

**RANCANG BANGUN SISTEM POSISI DAN *GROUND*
STATION MONITORING PADA SOLAR CELL UAV
(*UNMANNED AERIAL VEHICLE*)**

TUGAS AKHIR



**DWI HERIYANTO
SERKA NAV NRP 110115**

**PROGRAM STUDI D-3 TEKNIK ELEKTRONIKA
DIREKTORAT PEMBINAAN DIPLOMA
SEKOLAH TINGGI TEKNOLOGI ANGKATAN LAUT**

**SURABAYA
DESEMBER 2020**

**RANCANG BANGUN SISTEM POSISI DAN *GROUND*
STATION MONITORING PADA *SOLAR CELL* UAV
(*UNMANNED AERIAL VEHICLE*)**

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Ahli Madya
Program Studi D-3 Teknik Elektronika
STTAL**

**DWI HERIYANTO
SERKA NAV NRP 110115**

**PROGRAM STUDI D-3 TEKNIK ELEKTRONIKA
DIREKTORAT PEMBINAAN DIPLOMA
SEKOLAH TINGGI TEKNOLOGI ANGKATAN LAUT**

**SURABAYA
DESEMBER 2020**

LEMBAR PERSETUJUAN DOSEN PEMBIMBING

Judul Tugas Akhir : Rancang Bangun Sistem Posisi dan *Ground Station Monitoring* pada *Solar Cell UAV*
(*Unmanned Aerial Vehicle*)
Nama Mahasiswa : Dwi Heriyanto
Pangkat/Korps/NRP : Serka Nav NRP 110115
Program Studi : Diploma 3 Teknik Elektronika

Tugas Akhir dengan judul dan atas nama mahasiswa tersebut telah disetujui untuk dapat diujikan, sebagai bagian dari persyaratan tugas akhir untuk memperoleh gelar Ahli Madya Teknik pada Program Studi Teknik Elektronika, Direktorat Pembinaan Diploma Sekolah Tinggi Teknologi Angkatan Laut.

DOSEN PEMBIMBING

Pembimbing I : Dr. Rachmad S., S.T., M.T. (.....)
Lektor III/d
NIP 196905291995121001
Pembimbing II : Endin Tri Hartanto, S.T., M.T. (.....)
Mayor Laut (E) NRP 15625/P
Ditetapkan di : Surabaya
Tanggal : Desember 2020

Mengetahui
Direktur Pembinaan Diploma

Dr. Ahmadi, S.Si., M.T.
Kolonel Laut (KH) NRP 10053/P

LEMBAR PENGESAHAN UJIAN TUGAS AKHIR

Judul Tugas Akhir : Rancang Bangun Sistem Posisi dan *Ground Station Monitoring* pada *Solar Cell UAV*
(*Unmanned Aerial Vehicle*)

Nama Mahasiswa : Dwi Heriyanto

Pangkat/Korps/NRP : Serka Nav NRP 110115

Program Studi : Diploma 3 Teknik Elektronika

Telah/Tidak*) berhasil dipertahankan dihadapan Dewan Penguji dan Belum/Sudah*) diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Ahli Madya dalam Program Studi Teknik Elektronika, Direktorat Pembinaan Diploma Sekolah Tinggi Teknologi Angkatan Laut.

DEWAN PENGUJI

Ketua : Dr. Rachmad S., S.T., M.T. (.....)
Lektor III/d NIP 196905291995121001

Anggota : Bagiyo, S.T., M.T. (.....)
Ass Ahli III/A 9818087

Anggota : Sunarta, S.T., M.T. (.....)
Letkol Laut (E) NRP 12898/P

Anggota : Erpan Sahiri, S.T., M.T. M.Tr (Hanla) (.....)
Letkol Laut (E) NRP 13481/P

Anggota : Bagus Irawan, S.T., M.MT. (.....)
Mayor Laut (E) NRP 15618/P

Ditetapkan di : Surabaya

Tanggal : Desember 2020

Mengetahui
Direktur Pembinaan Diploma

Dr. Ahmadi, S.Si., M.T.
Kolonel Laut (KH) NRP 10053/P

PERNYATAAN ORISINALITAS

Yang bertanda tangan dibawah ini :

Judul Tugas Akhir : Rancang Bangun Sistem Posisi dan *Ground Station Monitoring* pada *Solar Cell UAV*
(*Unmanned Aerial Vehicle*)

Nama Mahasiswa : Dwi Heriyanto

Pangkat/Korps/NRP : Serka Nav NRP 110115

Program Studi : Diploma 3 Teknik Elektronika

Dengan ini saya menyatakan bahwa dalam Tugas Akhir ini tidak terdapat karya atau bagian karya yang pernah diajukan untuk memperoleh gelar Ahli Madya jenjang apapun disuatu Perguruan Tinggi, dan sepanjang pengetahuan saya juga tidak terdapat istilah, frasa, kalimat, paragraf, sub bab, atau bab dari karya yang pernah ditulis atau diterbitkan, kecuali yang secara tertulis dirujuk dalam naskah ini dan disebutkan dalam daftar referensi.

Apabila di kemudian hari terbukti bahwa terdapat plagiat dalam Tugas Akhir ini, saya bersedia menerima sanksi sesuai ketentuan peraturan dan undang-undang yang berlaku.

Surabaya, Desember 2020

Penulis

SURAT PENGALIHAN HAK CIPTA

Yang bertanda tangan dibawah ini :

Nama : Dwi Heriyanto

Alamat : Lingkungan Booto RT 001/RW 001 Kota Mojokerto Jawa Timur

Adalah Pihak I selaku pencipta, dengan ini menyerahkan karya ciptaan saya kepada :

Nama : Sekolah Tinggi Teknologi Angkatan Laut

Alamat : Jl. Bumimoro Moro Krembangan Surabaya

Adalah Pihak II selaku Pemegang Hak Cipta berupa Tugas Akhir dengan Judul “Rancang Bangun Sistem Posisi dan *Ground Station Monitoring* pada *Solar Cell UAV (Unmanned Aerial Vehicle)*” untuk didaftarkan di Direktorat Hak Cipta, Desain Industri, Desain Rancang Bangun Sistem Posisi dan *Ground Station Monitoring* pada *Solar Cell UAV (Unmanned Aerial Vehicle)* Tata Letak dan Sirkuit Terpadu dan Rahasia Dagang, Direktorat Jenderal Hak Kekayaan Intelektual, Kementerian Hukum dan Hak Azasi Manusia R.I. Demikianlah surat pengalihan hak ini kami buat, agar dapat dipergunakan sebagaimana mestinya.

Pemegang Hak Cipta

Surabaya, Desember 2020
Pencipta

Dr. Ahmadi, S.Si., M.T.
Kolonel Laut (KH) NRP 10053/P

Dwi Heriyanto
Serka Nav NRP 110115

© Hak Cipta Milik STTAL
Dilindungi Undang-Undang

“Dilarang mengutip sebagian atau seluruh tesis/skripsi/tugas akhir ini tanpa mencantumkan atau menyebutkan sumbernya. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik, atau tinjauan suatu masalah; dan pengutipan tersebut tidak merugikan kepentingan STTAL”

Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apa pun tanpa izin STTAL

KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Dengan memanjatkan puji syukur kehadirat Tuhan Yang Maha Kuasa, yang telah mengkaruniakan kekuatan dan kesehatan kepada penulis sehingga dapat menyelesaikan Tugas Akhir ini tepat pada waktunya dengan judul “Rancang Bangun Sistem Posisi dan *Ground Station Monitoring* pada *Solar Cell UAV (Unmanned Aerial Vehicle)*”

Tugas Akhir ini merupakan salah satu persyaratan untuk menyelesaikan pendidikan bagi Bintara Siswa Prodi Diploma 3 Teknik Elektronika Angkatan XIII TA 2020, menyadari dengan sepenuhnya bahwa Tugas Akhir ini masih jauh dari kesempurnaan baik mengenai isi, pembahasan maupun penyusunannya, hal ini tentu disebabkan oleh keterbatasan pada diri penulis, oleh sebab itu penulis sangat mengharapkan adanya koreksi dan saran yang konstruktif dari berbagai pihak guna penyempurnaan.

Penulis menyadari bahwa penulisan Tugas Akhir ini tidak dapat terselesaikan dengan baik dan tepat waktu, tanpa adanya bantuan dari pihak lain, oleh sebab itu penulis mengucapkan terima kasih kepada :

1. Komandan STTAL serta seluruh Staf STTAL atas kesempatan yang telah diberikan kepada penulis untuk dapat mengikuti pendidikan Diploma III di Prodi Teknik Elektronika STTAL.
2. Direktur Pembinaan Diploma, Kaprodi Diploma 3 Teknik Elektronika, Dosen dan seluruh Staf STTAL yang telah membaktikan segenap tenaga dan pemikirannya dalam mendukung proses pembelajaran.
3. Bapak Dr. Rachmad S., S.T., M.T. dan Mayor Laut (E) Endin Tri Hartanto, S.T., M.T. selaku Dosen Pembimbing yang telah memberikan waktu, arahan, bimbingan dan bantuan selama pembuatan Tugas Akhir ini.

4. Seluruh rekan-rekan Diploma 3 Teknik Elektronika XIII STTAL atas kebersamaan dalam suka maupun duka, kerjasama, dukungan dan bantuannya selama ini.

5. Istri tercinta Nani Masita Inayati dan putra kami terkasih Loya Zilgar Maheriyanto dan Sultan Zulkhair Maheriyanto yang senantiasa mendampingi, menemani, mendukung serta memberikan semangat kepada kami selama penyusunan Tugas Akhir ini.

6. Semua pihak yang tidak mungkin penulis sebutkan satu persatu yang telah membantu dalam penyelesaian Tugas Akhir ini.

Penulis menyadari bahwa Tugas Akhir ini masih belum sempurna dan memerlukan perbaikan, oleh karena itu dengan mohon adanya kritikan, saran, dan masukan serta koreksi dari semua pihak guna penyempurnaan isi proposal ini. Akhirnya dengan segala kerendahan hati penulis berharap semoga Tugas Akhir ini dapat bermanfaat bagi pembaca dan menjadi sumbangan pemikiran yang berguna bagi TNI Angkatan Laut di masa yang akan datang.

Wassalamu'alaikum Wr. Wb.

Surabaya, Desember 2020

Dwi Heriyanto
Serka Nav NRP 110115

ABSTRAK

UAV merupakan teknologi kendaraan terbang tanpa awak yang dikembangkan dan digunakan untuk mempermudah pekerjaan. Kebutuhan sumber daya elektrik UAV yang besar maka digunakan solar sel sebagai daya pendukung. Penggunaan UAV tidak terlepas dari sistem pemetaan untuk pemantau keberadaan UAV. Pemetaan uav merupakan kegiatan untuk mendefinisikan posisi UAV berdasar lintang bujur bumi. Perangkat GPS digunakan untuk menunjukkan lokasi UAV dalam koordinat bumi lintang bujur. Untuk mendapatkan sikap UAV terhadap bumi digunakan perangkat *Gyroscope*. *Gyroscope* mengukur kecepatan sudut pada tiga sumbu. Dengan mengintegrasikan kecepatan sudut dari waktu ke waktu dapat diperoleh sudut yang dapat digunakan untuk mengukur perubahan *roll*, *pitch* dan *yaw*. Untuk menunjang sistem pemetaan digunakan mikrokontroler Arduino Mega 2560 sebagai sistem pengolahan data sensor. Untuk mengirimkan data ke ground station digunakan perangkat nirkabel NRF24L01. jarak jangkauan maksimal 200 meter sistem monitoring posisi pesawat UAV dapat bekerja dengan baik tanpa terhalang bangunan. Berdasarkan pengujian sistem *ground station* yang telah dibuat oleh penulis, diperoleh sistem yang dapat memonitor posisi dan sikap UAV saat melakukan misi penerbangan.

Kata Kunci: UAV, *solar sel*, GPS, *Gyroscope*, NRF24L01, *ground station*, *roll*, *pitch yaw*

ABSTRACT

UAV is an unmanned flying vehicle technology developed and used to simplify work. The need for a large UAV electrical resource is used solar cells as supporting power. The use of UAVs cannot be separated from the mapping system for monitoring the presence of UAVs. UAV mapping is an activity to define the position of the UAV based on the latitude and longitude of the earth. The GPS device is used to show the UAV location in latitude longitude earth coordinates. To get the UAV's attitude towards the earth a gyroscope is used. The gyroscope measures angular velocity on three axes. By integrating the angular velocity over time, an angle can be obtained which can be used to measure changes in roll, pitch and yaw. To support the mapping system, the Arduino Mega 2560 microcontroller is used as a sensor data processing system. To transmit data to the ground station, the NRF24L01 wireless device is used. The maximum range distance of 200 meters, the UAV aircraft position monitoring system can work properly without being obstructed by buildings. Based on the ground station system testing that has been made by the author, a system that can monitor the position and attitude of the UAV during flight missions is obtained.

Keywords: UAV, solar sel, GPS, Gyroscope, NRF24L01, ground station, roll, pitch yaw

DAFTAR ISI

HALAMAN JUDUL	ii
LEMBAR PERSETUJUAN DOSEN PEMBIMBING	iii
LEMBAR PENGESAHAN UJIAN TUGAS AKHIR	iv
PERNYATAAN ORISINALITAS	v
SURAT PENGALIHAN HAK CIPTA	vi
KATA PENGANTAR	viii
ABSTRAK	x
<i>ABSTRACT</i>	xi
DAFTAR ISI	xi
DAFTAR GAMBAR	xv
DAFTAR TABEL	xvii
DAFTAR LAMPIRAN	xviii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	3
1.3 Tujuan Penelitian	3
1.4 Manfaat Penelitian	3
1.5 Batasan Masalah	4
BAB 2 TINJAUAN PUSTAKA	4
2.1 Penelitian Terdahulu	5
2.2 Landasan Teori	6
2.2.1 <i>Unmanned Aerial Vehicle (UAV)</i>	6
2.2.2 Jenis UAV	6
2.2.3 Dasar Pergerakan UAV	7

2.2.4	<i>Global Positioning System (GPS)</i>	7
2.2.5	Mikrokontroler Arduino	11
2.2.6	Sensor <i>Gyroscope</i> dan <i>Accelerometer</i>	12
BAB 3 METODE PENELITIAN		14
3.1.	Desain Penelitian	15
3.2	Prosedur Penelitian.....	15
3.2.1	Waktu dan Tempat Penelitian.....	15
3.2.2	Alat dan Bahan Penelitian	15
3.2.3	Rancangan Penelitian	16
3.2.4	Pengumpulan Data.....	26
3.2.5	Pengolahan Data.....	26
BAB 4 IMPLEMENTASI DAN PENGUJIAN SISTEM		27
4.1	Implementasi	28
4.1.1	Implementasi Perangkat Komunikasi.....	28
4.1.2	Implementasi Perangkat Proses	29
4.2	Pengujian.....	30
4.2.1	Pengujian Perangkat Keras.....	30
4.2.1.1	Pengujian Modul GPS	30
4.2.1.2	Pengujian Kompas MPU9250	32
4.2.1.4	Pengujian Sensor Barometer BMP280.....	35
4.2.1.5	Pengujian Perangkat Wireless	37
4.2.1.6	Pengujian Sistem <i>Power Management</i> pada GCS.....	39
4.2.1.7	Pengujian Ground Station Monitoring.....	40
BAB 5 KESIMPULAN DAN SARAN		50
5.1	Kesimpulan.....	50
5.2	Saran	50

DAFTAR PUSTAKA.....	51
---------------------	----

DAFTAR GAMBAR

Gambar 2.1	Pergerakan UAV	7
Gambar 2.2	Satelit GPS	9
Gambar 2.3	Blok diagram Arduino Mega 2560	12
Gambar 2.4	Prinsip kerja <i>gyroscope</i>	12
Gambar 3.1	<i>Flowchart</i> sistem posisi di UAV <i>Solar Cell</i>	19
Gambar 3.2	<i>Flowchart</i> sistem pada <i>ground station</i>	20
Gambar 3.3	Blok diagram sistem posisi dan <i>ground station</i>	21
Gambar 3.4	GPS UBLOK NEO 6M	23
Gambar 3.5	Sensor MPU 6050	23
Gambar 3.6	Modul radio NRF24L01	24
Gambar 4.1	Fitur-fitur <i>Ground Station Monitoring</i>	27
Gambar 4.2	Diagram blok komunikasi data UAV Dan GCS.....	28
Gambar 4.3	<i>Wiring</i> diagram pengujian GPS.....	30
Gambar 4.4	Pengujian GPS Ublox Neo-8M.....	31
Gambar 4.5	Diagram blok kompas.....	32
Gambar 4.6	Pin-pin kompas.....	32
Gambar 4.7	<i>Wiring</i> diagram pengujian kompas.....	33
Gambar 4.8	Pengujian kompas MPU9250.....	33
Gambar 4.9	Tampilan Arduino perhitungan ketinggian (<i>altitude</i>).....	35
Gambar 4.10	Hasil pengujian BMP280 pada terminal monitor.....	35
Gambar 4.11	Gambar <i>wiring digaram</i> modul nirkabel NRF24L01.....	36
Gambar 4.12	Pengujian penerimaan data dari Master dan pengiriman kembali ke Master.....	37
Gambar 4.13	Diagram Blok Sistem <i>Power Management</i>	38
Gambar 4.14	Instrumen Ukur <i>Solar Cell</i>	39
Gambar 4.15	Instrumen Ukur Baterai.....	40
Gambar 4.16	Tampilan pengaturan komunikasi serial pada GCS.....	40
Gambar 4.17	Tampilan koneksi USB berhasil.....	41
Gambar 4.18	Tampilan data penerimaan GCS.....	41

Gambar 4.19 Tampilan data orientasi UAV.....	43
Gambar 4.20 Tampilan kegagalan data GPS.....	43
Gambar 4.21 Tampilan marker UAV pada sudut bumi 0 derajat.....	44
Gambar 4.22 Tampilan marker UAV pada sudut bumi 90 derajat.....	45
Gambar 4.23 Tampilan marker UAV pada sudut bumi 180 derajat.....	45
Gambar 4.24 Tampilan marker UAV pada koordinat -7.446048, 112.717737.....	46
Gambar 4.25 <i>Gound Station Monitoring</i>	46
Gambar 4.26 Instrumen <i>avionik</i> pesawat UAV.....	47
Gambar 4.27 Pesawat UAV <i>Take Off</i>	47
Gambar 4.28 Lokasi terbang pesawat UAV.....	48
Gambar 4.29 Tampilan <i>Ground Station Monitoring</i>	48

DAFTAR TABEL

Tabel 3.1	Alat dan bahan.....	15
Tabel 4.1	Pengujian Modul NRF24L01.....	38

DAFTAR LAMPIRAN

LAMPIRAN - 1 <i>Listing</i> program <i>Ground Station Monitoring</i>	51
LAMPIRAN - 2 <i>Listing</i> Program <i>Arduino Power Management</i>	58

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi yang semakin pesat di persenjataan di TNI Angkatan Laut, membuat banyak kegiatan penelitian yang ditujukan agar dapat memudahkan berbagai pekerjaan manusia. Salah satunya adalah penelitian *Unmanned Aerial Vehicle* (UAV) yang ditujukan untuk memberikan kemudahan bagi personil TNI AL dalam melakukan beberapa tugas melalui jalur udara. Dimana UAV merupakan teknologi kendaraan terbang tanpa awak yang dikembangkan dan digunakan untuk mempermudah pekerjaan. Dewasa ini, teknologi UAV pada umumnya sudah diterapkan dalam bidang fotografi melalui jalur udara pemetaan. Dan hingga saat ini, penelitian terkait dengan UAV masih terus dilakukan untuk bidang lain yang membantu memudahkan pekerjaan manusia di jalur udara. Sebagai contoh bidang yang saat ini masih dikembangkan adalah bidang pada sistem pemetaan, bidang logistik, pertahanan dan keamanan, dan beberapa bidang lainnya.

Dengan kebutuhan daya yang banyak, tentunya akan semakin baik apabila dibutuhkan sumber daya tambahan pada UAV. Maka dari itu, dipilihlah *Solar Cell*, sebagai sumber daya tambahan tersebut. Dimana pemilihan *Solar Cell* sebagai sumber daya tambahan dikarenakan faktor cuaca dan iklim dari wilayah negara Indonesia, yang mendukung kinerja *Solar Cell* dalam penambahan daya.

Dengan kondisi cuaca yang panas disertai iklim tropis, *Solar Cell* mampu menjawab ketersediaan dalam penambahan daya tersebut. Selain itu, karakteristik pada *Solar Cell* yang tipis dengan dimensi luas tertentu, membuat adanya pertimbangan khusus akan pemilihan jenis UAV itu sendiri. Maka dari itu, dipilihlah UAV jenis *fixed wing*. Dimana *Solar Cell* bisa digunakan dengan jumlah yang banyak dan mampu direkatkan pada sayap. Dengan pemilihan pada *Solar Cell* sebagai sumber daya tambahan dan *fixed wing* sebagai jenis UAV yang baik untuk penambahan *Solar Cell*

tersebut, maka diharapkan akan mampu menambah *flight time* saat UAV melakukan penerbangan. Secara spesifik *Solar Cell* berfungsi untuk melakukan pengisian baterai dan sebagai sumber daya utama saat terbang. Pengisian akan dilakukan jika sumber daya *Solar Cell* telah memenuhi konsumsi daya sistem propulsi dan konsumsi sistem propulsi tidak melebihi kapasitas daya maksimal dari *Solar Cell*. Saat UAV melakukan *take off* maka sumber daya akan di serap dari sumber baterai.

Penggunaan UAV tidak terlepas dari sistem monitoring untuk memantau keberadaan UAV. Posisi UAV menjadi sangat penting berkaitan dengan tujuan UAV pada area tertentu. *Monitoring* UAV merupakan kegiatan untuk mendefinisikan posisi UAV berdasar lintang bujur bumi. Koordinat yang digunakan 2 dimensi mewakili posisi UAV berdasar sumbu *x* dan sumbu *y*. Saat UAV melakukan misi penerbangan UAV akan mengalami perpindahan posisi, dimana posisi lintang bujur secara *linier* akan ikut berubah.

Untuk mendapatkan informasi posisi UAV, digunakanlah perangkat GPS. Perangkat GPS digunakan untuk menunjukkan lokasi UAV dalam koordinat bumi lintang bujur. Sistem GPS terdiri dari tiga bagian, yakni satelit di angkasa, stasiun pengendali di bumi, dan *receiver* sebagai perangkat penerima sinyal satelit GPS yang terpasang pada UAV. Ketika satelit mengorbit, memancarkan sinyal ditangkap oleh penerima GPS di bumi. Alat penerima akan menghitung jarak dengan satelit dengan mengukur interval waktu yang dibutuhkan untuk menerima sinyal dari masing-masing satelit.

Informasi jarak dipakai untuk menentukan posisi penerima GPS menggunakan teknik *trilateration*, yaitu sebuah teknik untuk mencari titik persilangan di antara radius jarak tiga satelit GPS terdekat. Ketika ketiga lingkaran jarak ini bertemu, maka itulah posisi penerima GPS. Untuk mendapatkan lokasi dan *tracking* pergerakan garis bujur dan lintang, penerima GPS memerlukan sinyal tiga satelit. Untuk mendapatkan sikap UAV terhadap bumi digunakan perangkat *Gyroscope*. *Gyroscope* mengukur kecepatan sudut pada tiga sumbu. Dengan mengintegrasikan

kecepatan sudut dari waktu ke waktu dapat diperoleh sudut yang dapat digunakan untuk mengukur perubahan *roll*, *pitch* dan *yaw*. Dalam mewujudkan sistem penentuan posisi dan sikap UAV, penulis mengangkatnya menjadi sebuah tema penelitian dengan judul penelitian Rancang Bangun Sistem Posisi dan *Ground Station Monitoring* pada *Solar Cell UAV (Unmanned Aerial Vehicle)*. Sebuah penelitian dalam rangka memberikan informasi posisi dan sikap UAV terhadap bumi dan dapat dimonitor secara nirkabel pada *ground station*.

1.2 Perumusan Masalah

Pada penelitian ini perlu dirumuskan permasalahan-permasalahan dalam Rancang Bangun Sistem Posisi dan *Ground Station Monitoring* pada *Solar Cell UAV (Unmanned Aerial Vehicle)*.

Adapun beberapa permasalahan tersebut adalah :

- a. Bagaimana rancangan UAV *Solar Cell* untuk mendapatkan posisi dalam lintang dan bujur menggunakan GPS yang di monitoring di *Ground Station*.
- b. Bagaimana rancangan UAV *Sollar Cell* untuk mendapatkan sikap dalam *yaw*, *pitch* dan *roll* menggunakan *Gyroscope* yang di monitoring di *Ground Station*.

1.3 Tujuan Penelitian

Adapun tujuan dari Rancang Bangun Sistem Posisi dan *Ground Station Monitoring* pada *Solar Cell UAV (Unmanned Aerial Vehicle)* adalah:

- a. Merancang *system monitoring* UAV *Solar Cell*
- b. Mengukur dan *monitoring* posisi, sikap, kecepatan UAV *Solar Cell* yang di *monitoring* di *Ground Station*.

1.4 Manfaat Penelitian

Adapun manfaat Rancang Bangun Sistem Posisi dan *Ground Station Monitoring* pada *Solar Cell UAV (Unmanned Aerial Vehicle)* adalah:

a. Bagi TNI AL

Sebagai salah satu cara komunikasi data via radio komunikasi dalam kegiatan pengambilan data yang efektif dan efisien, agar dapat memantau keadaan di daerah latihan.

b. Bagi STTAL

Sebagai penerapan sistem komunikasi yang berkaitan dengan materi pendidikan pada system telekomunikasi, sebagai penerapan materi pemrograman Arduino mega 2560 dan diharapkan rencang bangun ini menjadi produk unggulan STTAL pada Pameran industri pertahanan dalam negeri.

c. Bagi diri sendiri

Memberikan ilmu pengetahuan tentang komunikasi data untuk kepentingan tugas kedepan.

1.5 Batasan Masalah

Adapun batasan Rancang Bangun Sistem Posisi dan *Ground Station Monitoring* pada *Solar Cell UAV (Unmanned Aerial Vehicle)* adalah:

- a. Jarak jangkauan kendali (*range*) komunikasi tidak lebih dari 500 meter
- b. Menggunakan Arduino Mega 2560
- c. Menggunakan Telemetry NRF24L01

BAB 2 TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Penelitian dalam bidang ini sudah dilakukan oleh beberapa peneliti terkait dengan permasalahan desain UAV *Fixed Wing* maupun penggunaan *Solar cell* sebagai sumber daya tambahan pada *Fixed Wing*. Seperti penelitian yang dilakukan oleh Scott Morton, Ruben D'Sa, dan Nikolaos Papanikolopoulos (2015) yang berjudul *Solar Powered UAV Design and Experiments*. Dimana dalam penelitian yang telah dikerjakan membahas tentang desain pada sebuah bentuk UAV berukuran kecil dengan pemasangan *Solar Cell* pada sayap yang memiliki ukuran rentang sayap sebesar 4 meter serta beban total UAV sebesar 3,1 Kg. Dan daya yang dihasilkan maksimum sebesar 180 Watt. Adapun dari daya yang dihasilkan menjadi kelebihan pada penelitian ini. Namun kekurangannya adalah tidak ada pembahasan lebih lanjut mengenai *Flight Time*. Dikarenakan pada penelitian ini lebih terfokus kepada daya yang dihasilkan oleh UAV saat dipasang *Solar Cell* pada sayap. Serta adanya faktor lain yang memengaruhi seperti efisiensi *aerodinamis*, dan ketahanan pesawat saat terbang sehingga perhitungan *Flight Time* pada penelitian ini dibatasi.

Kemudian penelitian yang dilakukan oleh Md. Shaiful Islam, Mohammad Mukit Hasan, Mir Md. Al Kamah Tamal, Md. Jamal Mian, dan Md. Touhidur Rahman Evan (2010) yang berjudul *Detail Solidworks Design and Simulation of an Unmanned Air Vehicle*. Pada penelitian ini diperlihatkan bagaimana sebuah UAV jenis *Fixed Wing* didesain menggunakan *software Solidworks*. Desain UAV ini kemudian disimulasikan dengan melakukan uji tekanan pada sayap. Pada penelitian ini, dibahas mengenai bagaimana sebuah desain pada UAV dikerjakan dan disimulasikan dengan teratur. Namun desain yang dikerjakan tidak membahas lebih lanjut bagaimana spesifikasi dari kebutuhan motor penggerak dan sumber daya yang akan digunakan. Dan penelitian ini dapat dijadikan referensi saat melakukan proses desain pada UAV melalui *software Solidworks*.

Selanjutnya penelitian yang dilakukan Ariyono Setiawan, Sudrajat, Ridho Rinaldi, Didi Hariyanto (2000) yang berjudul *Perancangan Sistem Kontrol Pesawat Model Tanpa Awak UAV Berbasis Arduino* saran dari

penulis sampaikan berkaitan dengan Penelitian ke depannya perlu ditambahkan pemasangan sistem kontrol pada pesawat UAV pesawat model elektrik.

2.2 Landasan Teori

Memberikan informasi mengenai hubungan dari masalah yang melatarbelakangi penelitian ini dengan sumber-sumber kepustakaan sebagai dasar penelitian, mencakup jurnal maupun artikel yang relevan.

2.2.1 *Unmanned Aerial Vehicle (UAV)*

Unmanned Aerial Vehicle (UAV) adalah pesawat yang terbang tanpa operator di dalamnya dan dapat terbang secara *autonomous* dengan mengolah data dari *ground station*. UAV dapat dikendalikan secara manual melalui radio kontrol atau secara otomatis dengan mengolah data sensor sehingga dapat terbang sesuai dengan keperluan pengguna.

2.2.2 Jenis UAV

Secara umum pembagian jenis UAV dilakukan menurut posisi sayap dan sumber tenaga pesawat.

- a. Jenis UAV berdasarkan posisi sayap.

Posisi sayap UAV terbagi dalam 3 bagian, meliputi:

- 1) *High wing*. Sayap pesawat berada di atas *fuselage*.
- 2) *Middle wing*. Sayap pesawat berada di tengah *fuselage* dalam posisi sejajar.
- 3) *Low wing*. Sayap pesawat berada di bawah *fuselage*.

- b. Jenis pesawat berdasarkan sumber tenaga penggerak propeller:

- 1) *Combustion engine*. Merupakan pesawat menggunakan bahan bakar cair pada *engine*.
- 2) Elektrik. Sumber tenaga pesawat menggunakan suplai daya dari baterai.

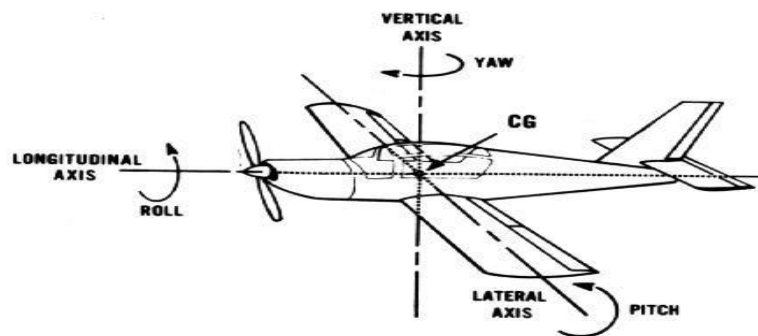
- c. Jenis UAV berdasarkan berat meliputi:

- 1) *Super heavy*. Berat diatas 2000 Kg.
- 2) *Heavy*. Berat antara 200 - 2000 Kg.
- 3) *Medium*. Berat antara 50 - 200 Kg.
- 4) *Light*. Berat antara 5 - 50 Kg.
- 5) *Micro*. Berat kurang dari 5 Kg

2.2.3 Dasar Pergerakan UAV

UAV memiliki 6 derajat kebebasan pergerakan di udara. Meliputi sumbu longitudinal, vertikal dan lateral. Titik pertemuan sumbu-sumbu tersebut merupakan *Center of Gravity* (CG) dalam kestabilan dan manuver pesawat. Manuver yang dilakukan antara lain:

- a. *Rolling*, gerak pesawat terhadap sumbu longitudinal dengan menggunakan aileron.
- b. *Pitching*, gerak pesawat terhadap sumbu lateral dengan menggunakan elevator.
- c. *Yawing*, gerak pesawat terhadap sumbu vertikal dengan menggunakan *rudder*.



Gambar 2.1 Pergerakan UAV

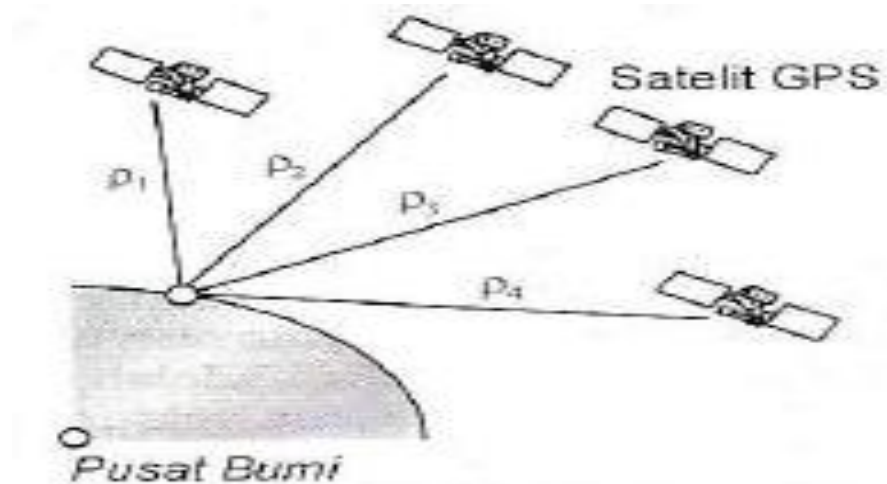
2.2.4 Global Positioning System (GPS)

Global Positioning System (GPS) adalah sistem radio navigasi yang berbasiskan satelit yang saling berhubungan yang berada di orbitnya. Satelit-satelit itu milik *Department of Defense* (Departemen Pertahanan) Amerika Serikat yang pertama kali diperkenalkan mulai tahun 1978 dan pada tahun 1994 sudah memakai 24 satelit. Untuk dapat mengetahui posisi seseorang maka diperlukan alat yang bernama *GPS Receiver* yang

berfungsi untuk menerima sinyal yang dikirim dari satelit GPS. Posisi diubah menjadi titik yang dikenal dengan nama *Way-point* nantinya akan berupa titik-titik koordinat lintang dan bujur dari posisi seseorang atau suatu lokasi kemudian di tampilkan di layar pada peta elektronik.

GPS merupakan sistem untuk menentukan letak di permukaan bumi dengan bantuan penyelarasan (*synchronization*) sinyal satelit. Sistem ini menggunakan 24 satelit yang mengirimkan sinyal gelombang mikro ke bumi. Sinyal ini diterima oleh alat penerima di permukaan, dan digunakan untuk menentukan letak, kecepatan, arah, dan waktu. Sistem yang serupa dengan GPS antara lain *GLONASS Rusia, Galileo Uni Eropa, IRNSS India* Sistem GPS, yang nama aslinya adalah *Navigation Satellite Timing and Ranging Global Positioning System* (NAVSTAR GPS) mempunyai tiga segmen yaitu : satelit, pengontrol, dan penerima. Satelit GPS yang mengorbit bumi, dengan orbit dan kedudukan yang tetap (koordinatnya pasti), seluruhnya berjumlah 24 buah dimana 21 buah aktif bekerja dan 3 buah sisanya adalah cadangan.

Untuk dapat mengetahui posisi seseorang maka diperlukan alat yang diberi nama GPS *reciever* yang berfungsi untuk menerima sinyal yang dikirim dari satelit GPS. Posisi diubah menjadi titik yang dikenal dengan nama *Way-point*, nantinya akan berupa titik-titik koordinat lintang dan bujur dari posisi seseorang atau suatu lokasi kemudian di layar pada peta elektronik. Sejak tahun 1980, layanan GPS yang dulunya hanya untuk keperluan militer mulai terbuka untuk publik.



Gambar 2.2 Satelit GPS

Gambar 2.2 Satelit GPS menjelaskan bahwa satelit-satelit ini bisa mengorbit pada ketinggian sekitar 12.000 Mil dari permukaan bumi. Posisi ini sangat ideal karena satelit dapat menjangkau area *coverage* yang lebih luas. Satelit-satelit ini akan selalu berada posisi yang bisa menjangkau semua area di atas permukaan bumi sehingga dapat meminimalkan terjadinya *blank spot* (area yang tidak terjangkau oleh satelit). Setiap satelit mampu mengelilingi bumi hanya dalam waktu 12 jam. Sangat cepat, sehingga mereka selalu bisa menjangkau dimana pun posisi seseorang di atas permukaan bumi.

Bagian yang paling penting dalam sistem navigasi GPS adalah beberapa satelit yang berada di orbit bumi atau yang sering kita sebut di ruang angkasa. Satelit GPS saat ini berjumlah 24 unit yang semuanya dapat memancarkan sinyal ke bumi yang lalu dapat ditangkap oleh alat

penerima sinyal tersebut atau *GPS Tracker*. Selain satelit terdapat 2 sistem lain yang saling berhubungan, sehingga jadilah 3 bagian penting dalam sistem GPS :

a. *GPS Control Segment*

Control segment GPS terdiri dari lima stasiun yang berada di pangkalan *Falcon Air Force, Colorado Springs, Ascension Island, Hawaii, Diego Garcia dan Kwajalein*. Kelima stasiun ini adalah mata dan telinga bagi GPS. Sinyal-sinyal dari satelit diterima oleh bagian kontrol, kemudian dikoreksi, dan dikirimkan kembali ke satelit. Data koreksi lokasi yang tepat dari satelit ini disebut data *ephemeris*, yang kemudian nantinya dikirimkan ke alat navigasi yang kita miliki tersebut atau *GPS Tracker*.

b. *GPS Space Segment*

Space Segment adalah terdiri dari sebuah jaringan satelit yang terdiri dari beberapa satelit yang berada pada orbit lingkaran yang terdekat dengan tinggi nominal sekitar 20.183 Km di atas permukaan bumi. Sinyal yang dipancarkan oleh seluruh satelit tersebut dapat menembus awan, plastik dan kaca, namun tidak bisa menembus benda padat seperti tembok dan rapatnya pepohonan. Terdapat 2 jenis gelombang yang hingga saat ini digunakan sebagai alat navigasi berbasis satelit. Masing-masingnya adalah gelombang L1 dan L2 dimana L1 berjalan pada frekuensi 1575.42 MHz yang bisa digunakan oleh masyarakat umum, dan L2 berjalan pada frekuensi 1227.6 Mhz dimana jenis ini hanya untuk kebutuhan militer saja.

c. *GPS User Segment*

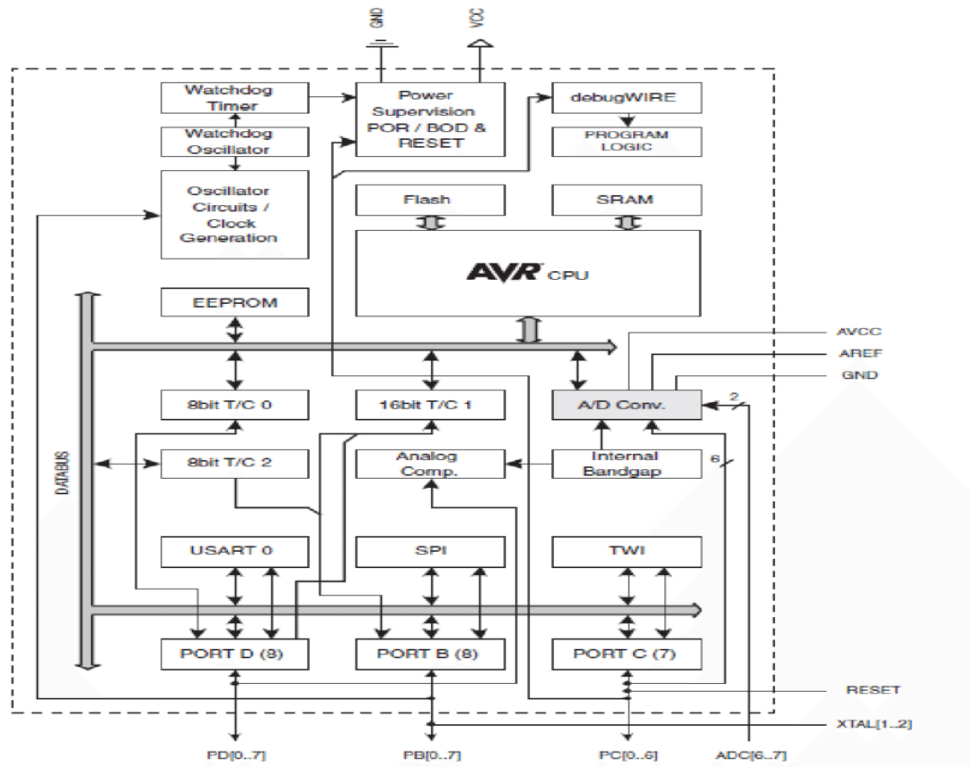
User segment terdiri dari antenna dan *prosesor receiver* yang menyediakan *positioning*, kecepatan dan ketepatan waktu ke pengguna. Bagian ini menerima data dari satelit-satelit melalui sinyal radio yang dikirimkan setelah mengalami koreksi oleh stasiun pengendali (*GPS Control Segment*).

2.2.5 Mikrokontroler Arduino

Arduino dikatakan sebagai sebuah platform dari *physical computing* yang bersifat *open source*. Ada banyak *project* dan alat-alat yang dikembangkan oleh akademisi dan profesional dengan menggunakan Arduino, selain itu juga ada banyak modul-modul pendukung (sensor, tampilan, penggerak dan sebagainya) yang dibuat oleh pihak lain untuk bisa disambungkan dengan Arduino.

Arduino berevolusi menjadi sebuah *platform* karena ia menjadi pilihan dan acuan bagi banyak praktisi. Salah satu yang membuat arduino memikat hati banyak orang adalah karena sifatnya *open source*, baik untuk *hardware* maupun *software*-nya. Komponen utama didalam papan Arduino adalah sebuah *microcontroller* 8 bit dengan merk Atmega yang dibuat oleh perusahaan *Atmel Corporation*. Berbagai papan Arduino menggunakan tipe *Atmega* yang berbeda-beda tergantung dari spesifikasinya, sebagai contoh Arduino Uno menggunakan Atmega328 sedangkan Arduino Mega 2560 yang lebih canggih menggunakan Atmega2560.

Gambar 2.3 Blok Diagram Arduino Mega 2560 ini menjelaskan bahwa *Board* ini sudah sangat lengkap, sudah memiliki segala sesuatu yang dibutuhkan untuk sebuah *mikrokontroler*, memiliki pin I/O yang cukup banyak, sejumlah 54 buah digital I/O pin (15 pin diantaranya adalah PWM), 16 pin *analog input*, 4 pin UART (*serial port hardware*). Arduino Mega 2560 dilengkapi dengan sebuah *oscillator* 16 Mhz, sebuah *port* USB, *power jack* DC, ICSP *header*, dan tombol *reset*



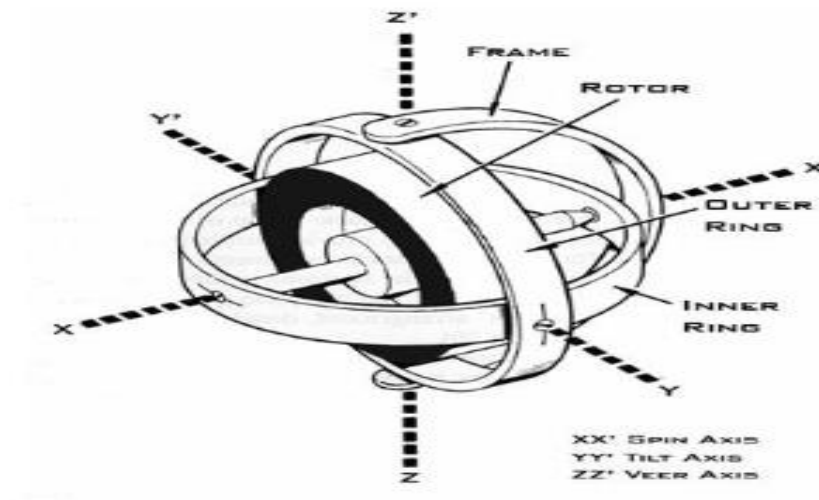
Gambar 2.3 Blok Diagram Arduino Mega 2560

Dengan penggunaan yang cukup sederhana, anda tinggal menghubungkan power dari USB ke PC anda atau melalui adaptor AC/DC ke konektor DC. Sifat *open source* arduino juga banyak memberikan keuntungan tersendiri untuk kita dalam menggunakan *board* ini, Bahasa pemrograman arduino merupakan bahasa C yang sudah disederhanakan menggunakan *syntax* bahasa pemrogramannya sehingga mempermudah kita dalam mempelajari dan mendalami *mikrokontroler*.

2.2.6 Sensor Gyroscope dan Accelerometer

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

Gambar 2.4 menjelaskan *Gyrosensor* mendeteksi gerakan sesuai gravitasi, atau dengan kata lain mendeteksi gerakan pengguna. Sebelum digunakan, sensor *gyroscope* terlebih dahulu dilakukan proses kalibrasi dengan menggunakan bandul. Proses kalibrasi tersebut berfungsi untuk memperoleh nilai faktor kalibrasi.



Gambar 2.4 Prinsip kerja *gyroscope*

Gyroscope memiliki keluaran berupa kecepatan sudut dari arah 3 sumbu yaitu: sumbu x yang nantinya akan menjadi sudut ϕ (kanan dan kiri), sumbu y nantinya menjadi sudut θ (atas dan bawah), dan sumbu z nantinya menjadi sudut ψ (depan dan belakang). Sensor *Accelerometer* yaitu sebuah *transducer* yang bekerja untuk mengukur dan mendeteksi adanya percepatan dan getaran akibat gravitasi bumi. Sensor ini mampu

bekerja untuk mengukur suatu getaran yang terjadi pada benda atau objek tertentu seperti mesin, kendaraan, bangunan, dan lebih hebatnya lagi bisa mengukur getaran di dalam bumi tanpa pengaruh gravitasi bumi.

Berdasarkan ilmu fisika dan hukum fisika bahwa bila konduktor digerakan ke medan magnet ataupun sebaliknya, maka yang terjadi adalah akan menimbulkan suatu tegangan induksi pada konduktor yang ada. *Accelerometer sensor* yang diletakkan di permukaan bumi itu bisa mendeteksi suatu percepatan atau sebuah ukuran gravitasi pada bumi di titik pusat *vertical*. Untuk pendeteksi bumi pada titik *vertical*, karena disebabkan adanya pergerakan percepatan dari *horizontal*, oleh karena itu *Sensor Accelerometer* ini akan mengukur percepatannya pada waktu itu atau secara langsung ketika percepatan itu bergerak secara *horizontal*. Karena *Sensor Accelerometer ini* dibuat oleh perusahaan atau pabriknya yang dibuat secara berbeda mulai dari jenis dan spesifikasinya karena alat ini sangat khusus untuk setiap digunakan. Sekarang pada saat ini, hampir semua alat ini berpenampilan *digital*.

BAB 3 METODE PENELITIAN

Pada Sub bab ini akan dijelaskan langkah-langkah penelitian, analisa Rancang Bangun Sistem Posisi dan *Ground Station Monitoring* pada *Solar Cell UAV (Unmanned Aerial Vehicle)*.

3.1. Desain Penelitian

Jenis penelitian pada tugas akhir ini adalah penelitian rancang bangun sistem, yaitu sebuah rancang bangun sistem identifikasi lokasi ranjau menggunakan perangkat GPS dan perangkat Arduino yang terdiri dari beberapa prosedur atau langkah-langkah dalam rancang bangun yang berkontribusi dalam mewujudkan sebuah perangkat yang mampu untuk mendapatkan data lokasi menggunakan perangkat GPS dengan unit proses menggunakan Arduino.

3.2 Prosedur Penelitian

Untuk melaksanakan penelitian dengan baik, maka persiapan langkah-langkah untuk menemukan sumber data serta melakukan penelitian sistem Rancang Bangun Sistem Posisi dan *Ground Station Monitoring* pada *Solar Cell UAV (Unmanned Aerial Vehicle)* yang mencakup instrumen penelitian, data primer, data sekunder, serta pengujian keabsahan dan keterandalan data.

3.2.1 Waktu dan Tempat Penelitian

Lokasi penelitian merupakan suatu tempat atau wilayah dimana penelitian tersebut akan dilakukan. Adapun penelitian yang dilakukan oleh penulis mengambil lokasi di Kampus STTAL Moro Krembangan Surabaya. Waktu yang digunakan dalam penelitian ini mulai bulan Juli 2020 hingga akhir Desember 2020.

3.2.2 Alat dan Bahan Penelitian

Alat dan bahan yang diperlukan untuk mendukung proses penelitian ini dan penerapan sistem.

Tabel 3.1 Alat dan bahan

No	Nama Alat dan Bahan	Tipe
1	Microcontroler	Arduino Nano 328
2	Microcontroler	Arduino Mega

3	Sensor IMU	MPU6050
4	GPS	Neo 6M
5	Telemetry	NRF24L01
6	Telemetry	NRF24L01

3.2.3 Rancangan Penelitian

Rancangan perangkat dan program berdasarkan permasalahan penelitian dikerjakan dengan metode pengembangan *prototype*. Untuk menunjang sistem pemetaan ini digunakan mikrokontroler Arduino Mega 2560 sebagai sistem pengolahan data sensor. Untuk mengirimkan data ke *ground station* digunakan perangkat nirkabel NRF24L01. Arduino Mega 2560 adalah pengendali mikro *single-board* yang bersifat *open-source*, diturunkan dari *wiring platform*, dirancang untuk memudahkan penggunaan elektronik dalam berbagai bidang. *Hardware* memiliki prosesor AtmelAVR, dan *software* memiliki bahasa pemrograman sendiri dan menggunakan Atmega2560 sebagai Mikrokontrolernya, memiliki 56 pin I/O digital dan Atmega2560 memiliki memori flash terprogram 256 KB, RAM statis 8 KB dan EEPROM 1 KB.

Kebutuhan perangkat keras menjelaskan perangkat atau komponen elektronik yang dibutuhkan pada penelitian ini. Perangkat keras yang digunakan pada penelitian ini harus dapat memenuhi semua kebutuhan minimal. Semua perangkat keras yang dibutuhkan pada penelitian dapat dijelaskan sebagai berikut :

a. Radio Transceiver NRF24

Sebuah modul *transceiver* yang bekerja pada frekuensi 2.4 Ghz menggunakan sistem modulasi GFSK. Modulasi GFSK yaitu memberikan frekuensi yang berbeda untuk operator tergantung pada bit yang ditransmisikan. Oleh karena itu, ketika 0 bit ditransmisikan, operator mengasumsikan frekuensi sesuai dengan 0 bit selama durasi satu bit. Modul *trabsceiver* menyediakan 125 kanal berbeda yang dapat digunakan sebagai modem yang dapat

bekerja secara independen pada satu tempat. Setiap saluran dapat memiliki hingga 6 alamat, atau setiap unit dapat berkomunikasi dengan hingga 6 unit lainnya secara bersamaan. Konsumsi daya NRF24 12mA selama *transmisi*, tegangan pengoperasian NRF24 adalah dari 1,9 hingga 3,6V. NRF24 dapat dihubungkan ke Arduino tanpa menggunakan konverter level logika apa pun.

b. GPS M6N

GPS adalah sistem radio navigasi satelit yang dikembangkan oleh *the U.S Dept. of Defense* (DOD) untuk keperluan navigasi global segala cuaca dimuka bumi pada sembarang waktu. Sistem ini memungkinkan pemakai GPS menentukan posisi, kecepatan gerak dalam koordinat tiga dimensi dan waktu dengan teliti. Sistem radio navigasi satelit ini terdiri dari tiga bagian yaitu : *Space Segment*, *Control Segment*, dan *User Segment*. Penentuan posisi GPS digambarkan dengan menggunakan nilai koordinat X dan Y atau garis bujur dan garis lintang (*longitude/latitude*). NEO-6M adalah salah satu modul GPS yang masuk dalam salah satu seri GPS UBLOX NEO-6 yang memiliki kinerja tinggi, *receiver* yang fleksibel dan memiliki fitur berbagai pilihan konektivitas hanya dalam miniatur 16 x 12,2 x 2,4 mm.

c. Sensor IMU

IMU adalah bagian dari navigasi system yang dikenal sebagai *Inertial Navigation System* (INS). IMU bekerja dengan memanfaatkan sensor *gyro* dan *accelerometer* untuk mendapatkan sikap *client* kapal terhadap bumi. Untuk mendapat sikap kapal pada sumbu *yaw* menggunakan pengukuran sensor gyro sementara itu mendapatkan sikap kapal terhadap sumbu *roll* dan *pitch* menggunakan kombinasi pengukuran sensor gyro dan *accelerometer*.

d. Arduino Mega

Unit pengolahan pembacaan sensor GPS, IMU dan pengolahan komunikasi pada UAV.

Kebutuhan perangkat lunak pada penelitian ini dapat dijelaskan sebagai berikut :

a. Arduino IDE 1.8

Merupakan editor mikrokontroler Arduino yang digunakan sebagai perangkat lunak pembuatan program Arduino pada sisi *client* kapal maupaun sisi *gateway* dan *ground station*.

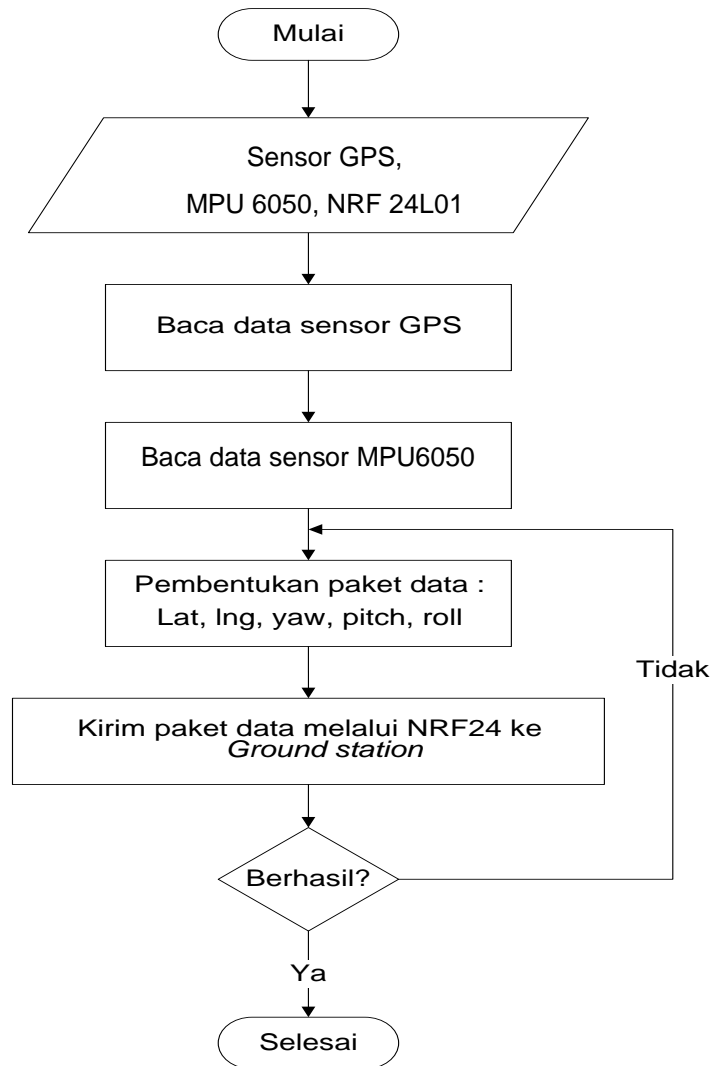
b. Visual Studio 2012 C#

Merupakan editor program yang berjalan pada PC maupun laptop untuk pembuatan program PC maupun laptop pada sisi *ground station monitoring*. Program yang menggunakan bahasa pemrograman menggunakan C# memiliki fitur untuk menampilkan semua data yang dikirim dari UAV untuk ditampilkan dalam bentuk *user interface*.

c. Windows 7 32 bit

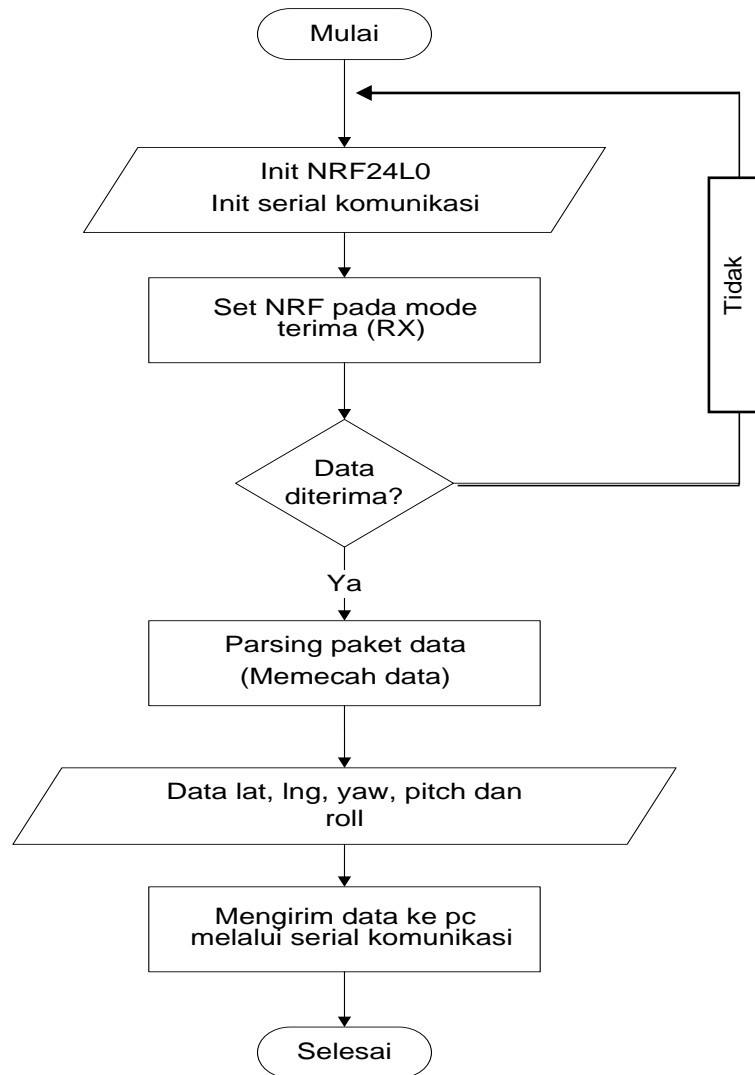
Merupakan *operating system* yang berjalan pada PC maupun laptop *ground station*.

Gambar 3.1 menjelaskan alur sistem dan proses kerja UAV, awal mula sistem akan melakukan inisialisasi GPS, MPU6050 dan NRF24L01. GPS diatur pada pin Arduino pada mode serial komunikasi. Sensor MPU6050 diatur pada mode komunikasi I2C dihubungkan ke pin SDA dan SCL. Sementara modul NRF dihubungkan ke Arduino melalui komunikasi SPI. Pada alur sistem menjelaskan proses pembacaan data GPS, MPU6050. Data koordinat yang diperoleh dari GPS dan data orientasi UAV yang diperoleh dari MPU6050 akan dikirimkan ke GCS melalui NRF24L01. Data dapat dikirimkan setiap 1 detik. Sehingga perubahan posisi UAV yang akan selalu dapat termonitor. Sistem dan proses kerja di UAV *Solar Cell* dijelaskan pada *flowchart* dibawah sebagai berikut:

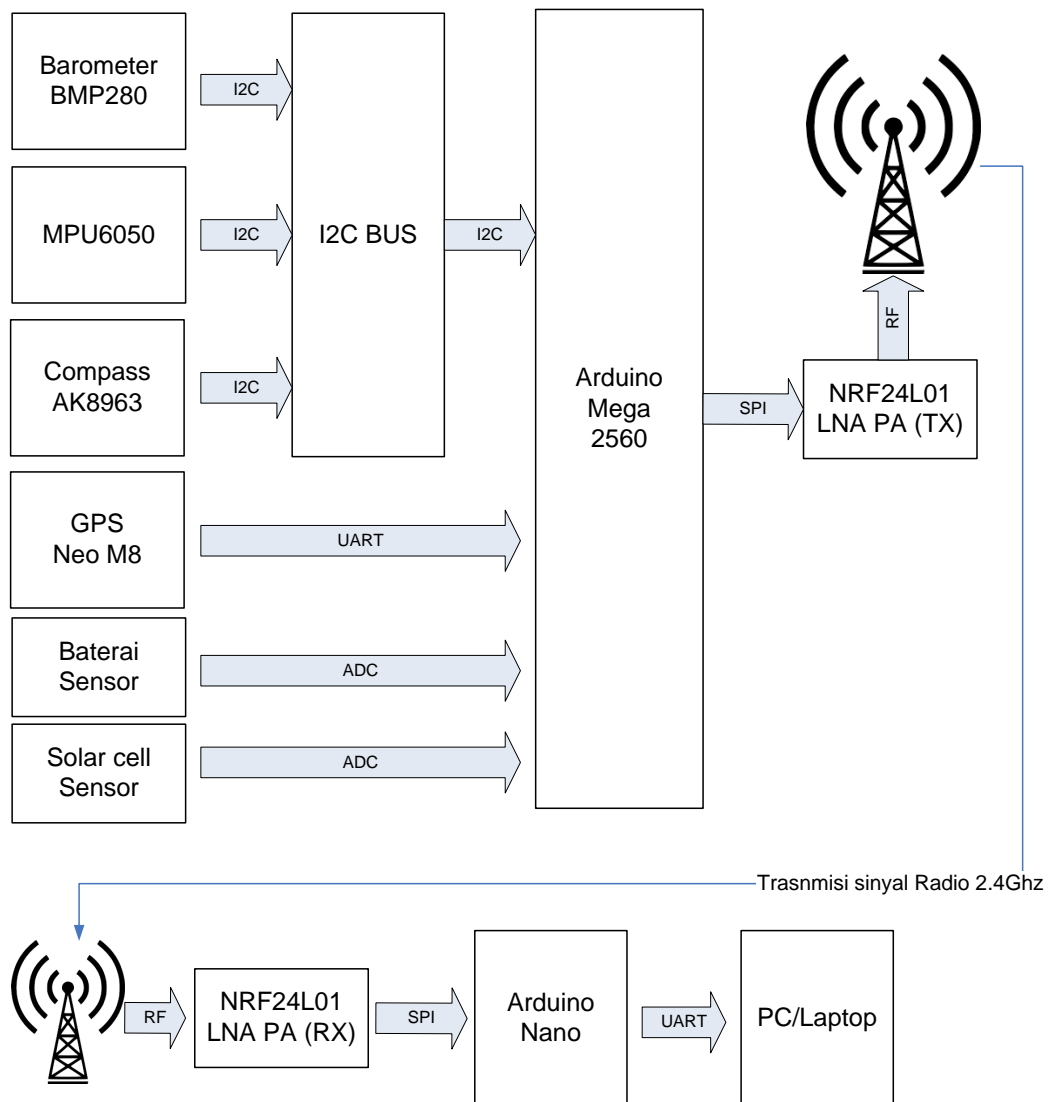


Gambar 3.1 *Flowchart* sistem posisi di UAV *Solar Cell*

Pada Gambar 3.2 dibawah ini menjelaskan alur sistem dan proses kerja pada *Ground Station Monitoring* yang dijalankan pada perangkat laptop atau *personal computer*. Pada alur sistem ini, NRF24L01 menerima data yang dikirimkan oleh perangkat UAV. Data dilakukan proses parsing atau memecah data berdasarkan karakter tertentu. Data hasil pemecahan akan ditampilkan dalam monitor *Ground Station*. Data lokasi akan ditampilkan dalam bentuk peta elektronik 2 dimensi. Sementara data orientasi UAV akan ditampilkan dalam bentuk GUI *Avionik*. Sistem dan proses kerja pada *Ground Station Monitoring* yang menggunakan pada perangkat laptop atau *personal computer*.



Gambar 3.2 *Flowchart* sistem pada *ground station*



Gambar 3.3 Blok diagram sistem posisi dan *ground station*

Gambar 3.3 menjelaskan tentang Blok diagram sistem posisi dan *ground station* perangkat keras dan perangkat lunak dibuat per bagian agar mudah dalam analisa dan pengujian. Perangkat keras dibuat secara bertahap dan dirangkai ke perangkat Arduino melalui saluran komunikasi yang sesuai. Perangkat GPS dihubungkan ke saluran komunikasi TTL arduino dengan *baudrate* 9600 Bps, sensor IMU terhubung ke Arduino melalui saluran komunikasi I2C. Untuk mentransmisikan data ke modem melalui saluran TTL komunikasi yang ditentukan pada Serial 0 Arduino.

Pembuatan perangkat lunak pada penelitian ini dapat dijabarkan sebagai berikut:

a. *UAV*

Perangkat lunak dibuat menggunakan bahasa pemrograman C++ pada *editor* program Arduino IDE 1.8.

Perangkat lunak dibuat dengan masing-masing fungsi sebagai berikut :

- 1) Mengakses sensor GPS pada saluran komunikasi TLL dan mengekstrak informasi lokasi pada format NMEA.
- 2) Mengakses sensor IMU melalui saluran komunikasi I2C dengan alamat 07H untuk mendapatkan informasi sumbu *yaw*, *pitch* dan *roll*.
- 3) Berkomunikasi dengan telemetry NRF24L01 melalui saluran komunikasi SPI.

b. *Ground Station*

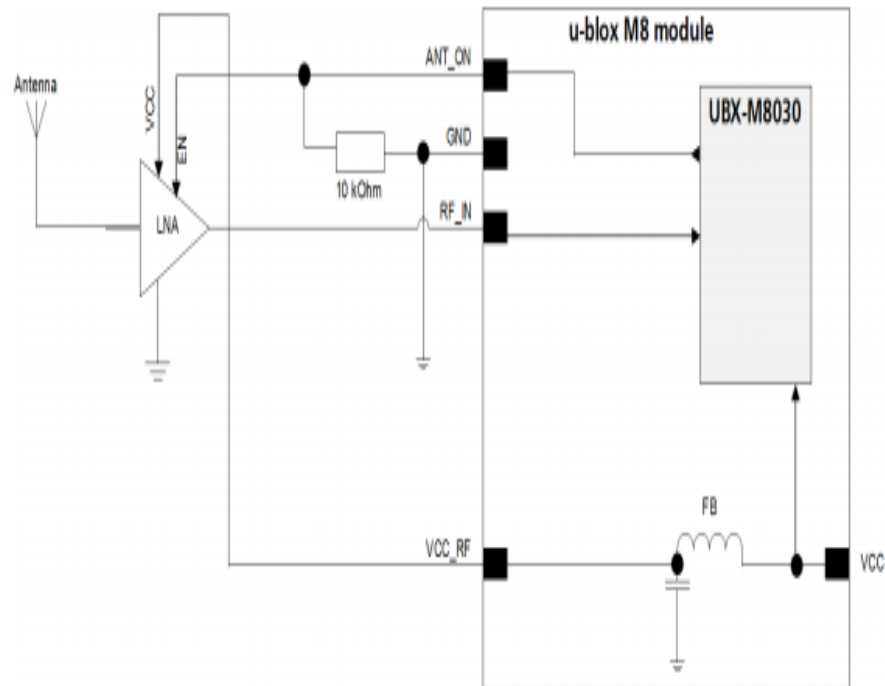
Perangkat lunak dibuat menggunakan bahasa pemrograman C# pada *editor Visual Studio* 2012. Perangkat lunak dibuat dengan masing-masing fungsi dan fitur sebagai berikut:

- 1) Menampilkan lokasi UAV pada peta elektronik menggunakan *library* GMAP.NET.
- 2) Menampilkan informasi sikap UAV dalam derajat pada antar muka program.
- 3) Menampilkan informasi kecepatan UAV dalam km/jam pada antar muka program.

Jenis – jenis modul yang dipergunakan pada penelitian ini dapat dijelaskan meliputi:

a. GPS UBLOX NEO-6

NEO-6M adalah salah satu modul GPS yang masuk dalam salah satu seri GPS UBLOX NEO-6 yang memiliki kinerja tinggi, *receiver* yang fleksibel, murah, dan menawarkan berbagai pilihan konektivitas hanya dalam miniatur 16 x 12,2 x 2,4 mm.



Gambar 3.4 GPS U - BLOK NEO6M

Gambar 3.4 GPS U - BLOK NEO6M menjelaskan arsitektur yang *compact* dan pilihan memori membuat NEO-6M ideal untuk dioperasikan dengan baterai perangkat *mobile*.

Mesin 50-channel u-blox 6 menawarkan *Time-To-FirstFix* (TTFF) di bawah 1 detik. Mesin akuisisi yang memiliki 2 juta *correlators* ini memungkinkan untuk menemukan satelit secara langsung. Serta dengan desain dan teknologi yang inovatif menjadikan NEO-6M sebuah navigasi yang paling baik bahkan di lingkungan yang ekstrim.

b. Modul MPU 6050 6-Axis

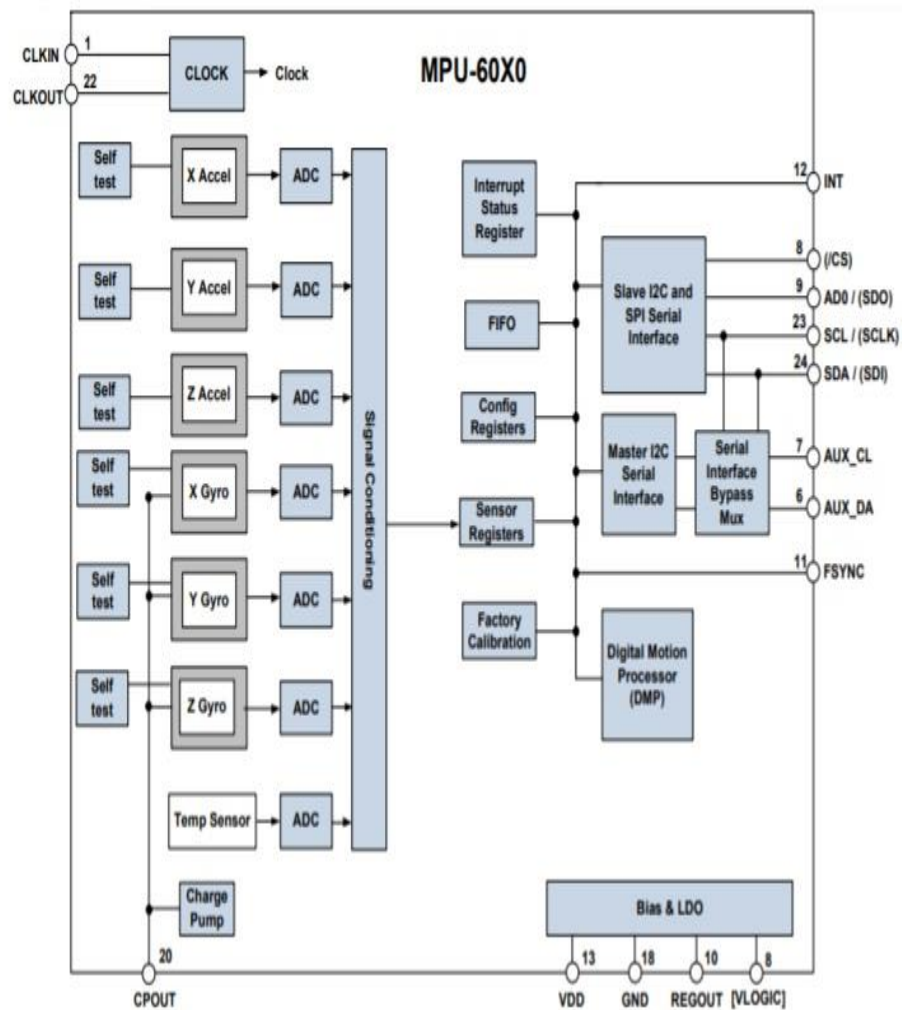
Modul MPU-6050 menerapkan teknologi *Motion Fusion* dan *run-time calibration firmware* yang menjamin kinerja optimal bagi pengguna. Dengan adanya *Digital Motion Processor* modul ini dapat diintegrasikan dengan magnetometer atau sensor lainnya lewat antarmuka I2C. Bentuk fisik modul IMU MPU6050 digunakan untuk memproses algoritma gerakan yang kompleks secara *internal* tanpa membebani kerja *mikroprosesor* utama.

Berikut spesifikasi modul IMU MPU6050:

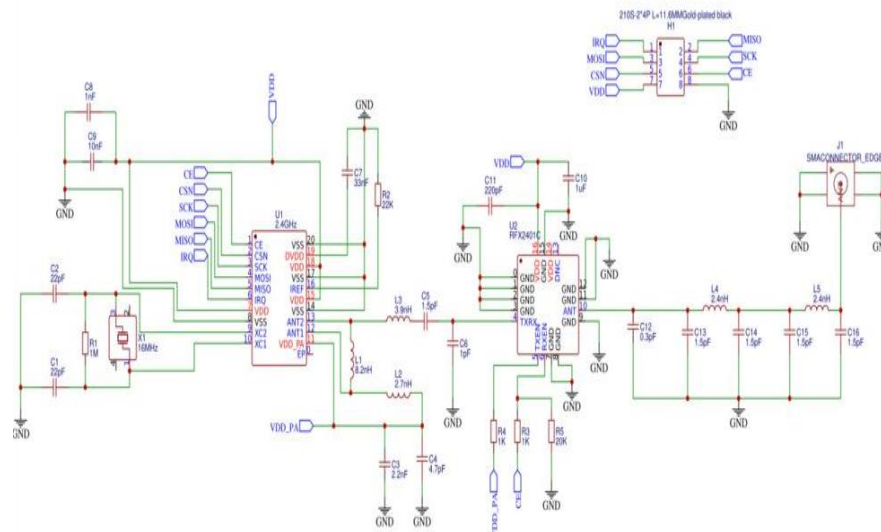
- 1) Berbasis *chip* MPU-6050
- 2) *Supply* tegangan berkisar 3-5 V
- 3) *Gyroscope range* + 250 500 1000 2000 ° / s
- 4) *Acceleration range*: $\pm 2 \pm 4 \pm 8 \pm 16$ g
- 5) *Communication* standard I2C

Sensor ini sangat akurat dengan fasilitas *hardware internal* 16 bit ADC untuk setiap kanalnya. Sensor ini akan menangkap nilai kanal axis X, Y dan Z bersamaan dalam satu waktu.

Gambar 3.5 menjelaskan sensor MPU-6050 terdapat sebuah MEMS *Accelerometer* dan sebuah MEMS *Gyro* yang saling terintegrasi.



Gambar 3.5 Sensor MPU 6050



Gambar 3.6 Modul radio NRF24L01

Gambar 3.6 menjelaskan tentang Modul Radio NRF24L01 merupakan modul NRF24L01 menggunakan GFSK modulasi. Pengguna dapat mengkonfigurasi parameter seperti frekuensi *channel*, keluaran dan daya kecepatan transmisi di udara

c. Modul NRF24L01

Modul NRF24L01 merupakan modul radio jarak jauh mendukung 2,4 GHz *transmisi* data nirkabel dengan konsumsi daya yang rendah. Pada modul ini memiliki tiga mode operasi yaitu mode *stand by*, mode RX, dan mode TX. NRF24L01 mendukung kecepatan transmisi 250 Kbps, 1 Mbps & 2 Mbps. Spesifikasi Modul NRF24L01 menggunakan GFSK modulasi. Pengguna dapat mengkonfigurasi parameter seperti frekuensi *channel*, keluaran dan daya kecepatan transmisi diudara. NRF24L01 mendukung kecepatan transmisi 250 Kbps, 1 Mbps & 2 Mbps. Modul ini menggunakan antarmuka SPI untuk berkomunikasi. Memiliki 8 pin yaitu: GND, VDD, CE, CSN, SCK, MOSI, MISO, dan IRQ.

Berikut spesifikasi dan fitur dari modul NRF24L01:

- 1) Menggunakan pita 2,4 GHz
- 2) Terdapat 126 kanal RF Menggunakan modulasi GFSK
- 3) Kecepatan data 250 Kbps, 1 Mbps, dan 2 Mbps
- 4) Antarmuka menggunakan 4 pin SPI

5) Tegangan 1,9 – 3,6 V

3.2.4 Pengumpulan Data

Pada penelitian system Rancang Bangun Sistem Posisi dan *Ground Station Monitoring* pada *Solar Cell UAV (Unmanned Aerial Vehicle)* penulis melakukan studi dengan mengumpulkan data. Metode pengumpulan data penelitian yang akan dilakukan penulis, yaitu:

a. Wawancara

Cara ini dilakukan dengan melakukan diskusi terhadap pihak yang berkompeten mengenai komunikasi data secara nirkabel.

b. Observasi

Cara ini dilakukan penulis dengan melakukan penelitian secara langsung di lapangan.

c. Dokumentasi

Cara ini dilakukan dengan mengumpulkan data dari buku, catatan, dan hasil-hasil penelitian pada instansi terkait mengenai *Global Positioning System*.

3.2.5 Pengolahan Data

Pengolahan data merupakan proses yang dilakukan penulis setelah memperoleh beberapa informasi yang dikumpulkan pada proses pengumpulan data. Data yang diperoleh akan dilakukan kajian dan menjadi referensi pada proses analisis rancangan dan perancangan sistem.

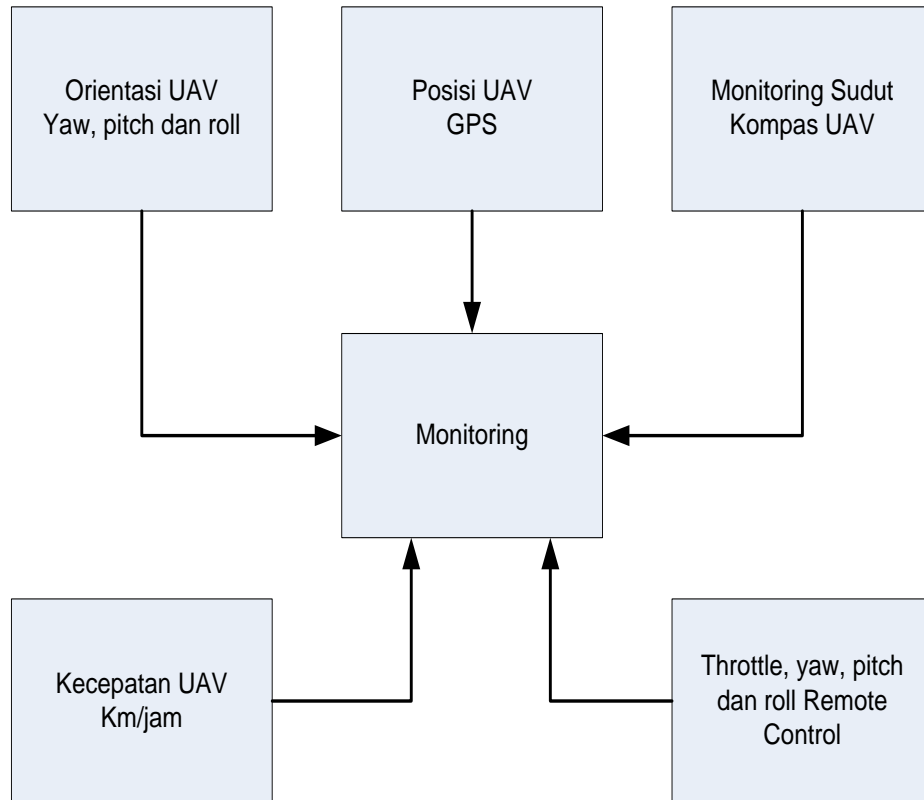
BAB 4 IMPLEMENTASI DAN PENGUJIAN SISTEM

Pada bab ini akan dijelaskan mengenai analisa dan pengujian sistem yang telah dibuat berdasarkan perancangan pada bab sebelumnya. Pengujian dilakukan melalui beberapa tahap pengujian pada masing-masing sistem untuk mendapatkan hasil keluaran yang sesuai dengan tujuan Rancang Bangun Sistem Posisi dan *Ground Station Monitoring* pada *Solar Cell UAV (Unmanned Aerial Vehicle)*. Yang tidak sesuai akan dilakukan analisa dan perbaikan hingga hasil akhir sesuai dengan tujuan rancang bangun.

Sistem monitoring dan pemetaan UAV *solar cell* terdiri dari beberapa fitur yang dapat disebutkan sebagai berikut:

- a. *Monitoring* posisi UAV berdasarkan koordinat *latitude* dan *longitude*, yang diperoleh dari data GPS.
- b. Menampilkan posisi UAV pada peta elektronik.
- c. *Monitoring* orientasi UAV yang terdiri dari sikap *yaw*, *pitch* dan *roll*.
- d. *Monitoring* sudut UAV terhadap sumbu bumi berdasarkan data kompas.
- e. *Monitoring* kecepatan perpindahan posisi UAV dalam satuan km/jam berdasarkan data GPS.
- f. *Monitoring throttle*, *yaw*, *pitch* dan *roll remote control*.

Fitur-fitur yang tersedia pada *Ground Station Monitoring* Fitur-fitur tersebut digunakan untuk memonitor status dan kondisi UAV saat melaksanakan penerbangan.



Gambar 4.1 Fitur-fitur *Ground Station Monitoring*

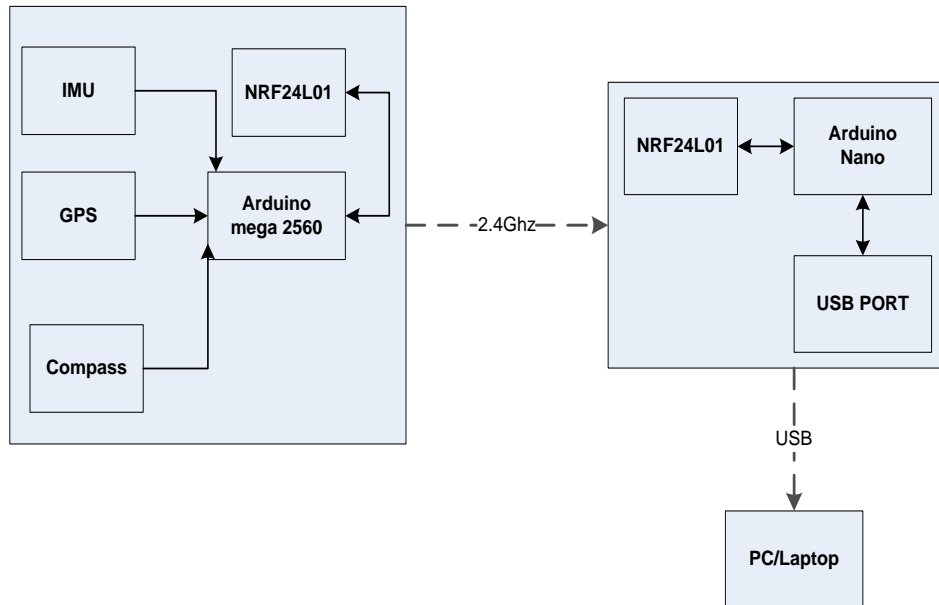
Gambar 4.1 menjelaskan Data-data yang disajikan diperoleh dari pesawat UAV yang dikirimkan ke GCS melalui komunikasi wireless pada frekuensi 2.4Ghz.

4.1 Implementasi

Pada tahap implementasi yang dilakukan pada Rancang Bangun Sistem Posisi dan *Ground Station Monitoring* pada *Solar Cell UAV (Unmanned Aerial Vehicle)* yang berada di *Ground Control Station (GCS)* dapat dijabarkan pada langkah-langkah implementasi berikut ini.

4.1.1 Implementasi Perangkat Komunikasi

Perangkat komunikasi yang digunakan pada Rancang Bangun Sistem Posisi dan *Ground Station Monitoring* pada *Solar Cell UAV (Unmanned Aerial Vehicle)*. Merupakan perangkat komunikasi data yang bertujuan untuk mengirimkan dan menerima data dari perangkat pesawat UAV.



Gambar 4.2 Diagram Blok Komunikasi Data UAV Dan GCS

Gambar 4.2 menjelaskan diagram blok sistem komunikasi pada perangkat UAV sebagai pengirim dan diagram blok sistem komunikasi pada penerima *Ground Control Station* (GCS).

Agar perangkat komunikasi dapat bekerja dengan baik sesuai dengan fungsi dan tujuan penggunaan perangkat, maka harus dihubungkan dengan pin-pin mikrokontroler sesuai dengan instalasi perangkat yang telah dijelaskan pada bab sebelumnya. Selain hubungan secara elektrik dengan mikrokontroler, agar dapat bekerja dengan baik maka pada perangkat mikrokontroler harus dilakukan pemrograman menggunakan bahasa C++ dengan perantara editor arduino IDE.

4.1.2 Implementasi Perangkat Proses

Perangkat atau unit proses untuk memproses data yang akan dikirimkan ke *Ground Control Station* (GCS) melalui NRF24L01 berintikan *chip processor* Atmega 2560 yang dikemas pada sebuah *board* dengan nama arduino Mega 2560. Kemampuan arduino mega untuk melakukan proses pembacaan sensor pada implementasi 4.1.1 diyakini dapat melakukan proses pembacaan dengan baik. Berdasarkan *datasheet chip*, diperoleh informasi *flash memory* yang cukup besar yaitu 256kb, sehingga mampu untuk menampung program yang cukup besar. Secara umum pada

penerapan mikokontroler ini untuk menjalankan semua fungsi proses, dapat dijelaskan sebagai berikut.

- a. Melakukan proses pembacaan sensor *inersia*, GPS, *compass*, dan *barometer*.
- b. Melakukan komunikasi data melalui telemetry NRF24L01.

Perangkat penerima pada GCS yang terdiri dari Arduino Nano dan NRF24L01, merupakan unit proses penerima data yang dikirimkan oleh UAV dan disalurkan ke PC/laptop melalui saluran komunikasi USB.

4.2 Pengujian

Pengujian merupakan langkah yang dilakukan setelah proses implementasi perangkat keras dilakukan. Pengujian dilakukan perbagian untuk memastikan bahwa setiap perangkat yang di pasang dapat berjalan dengan baik, sehingga jika ditemui permasalahan maka proses *trouble shooting* dapat dengan cepat diperbaiki. Sesuai dengan perancangan perangkat keras dan lunak maka proses pengujian dapat dibagi menjadi dua bagian penting, yaitu pengujian secara parsial dan pengujian secara keseluruhan. Pengujian parsial meliputi pengujian komponen sensor dan perangkat keras pendukung serta perangkat lunak yang berkaitan.

4.2.1 Pengujian Perangkat Keras

Pada pengujian ini, penulis akan melakukan pengujian komponen dan menghubungkan ke mikrokontroler Arduino mega 2560 serta mengamati proses kerja perangkat dengan mengisi program uji pada mikrokontroler. Hasil pengujian untuk membuktikan berhasil tidaknya perangkat bekerja dapat dilihat pada terminal monitor perangkat lunak Arduino IDE.

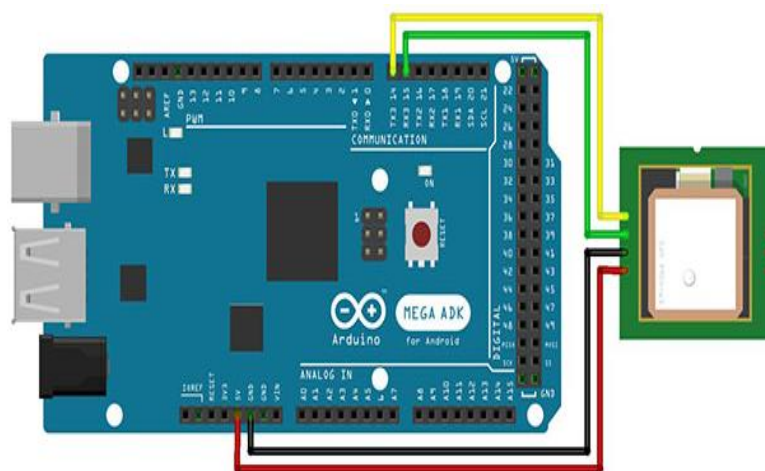
4.2.1.1 Pengujian Modul GPS

Modul GPS memegang peranan dalam menginformasikan lokasi roket, data keluaran GPS merupakan koordinat lokasi yang diwakili dengan *variabel latitude dan longitude*. Syarat koordinat lokasi GPS dinyatakan fix

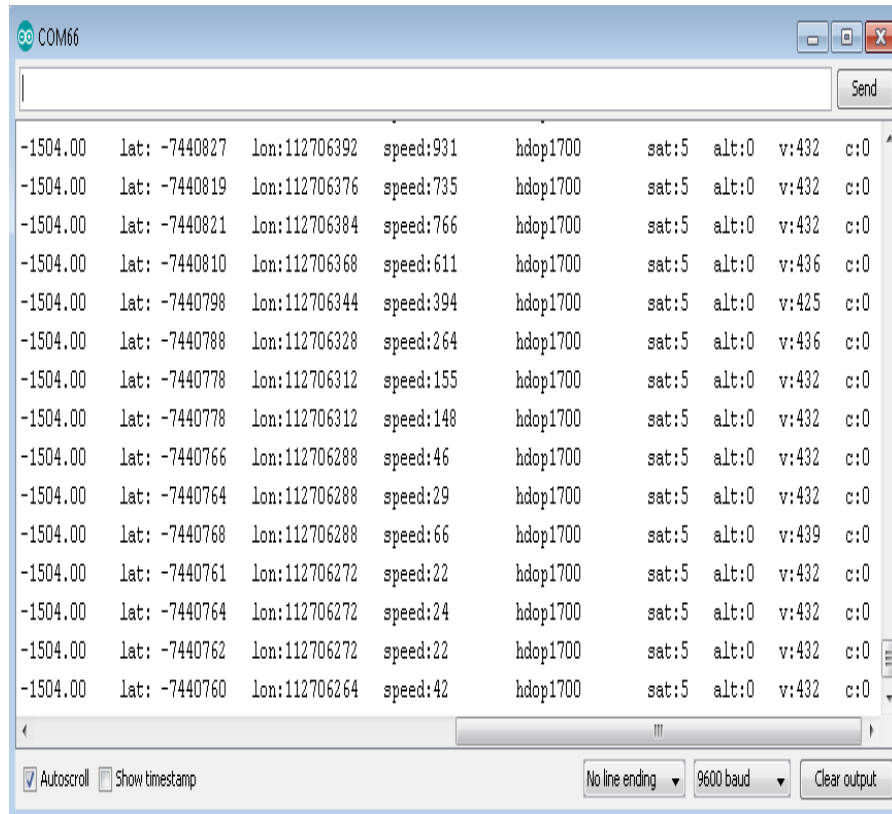
adalah minimal mendapat sinyal dari 3 satelit GPS dari luar bumi. Kondisi lingkungan yang mendukung sangat mempengaruhi penerimaan sinyal GPS. Semakin banyak sinyal GPS yang diperoleh dari satelit maka lokasi roket akan semakin akurat. Sinyal GPS berisi 3 jenis informasi berbeda:

- a. Kode pseudorandom adalah sebuah I.D. kode yang berfungsi untuk mengidentifikasi satelit mana yang mengirimkan informasi.
- b. Data Ephemeris diperlukan untuk menentukan posisi satelit dan memberikan informasi penting tentang kesehatan satelit, tanggal dan waktu saat ini.
- c. Data Almanac memberitahu penerima GPS di mana setiap satelit GPS secara *real time* menunjukkan informasi orbit untuk satelit itu dan setiap satelit lain dalam sistem.

Pada Gambar 4.3 menjelaskan pengujian modul GPS Ublox Neo-08M yang menghubungkan GPS dengan perangkat Arduino seperti *wiring diagram*. Transaksi data antar GPS dengan mikrokontroler melalui komunikasi Serial 1 Arduino Mega 2560, *default* baud rate GPS adalah 9600, sehingga pada pengaturan *baudrate* program menyesuaikan dengan baudrate GPS. Pengujian modul GPS Ublox Neo-08M yang menghubungkan GPS dengan perangkat Arduino seperti *wiring diagram*.



Gambar 4.3 Wiring Diagram Pengujian GPS.

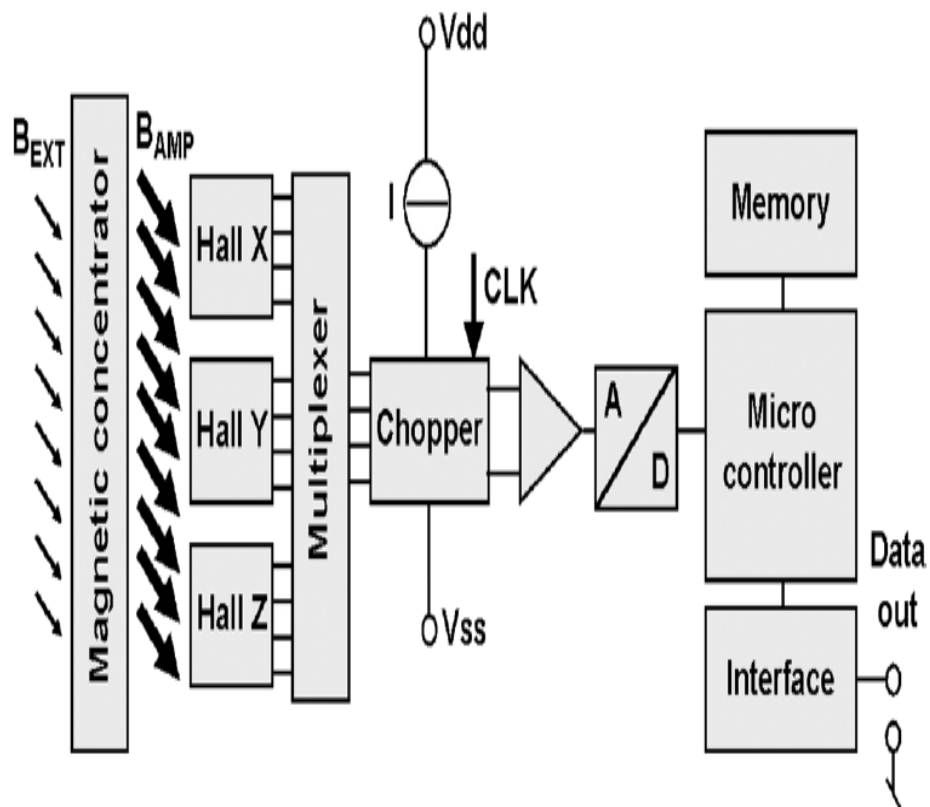


Gambar 4.4 Pengujian GPS *Ublox Neo-8M*

Gambar 4.4 menjelaskan mengenai hasil pengujian dapat dijelaskan dalam bentuk pengujian GPS, pengujian ini menggunakan fitur terminal monitor pada arduino IDE untuk menampilkan hasil pembacaan data lokasi GPS dalam format koordinat *latitude* dan *longitude*.

4.2.1.2 Pengujian Kompas MPU9250

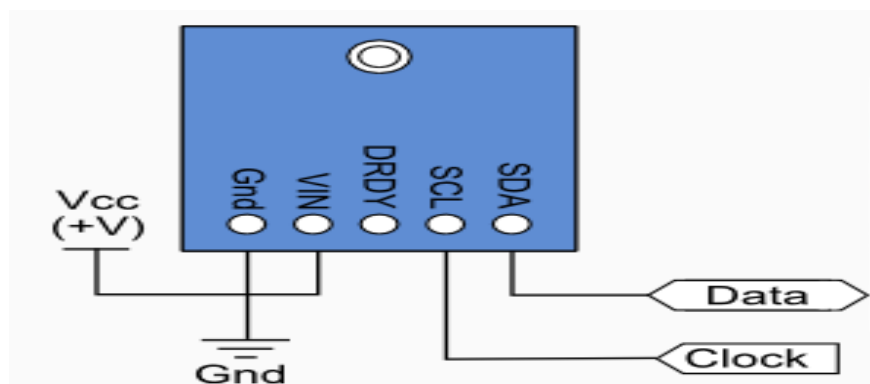
Modul magnetometer 3 sumbu merupakan modul kompas yang sangat mudah digunakan. Hanya membutuhkan 2 pin daya dan 2 pin data perangkat ini sudah bisa digunakan. 2 pin data dihubungkan ke perangkat Arduino pada *port* I2C yaitu SCL dan SDA. Prinsip kerja kompas digital sangat bergantung pada medan magnet pada sumbu XYZ. *Konsentrator Magnetic* menerima dan menguatkan medan magnet dan mengubahnya menjadi besarnya medan magnet dalam tiga sumbu, sumbu XYZ. Keluaran dalam bentuk sinyal analog.



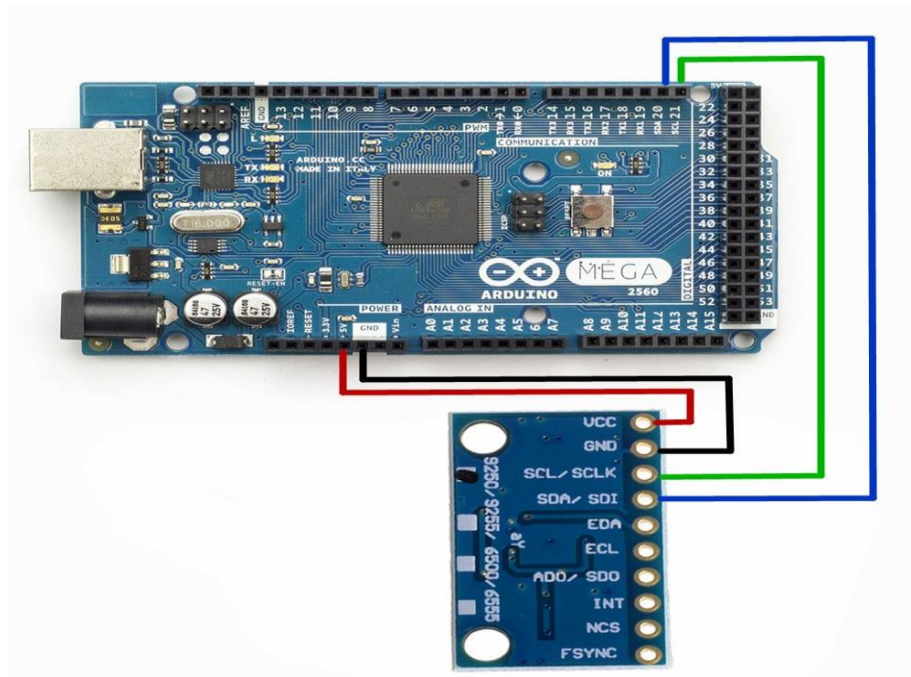
Gambar 4.5 Diagram Blok Kompas.

Gambar 4.5 menjelaskan ditetapkannya tentang sinyal *Analog Multiplexing* akan melakukan sampling untuk mengkonversi ke sinyal digital. Sinyal digital dilakukan proses perhitungan menggunakan mikrokontroler.

Gambar 4.6 menjelaskan tentang sebuah modul kompas dapat dihubungkan ke perangkat Arduino melalui pin-pin kompas.



Gambar 4.6 Pin-pin Kompas.



Gambar 4.7 Wiring Diagram Pengujian Kompas

Gambar. 4.7 menjelaskan mengenai Modul kompas MPU9250 terhubung ke Arduino Mega 2560. Hasil pengujian dapat dijelaskan dalam bentuk gambar-gambar pengujian, pada pengujian ini peneliti menggunakan fitur terminal monitor pada arduino ide untuk menampilkan hasil pembacaan sensor kompas dalam 3 sumbu x,y dan z.

Gambar 4.8 berikut merupakan hasil pengujian pembacaan sensor kompas dan ditampilkan pada terminal monitor.

Imu:::353.00	65.22	p: -2.10
Imu:::354.00	65.23	p: -2.10
Imu:::353.00	65.27	p: -2.13
Imu:::353.00	65.29	p: -2.10
Imu:::354.00	65.32	p: -2.11
Imu:::353.00	65.34	p: -2.09
Imu:::352.00	65.36	p: -2.09
Imu:::353.00	65.38	p: -2.08
Imu:::354.00	65.40	p: -2.10

Gambar 4.8 Pengujian Kompas MPU9250

4.2.1.3 Pengujian Sensor Barometer BMP280

Seperti diketahui bahwa tekanan udara dikenal sebagai tekanan atmosfer, adalah tekanan yang disebabkan oleh tekanan udara yang menekan Bumi. Sebuah kolom udara naik dari permukaan Bumi ke puncak atmosfer. Udara di atmosfer memiliki massa, sehingga gravitasi menyebabkan berat kolom itu untuk menekan di permukaan.

Tekanan yang diciptakan oleh kolom udara 1 × 1 inci yang mencapai puncak atmosfer didefinisikan sebagai satu tekanan atmosfer (atm). Kolom udara ini memiliki berat 66,7 Kg, itulah sebabnya mengapa satu atm sama dengan 66,7 pon per inci persegi (psi). Satuan SI untuk tekanan adalah Pascal (Pa). Satu Pascal didefinisikan sebagai satu Newton gaya per meter persegi. Pembacaan tekanan keluaran BMP280 dalam satuan pascal, oleh karenanya perlu proses dikonversi ke hectoPascals (hPa) menggunakan *library* program yang peneliti gunakan.

BMP280 adalah sensor *piezoresistif* yang mendeteksi tekanan. Sensor piezoresistif terdiri dari bahan semikonduktor yang mengubah resistensi ketika gaya mekanis seperti tekanan atmosfer. BMP280 mengukur tekanan dan temperatur, karena suhu mengubah densitas gas seperti udara. Pada suhu yang lebih tinggi, udara tidak sepadat dan berat, jadi itu mengurangi tekanan pada sensor. Pada suhu yang lebih rendah, udara lebih padat dan berat lebih banyak, sehingga memberikan lebih banyak tekanan pada sensor.

Sensor menggunakan pengukuran suhu secara *real time* untuk mengkompensasi pembacaan tekanan pada perubahan dalam densitas udara. BMP280 berkomunikasi dengan Arduino melalui pin I2C, pada *board* mega 2560 yang digunakan, pin SDA terletak pada pin 20 dan pin SCL terletak pada pin 21. Sebagai suplai daya dc pin daya sensor BMP280 dihubungkan pada pin vcc 3.3Vdc dan GND Arduino Mega 2560.

```

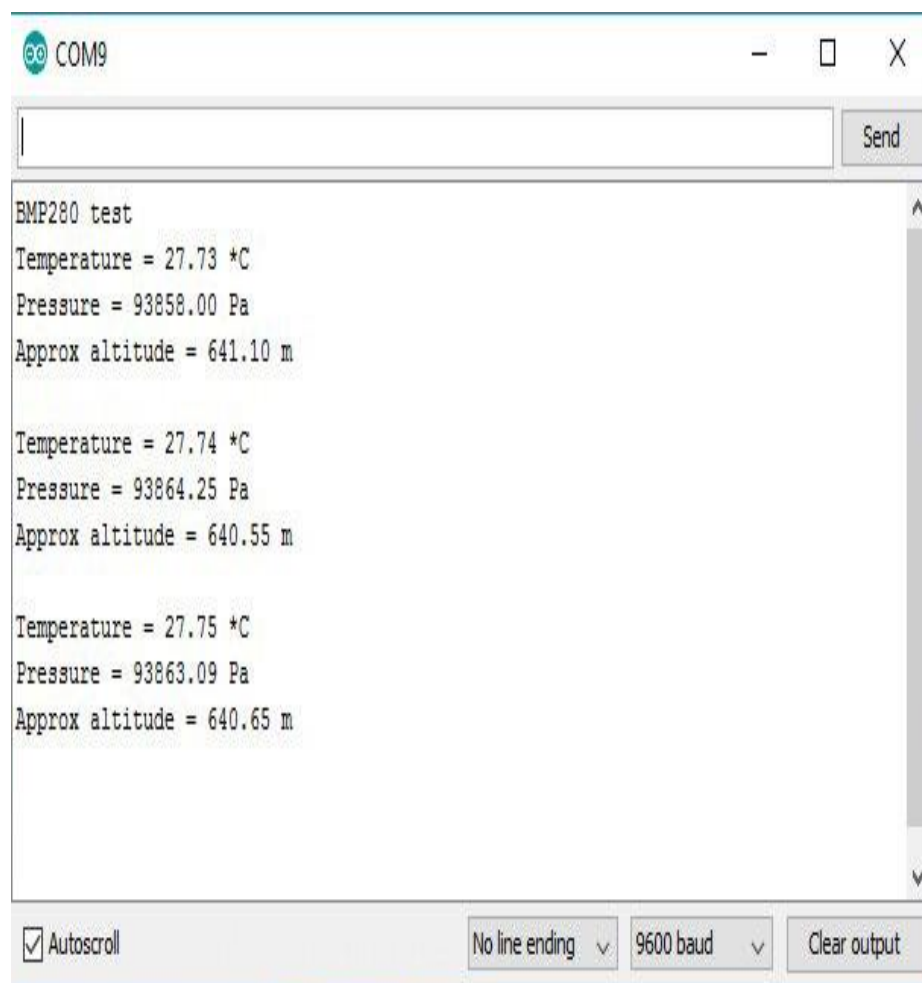
Po=1013.25 //(Average sea-level pressure)
p = bmp.readPressure();
T = bmp.readTemperature();
altitude=(((p/Po)^(1/5.2557))*(T+273.15))/0.0065

```

Gambar 4.9 Tampilan program Arduino perhitungan ketinggian (*altitude*)

Gambar 4.9 berikut merupakan gambar kode program pembacaan sensor BMP pada editor Arduino IDE 1.8.0.

Gambar 4.10 merupakan hasil pengujian sensor BMP180 pada terminal monitor arduino IDE.



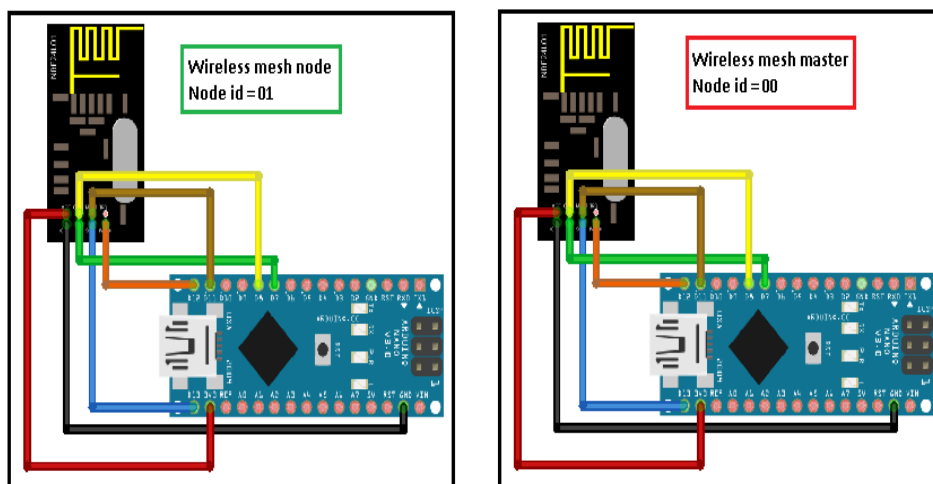
Gambar 4.10 Hasil Pengujian BMP280 Pada Terminal Monitor

4.2.1.4 Pengujian Perangkat Wireless

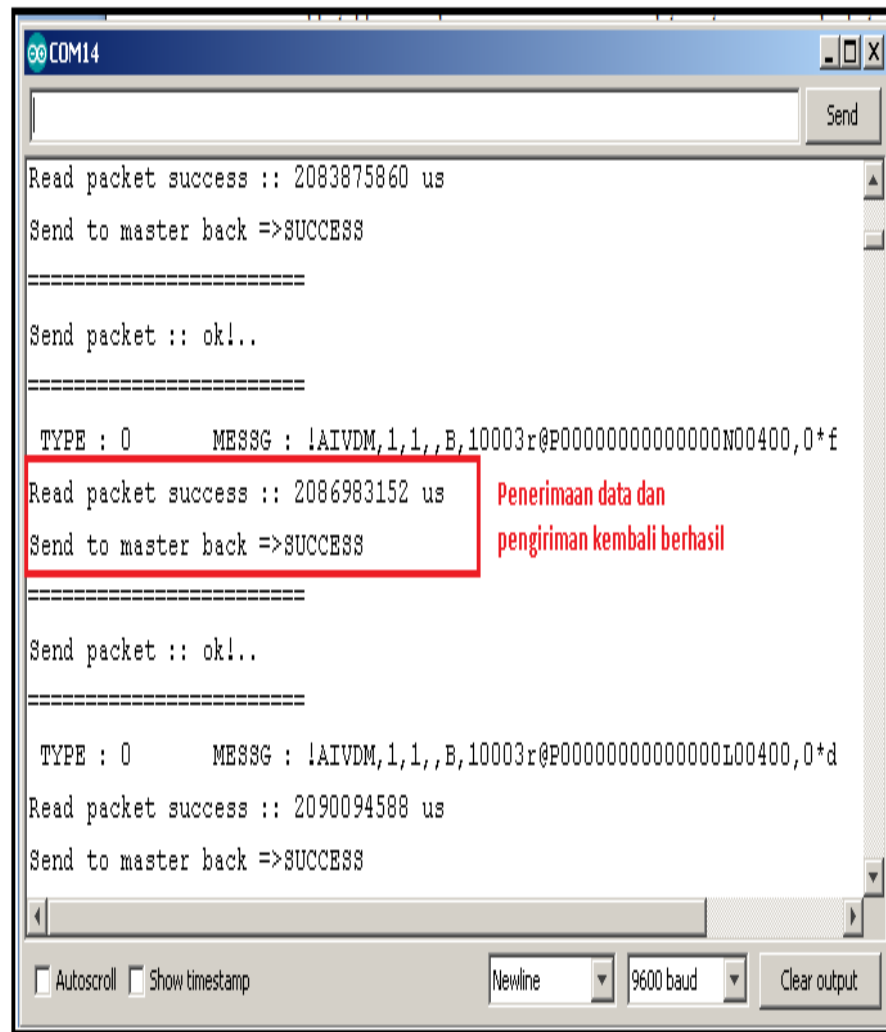
Pada pengujian perangkat *Wireless*, penulis menggunakan NRF24L01 2.4GHz yang terdiri dari antenna ping tale, beroperasi pada pita 2,4 GHz ISM dengan lebar saluran 1 MHz yang terdapat 125 saluran yang dapat digunakan. Karena perangkat Wi-Fi menggunakan sebagian besar saluran yang rendah sehingga diatur menggunakan saluran tertinggi pada perangkat ini untuk menghindari interferensi dengan saluran WIFI. Modul NRF24L01 dihubungkan dengan unit mikrokontroler menggunakan komunikasi SPI.

Untuk melakukan percobaan ini, penulis menggunakan 2 perangkat yang berfungsi sebagai *Wireless mesh node* dan *Wireless mesh master*. Hal penting yang perlu dilakukan adalah mengatur Node id masing-masing perangkat. Untuk *Wireless mesh node*, seperti yang telah disebutkan diatas didefinisikan pada Node id 01 dan *Wireless mesh master* didefinisikan pada Node id 00.

Gambar 4.11 merupakan gambar wiring diagram pengujian modul nirkabel NRF24L01. Pada pengujian yang dilakukan *Wireless mesh master* akan mengirimkan data timer yang terintegrasi pada chip mikrokontroler arduino Nano 328p ke *Wireless mesh node* 00. *Wireless mesh node* akan mereplay data yang diterima ke *Wireless mesh master* kembali.



Gambar 4.11 Gambar *wiring digaram* pengujian modul nirkabel NRF24L01



Gambar 4.12 Pengujian penerimaan data dari Master dan pengiriman kembali ke Master

Gambar 4.12 merupakan hasil pengujian penerimaan data dari Master node 00 ke node 01 dan proses pengiriman kembali ke Master node. Hal ini bertujuan untuk mengetahui apakah komunikasi dapat berjalan dengan baik dan untuk mengukur waktu yang dibutuhkan pada proses pengiriman data atau yang disebut *End Delay Measurement*.

Hasil pengujian jarak Modul NRF24L01 untuk kemampuan pengiriman data dalam kondisi tanpa terhalang bangunan ataupun sesuatu yang menghalangi.

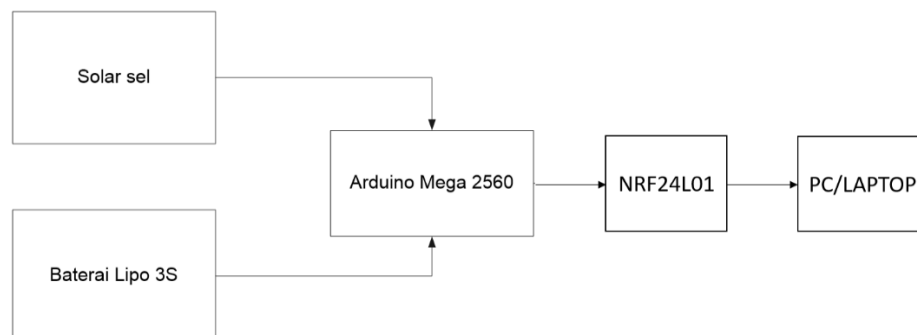
Tabel 4.1 Pengujian Modul NRF24L01

No	Jarak (meter)	Penerimaan
1.	1	Baik
2.	10	Baik
3.	50	Baik
4.	100	Baik
5.	200	Baik
6.	210	Tidak baik

Dari tabel 4.1 dapat diperoleh penerimaan komunikasi data yang baik pada jarak 1 – 200 meter, untuk jarak lebih dari 200 meter penerimaan data pada *receiver* yang diterima di GCS tidak baik, hal itu ditunjukan tidak adanya keterangan yang tampil pada peta elektronik di *ground station*.

4.2.1.5 Pengujian Sistem *Power Management* pada GCS

Pada tahap pengujian penulis akan menguji sistem yang terintegrasi secara keseluruhan. Data pengujian diperoleh melalui *ground station*. Gambar 4.13 Menjelaskan diagram blok sistem *power management* yang terintegrasi dengan sistem UAV yang terhubung dengan GCS.



Gambar 4.13 Diagram Blok Sistem *Power Management*

Secara umum pada penerapan mikokontroler ini untuk menjalankan semua fungsi proses, dapat dijelaskan sebagai berikut:

- a. Melakukan proses pembacaan sensor arus.
- b. Melakukan komunikasi data melalui telemetry NRF24L01 menggunakan komunikasi SPI.

Perangkat penerima pada GCS yang terdiri dari Arduino Nano dan NRF24L01, merupakan unit proses penerima data yang dikirimkan oleh UAV dan disalurkan ke PC/laptop melalui saluran komunikasi USB. Alur kerja sistem dapat dijelaskan sebagai berikut:

- a. Sistem melakukan pengecekan sumber *Solar Cell*, jika nilai tegangan solar melebihi 16.8 volt maka sumber tegangan solar digunakan untuk mengisi baterai.
- b. Sistem melakukan pengecekan besar tegangan baterai, jika tegangan baterai kurang dari 16.8V maka *Solar Cell* mengisi baterai. Arus pengisian baterai dibatasi 1A.
- c. Jika tegangan *Solar Cell* lebih kecil dari tegangan baterai, maka sumber daya utama diambil dari baterai.
- d. Jika tegangan solar cukup menyuplai daya motor UAV saat terbang, maka sumber baterai akan diputus.

Gambar 4.14 menunjukkan hasil pengukuran sumber daya elektrik saat UAV terbang. Berdasarkan pengukuran daya *Solar Cell*, diperoleh nilai tegangan *Solar Cell* sebesar 14.85 Volt dc. Arus yang terserap ke beban motor sebesar 12.36A dan daya yang terserap sebesar 183.52 Watt.

Solar measurement	
Voltage	14.85 V
Current	12.36 A
Power	183.52 W

Gambar 4.14 Instrumen Ukur *Solar Cell*

Battery charge	
Voltage	15.14 V
Current	0.04 A
Power	0.61 W

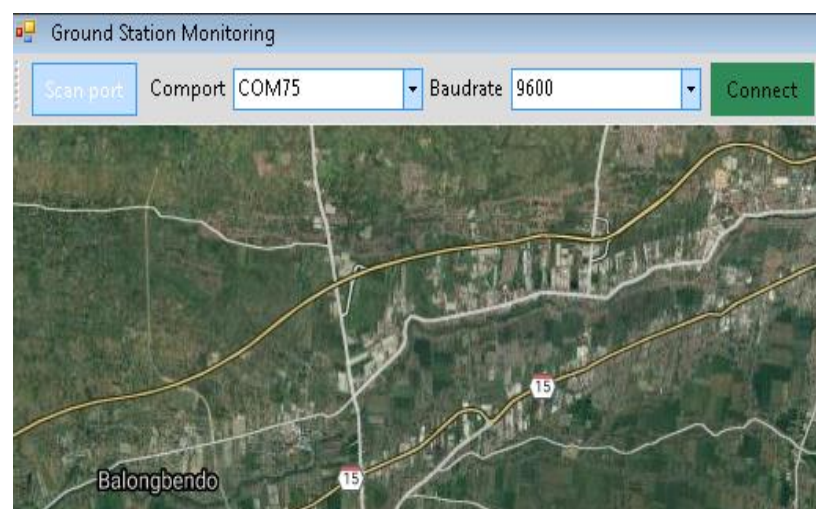
Gambar 4.15 Instrumen Ukur Baterai

Pada Gambar 4.15 ditunjukkan pengukuran sumber daya baterai, saat dilakukan uji terbang terlihat bahwa tegangan *Solar Cell* lebih kecil dari baterai, sehingga saat uji ini baterai tidak dilakukan pengisian daya, namun baterai secara periodik terserap sumber dayanya untuk mendukung *Solar Cell*.

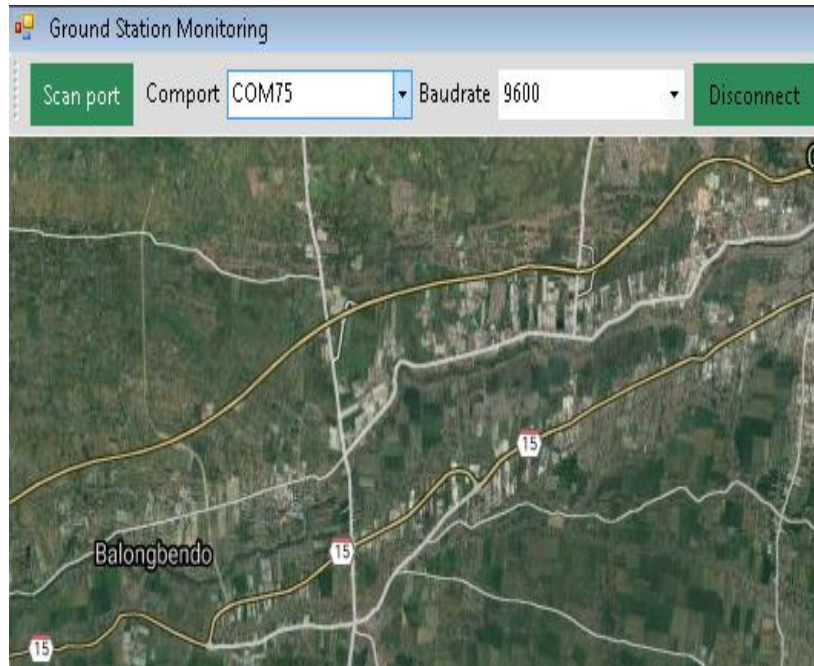
4.2.1.6 Pengujian Ground Station Monitoring

Pada pengujian *Ground Station Monitoring* penulis menghubungkan perangkat penerima pada port USB dan mengatur *comport* pada port 75 dan *baudrate* 9600. Interval waktu pengiriman data dari perangkat UAV adalah 1000 ms yang artinya GCS dapat menerima data setiap 1000 ms.

Gambar 4.16 menjelaskan proses menghubungkan penerima pada GCS dan pengaturan yang dilakukan agar penerima dapat terhubung dengan program GCS secara tepat.



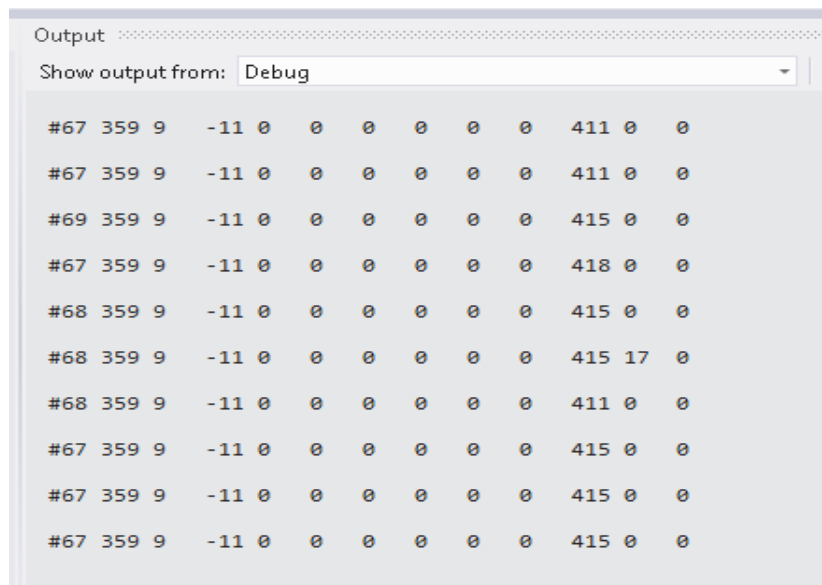
Gambar 4.16 Tampilan pengaturan komunikasi serial pada GCS



Gambar 4.17 Tampilan koneksi USB berhasil

Gambar 4.17 menjelaskan jika koneksi ke USB berhasil maka *button Connect* akan menampilkan keterangan *disconnect*.

Gambar 4.18 menjelaskan pengujian penerimaan data pada GCS dapat ditampilkan pada *output* program Visual Studio. Data yang diperoleh oleh GCS tersebut dilakukan proses *parsing* atau memecah data berdasarkan karakter spasi.



Gambar 4.18 Tampilan Data Penerimaan GCS

Data awal # didefinisikan sebagai *header* data dan karakter `\n` didefinisikan sebagai *tail* atau akhiran data. Data yang diperoleh dilakukan parsing sehingga diperoleh 12 data yang mewakili variabel data *compass*, *pitch*, *roll*, *latitude*, *longitude*, *speed*, *altitude*, jumlah satelit, akurasi GPS (Hdop), tegangan baterai, arus baterai dan tegangan solar panel. Proses *parsing* data dijelaskan pada kode program C# berikut:

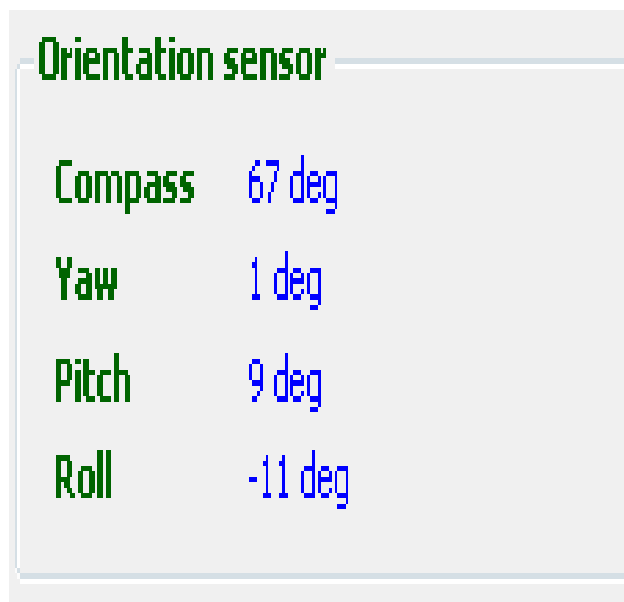
```
String data = serial.ReadLine();
data.Trim();
data.Replace("\r", "");
data.Replace("\n", "");
data.Replace("\r", "");
data.Replace("\n", "");
data.Replace("\r\n", "");
try{
    Invoke((MethodInvoker)delegate {
        Console.WriteLine(data);
        if (data.Contains("#")){
            data = data.Replace("#", "");
            String [] parse = data.Split('\t');
            //compass,yaw,pitch,roll          lat,lng,speed,sat,hdop,altitude
            bat_voltage, current, solar_voltage
            if ( parse.Length > 0) {
                lblCompass.Text = parse[0] + " deg";
                lblYaw.Text = parse[1] + " deg";
                lblPitch.Text = parse[2] + " deg";
                lblRoll.Text = parse[3] + " deg";
                lat = (double)Convert.ToInt32(parse[4]) / 1000000;
                lng = (double)Convert.ToInt32(parse[5]) / 1000000;
                lblLat.Text = string.Format("{0:0.000000}", lat);
                lblLng.Text = string.Format("{0:0.000000}", lng);
                double speed = (double)Convert.ToInt32(parse[6]) / 100;
```

```

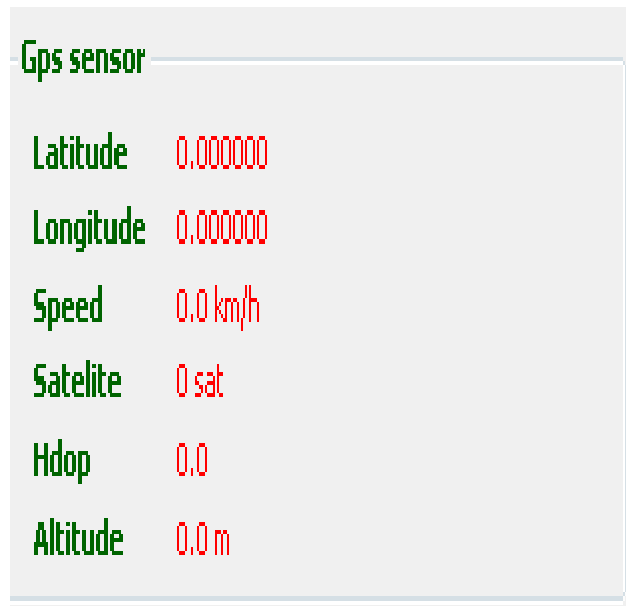
lblSpeed.Text = string.Format("{0:0.0}", speed) + " km/h";
lblSat.Text = parse[7] + " sat";
double hdop = (double)Convert.ToInt32(parse[8]) / 100;
lblHdop.Text = string.Format("{0:0.0}", hdop) + "";
double altitude = (double)Convert.ToInt32(parse[9]) / 100;
lblAltitude.Text = string.Format("{0:0.0}", altitude) + " m";
double voltage = (double)Convert.ToInt32(parse[10]) / 100;
lblVoltage.Text = string.Format("{0:0.0}", voltage) + " V";
double current = (double)Convert.ToInt32(parse[11]) / 100;
lblCurrent.Text = string.Format("{0:0.0}", current) + " A";
double power = voltage * current;
lblPower.Text = string.Format("{0:0.0}", power) + " W";
}
}

```

Gambar 4.19 menjelaskan mengenai tampilan orientasi UAV ditampilkan pada tampilan antar muka pengguna. Data orientasi menjelaskan nilai kompas, *yaw*, *pitch* dan *roll* UAV dalam derajat. Data kompas menunjukkan sudut UAV terhadap sumbu bumi berdasarkan data kompas. Data *yaw*, *pitch* dan *roll* diperoleh dari data GYRO MPU9250.



Gambar 4.19 Tampilan Data Orientasi UAV



Gambar 4.20 Tampilan Kegagalan Data GPS

Gambar 4.20 merupakan saat data GPS belum diperoleh karena factor cuaca seperti mendung sehingga penerima GPS tidak dapat menerima data satelit.

Gambar 4.21 merupakan hasil uji UAV yang ditampilkan pada peta elektronik sampai Gambar 4.22. Pada hasil uji dijelaskan marker UAV pada peta pada orientasi 0 °, 90 ° dan 180 °.



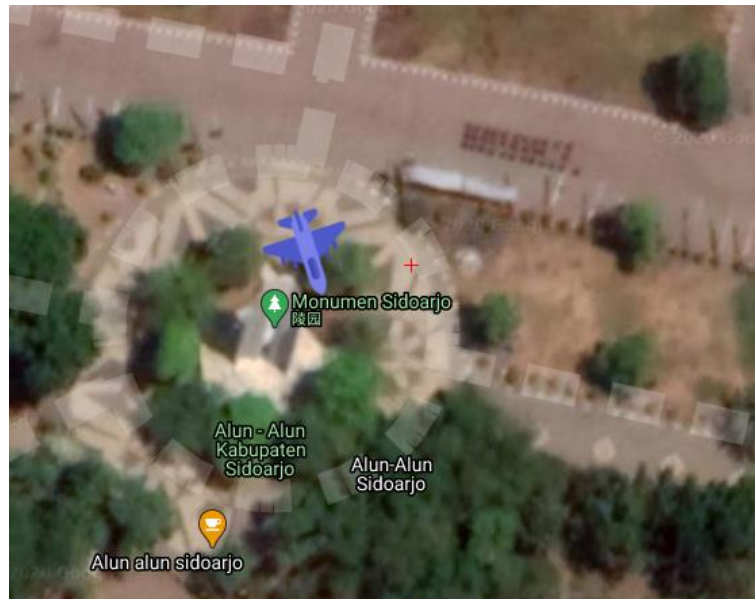
Gambar 4.21 Tampilan marker UAV pada sudut bumi 0 derajat



Gambar 4.22 Tampilan marker UAV pada sudut bumi 90 derajat



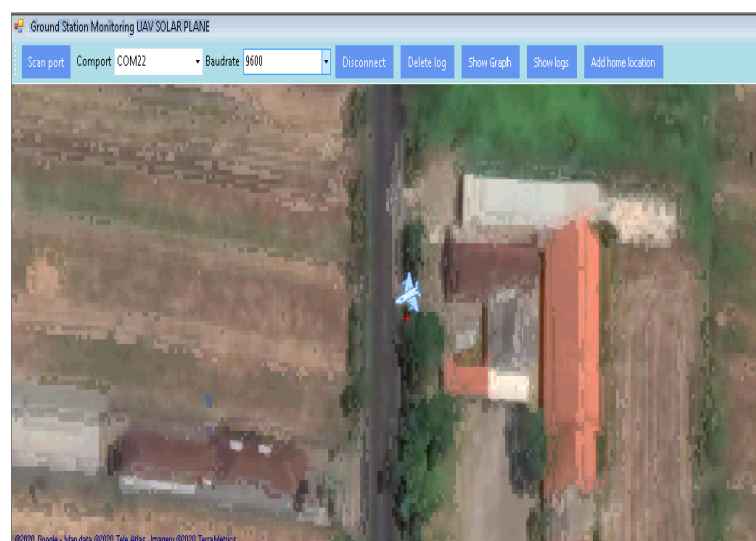
Gambar 4.23 Tampilan marker UAV pada sudut bumi 180 derajat



Gambar 4.24 Tampilan marker UAV pada koordinat -7.446048, 112.717737

Gambar 4.24 menunjukkan hasil pengujian lokasi UAV pada koordinat -7.446048, 112.717737 di lokasi alun-alun Sidoarjo. Saat GPS telah mendapatkan sinyal satelit maka, posisi UAV diletakkan sesuai dengan koordinat GPS. Pada pengujian ini penulis melakukan pengujian GPS UAV di lokasi alun-alun Sidoarjo.

Pada Gambar 4.25 menunjukkan bahwa *ground station* telah terhubung dengan pesawat UAV dengan dibuktikan lokasi pesawat UAV dapat ditampilkan pada peta.



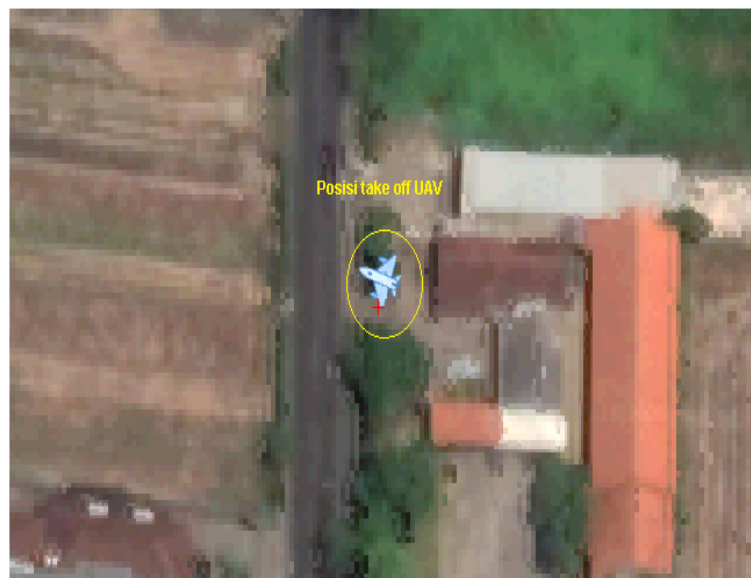
Gambar 4.25 Gound Station Monitoring

Solar UAV data	
Orientation sensor	
Compass	251 deg
Yaw	8 deg
Pitch	3 deg
Roll	12 deg
Gps sensor	
Latitude	-7.427646
Longitude	112.681352
Speed	1.20 km/h
Satelite	8 sat
Hdop	1.86
Altitude	-27.60 m

Gambar 4.26 Instrumen *Avionik* Pesawat UAV

Gambar 4.26 menjelaskan Instrumen *Avionik* Pesawat UAV dan data orientasi pesawat UAV. Pengujian *ground station* dan sekaligus pengujian terbang pesawat UAV yang terpantau melalui *ground station* di jelaskan oleh penulis pada pengujian selanjutnya. Pengujian integrasi *ground station* merupakan tahap akhir dari keseluruhan rancang bangun *monitoring* pesawat UAV *Solar Cell*. Pada pengujian ini penulis melakukan persiapan dengan menyiapkan program *ground station monitoring*.

Gambar 4.27 menunjukan posisi pesawat UAV saat melakukan *take off*. Dimana pelunuran pesawat diawali dari seberang jalan.



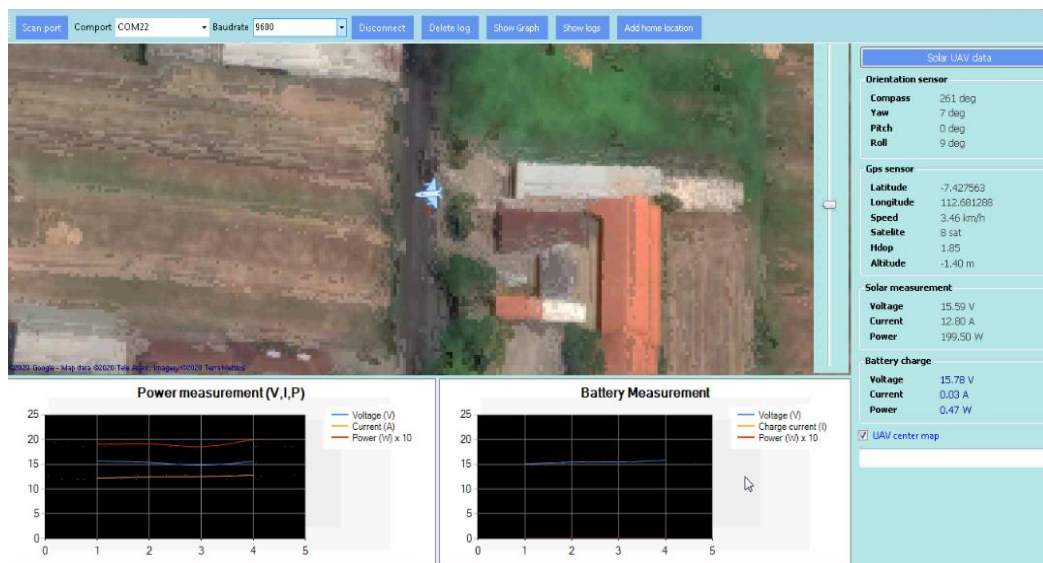
Gambar 4.27 Pesawat UAV *Take Off*



Gambar 4.28 Lokasi Terbang Pesawat UAV

Gambar 4.28 menunjukkan setelah pesawat UAV melakukan *take off*, maka pesawat UAV akan terbang ke lokasi.

Gambar 4.29 menjelaskan Tampilan *Ground Station Monitoring* secara keseluruhan. Sehingga *ground station* dapat digunakan untuk *Monitoring* posisi UAV berdasarkan koordinat *latitude* dan *longitude*, yang diperoleh dari data GPS.



Gambar 4.29 Tampilan *Ground Station Monitoring*

Menampilkan posisi UAV pada peta elektronik. *Monitoring* orientasi UAV yang terdiri dari sikap *yaw*, *pitch* dan *roll*. *Monitoring* sudut UAV terhadap sumbu bumi berdasarkan data kompas. *Monitoring* kecepatan perpindahan posisi UAV dalam satuan km/jam berdasarkan data GPS.

Monitoring throttle, yaw, pitch dan roll remote control. Pengujian dilakukan perbagian untuk memastikan bahwa setiap perangkat yang di pasang dapat berjalan dengan baik, sehingga jika ditemui permasalahan maka proses *trouble shooting* dapat dengan cepat dilakukan perbaikan. Pengujian *ground station* dan sekaligus pengujian terbang pesawat UAV yang terpantau melalui *ground station*. Pengujian integrasi *ground station* merupakan tahap akhir dari keseluruhan rancang bangun *monitoring* pesawat UAV *Solar Cell*.

BAB 5

KESIMPULAN DAN SARAN

1.1 Kesimpulan

Berdasarkan pengujian pada penelitian Rancang Bangun Sistem Posisi dan *Ground Station Monitoring* pada *Solar Cell UAV (Unmanned Aerial Vehicle)*. Penulis dapat menjelaskan kesimpulan sebagai berikut:

- a. Berdasarkan hasil pengujian sistem monitoring UAV Solar Panel dapat diketahui bahwa perencanaan system komunikasi data menggunakan perangkat nirkabel NRF24L01 jarak jangkauan maksimal 200 meter sistem monitoring posisi pesawat UAV dapat bekerja dengan baik tanpa terhalang bangunan. Hal ini dibuktikan perubahan atau perpindahan UAV dapat termonitor dan dapat ditampilkan pada peta elektronik.
- b. Faktor cuaca dan lokasi penempatan GPS mempengaruhi menerima data dari satelit yang bisa di monitoring di GCS.

1.2 Saran

Berdasarkan hasil penelitian Rancang Bangun Sistem Posisi dan *Ground Station Monitoring* pada *Unmanned Aerial Vehicle (UAV) Solar Cell* Berbasis *Global Positioning System (GPS)*, penulis menyusun saran bagi pengembangan penelitian ini agar diperoleh sistem monitoring yang lebih baik, yaitu:

- a. Mengembangkan sistem komunikasi yang dapat termonitor hingga ratusan km. Penulis menyarankan menggunakan jaringan internet dengan jaringan 3G maupun 4G. Sebagai jembatan pertukaran data dapat menggunakan *server* maupun *cloud server*.
- b. Menambahkan perangkat kamera agar melengkapi fungsi pesawat UAV sebagai pengawas.

DAFTAR PUSTAKA

Abidin, Hasanuddin Z. Geodesi Satelit. Institut Teknologi Bandung. Pradnya Paramita. Jakarta : 2001

Ariyono Setiawan, Sudrajat, Ridho Rinaldi, Didi Hariyanto, "Perancangan Sistem Kontrol Pesawat Model Tanpa Awak (*Unmanned Aerial Vehicle*) Berbasis Arduino".

Djuandi, Feri, 2011. "Pengenalan Arduino". Jakarta: Penerbit Elexmedia.

Md. Shaiful Islam, Mohammad Mukit Hasan, Mir Md. Al Kamah Tamal, Md. Jamal Mian, dan Md. Touhidur Rahman Evan, "*Detail Solidworks Design and Simulation of an Unmaned Air Vehicle*"

Scott Morton, Ruben D'Sa, dan Nikolaos Papanikolopoulos, "Solar Powered UAV: *Design and Experiments*"

Kamus Populer. Pengertian GPS Cara Kerja GPS dan Fungsi GPS. Tanggal Akses 7 Juni 2020.

Sumber: www.arduino.cc/en/Products/Counterfeit, 2014

<http://aeroengineering.co.id/2016/01/control-surface-pada-pesawat-terbang/>

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

<https://www.electroschematics.com/14423/neo-6m-gps-module/>

<http://www.vcc2gnd.com/sku/MPU-6050>

<https://lastminuteengineers.com/nrf24l01-arduino-wireless-communication/>

LAMPIRAN

Listing program *Ground Station Monitoring*

```
Public GMapOverlay polyOverlay = new GMapOverlay("polygons");
private          string[]          mapType          =          {
"ArcGIS_DarbAE_Q2_2011_NAVTQ_Eng_V5_Map","ArcGIS_Imagery_World_2
D_Map",
"ArcGIS_ShadedRelief_World_2D_Map","ArcGIS_StreetMap_World_2D_Map",

"ArcGIS_Topo_US_2D_Map","ArcGIS_World_Physical_Map","ArcGIS_World_Sh
aded_Relief_Map",

"ArcGIS_World_Street_Map","ArcGIS_World_Terrain_Base_Map","ArcGIS_Worl
d_Topo_Map",

"BingHybridMap","BingMap","BingSatelliteMap","CloudMadeMap","CzechHistory
Map",

"CzechHybridMap","CzechMap","CzechSatelliteMap","CzechTurstMap","EmptyP
rovider",

"GoogleChinaHybridMap","GoogleChinaMap","GoogleChinaSatelliteMap",

"GoogleChinaTerrainMap","GoogleHybridMap","GoogleKoreaHybridMap",

"GoogleKoreaMap","GoogleKoreaSatelliteMap","GoogleMap","GoogleSatelliteMa
p",

          "GoogleTerrainMap","LatviaMap"};

double lat,lng;
double[] location_gateway = new double[2];
private Bitmap [] bitmap = new Bitmap[6];
Boolean showNodes = false;
public Form1()
{
    InitializeComponent();
    String bmp_0 = Path.GetDirectoryName(Application.ExecutablePath) +
"/uav2.png";
    String bmp_1 = Path.GetDirectoryName(Application.ExecutablePath) +
"/home.png";
    String bmp_2 = Path.GetDirectoryName(Application.ExecutablePath) +
"/2.png";
    String bmp_3 = Path.GetDirectoryName(Application.ExecutablePath) +
"/3.png";
```

```

        String bmp_4 = Path.GetDirectoryName(Application.ExecutablePath) +
"/4.png";
        String bmp_5 = Path.GetDirectoryName(Application.ExecutablePath) +
"/5.png";
        bitmap[0] = new Bitmap(bmp_0);
        bitmap[1] = new Bitmap(bmp_1);
        bitmap[2] = new Bitmap(bmp_2);
        bitmap[3] = new Bitmap(bmp_3);
        bitmap[4] = new Bitmap(bmp_4);
        bitmap[5] = new Bitmap(bmp_5);
    }

    private Bitmap RotateImage(Bitmap bmp, float angle)
    {
        angle = angle - 90;
        if (angle < 0) angle += 360;
        if (angle > 360) angle -= 360;
        Bitmap rotatedImage = new Bitmap(bmp.Width, bmp.Height);
        rotatedImage.SetResolution(bmp.HorizontalResolution,
bmp.VerticalResolution);

        using (Graphics g = Graphics.FromImage(rotatedImage))
        {
            // Set the rotation point to the center in the matrix
            g.TranslateTransform(bmp.Width / 2, bmp.Height / 2);
            // Rotate
            g.RotateTransform(angle);
            // Restore rotation point in the matrix
            g.TranslateTransform(-bmp.Width / 2, -bmp.Height / 2);
            // Draw the image on the bitmap
            g.DrawImage(bmp, new Point(0, 0));
        }

        return rotatedImage;
    }

    private void Form1_Load(object sender, EventArgs e)
    {
        gMapControl1.MapProvider =
GMap.NET.MapProviders.GoogleChinaSatelliteMapProvider.Instance;
        GMap.NET.GMaps.Instance.Mode =
GMap.NET.AccessMode.ServerAndCache;
        gMapControl1.Position = new GMap.NET.PointLatLng(-7.453586,
112.727134);
        gMapControl1.MinZoom = 0;
        gMapControl1.MaxZoom = 24;
        gMapControl1.Zoom = 12;
        gMapControl1.DragButton = MouseButtons.Left;
        gMapControl1.Overlays.Add(nodeOverlay_1);
        gMapControl1.Overlays.Add(nodeOverlay_2);
        gMapControl1.Overlays.Add(nodeOverlay_3);
        gMapControl1.Overlays.Add(nodeOverlay_4);
        gMapControl1.Overlays.Add(nodeOverlay_5);
    }

```



```

gMapControl1.Overlays.Add(node_gateway);
gMapControl1.Overlays.Add(polyOverlay);

serial = new SerialPort();

resetDataIndex(0);
resetDataIndex(1);
resetDataIndex(2);
resetDataIndex(3);
resetDataIndex(4);

chartPower.ChartAreas[0].AxisY.Maximum = 25;
chartPower.ChartAreas[0].AxisY.Minimum = 0;
chartBatt.ChartAreas[0].AxisY.Maximum = 25;
chartBatt.ChartAreas[0].AxisY.Minimum = 0;

//chartOrientation.ChartAreas[0].AxisY.Maximum = 25;
//chartOrientation.ChartAreas[0].AxisY.Minimum = 0;
for (int i = 0; i < mapType.Length; i++)
{
    cmbMapType.Items.Add(mapType[i]);
}

}

Int16 compass;
int chart_index = 0, chart_i = 0, loc_hist=0;
double speed;

double i_test = 8.0;
double v_test = 16.0;

private void datareceive(object sender, SerialDataReceivedEventArgs e)
{
    String data = serial.ReadLine();
    data.Trim();
    data.Replace("\r", "");
    data.Replace("\n", "");
    data.Replace("\r", "");
    data.Replace("\n", "");
    data.Replace("\r\n", "");
    DateTime now = DateTime.Now;
    try {
        Invoke((MethodInvoker)delegate {
            if (data.Contains("**") || data.Contains("#") || data.Contains("?")){
                Console.WriteLine(data);
                int what = 0;
                if (data.Contains("#")) what = 0;
                if (data.Contains("**")) what = 1;
                if (data.Contains("?")) what = 2;
            }
        });
    }
}

```

```

        data = data.Replace("*", ""); data = data.Replace("#", ""); data =
data.Replace("?", "");
        String [] parse = data.Split('\t');
        if ( parse.Length == 8 ) {
            if ( what == 0 ) {
                txtLog1.AppendText(data + "\r\n");
                /*tx.data[1] = (int16_t)compass;
                tx.data[2] = (int16_t)yaw;
                tx.data[3] = (int16_t)pitch;
                tx.data[4] = (int16_t)roll;
                tx.data[5] = flat;
                tx.data[6] = flon;
                tx.data[7] = speed_;*/
                compass = Int16.Parse(parse[0]);
                lblCompass.Text = parse[0] + " deg";
                lblYaw.Text = parse[1] + " deg";
                lblPitch.Text = parse[2] + " deg";
                lblRoll.Text = parse[3] + " deg";
                lat = (double)Convert.ToInt32(parse[4]) / 1000000;
                lng = (double)Convert.ToInt32(parse[5]) / 1000000;
                lblLat.Text = string.Format("{0:0.000000}", lat);
                lblLng.Text = string.Format("{0:0.000000}", lng);
                speed = (double)Convert.ToInt32(parse[6]) / 100;
                lblSpeed.Text = string.Format("{0:0.00}", speed) + " km/h";
                if (loc_hist < 200)
                {
                    chartLocation.Series[0].Points.AddXY((lat),lng);
                    loc_hist++;
                }
                else
                {
                    loc_hist = 0;
                    chartLocation.Series[0].Points.Clear();
                }
            }
            if (chart_i < 100)
            {

```

```

chartOrientation.Series[0].Points.AddY((float)(Convert.ToInt16(parse[1])));

```

```

chartOrientation.Series[1].Points.AddY((float)(Convert.ToInt16(parse[2])));

```

```

chartOrientation.Series[2].Points.AddY((float)(Convert.ToInt16(parse[3])));

```

```

        chart_i++;
    }
    else
    {
        chart_i = 0;
        chartOrientation.Series[0].Points.Clear();
        chartOrientation.Series[1].Points.Clear();
        chartOrientation.Series[2].Points.Clear();
    }
}

```

```

    }

    } else if (what == 1) {

        txtLog2.AppendText(data + "\r\n");
        lblSat.Text = parse[0] + " sat";
        double hdop = (double)Convert.ToInt32(parse[1]) / 100;
        lblHdop.Text = string.Format("{0:0.00}", hdop) + "";
        double altitude = (double)Convert.ToInt32(parse[2]) / 100;
        lblAltitude.Text = string.Format("{0:0.00}", altitude) + " m";
        gMapControl1.Overlays.Remove(nodeOverlay_1);
        if (nodeOverlay_1.Markers.Count() > 0)
nodeOverlay_1.Clear();
        gMapControl1.Overlays.Add(nodeOverlay_1);
        GMarkerGoogle markerNode = new GMarkerGoogle(new
PointLatLng(lat, lng), RotateImage(bitmap[0], compass));
        GMap.NET.PointLatLng point = new PointLatLng(lat, lng);
        markerNode.Position = point;
        markerNode.ToolTipText = "UAV\r\n" + "GPS : " + lblLat.Text +
", " + lblLng.Text + "\r\n" +
                                "Compas : " + lblCompass.Text + "\r\nYaw : " +
lblYaw.Text + "\r\n" +
                                "Pitch : " + lblPitch.Text + ", Roll : " +
lblYaw.Text + "\r\n" +
                                "Speed : " + lblSpeed.Text + ", Satellite : " +
lblSat.Text + "\r\n";
        markerNode.ToolTipMode =
MarkerToolTipMode.OnMouseOver;
        nodeOverlay_1.Markers.Add(markerNode);

        if (chk.Checked) gMapControl1.Position = point;

        double voltage = (double)Convert.ToInt32(parse[3]) / 100;

        Random random = new Random();
        v_test = random.NextDouble() + 15.0;
        voltage = voltage - random.NextDouble();
        Random random_ = new Random();
        i_test = random.NextDouble() + 12.0;

        lblVoltage.Text = string.Format("{0:0.00}", voltage) + " V";

        double current = i_test;// (double)Convert.ToInt32(parse[4]) /
100;
        lblCurrent.Text = string.Format("{0:0.00}", current) + " A";
        double power = voltage * current;
        lblPower.Text = string.Format("{0:0.00}", power) + " W";
        double current_charge = (double)Convert.ToInt32(parse[5]) /
100;

```

```

        lblBattCurrentCharge.Text = string.Format("{0:0.00}",
current_charge) + " A";
        double batt_voltage = v_test;//
(double)Convert.ToInt32(parse[6]) / 100;
        lblBattVoltage.Text = string.Format("{0:0.00}", batt_voltage) +
" V";
        lblBatPower.Text = string.Format("{0:0.00}", batt_voltage *
current_charge) + " W";
        if (chart_index < 100)
        {
            chartPower.Series[0].Points.AddY((float)(voltage));
            chartPower.Series[1].Points.AddY((float)(current));
            chartPower.Series[2].Points.AddY((float)power/10);
            chartYawRate.Series[0].Points.AddY((float)speed);

            chartBatt.Series[0].Points.AddY((float)(batt_voltage));
            chartBatt.Series[1].Points.AddY((float)(current_charge));
            chartBatt.Series[2].Points.AddY((float)(batt_voltage *
current_charge) / 10);
            chartAltitude.Series[0].Points.AddY((float)(altitude));
            chart_index++;
        }
        else
        {
            chartBatt.Series[0].Points.Clear();
            chartBatt.Series[1].Points.Clear();
            chartBatt.Series[2].Points.Clear();
            chartPower.Series[0].Points.Clear();
            chartPower.Series[1].Points.Clear();
            chartPower.Series[2].Points.Clear();
            chartAltitude.Series[0].Points.Clear();
            chartYawRate.Series[0].Points.Clear();
            chart_index = 0;
        }

        tx.data[1] = satelites;
        tx.data[2] = hdop_;
        tx.data[3] = altitude_;
        tx.data[4] = solar_voltage; //v source
        tx.data[5] = current; //i source
        tx.data[6] = current_charge;
        tx.data[7] = bat_voltage; */

    } else if (what == 2) {
        txtLog3.AppendText(data + "\r\n");
        Int16 y = Convert.ToInt16(parse[0]);

```

LAMPIRAN - 2 Listing *Program Arduino Power Management*

```
void ANALOG_READ () {
    solar_voltage = ((float)analogRead ( PIN_VSOURCE )/1023.0f)*5.0f *
(11.58f/1.6275f) * 100;
    current = (((float)analogRead ( PIN_ISOURCE )/1023.0f) * 5.0f ) *
(184.0f/3.3f) * 100;
    int16_t vbat = ((float)analogRead ( PIN_VBATT )/1023.0f) * 5100;
    vbat = map ( vbat, 0, 4300, 0, 1720 );
    if ( vbat < 0 ) vbat = 0;
    if ( counter < 1000 ) {
        avg_offset += analogRead ( PIN_ICHARGE );
        counter++;
    } else {
        double avg_curr_offset = avg_offset /1000;
        float v = (((float)(analogRead ( PIN_ICHARGE )- avg_curr_offset) /
1023.0f) * 5.0f );
        if ( v < 0 ) v = 0;
        if ( average < 10 ) {
            avg_current_charging += ( v / SENSISITIVITY ) * 100;
            avg_voltage_charging += ( vbat /1 );
            average++;
        } else {
            current_charge = avg_current_charging/10;
            bat_voltage = avg_voltage_charging/10;
            avg_current_charging = 0;
            avg_voltage_charging = 0;
            average = 0;
        }
    }
    /*if ( millis() - current_timer >= 100 ) {
        Serial.print(bat_voltage);
        Serial.print("\t");
    }
```

```

        Serial.println( current_charge );
        current_timer = millis();
    }*/
}
//delay (5);
}

//*****

#define PIN_BATT_CHARGE 8//PWM
#define PIN_BATT_USE 9//DIGITAL
#define PIN_CELL_USE 10//DIGITAL
int PWM_CHARGE = 0;
int16_t PEM_CHARGE_MAX = 0;
int16_t PEM_CHARGE_MIN = 255;
#include <PID_v1.h>
double Kpc = 0.5, Kic = 0.75, Kdc = 0.0040;
double Setpoint, Input, Output;
PID PID_CHARGE(&Input, &Output, &Setpoint, Kpc, Kic, Kdc, DIRECT);
float pid_charge_error;
float pid_charge_value;
float CP, CI, CD;
float previous_error_pid_C, previous_I_C;

//*****

boolean SOLAR_PWR = false;
#define CURRENT_CHARGE 200 //2A
void MANAGE_POWER () {
    if ( solar_voltage >= 1700 ) {
        USE_BATTERY_OFF ();
        SOLAR_PWR = true;
    }
    if ( bat_voltage < 1680 ) {
        pid_charge_error = (CURRENT_CHARGE) - current_charge;
        CP = pid_charge_error;
    }
}

```

```

    CI = CI + previous_I_C;
    CD = pid_charge_error-previous_error_pid_C;
    pid_charge_value = ((Kpc)*CP) + ((Kic)*CI) + ((Kdc)*CD);
    previous_I_C= CI;
    previous_error_pid_C = pid_charge_error;
    pid_charge_value = constrain(pid_charge_value,0,255);
    analogWrite ( PIN_BATT_CHARGE, 255-pid_charge_value );
} else {
    previous_I_C = 0;
    previous_error_pid_C = 0;
    analogWrite ( PIN_BATT_CHARGE, 255 );
    CI = 0;
    CD = 0;
}

    LED_USE_SOLAR_PANEL();
} else {
    LED_BATERAY_USE_BLINK ();
    USE_BATTERAY_ON ();
}
}

void USE_BATTERAY_ON () {
    digitalWrite ( PIN_BATT_USE, LOW );
}

void USE_BATTERAY_OFF () {
    digitalWrite ( PIN_BATT_USE, HIGH );
}

boolean USE_BATERAY = false;
#define BUZZER_PIN 29
#define LED_PIN 31
#define LEDSOLAR_PIN 30

void LED_USE_SOLAR_PANEL (){

```

```

    if ( millis() - led_timer_ >= 500 ) {
        digitalWrite ( LED_PIN, led_state_batt );
        led_state_batt = !led_state_batt;
        led_timer_ = millis();
    }
}

void LED_BATERAY_USE_BLINK () {
    if ( millis() - led_timer_ >= 100 ) {
        digitalWrite ( LED_PIN, led_state_batt );
        led_state_batt = !led_state_batt;
        led_timer_ = millis();
    }
}

void LED_BATERAY_USE_OFF () {
    digitalWrite ( LED_PIN, HIGH );
}

void LED_SOLAR_USE_BLINK () {
    if ( millis() - led_timer_ >= 500 ) {
        digitalWrite ( LED_PIN, led_state_batt );
        led_state_batt = !led_state_batt;
        led_timer_ = millis();
    }
}

void LED_SOLAR_USE_OFF () {
    digitalWrite ( LEDSOLAR_PIN, HIGH );
}

void loop() {
    ANALOG_READ ();
    MANAGE_POWER ();
}

```


LAMPIRAN - 1 Listing *Program Ground Station Power Monitoring*

```
lblVoltage.Text = string.Format("{0:0.00}", voltage) + " V";
    double current = i_test;
lblCurrent.Text = string.Format("{0:0.00}", current) + " A";
    double power = voltage * current;
    lblPower.Text = string.Format("{0:0.00}", power) + "
W";
        double current_charge =
(double)Convert.ToInt32(parse[5]) / 100;
    lblBattCurrentCharge.Text = string.Format("{0:0.00}",
current_charge) + " A";
        double batt_voltage = v_test;//
(double)Convert.ToInt32(parse[6]) / 100;
    lblBattVoltage.Text = string.Format("{0:0.00}",
batt_voltage) + " V";
    lblBatPower.Text = string.Format("{0:0.00}",
batt_voltage * current_charge) + " W";
    if (chart_index < 100)
    {
        chartPower.Series[0].Points.AddY((float)(voltage));
        chartPower.Series[1].Points.AddY((float)(current));
        chartPower.Series[2].Points.AddY((float)power/10);
        chartYawRate.Series[0].Points.AddY((float)speed);

chartBatt.Series[0].Points.AddY((float)(batt_voltage));

chartBatt.Series[1].Points.AddY((float)(current_charge));
        chartBatt.Series[2].Points.AddY((float)(batt_voltage
* current_charge) / 10);
        chartAltitude.Series[0].Points.AddY((float)(altitude));
        chart_index++;
    }
    else
    {
        chartBatt.Series[0].Points.Clear();
        chartBatt.Series[1].Points.Clear();
        chartBatt.Series[2].Points.Clear();
        chartPower.Series[0].Points.Clear();
        chartPower.Series[1].Points.Clear();
        chartPower.Series[2].Points.Clear();
        chartAltitude.Series[0].Points.Clear();
        chartYawRate.Series[0].Points.Clear();
        chart_index = 0;
    }
}
```

```

/*
tx.data[0] = DATA_STATE;
tx.data[1] = satellites;
tx.data[2] = hdop_;
tx.data[3] = altitude_;
tx.data[4] = solar_voltage; //v source
tx.data[5] = current; //i source
tx.data[6] = current_charge;
tx.data[7] = bat_voltage; */

} else if (what == 2) {
    txtLog3.AppendText(data + "\r\n");
    Int16 y = Convert.ToInt16(parse[0]);
    Int16 p = Convert.ToInt16(parse[1]);
    Int16 r = Convert.ToInt16(parse[2]);
    Int16 t = Convert.ToInt16(parse[3]);
    //try
    //{
        /*pbThrottle.Value = t - 1000;
        pbPitch.Value = p - 1000;
        pbRoll.Value = r - 1000;
        pbYaw.Value = y - 1000;*/
    /*}
    catch (Exception ex)
    {

    }*/

}

/*int dataIndex = 1;
if (dataGridViewClient.RowCount > 0)
{
    dataGridViewClient.Rows.Clear();
    dataIndex = 1;
}

```

BIOGRAFI



Nama : Dwi Heriyanto

Pangkat/Korps/NRP : Serka NAV NRP 110115

Alamat : Lingkungan Booto RT
001 /RW 001 Kota
Mojokerto Jawa Timur

Merupakan anak kedua dari tiga bersaudara pasangan Bapak Marli (ALM) dan Ibu Nasri. Menempuh pendidikan umum SD,SLTP di Kota Gresik dan SMA di Kota Sidoarjo jawa timur. Menikah dengan Nani Masita Inayati Pada tanggal 18 Maret 2012 dan dikaruniai anak bernama Loya Zilgar Maheriyanto dan Sultan Zulkhair Maheriyanto. Mengikuti pendidikan Militer DikCaba PK XXVI 2006 di Surabaya. Dinas pertama ditempatkan di KRI SIRIBUA 859 (Satkamla) Lantamal I Belawan, hingga tahun 2017 Desember melaksanakan tugas belajar di Sekolah Tinggi Teknologi Angkatan Laut (STTAL) Surabaya, Program Studi D3 Tekhik Elektronika Angkatan ke XIII.