

BAB II

TINJAUAN PUSTAKA

Penelitian tentang aplikasi monitoring muatan balon atmosfer sudah banyak dilakukan. Ardhi [2015] melakukan penelitian tentang “*Rancang Bangun Antarmuka Penyimpanan Profil Atmosfer Transmisi Radiosonde Berbasis LabVIEW*”. Pada penelitiannya ini menjelaskan bahwa *data logger* sebagai perekam data pengamatan secara *realtime*. *LabVIEW* yang dilengkapi simulasi membantu perancang untuk melihat respon berbagai kombinasi konstanta dengan variasi input yang berbeda. Penggunaan *LabVIEW* ini sangat membantu perancang dalam perancangan antarmuka dan data logger yang baik, handal dan sederhana. Proyek akhir ini bertujuan untuk membuat dan mengimplementasikan suatu sistem yang dapat menampilkan antarmuka dan menyimpan data profil atmosfer dengan perangkat lunak *LabVIEW*. Metode penelitian yang digunakan pada penelitian ini meliputi studi literatur, perancangan perangkat lunak, dan pengujian. *LabVIEW* yang digunakan pada perancangan ini adalah *LabVIEW Student Version*. Dalam penelitian ini telah berhasil dibuat sebuah antarmuka untuk menampilkan data profil atmosfer dengan data yang diperoleh dari transmisi radiosonde secara *realtime*. Selain menampilkan parameter yang diperoleh dari transmisi *radiosonde*, antarmuka ini juga dapat melakukan operasi perhitungan untuk mencari nilai *bearing degree* dan ketinggian berdasarkan tekanan udara yang kemudian ditampilkan dan disimpan pada *database*. [1]

Wiyagi, Danar, dan Tri [2016] melakukan penelitian tentang “*High Altitude Balloon Payload Design for Atmospheric Observation*”. Pada penelitiannya ini menjelaskan bahwa Teknik Elektro Universitas Muhammadiyah Yogyakarta telah merancang dan mengembangkan muatan untuk *High Altitude Balon*. Payload ini memiliki fungsi yang mengukur parameter atmosfer secara vertikal dan mengirimkan data ke *ground station* di bumi dengan menggunakan telemetri. Parameter yang diukur adalah tekanan udara, suhu dan kelembaban relatif. Selain mengirimkan parameter ini, payload juga memiliki misi memantau posisi muatan sebenarnya dengan menggunakan GPS, transmisi video real time, pengambilan data secara reguler dan pengiriman gambar dari muatan. Payload ini diuji pada Kompetisi Balon Tontonan Atmosfer tahunan (KOMBAT) 2016. [2]

Danar [2016] melakukan penelitian tentang “*Teknik Pengiriman Citra pada Muatan Balon Atmosfer*”. Pada penelitiannya ini menjelaskan bahwa Atmosfer adalah lapisan campuran gas yang membungkus permukaan bumi dan berfungsi sebagai pelindung kehidupan makhluk hidup di bumi. Semakin banyaknya kandungan CO₂ di bumi meningkatkan suhu atmosfer permukaan bumi. Berdasarkan permasalahan di atas dibuatlah prototipe muatan balon atmosfer sebagai alat pengukur untuk mengetahui profil atmosfer permukaan bumi secara horizontal dan vertikal yang dilengkapi dengan pengiriman citra yang nantinya setiap data yang berupa parameter suhu, kelembaban udara, tekanan udara, ketinggian, garis lintang, garis bujur, dan gambar akan dikirimkan ke *ground station* secara serial Penelitian dilakukan dengan

menggunakan beberapa metode pengujian sesuai dengan tipe sensor yang digunakan. Semua data yang diperoleh dari muatan balon atmosfer kemudian di analisis sehingga diketahui persentase error dari sensor yang digunakan serta untuk mengetahui keandalan sistem yang dibuat. [3]

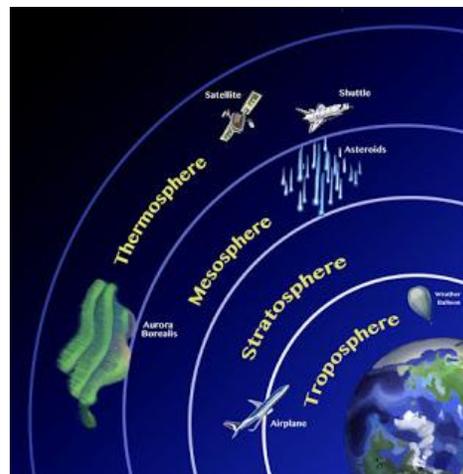
Horan [2017] melakukan penelitian tentang “*Using Labview to Design a Payload Control System*”. Pada penelitiannya ini menjelaskan bahwa mengembangkan sebuah paket perangkat lunak dengan kombinasi antara GS dan *flight control* menggunakan LabVIEW. Sistem komputer ini digunakan untuk memperoleh data dari sensor, mengendalikan jalur komunikasi, menyediakan kemampuan akuisisi data otomatis, dan menyediakan antarmuka pengguna. [4]

Pengujian software secara umum dan *Graphical User Interface* (GUI) khususnya adalah salah satu tantangan utama dalam siklus hidup dari setiap sistem perangkat lunak. Pengujian GUI secara *inheren* lebih sulit daripada pengujian antarmuka tradisional (*remote control*) dan *command-line*. Beberapa faktor yang membuat pengujian GUI berbeda dari pengujian perangkat lunak tradisional dan secara signifikan lebih sulit adalah: sejumlah besar objek, tampilan yang berbeda dan nuansa objek, banyak parameter yang terkait dengan setiap objek, pengungkapan progresif, input yang kompleks dari berbagai sumber, dan output grafis. Teknik – teknik pengujian yang sudah ada perlu disesuaikan / ditingkatkan untuk GUI, dan teknik pengujian baru yang diinginkan bertujuan untuk menciptakan sesuatu yang lebih efisien dan efektif (Zafar Singhera, Ellis Horowitz dan Abad Shah, 2009).

2.1. Atmosfer

Bumi memiliki lapisan gas yang disebut atmosfer yang berada pada ketinggian 0 km hingga 1000 km dari atas permukaan laut. Atmosfer memiliki susunan lapisan yang dapat dibedakan berdasarkan karakteristik seperti komposisi gas, suhu, dan tekanan. Perpindahan antara lapisan yang satu dengan yang lain berlangsung secara bertahap (Sasmito,1998).

Lapisan atmosfer terbagi menjadi empat lapisan seperti tampilan gambar 2.1 yaitu troposfer (0-18 kmdpl), stratosfer (18-49 kmdpl), mesosfer (49-82 kmdpl), termosfer (82-800 kmdpl), dan eksosfer (800-1000 kmdpl). Atmosfer mengandung beberapa campuran gas-gas yang menutupi seluruh permukaan bumi. Atmosfer bumi tersusun dari nitrogen (78,17%), oksigen (20,97%), dengan sedikit argon (0,9%), karbondioksida (0,0357%), uap air, dan gas lainnya.



Gambar 2.1. Lapisan Atmosfer Bumi

(Sumber : <http://www.edubio.info/2017/03/lapisan-lapisan-atmosfer.html>)

Perubahan musim yang tidak menentu akhir-akhir ini berpengaruh terhadap kandungan uap air yang tersimpan pada atmosfer. Ketika di Indonesia terjadi musim kemarau maka kandungan uap air yang ada di atmosfer akan menipis, sebaliknya apabila di Indonesia sedang terjadi musim penghujan maka kandungan uap air di atmosfer meningkat

Sebagai media lingkungan, atmosfer berfungsi untuk menampung berbagai macam gas yang dihasilkan oleh aktivitas manusia seperti Oksigen, Karbon dioksida, dan uap air. Keberadaan berbagai macam gas tersebut apabila sesuai kadar maka tidak akan berpengaruh banyak terhadap aktivitas manusia namun sebaliknya apabila keberadaan gas-gas tersebut melebihi ukuran yang seharusnya maka dikhawatirkan dapat membahayakan umat manusia dan kehidupan di bumi.

2.2. High Altitude Balloon

High Altitude Balloon (HAB) adalah sebuah balon besar tak berawak biasanya terisi dengan helium atau hidrogen yang dilepaskan ke stratosfer. Balon ini mampu membawa sekelompok objek hingga ketinggian 100.000 *feet* dari permukaan bumi. Objek yang biasa dibawa oleh HAB disebut *payload*. Tujuan dari *payload* berbeda-beda seperti pengambilan data citra maupun parameter-parameter atmosfer. *Payload* harus mencakup pemancar radio, sistem navigasi, dan komunikasi *tracker*.



Gambar 2.2. *High Altitude balloon*

HAB seperti gambar 2.2 sangat berguna pada bidang penelitian. Hal tersebut terbukti dengan yang dilakukan oleh peneliti di Amerika. National Weather Service (NWS) menerbangkan HAB dua kali dalam sehari untuk mengukur karakteristik atmosfer guna membantu model perkiraan cuaca. Jika terjadi cuaca ekstrem, mereka akan meluncurkan HAB tiga kali dalam sehari baik di stasiun cuaca maupun di bandara seluruh negeri. HAB ini mengagumkan karena memungkinkan akses cepat dalam mengetahui kondisi cuaca dan parameter atmosfer dengan cepat.

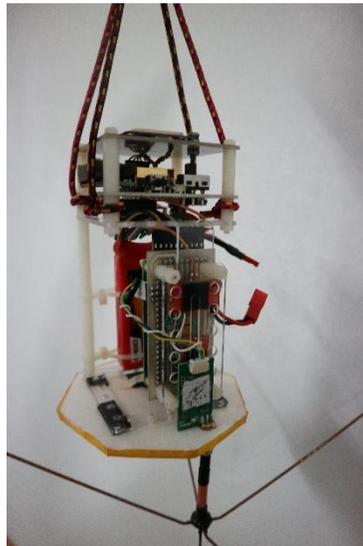
2.3. Muatan Balon Atmosfer (*Payload*)

Payload adalah Muatan balon atmosfer merupakan sebuah alat yang diterbangkan dengan wahana balon atmosfer. Muatan balon atmosfer dapat mengukur parameter-parameter atmosfer vertikal dan mengirimkan datanya melalui *radio telemetry* kepada sistem penerima di permukaan bumi. *Payload* pada muatan Mr. Cilindro terbuat dari *polyfoam*. *Polyfoam* adalah jenis foam yang keras dan kuat. Struktur yang keras ini memberikan ketahanan muatan terhadap benturan yang mungkin terjadi saat kompetisi. Sifat *foam* yang merupakan penahan suhu yang baik memberikan kestabilan temperatur pada rangkaian elektronik dalam muatan. Bobot *polyfoam* yang sangat ringan menjadi poin lebih dalam pemilihan bahan ini. Tampilan *payload* Mr. Cilindro seperti gambar 2.3.



Gambar 2.3. *Payload* Mr, Cilindro

Model *payload* (Gambar 2.3) adalah struktur silinder dengan dimensi 9 x 13 cm. Ketebalan *polyfoam* pada sisi atas dan bawah adalah 2 mm. Sedangkan pada sisi selimut memiliki ketebalan 2 mm. *Polyfoam* dibungkus dengan isolasi berwarna kuning serta bentuk ini memungkinkan *payload* mampu berputar secara bebas. Selain bentuk *cover*, *payload* juga mempunyai komponen penyusun elektronik. Komponen elektronik inilah yang membuat *payload* dapat mengukur parameter-parameter atmosfer. Gambar 2.4. adalah susunan komponen elektronik yang ada pada *payload*:

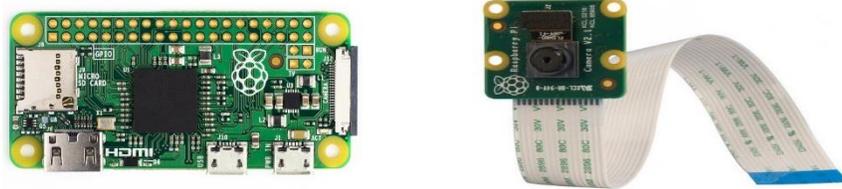


Gambar 2.4. Rangkaian Elektronik Mr. Cilindro

1.1.1. Bagian Pengolah Data

Raspberry Pi Zero menggunakan prosesor ARM 700MHz dan sistem operasi Linux. Dengan spesifikasi tinggi tersebut diharapkan proses pengolahan gambar baik *capturing*, konversi, dan penggabungan dengan data sensor dapat dilakukan dengan cepat.

Untuk capture foto menggunakan kamera RaspiCam v.2. RaspiCam v.2 adalah kamera yang didesain khusus untuk digunakan pada SBC Raspberry Pi. Kamera ini memiliki sensor Sony IMX219 dengan resolusi hingga 8 MP dengan fokus tetap. Dengan ukuran relatif kecil (25 x 24 mm) serta ringan .



Gambar 2.5. Raspberry Pi Zero dan RaspiCam v2

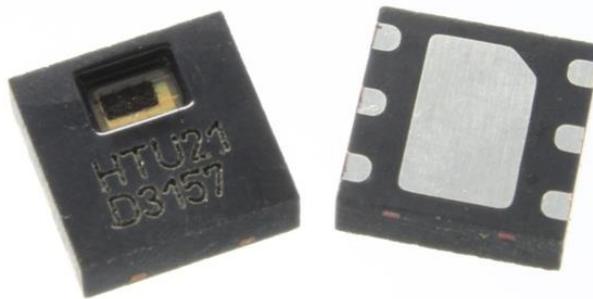
(Sumber : <https://www.raspberrypi.org/products/raspberry-pi-zero/>)

1.1.2. *HTU21D Humidity Sensor*

HTU21D adalah sensor kelembaban relatif digital dengan performa tinggi, low cost dan mudah digunakan. Sensor ini ideal untuk penginderaan lingkungan, data logging dan sempurna untuk stasiun cuaca. Protokol pembacaan data menggunakan I²C yang hanya membutuhkan dua jalur data saja. Berikut adalah spesifikasi sensor HTU21D.

Tabel 2.1. Spesifikasi *HTU21D Humidity Sensor*

Ratings	Symbol	Value	Unit
Storage Temperature	T_{stg}	-40 to 125	°C
Supply Voltage (Peak)	V_{cc}	3.8V	V_{dc}
Humidity Operating Range	RH	0 to 100	%RH
Temperature Operating Range	T_a	-40 to +125	°C
VDD to GND		-0.3 to 3.6V	V
Digital I/O pins (DATA/SCK) to VDD		-0.3 to VDD+0.3	V
Input current on any pin		-10 to +10	mA



Gambar 2.6. *HTU21D Humidity Sensor*

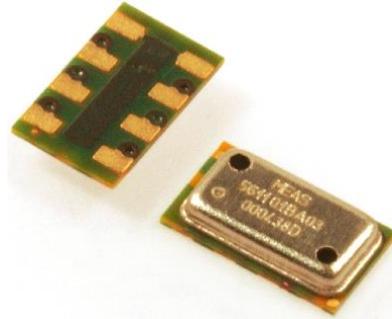
(Sumber : <https://www.mouser.com/new/measurement-specialties/measurement-specialties-HTU21D/>)

1.1.3. *MS5611 Pressure Sensor*

MS5611 adalah generasi baru dari sensor altimeter resolusi tinggi. sensor ini digunakan untuk altimeter dan variometers dengan resolusi ketinggian 10 cm. Modul sensor termasuk sensor tekanan linearitas yang tinggi dan ADC 24 bit. Sensor ini menghasilkan nilai tekanan dan suhu yang tepat. Berikut adalah spesifikasi sensor MS5611.

Tabel 2.2. Sepesifikasi *MS5611 Pressure Sensor*

Sensor Performances ($V_{DD} = 3\text{ V}$)				
Pressure	Min	Typ	Max	Unit
Range	10		1200	mbar
ADC	24			bit
Resolution (1)	0.065 / 0.042 / 0.027 / 0.018 / 0.012			mbar
Accuracy 25°C, 750 mbar	-1.5		+1.5	mbar
Error band, -20°C to +85°C 450 to 1100 mbar (2)	-2.5		+2.5	mbar
Response time (1)	0.5 / 1.1 / 2.1 / 4.1 / 8.22			ms
Long term stability		±1		mbar/yr



Gambar 2.7. *MS5611 Pressure Sensor*

(Sumber : [http://www.te.com/usa-en/product-CAT BLPS0036.html](http://www.te.com/usa-en/product-CAT%20BLPS0036.html))

1.1.4. *DS18S20 Temperature Sensor*

DS18S20 adalah sensor temperatur terkalibrasi yang dapat mengukur suhu mulai -55°C to $+125^{\circ}\text{C}$ (-67°F to $+257^{\circ}\text{F}$). Dengan 9-Bit Resolusi dan tidak perlu menambahkan komponen eksternal yang lain. Sehingga jika muatan berada dalam kondisi yang ekstrim maka sensor temperatur masih dapat berfungsi dengan baik.



Gambar 2.8. *DS18S20 Temperature Sensor*

(Sumber : <https://www.sparkfun.com/products/retired/8366>)

1.1.5. *GPS U BLOX M8N*

GPS U BLOX M8N adalah sensor yang dapat memberitahu koordinat posisi muatan di udara. Sensor ini memberikan performa terbaik dan integrasi RF. Selain itu, sensor ini memiliki kinerja tinggi

dalam konsumsi daya yang rendah. Berikut adalah spesifikasi sensor GPS U BLOX M8N.

Highlights

- Concurrent reception of up to 3 GNSS (GPS, Galileo, GLONASS, BeiDou)
- Industry leading -167 dBm navigation sensitivity
- Security and integrity protection
- Supports all satellite augmentation systems
- Advanced jamming and spoofing detection
- Product variants to meet performance and cost requirements
- Backward compatible with NEO-7 and NEO-6 families



Gambar 2.9. GPS U BLOX M8N

(Sumber : <https://www.u-blox.com/en/product/neo-m8-series>)

1.1.6. Power Supply

Baterai yang digunakan dalam muatan ini adalah baterai dengan kapasitas 1000 mAh 2 cell 7,4V dengan jenis Lithium-polymer. Memiliki dimensi yang kecil, bermassa ringan dan memiliki kepadatan energi yang tinggi. Selain itu memiliki masa simpan yang panjang, sehingga diharapkan kebutuhan energi muatan dapat tercukupi.

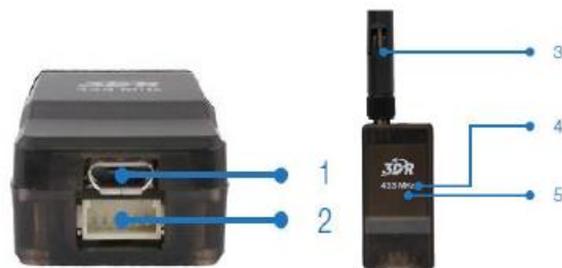


Gambar 2.10. Baterai Lipo Turnigi 1000mAh 2s

(Sumber : <http://buaya-instrument.com/turnigy-1000mah-2s.html>)

1.1.7. Bagian *Telemetry* Radio

Radio Telemetry 3DR adalah radio telemetri UHF (433MHz-438MHz) yang memiliki power output hingga 100mW. Radio ini berbasis module Hope RF HM-TRP. Dengan koreksi bit error hingga 25% dan didukung teknologi *Frequency Hopping Spread Spectrum* maka radio ini memiliki performa yang sangat baik untuk aplikasi balon atmosfer. Protokol komunikasi data menggunakan model UART (Serial Data) sehingga mudah untuk digunakan.



Gambar 2.11. *Radio Telemetry 3DR*

(Sumber : <http://ardupilot.org/copter/docs/common-siktelemetry-radio.html>)

Tabel 2.3. Keterangan Gambar 2.11

No	Descriptions
1	Micro-USB port
2	6-wire cable connector
3	Antenna
4	Frequency
5	LED indicator

2.4. Aplikasi

Secara umum pengertian aplikasi menurut kamus besar bahasa Indonesia (1998:52) adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu. Supriyanto (2005:117-118) mendefinisikan *software* aplikasi sebagai sebuah perangkat lunak program yang mempunyai aktivitas mengolah perintah yang diperlukan dalam melaksanakan permintaan *user* dengan tujuan tertentu. Aplikasi (*application*) adalah software yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas-tugas tertentu, misalnya *LabVIEW*, *Foxview*, *Logixpro*, dan *Android Studio*. Perangkat lunak aplikasi terdiri dari beberapa sistem antara lain:

1. *Programming language* (bahasa pemrograman).
2. *Packet program* (program paket).
3. *Application program* (program aplikasi).
4. *Utility program* (program utilitas).
5. *Entertainment, games*, dan lain-lain

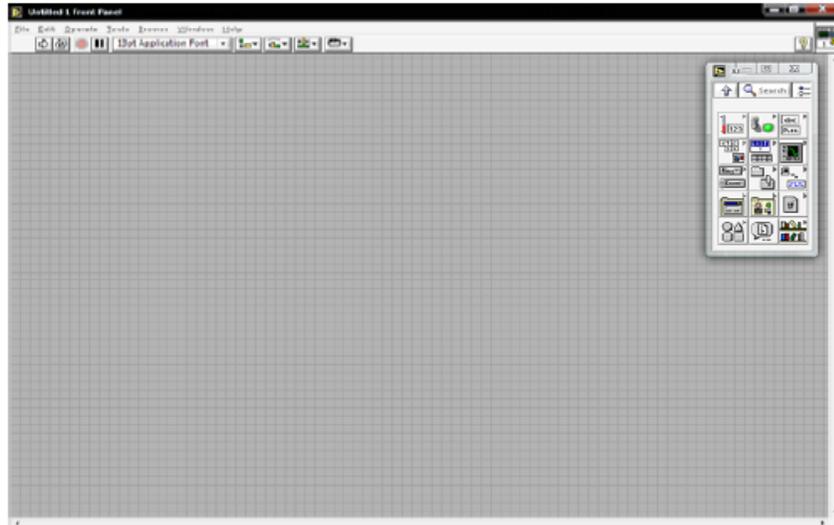
Aplikasi yang akan dibahas pada tugas akhir ini adalah *application program* (program aplikasi). Program aplikasi yang digunakan yaitu *LabVIEW*. *LabVIEW* digunakan untuk membangun aplikasi dari muatan Mr. Cilindro dalam bentuk GUI (*Graphical User Interface*). Berikut penjelasan tentang *LabVIEW* dan GUI (*Graphical User Interface*):

2.4.1. *LabVIEW*

LabVIEW merupakan sebuah perangkat lunak pemrograman yang dibuat oleh *National Instruments* dengan konsep yang lebih sederhana. *LabVIEW* mempunyai fungsi dan peranan yang sama dengan aplikasi pemrograman lainnya seperti C++, matlab, atau *visual basic*. *LabVIEW* menggunakan Bahasa pemrograman berbasis grafis atau blok diagram berbeda dengan aplikasi pemrograman lainnya menggunakan *text*. *LabVIEW* dikenal dengan *Virtual Instruments* karena penampilannya dan operasinya dapat meniru sebuah *instrument*. Pada *LabVIEW*, *user* pertama-tama membuat *user interface* dengan menggunakan fitur control dan indikator. Kontrol adalah *knobs*, *push buttons*, *dials* dan peralatan inputan lainnya sedangkan indikator adalah *graphs*, *LED*, dan peralatan *display* lainnya. Setelah menyusun *front panel*, *user* selanjutnya menyusun blok diagram yang berisi kode-kode untuk mengontrol *front panel*. Berikut komponen-komponen utama dari perangkat lunak *LabVIEW* yaitu:

1. *Front Panel*

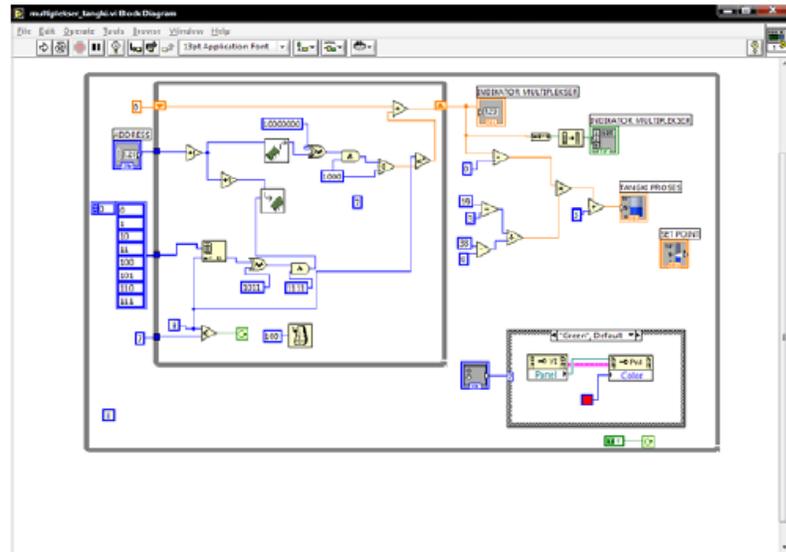
Front Panel merupakan bagian *user interface* yang berfungsi sebagai tempat membuat sebuah VI (*Virtual Instrument*) yang mengandung fitur *control* dan *indicator*. Tampilan dari *front panel* dapat dilihat pada Gambar 2.12.



Gambar 2.12. *Front Panel*

2. Blok Diagram dari VI (*Virtual Instrument*)

Blok diagram merupakan fitur yang berfungsi sebagai tempat *source code*. *Source code* berfungsi sebagai intruksi untuk *front panel*. Tampilan dari blok diagram dari *VI (Virtual Instrument)* dapat dilihat pada Gambar 2.13.



Gambar 2.13. Blok Diagram dari VI (*Virtual Instrument*)

3. Control Palette

Control palette adalah wadah untuk beberapa *control* dan *indicator* pada *front panel*. *Control palette* hanya terdapat pada *front panel*. Menampilkan *control palette* dapat dilakukan dengan klik *windows* kemudian klik *show control palette*. Tampilan dari *control palette* dapat dilihat pada gambar 2.14.



Gambar 2.14. Control Palette

4. *Functions Pallete*

Functions pallete mempunyai fungsi sebagai fitur untuk membangun sebuah blok diagram. *Functions pallete* hanya terdapat pada blok diagram. Dalam menampilkan *functions pallete* dapat dilakukan dengan klik *windows* kemudian *show functions pallete*. Tampilan dari *functions pallete* dapat dilihat pada gambar 2.15.

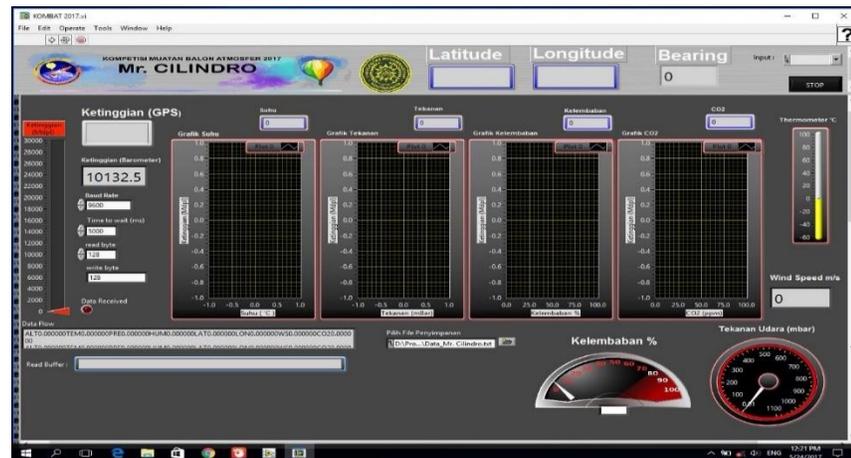


Gambar 2.15. *Functions Pallete*

2.4.2. GUI (*Graphical User Interface*)

GUI (*Graphical User Interface*) adalah antarmuka pada sistem operasi atau komputer yang menggunakan menu grafis agar mempermudah para pengguna-nya untuk berinteraksi dengan komputer atau sistem operasi. Graphical User Interface (GUI) adalah bagian penting dari setiap aplikasi perangkat lunak bagi pengguna pada saat ini dan GUI juga harus memenuhi kriteria

berikut : *significant design, development, dan testing activities* (Zafar Singhera, Ellis Horowitz dan Abad Shah, 2009). Dalam hal ini dengan memanfaatkan GUI (*Graphical User Interface*) program akan lebih jauh lebih menarik. Selain itu program akan memiliki tampilan lebih efektif dan interaktif (kamsyakawuni, 2010:35). Salah satu contoh tampilan dari GUI (*Graphical User Interface*) dapat dilihat pada Gambar 2.16.



Gambar 2.16. GUI (*Graphical User Interface*) Mr. Cilindro 2016

2.5. Metode Pengembangan Perangkat Lunak

Metodologi pengembangan perangkat lunak atau paradigma rekayasa perangkat lunak adalah strategi yang dilakukan dalam hal memadukan proses, metode, dan *tools*. Metode-metode rekayasa perangkat lunak menawarkan teknik untuk membangun sebuah perangkat lunak. Metode-metode perangkat lunak berkaitan

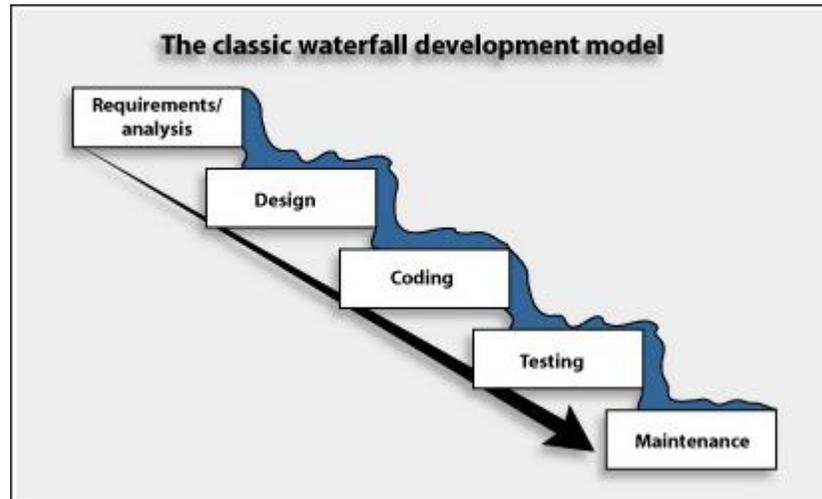
dengan serangkaian tugas yang luas menyangkut analisis kebutuhan, konstruksi program, desain, pengujian, dan pemeliharaan (Rogers S. Pressman, 2002).

Dalam hal menyelesaikan permasalahan di dalam pengembangan perangkat lunak, hal yang harus dilakukan perancang adalah menggabungkan strategi pengembangan yang melingkupi lapisan proses, metode, dan alat bantu. Model proses perancang perangkat lunak dapat disesuaikan berdasarkan sifat aplikasi dan proyeknya, metode, *control*, dan alat bantu yang dipakai.

Berikut metode-metode pengembangan perangkat lunak yaitu metode *waterfall*, *prototype*, RAD dan model Sekuensial Linier. Metode-metode tersebut akan menjadi acuan dalam penelitian ini. Berikut metode-metode tersebut :

2.5.1. Metode *Waterfall*

Model *waterfall* merupakan salah satu model klasik bersifat sistematis. Model ini dikerjakan secara berurutan. Penggunaan model ini dalam penerapan kehidupan sehari-hari sangatlah memakan waktu dan sangat sedikit di pakai dalam membuat software.



Gambar 2.17. Metode *Waterfall*

(Sumber: <https://www.academia.edu/13871195/>)

Kelebihan metode *waterfall*:

- Mudah diaplikasikan.
- Memberikan *template* tentang metode analisis, desain, kode, pengujian, serta pemeliharaan.
- Sangat sesuai digunakan untuk peroduk perangkat lunak yang sudah jelas kebutuhannya sehingga meminimalisir kesalahan.

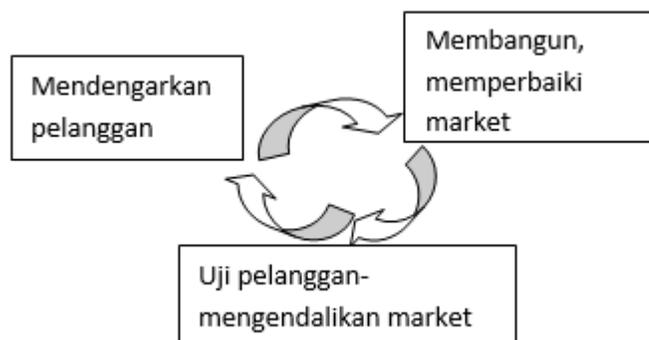
Kekurangan metode *waterfall*:

- Pembagian proyek yang dilakukan menjadi tahap-tahap yang tidak fleksibel sehingga komitmen harus dilakukan pada tahap awal proses..

- Sangat tidak fleksibel untuk permintaan yang berubah-ubah dari *customer*.
- *Customer* harus sabar menanti produk selesai. Hal ini dikarenakan dikerjakan secara tahap per tahap yaitu menyelesaikan tahap awal baru ke tahap selanjutnya.

2.5.2. *Prototype*

Metode ini dimulai dengan pengumpulan kebutuhan. Pendekatan *prototype metode* digunakan jika pemakai hanya mendefenisikan objektif umum dari perangkat lunak tanpa merinci kebutuhan input, pemrosesan dan outputnya, sementara pengembang tidak begitu yakin akan efisiensi algoritma, adaptasi sistem operasi, atau bentuk antarmuka manusia-mesin yang harus diambil.



Gambar 2.18. Metode *Prototype*

Secara ideal *prototype* berfungsi sebagai sebuah mekanisme untuk mengidentifikasi kebutuhan perangkat lunak. Bila *prototype* yang sedang bekerja dibangun, pengembang harus menggunakan

fragmen-fragmen program yang ada atau mengaplikasikan alat-alat bantu (contoh: window manager) yang memungkinkan program yang bekerja agar dimunculkan secara cepat.

Proses pada metode *prototyping* dijelaskan dibawah ini:

- Pengumpulan

Developer dan klien bertemu untuk menentukan tujuan umum, kebutuhan yang dibutuhkan klien, dan gambaran kebutuhan berikutnya. Kebutuhan detail mungkin tidak dibicarakan pada awal pertemuan pengumpulan kebutuhan ini.

- Perancangan

Perancangan dilakukan dengan cepat serta mewakili aspek-aspek perangkat lunak yang diketahui. Rancangan ini menjadi dasar pembuatan *prototype*.

- Evaluasi *Prototype*

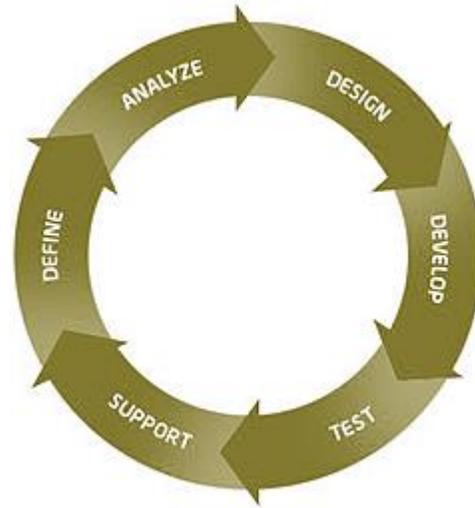
Klien bertugas untuk mengevaluasi terhadap *prototype* yang dibuat untuk memperjelas kebutuhan *software*.

Kelemahan metode *prototype*:

- Klien bisa saja tidak menyadari bahwa mungkin saja *prototype* dibuat terburu-buru. Sehingga rancangan yang diharapkan benar oleh klien dikerjakan dengan susunan yang salah.
- Pengembangan terkadang dapat menyebabkan implementasi sembarangan dikarenakan pekerjaan yang diinginkan bekerja dengan cepat.

2.5.3. RAD (*Rapid Application Development*)

RAD adalah sebuah metode pengembangan perangkat lunak secara *linear sequential* yang menekankan pengembangan perangkat lunak yang sangat singkat. Jika kebutuhan dipahami dengan baik, proses RAD memungkinkan tim pengembangan menciptakan “sistem fungsional yang utuh” dalam periode waktu yang sangat pendek (kira-kira 60-90 hari).



Gambar 2.19. Metode RAD

Pendekatan metode RAD menekankan beberapa hal berikut:

- Pemodelan Bisnis (*Business Modelling*)

Aliran informasi berada diantara fungsi-fungsi bisnis dimodelkan dengan cara menjawab pertanyaan-pertanyaan seperti: informasi apa yang yang mengendalikan proses bisnis? Kemana informasi itu pergi? Siapa yang memprosesnya?

- Pemodelan Data (*Data Modelling*)

Aliran informasi yang didefinisikan sebagai bagian dari fase pemodelan bisnis disaring ke dalam serangkaian objek data yang dibutuhkan untuk menopang bisnis tersebut. Karakteristik/atribut dari masing-masing objek diidentifikasi dan hubungan antara objek-objek tersebut didefinisikan.

- *Pemodelan Proses (Process Modelling)*

Aliran informasi didefinisikan sebagai fase pemodelan dan ditransformasikan untuk mencapai aliran informasi yang perlu bagi implementasi fungsi bisnis. Gambaran proses yang terjadi diciptakan untuk menambah, memodifikasi, menghapus, dan mendapatkan kembali sebuah objek data. Hal ini untuk mempermudah dalam pemodelan.

- *Pembuatan Aplikasi (Application Generation)*

Selain menciptakan perangkat lunak dengan menggunakan bahasa pemrograman generasi ketiga yang konvensional, RAD lebih banyak memproses kerja untuk memakai lagi komponen program yang telah ada atau menciptakan komponen yang bias dipakai lagi. Pada semua kasus, alat-alat Bantu otomatis dipakai untuk memfasilitasi konstruksi perangkat lunak

- Pengujian dan Pergantian (*Testing and Turnover*)

Dikarenakan proses RAD menekankan pada pemakaian kembali sehingga banyak komponen yang telah diuji. Hal ini mengurangi keseluruhan waktu pengujian tetapi setiap komponen baru harus diuji.

Kelemahan metode RAD:

- Pada proyek dengan skala besar, RAD membutuhkan sumber daya manusia yang mumpuni untuk membentuk sebuah tim RAD yang berkualitas baik.
- RAD sangat membutuhkan pengembangan sehingga klien yang mempunyai komitmen dalam aktivitas *rapid-fire* untuk melaksanakan aktivitas dalam hal melengkapi sistem dalam kerangka waktu yang singkat. Jika komitmen tidak punyai oleh klien maka proyek RAD dianggap gagal.
- RAD tidak fleksibel pada semua aplikasi dikarenakan bila sistem tidak dapat dimodulkan dengan teratur maka pembangunan komponen penting pada RAD akan menjadi sangat bermasalah. RAD menjadi tidak sesuai jika resiko teknis tinggi.

2.6. Map

Menurut E. Prahasta, (2001) pemetaan adalah suatu proses menyajikan informasi muka bumi yang berupa fakta (dunia nyata, baik bentuk permukaan buminya maupun sumberdaya alamnya, berdasarkan skala peta, sistem proyeksi peta, serta simbol-simbol dari unsur muka bumi yang disajikan. Peta adalah representasi visual dari keseluruhan area atau sebagian area, biasanya ditunjukkan pada permukaan datar. Peta sering digunakan untuk menggambarkan fitur spesifik dan rinci dari area tertentu. Bentuk data peta yang dikenal dan masih tersedia hingga saat ini terdiri dari dua jenis yaitu; data konvensional dan data digital.

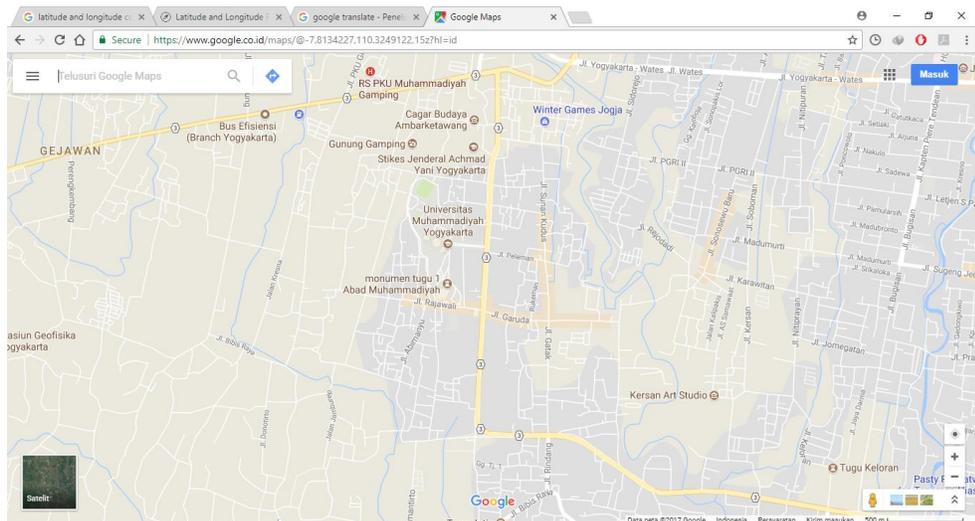
Peta konvensional adalah peta kertas hasil teknologi analog. Peta semacam ini cukup sulit untuk dimutakhirkan, karena praktis seluruhnya harus digambar ulang, tidak cukup bagian yang berubah saja. Selain itu penggunaannya juga terbatas, tidak mudah ditampilkan dalam format berbeda, dan tidak bisa langsung diproses dengan teknologi digital lainnya.



Gambar 2.20. Peta konvensional

(Sumber: <http://fhd-planning.blogspot.co.id/2016/07/peta-konvensional.html>)

Sedangkan pemetaan digital adalah menggambarkan permukaan bumi menggunakan perangkat komputer dengan acuan pada data koordinat. Pada intinya pemetaan digital merupakan proses pengolahan objek-objek pada peta yang menggunakan format digital maka membutuhkan sebuah *hardware* komputer dan *software* yang berkaitan dengan pemetaan digital. Perangkat lunak yang biasa digunakan dalam pembuatan pemetaan digital seperti *land desktop*, *auto cad map*, *arc view*, *map info professional*, dan lain-lain.



Gambar 2.21. Peta Digital

(Sumber: <https://www.google.co.id/maps/@-7.8032376,110.3249238,15z?hl=id>)

Peta memiliki koordinat yaitu *latitude* (lintang) dan *longitude* (bujur). Lintang dan bujur adalah sebuah unit yang mewakili koordinat pada sistem koordinat geografis. Hal ini sama seperti rumah yang memiliki sebuah alamat rumahnya secara detail. Setiap titik di permukaan bumi dapat ditentukan oleh koordinat

dengat garis lintang dan bujur. Oleh sebab itu, menggunakan garis lintang dan bujur dapat menentukan hamper seluruh titik yang ada di bumi.

Lintang mempunyai simbol phi sehingga ini menunjukkan sudut antara garis lurus pada titik tertentu pada bidang ekuator. Lintang ditentukan dengan derajat yang dimulai dari 0° berakhir dengan 90° pada kedua sisi khatulistiwa. Oleh sebab itu membuat garis lintang utara dan selatan. Khatulistiwa adalah garis dengan garis lintang 0° . Bujur memiliki simbol lambda dan merupakan koordinat sudut lain yang menentukan posisi suatu titik pada permukaan bumi. Garis bujur dapat diartikan sebagai sudut yang menunjuk ke barat atau timur dari meridian Greenwich yang diambil sebagai meridian utama. Bujur dapat diartikan sudut maksimum 180° BT dari meridian perdana dan 180° BB dari meridian perdana.

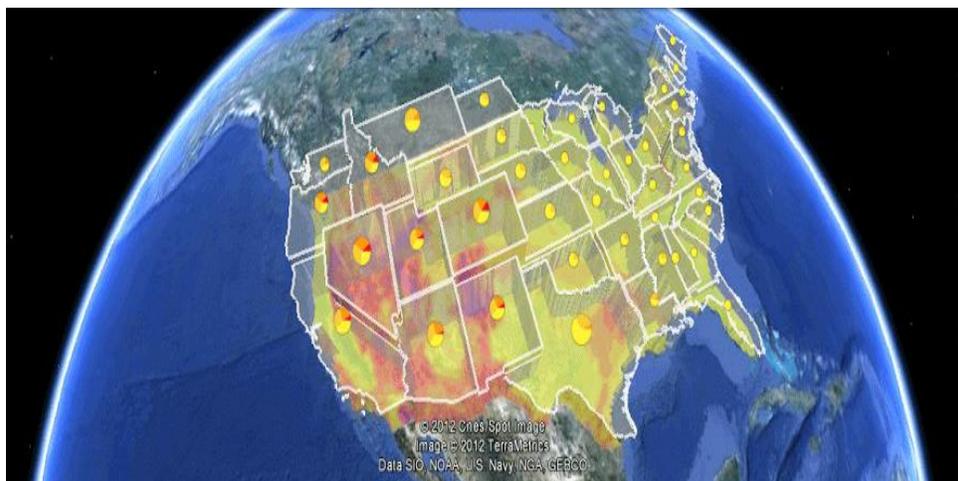
Garis lintang dan bujur diukur dalam derajat sehingga pada gilirannya dibagi menjadi beberapa menit serta detik. Sebagai contoh zona tropis yang terletak di selatan dan utara dari khatulistiwa ditentukan oleh batas $23^\circ 26'13,7''$ S dan $23^\circ 26'13,7''$ N, atau contoh lainnya koordinat geografis gunung Ngauruhoe di Selandia Baru yang terkenal dengan area syuting film *Lord of the Rings*, memiliki koordinat geografis $39^\circ 09'24.6''$ S $175^\circ 37'55.8''$ E.

2.7. KML (*Keyhole Markup Language*)

KML (*Keyhole Markup Language*) adalah XML (*Extensible Markup Language*) yang bertujuan fokus pada visualisasi grafis termasuk anotasi peta dan citra. Visualisasi geografis mencakup tidak hanya penyajian data grafik di peta

dunia tetapi dalam hal navigasi kendali dalam mengarahkan pengguna saat penggunaan peta. Data KML dapat ditampilkan dalam *Earth browser* seperti *Google Earth* dengan menggunakan proyeksi silinder sederhana untuk basis pencitraannya. Proyeksi peta ini cukup sederhana dengan sistem paralel dan meridian merupakan ekuidistan, garis horizontal dan kedua garis memotong pada sudut tegak lurus. Proyeksi ini juga dikenal sebagai Lintang/Bujur WGS84.

KML (*Keyhole Markup Language*) memiliki beberapa *setting* struktur seperti lokasi, foto, *polygon*, bentuk 3D, dan lain-lain untuk ditampilkan pada *google earth*, *google maps*, dan *mobile*. Pada setiap lokasi mempunyai *latitude* (lintang) dan *longitude* (bujur). Sehingga data XML (*Extensible Markup Language*) inilah dapat dipetakan ke dalam *Google Earth* menjadi sebuah *layer* KML.



Gambar 2.22. KML (*Keyhole Markup Language*)

(Sumber: <https://developers.google.com/kml/>)

Berikut contoh skrip dari kml:

```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta name="viewport" content="initial-scale=1.0, user-scalable=no">
5     <meta charset="utf-8">
6     <title>KML Click Capture Sample</title>
7     <style>
8       html, body {
9         height: 370px;
10        padding: 0;
11        margin: 0;
12      }
13      #map {
14        height: 360px;
15        width: 300px;
16        overflow: hidden;
17        float: left;
18        border: thin solid #333;
19      }
20      #capture {
21        height: 360px;
22        width: 480px;
23        overflow: hidden;
24        float: left;
25        background-color: #ECECFB;
26        border: thin solid #333;
27        border-left: none;
28      }
29    </style>
30  </head>
31  <body>
32    <div id="map"></div>
33    <div id="capture"></div>
34    <script>
35      var map;
36      var src = 'https://developers.google.com/maps/documentation/javascript/examples/kml/westcampus.kml';
37
38      function initMap() {
39        map = new google.maps.Map(document.getElementById('map'), {
40          center: new google.maps.LatLng(-19.257753, 146.823688),
41          zoom: 2,
42          mapTypeId: 'terrain'
43        });
44
45        var kmlLayer = new google.maps.KmlLayer(src, {
46          suppressInfoWindows: true,
47          preserveViewport: false,
48          map: map
49        });
50        kmlLayer.addListener('click', function(event) {
51          var content = event.featureData.infoWindowHtml;
52          var testimonial = document.getElementById('capture');
53          testimonial.innerHTML = content;
54        });
55      }
56    </script>
57    <script async defer
58      src="https://maps.googleapis.com/maps/api/js?key=YOUR_API_KEY&callback=initMap">
59    </script>
60  </body>
61 </html>

```

Gambar 2.23. Contoh Script KML (Keyhole Markup Language)

2.8. Harvesine Formula

Harvesine formula adalah persamaan yang sangat penting pada navigasi. Formula ini dapat menghitung jarak diantara dua titik pada lingkaran bola dari setiap garis bujur (*longitude*) dan garis lintang (*latitude*). *Harvesine formula* adalah sebuah formula umum pada trigonometri lingkaran bola. Dimana *harvesine formula* menghubungkan sisi dan sudut segitiga bola untuk menentukan jarak pada kedua sisi tersebut.

Harversine formula pada penelitian ini digunakan untuk perhitungan jarak antara dua titik koordinat yang dikirimkan oleh GPS muatan. Koordinat yang dimaksud yaitu koordinat *user* dan koordinat tujuan sehingga dapat menjadi sebuah acuan dalam perbandingan jarak pada penentuan jarak terdekat maupun jarak terjauh. Rumus *harvesine* yang berlaku yaitu:

$$a = \sin^2(\Delta\phi/2) + \cos \phi_1 \cdot \cos \phi_2 \cdot \sin^2(\Delta\lambda/2)$$

$$c = 2 \cdot \text{atan2}(\sqrt{a}, \sqrt{1-a})$$

$$d = R \cdot c$$

Keterangan:

d = Jarak antara dua titik

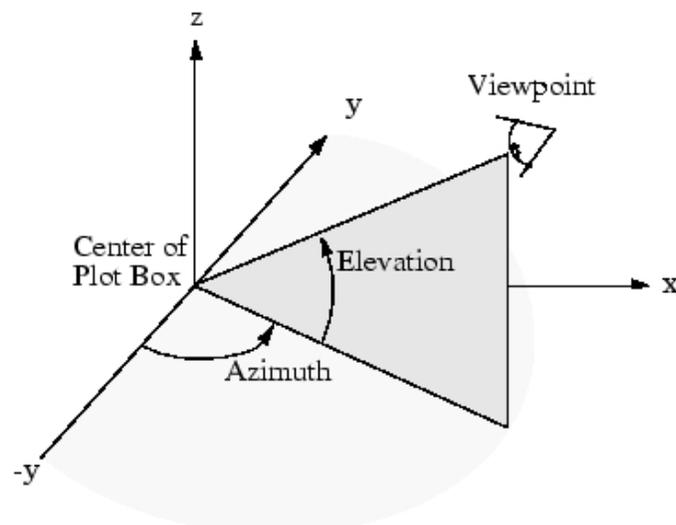
R = Radius dari lingkaran bulat (radius = 6,371km)

ϕ_1, ϕ_2 = Latitude dari titik 1, latittude dari titik 2

λ_1, λ_2 = Longitude dari titik 1, longitude dari titik 2

2.9. Sudut Elevasi dan Sudut Azimut

Azimuth merupakan sudut horizontal yang diukur searah jarum jam dari suatu garis dasar utara dalam sebuah lingkaran. Nilai sudut dari 1^0 sampai 360^0 atau 1 sampai 6400 mil. Utara yang dimaksud bisa saja utara sebenarnya, utara magnetis ataupun utara *grid* atau peta. Sedangkan sudut elevasi adalah sudut yang terbentuk antara garis lurus mendatar dengan posisi pengamat ke atas. Sudut azimuth dan sudut elevasi digunakan pada penelitian sebagai acuan *tracking* antenna terhadap pergerakan muatan balon atmosfer selama berada di atmosfer. Berikut contoh dari sudut azimuth dan sudut elevasi seperti pada gambar 2.24.



Gambar 2.24. Sudut Azimuth dan Sudut Elevasi

(Sumber: <http://www.bu.edu/tech/support/research/training-consulting/online-tutorials/visualization-with-matlab>)