

BAB III

RANCANG BANGUN PERANGKAT LUNAK

3.1. Metode Rancang Bangun Perangkat Lunak

Metodologi yang digunakan dalam pembuatan perangkat ini, yaitu:

a. Studi Literatur

Sebagai landasan dalam membuat perangkat lunak ini, diperlukan teori penunjang yang memadai mengenai ilmu dasar pembuatan antarmuka dan penyimpanan data. Teori penunjang tersebut dapat bersumber dari buku pegangan, jurnal ilmiah, dan media *online*.

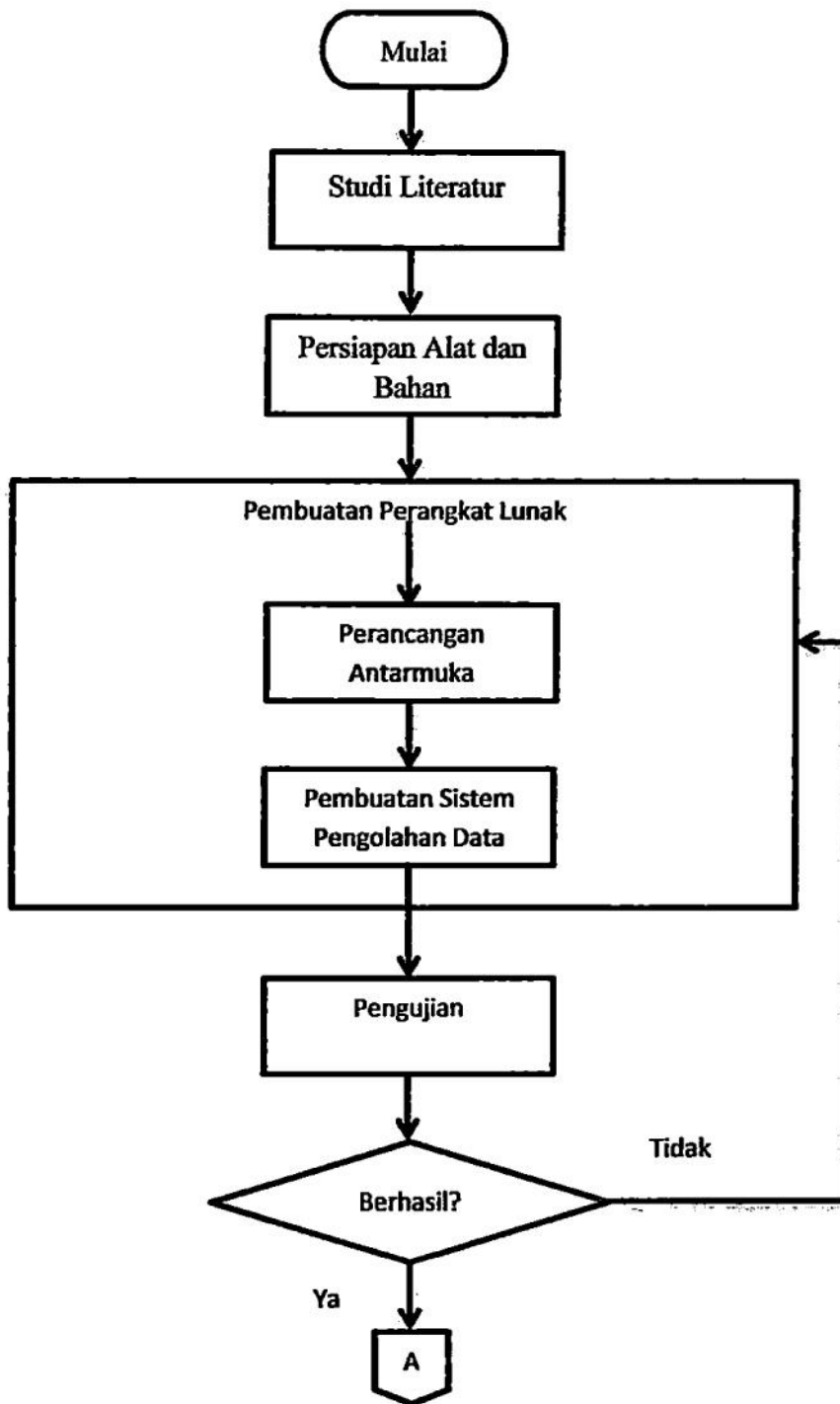
b. Pembuatan Perangkat Lunak

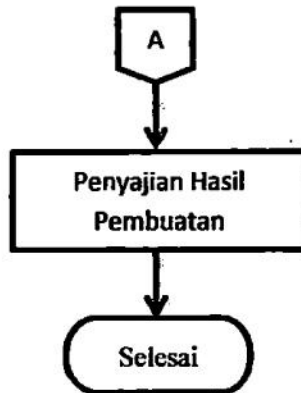
Pada tahap ini akan dilakukan pembuatan perangkat lunak, yaitu antarmuka dan sistem penyimpanan data profil atmosfer dimana data yang diperoleh merupakan data yang dikirimkan oleh *radiosonde* Rev-Gaardan dengan struktur data yang telah ditentukan sebelumnya.

c. Pengujian Hasil Pembuatan

Pada tahap ini akan dilakukan pengujian hasil pembuatan perangkat lunak dari segi kemampuan menampilkan dan menyimpan data hasil transmisi. Jika belum berhasil, maka akan dilakukan pengecekan ulang pada bagian yang mengalami gangguan.

Secara garis besar proses pembuatan ini dapat digambarkan dalam sebuah diagram alir yang dapat dilihat pada gambar 3.1.

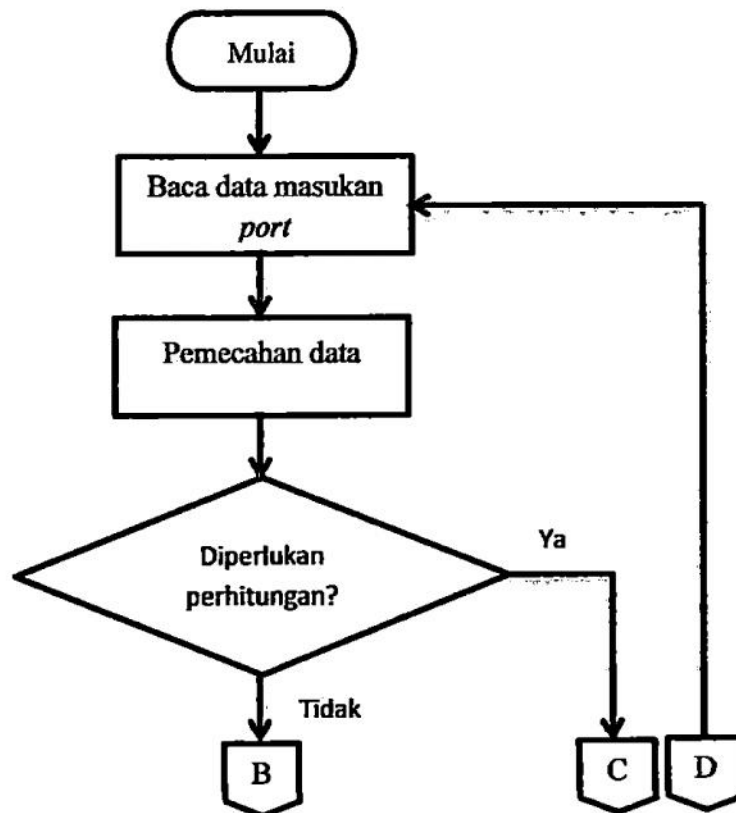


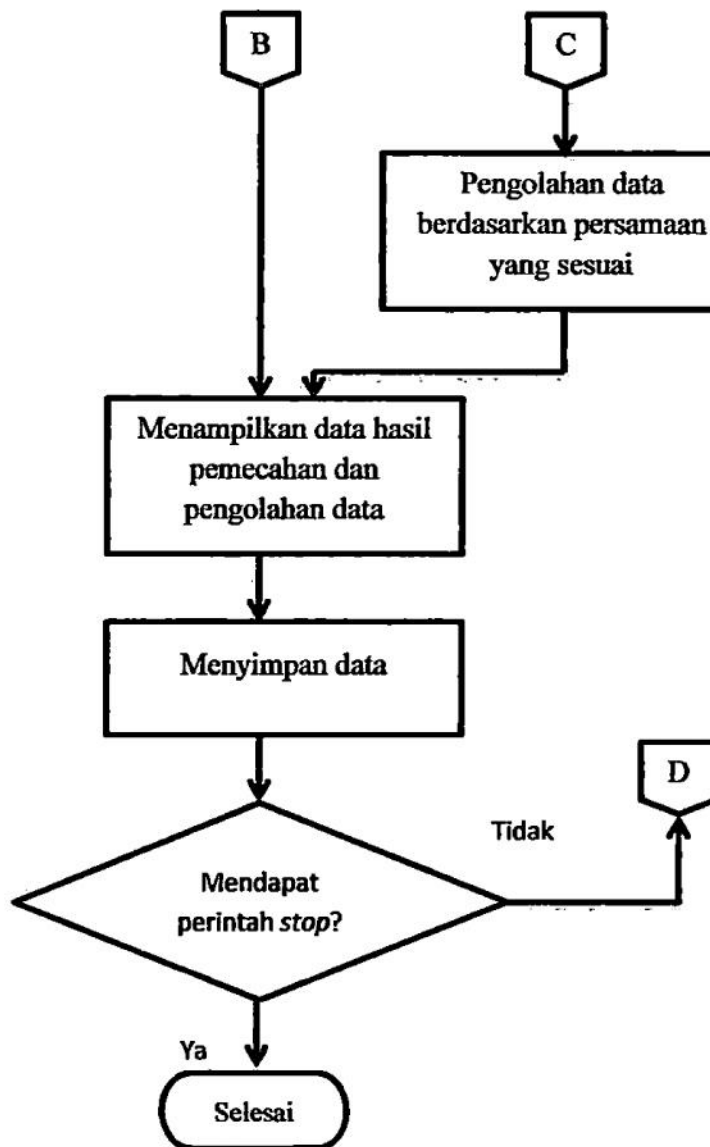


Gambar 3.1 Diagram alir pembuatan perangkat lunak

3.2. Perinsip Kerja Perangkat Lunak

Secara garis besar, jalannya sistem perangkat lunak ini digambarkan dalam sebuah diagram alir yang dapat dilihat pada gambar 3.2.





Gambar 3.2 Diagram alir jalannya program

3.3. Perancangan Antarmuka

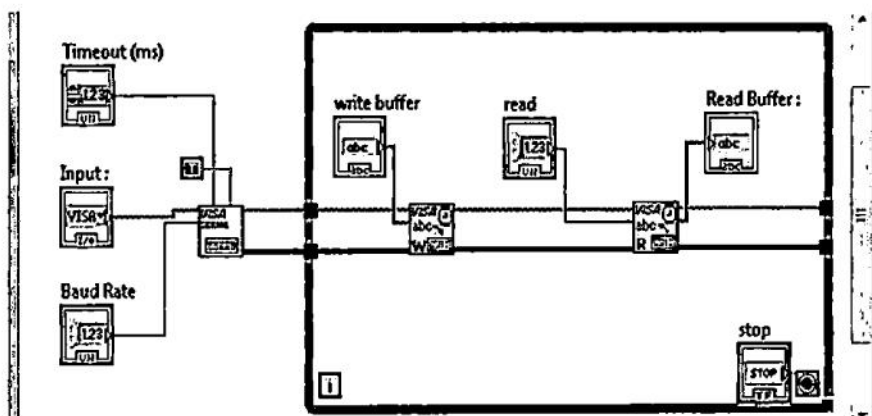
Dalam perancangan antarmuka pada penelitian ini digunakan perangkat lunak *LabVIEW Student Edition*. *LabVIEW Student Edition* memiliki fungsi yang sama dengan *LabVIEW Full Development*. Penggunaan *LabVIEW Student Edition* ini bertujuan untuk menghindari pembajakan *software*. *LabVIEW Student Edition* disediakan secara gratis oleh pihak National Instrument yang dapat diunduh pada tautan <https://www.ni.com/>.

3.3.1. Komunikasi Serial LabVIEW

Tahap pertama perancangan ini adalah bagaimana agar LabVIEW mampu berkomunikasi secara serial melalui *port* pada computer agar LabVIEW mampu berkomunikasi secara serial maka digunakan fungsi VISA. Dalam perancangan ini digunakan 3 macam fungsi VISA, yaitu :

- VISA Configure Serial Port*. Fungsi ini digunakan untuk mengatur setting komunikasi serial di awal sebelum komunikasi dilangsungkan, termasuk saluran yang digunakan (*Visa resource name*), kecepatan komunikasi (*baud rate*), jumlah data bit, *stop bit*, dan lain-lain.
- VISA Write*. Fungsi ini digunakan untuk menulis atau mengirimkan data dari *write buffer* ke suatu alat atau *hardware interface* yang ditentukan oleh *VISA resource name*.
- VISA Read*. Fungsi ini digunakan untuk membaca sejumlah *byte* data dari alat atau *hardware interface* yang ditentukan oleh *VISA Resource name*.

Agar sistem dapat berjalan terus menerus, maka perlu ditambahkan struktur *While Loop*, sehingga ikon-ikon di dalamnya terus di eksekusi hingga terminal masukan *conditional (stop if true)* mendapat nilai *False*, dan perulangan akan berhenti ketika mendapat nilai *True*. Berikut ini merupakan gambar program pada jendela *block diagram* untuk komunikasi serial pada LabVIEW:

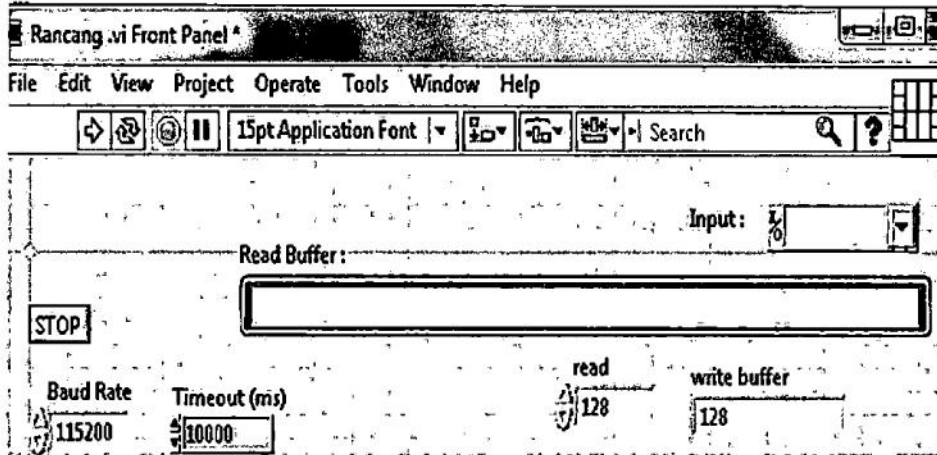


Gambar 3.3 Block Diagram komunikasi serial LabVIEW

Pada program tersebut terdapat pengaturan *time out*, *baud rate*, *write buffer byte*, dan *read buffer byte*. Setiap pengaturan tersebut memiliki fungsi masing-masing, yaitu:

- a. *Time Out*: Berfungsi untuk mengatur lamanya waktu jeda (dalam mili detik) pembacaan data *port* pada program. Semisal pada program diberikan pengaturan *time out* dengan nilai 5000, maka apabila program tidak menerima data dalam waktu 5000 mili detik, maka program akan memberikan nilai pembacaan dengan nilai 0.
- b. *Baud rate*: Berfungsi untuk mengatur kecepatan aliran data. Nilai kecepatan dari program dan *hardware* harus sama agar komunikasi dapat berjalan dengan baik.
- c. *Write buffer byte*: Berfungsi untuk mengatur jumlah *byte* yang dibaca oleh *VISA Write*.
- d. *Read buffer byte*: Berfungsi untuk mengatur jumlah *byte* yang dibaca oleh *VISA Read*.

Agar pengaturan tersebut dapat diatur sewaktu-waktu tanpa membuka jendela *block diagram*, maka fungsi-fungsi tersebut dirancang menjadi sebuah *control* dan ditampilkan pada jendela *Front panel*. Untuk mengetahui data mentah yang masuk ke program, diberikan sebuah indikator pada *VISA Serial*. Berikut ini tampilan antarmuka yang terbentuk dari *block diagram* gambar 3.3 :



Gambar 3.4 Tampilan antarmuka pada jendela *Front Panel*

3.3.2. Parsing Data

Data yang diterima dari komunikasi serial LabVIEW merupakan data mentah yang mengandung beberapa informasi yang akan diolah pada program. Dalam satu kali komunikasi, program mendapatkan satu baris data yang berisi beberapa nilai-nilai seperti nilai ketinggian, suhu, tekanan udara, kelembaban relatif sertera koordinat. Setiap nilai tersebut memiliki *header* masing-masing dan perlu dipisahkan, atau disebut sebagai dengan proses *parsing data*. Karena format data yang diterima sudah diketahui, maka digunakan fungsi *scan from string*. Fungsi ini hanya dapat digunakan ketika susunan data dan *headernya* sudah diketahui. Muatan Rev-Gaardan memiliki format pengiriman data seperti berikut::

ALT(x1)TEM(x2)PRE(x3)HUM(x4)LAT(x5)LON(x6)

Dimana : x1 = nilai ketinggian

x2 = nilai suhu

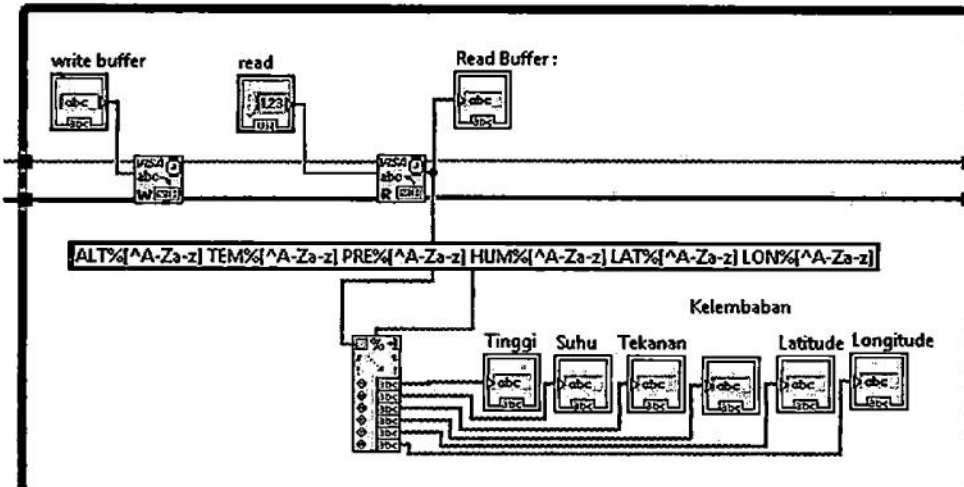
x3 = nilai tekanan udara

x4 = nilai kelembaban relatif

x5 = nilai latitude

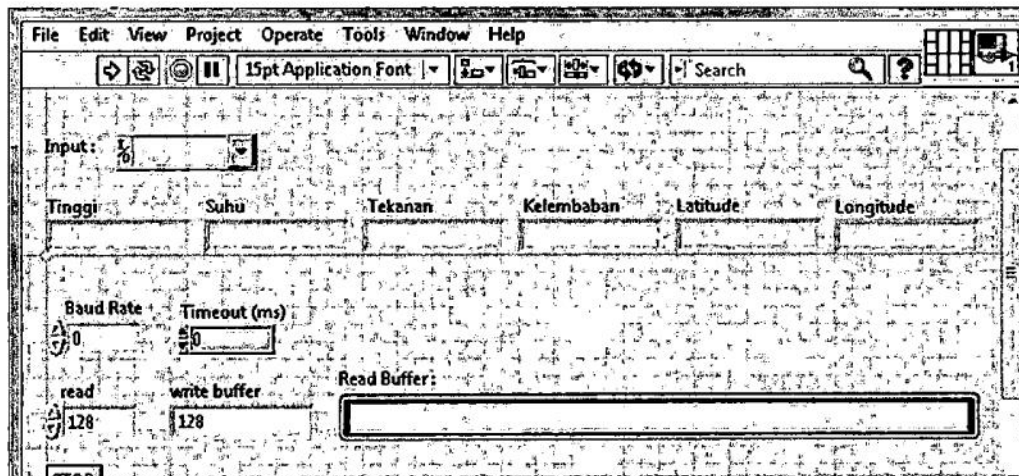
x6 = nilai longitude

Sehingga program pada jendela block diagram dirancang seperti berikut:



Gambar 3.5 Block Diagram parsing data

Data hasil *parsing data* kemudian ditampilkan dalam bentuk indikator *string*, berikut tampilan pada jendela *front panel*:



Gambar 3.6 Tampilan pada *Front Panel* setelah *parsing data*

3.3.3. Konversi Tekanan Ke Ketinggian

Nilai ketinggian dapat diperoleh dengan dua cara, yaitu dari GPS dan dari konversi dari tekanan. Nilai ketinggian yang diperoleh dari hasil komunikasi serial LabVIEW merupan nilai yang diperoleh dari GPS, dan untuk memperoleh ketinggian dari tekanan diperoleh dengan persamaan berikut:

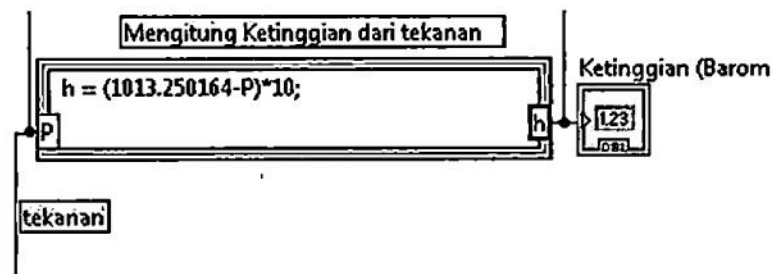
$$h = ((76 * 13,332239) - p_h) * 10 \text{ m} \dots\dots\dots (3.1)$$

$$h = (1013,250164 - p_h) * 10 \text{ m} \dots\dots\dots (3.2)$$

dimana h = Ketinggian (dalam meter)

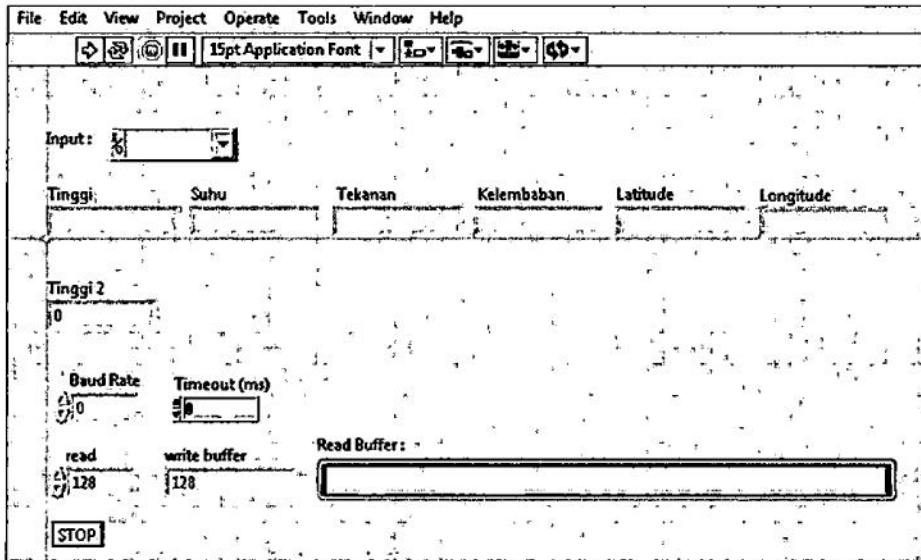
p_h = Tekanan udara pada ketinggian tersebut (dalam mbar)

Untuk melakukan perhitungan pada LabVIEW maka digunakan fungsi *Formula Node*. Masukan pada *Formula Node* adalah berupa data *number*, maka data hasil *parsing data* harus di ubah dulu dari *string* menjadi *number*. Berikut operasi perhitungan konversi tekanan ke ketinggian pada program :



Gambar 3.7 Operasi perhitungan konversi tekanan ke ketinggian

Data hasil konversi kemudian ditampilkan dalam bentuk indikator *number*, berikut tampilan pada jendela *front panel*.



Gambar 3.8 Tampilan pada *Front Panel* setelah konversi tekanan ke ketinggian

3.3.4. Perhitungan Sudut Perpindahan (*bearing degree*)

Sudut perpindahan atau *bearing degree* diperoleh dari perubahan posisi *latitude* dan *longitude* muatan *radiosonde*. Nilai sudut perpindahan dalam derajat atau *bearing degree* diperoleh dengan persamaan berikut:

$$\theta = \text{atan2}(\sin(\Delta\text{long}) \cdot \cos(\text{lat2}), \cos(\text{lat1}) \cdot \sin(\text{lat2}) - \sin(\text{lat1}) \cdot \cos(\text{lat2}) \cdot \cos(\Delta\text{long})) \dots (3.3)$$

dimana θ : *Bearing degree*

lat1 : posisi *latitude* muatan mula-mula

lat2 : posisi *latitude* muatan akhir

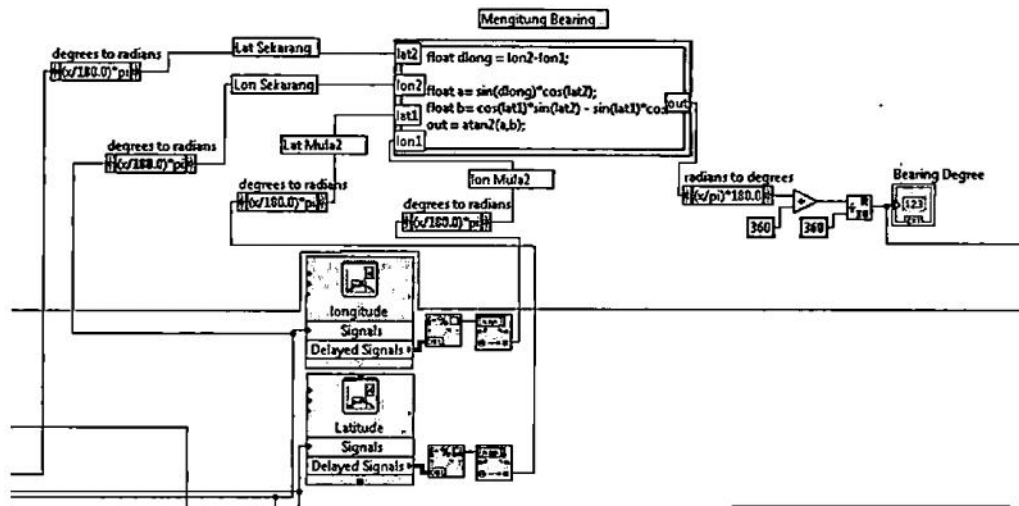
lon1 : posisi *longitude* muatan mula-mula

lon2 : posisi *longitude* muatan akhir

Δlong : Perubahan letak *longitude* ($\text{long2} - \text{long1}$)

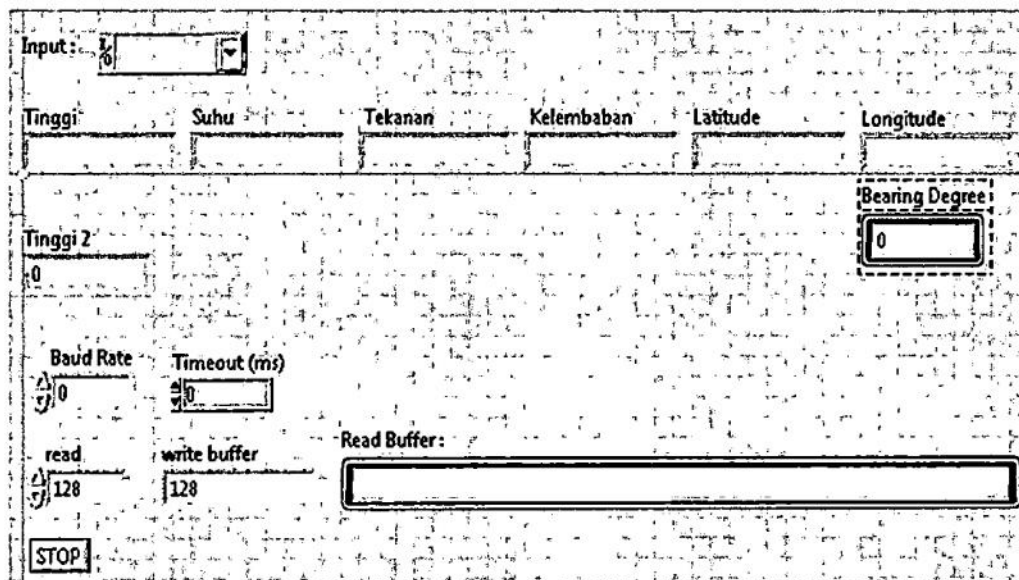
Pada pemrograman LabVIEW, untuk memperoleh nilai sebelumnya digunakan fungsi *delay value express VI*. Data keluaran dari fungsi ini adalah

aliran data yang ditunda sebanyak 1 tahap aliran. Berikut operasi perhitungan sudut perpindahan dalam derajat (*bearing degree*) pada program :



Gambar 3.9 Operasi perhitungan *bearing degree*

Data hasil operasi kemudian ditampilkan dalam bentuk indikator *number*, berikut tampilan pada jendela *front panel*:



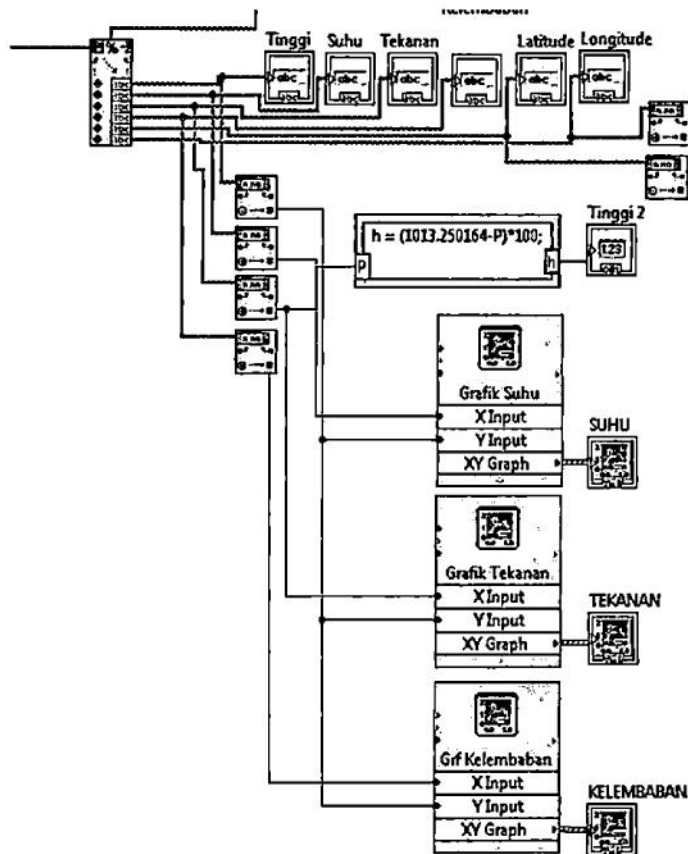
Gambar 3.10 Tampilan pada *Front Panel* setelah diperoleh *bearing degree*

3.3.5. Perancangan Grafik dan Indikator

Untuk memudahkan pemantauan perubahan data profil atmosfer, diperlukan adanya grafik dan indikator pada bagian antarmuka, sehingga pengguna dapat memahami data dengan mudah.

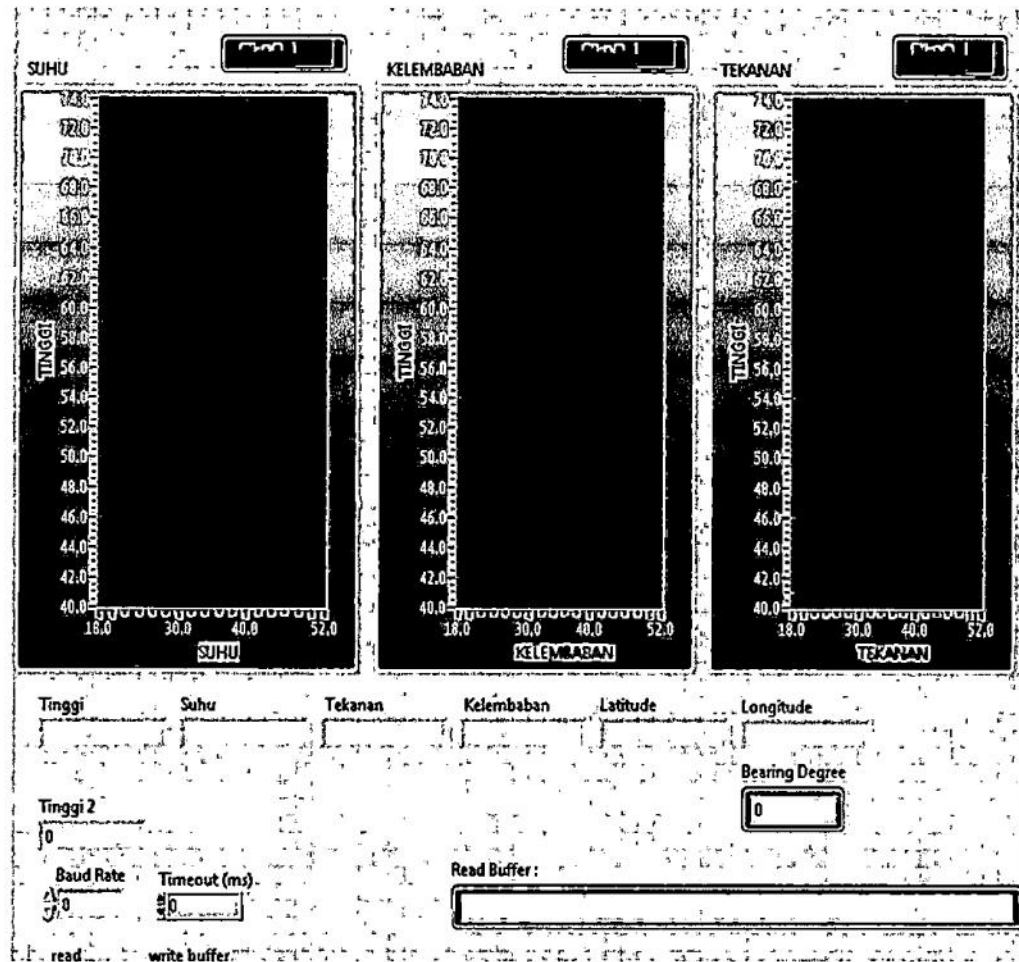
a. Pembuatan Grafik

Pada program ini digunakan fungsi *Build XY graph express VI*. Fungsi ini dipilih karena mampu menampilkan grafik yang tidak berbasis pada waktu dan memasukkan *input* dengan memisahkan nilai-nilai pada koordinat sumbu X dengan sumbu Y. Berikut ini perancangan grafik pada jendela *block diagram*:



Gambar 3.11 Perancangan grafik pada program

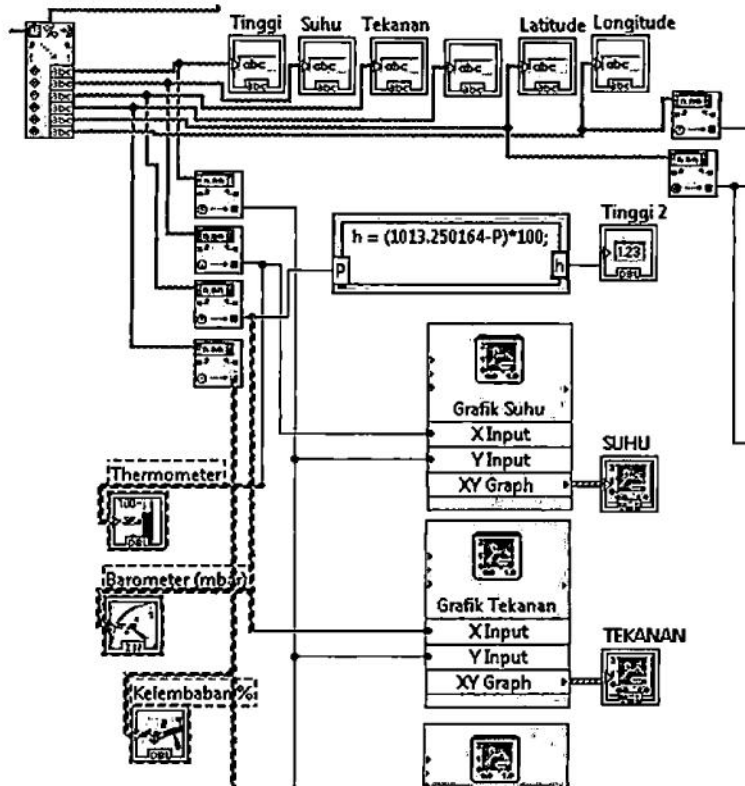
Berikut ini tampilan grafik pada jendela *front panel* yang terbentuk dari block diagram diatas:



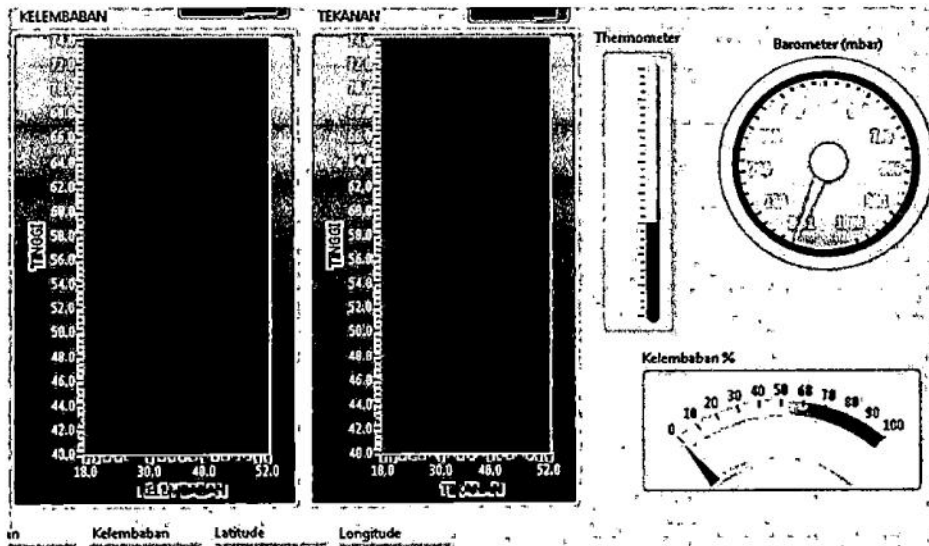
Gambar 3.12 Tampilan grafik pada jendela *front panel*

b. Pembuatan Indikator

Penambahan indikator bertujuan untuk menampilkan informasi data ke pengguna dalam bentuk skala. Berikut ini perancangan indikator pada jendela *block diagram* dan pada jendela *front panel*:



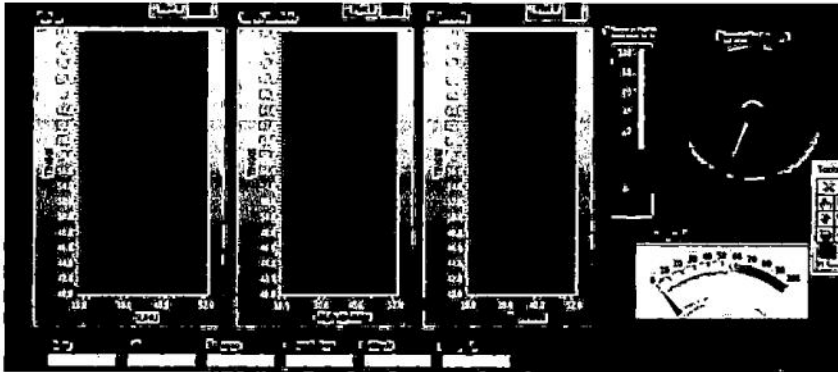
Gambar 3.13 Perancangan Indikator pada *block diagram*



Gambar 3.14 Indikator pada jendela *front panel*

c. Pengaturan Warna

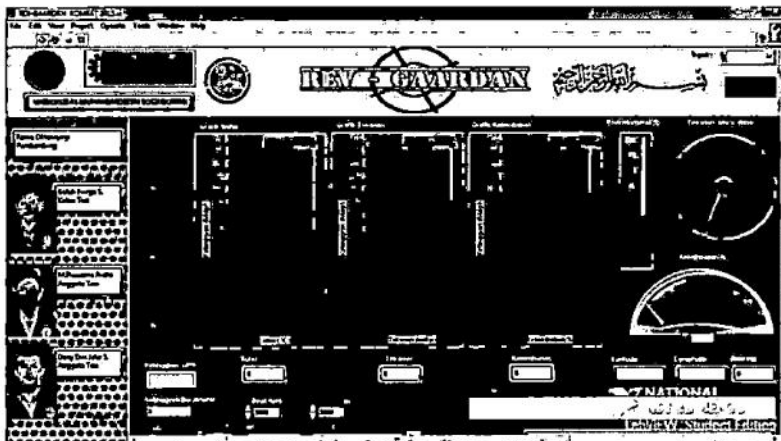
Pengaturan warna bertujuan untuk membuat tampilan antarmuka lebih menarik. Pengaturan warna hanya dapat dilakukan pada jendela *front panel*. Pengaturan warna dilakukan dengan *tool set color* yang terdapat pada *tools pallet*. Berikut ini tampilan antarmuka setelah dilakukan pengaturan warna :



Gambar 3.15 *Front panel* setelah pengaturan warna

d. Menyisipkan Gambar

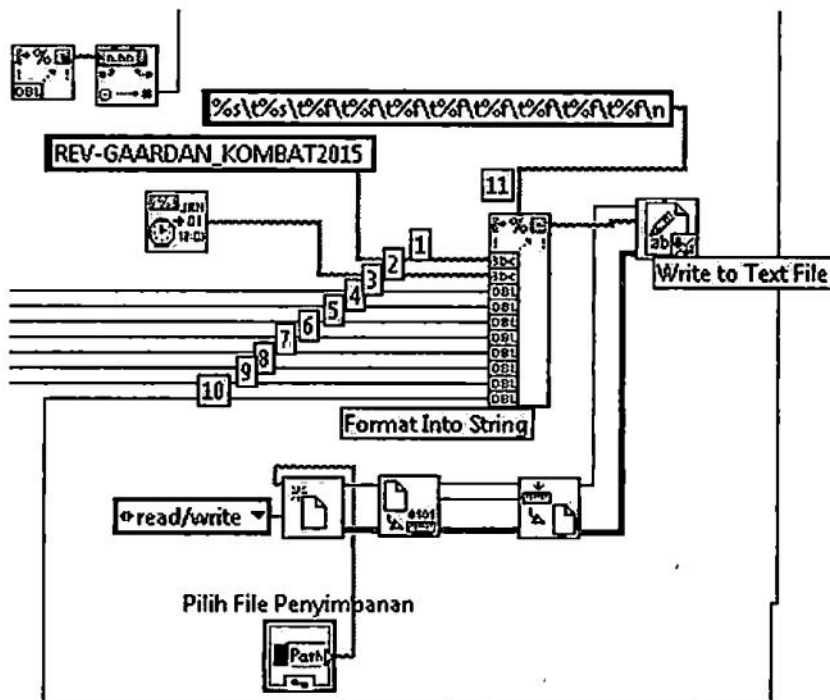
Penyisipan gambar juga dapat dilakukan untuk membuat tampilan antarmuka lebih menarik. Berikut ini tampilan akhir antarmuka tim Rev-Gaardan yang digunakan pada Kompetisi Muatan Balon Atmosfer 2015 setelah pengaturan letak, warna, dan penyisipan gambar pada jendela *front panel*.



Gambar 3.16 Tampilan antarmuka akhir

3.4. Perancangan *Data Logger*

Untuk menyimpan data hasil pengukuran dan perhitungan profil atmosfer diperlukan sebuah *data logger*. *Data logger* pada program ini menggunakan fungsi *Write To Text File*. Fungsi dipilih karena lebih efektif dibandingkan dengan metode penyimpanan menggunakan fungsi yang lain. Dengan fungsi ini data yang diperoleh dapat disimpan dalam bentuk *file .txt*, dan untuk memudahkan dalam pembacaan data hasil penyimpanan tersebut, maka digunakan *tab delimiter* sebagai pemisah pada setiap data (waktu, ketinggian, suhu, kelembaban, dan koordinat). Berikut ini perancangan *data logger* pada jendela *block diagram*.



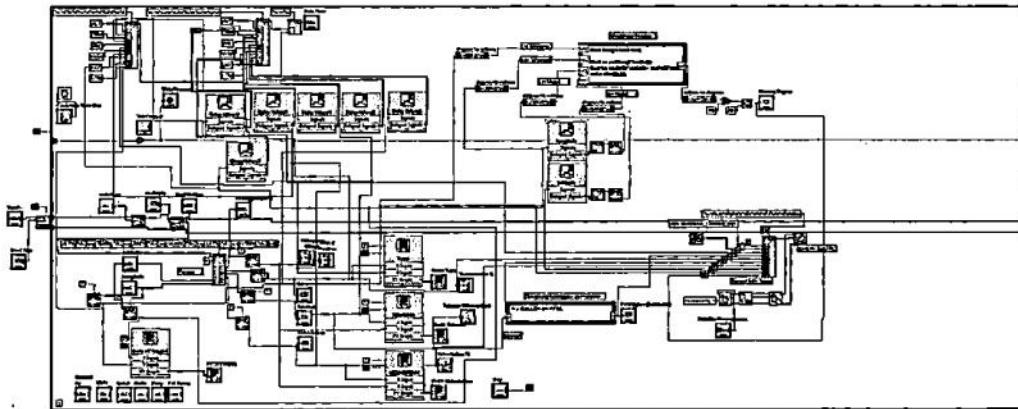
Gambar 3.17 Sistem penyimpanan data

Gambar diatas merupakan gambar sistem penyimpanan yang terdapat pada jendela *block diagram*. Sistem tersebut menggunakan 2 buah fungsi utama, yaitu *Write to Text File* dan *Format Into String*. Seluruh data dikumpulkan melalui fungsi *Format Into String* baru kemudian data *string* yang telah dikumpulkan disimpan dalam bentuk teks dengan fungsi *Write to Text File*.

Keterangan angka pada gambar:

- 1: Identitas
- 2: Waktu
- 3: Nilai ketinggian dari *GPS*
- 4: Nilai ketinggian dari konversi tekanan
- 5: Nilai masukan suhu
- 6: Nilai masukan tekanan
- 7: Nilai masukan kelembaban
- 8: Nilai masukan *altitude*
- 9: Nilai masukan *longitude*
- 10: Nilai masukan *bearing*
- 11: Format penulisan *string*

3.5. *Block Diagram* keseluruhan



Gambar 3.18 *Block Diagram* keseluruhan

3.6. Pembuatan *File .Exe*

Setelah seluruh program selesai dibuat, langkah terakhir dari perancangan ini adalah membuat *file Exe*. Tujuan dari dibuatnya *file Exe* ini adalah pengguna program tidak akan dapat mengubah atau melihat kode program yang dibuat oleh pemrogram. Pembuatan *file Exe* ini dapat dilakukan dengan *tools Build Application (EXE) From VI*. Setelah seluruh proses selesai, maka akan didapat *file Exe* yang dapat dijalankan tanpa dapat diubah kode programnya lagi dan program akan terus berjalan selama program belum mendapat perintah *stop* dari penggunanya.