroler STM32 yang merupakan keluarga dari 32-bit sirkuit terpadu mikrokontroler oleh STMicroelectronics yang secara spesifik dikembangkan untuk aplikasi (*Cortex-A*), *embedded* (*Cortex-M*) dan *real-time* (*Cortex-R*). Mikrokontroler ini cocok untuk melakukan pekerjaan *embedded* karena mengombinasikan performa tinggi dengan peripheral kelas satu, yang membuat kinerjanya STM32 lebih unggul dibandingkan dengan Arduino.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, maka rumusan masalah yang akan dikaji dalam laporan ini adalah sebagai berikut.

1. Bagaimana proses pengujian pembacaan sensor *accelerometer*, *gyroscope*, dan modul GPS menggunakan mikrokontroler STM32F411?
2. Bagaimana hasil analisa pembacaan sensor *accelerometer*, *gyroscope* dan modul GPS menggunakan mikrokontroler STM32F411?
3. Apakah mikrokontroler STM32F411 berpotensi dapat digunakan sebagai alternatif *flight controller* pada drone?

1.3 Tujuan Praktik Kerja Lapangan

Berdasarkan rumusan masalah yang telah diuraikan, maka tujuan dari Praktik Kerja Lapangan ini adalah sebagai berikut.

1. Mengetahui proses pengujian pembacaan sensor *accelerometer*, *gyroscope*, dan modul GPS menggunakan mikrokontroler STM32F411
2. Mengetahui hasil analisa pembacaan sensor *accelerometer*, *gyroscope*, dan modul GPS menggunakan mikrokontroler STM32F411
3. Mengetahui apakah mikrokontroler STM32F411 dapat berpotensi digunakan sebagai alternatif *flight controller* pada drone

1.4 Manfaat Praktik Kerja Lapangan

Manfaat dari dilaksanakannya kegiatan Praktik Kerja Lapangan ini dibagi menjaditiga, yaitu :

1.4.1 Bagi Mahasiswa

- a. Mahasiswa dapat menerapkan ilmu yang didapatkan selama perkuliahan pada industri atau lembaga tempat Kerja Praktik dilaksanakan.
- b. Menyiapkan diri mahasiswa dalam menghadapi dunia kerja pada masa yang akan datang.
- c. Meningkatkan pemahaman ilmu pengetahuan yang terkait dengan teknologi penerbangan.

1.4.2 Bagi Universitas Ahmad Dahlan (UAD)

- a. Sebagai sarana perkembangan IPTEK, khususnya di dunia Fisika sebagai bahan pertimbangan dalam penyusunan program bagi Universitas Ahmad Dahlan (UAD).
- b. Sebagai bahan masukan dan evaluasi program pendidikan di UAD untuk menghasilkan tenaga-tenaga terampil sesuai kebutuhan industri dan bidang masing-masing.

1.4.3 Bagi PUSTEKBANG - BRIN

- a. Sebagai wadah untuk menjalin kerja sama dengan Perguruan Tinggi, khususnya Universitas Ahmad Dahlan.
- b. Sebagai sarana penyebaran ilmu pengetahuan dan teknologi kedirgantaraan khususnya teknologi penerbangan bagi dunia pendidikan, khususnya pada kalangan mahasiswa.
- c. Sebagai sarana informasi terkait dengan kriteria tenaga kerja yang dibutuhkan di PUSTEKBANG-BRIN

1.5 Lokasi dan Waktu Praktik Kerja Lapangan

Praktik kerja lapangan dilaksanakan di Pusat Riset Teknologi Penerbangan, Badan Riset dan Inovasi Nasional (PUSTEKBANG – BRIN) yang beralamatkan di Jl. Raya LAPAN, Sukamulya, Rumpin Bogor, Jawa Barat 16350. PKL ini dilaksanakan secara *offline* selama 34 hari yaitu mulai tanggal 8 Agustus 2022 sampai dengan tanggal 10 September 2022.

BAB II

TINJAUAN INSTANSI

2.1 BADAN RISET DAN INOVASI NASIONAL (BRIN)

2.1.1 Pendahuluan

Badan Riset dan Inovasi Nasional (disingkat BRIN) adalah lembaga pemerintah non-kementerian yang berada di bawah dan bertanggung jawab kepada Presiden Indonesia melalui menteri yang membidangi urusan pemerintahan di bidang riset dan teknologi. Lembaga ini pertama kali dibentuk oleh Presiden Joko Widodo melalui Peraturan Presiden Nomor 74 Tahun 2019 yang melekat kepada Kementerian Riset dan Teknologi (Kemristek) sehingga Menteri Riset dan Teknologi juga bertindak sebagai Kepala BRIN. Saat ini, BRIN memiliki Ketua Dewan Pengarah dari BPIP yaitu Megawati Soekarnoputri. Pada 28 April 2021, berdasarkan Peraturan Presiden Nomor 33 Tahun 2021 tentang Badan Riset dan Inovasi Nasional, BRIN menjadi lembaga yang berdiri sendiri dengan mengintegrasikan Kementerian Riset dan Teknologi dan 4 (empat) lembaga pemerintahan non kementerian (LPNK) yakni Badan Pengkajian dan Penerapan Teknologi (BPPT), Badan Tenaga Nuklir Nasional (BATAN), Lembaga Penerbangan dan Antariksa Nasional (LAPAN), dan Lembaga Ilmu Pengetahuan Indonesia (LIPI). Peraturan Presiden Nomor 33 Tahun 2021 tentang Badan Riset dan Inovasi Nasional kemudian dicabut dan digantikan oleh Peraturan Presiden Nomor 78 Tahun 2021 tentang Badan Riset dan Inovasi Nasional.

2.1.2 Sejarah Singkat

Pada 5 Mei 2021, Presiden Joko Widodo menandatangani Peraturan Presiden Nomor 33 Tahun 2021, yang secara efektif menetapkan BRIN sebagai satu-satunya badan penelitian nasional, meneruskan Komite Inovasi Nasional. Peraturan tersebut memutuskan bahwa semua lembaga penelitian milik pemerintah yang meliputi Lembaga Ilmu Pengetahuan Indonesia (LIPI), Badan Pengkajian dan Penerapan Teknologi (BPPT), Badan Tenaga Nuklir Nasional (BATAN), dan Lembaga Penerbangan dan Antariksa Nasional (LAPAN) serta unit kerja yang melaksanakan tugas dan fungsi riset di lingkungan kementerian/lembaga pemerintah bergabung menjadi BRIN.

2.1.3 Visi dan Misi BRIN

a. Visi

Terwujudnya Badan Riset dan Inovasi Nasional yang andal, profesional, inovatif, dan berintegritas dalam pelayanan kepada Presiden dan Wakil Presiden, untuk

mewujudkan Visi dan Misi Presiden : “Indonesia Maju yang Berdaulat, Mandiri, dan Berkepribadian berlandaskan Gotong Royong”.

b. Misi

1. Memberikan dukungan teknis dan administrasi serta analisis yang cepat, akurat dan responsif, kepada Presiden dan Wakil Presiden dalam menyelenggarakan penelitian, pengembangan, pengkajian dan penerapan, serta invensi dan inovasi, penyelenggaraan ketenaganukliran, dan penyelenggaraan keantariksaan secara nasional yang terintegrasi serta melakukan *monitoring* pengendalian dan evaluasi terhadap pelaksanaan tugas dan fungsi BRIDA
2. Meningkatkan kualitas sumber daya manusia dan prasarana riset dan inovasi penyelenggaraan ketenaganukliran, dan keantariksaan secara nasional yang terintegrasi dan pembinaan terhadap pelaksanaan tugas dan fungsi BRIDA
3. Menyelenggarakan pelayanan yang efektif dan efisien di bidang pengawasan, administrasi umum, informasi, dan hubungan kelembagaan

2.1.4 Tugas dan Fungsi BRIN

BRIN bertugas melakukan *monitoring*, pengendalian, dan evaluasi terhadap pelaksanaan tugas dan fungsi Badan Riset dan Inovasi Daerah (BRIDA) sesuai denganketentuan peraturan perundang-undangan. Berdasarkan Perpres No. 78/2021 tentang BRIN yang ditandatangani Presiden Jokowi pada 24 Agustus 2021, pada pasal 3 BRIN mempunyai tugas membantu Presiden dalam menyelenggarakan tugas pemerintahan di bidang penelitian, pengembangan, pengkajian, dan penerapan serta invensi dan inovasi, penyelenggaraan ketenaganukliran, dan penyelenggaraan keantariksaan secara nasional yang terintegrasi.

Sementara dalam pasal 4, BRIN menyelenggarakan 14 fungsi, sebagai berikut:

1. Pelaksanaan penelitian, pengembangan, pengkajian, dan penerapan serta invensi dan inovasi dalam rangka penyusunan rekomendasi perencanaan pembangunan nasional berdasarkan hasil kajian ilmiah dengan berpedoman pada nilai Pancasila
2. Perumusan dan penetapan kebijakan di bidang riset dan inovasi yang meliputi rencana induk pemajuan ilmu pengetahuan dan teknologi, dan peta jalan penelitian, pengembangan, pengkajian, penerapan, serta invensi dan inovasi, penyelenggaraan ketenaganukliran, dan penyelenggaraan keantariksaan
3. Perumusan, penetapan, dan pelaksanaan kebijakan di bidang pembinaan, pengembangan kompetensi, pengembangan profesi, manajemen talenta, dan pengawasan dan pengendalian sumber daya manusia ilmu pengetahuan dan teknologi, infrastruktur riset dan inovasi, fasilitas riset dan inovasi pemanfaatan riset dan inovasi
4. Pengintegrasian sistem penyusunan perencanaan, program, anggaran, kelembagaan, dan sumber daya penelitian, pengembangan, pengkajian, dan

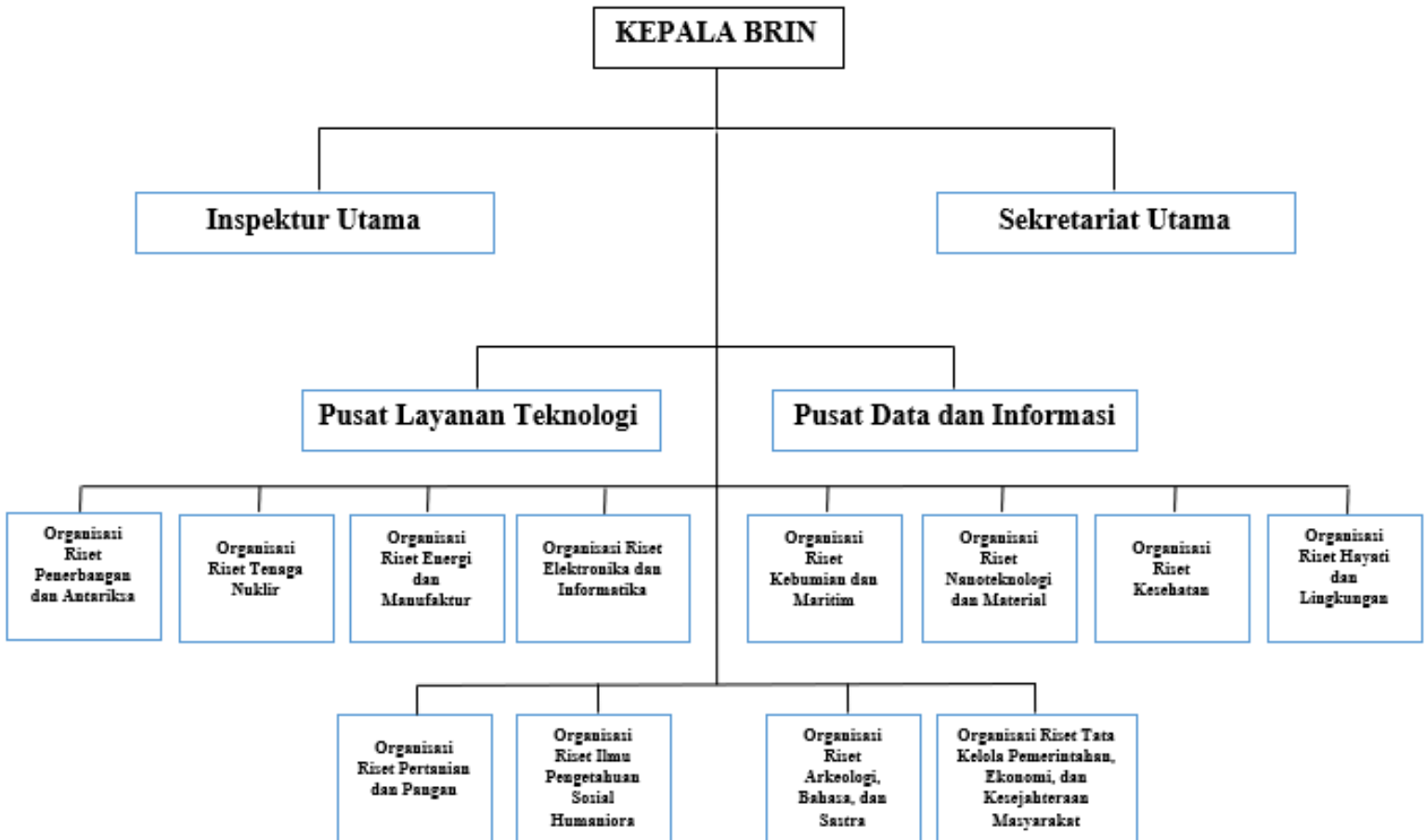
- penerapan, invensi dan inovasi, penyelenggaraan ketenaganukliran, dan penyelenggaraan keantariksaan
5. Penyelenggaraan penelitian, pengembangan, pengkajian, dan penerapan, serta invensi dan inovasi, penyelenggaraan ketenaganukliran, dan penyelenggaraan keantariksaan
 6. Pengawasan dan pengendalian penelitian, pengembangan, pengkajian, dan penerapan, serta invensi dan inovasi, penyelenggaraan ketenaganukliran, dan penyelenggaraan keantariksaan secara menyeluruh dan berkelanjutan
 7. Pelaksanaan koordinasi pengabdian kepada masyarakat berbasis penelitian, pengembangan, pengkajian, dan penerapan, serta invensi dan inovasi yang dihasilkan oleh kelembagaan ilmu pengetahuan dan teknologi
 8. Pelaksanaan pembangunan, pengelolaan, dan pengembangan sistem informasi penelitian, pengembangan, pengkajian, dan penerapan, serta invensi dan inovasi, penyelenggaraan ketenaganukliran, dan penyelenggaraan keantariksaan
 9. Pelaksanaan penelitian, pengembangan, invensi, dan inovasi kebijakan yang mengakui, menghormati, mengembangkan dan melestarikan keanekaragaman pengetahuan tradisional, kearifan lokal, sumber daya alam hayati dan nirhayati, serta budaya sebagai bagian dari identitas bangsa
 10. Pemberian fasilitasi, bimbingan teknis, pembinaan, dan supervisi serta pemantauan dan evaluasi di bidang penelitian, pengembangan, pengkajian, dan penerapan, serta invensi dan inovasi, penyelenggaraan ketenaganukliran, dan penyelenggaraan keantariksaan
 11. Pemantauan, pengendalian, dan evaluasi terhadap pelaksanaan tugas dan fungsi BRIDA
 12. Pembinaan dan pemberian dukungan administrasi dan teknis kepada seluruh unsur organisasi di lingkungan BRIN
 13. Pengawasan atas pelaksanaan tugas di lingkungan BRIN
 14. Pelaksanaan fungsi lain yang diberikan oleh Presiden.

2.1.5 Struktur Organisasi dan Tata Kerja BRIN

Struktur organisasi dan tata kerja (SOTK) BRIN diatur dalam Peraturan Badan Riset dan Inovasi Nasional Republik Indonesia Nomor 1 Tahun 2021. Menurut peraturan tersebut, SOTK BRIN terdiri dari jabatan-jabatan berikut.

1. Kepala BRIN
2. Wakil Kepala BRIN
3. Sekretariat Utama
4. Inspektorat Utama
5. Deputi Bidang Kebijakan Pembangunan
6. Deputi Bidang Kebijakan Riset dan Inovasi
7. Deputi Bidang Sumber Daya Manusia Ilmu Pengetahuan dan Teknologi
8. Deputi Bidang Infrastruktur Riset dan Inovasi

9. Deputi Bidang Fasilitasi Riset dan Inovasi
10. Deputi Bidang Pemanfaatan Riset dan Inovasi
11. Deputi Bidang Riset dan Inovasi Daerah
12. Inspektorat Utama
13. Organisasi Riset



Gambar 2.1. Bagan Struktur Organisasi BRIN

(Sumber : www.brin.go.id)

Struktur Organisasi BRIN seperti yang ditunjukkan pada **Gambar 2.1** merupakan hasil pengintegrasian empat lembaga riset, yakni Lembaga Ilmu Pengetahuan Indonesia (LIPI), Badan Pengkajian dan Penerapan Teknologi (BPPT), Badan Tenaga Nuklir Nasional (Batn), dan Lembaga Penerbangan dan Antariksa Nasional (LAPAN). Adapun pelaksanaan Praktik Kerja Lapangan dalam laporan ini dilaksanakan di bagian organisasi riset bidang penerbangan dan antariksa yang dulu merupakan bagian dari LAPAN. Organisasi riset penerbangan dan antariksa terbagi lagi menjadi beberapa pusat riset, salah satunya yaitu pusat riset teknologi penerbangan yang menjadi lokasi utama proses pengujian dan pengambilan data.

2.1.6 Organisasi Riset BRIN

Organisasi Riset (OR) merupakan organisasi non-struktural yang menyelenggarakan teknis penelitian, pengembangan, pengkajian, dan penerapan, serta invensi dan inovasi, penyelenggaraan ketenaganukliran, dan/atau penyelenggaraan keantariksaan. Peran yang dilaksanakan oleh OR adalah seperti peran kampus di Kemendikbudristek. Jika kampus menjadi pelaksana fungsi pendidikan di Kemendikbudristek, maka OR menjadi pelaksana fungsi riset di BRIN. Pada tahun 2022 terdapat 12 Organisasi Riset di BRIN berdasarkan bidang keilmuan, yaitu:

1. Organisasi Riset Tenaga Nuklir
2. Organisasi Riset Kebumihan dan Maritim
3. Organisasi Riset Ilmu Pengetahuan Sosial dan Humaniora
4. Organisasi Riset Penerbangan dan Antariksa
5. Organisasi Riset Arkeologi, Bahasa, dan Sastra
6. Organisasi Riset Tata Kelola Pemerintahan, Ekonomi, dan Kesejahteraan Masyarakat
7. Organisasi Riset Pertanian dan Pangan
8. Organisasi Riset Hayati dan Lingkungan
9. Organisasi Riset Kesehatan
10. Organisasi Riset Energi dan Manufaktur
11. Organisasi Riset Elektronika dan Informatika
12. Organisasi Riset Nanoteknologi dan Material

2.1.7 Logo BRIN

Meleburnya beberapa lembaga riset yakni Lembaga Ilmu Pengetahuan Indonesia (LIPI), Badan Pengkajian dan Penerapan Teknologi (BPPT), Badan Tenaga Nuklir Nasional (BATAN), serta Lembaga Penerbangan dan Antariksa Nasional (LAPAN) menjadi Badan Riset dan Inovasi Nasional (BRIN) menjadi latar belakang diluncurkannya logo BRIN yang secara resmi diunggah pada tanggal 10 Agustus 2021. Seperti pada **Gambar 2.2**, logo BRIN yang berwarna merah menggambarkan BRIN sebagai institusi berlandaskan IPTEK yang bisa memandu Indonesia bergerak maju. Adapun bentuk logo BRIN merupakan vektor yang menggambarkan aspek yang diusung BRIN. Yakni vektor logo manusia, bintang yang menggambarkan angkasa dan ilmu pengetahuan, lingkaran yang menggambarkan *biodiversity*, sirip yang menggambarkan fauna dan daun yang menggambarkan flora.



Gambar 2.2. Logo BRIN
(sumber : www.brin.go.id)

2.2 PUSAT RISET TEKNOLOGI PENERBANGAN (PUSTEKBANG)

Pusat Riset Teknologi Penerbangan (PUSTEKBANG) adalah salah satu fasilitas dan pusat riset yang dimiliki oleh BRIN. Kedudukannya dibawah Deputi Bidang Teknologi Penerbangan dan Antariksa, dan dipimpin oleh seorang kepala yang bertanggungjawab kepada Deputi Bidang Teknologi Penerbangan dan Antariksa.

2.2.1 Visi dan Misi

a. Visi

Pusat unggulan Penerbangan dan Antariksa untuk mewujudkan Indonesia yang maju dan mandiri

b. Misi

1. Meningkatkan kualitas litbang penerbangan dan antariksa bertaraf internasional.
2. Meningkatkan kualitas produk teknologi dan informasi di bidang penerbangan dan antariksa dalam memecahkan permasalahan nasional.
3. Melaksanakan dan mengatur penyelenggaraan keantariksaan untuk kepentingan nasional

2.2.2 Tugas dan Fungsi

Berdasarkan Peraturan Kepala LAPAN Nomor 8 Tahun 2015 tentang Organisasi dan Tata Kerja LAPAN, Pasal 86, Pusat Teknologi Penerbangan mempunyai tugas pokok: melaksanakan penelitian, pengembangan, perekayasaan dan pemanfaatan serta penyelenggaraan keantariksaan di bidang teknologi aeronautika.

Adapun fungsi – fungsinya meliputi:

- a.** Penyusunan rencana, program, kegiatan, dan anggaran di bidang teknologi

aeronautika.

- b. Penyiapan bahan rumusan kebijakan teknis di bidang teknologi aeronautika.
- c. Penyusunan dan pelaksanaan program nasional penguasaan dan pengembangan teknologi aeronautika.
- d. Penelitian, pengembangan, dan perekayasaan teknologi aeronautika.
- e. Pengelolaan fasilitas penelitian, pengembangan, perekayasaan, dan pemanfaatan di bidang teknologi aeronautika.
- f. Pelaksanaan kegiatan diseminasi hasil penelitian, pengembangan, perekayasaan, dan pemanfaatan di bidang teknologi aeronautika
- g. Pelaksanaan kegiatan penjalaran teknologi di bidang teknologi aeronautika
- h. Pembinaan dan pemberian bimbingan di bidang penelitian, pengembangan, perekayasaan, dan pemanfaatan teknologi aeronautika.
- i. Pelaksanaan kerja sama teknis di bidang teknologi aeronautika.
- j. Pelaksanaan alih teknologi di bidang teknologi aeronautika
- k. Pelaksanaan administrasi keuangan, penatausahaan barang milik negara, pengelolaan rumah tangga, sumber daya manusia aparatur, dan tata usaha pusat.

2.2.3 Struktur Organisasi PUSTEKBANG

Pusat Riset Teknologi Penerbangan terdiri atas :

- a. Bagian Administrasi
- b. Bagian Program dan Fasilitas
- c. Bagian Diseminasi
- d. Bagian Aerodinamika
- e. Bagian Aerostruktur
- f. Bagian Propulsi
- g. Bagian Avionik

2.2.4 Lokasi PUSTEKBANG

Pusat Riset Teknologi penerbangan Bertempat di Jl. Raya Lapan, Sukamulya, Rumpin. RT 7 / RW 4 Sukamulya. Sukamulya. Kec. Rumpin, Kab. Bogor, Prov. Jawa Barat. 16350. Telepon: (021) 75790083. Email : pustekbang@lapan.go.id

BAB III

TINJAUAN PUSTAKA

3.1 Drone

3.1.1 Pengertian Drone

Drone merupakan salah satu jenis pesawat tanpa pilot atau biasa dikenal dengan *Unmanned Aerial Vehicle* (UAV). Seperti pada **Gambar 3.1**, pesawat ini dikendalikan secara otomatis melalui program komputer yang dirancang, atau melalui kendali jarakjauh dari pilot yang terdapat di dataran atau di kendaraan lainnya. Awalnya UAV merupakan pesawat yang dikendalikan jarak jauh, namun sistem otomatis kini mulai banyak diterapkan. Perkembangan teknologi membuat drone juga mulai banyak diterapkan untuk kebutuhan sipil, terutama di bidang bisnis, industri dan logistik.

Dunia industri bisnis, drone telah diterapkan dalam berbagai layanan seperti pengawasan Infrastruktur, pengiriman paket barang, pemadam kebakaran hutan, eksplorasi bahan tambang, pemetaan daerah pertanian, dan pemetaan daerah industri. Meski alat canggih ini pada awalnya hanya di gunakan oleh anggota militer saja, kini alat ini telah banyak digunakan oleh seluruh pihak secara meluas. Badan pemerintahan juga memanfaatkan alat canggih ini untuk dapat menghubungkan intelijen dengan pertanian. Namun saat ini untuk masyarakat awam sekalipun juga sudah dapat menggunakan drone (Indreswari, 2016).



Gambar 3.1. Drone untuk fotografi
(Sumber : <https://eraspace.com/>)

3.1.2 Jenis – jenis Drone

Berdasarkan jenisnya, terdapat dua jenis drone, yaitu *multicopter* dan *fixed wing*. *Fixed Wing* memiliki bentuk seperti pesawat terbang biasa yang dilengkapi sistem sayap seperti pada **Gambar 3.2**. Tipe *fixed-wing* memerlukan desain aerodinamika

pada sayap dan badannya sehingga perancangannya cukup rumit. *Multicopter* yaitu jenis drone yang memanfaatkan putaran baling-baling untuk terbang seperti pada **Gambar 3.3**. *Multicopter* dibagi menjadi dua yaitu *single-rotor* dan *multi-rotor*. Tipe *single-rotor* berbentuk seperti helikopter menggunakan baling-baling tunggal, sedangkan *multi-rotor* menggunakan 3 sampai 8 baling-baling (Indreswari, 2016).



Gambar 3.2. Drone *fixed wing* (Indreswari, 2016)



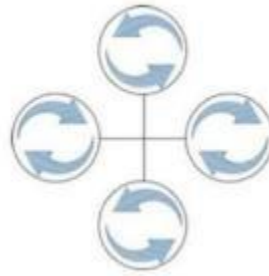
Gambar 3.3. Drone *Multicopter*

(Sumber : <http://www.library.upnvj.ac.id/pdf/2s1teknikinformatika/bab2.pdf>)

3.1.3 Prinsip Kerja Drone

Drone terutama jenis *multirotor* ketika beroperasi menggunakan empat buah rotor dan empat buah baling-baling. Dua rotor berputar melawan arah jarum jam dan dua lainnya searah jarum jam. Karena kombinasi ini lah sistem mampu mengendalikan gaya angkat pesawat, melalui kondisi torsi yang seimbang. Sedangkan ketika dilihat dari segi “*vertical take off and landing aircraft (VTOL)*” *multirotor* mampu melakukan *pitch*, *yaw*, dan *roll* menggunakan kontrol kendali empat rotor.

Quadcopter memiliki ketentuan tersendiri terhadap variasi kecepatan perputaran setiap baling-balingnya untuk dapat melakukan pergerakan tertentu. Dalam melakukan terbang melayang di udara, yang dikenal dengan sebutan *hover*, *quadcopter* perlu menggerakkan keempat baling-balingnya dengan kecepatan yang sama. Pada **Gambar 3.4** menunjukkan setiap pasangan baling-baling memiliki arah gaya dorong yang memiliki fungsi berbeda, satu pasang sebagai pendorong (*Pusher*) dan satu pasang sebagai penarik (*Puller*). Dengan adanya pergerakan yang menghasilkan gaya dorong yang melawan gaya gravitasi tersebut, *quadcopter* dapat terbang melayang di udara dengan stabil (Munir hutama, 2015).



Gambar 3.4. Prinsip kerja *propeller multicopter* (Munir Hutama, 2015).

Gerakan dasar pada *quadcopter* terdiri dari empat gerakan, yaitu; *Roll* (bergerak menyamping), *Pitch* (bergerak maju mundur), *Yaw* (bergerak secara berotasi) dan *Throttle* (bergerak ke atas *landing* dan *take off*).

1. *Throttle*

Throttle merupakan penaikan kecepatan secara keseluruhan pada motor, di mana ketika motor *full throttle* maka *quadcopter* akan terjadi momen angkat pada *body quadcopter*. dan begitu sebaliknya jika motor menurunkan kecepatan maka akan terjadi posisi turun.

2. *Pitch*

Pitch merupakan pergerakan maju dan mundur pada *quadcopter*, di mana dua buah motor yang berada di posisi belakang berputar lebih cepat dibanding dengan motor yang berada di depan sehingga terjadi kemiringan ke arah depan dan akan terdorong ke depan dan akan terjadi gerakan maju, begitupun sebaliknya untuk bergerak mundur.

3. *Roll*

Roll merupakan gerakan menyamping, baik ke arah kiri atau kanan, untuk dapat bergerak menyamping perlu adanya perubahan kecepatan pada satu pasang motor, misal *quadcopter* akan gerak ke samping kanan maka motor sebelah kiri menaikkan kecepatan sehingga terjadi momen miring dan secara tidak langsung bergerak ke samping kanan begitu pun sebaliknya.

4. *Yaw*

Pergerakan ini yaitu di mana *quadcopter* berputar dalam satu poros dilakukan dengan cara menurunkan kecepatan sepasang motor, atas-bawah atau kiri - kanan, dan menaikkan kecepatan satu pasangan motor yang sedikit lambat kecepatannya dibandingkan pasangan motor yang lain (Achlisson, 2021).

3.2 *Flight Controller*

Flight Controller (FC) adalah kumpulan dari berbagai komponen-komponen dan sensor yang berfungsi untuk menjaga drone atau pesawat RC tetap seimbang dan dapat terkendali. Penggunaan teknologi *flight controller* saat ini sudah sangat maju dan mudah diakses datanya sehingga memungkinkan teknologi cepat berkembang di semua kalangan (Setiawan, dkk, 2017). Penggunaan *flight controller* mutlak adanya pada *multicopter*, sedangkan pada *fixed-wing* menjadi kebutuhan yang opsional karena masih dapat dikendalikan secara manual tanpa adanya bantuan *flight controller*. Banyak produsen *flight controller* bermunculan seperti Pixhawk, APM dan lainnya yang kerap digunakan dalam aktivitas pemetaan lahan oleh beberapa perusahaan, *flight controller* yang merupakan produk manufaktur ini memiliki fungsi yang lebih kompleks karena sudah dikembangkan oleh banyak orang yang umumnya digunakan pada dunia pemetaan suatu wilayah melalui udara, sehingga memiliki kemampuan untuk terbang secara auto berdasarkan jalur misi yang sudah dibuat sebelumnya dan melakukan pemotretan gambar berkala untuk kemudian hasil dari misi tersebut akan disatukan menjadi sebuah peta yang dapat disajikan untuk kebutuhan selanjutnya.



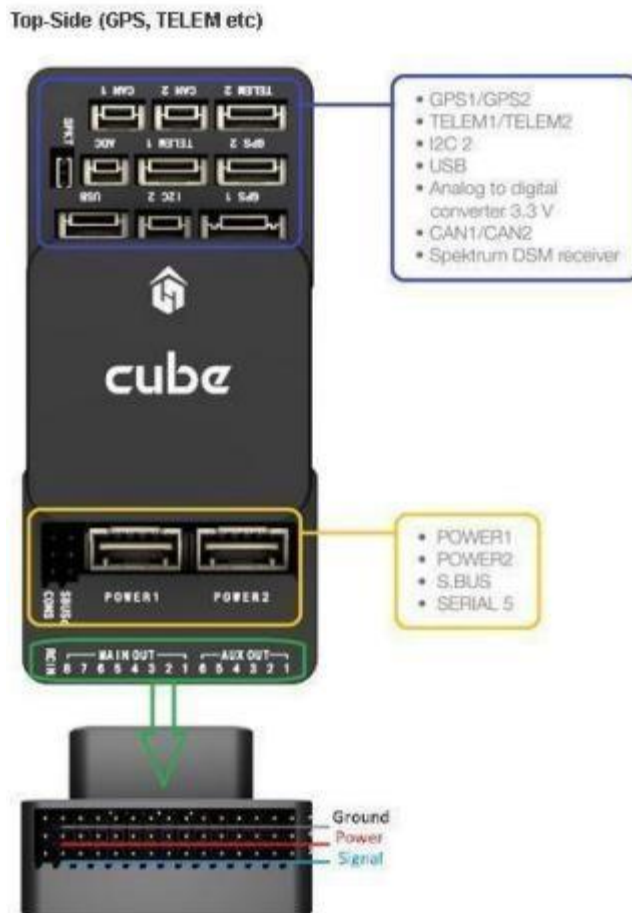
Gambar 3.5. Pixhawk *Cube*

(Sumber : <https://www.doc.px4.io>)

Flight controller bertugas mengumpulkan dan mengelola informasi ketinggian, kecepatan, arah, kemiringan dan mengusahakan agar drone tetap dalam keadaan

yang sesuai dengan yang diharapkan. Adapun jenis sensor umum yang ada dalam sebuah *flight controller* adalah sebagai berikut :

- *Accellerometer*, sensor pendeteksi kecepatan dan perubahan posisi
- *Magnetometer*, sensor pendeteksi ke arah mana *flight controller* menghadap (kompas)
- *Gyroscope*, sensor pendeteksi perubahan arah gerak
- *Barometer*, sensor pendeteksi perubahan tekanan udara (penentu ketinggian), dan biasanya juga sensor GPS.



Gambar 3.6. Konfigurasi Pixhawk *Cube*
(sumber : ardupilot.org)

3.3 Mikrokontroler STM32F411

STM32F411 adalah salah satu tipe dari mikrokontroler STM32 yang merupakan keluarga dari 32-bit sirkuit terpadu mikrokontroler oleh *STMicroelectronics* yang secara spesifik dikembangkan untuk aplikasi (*Cortex-A*), *embedded* (*Cortex-M*) dan *real-time* (*Cortex-R*). Mikrokontroler ini cocok untuk melakukan pekerjaan

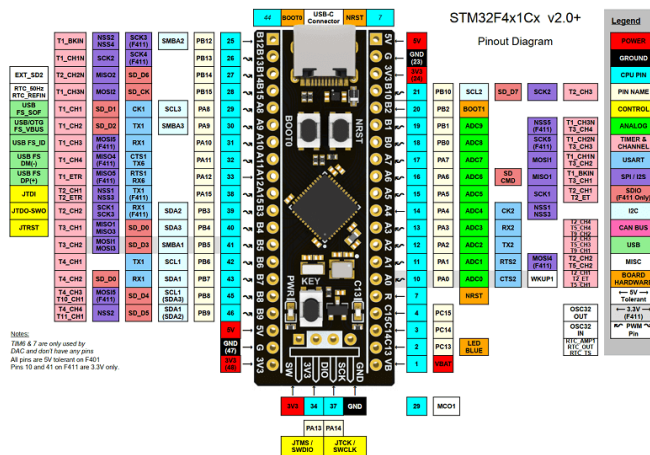
embedded karena mengombinasikan performa tinggi dengan peripheral kelas satu, yang membuat kinerjanya STM32 lebih unggul dibandingkan dengan Arduino.

Mikrokontroler STM32F411 merupakan bagian dari lini STM32 *Dynamic Efficiency*. Perangkat ini adalah level awal seri F4 dengan performa tinggi dan menawarkan keseimbangan terbaik antara konsumsi daya dinamis (dalam mode berjalan) dan performa pemrosesan, sekaligus mengintegrasikan sejumlah besar fitur bernilai tambah dalam paket berukuran kecil 3 x 3 mm.

Dengan *Batch Acquisition Mode* (BAM) baru, yang mengoptimalkan konsumsi daya untuk pengelompokan data, STM32F411 membawa efisiensi dinamis ke tingkat yang baru. BAM ini memungkinkan pertukaran kumpulan data melalui komunikasi periferil dengan perangkat lain (termasuk CPU) dalam mode hemat daya.

Berikut spesifikasi dari Mikrokontroler STM32F411 :

- *Memory Flash* 512 KB
- RAM 128 KB
- Sumber tegangan : melalui USB atau dari sumber tegangan eksternal 5V
- 3x USART berjalan hingga 12,5 Mbit/dtk
- 5x SPI (dengan I2S) berjalan hingga 50 Mbit/dtk
- 3x I²C hingga 1Mbps
- 1x USB 2.0 OTG kecepatan penuh
- 2x full duplex I²S hingga 32-bit/192KHz
- 3x simpleks I²S hingga 32-bit/192KHz
- ADC 12-bit mencapai 2,4 MSPS
- 11 timer, 16- dan 32-bit, beroperasi hingga 100 MHz



Gambar 3.7. Konfigurasi Pin STM32F411
(Sumber : <https://www.dfrobot.com/>)

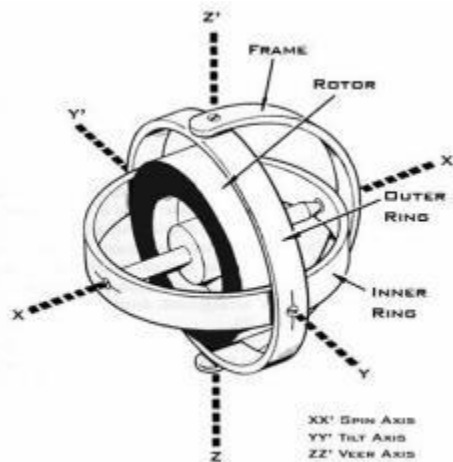
3.4 Sensor Accelerometer

Accelerometer adalah alat yang digunakan untuk mengukur percepatan, mendeteksi dan mengukur getaran (vibrasi), dan mengukur percepatan akibat gravitasi (inklinasi). *Accelerometer* dapat digunakan untuk mengukur getaran pada mobil, mesin, bangunan, dan instalasi pengamanan.

Accelerometer juga dapat diaplikasikan pada pengukuran aktivitas gempa bumi dan peralatan-peralatan elektronik, seperti permainan 3 dimensi, *mouse* komputer, dan telepon. Untuk aplikasi yang lebih lanjut, sensor ini banyak digunakan untuk keperluan navigasi. Percepatan merupakan suatu keadaan berubahnya kecepatan terhadap waktu. Bertambahnya suatu kecepatan dalam suatu rentang waktu disebut percepatan (*acceleration*). Namun jika kecepatan semakin berkurang daripada kecepatan sebelumnya, disebut perlambatan (*deceleration*). Percepatan juga bergantung pada arah/orientasi karena merupakan penurunan kecepatan yang merupakan besaran vektor. Berubahnya arah pergerakan suatu benda akan menimbulkan percepatan pula. Untuk memperoleh data jarak dari sensor *accelerometer*, diperlukan proses integral ganda terhadap keluaran sensor (Riyadi, dkk , 2011).

3.5 Sensor Gyroscope

Gyroscope atau *Gyro* adalah perangkat untuk mengukur atau mempertahankan orientasi, dengan prinsip ketetapan momentum sudut. Mekanismenya adalah sebuah roda berputar dengan piringan didalamnya yang tetap stabil. *Gyroscope* sering digunakan pada robot atau heli dan alat-alat canggih lainnya. *Gyroscope* adalah berupa sensor *gyro* seperti pada **Gambar 3.8** yang berfungsi untuk menentukan orientasi gerak dengan bertumpu padaroda atau cakram yang berotasi dengan cepat pada sumbu.



Gambar 3.8. Prinsip Kerja *Gyroscope*

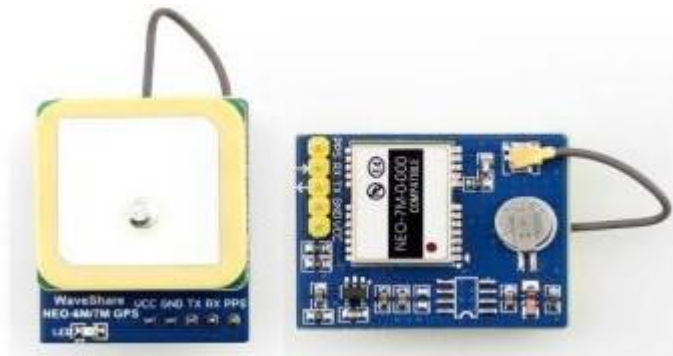
(Sumber:http://www.dutchops.com/Portfolio_Marcel/Articles/Instruments/Gyroscopic_Instruments/Theory_Gyroscopes.htm)

Gyro sensor bisa mendeteksi gerakan sesuai gravitasi, atau dengan kata lain mendeteksi gerakan pengguna. Sebelum digunakan, sensor *gyroscope* terlebih dahuludilakukan proses kalibrasi dengan menggunakan bandul. Proses kalibrasi tersebut berfungsi untuk memperoleh nilai faktor kalibrasi. *Gyroscope* memiliki keluaran berupakecepatan sudut dari arah 3 sumbu yaitu: sumbu x yang nantinya akan menjadi sudut *phi* (kanan dan kiri), sumbu y nantinya menjadi sudut *theta* (atas dan bawah), dan sumbu z nantinya menjadi sudut *psi* (depan dan belakang).

3.6 Modul GPS

Modul GPS berukuran 25x35 mm dan memiliki antena dengan ukuran 25x25 mm berfungsi sebagai penerima *GPS (Global Positioning System Receiver)* yang dapat mendeteksi lokasi dengan menangkap dan memroses sinyal dari satelit navigasi. Aplikasi dari modul ini melingkupi sistem navigasi, sistem keamanan terhadap kemalingan pada kendaraan / perangkat bergerak, akuisisi data pada sistem pemetaan medan, penjejak lokasi */location tracking*, dsb.

Modul ini kompatibel dengan APM2 dan APM2.5 dengan EEPROM terpadu yang dapat digunakan untuk menyimpan data konfigurasi. Antarmuka menggunakan serial TTL (RX/TX) yang dapat diakses dari mikrokontroler yang memiliki fungsi UART atau emulasi serial TTL (pada Arduino dapat menggunakan pustaka komunikasi serial / *serial communication library* yang sudah tersedia dalam paket Arduino IDE). *Baud rate* di set secara *default* di 9600 bps.



Gambar 3.9. Modul GPS tipe NEO-7M
(Sumber: <https://www.nyebarilmu.com/>)

BAB IV METODOLOGI

4.1 Tempat dan Waktu Pelaksanaan

Kegiatan Praktik Kerja Lapangan ini berlangsung selama 5 Minggu terhitung mulai tanggal 08 Agustus - 10 September 2022, dimana satu minggu pertama digunakan untuk studi literatur dan mencari referensi tentang tema yang akan dikerjakan yaitu tentang mikrokontroler STM32F411, Minggu kedua digunakan untuk mempelajari berbagai sensor yang akan digunakan yaitu mempelajari prinsip kerja sensor, tipe – tipenya serta codingan sensor agar terhubung ke mikrokontroler STM32F411 dan dapat dilakukan pembacaan data. Di minggu ketiga digunakan untuk melakukan perancangan sistem *hardware* berupa rangkaian sensor ke mikrokontroler dan perancangan sistem *software* berupa codingan program di Arduino IDE. Di minggu keempat pelaksanaan, digunakan untuk melakukan pengujian dan pengambilan data, dan di minggu kelima digunakan untuk evaluasi dan penyusunan laporan. Pelaksanaan ini dilakukan di Gedung Avionik di Lembaga Pusat Teknologi Penerbangan dan Antariksa Nasional (PUSTEKBANG-LAPAN) Rumpin, Kabupaten Bogor dan untuk pengambilan data dilakukan setelah proses pengujian dan perancangan *hardware* dan *software*. Diagram blok pelaksanaan Praktik Kerja Lapangan ditunjukkan pada **Gambar 4.1**.

4.2 Alat dan Bahan

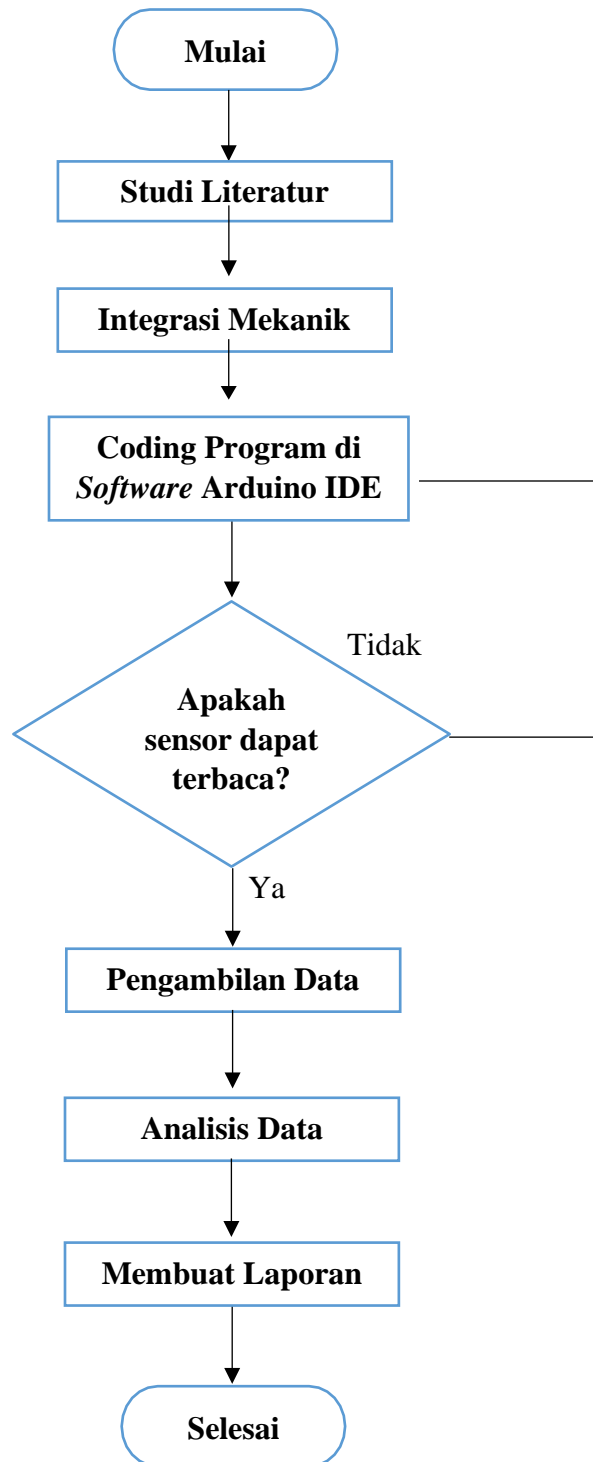
Adapun alat dan bahan yang digunakan dalam Praktik Kerja Lapangan (PKL) yaitu:

- a. Mikrokontroler STM32F411
- b. ST-Link V2 *driver*
- c. Sensor *Accelerometer* ADXL345
- d. Sensor *Gyroscope* GY-25
- e. Sensor IMU GY-521(MPU6050)
- f. Modul GPS Neo-7M
- g. *Breadboard*
- h. Kabel *Jumper*
- i. Kabel USB Type-C
- j. Software Arduino IDE

4.3 Metode Pelaksanaan Praktik Kerja Lapangan

Metode yang digunakan dalam praktik kerja lapangan ini berupa metode pembacaan hasil sensor dan analisa. Proses ini dimulai dengan integrasi mekanik berupa perancangan sistem *hardware* atau dan perancangan sistem *software* antara komponen

sensor dan mikrokontroler STM32F411. Setelah proses perancangan, dilakukan proses pembacaan hasil sensor dan analisa hasil yang didapat. Adapun metode pelaksanaan penelitian ditunjukkan pada diagram alir **Gambar 4.1**:

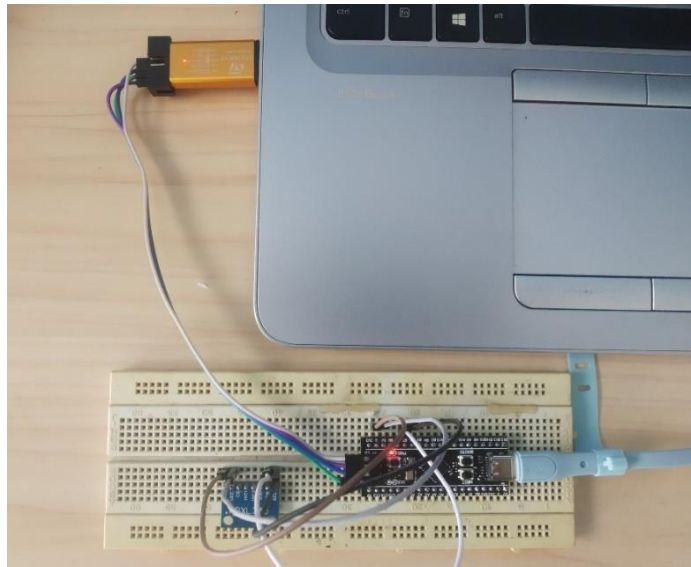


Gambar 4.1. Diagram alir metode Praktik Kerja Lapangan

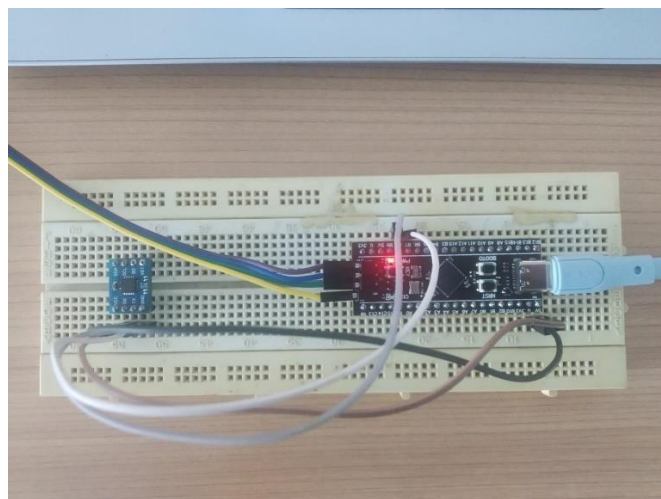
4.4 Metode Integrasi Sistem

4.4.1 Integrasi *Hardware* Rangkaian

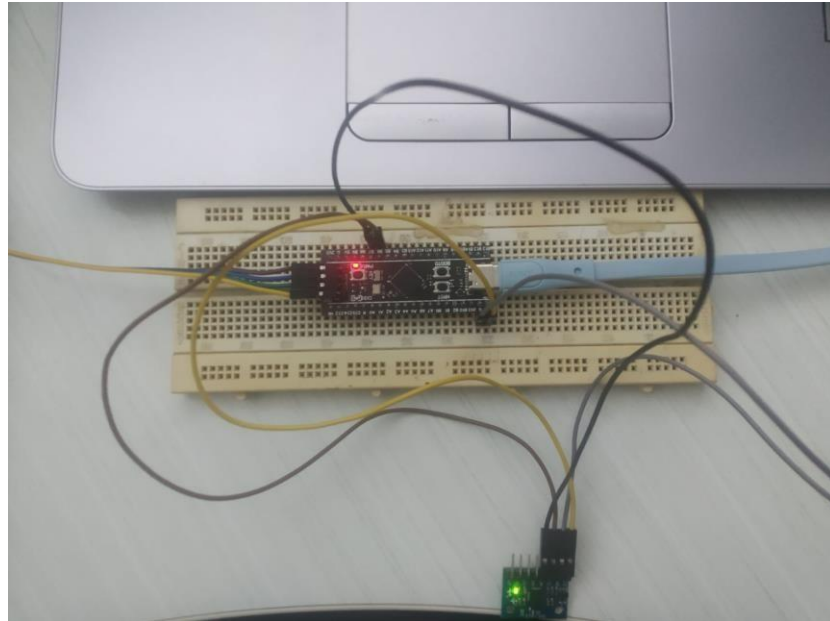
Pada integrasi *hardware* ini dilakukan proses perangkaian antara mikrokontroler STM32F411 dengan komponen sensor yang akan di baca hasilnya. Perangkaian ini dilakukan menggunakan komponen *breadboard* yang dihubungkan dengan kabel *jumper*. Adapun rangkaian mekanik antara mikrokontroler dan sensor ditunjukkan pada **Gambar 4.2**, **Gambar 4.3**, **Gambar 4.4**, dan **Gambar 4.5**.



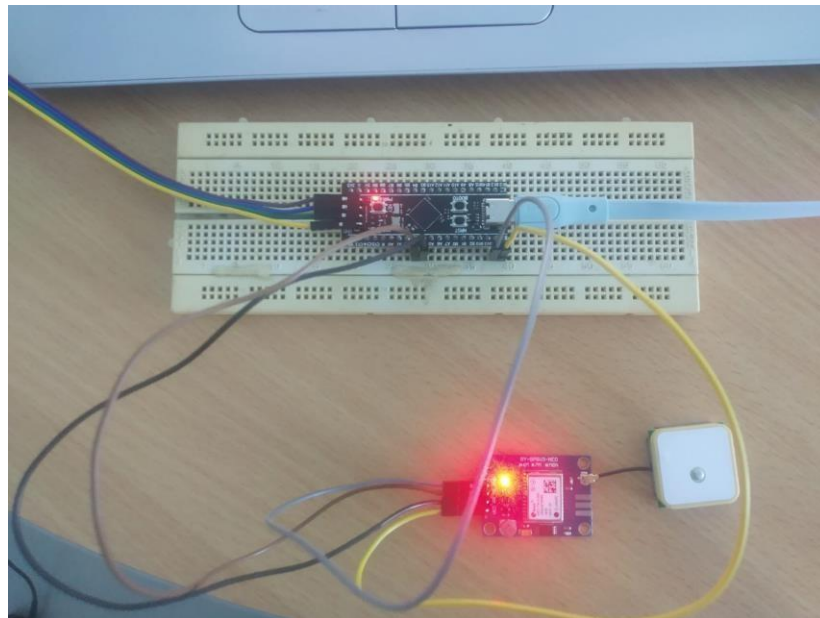
Gambar 4.2. Rangkaian Sensor *Accelerometer* ADXL345



Gambar 4.3. Rangkaian Sensor *Gyroscope* GY-25



Gambar 4.4. Rangkaian Sensor IMU GY-521(MPU6050)

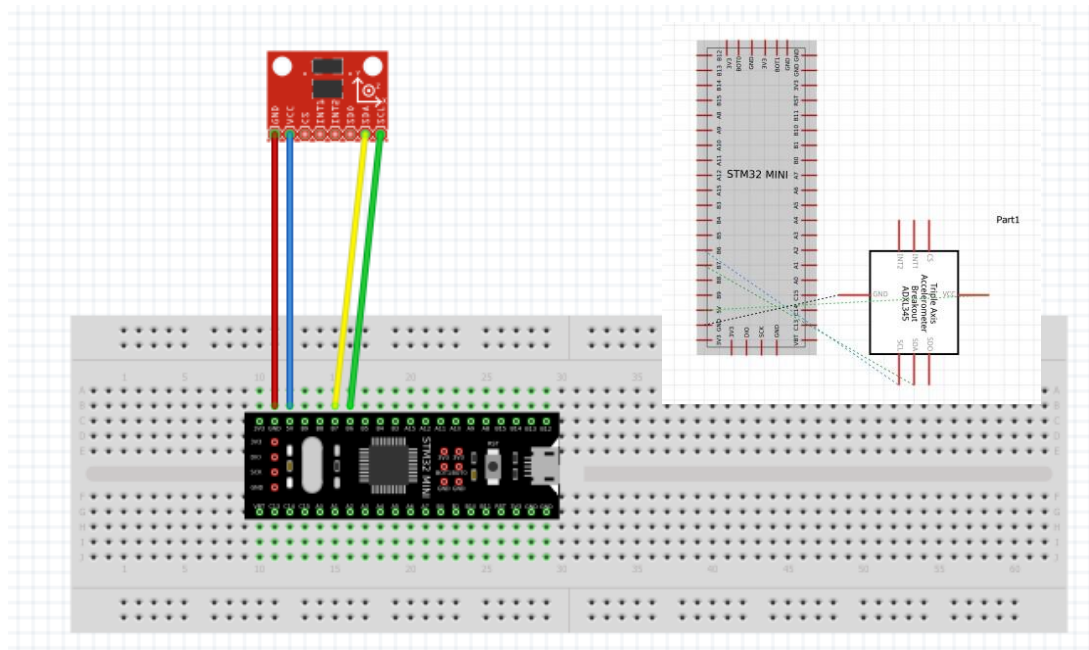


Gambar 4.5. Rangkaian Modul GPS Neo-7m

4.4.2 Integrasi *Software* dan Konfigurasi Pin

Pada proses integrasi *software*, digunakan aplikasi Arduino IDE dan *software* Fritzing untuk membantu proses perangkaian dan pengambilan data pada penelitian ini. Selain itu, agar hasil pembacaan data akurat dan optimal, proses perangkaian komponen didasarkan pada konfigurasi pin out antara sensor – sensor dan mikrokontroler STM32F411. Proses tersebut di jabarkan sebagai berikut :

- a. Rangkaian dan Konfigurasi Pin Sensor *Accelerometer* ADXL345



Gambar 4.6. Skematik Rangkaian Sensor *Accelerometer* ADXL345 di *Software* Fritzing

Tabel 4.1. Konfigurasi Pin Sensor *Accelerometer* ADXL345 dan Mikrokontroler STM32F411

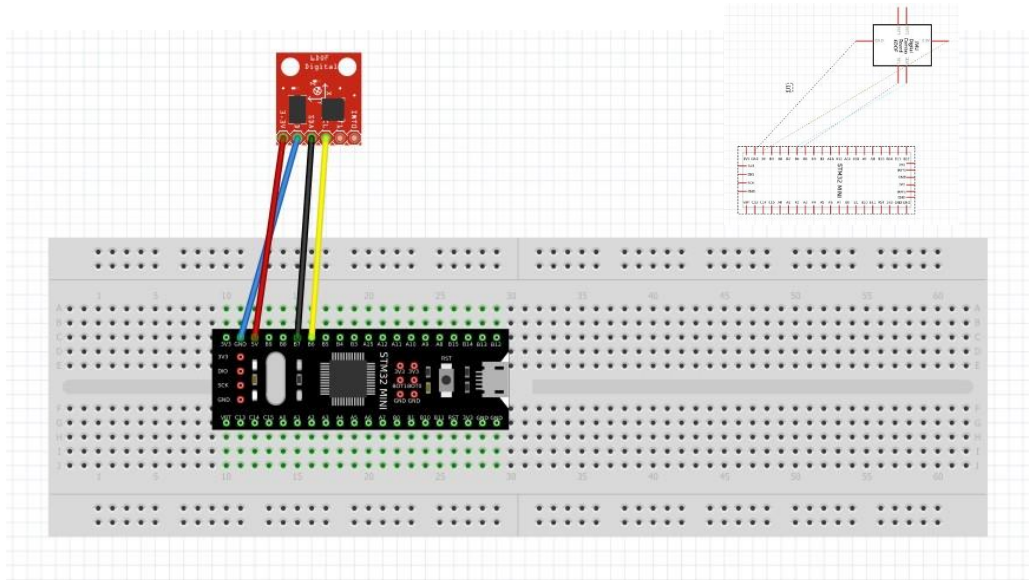
ADXL345	STM32F411
G	G
Vcc	5v
SDA	B7
SCL	B6

Pada **Gambar 4.6** dan **Tabel 4.1** menunjukkan skematik rangkaian dan konfigurasi pin out antara sensor *Accelerometer ADXL345* dan Mikrokontroler *STM32F411*. Sirkuit untuk sensor *Accelerometer ADXL345* cukup mudah dan tidak memerlukan komponen atau kabel yang rumit agar dapat terhubung ke Mikrokontroler *STM32F411*. Langkah – langkah untuk merangkai rangkaiannya sebagai berikut:

- Menyambungkan pin GND dari ADXL345 ke Pin GND di STM32F411
- Menyambungkan pin VCC dari ADXL345 ke Pin 5v di STM32F411
- Menyambungkan pin SDA dari ADXL345 ke Pin B7 di STM32F411
- Menyambungkan pin SCL dari ADXL345 ke Pin B6 di di STM32F411

Pin GND merupakan pin *ground* untuk catu daya digital, pin VCC merupakan pin yang berfungsi sebagai pin masukan catu daya digital. 2. Sedangkan pin SDA dan SCL merupakan pin yang berfungsi sebagai antarmuka digital yang menggunakan komunikasi I2C atau SPI.

b. Rangkaian dan Konfigurasi Pin Sensor *Gyroscope GY-25*



Gambar 4.7. Skematik Rangkaian Sensor *Gyroscope GY-25* di *Software Fritzing*

Tabel 4.2. Konfigurasi Pin Sensor *Gyroscope GY-25* dan Mikrokontroler *STM32F411*

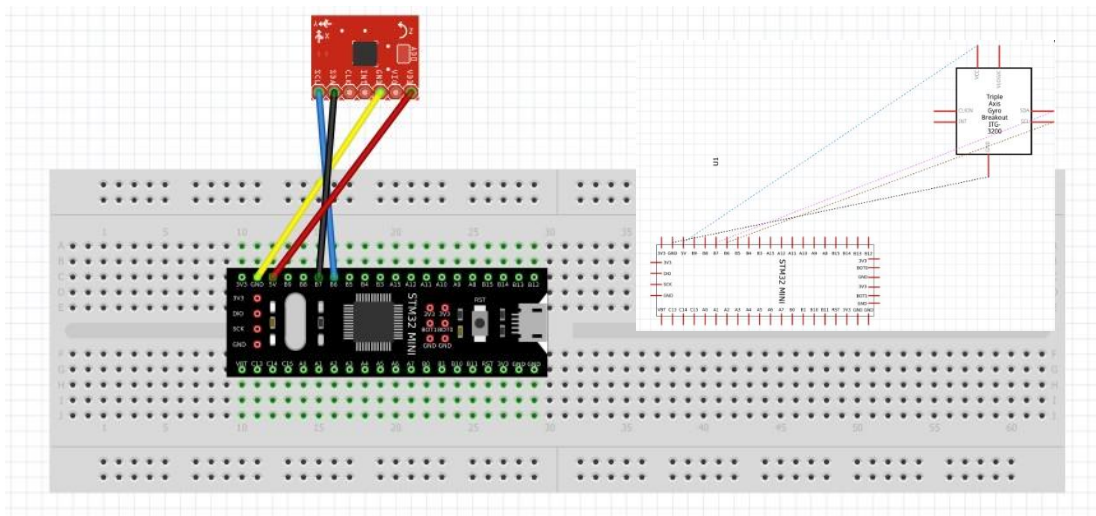
GY-25	STM32F411
G	G
Vcc	5v
SDA	B7
SCL	B6

Pada **Gambar 4.7** dan **Tabel 4.2** menunjukkan skematik rangkaian dan konfigurasi pin out antara sensor *Gyroscope* GY-25 dan Mikrokontroler STM32F411. Langkah – langkah untuk merangkai rangkaiannya sebagai berikut:

- Menyambungkan pin GND dari GY-25 ke Pin GND di STM32F411
- Menyambungkan pin VCC dari GY-25 ke Pin 5v di STM32F411
- Menyambungkan pin SDA dari GY-25 ke Pin B7 di STM32F411
- Menyambungkan pin SCL dari GY-25 ke Pin B6 di di STM32F411

Pin GND merupakan pin *ground* untuk catu daya digital, pin VCC merupakan pin yang berfungsi sebagai pin masukan catu daya digital. 2. Sedangkan pin SDA dan SCL merupakan pin I2C yang berfungsi sebagai data kirim dan terima serial.

c. Rangkaian dan Konfigurasi Pin Sensor IMU GY-521(MPU6050)



Gambar 4.8. Skematik Rangkaian Sensor IMU GY-521(MPU6050) di *Software Fritzing*

Tabel 4.3. Konfigurasi Pin Sensor IMU GY-521(MPU6050) dan Mikrokontroler STM32F411

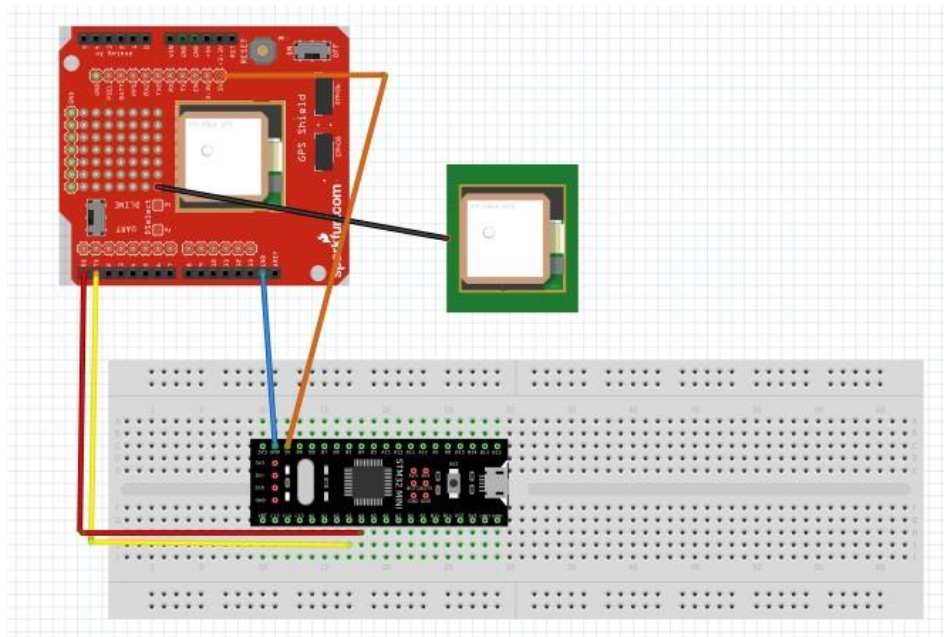
IMU GY-521(MPU6050)	STM32F411
G	G
Vcc	5v
SDA	B7
SCL	B6

Pada **Gambar 4.8** dan **Tabel 4.3** menunjukkan skematik rangkaian dan konfigurasi pin out antara sensor IMU GY-521(MPU6050) dan Mikrokontroler STM32F411. Langkah – langkah untuk merangkai rangkaiannya sebagai berikut:

- Menyambungkan pin GND dari GY-521(MPU6050) ke Pin GND di STM32F411
- Menyambungkan pin VCC dari GY-521(MPU6050) ke Pin 5v di STM32F411
- Menyambungkan pin SDA dari GY-521(MPU6050) ke Pin B7 di STM32F411
- Menyambungkan pin SCL dari GY-521(MPU6050) ke Pin B6 di di STM32F411

Pin GND merupakan pin *ground* untuk catu daya digital, pin VCC merupakan pin yang berfungsi sebagai pin masukan catu daya digital. 2. Sedangkan pin SDA dan SCL merupakan pin yang berfungsi sebagai antarmuka digital yang menggunakan komunikasi I2C atau SPI.

d. Rangkaian dan Konfigurasi Pin Modul GPS Neo-7m



Gambar 4.9. Skematik Rangkaian Modul GPS Neo-7m di *Software Fritzing*

Tabel 4.4. Konfigurasi Pin Modul GPS Neo-7m dan Mikrokontroler STM32F411

Modul GPS Neo-7m	STM32F411
G	G
Vcc	5v
RX	A4
TX	A3

Pada **Gambar 4.9** dan **Tabel 4.4** menunjukkan skematik rangkaian dan konfigurasi pin out antara Modul GPS Neo-7m dan Mikrokontroler STM32F411. Langkah – langkah untuk merangkai rangkaiannya sebagai berikut:

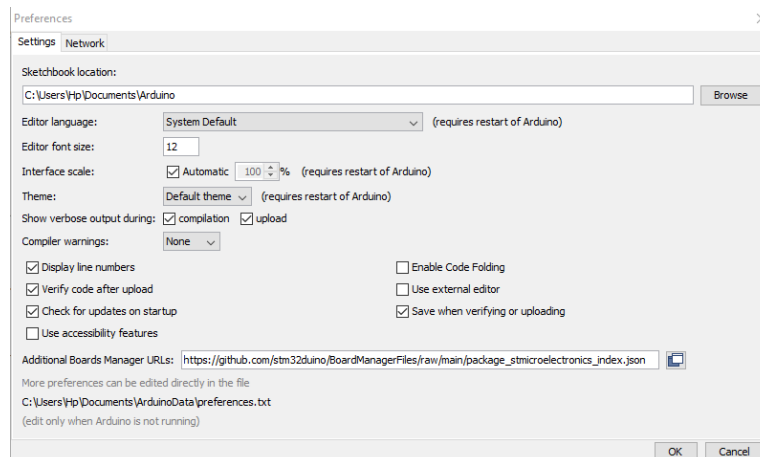
- Menyambungkan pin GND dari Modul GPS Neo-7m ke Pin GND di STM32F411
- Menyambungkan pin VCC dari Modul GPS Neo-7m ke Pin 5v di STM32F411
- Menyambungkan pin RX dari Modul GPS Neo-7m ke Pin A4 di STM32F411
- Menyambungkan pin TX dari Modul GPS Neo-7m ke Pin A3 di di STM32F411

Pin GND merupakan pin *ground* untuk catu daya digital, pin VCC merupakan pin yang berfungsi sebagai pemasok daya untuk modul. Sedangkan pin RX dan TX merupakan pin yang berfungsi sebagai pemancar dan penerima komunikasi serial.

4.4.3 Codingan Program dan Konfigurasi *Tools* di *Software* Arduino IDE

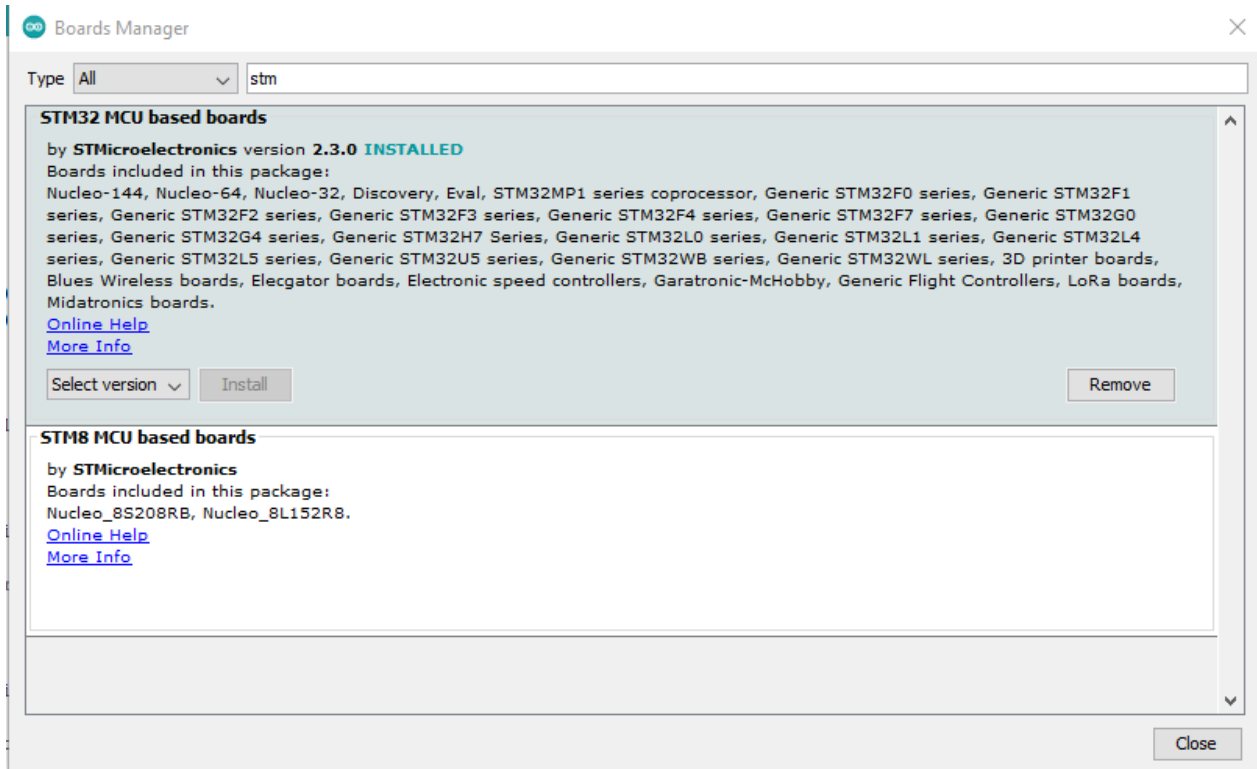
Pembacaan dari sensor *Accelerometer*, *Gyroscope*, dan Modul GPS dengan Mikrokontroler STM32F411 menggunakan bantuan *software* Arduino IDE yang ditampilkan melalui *tools serialmonitor* dan *serial plotter*. Namun sebelum dilakukan pembacaan sensor, ada beberapa langkah – langkah yang harus dilakukan agar hasil dari sensor – sensor dapat terbaca dengan baik di *software* Arduino IDE. Langkah – langkah tersebut dijabarkan melalui proses berikut:

- 1) Menginstal *library* STM32 ke *software* Arduino IDE dengan cara membuka *software* Arduino IDE, kemudian klik *File > Preferences* dan menambahkan URL ke kolom *Additional Board Manager URLs* seperti pada **Gambar 4.10**



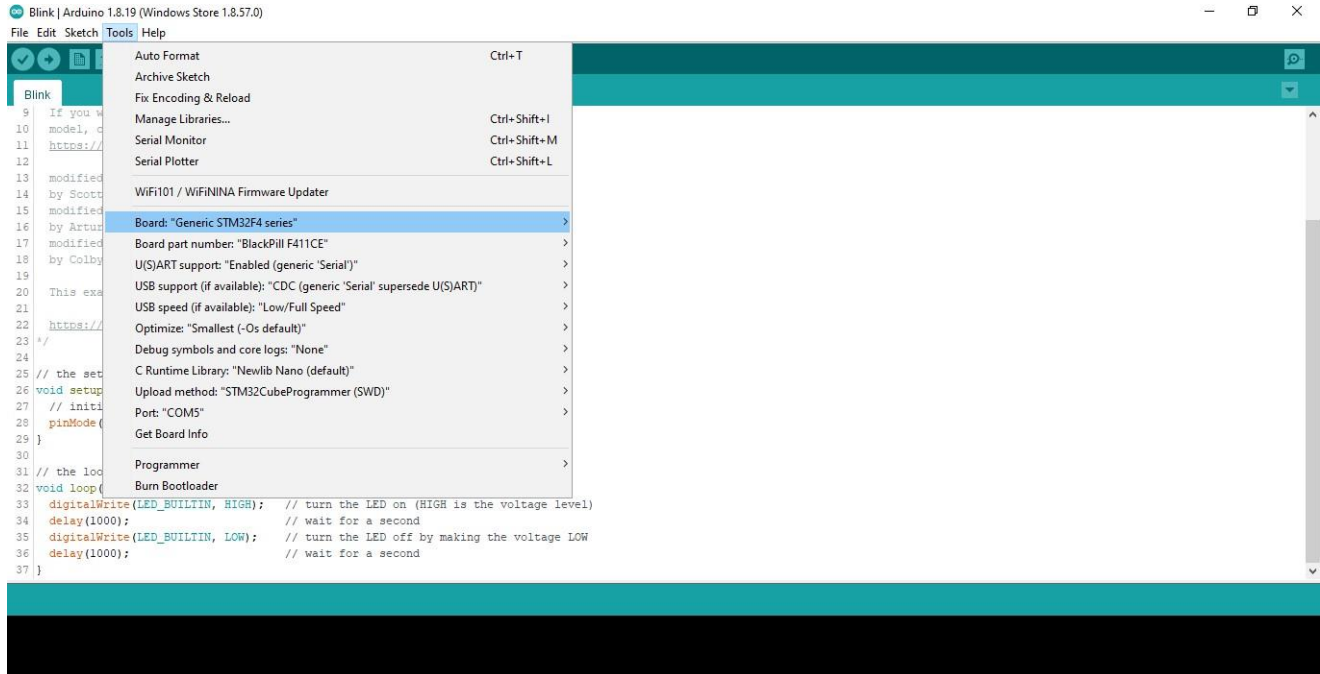
Gambar 4.10. Link URL *library* STM32 di *software* Arduino IDE

- 2) Selanjutnya adalah menginstal *Boards Manager* STM32 dengan cara klik *Tools* > *Board* > *Boards Manager* pada *software* Arduino IDE, kemudian *search boards* STM32 dan menginstal versi terbaru yang ada seperti pada gambar **Gambar 4.11**.



Gambar 4.11. *Boards Manager* STM32 di *software* Arduino IDE

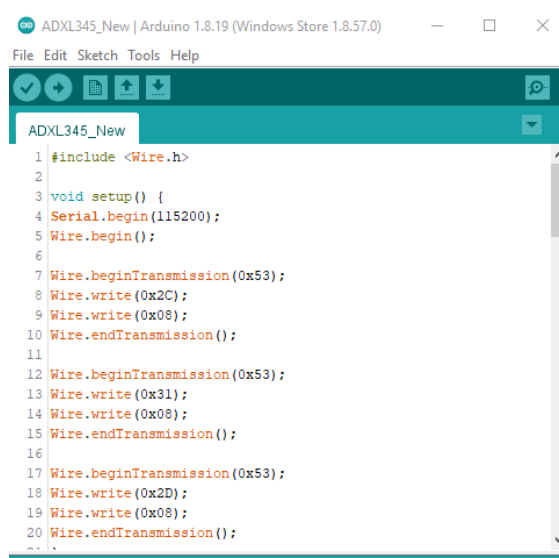
- 3) Langkah selanjutnya adalah melakukan konfigurasi *Tools* agar Mikrokontroler STM32F411 dapat terhubung di *software* Arduino IDE dan dapat dilakukan pembacaan data melalui *serial monitor* dan *serial plotter*. Konfigurasi diawali dengan klik *Tools* > *Board* > *STM32 Board* dan memilih *Generic STM32F4 series*, kemudian klik *Tools* > *Board Part Number* > *BlackPill F411CE*. Untuk sambungan ke laptop, STM32F411 menggunakan kabel USB *type-C* yang dibantu dengan *ST-Link V2 driver*. Konfigurasi *Tools* dari Mikrokontroler STM32F411 secara lengkap ditunjukkan pada **Gambar 4.12**.



Gambar 4.12. konfigurasi *Tools* STM32F411 di *software* Arduino IDE

4) Codings Program

Langkah terakhir agar sensor dapat terbaca dan ditampilkan outputnya melalui *serial monitor* dan *serial plotter* di *software* Arduino IDE adalah melakukan coding program. Codings ini berdasarkan dengan tipe masing-masing sensor yang sudah disesuaikan dengan mikrokontroler STM32F411 dan menggunakan bahasa C. Di awali dengan mendefinisikan *library* sensor dan diakhiri dengan perintah menampilkan hasil di *serial monitor*. Salah satu codingan sensor ditampilkan pada **Gambar 4.13** dan secara lengkap codingan program dari semua sensor pada laporan ini dicantumkan pada lampiran.



Gambar 4.13. Codings sensor menggunakan mikrokontroler STM32F411 di *software* Arduino IDE

4.5 Prosedur Pengambilan Data

Pengambilan data pada Praktik Kerja Lapangan ini menggunakan metode pembacaan sensor yang hasilnya ditampilkan melalui *tools serial monitor* pada *software* Arduino IDE, adapun prosedur – prosedur pengambilan data diuraikan sebagai berikut:

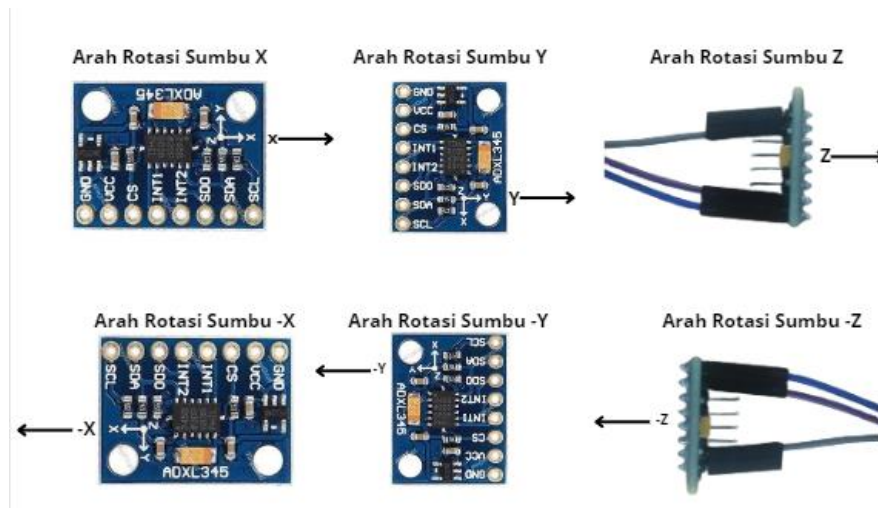
- a. Menyiapkan komponen
- b. Melakukan perangkaian antara sensor dan mikrokontroler
- c. Setelah semua sensor dirangkai dan saling terhubung, selanjutnya adalah menulis *script* program untuk membaca hasil sensor di *software* Arduino IDE
- d. Jika semua program sudah ditulis, klik menu *verify/compile* untuk mengecek apakah masih ada perintah yang *error* atau tidak, selanjutnya klik menu *upload* pada sensor jika tidak ada lagi *error* pada program dan pastikan sensor dan mikrokontroler sudah dihubungkan ke port USB pada laptop
- e. Setelah itu hasil pembacaan sensor akan ditampilkan melalui *tools serial monitor* dan langkah selanjutnya adalah proses pengambilan data sesuai dengan hasil yang ditampilkan pada *serial monitor*

Adapun nilai pembacaan sensor yang diambil sebagai data pada Praktik Kerja Lapangan ini disesuaikan dengan masing – masing tipe sensor. Sensor yang digunakan merupakan tipe sensor yang biasanya menjadi komponen utama dalam sebuah *flight controller* drone, prosedur pengambilan data di tiap sensor dijabarkan sebagai berikut:

4.5.1 Pengambilan Data Sensor *accelerometer* ADXL345

Sensor *accelerometer* ADXL345 berfungsi untuk melakukan pembacaan nilai akselerasi atau perubahan posisi pada sebuah benda. Dalam laporan ini, nilai *accelerometer* ditampilkan dengan enam arah rotasi sumbu yaitu sumbu X, sumbu Y dan sumbu Z serta arah sebaliknya yang dinotasikan dengan tanda (-) dengan satuan meter per sekon kuadrat (m/s^2). Namun sebelum hasil pembacaan sensor ditampilkan, ada beberapa langkah – langkah yang harus dilakukan yaitu:

- Merangkai sensor ke mikrokontroler sesuai dengan rangkaian yang ada pada **Gambar 4.6** dengan konfigurasi pin sesuai pada **Tabel 4.1**
- Menuliskan *script* program pada *software* Arduino IDE sesuai yang tertera pada **Lampiran 1**
- Mengarahkan sensor sesuai dengan arah rotasi sumbu yang ingin dibaca nilainya seperti yang tertampil pada **Gambar 4.14**

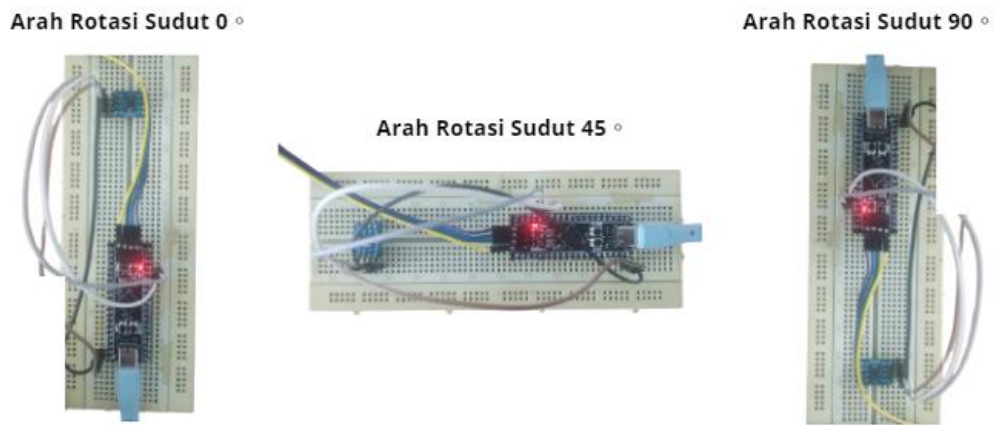


Gambar 4.14. Arah rotasi sumbu pada sensor *accelerometer* ADXL345

4.5.2 Pengambilan Data Sensor *Gyroscope* GY-25

Sensor *Gyroscope* GY-25 merupakan sensor IMU dengan *output* data berupa *roll*, *pitch* dan *yaw*. Namun pada penelitian ini, pengujian hanya berfokus pada pembacaan nilai *yaw* yakni *output* berupa derajat kemiringan sensor. *Yaw* inilah yang nantinya dapat digunakan untuk menggantikan sensor kompas sehingga posisi kemiringan drone dapat terbaca. Namun sebelum hasil pembacaan sensor ditampilkan pada *tools serial monitor*, ada beberapa langkah – langkah yang harus dilakukan yaitu:

- Merangkai sensor ke mikrokontroler sesuai dengan rangkaian yang ada pada **Gambar 4.7** dengan konfigurasi pin sesuai pada **Tabel 4.2**
- Menuliskan *script* program pada *software* Arduino IDE sesuai yang tertera pada **Lampiran 2**
- Mengarahkan sensor sesuai dengan arah rotasi sudut yang ingin dibaca nilainya seperti yang tertampil pada **Gambar 4.15**

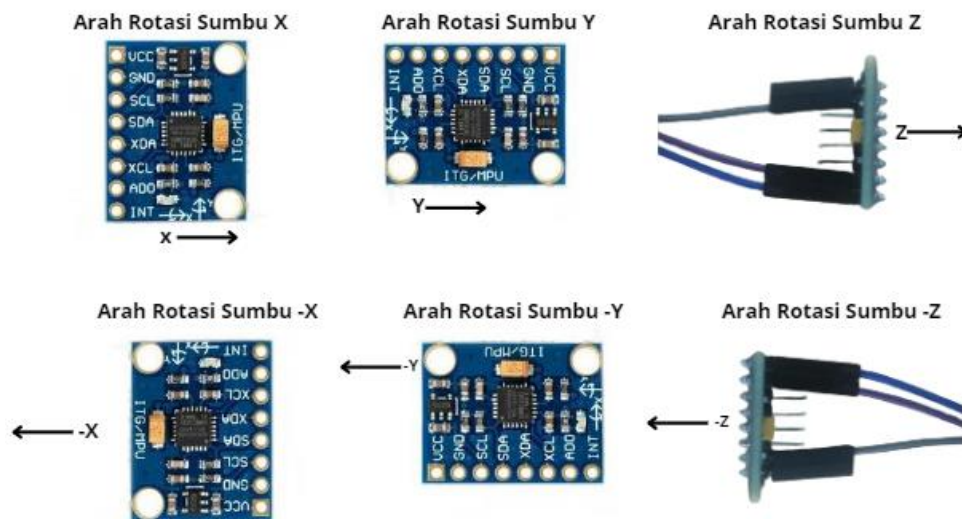


Gambar 4.15. Arah rotasi sudut pengambilan data pada sensor *gyroscope* GY-25

4.5.3 Pengambilan Data Sensor IMU GY-521(MPU6050)

Pada penelitian ini, Sensor IMU GY-251 digunakan untuk membaca nilai *accelerometer* dan *gyroscope* yang mana hasilnya ditampilkan melalui *serial monitor* dalam satuan m/s^2 untuk nilai *accelerometer* dan satuan derajat per detik ($^{\circ}/s$) untuk nilai *gyroscope*. Namun agar hasil pembacaan sensor muncul di *tools serial monitor*, ada beberapa langkah – langkah yang harus dilakukan yaitu:

- Merangkai sensor ke mikrokontroler sesuai dengan rangkaian yang ada pada **Gambar 4.8** dengan konfigurasi pin sesuai pada **Tabel 4.3**
- Menuliskan *script* program pada *software* Arduino IDE sesuai yang tertera pada **Lampiran 3**
- Mengarahkan sensor sesuai dengan arah rotasi sumbu yang ingin dibaca nilainya seperti yang tertampil pada **Gambar 4.16**



Gambar 4.16. Arah rotasi sumbu pada sensor IMU GY-521

4.5.4 Pengambilan Data Modul GPS Neo-7m

Modul GPS pada penelitian ini digunakan untuk memperoleh lokasi serta memberikan informasi waktu, secara terus menerus tanpa bergantung pada waktu dan cuaca tertentu. Lokasi pada GPS sendiri dinyatakan dalam suatu titik koordinat yang mana nantinya titik koordinat tersebut dapat dimanfaatkan untuk menampilkan lokasi dimana keberadaan GPS tersebut. Namun sebelum hasil pembacaan Modul GPS ditampilkan pada *tools serial monitor*, ada beberapa langkah – langkah yang harus dilakukan yaitu:

- Merangkai sensor ke mikrokontroler sesuai dengan rangkaian yang ada pada **Gambar 4.9** dengan konfigurasi pin sesuai pada **Tabel 4.4**

- Menuliskan *script* program pada *software* Arduino IDE sesuai yang tertera pada **Lampiran 4**
- Meletakkan modul GPS pada 2 titik lokasi yang berbeda, hal ini bertujuan untuk membandingkan hasil pembacaan modul. Adapun lokasi pengambilan data ditunjukkan pada **Gambar 4.17 dan Gambar 4.18**



Gambar 4.17. Gedung utama Pustekbang-BRIN lokasi pengambilan data 1 GPS



Gambar 4.18. Gedung Avionik lokasi pengambilan data 2 GPS

BAB V

HASIL DAN PEMBAHASAN

5.1 Hasil

5.1.1 Hasil Pembacaan Sensor *Accelerometer* ADXL345

Tabel 5.1. Data hasil pembacaan sensor *Accelerometer* ADXL345 untuk arah sumbu x

Data ke	Arah Sumbu	Nilai Pembacaan Sensor		
		Sumbu X (m/s ²)	Sumbu Y (m/s ²)	Sumbu Z (m/s ²)
1	x	1.07	0.07	0.02
2	x	1.08	0.10	0.03
3	x	1.07	0.07	0.02
4	x	1.06	0.09	0.01
5	x	1.06	0.05	0.01

Tabel 5.2. Data hasil pembacaan sensor *Accelerometer* ADXL345 untuk arah sumbu -x

Data ke	Arah Sumbu	Nilai Pembacaan Sensor		
		Sumbu X (m/s ²)	Sumbu Y (m/s ²)	Sumbu Z (m/s ²)
1	-x	-0.96	0.01	0.22
2	-x	-0.95	0.03	0.24
3	-x	-0.99	0.01	0.22
4	-x	-0.95	0.01	0.22
5	-x	-0.95	0.02	0.24

Tabel 5.3. Data hasil pembacaan sensor *Accelerometer* ADXL345 untuk arah sumbu y

Data ke	Arah Sumbu	Nilai Pembacaan Sensor		
		Sumbu X (m/s ²)	Sumbu Y (m/s ²)	Sumbu Z (m/s ²)
1	y	0.09	1.04	0.01
2	y	0.09	1.04	0.01
3	y	0.08	1.04	0.01
4	y	0.08	1.04	0.01
5	y	0.09	1.03	0.02

Tabel 5.4. Data hasil pembacaan sensor *Accelerometer* ADXL345 untuk arah sumbu -y

Data ke	Arah Sumbu	Nilai Pembacaan Sensor		
		Sumbu X (m/s ²)	Sumbu Y (m/s ²)	Sumbu Z (m/s ²)
1	-y	0.07	-1.05	0.18
2	-y	0.06	-1.06	0.17
3	-y	0.07	-1.06	0.16
4	-y	0.06	-1.05	0.16
5	-y	0.06	-1.05	0.17

Tabel 5.5. Data hasil pembacaan sensor *Accelerometer* ADXL345 untuk arah sumbu z

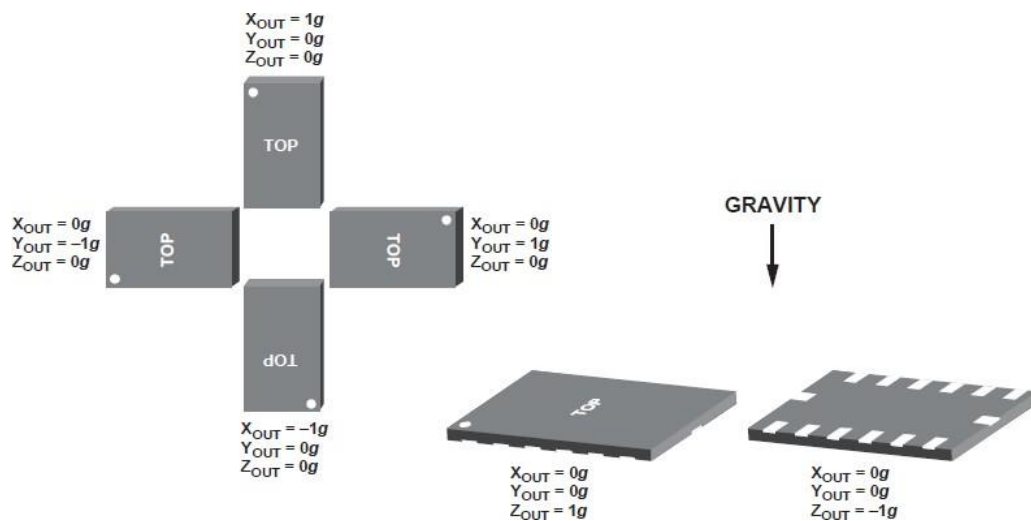
Data ke	Arah Sumbu	Nilai Pembacaan Sensor		
		Sumbu X (m/s ²)	Sumbu Y (m/s ²)	Sumbu Z (m/s ²)
1	z	0.04	0.20	1.04
2	z	0.04	0.21	1.04
3	z	0.04	0.20	1.04
4	z	0.04	0.21	1.03
5	z	0.04	0.20	1.04

Tabel 5.6. Data hasil pembacaan sensor *Accelerometer* ADXL345 untuk arah sumbu -z

Data ke	Arah Sumbu	Nilai Pembacaan Sensor		
		Sumbu X (m/s ²)	Sumbu Y (m/s ²)	Sumbu Z (m/s ²)
1	-z	0.07	0.22	-1.03
2	-z	0.08	0.14	-0.98
3	-z	0.09	0.13	-0.96
4	-z	0.10	0.12	-0.96
5	-z	0.12	0.12	-0.96

Pembahasan:

Berdasarkan hasil pembacaan sensor *accelerometer* ADXL345 yang ditampilkan pada **Tabel 5.1 - Tabel 5.6**, sensor melakukan pengukuran percepatan pada tiga arah yang saling tegak lurus yakni sumbu X, Y, dan Z serta sumbu -X, -Y dan -Z untuk arah yang berlawanan. Hasil ini memungkinkan sensor untuk mendeteksi gerakan dan perubahan orientasi dari objek atau perangkat yang ditempatkan di mana sensor tersebut dipasang. Pada *flight controller* drone, sensor *accelerometer* biasanya berfungsi untuk mendeteksi perubahan posisi drone dan menilai gerakan serta percepatannya.



Gambar 5.1. Respon *output* sensor *accelerometer* ADXL345 dan orientasi terhadap gravitasi

Berdasarkan **Gambar 5.1**, nilai pembacaan sensor ADXL345 menyesuaikan dengan arah orientasi sensornya dengan satuan nilai berupa g (satuan percepatan gravitasi), dimana 1g merupakan satu satuan percepatan rata-rata gravitasi bumi yaitu sebesar $9,8\text{m/s}^2$. Hasil pembacaan sensor yang dilampirkan pada tabel **Tabel 5.1 - Tabel 5.6** menunjukkan hasil yang sesuai dengan respon *output* sensor yang menunjukkan bahwa sensor dan mikrokontroler sudah bekerja dengan baik sesuai dengan perintah di program.

5.1.1.1 Analisis Data Hasil Pembacaan Sensor Accelerometer ADXL345

Tabel 5.7. Hasil Pengolahan data hasil pembacaan Sensor ADXL345 untuk sumbu x dan -x

No	Hasil Pembacaan Sensor					
	Sumbu x			Sumbu -x		
	x (m/s ²)	(x-x̄) (m/s ²)	(x-x̄) ² (m/s ²)	x (m/s ²)	(x-x̄) (m/s ²)	(x-x̄) ² (m/s ²)
1	1.07	0.002	4E-06	-0.96	0	0
2	1.08	0.012	0.000144	-0.95	0.01	0.0001
3	1.07	0.002	4E-06	-0.99	-0.03	0.0009
4	1.06	-0.008	6.4E-05	-0.95	0.01	0.0001
5	1.06	-0.008	6.4E-05	-0.95	3.85	14.8225
Σ	5.34	-1.11E-15	0.00028	-4.8	3.84	0.0011
Rata2 x	1.068			-0.96		
Ralat x	0.00374166			0.0074162		

Hasil pengolahan data seperti yang ditunjukkan pada **Tabel 5.7** merupakan hasil pengolahan data hasil pembacaan sensor ADXL345 untuk sumbu x dan -x. Pada pengukuran yang dilakukan didapatkan bahwa selama 5 kali pembacaan sensor, nilai rata – rata sumbu x yaitu sebesar 1.068 dengan nilai ralat sebesar 0.00374166 dan rata – rata sumbu -x sebesar -0.96 dengan nilai ralat 0.0074162 yang menunjukkan bahwa hasil pengukuran hampir akurat dan mendekati nilai asli dimana nilai yang seharusnya yaitu sebesar 1g.

Tabel 5.8. Hasil Pengolahan data hasil pembacaan Sensor ADXL345 untuk sumbu y dan -y

No	Hasil Pembacaan Sensor					
	Sumbu x			Sumbu -x		
	x (m/s ²)	(x- \bar{x}) (m/s ²)	(x- \bar{x}) ² (m/s ²)	x (m/s ²)	(x- \bar{x}) (m/s ²)	(x- \bar{x}) ² (m/s ²)
1	1.04	0.002	4E-06	-1.05	0.004	0.000016
2	1.04	0.002	4E-06	-1.06	-0.006	3.6E-05
3	1.04	0.002	4E-06	-1.06	-0.006	3.6E-05
4	1.04	0.002	4E-06	-1.05	0.004	0.000016
5	1.03	-0.008	6.4E-05	-1.05	4.22	17.8084
Σ	5.19	0	8E-05	-5.27	4.216	0.000104
Rata2 x	1.038			-1.054		
Ralat x	0.002			0.00228035		

Hasil pengolahan data seperti yang ditunjukkan pada **Tabel 5.8** merupakan hasil pengolahan data hasil pembacaan sensor ADXL345 untuk sumbu y dan -y. Pada pengukuran yang dilakukan didapatkan bahwa selama 5 kali pembacaan sensor, nilai rata – rata sumbu y yaitu sebesar 1.038 dengan nilai ralat sebesar 0.002 dan rata – rata sumbu -y sebesar -1.054 dengan nilai ralat 0.00228035 yang menunjukkan bahwa hasil pengukuran hampir akurat dan mendekati nilai asli dimana nilai yang seharusnya yaitu sebesar 1g.

Tabel 5.9. Hasil Pengolahan data hasil pembacaan Sensor ADXL345 untuk sumbu z dan -z

No	Hasil Pembacaan Sensor					
	Sumbu x			Sumbu -x		
	x (m/s ²)	(x- \bar{x}) (m/s ²)	(x- \bar{x}) ² (m/s ²)	x (m/s ²)	(x- \bar{x}) (m/s ²)	(x- \bar{x}) ² (m/s ²)
1	1.04	0.002	4E-06	-1.03	-0.052	0.002704
2	1.04	0.002	4E-06	-0.98	-0.002	4E-06
3	1.04	0.002	4E-06	-0.96	0.018	0.000324
4	1.03	-0.008	6.4E-05	-0.96	0.018	0.000324
5	1.04	0.002	4E-06	-0.96	3.93	15.4449
Σ	5.19	0	8E-05	-4.89	3.912	0.003356
Rata2 x	1.038			-0.978		
Ralat x	0.002			0.01295376		

Hasil pengolahan data seperti yang ditunjukkan pada **Tabel 5.9** merupakan hasil pengolahan data hasil pembacaan sensor ADXL345 untuk sumbu z dan -z. Pada pengukuran yang dilakukan didapatkan bahwa selama 5 kali pembacaan sensor, nilai rata – rata sumbu z yaitu sebesar 1.038 dengan nilai ralat sebesar 0.002 dan rata – rata sumbu -z sebesar -0.978 dengan nilai ralat 0.01295376 yang menunjukkan bahwa hasil pengukuran hampir akurat dan mendekati nilai asli dimana nilai yang seharusnya yaitu sebesar 1g.

5.1.2 Hasil Pembacaan Sensor *Gyroscope* GY-25

Tabel 5.10. Data hasil pembacaan sensor *Gyroscope* GY-25

Arah Hadap (°)	Data ke	Nilai Pembacaan Sensor (°)
0°	1	0.00
	2	0.01
	3	0.02
	4	0.04
	5	0.08
45°	1	45.97
	2	46.67
	3	47.44
	4	48.21
	5	48.87
90°	1	89.93
	2	90.05
	3	90.15
	4	90.26
	5	90.36

Sensor *Gyroscope* GY-25 merupakan sensor IMU dengan *output* data berupa *roll*, *pitch* dan *yaw*. Namun pada penelitian ini, pengujian hanya berfokus pada pembacaan nilai *yaw* yakni *output* berupa derajat kemiringan sensor. *Yaw* inilah yang nantinya dapat digunakan untuk menggantikan sensor kompas sehingga posisi kemiringan drone dapat terbaca. Dari hasil pengujian yang dilampirkan pada **Tabel 5.10**, dilakukan pengujian menggunakan 3 kemiringan sudut yakni 0°, 45° dan 90°. Ketika sensor di putar sesuai arah sudut tersebut maka akan terbaca nilai sudut kemiringannya. Namun dari data yang didapatkan pada **Tabel 5.10**, hasilnya tidak pas sesuai dengan arah sudut. Hal ini disebabkan karena sensor yang bergerak dan tidak stabil ketika dilakukan pengujian.

5.1.2.1 Analisis Data Hasil Pembacaan Sensor Gyroscope GY-25

Tabel 5.11. Hasil Pengolahan data hasil pembacaan Sensor Gyroscope GY-25

No.	Arah Hadap								
	0°			45°			90°		
	x (°)	(x-x̄) (°)	(x-x̄) ² (°)	x (°)	(x-x̄) (°)	(x-x̄) ² (°)	x (°)	(x-x̄) (°)	(x-x̄) ² (°)
1	0	-0.03	0.0009	45.97	-1.462	2.137444	89.93	-0.22	0.0484
2	0.01	-0.02	0.0004	46.67	-0.762	0.580644	90.05	-0.1	0.01
3	0.02	-0.01	0.0001	47.44	0.008	6.4E-05	90.15	0	0
4	0.04	0.01	1E-04	48.21	0.778	0.605284	90.26	0.11	0.0121
5	0.08	0.05	0.0025	48.87	1.438	2.067844	90.36	0.21	0.0441
∑	0.15	0	0.004	237.16	-1.42109	5.39128	450.75	-1.4E-	0.1146
Rata2	0.03			47.432			90.15		
Ralat	0.014142			0.519196			0.075697		

Hasil pengolahan data seperti yang ditunjukkan pada **Tabel 5.11** merupakan hasil pengolahan data hasil pembacaan sensor GY-25 untuk arah hadap sudut 0°, 45° dan 90°. Berdasarkan hasil pengukuran sebanyak 5 kali, didapatkan nilai rata – rata untuk

sudut 0° sebesar 0.03° dengan nilai ralat sebesar 0.014142, sudut 45° memiliki rata – rata pengukuran sebesar 47.432° dengan nilai ralat 0.519196 serta untuk arah sudut 90° memiliki rata – rata pengukuran sebesar 90.15° dengan nilai ralat sebesar 0.075697. Berdasarkan hasil tersebut dapat disimpulkan bahwa hasil pengukuran memiliki nilai yang tidak sama persis dengan nilai arah sudut yang sebenarnya, hal ini disebabkan oleh beberapa faktor yang salah satunya karena ketika dilakukan pengukuran, posisi sensor masih dalam keadaan bergerak sehingga hasil yang terbaca tidak akurat.

5.1.3 Hasil Pembacaan Sensor IMU GY-521(MPU6050)

Tabel 5.12. Data hasil pembacaan sensor IMU GY-521(MPU6050) untuk arah sumbu x

Data ke	Arah Sumbu	Pembacaan Nilai <i>Accelerometer</i>			Pembacaan Nilai <i>Gyroscope</i>		
		Sumbu aX (m/s ²)	Sumbu aY (m/s ²)	Sumbu aZ (m/s ²)	Sumbu gX (°/s)	Sumbu gY (°/s)	Sumbu gZ (°/s)
1	x	17528	496	-6288	3	240	161
2	x	16840	260	-6384	-424	336	186
3	x	16804	496	-5848	-51	-436	238
4	x	17076	768	-6288	-443	-301	206
5	x	16840	-220	-5656	1228	1497	403
6	x	17332	-436	-2572	-476	750	544
7	x	17000	-468	-2588	-235	-423	306
8	x	17048	-372	-2440	-420	-116	135
9	x	17160	-500	-2352	-396	-129	176
10	x	17144	-532	-3112	-281	74	106

Tabel 5.13. Data hasil pembacaan sensor IMU GY-521(MPU6050) untuk arah sumbu -x

Data ke	Arah Sumbu	Pembacaan Nilai <i>Accelerometer</i>			Pembacaan Nilai <i>Gyroscope</i>		
		Sumbu aX (m/s ²)	Sumbu aY (m/s ²)	Sumbu aZ (m/s ²)	Sumbu gX (°/s)	Sumbu gY (°/s)	Sumbu gZ (°/s)

1	-x	-13684	-3868	3328	-677	-174	402
2	-x	-13596	-3724	3552	10	277	147
3	-x	-13532	-3952	3576	-13	408	177
4	-x	-13628	-4088	3008	-1755	-678	255
5	-x	-13656	-4016	3208	-119	228	187
6	-x	-13692	-3944	3136	-497	179	96
7	-x	-13744	-3856	3464	-72	-9	176
9	-x	-13760	-3980	2944	-607	-2	268
10	-x	-13872	-3932	2940	-1630	-779	233

Tabel 5.14. Data hasil pembacaan sensor IMU GY-521(MPU6050) untuk arah sumbu y

Data ke	Arah Sumbu	Pembacaan Nilai Accelerometer			Pembacaan Nilai Gyroscope		
		Sumbu aX (m/s ²)	Sumbu aY (m/s ²)	Sumbu aZ (m/s ²)	Sumbu gX (°/s)	Sumbu gY (°/s)	Sumbu gZ (°/s)
1	y	1736	15480	-704	-2831	2194	271
2	y	1424	15912	-960	-1341	-626	542
3	y	1460	15948	-1696	-225	-765	686
4	y	1816	15844	-2240	-375	-394	757
5	y	2068	15868	-2020	-1059	-799	864
6	y	1236	16524	-1480	-2232	318	118
7	y	1516	15848	-1144	946	1296	-1337
8	y	1276	15936	-1276	387	763	502
9	y	948	15720	-292	-3082	-2115	1547
10	y	940	15924	-1040	-1036	-109	149

Tabel 5.15. Data hasil pembacaan sensor IMU GY-521(MPU6050) untuk arah sumbu -y

Data ke	Arah Sumbu	Pembacaan Nilai Accelerometer			Pembacaan Nilai Gyroscope		
		Sumbu aX (m/s ²)	Sumbu aY (m/s ²)	Sumbu aZ (m/s ²)	Sumbu gX (°/s)	Sumbu gY (°/s)	Sumbu gZ (°/s)
1	-y	1168	-16316	-252	-353	27	234

2	-y	1180	-16428	-396	-360	51	223
3	-y	1092	-16508	-436	-221	43	178
4	-y	1132	-16328	-520	-442	12	252
5	-y	1084	-16456	-492	-193	34	203
6	-y	1144	-16276	-368	-467	39	245
8	-y	1016	-16280	-52	-875	-110	353
9	-y	1048	-16428	-388	-102	72	196
10	-y	764	-16344	-544	-550	14	316

Tabel 5.16. Data hasil pembacaan sensor IMU GY-521(MPU6050) untuk arah sumbu z

Data ke	Arah Sumbu	Pembacaan Nilai Accelerometer			Pembacaan Nilai Gyroscope		
		Sumbu aX (m/s ²)	Sumbu aY (m/s ²)	Sumbu aZ (m/s ²)	Sumbu gX (°/s)	Sumbu gY (°/s)	Sumbu gZ (°/s)
1	z	-292	-344	13220	-353	30	208
2	z	-380	-328	12972	-344	28	182
3	z	-460	-296	13152	-348	72	189
4	z	-324	-264	13084	-340	20	210
5	z	-392	-256	13128	-346	60	219
6	z	-284	-324	13096	-365	35	211
7	z	-308	-364	13116	-475	-13	205
8	z	-248	-348	13164	-377	35	177
9	z	-236	-344	13168	-329	64	210
10	z	-304	-296	13148	-379	55	195

Tabel 5.17. Data hasil pembacaan sensor IMU GY-521(MPU6050) untuk arah sumbu -z

Data ke	Arah Sumbu	Pembacaan Nilai Accelerometer			Pembacaan Nilai Gyroscope		
		Sumbu aX (m/s ²)	Sumbu aY (m/s ²)	Sumbu aZ (m/s ²)	Sumbu gX (°/s)	Sumbu gY (°/s)	Sumbu gZ (°/s)
1	-z	840	-756	-19764	-309	12	263
2	-z	840	-628	-19908	-237	311	263

3	-z	832	-844	-19788	-267	-123	377
4	-z	772	-664	-19968	-490	-335	302
6	-z	700	-452	-19944	-271	-67	287
7	-z	784	-532	-19836	-380	-41	246
8	-z	732	-536	-20048	-322	274	213
9	-z	836	-492	-20072	-285	427	268
10	-z	472	-652	-20040	-442	-122	33

Modul GY-521 merupakan sebuah modul berinti MPU-6050 yang merupakan 6 *axis Motion Processing Unit* dengan penambahan *voltage regulator* dan beberapa komponen tambahan lainnya yang membuat modul ini dapat digunakan dengan *supply* 3-5 V DC. Modul ini memiliki *interface* I2C yang dapat disambungkan langsung ke mikrokontroler yang juga memiliki konektifitas I2C. Sensor GY-521 berisi 3 sumbu sudut *accelerometer* dan 3 sumbu sudut *gyroscope*. Ini mengubah 16 bit konverter analog-digital untuk digitalisasi *accelerometer* dan nilai *gyroscope* menggunakan gerakan digital bawaannya. Untuk melacak gerakan cepat dan lambat dari benda yang diam, skala deteksi posisi sudut dapat diprogram untuk nilai ± 250 °/dtk, ± 500 °/dtk, ± 1000 °/dtk dan ± 2000 °/dtk, skala deteksi percepatan sudut dapat diprogram ke nilai ± 2 g, ± 4 g, ± 8 g, ± 16 g. Perubahan pergerakan pembacaan dari *accelerometer* dan *gyroscope* akan diproses oleh mikrokontroler STM32F411 dan mengkonversi sinyal tersebut kedalam nilai integer *accelerator* (aX, aY) dan *gyroscope* (gX, gY).

Hasil data yang ditampilkan seperti pada **Tabel 5.12 – Tabel 5.17** merupakan pembacaan nilai *accelerometer* dan *gyroscope* dengan masing – masing sumbu X, Y dan Z menggunakan sensor IMU GY-521. Berdasarkan data percobaan, hasil pembacaan sensor menunjukkan hasil yang sesuai dengan perintah program. Ketika sensor diarahkan ke sumbu tertentu, maka nilai yang ditampilkan akan berbeda dan

cenderung lebih besar dari nilai untuk sumbu yang lain. Hal ini dikarenakan sensor di program untuk bekerja dengan membaca nilai sinyal x, y dan z dari MPU6050 melalui bus I2C, yang dipetakan menjadi sumbu x dan sumbu y dari nilai -17000 <-> 17000 hingga -50 <-> 50.

5.1.4 Hasil Pembacaan Modul GPS Neo-7m

Tabel 5.18. Data hasil pembacaan Modul GPS Neo-7m di Gedung Utama Pustekbang BRIN

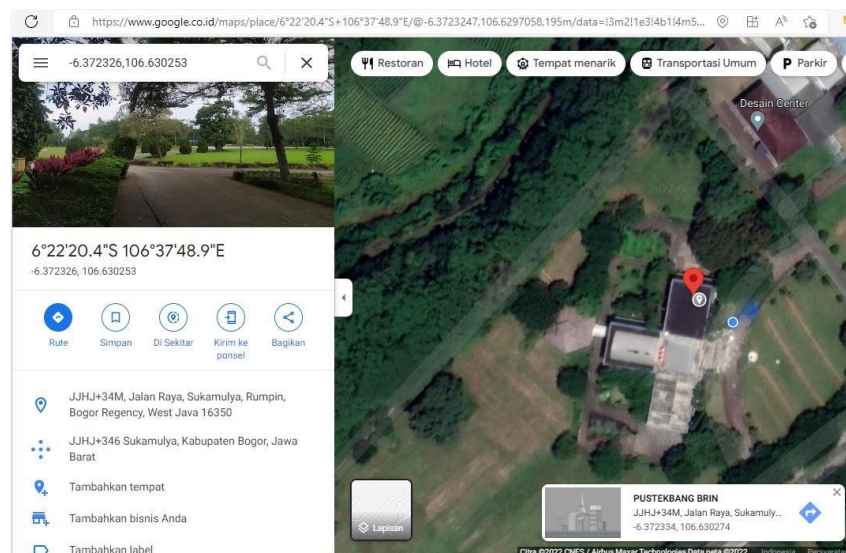
Data ke	Date/Time	Hasil Pembacaan Modul GPS	
		Koordinat (°)	Altitude (m)
1	9/6/2022, 04:35:57	-6.372194, 106.630248	77.50
2	9/6/2022, 04:35:58	-6.372201, 106.630253	77.50
3	9/6/2022, 04:35:59	-6.372212, 106.630265	77.50
4	9/6/2022, 04:36:00	-6.372215, 106.630269	77.50
5	9/6/2022, 04:36:01	-6.372218, 106.630270	77.50
6	9/6/2022, 05:04:59	-6.372185, 106.630487	77.50
7	9/6/2022, 05:05:02	-6.372191, 106.630477	77.50
8	9/6/2022, 05:05:04	-6.372193, 106.630498	77.50
9	9/6/2022, 05:05:05	-6.372189, 106.630508	77.50
10	9/6/2022, 05:05:06	-6.372189, 106.630516	77.50

Tabel 5.19. Data hasil pembacaan Modul GPS Neo-7m di Gedung Avionik Pustekbang BRIN

Data ke	Date/Time	Hasil Pembacaan Modul GPS	
		Koordinat (°)	Altitude (m)
1	9/9/2022, 08:25:27	-6.374185, 106.628061	130.70
2	9/9/2022, 08:25:29	-6.374183, 106.628055	130.70
3	9/9/2022, 08:25:30	-6.374183, 106.628054	130.70

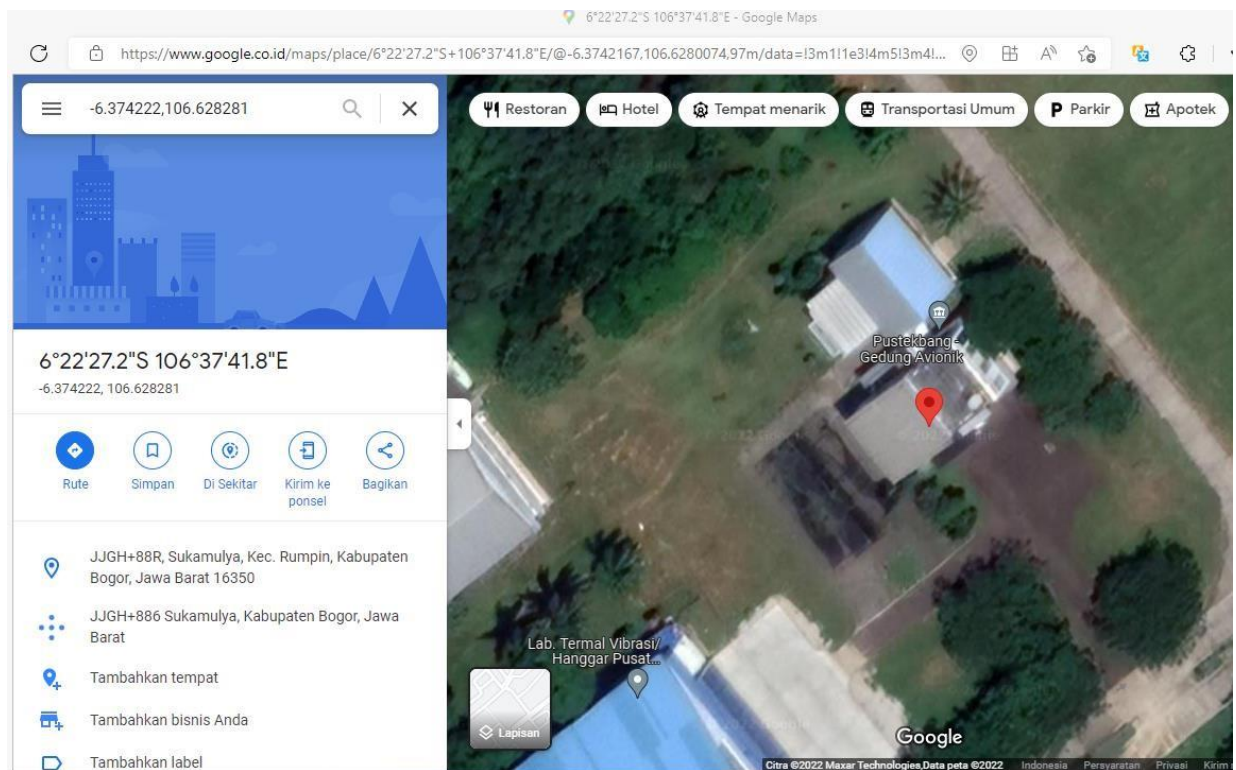
4	9/9/2022, 08:25:31	-6.374183, 106.628053	130.70
5	9/9/2022, 08:25:33	-6.374198, 106.628084	130.70
6	9/9/2022, 08:25:35	-6.374218, 106.628123	130.70
7	9/9/2022, 08:25:36	6.374225, 106.628135	130.70
8	9/9/2022, 08:25:37	-6.374230, 106.628146	130.70
9	9/9/2022, 08:25:38	-6.374235, 106.628155	130.70
10	9/9/2022, 08:25:39	-6.374239, 106.628162	130.70

Global Positioning System (GPS) merupakan sistem navigasi atau penentuan lokasi. Sistem GPS sendiri digunakan untuk memperoleh lokasi dan juga bisa berupa kecepatan serta memberikan informasi waktu, secara terus menerus tanpa bergantung pada waktu dan cuaca tertentu. Lokasi pada GPS sendiri dinyatakan dalam suatu titik koordinat yang mana nantinya titik koordinat tersebut dapat dimanfaatkan untuk menampilkan lokasi dimana keberadaan GPS tersebut. Seperti pada **Tabel 5.18** dan **Tabel 5.19**, hasil pembacaan modul GPS menunjukkan beberapa data yakni berupa tanggal dan jam pengambilan data, titik koordinat pengambilan data serta *altitude* (ketinggian) lokasi dari permukaan laut.



Gambar 5.2. Titik koordinat lokasi 1 pengambilan data melalui *website* google maps

Gambar 5.2 menunjukkan titik koordinat lokasi pengambilan data yang pertama, dimana data koordinat ini akan digunakan sebagai acuan untuk melihat apakah modul GPS melakukan pembacaan yang akurat atau tidak. Berdasarkan data pengujian pada **Tabel 5.18** dapat disimpulkan bahwa hasil yang didapatkan pada pengujian ini menunjukkan data yang hampir akurat, hasil yang paling mendekati ditampilkan pada pembacaan data yang ke-5.



Gambar 5.3. Titik koordinat lokasi 2 pengambilan data melalui *website* google maps

Gambar 5.3 menunjukkan titik koordinat lokasi pengambilan data yang kedua, dimana data koordinat ini akan digunakan sebagai acuan untuk melihat apakah modul GPS melakukan pembacaan yang akurat atau tidak. Berdasarkan data pengujian pada **Tabel**

5.19 dapat disimpulkan bahwa hasil yang didapatkan pada pengujian ini menunjukkan data yang juga hampir akurat, hasil yang paling mendekati ditampilkan pada pembacaan data yang ke-7.

5.1.5 Analisa Kinerja Mikrokontroler STM32F411 dalam Melakukan Pembacaan

Sensor

Mikrokontroler STM32F411 adalah salah satu dari serangkaian mikrokontroler yang dikembangkan oleh perusahaan *STMicroelectronics*. Mikrokontroler ini termasuk dalam keluarga STM32F4, yang memiliki fitur-fitur yang cukup kuat dan serbaguna untuk berbagai aplikasi. Dalam penelitian ini, penulis melakukan pengujian dan analisa kinerja dari mikrokontroler STM32F411 dalam melakukan pembacaan sensor – sensor. Jika menunjukkan kinerja yang baik, mikrokontroler ini nantinya akan dikembangkan sebagai alternatif *flight controller* pada drone.

Berdasarkan pengujian dan hasil pembacaan sensor yang dilakukan, didapatkan beberapa keunggulan dari Mikrokontroler STM32F411:

1. Mikrokontroler ini menggunakan arsitektur ARM Cortex-M4 dengan unit *Floating Point Unit* (FPU), yang memungkinkan untuk melakukan operasi *floating-point* (bilangan pecahan) dengan cepat. Kecepatan prosesor bisa mencapai beberapa ratus *megahertz*.
2. STM32F411 memiliki berbagai macam memori, termasuk *Flash memory* untuk menyimpan program (kode), RAM untuk menyimpan data sementara, dan EEPROM (*non-volatile memory*) untuk menyimpan data yang harus tetap ada bahkan ketika daya dimatikan.
3. Mikrokontroler ini memiliki banyak pin I/O yang dapat dikonfigurasi untuk

berbagai tugas, termasuk input analog dan digital, output PWM, komunikasi serial seperti UART, SPI, dan I2C.

4. STM32F411 mendukung berbagai protokol komunikasi, termasuk UART, SPI, I2C, CAN (*Controller Area Network*), USB, dan *Ethernet*.
5. Mikrokontroler ini dilengkapi dengan berbagai periferal seperti *timer*, *watchdog*, ADC (*Analog-to-Digital Converter*) dan DAC (*Digital-to-Analog Converter*).
6. STM32F411 memiliki fitur keamanan seperti *memory protection unit* (MPU) untuk mencegah akses ilegal ke memori, serta fitur-fitur kriptografi untuk mengamankan data.
7. Mikrokontroler ini dapat diprogram menggunakan berbagai bahasa pemrograman, termasuk bahasa C/C++ dengan bantuan perangkat lunak dan perangkat keras yang sesuai. Dalam penelitian ini, penulis menggunakan bantuan *software* Arduino IDE untuk menulis perintah pada sensor – sensor.

Pengembangan sebuah mikrokontroler menjadi *flight controller* pada drone harus memenuhi beberapa syarat seperti harus memiliki prosesor yang cukup kuat untuk menangani algoritma kontrol yang kompleks dan pengolahan data sensor secara *real-time*, memori yang cukup juga diperlukan untuk menyimpan program dan data, mendukung komunikasi dengan perangkat eksternal seperti remote control transmitter, sistem GPS, dan modul telemetri. Port dan protokol komunikasi seperti PWM, PPM, S.Bus, UART, I2C dan SPI. Berdasarkan persyaratan tersebut, maka dapat disimpulkan bahwa mikrokontroler STM32F411 berpotensi dapat dijadikan sebagai alternatif *flight controller* pada drone karena memiliki spesifikasi yang mumpuni sesuai dengan yang dibutuhkan oleh sebuah *flight controller*.

BAB VI PENUTUP

6.1 Kesimpulan

Telah dilakukan pengujian dan analisa pembacaan sensor *accelerometer*, *gyroscope* dan modul GPS menggunakan mikrokontroler STM32F411, dan didapatkan kesimpulan sebagai berikut:

1. Hasil pembacaan dari sensor *accelerometer*, *gyroscope* dan modul GPS menggunakan mikrokontroler STM32F411 ditampilkan melalui *tools serial monitor* pada *software* Arduino IDE. Data yang ditampilkan pada sensor ADXL345 berupa nilai percepatan pada tiga arah, sensor GY-25 menampilkan data berupa derajat kemiringan, sensor IMU GY-521(MPU6050) menampilkan nilai akselerasi dan giroskop dan modul GPS menampilkan data koordinat lokasi.
2. Berdasarkan analisa dari hasil pembacaan sensor, sensor menunjukkan kinerja yang baik karena mampu melakukan pembacaan nilai sesuai dengan perintah pada program. Sensor ADXL345 digunakan untuk melakukan pembacaan nilai percepatan linier dalam tiga sumbu (x, y dan z), sensor *gyroscope* GY-25 digunakan untuk mengukur derajat kemiringan layaknya kompas, sensor IMU GY-521(MPU6050) melakukan pengukuran nilai percepatan dan *gyroscope*, serta Modul GPS yang digunakan untuk melakukan pembacaan posisi titik koordinat dan *altitude*.
3. Berdasarkan hasil pengujian dengan melakukan pembacaan sensor, maka dapat disimpulkan bahwa mikrokontroler STM32F411 berpotensi dapat dijadikan sebagai alternatif *flight controller* pada drone karena memiliki spesifikasi yang mumpuni sesuai dengan yang dibutuhkan oleh sebuah *flight controller*.

6.2 Saran

Berdasarkan penelitian yang telah dilakukan, terdapat beberapa kekurangan sehingga diperlukan pengembangan lebih lanjut. Maka saran untuk penelitian selanjutnya adalah sebagai berikut:

1. Perlu ditambahkan komponen sensor – sensor lain yang diperlukan untuk sebuah *flight controller* pada drone seperti sensor barometer, sensor magnetometer, sensor ultrasonik, dan sensor yang lain.
2. Hasil pembacaan sensor pada penelitian ini masih dilakukan secara terpisah sehingga selanjutnya perlu dilakukan penggabungan beberapa sensor agar hasil pembacaan sensor menjadi satu dan pada codingan program diberi perintah untuk melakukan pembacaan sensor secara bersamaan.

DAFTAR PUSTAKA

- Achlison, U. 2021. *Prinsip Kerja Drone*. Diakses pada tanggal 12 September 2023 melalui <https://teknik-elektronika-d3.stekom.ac.id/informasi/baca/PRINSIP-KERJA-DRONE/00b7858a85fe7328455145a1552e17378e980fe7>
- Hutama, M. 2015. *Prinsip Kerja Quadcopter*. Diakses pada tanggal 12 September 2023 melalui <http://www.insinyoer.com/prinsip-kerja-quadcopter>
- Indreswari, S. 2016. *Peran Drone/Unmanned Aerial Vehicle (UAV) buatan STTKD dalam dunia penerbangan*. Prosiding Seminar Nasional "Pengembangan Kebijakan dan Regulasi Penerbangan dan Antariksa : Problema dan Tantangan". pp. 104- 112
- Pangestu, M. I. 2018. *Penggunaan MultiWii Sebagai Flight Control Pada Voice Control Quadcopter*. Tugas Akhir, Universitas Sriwijaya
- Riyadi, M. dkk. 2011. *Pendeteksi Posisi Menggunakan Sensor Accelerometer MA7260Q Berbasis Mikrokontroler Atmega 32*. Undergraduate thesis, Diponegoro University
- Setiawan, A. dkk. 2017. *Perancangan Sistem Kontrol Pesawat Model Tanpa Awak (Unmanned Aerial Vehicle) Berbasis Arduino*. Prosiding Seminar Nasional Inovasi Teknologi Penerbangan (SNITP), ISSN : 2548-8112. ATKP Surabaya

LAMPIRAN

- Lampiran 1. Listing Program Sensor *Accelerometer* ADXL345 di *Software* Arduino IDE

```
#include <Wire.h>

void setup() {
  Serial.begin(115200);
  Wire.begin();

  Wire.beginTransmission(0x53);
  Wire.write(0x2C);
  Wire.write(0x08);
  Wire.endTransmission();

  Wire.beginTransmission(0x53);
  Wire.write(0x31);
  Wire.write(0x08);
  Wire.endTransmission();

  Wire.beginTransmission(0x53);
  Wire.write(0x2D);
  Wire.write(0x08);
  Wire.endTransmission();
}

void loop() {
  Wire.beginTransmission(0x53);
  Wire.write(0x32);
  Wire.endTransmission();
  Wire.requestFrom(0x53, 1);
  byte x0 = Wire.read();

  Wire.beginTransmission(0x53);
  Wire.write(0x33);
  Wire.endTransmission();
  Wire.requestFrom(0x53, 1);
  byte x1 = Wire.read();
  x1 = x1 & 0x03;

  uint16_t x = (x1 << 8) + x0;
  int16_t xf = x;
  if(xf > 511)
  {
    xf = xf - 1024;
  }
  float xa = xf * 0.004;
  Serial.print("Sumbu X = ");
```

```

Serial.print(xa);
Serial.print(" g(m/s^2)");
Serial.println();

Wire.beginTransmission(0x53);
Wire.write(0x34);
Wire.endTransmission();
Wire.requestFrom(0x53, 1);
byte y0 = Wire.read();

Wire.beginTransmission(0x53);
Wire.write(0x35);
Wire.endTransmission();
Wire.requestFrom(0x53, 1);
byte y1 = Wire.read();
y1 = y1 & 0x03;

uint16_t y = (y1 << 8) + y0;
int16_t yf = y;
if(yf > 511)
{
yf = yf - 1024;
}
float ya = yf * 0.004;
Serial.print("Sumbu Y = ");
Serial.print(ya);
Serial.print(" g(m/s^2)");
Serial.println();

Wire.beginTransmission(0x53);
Wire.write(0x36);
Wire.endTransmission();
Wire.requestFrom(0x53, 1);
byte z0 = Wire.read();

Wire.beginTransmission(0x53);
Wire.write(0x37);
Wire.endTransmission();
Wire.requestFrom(0x53, 1);
byte z1 = Wire.read();
z1 = z1 & 0x03;

uint16_t z = (z1 << 8) + z0;
int16_t zf = z;
if(zf > 511)
{
zf = zf - 1024;
}

```

```

}
float za = zf * 0.004;
Serial.print("Sumbu Z = ");
Serial.print(za);
Serial.print(" g(m/s^2)");
Serial.println();
Serial.println();
delay(500);
}

```

- **Lampiran 2.** Listing Program Sensor *Gyroscope GY-25* di *Software Arduino IDE*

```

String dataInIMU;
String dataIMU[10];
int i;
boolean parsing = false;
float hadap;

void setup() {
  Serial.begin(115200);

  Serial.print("Calibrating IMU\n");
  Serial.print("Please wait...\n");

  dataInIMU="";

  // correction mode, Kalibrasi roll pitch, tahan beberapa detik sebelum melakukan kalibrasi
  Serial.write(0xA5);
  Serial.write(0x54);

  delay(2000); // Jeda sebelum kalibrasi heading

  // correction mode, kalibrasi kompas. Kompas akan diset ke 0 ketika perintah ini dikirim
  Serial.write(0xA5);
  Serial.write(0x55);

  delay(100); // Jeda sebelum konfigurasi output

  // Kirim data sensor secara realtime dalam bentuk ASCII
  Serial.write(0xA5);
  Serial.write(0x53);

  delay(1000); // Jeda sebentar
}

void loop() {
  if (Serial.available() > 0) {

```

```

char inChar = (char)Serial.read();
dataInIMU += inChar;
if (inChar == '\n') {
  parsing = true;
}
}

if (parsing) {
  parsingData();
  parsing = false;
  dataInIMU = "";
}
}

void parsingData() {
  int j = 0;

  dataIMU[j] = "";
  //proses parsing data
  for (i = 1; i < dataInIMU.length(); i++) {
    // pengecekan tiap karakter dengan karakter (#) (,) (=)
    if ((dataInIMU[i] == '#') || (dataInIMU[i] == ',') || (dataInIMU[i] == '='))
    {
      // increment variabel j, digunakan untuk merubah index array penampung
      j++;
      dataIMU[j] = ""; //inisialisasi variabel array dataIMU[j]
    }
    else
    {
      //proses tampung data saat pengecekan karakter selesai.
      dataIMU[j] = dataIMU[j] + dataIMU[i];
    }
  }
}

hadap = dataIMU[1].toFloat();
if (hadap > 0) {
  float selisih = 180 - hadap;
  hadap = 180 + selisih;
  if (hadap > 360) {
    hadap = 0;
  }
}

if (hadap < 0) {
  hadap = hadap * -1;
}

Serial.print("Arah Hadap : ");

```

```

Serial.print(hadap);
Serial.print(" ");
Serial.print("\n");
}

```

- **Lampiran 3.** Listing Program Sensor IMU GY-521(MPU-6050) di *Software* Arduino IDE

```

#include "Wire.h" // This library allows you to communicate with I2C devices.

const int MPU_ADDR = 0x68; // I2C address of the MPU-6050. If AD0 pin is set to HIGH, the I2C
address will be 0x69.

int16_t accelerometer_x, accelerometer_y, accelerometer_z; // variables for accelerometer raw data
int16_t gyro_x, gyro_y, gyro_z; // variables for gyro raw data
int16_t temperature; // variables for temperature data

char tmp_str[7]; // temporary variable used in convert function

char* convert_int16_to_str(int16_t i) { // converts int16 to string. Moreover, resulting strings will
have the same length in the debug monitor.
    sprintf(tmp_str, "%6d", i);
    return tmp_str;
}

void setup() {
    Serial.begin(9600);
    Wire.begin();
    Wire.beginTransmission(MPU_ADDR); // Begins a transmission to the I2C slave (GY-521 board)
    Wire.write(0x6B); // PWR_MGMT_1 register
    Wire.write(0); // set to zero (wakes up the MPU-6050)
    Wire.endTransmission(true);
}

void loop() {
    Wire.beginTransmission(MPU_ADDR);
    Wire.write(0x3B); // starting with register 0x3B (ACCEL_XOUT_H) [MPU-6000 and MPU-6050
Register Map and Descriptions Revision 4.2, p.40]
    Wire.endTransmission(false); // the parameter indicates that the Arduino will send a restart. As a
result, the connection is kept active.
    Wire.requestFrom(MPU_ADDR, 7*2, true); // request a total of 7*2=14 registers

    // "Wire.read()<<8 | Wire.read();" means two registers are read and stored in the same variable
    accelerometer_x = Wire.read()<<8 | Wire.read(); // reading registers: 0x3B (ACCEL_XOUT_H)
and 0x3C (ACCEL_XOUT_L)
    accelerometer_y = Wire.read()<<8 | Wire.read(); // reading registers: 0x3D (ACCEL_YOUT_H)
and 0x3E (ACCEL_YOUT_L)
    accelerometer_z = Wire.read()<<8 | Wire.read(); // reading registers: 0x3F (ACCEL_ZOUT_H)
and 0x40 (ACCEL_ZOUT_L)
}

```

```

temperature = Wire.read()<<8 | Wire.read(); // reading registers: 0x41 (TEMP_OUT_H) and 0x42
(TEMP_OUT_L)
gyro_x = Wire.read()<<8 | Wire.read(); // reading registers: 0x43 (GYRO_XOUT_H) and 0x44
(GYRO_XOUT_L)
gyro_y = Wire.read()<<8 | Wire.read(); // reading registers: 0x45 (GYRO_YOUT_H) and 0x46
(GYRO_YOUT_L)
gyro_z = Wire.read()<<8 | Wire.read(); // reading registers: 0x47 (GYRO_ZOUT_H) and 0x48
(GYRO_ZOUT_L)

// print out data
Serial.print("aX = "); Serial.print(convert_int16_to_str(accelerometer_x));
Serial.print(" | aY = "); Serial.print(convert_int16_to_str(accelerometer_y));
Serial.print(" | aZ = "); Serial.print(convert_int16_to_str(accelerometer_z));
// the following equation was taken from the documentation [MPU-6000/MPU-6050 Register Map
and Description, p.30]
Serial.print(" | tmp = "); Serial.print(temperature/340.00+36.53);
Serial.print(" | gX = "); Serial.print(convert_int16_to_str(gyro_x));
Serial.print(" | gY = "); Serial.print(convert_int16_to_str(gyro_y));
Serial.print(" | gZ = "); Serial.print(convert_int16_to_str(gyro_z));
Serial.println();

// delay
delay(1000);
}

```

- **Lampiran 4.** Listing Program Modul GPS di *Software* Arduino IDE

```

#include <TinyGPS++.h>
#include <SoftwareSerial.h>

static const int RXPin = 3, TXPin = 4;
static const uint32_t GPSBaud = 9600;

// The TinyGPS++ object
TinyGPSPlus gps;

// The serial connection to the GPS device
SoftwareSerial ss(RXPin, TXPin);

void setup()
{
  Serial.begin(115200);
  ss.begin(GPSBaud);

  Serial.println(F("A simple demonstration of TinyGPS++ with an attached GPS module"));
  Serial.print(F("Testing TinyGPS++ library v. ")); Serial.println(TinyGPSPlus::libraryVersion());
  Serial.println(F("by Mikal Hart"));
  Serial.println();
}

```

```

}

void loop()
{
// This sketch displays information every time a new sentence is correctly encoded.
while (ss.available() > 0)
if (gps.encode(ss.read()))
displayInfo();

if (millis() > 5000 && gps.charsProcessed() < 10)
{
Serial.println(F("No GPS detected: check wiring."));
while(true);
}
delay(100);
}

void displayInfo()
{
Serial.print(F("Location: "));
if (gps.location.isValid())
{
Serial.print(gps.location.lat(), 6);
Serial.print(F(", "));
Serial.print(gps.location.lng(), 6);

Serial.print(F("- altitude: "));
if (gps.altitude.isValid())
Serial.println(gps.altitude.meters());
}
else
{
Serial.print(F("INVALID"));
}

Serial.print(F(" Date/Time: "));
if (gps.date.isValid())
{
Serial.print(gps.date.month());
Serial.print(F("/"));
Serial.print(gps.date.day());
Serial.print(F("/"));
Serial.print(gps.date.year());
}
else
{
Serial.print(F("INVALID"));
}
}

```

```

}

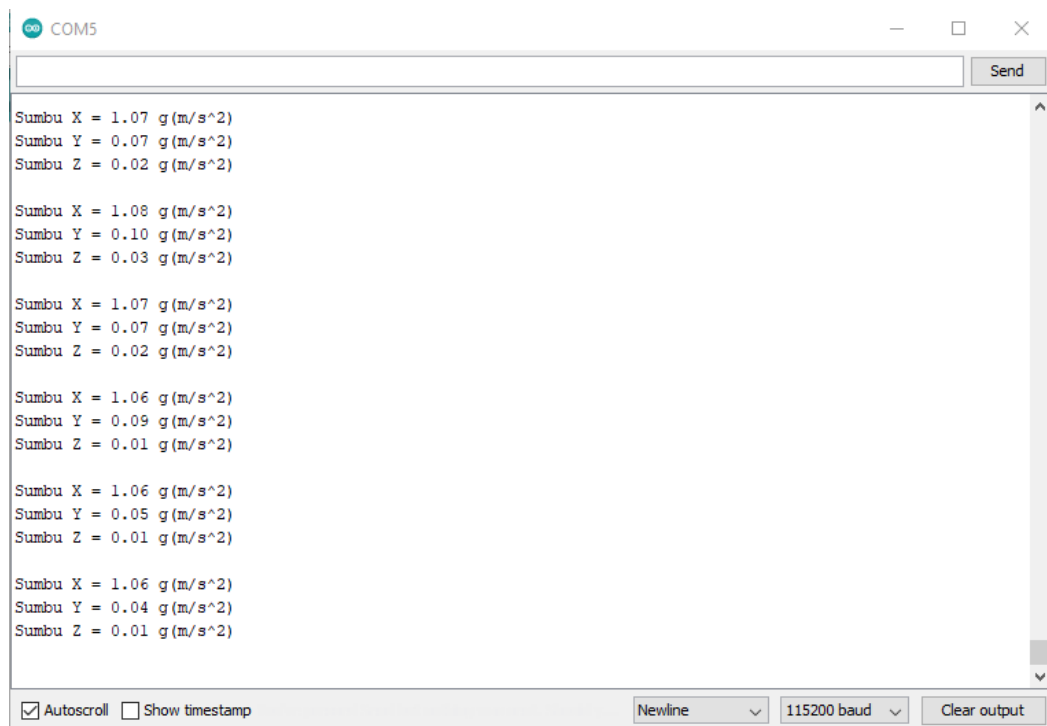
Serial.print(F(" "));
if (gps.time.isValid())
{
if (gps.time.hour() < 10) Serial.print(F("0"));
Serial.print(gps.time.hour());
Serial.print(F(":"));
if (gps.time.minute() < 10) Serial.print(F("0"));
Serial.print(gps.time.minute());
Serial.print(F(":"));
if (gps.time.second() < 10) Serial.print(F("0"));
Serial.print(gps.time.second());
Serial.print(F("."));
if (gps.time.centisecond() < 10) Serial.print(F("0"));
Serial.print(gps.time.centisecond());
}
else
{
Serial.print(F("INVALID"));
}

Serial.println();
}

```

- **Lampiran 5.** Screenshot Data Hasil Pembacaan Sensor *accelerometer* ADXL345 di *software* Arduino IDE

Sumbu x



Sumbu -x

COM5

Send

Sumbu X = -0.96 g (m/s²)
Sumbu Y = 0.01 g (m/s²)
Sumbu Z = 0.22 g (m/s²)

Sumbu X = -0.95 g (m/s²)
Sumbu Y = 0.03 g (m/s²)
Sumbu Z = 0.24 g (m/s²)

Sumbu X = -0.99 g (m/s²)
Sumbu Y = 0.01 g (m/s²)
Sumbu Z = 0.22 g (m/s²)

Sumbu X = -0.95 g (m/s²)
Sumbu Y = 0.01 g (m/s²)
Sumbu Z = 0.22 g (m/s²)

Sumbu X = -0.95 g (m/s²)
Sumbu Y = 0.02 g (m/s²)
Sumbu Z = 0.24 g (m/s²)

Sumbu X = -0.95 g (m/s²)
Sumbu Y = 0.00 g (m/s²)
Sumbu Z = 0.21 g (m/s²)

Autoscroll Show timestamp

Newline 115200 baud Clear output

Sumbu y

COM5

Send

Sumbu X = 0.09 g (m/s²)
Sumbu Y = 1.04 g (m/s²)
Sumbu Z = 0.01 g (m/s²)

Sumbu X = 0.09 g (m/s²)
Sumbu Y = 1.04 g (m/s²)
Sumbu Z = 0.01 g (m/s²)

Sumbu X = 0.08 g (m/s²)
Sumbu Y = 1.04 g (m/s²)
Sumbu Z = 0.01 g (m/s²)

Sumbu X = 0.08 g (m/s²)
Sumbu Y = 1.04 g (m/s²)
Sumbu Z = 0.01 g (m/s²)

Sumbu X = 0.09 g (m/s²)
Sumbu Y = 1.03 g (m/s²)
Sumbu Z = 0.02 g (m/s²)

Sumbu X = 0.09 g (m/s²)
Sumbu Y = 1.04 g (m/s²)
Sumbu Z = 0.01 g (m/s²)

Autoscroll Show timestamp

Newline 115200 baud Clear output

Sumbu -y

The screenshot shows a terminal window titled "COM5" with a "Send" button in the top right. The main area contains six groups of three lines each, representing acceleration data for the Y-axis. Each group shows Sumbu X, Sumbu Y, and Sumbu Z values in g(m/s^2). The values for Sumbu Y are consistently around -1.05 to -1.06 g(m/s^2). At the bottom, there are checkboxes for "Autoscroll" (checked) and "Show timestamp" (unchecked), along with dropdown menus for "Newline" and "115200 baud", and a "Clear output" button.

```
Sumbu X = 0.07 g(m/s^2)
Sumbu Y = -1.05 g(m/s^2)
Sumbu Z = 0.18 g(m/s^2)

Sumbu X = 0.06 g(m/s^2)
Sumbu Y = -1.06 g(m/s^2)
Sumbu Z = 0.17 g(m/s^2)

Sumbu X = 0.07 g(m/s^2)
Sumbu Y = -1.06 g(m/s^2)
Sumbu Z = 0.16 g(m/s^2)

Sumbu X = 0.06 g(m/s^2)
Sumbu Y = -1.05 g(m/s^2)
Sumbu Z = 0.16 g(m/s^2)

Sumbu X = 0.06 g(m/s^2)
Sumbu Y = -1.05 g(m/s^2)
Sumbu Z = 0.17 g(m/s^2)

Sumbu X = 0.07 g(m/s^2)
Sumbu Y = -1.05 g(m/s^2)
Sumbu Z = 0.16 g(m/s^2)
```

Autoscroll Show timestamp Newline 115200 baud Clear output

Sumbu z

The screenshot shows a terminal window titled "COM5" with a "Send" button in the top right. The main area contains six groups of three lines each, representing acceleration data for the Z-axis. Each group shows Sumbu X, Sumbu Y, and Sumbu Z values in g(m/s^2). The values for Sumbu Z are consistently around 1.03 to 1.04 g(m/s^2). At the bottom, there are checkboxes for "Autoscroll" (checked) and "Show timestamp" (unchecked), along with dropdown menus for "Newline" and "115200 baud", and a "Clear output" button.

```
Sumbu X = 0.04 g(m/s^2)
Sumbu Y = 0.20 g(m/s^2)
Sumbu Z = 1.04 g(m/s^2)

Sumbu X = 0.04 g(m/s^2)
Sumbu Y = 0.21 g(m/s^2)
Sumbu Z = 1.04 g(m/s^2)

Sumbu X = 0.04 g(m/s^2)
Sumbu Y = 0.20 g(m/s^2)
Sumbu Z = 1.04 g(m/s^2)

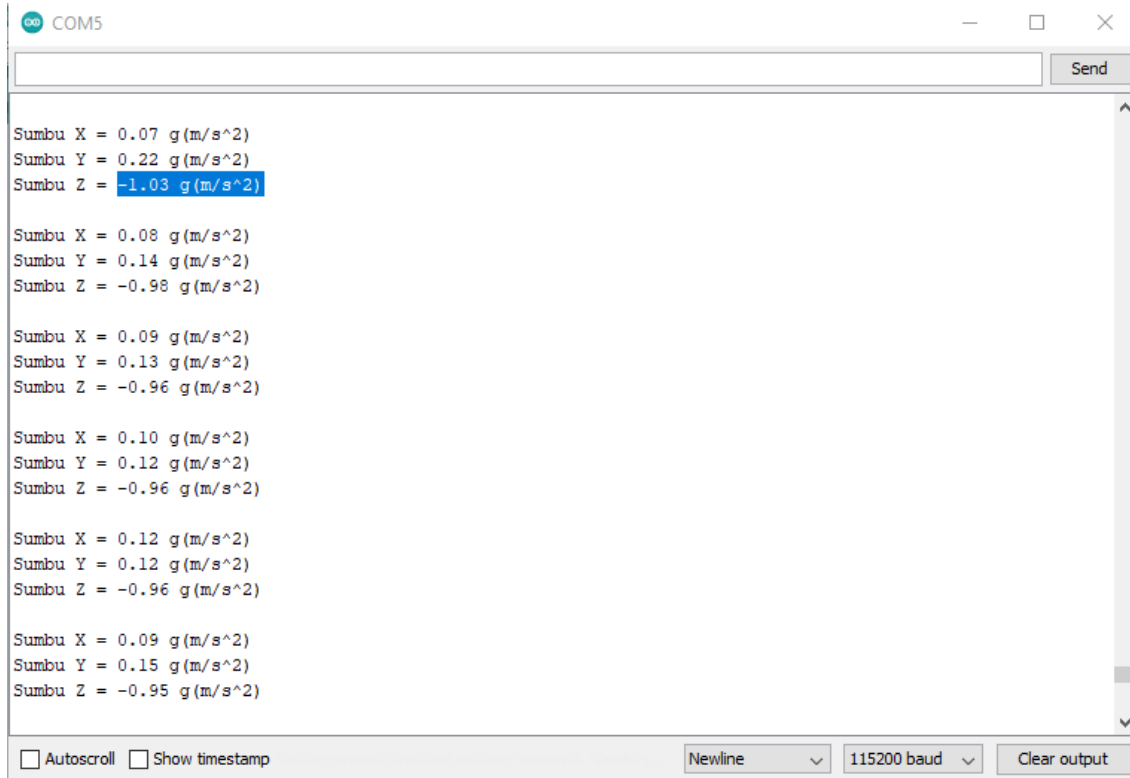
Sumbu X = 0.04 g(m/s^2)
Sumbu Y = 0.21 g(m/s^2)
Sumbu Z = 1.03 g(m/s^2)

Sumbu X = 0.04 g(m/s^2)
Sumbu Y = 0.20 g(m/s^2)
Sumbu Z = 1.04 g(m/s^2)

Sumbu X = 0.04 g(m/s^2)
Sumbu Y = 0.21 g(m/s^2)
Sumbu Z = 1.04 g(m/s^2)
```

Autoscroll Show timestamp Newline 115200 baud Clear output

Sumbu -z



COM5

```
Sumbu X = 0.07 g(m/s^2)
Sumbu Y = 0.22 g(m/s^2)
Sumbu Z = -1.03 g(m/s^2)

Sumbu X = 0.08 g(m/s^2)
Sumbu Y = 0.14 g(m/s^2)
Sumbu Z = -0.98 g(m/s^2)

Sumbu X = 0.09 g(m/s^2)
Sumbu Y = 0.13 g(m/s^2)
Sumbu Z = -0.96 g(m/s^2)

Sumbu X = 0.10 g(m/s^2)
Sumbu Y = 0.12 g(m/s^2)
Sumbu Z = -0.96 g(m/s^2)

Sumbu X = 0.12 g(m/s^2)
Sumbu Y = 0.12 g(m/s^2)
Sumbu Z = -0.96 g(m/s^2)

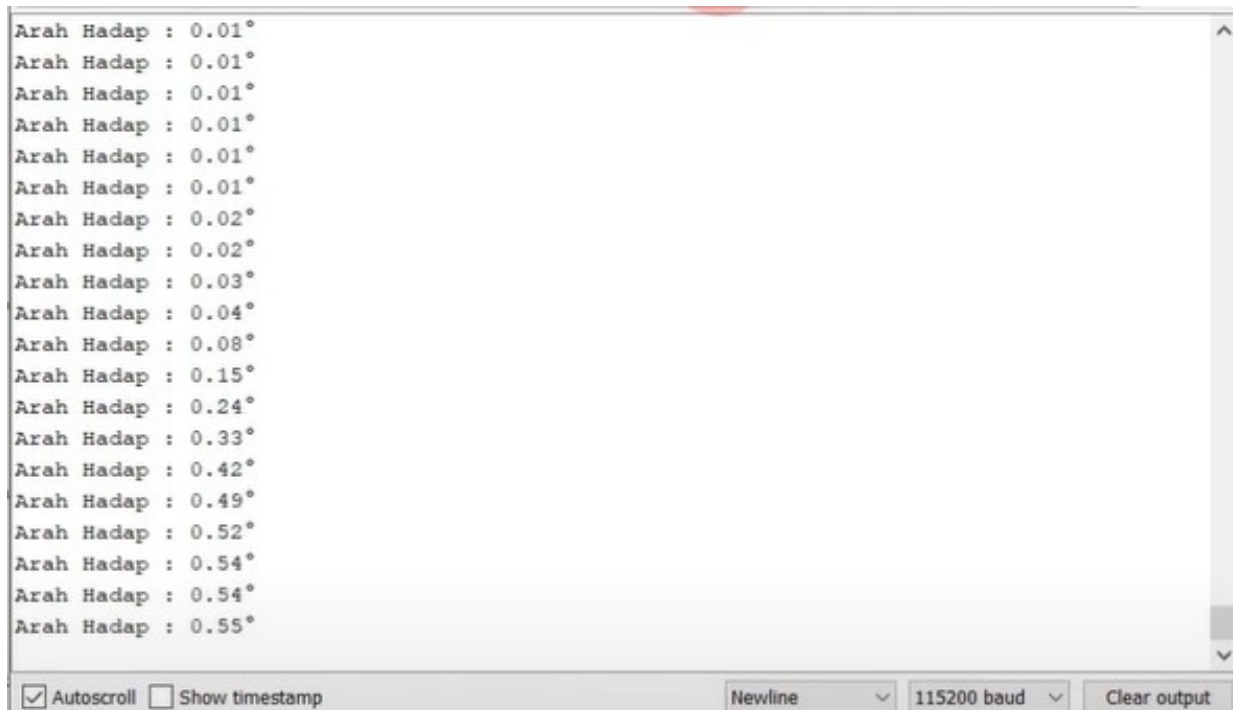
Sumbu X = 0.09 g(m/s^2)
Sumbu Y = 0.15 g(m/s^2)
Sumbu Z = -0.95 g(m/s^2)
```

Autoscroll Show timestamp

Newline 115200 baud Clear output

- **Lampiran 6.** Screenshot Data Hasil Pembacaan Sensor Gyroscope GY-25 di software Arduino IDE

Arah Sudut 0°



```
Arah Hadap : 0.01°
Arah Hadap : 0.01°
Arah Hadap : 0.01°
Arah Hadap : 0.01°
Arah Hadap : 0.01°
Arah Hadap : 0.02°
Arah Hadap : 0.02°
Arah Hadap : 0.03°
Arah Hadap : 0.04°
Arah Hadap : 0.08°
Arah Hadap : 0.15°
Arah Hadap : 0.24°
Arah Hadap : 0.33°
Arah Hadap : 0.42°
Arah Hadap : 0.49°
Arah Hadap : 0.52°
Arah Hadap : 0.54°
Arah Hadap : 0.54°
Arah Hadap : 0.55°
```

Autoscroll Show timestamp

Newline 115200 baud Clear output

Arah Sudut 45°

```
Arah Hadap : 45.97°
Arah Hadap : 46.67°
Arah Hadap : 47.44°
Arah Hadap : 48.21°
Arah Hadap : 48.87°
Arah Hadap : 49.50°
Arah Hadap : 50.02°
Arah Hadap : 50.49°
Arah Hadap : 50.91°
Arah Hadap : 51.25°
Arah Hadap : 51.52°
Arah Hadap : 51.71°
Arah Hadap : 51.81°
Arah Hadap : 51.87°
Arah Hadap : 51.93°
Arah Hadap : 52.01°
Arah Hadap : 52.09°
Arah Hadap : 52.23°
Arah Hadap : 52.35°
Arah Hadap : 52.47°
```

Autoscroll Show timestamp Newline 115200 baud Clear output

Arah Sudut 90°

```
Arah Hadap : 89.93°
Arah Hadap : 90.05°
Arah Hadap : 90.15°
Arah Hadap : 90.26°
Arah Hadap : 90.36°
Arah Hadap : 90.44°
Arah Hadap : 90.51°
Arah Hadap : 90.58°
Arah Hadap : 90.62°
Arah Hadap : 90.66°
Arah Hadap : 90.71°
Arah Hadap : 90.76°
Arah Hadap : 90.82°
Arah Hadap : 90.88°
Arah Hadap : 90.95°
Arah Hadap : 91.04°
Arah Hadap : 91.12°
Arah Hadap : 91.21°
Arah Hadap : 95.14°
Arah Hadap : 95.14°
```

Autoscroll Show timestamp Newline 115200 baud Clear output

- **Lampiran 7.** Screenshot Data Hasil Pembacaan Sensor IMU GY-521 di software Arduino IDE

Sumbu x

```

COM5
Send
aX = 17528 | aY = 496 | aZ = -6288 | tmp = 30.27 | gX = 3 | gY = 240 | gZ = 161
aX = 16840 | aY = 260 | aZ = -6384 | tmp = 30.32 | gX = -424 | gY = 336 | gZ = 186
aX = 16804 | aY = 496 | aZ = -5848 | tmp = 30.27 | gX = -51 | gY = -436 | gZ = 238
aX = 17076 | aY = 768 | aZ = -6288 | tmp = 30.32 | gX = -443 | gY = -301 | gZ = 206
aX = 16840 | aY = -220 | aZ = -5656 | tmp = 30.32 | gX = 1228 | gY = 1497 | gZ = 403
aX = 17332 | aY = -436 | aZ = -2572 | tmp = 30.32 | gX = -476 | gY = 750 | gZ = 544
aX = 17000 | aY = -468 | aZ = -2588 | tmp = 30.37 | gX = -235 | gY = -423 | gZ = 306
aX = 17048 | aY = -372 | aZ = -2440 | tmp = 30.46 | gX = -420 | gY = -116 | gZ = 135
aX = 17160 | aY = -500 | aZ = -2352 | tmp = 30.46 | gX = -396 | gY = -129 | gZ = 176
aX = 17144 | aY = -532 | aZ = -3112 | tmp = 30.41 | gX = -281 | gY = 74 | gZ = 106
aX = 17220 | aY = -648 | aZ = -2528 | tmp = 30.41 | gX = 247 | gY = 141 | gZ = 418
aX = 16932 | aY = -716 | aZ = -2904 | tmp = 30.46 | gX = -403 | gY = 126 | gZ = 258
aX = 17100 | aY = -504 | aZ = -2736 | tmp = 30.51 | gX = -686 | gY = 115 | gZ = 48
aX = 17044 | aY = -1016 | aZ = -2744 | tmp = 30.46 | gX = -592 | gY = 107 | gZ = 211
aX = 17344 | aY = -1176 | aZ = -2556 | tmp = 30.46 | gX = -437 | gY = -127 | gZ = 232
aX = 17296 | aY = -916 | aZ = -3316 | tmp = 30.46 | gX = 104 | gY = 275 | gZ = 281
aX = 16912 | aY = -1268 | aZ = -3108 | tmp = 30.51 | gX = -368 | gY = -64 | gZ = 218
aX = 17200 | aY = -796 | aZ = -2724 | tmp = 30.46 | gX = -330 | gY = -206 | gZ = 314
aX = 17148 | aY = -784 | aZ = -3120 | tmp = 30.60 | gX = -287 | gY = 24 | gZ = 235
aX = 17040 | aY = -940 | aZ = -3124 | tmp = 30.55 | gX = -366 | gY = -2 | gZ = 244
aX = 17280 | aY = -748 | aZ = -3152 | tmp = 30.60 | gX = 159 | gY = 55 | gZ = 285
aX = 17044 | aY = -804 | aZ = -3140 | tmp = 30.65 | gX = -422 | gY = -121 | gZ = 216
aX = 16972 | aY = -1212 | aZ = -3000 | tmp = 30.65 | gX = -534 | gY = -60 | gZ = 129
aX = 17780 | aY = -232 | aZ = -3152 | tmp = 30.65 | gX = -232 | gY = -53 | gZ = -44
aX = 17140 | aY = -872 | aZ = -3772 | tmp = 30.55 | gX = -403 | gY = -52 | gZ = 241
Autoscroll Show timestamp Newline 115200 baud Clear output

```

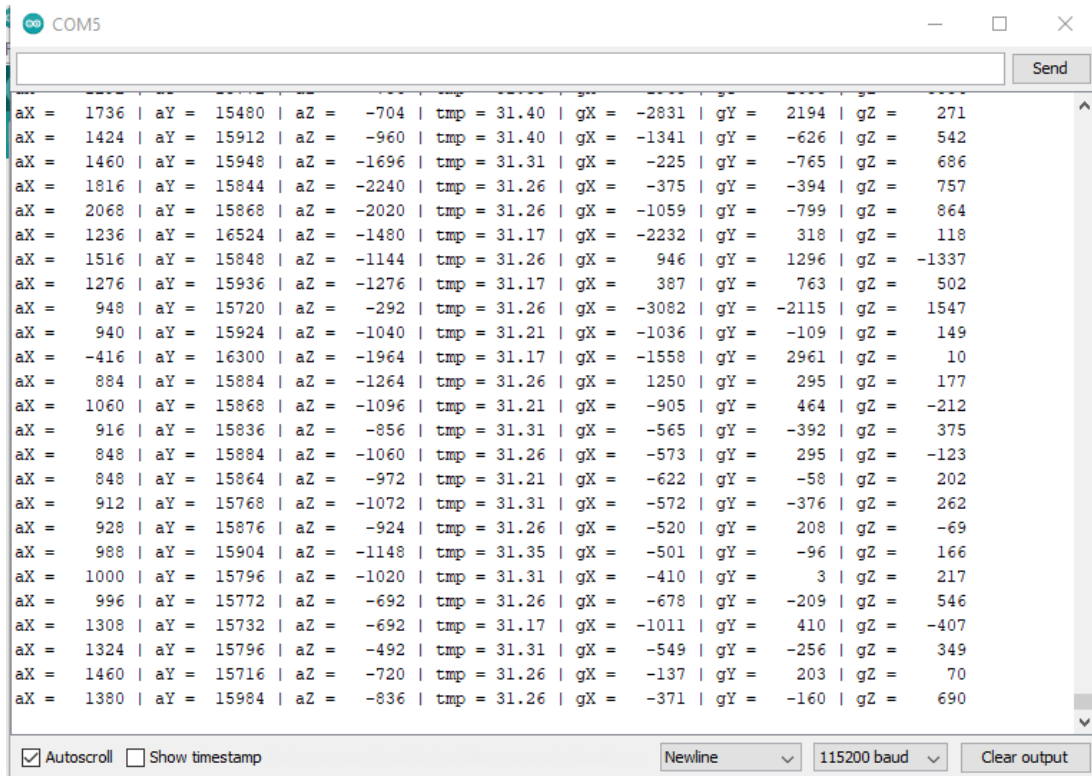
Sumbu -x

```

COM5
Send
aX = -13684 | aY = -3868 | aZ = 3328 | tmp = 30.65 | gX = -677 | gY = -174 | gZ = 402
aX = -13596 | aY = -3724 | aZ = 3552 | tmp = 30.60 | gX = 10 | gY = 277 | gZ = 147
aX = -13532 | aY = -3952 | aZ = 3576 | tmp = 30.69 | gX = -13 | gY = 408 | gZ = 177
aX = -13628 | aY = -4088 | aZ = 3008 | tmp = 30.69 | gX = -1755 | gY = -678 | gZ = 255
aX = -13656 | aY = -4016 | aZ = 3208 | tmp = 30.74 | gX = -119 | gY = 228 | gZ = 187
aX = -13692 | aY = -3944 | aZ = 3136 | tmp = 30.69 | gX = -497 | gY = 179 | gZ = 96
aX = -13744 | aY = -3856 | aZ = 3464 | tmp = 30.65 | gX = -72 | gY = -9 | gZ = 176
aX = -13740 | aY = -3988 | aZ = 3256 | tmp = 30.74 | gX = -385 | gY = -946 | gZ = 587
aX = -13760 | aY = -3980 | aZ = 2944 | tmp = 30.69 | gX = -607 | gY = -2 | gZ = 268
aX = -13872 | aY = -3932 | aZ = 2940 | tmp = 30.69 | gX = -1630 | gY = -779 | gZ = 233
aX = -13916 | aY = -3968 | aZ = 2704 | tmp = 30.79 | gX = -650 | gY = -176 | gZ = 241
aX = -13888 | aY = -3932 | aZ = 2876 | tmp = 30.79 | gX = -383 | gY = 110 | gZ = 276
aX = -13952 | aY = -3904 | aZ = 3084 | tmp = 30.79 | gX = -3212 | gY = -3012 | gZ = -2548
aX = -14268 | aY = -816 | aZ = 2764 | tmp = 30.74 | gX = -963 | gY = 5803 | gZ = -97
aX = -15492 | aY = -884 | aZ = -4328 | tmp = 30.74 | gX = -1922 | gY = -1826 | gZ = 2054
aX = -15548 | aY = -208 | aZ = -3656 | tmp = 30.84 | gX = -345 | gY = -1354 | gZ = 228
aX = -15488 | aY = -220 | aZ = -2656 | tmp = 30.88 | gX = -338 | gY = -167 | gZ = 235
aX = -15396 | aY = -216 | aZ = -1972 | tmp = 30.74 | gX = -351 | gY = -1623 | gZ = 201
aX = -15416 | aY = -284 | aZ = -1432 | tmp = 30.84 | gX = -333 | gY = 264 | gZ = 215
aX = -15472 | aY = -300 | aZ = -1784 | tmp = 30.69 | gX = -1220 | gY = 693 | gZ = 255
aX = -15396 | aY = -152 | aZ = -1720 | tmp = 30.84 | gX = -336 | gY = -450 | gZ = 193
aX = -15364 | aY = -212 | aZ = -1464 | tmp = 30.79 | gX = -339 | gY = 23 | gZ = 206
aX = -15412 | aY = -348 | aZ = -1408 | tmp = 30.79 | gX = -329 | gY = -484 | gZ = 172
aX = -15428 | aY = -220 | aZ = -1064 | tmp = 30.79 | gX = -325 | gY = 168 | gZ = 216
aX = -15248 | aY = -176 | aZ = -1344 | tmp = 30.74 | gX = -349 | gY = 270 | gZ = 217
Autoscroll Show timestamp Newline 115200 baud Clear output

```

Sumbu y

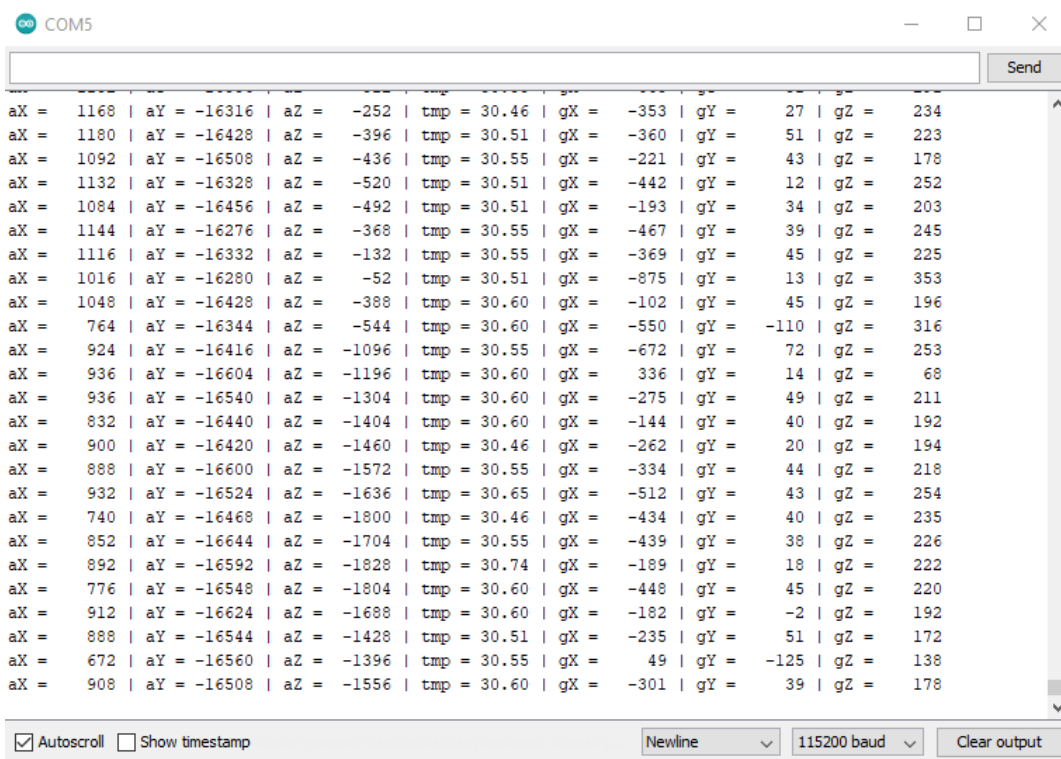


The screenshot shows a terminal window titled 'COM5' with a 'Send' button. The output consists of 30 lines of sensor data. Each line contains 10 values: aX, aY, aZ, tmp, gX, gY, and gZ. The data is as follows:

aX	aY	aZ	tmp	gX	gY	gZ
1736	15480	-704	31.40	-2831	2194	271
1424	15912	-960	31.40	-1341	-626	542
1460	15948	-1696	31.31	-225	-765	686
1816	15844	-2240	31.26	-375	-394	757
2068	15868	-2020	31.26	-1059	-799	864
1236	16524	-1480	31.17	-2232	318	118
1516	15848	-1144	31.26	946	1296	-1337
1276	15936	-1276	31.17	387	763	502
948	15720	-292	31.26	-3082	-2115	1547
940	15924	-1040	31.21	-1036	-109	149
-416	16300	-1964	31.17	-1558	2961	10
884	15884	-1264	31.26	1250	295	177
1060	15868	-1096	31.21	-905	464	-212
916	15836	-856	31.31	-565	-392	375
848	15884	-1060	31.26	-573	295	-123
848	15864	-972	31.21	-622	-58	202
912	15768	-1072	31.31	-572	-376	262
928	15876	-924	31.26	-520	208	-69
988	15904	-1148	31.35	-501	-96	166
1000	15796	-1020	31.31	-410	3	217
996	15772	-692	31.26	-678	-209	546
1308	15732	-692	31.17	-1011	410	-407
1324	15796	-492	31.31	-549	-256	349
1460	15716	-720	31.26	-137	203	70
1380	15984	-836	31.26	-371	-160	690

At the bottom of the window, there are controls: Autoscroll, Show timestamp, a dropdown menu set to 'Newline', a dropdown menu set to '115200 baud', and a 'Clear output' button.

Sumbu -y



The screenshot shows a terminal window titled 'COM5' with a 'Send' button. The output consists of 30 lines of sensor data. Each line contains 10 values: aX, aY, aZ, tmp, gX, gY, and gZ. The data is as follows:

aX	aY	aZ	tmp	gX	gY	gZ
1168	-16316	-252	30.46	-353	27	234
1180	-16428	-396	30.51	-360	51	223
1092	-16508	-436	30.55	-221	43	178
1132	-16328	-520	30.51	-442	12	252
1084	-16456	-492	30.51	-193	34	203
1144	-16276	-368	30.55	-467	39	245
1116	-16332	-132	30.55	-369	45	225
1016	-16280	-52	30.51	-875	13	353
1048	-16428	-388	30.60	-102	45	196
764	-16344	-544	30.60	-550	-110	316
924	-16416	-1096	30.55	-672	72	253
936	-16604	-1196	30.60	336	14	68
936	-16540	-1304	30.60	-275	49	211
832	-16440	-1404	30.60	-144	40	192
900	-16420	-1460	30.46	-262	20	194
888	-16600	-1572	30.55	-334	44	218
932	-16524	-1636	30.65	-512	43	254
740	-16468	-1800	30.46	-434	40	235
852	-16644	-1704	30.55	-439	38	226
892	-16592	-1828	30.74	-189	18	222
776	-16548	-1804	30.60	-448	45	220
912	-16624	-1688	30.60	-182	-2	192
888	-16544	-1428	30.51	-235	51	172
672	-16560	-1396	30.55	49	-125	138
908	-16508	-1556	30.60	-301	39	178

At the bottom of the window, there are controls: Autoscroll, Show timestamp, a dropdown menu set to 'Newline', a dropdown menu set to '115200 baud', and a 'Clear output' button.

Sumbu z

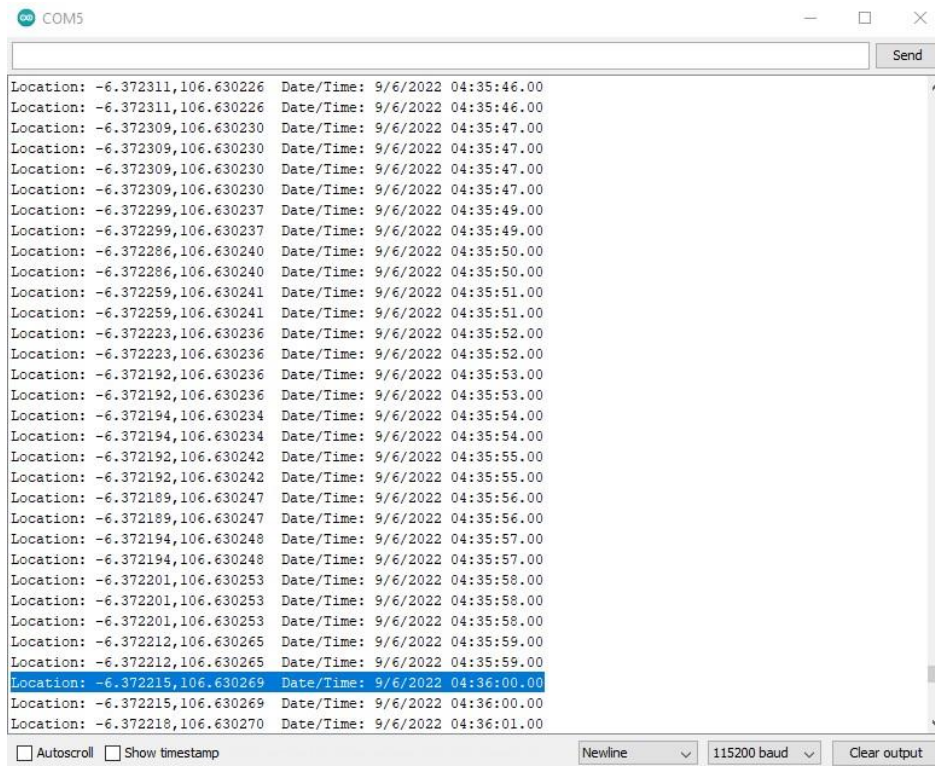
```
COM5
Send
aX = -292 | aY = -344 | aZ = 13220 | tmp = 30.32 | gX = -353 | gY = 30 | gZ = 208
aX = -380 | aY = -328 | aZ = 12972 | tmp = 30.32 | gX = -344 | gY = 28 | gZ = 182
aX = -460 | aY = -296 | aZ = 13152 | tmp = 30.32 | gX = -348 | gY = 72 | gZ = 189
aX = -324 | aY = -264 | aZ = 13084 | tmp = 30.27 | gX = -340 | gY = 20 | gZ = 210
aX = -392 | aY = -256 | aZ = 13128 | tmp = 30.37 | gX = -346 | gY = 60 | gZ = 219
aX = -284 | aY = -324 | aZ = 13096 | tmp = 30.41 | gX = -365 | gY = 35 | gZ = 211
aX = -308 | aY = -364 | aZ = 13116 | tmp = 30.32 | gX = -475 | gY = -13 | gZ = 205
aX = -248 | aY = -348 | aZ = 13164 | tmp = 30.37 | gX = -377 | gY = 35 | gZ = 177
aX = -236 | aY = -344 | aZ = 13168 | tmp = 30.32 | gX = -329 | gY = 64 | gZ = 210
aX = -304 | aY = -296 | aZ = 13148 | tmp = 30.37 | gX = -379 | gY = 55 | gZ = 195
aX = -284 | aY = -352 | aZ = 13132 | tmp = 30.32 | gX = -346 | gY = 26 | gZ = 204
aX = -324 | aY = -200 | aZ = 13120 | tmp = 30.27 | gX = -317 | gY = 40 | gZ = 219
aX = -276 | aY = -240 | aZ = 13124 | tmp = 30.22 | gX = -353 | gY = 42 | gZ = 222
aX = -240 | aY = -272 | aZ = 13088 | tmp = 30.22 | gX = -315 | gY = 44 | gZ = 216
aX = -276 | aY = -380 | aZ = 13072 | tmp = 30.32 | gX = -374 | gY = 51 | gZ = 210
aX = -284 | aY = -376 | aZ = 13108 | tmp = 30.32 | gX = -349 | gY = 72 | gZ = 193
aX = -252 | aY = -388 | aZ = 13208 | tmp = 30.27 | gX = -341 | gY = 21 | gZ = 228
aX = -196 | aY = -380 | aZ = 13100 | tmp = 30.18 | gX = -341 | gY = 52 | gZ = 237
aX = -212 | aY = -388 | aZ = 13176 | tmp = 30.22 | gX = -327 | gY = 19 | gZ = 208
aX = -264 | aY = -400 | aZ = 13100 | tmp = 30.18 | gX = -350 | gY = 42 | gZ = 214
aX = -348 | aY = -380 | aZ = 13048 | tmp = 30.27 | gX = -336 | gY = 29 | gZ = 202
aX = -188 | aY = -348 | aZ = 13156 | tmp = 30.08 | gX = -323 | gY = 48 | gZ = 212
aX = -276 | aY = -344 | aZ = 13224 | tmp = 30.22 | gX = -358 | gY = 43 | gZ = 209
aX = -216 | aY = -400 | aZ = 13184 | tmp = 30.22 | gX = -330 | gY = 32 | gZ = 229
aX = -252 | aY = -392 | aZ = 13096 | tmp = 30.22 | gX = -353 | gY = 50 | gZ = 194
 Autoscroll  Show timestamp
Newline 115200 baud Clear output
```

Sumbu -z

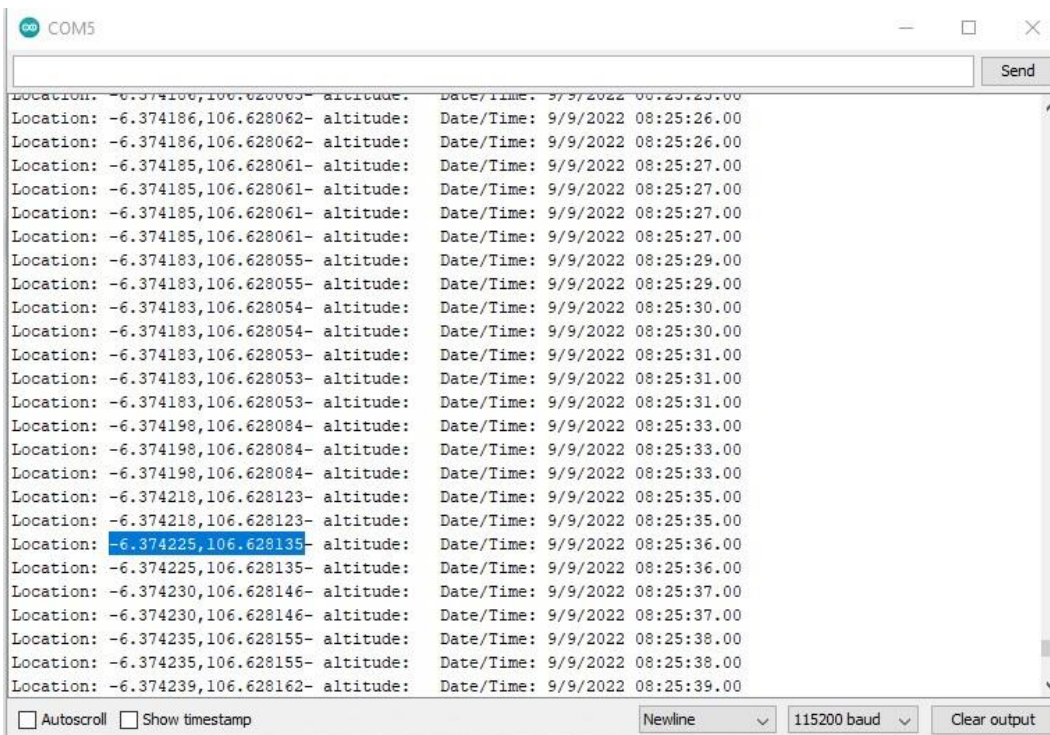
```
COM5
Send
aX = 840 | aY = -756 | aZ = -19764 | tmp = 30.51 | gX = -309 | gY = 12 | gZ = 263
aX = 840 | aY = -628 | aZ = -19908 | tmp = 30.51 | gX = -237 | gY = 311 | gZ = 263
aX = 832 | aY = -844 | aZ = -19788 | tmp = 30.51 | gX = -267 | gY = -123 | gZ = 377
aX = 772 | aY = -664 | aZ = -19968 | tmp = 30.46 | gX = -490 | gY = -335 | gZ = 302
aX = 808 | aY = -656 | aZ = -20040 | tmp = 30.46 | gX = -755 | gY = -467 | gZ = -205
aX = 700 | aY = -452 | aZ = -19944 | tmp = 30.51 | gX = -271 | gY = -67 | gZ = 287
aX = 784 | aY = -532 | aZ = -19836 | tmp = 30.51 | gX = -380 | gY = -41 | gZ = 246
aX = 732 | aY = -536 | aZ = -20048 | tmp = 30.41 | gX = -322 | gY = 274 | gZ = 213
aX = 836 | aY = -492 | aZ = -20072 | tmp = 30.41 | gX = -285 | gY = 427 | gZ = 268
aX = 472 | aY = -652 | aZ = -20040 | tmp = 30.46 | gX = -442 | gY = -122 | gZ = 33
aX = 868 | aY = -328 | aZ = -19432 | tmp = 30.51 | gX = -853 | gY = -2129 | gZ = -63
aX = -452 | aY = 128 | aZ = -20024 | tmp = 30.46 | gX = -442 | gY = -110 | gZ = 307
aX = -688 | aY = 156 | aZ = -19936 | tmp = 30.41 | gX = -357 | gY = -59 | gZ = 136
aX = -680 | aY = 144 | aZ = -19776 | tmp = 30.41 | gX = -245 | gY = 450 | gZ = 92
aX = -556 | aY = 24 | aZ = -19836 | tmp = 30.41 | gX = -415 | gY = -199 | gZ = 250
aX = -516 | aY = -108 | aZ = -19828 | tmp = 30.46 | gX = -480 | gY = -50 | gZ = 253
aX = -320 | aY = -252 | aZ = -19928 | tmp = 30.37 | gX = -531 | gY = -231 | gZ = 151
aX = -244 | aY = -148 | aZ = -19940 | tmp = 30.41 | gX = -509 | gY = -62 | gZ = -143
aX = -256 | aY = -220 | aZ = -20004 | tmp = 30.41 | gX = -437 | gY = 69 | gZ = 242
aX = -68 | aY = -192 | aZ = -19756 | tmp = 30.51 | gX = -853 | gY = -394 | gZ = 540
aX = -252 | aY = -364 | aZ = -20200 | tmp = 30.37 | gX = -399 | gY = 45 | gZ = 224
aX = -284 | aY = -252 | aZ = -19832 | tmp = 30.46 | gX = -138 | gY = 352 | gZ = 529
aX = -16 | aY = -436 | aZ = -20020 | tmp = 30.41 | gX = -440 | gY = -106 | gZ = 357
aX = 16 | aY = -548 | aZ = -19968 | tmp = 30.41 | gX = -329 | gY = 65 | gZ = 207
aX = 124 | aY = -376 | aZ = -20012 | tmp = 30.32 | gX = -459 | gY = -158 | gZ = 302
 Autoscroll  Show timestamp
Newline 115200 baud Clear output
```


- **Lampiran 8. Screenshot Data Hasil Pembacaan Modul GPS di software Arduino IDE**

GPS Pengambilan Data 1



GPS Pengambilan Data 2



- **Lampiran 9.** Dokumentasi Kegiatan Praktik Kerja Lapangan

Studi Literatur di Perpustakaan Pustekbang-BRIN



Proses Pengambilan Data



Sesi Diskusi Ilmiah Bersama Kepala Pustekbang-BRIN



Sesi Foto Bersama Semua Peserta PKL Pustekbang-BRIN

