BAB IV

HASIL DAN PEMBAHASAN

4.1. Perancangan Sistem

Rencana perancangan pengembangan sistem baru yaitu untuk memecahkan masalah GCS yang tidak dapat mendeteksi lokasi *real-time payload* di bumi dan menambah fitur baru dari sistem informasi GCS berupa penanaman sistem GPS.

4.1.1. Tujuan Perancangan Sistem

Tujuan dari perancangan sistem ini yaitu untuk proses rencana persiapan dari spesifikasi yang terperinci untuk pengembangan suatu sistem yang baru. Dimulai dengan perancangan sistem awal yang kemudian dimodifikasi sampai terdapat persetujuan dengan manajeman. Tahap perancangannya ini harus menyesuaikan semua rincian rencana pengembangan sistem baru yang telah ditentukan sebelumnya agar sistem yang baru dapat diimplementasikan dengan memuaskan.

Tujuan perancangan sistem secara umum yaitu untuk membentuk kerangka sistem baru yang dapat mengetahui posisi *payload* pada saat setelah roket meluncurkan *payload.* Sedangkan tujuan utama dari perancangan sistem yaitu untuk memberikan gambaran umum kepada pengguna sistem mengenai sistem informasi yang baru.

Penentuan persyaratan sistem dilakukan agar arah perancangan sistem dapat benar-benar terarah pada tujuan yang ditetapkan, oleh karena itu sistem yang

dirancang harus memenuhi batasan sistem dimana perancangan sistem ini merupakan kebutuhan fungsional dan persiapan untuk rancang bangun dan implementasi yang menggambarkan bagaimana suatu sistem dibentuk. Pada tahap perancangan sistem informasi dirancang dengan tujuan mengkomunikasikan kebutuhan kepada pemakai, bukan kepada *programmer*.

4.1.2. Gambaran Umum Sistem yang Diusulkan

Sistem informasi GCS posisi global *payload* perlu untuk dikembangkan mengingat sistem informasi GCS yang sudah ada sebelumnya belum dapat mendukung atau memberikan informasi di mana posisi *payload* secara *real-time* berada di bumi yang jatuh dan tertiup angin laut setelah diluncurkan oleh RUM, sehingga menyebabkan kerugian materi.

Untuk mengantisipasi masalah tersebut maka diperlukan suatu sistem informasi posisi global yang memudahkan pengguna mengetahui posisi global *payload* di bumi secara *real-time* dan mencegah kehilangan *payload*.

4.1.3. Perancangan Prosedur Yang Diusulkan

Prosedur baru yang diusulkan penulis dengan prosedur yang sudah ada terdapat perbedaan yaitu prosedur yang lama *payload* tidak terdapat sistem GPS, tidak mengirim data GPS dan tidak ada perangkat pengolah data GPS dari *payload* untuk diolah dan tampilkan dalam bentuk visual *user-friendly* yang menarik.

Berikut perancangan prosedur yang peneliti usulkan dalam pengembangan sistem informasi GCS ini :

1. *Payload* dihidupkan, setelah dihidupkan *payload* dalam kondisi *standby* untuk siap menerima perintah-perintah dari pengguna.
2. Pengguna memberikan suatu perintah melalui GUI GCS.
3. GCS mengkonversikan perintah tersebut kedalam format data yang dimengerti oleh *payload.*
4. *Payload* menerima perintah dan mengidentifikasi perintah tersebut untuk kemudian digunakan sebagai acuan apa yang harus dilakukan *payload* selanjutnya.
5. Jika perintah sama dengan "#0" maka *payload* kembali *standby*, atau sama dengan proses 1, jika tidak lanjut ke proses 6.
6. Apakah perintah sama dengan "#1"?, jika ya masuk ke proses 7, jika tidak masuk ke proses 16.
7. *Payload* mengukur atau mengambil data sensor-sensor *accelerometer* (percepatan), *gyroscope* (perubahan sudut), *magnetometer* (pengukur medan magnet) dan data posisi melalui modul GPS *receiver (latitude, longitude, altitude)*.
8. *Payload* mengenkapsulasi data-data tersebut untuk disajikan kedalam format pengiriman data yang telah ditentukan oleh juri kompetisi sebelumnya.
9. *Payload* mengirimkan data-data pengukuran 3 sensor dan GPS dalam format data KOMURINDO, melalui modul RF(*Radio Frequency*) untuk ditersukan ke modul RF penerima di *ground control station* yang ada di bumi.
10. Modul RF penerima di GCS meneruskan ke rangkaian *converter* untuk mengubah format data yang dimengerti oleh Laptop/PC.

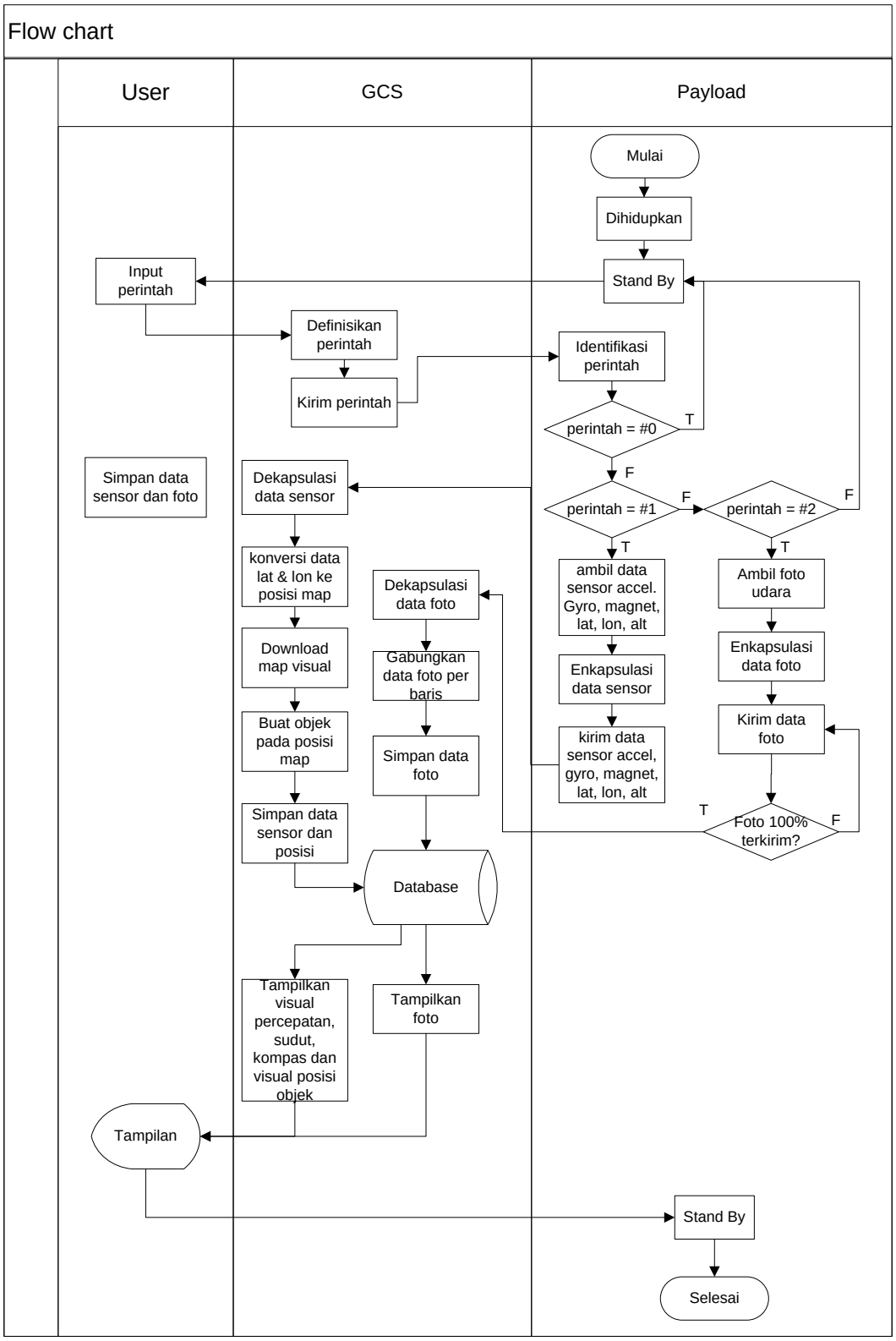
11. Perangkat lunak GCS mengolah data pengukuran 3 sensor dan GPS tersebut untuk kemudian didekapsulasi perbagian sensor.
12. Perangkat lunak GCS mengkonversi data *latitude*, *longitude* dan *altitude* menjadi posisi pada map.
13. Perangkat lunak GCS membuat objek visual *payload* pada posisi di mana *payload* tersebut berada.
14. Perangkat lunak GCS menyimpan data *history* sensor dan data posisi visual map ke dalam basis data SI GCS.
15. Perangkat lunak GCS menyajikan data tersebut kedalam bentuk visual yang dapat dimengerti oleh pengguna berupa visual map dan objek bergerak *payload*.
16. *Payload* mengambil gambar/foto udara 200x200 piksel dari ketinggian ±900 meter dpal.
17. *Payload* mengirim data foto per piksel kedalam format pengiriman data gambar yang telah ditentukan sebelumnya oleh panitia KOMURINDO.
18. Perangkat lunak GCS mendekapsulasi data foto udara tersebut kedalam unit data yang matang untuk disajikan.
19. Perangkat lunak GCS menggabungkan piksel data foto perbaris.
20. Perangkat lunak GCS menyimpan data foto ke dalam basis data SI GCS
21. Perangkat lunak GCS menampilkan data gambar foto perbaris yang diterima dari *payload*.
22. Apakah data foto telah terkirim 100%?, jika ya masuk ke proses 23, jika tidak kembali ke proses 17.

23. *Payload* dalam keadaan *standby* jatuh ke bumi.

4.1.3.1. Flowhart

Diagram alir atau *flowchart* digunakan untuk menggambarkan prosedur-prosedur yang ada untuk disajikan secara sederhana agar mudah dimengerti. Menggambarkan aliran dari satu proses ke proses lain atau dari suatu *method* ke *method* lain. Disamping itu diagram alir dapat membantu dimana letak kekurangan dari sistem yang ada dan bagaimana perbaikan sistem itu dapat dilakukan.

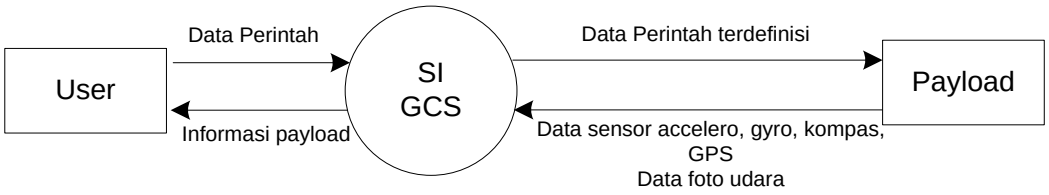
Berikut gambar diagram alir prosedur yang diusulkan :



Gambar 4. 1 Bagan alir sistem yang diusulkan

4.1.3.2. Diagram Konteks

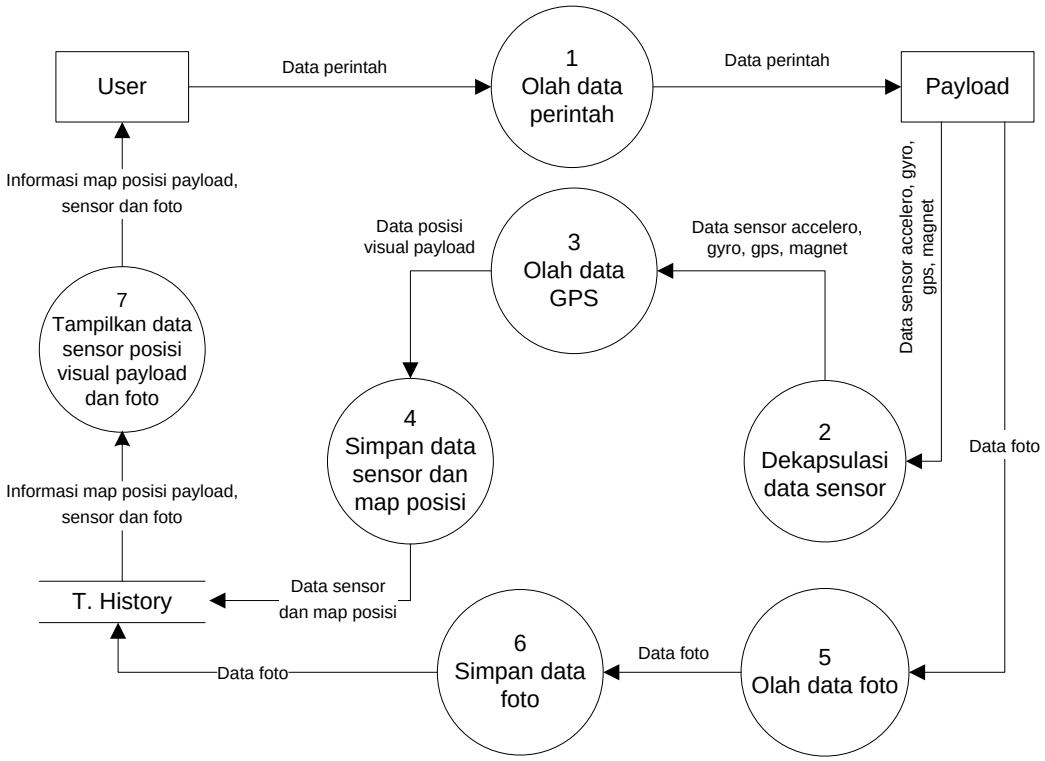
Diagram konteks dibuat untuk memperlihatkan interaksi proses sistem dengan lingkungan luar sistem. Pada diagram konteks akan terlihat arus data yang masuk ke dalam sistem dan arus data yang keluar dari sistem, arus data ini berasal dari entitas luar yang mempengaruhi sistem. Berikut gambaran diagram konteks yang diusulkan :



Gambar 4. 2 Diagram konteks GCS yang diusulkan

4.1.3.3. Diagram Arus Data

Diagram Arus Data (DAD) menggambarkan aliran data atau informasi secara lebih detail dan terperinci yang merupakan pengembangan dari diagram konteks. Berikut gambar DAD untuk GCS yang diusulkan :



Gambar 4. 3 DAD GCS yang diusulkan

4.1.3.4. Kamus Data

Kamus data menjelaskan data-data yang mengalir dalam sistem secara lengkap berdasarkan aliran data yang terdapat pada DAD. Berikut kamus data pada aplikasi SI GCS posisi ini :

1. Nama data : Data perintah

   Alias : -

   Aliran data : *User*->P1 , P1->*Payload*

   Volume : Setiap *user* mengklik tombol perintah

   Bentuk data : Virtual

   Struktur data :

2. Nama data : Data sensor *accelero, gyro, magnet,* GPS
   Alias : -
   Aliran data : Payload->P2 , P2->P3
   Volume : Setiap GCS menerima data sensor
   Bentuk data : Virtual
   Struktur data : Acel_X, Acel_Y, Acel_Z, Gyro_X, Gyro_Y, Gyro_Z, Magn_X, Magn_Y, Magn_Z, *Latitude, Longitude, Altitude*
3. Nama data : Data posisi visual *payload*
   Alias : -
   Aliran data : P3->P4
   Volume : Setiap GCS menerima data GPS
   Bentuk data : Visual map dan objek *payload*
   Struktur data : *Latitude, Longitude, Altitude*
4. Nama data : Data sensor dan map posisi
   Alias : -
   Aliran data : P4->T.History basis data
   Volume : Setiap aplikasi menyimpan data
   Bentuk data : Virtual
   Struktur data : ID_History, Tanggal, jam, Acel_X, Acel_Y, Acel_Z, Gyro_X, Gyro_Y, Gyro_Z, Magn_X, Magn_Y, Magn_Z, *Latitude, Longitude, Altitude*

5. Nama data : Data foto
   Alias : -
   Aliran data : *Payload*->P5 , P5->P6 , P6->T.History
   Volume : Setiap GCS menerima data foto
   Bentuk data : Data virtual foto berformat *.JPG
   Struktur data : Tanggal, jam, ID_Foto, Foto
6. Nama data : Informasi map posisi *payload,* sensor dan foto
   Alias : -
   Aliran data : T.History->P7 , P7->*User*
   Volume : Setiap user ingin menampilkan informasi *payload*
   Bentuk data : Grafik virtual, visual 3D, visual map dan foto virtual
   Struktur data : Tanggal, jam, Acel_X, Acel_Y, Acel_Z, Gyro_X, Gyro_Y, Gyro_Z, Magn_X, Magn_Y, Magn_Z, *Latitude, Longitude, Altitude,* Foto

4.1.4. Perancangan Basis Data

Untuk mendukung kebutuhan pengolahan data sistem, maka perlu ditentukan bagaimana rancangan basis datanya. Perancangan basis data ini dimaksudkan untuk mengidentifikasi kebutuhan-kebutuhan file basis data sistem yang dibutuhkan.

4.1.4.1. Normalisasi

Normalisasi merupakan suatu proses untuk mengubah suatu tabel kedalam beberapa tabel. Normalisasi biasa dipakai oleh perancang basis data untuk melakukan verifikasi terhadap tabel-tabel yang telah dibuat sehingga tidak menimbulkan masalah saat dihapus. Suatu tabel dikatakan berada dalam keadaan normal jika memenuhi kondisi-kondisi tertentu.

1. Bentuk tidak normal (*unnormal*)

   Bentuk tidak normal menghadirkan atribut-atribut apa saja yang ada pada kamus data, berikut semua atribut yang ada pada kamus data :

   Acel_X, Acel_Y, Acel_Z, Gyro_X, Gyro_Y, Gyro_Z, Magn_X, Magn_Y, Magn_Z, *Latitude, Longitude, Altitude, Latitude, Longitude, Altitude*, ID_History, Tanggal, jam, Acel_X, Acel_Y, Acel_Z, Gyro_X, Gyro_Y, Gyro_Z, Magn_X, Magn_Y, Magn_Z, *Latitude, Longitude, Altitude*, Tanggal, jam, ID_Foto, Foto, Tanggal, jam, Acel_X, Acel_Y, Acel_Z, Gyro_X, Gyro_Y, Gyro_Z, Magn_X, Magn_Y, Magn_Z, *Latitude, Longitude, Altitude*, Foto.

2. Bentuk normal pertama (1NF)

   Bentuk normal pertama terpenuhi apabila tidak ada atribut bernilai banak atau lebih dari satu atribut dengan domain bernilai sama seperti berikut :

   ID_History, Tanggal, jam, Acel_X, Acel_Y, Acel_Z, Gyro_X, Gyro_Y, Gyro_Z, Magn_X, Magn_Y, Magn_Z, *Latitude, Longitude, Altitude*, ID_Foto, Foto.

3. Bentuk normal kedua (2NF)

   Bentuk normal kedua terpenuhi jika pada sebuah tabel semua atribut yang tidak termasuk *primary key* memiliki ketergantungan fungsional pada *primary key* secara utuh.

   History = { ID_History*, Tanggal, jam, Acel_X, Acel_Y, Acel_Z, Gyro_X, Gyro_Y, Gyro_Z, Magn_X, Magn_Y, Magn_Z, *Latitude, Longitude, Altitude*, ID_Foto, Foto }

4. Bentuk normal ketiga (3NF)

   Bentuk normal ketiga terpenuhi apabila setiap atribut secara langsung bergantung dan ditentukan oleh *primary key*.

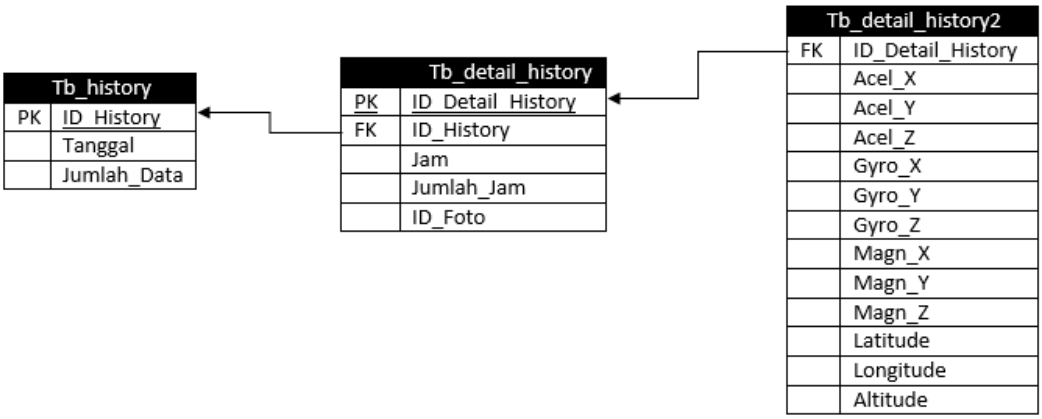
   History = { ID_History*, Tanggal }

   Detail_history = { ID_Detail_History*, Jam, ID_Foto, ID_History** }

   Detail_history2 = { ID_Detail_History**, Acel_X, Acel_Y, Acel_Z, Gyro_X, Gyro_Y, Gyro_Z, Magn_X, Magn_Y, Magn_Z, *Latitude, Longitude,Altitude* }

4.1.4.2. Tabel Relasi

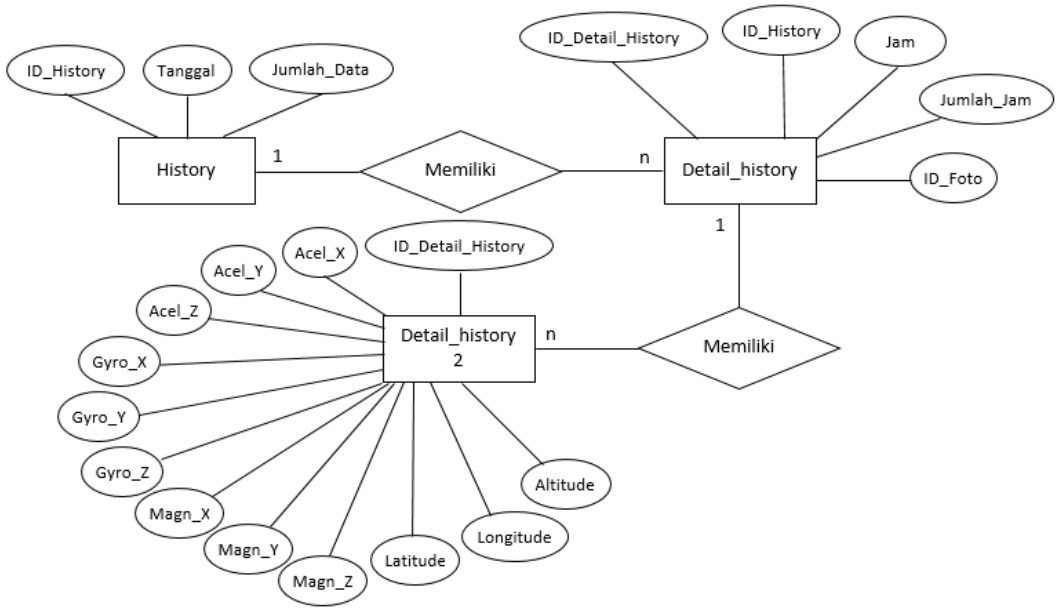
Menggambarkan hubungan antar file atau tabel yang satu dengan yang lainnya yang saling berhubungan. Berikut gambarannya :



Gambar 4. 4 Bagan relasi tabel

4.1.4.3. *Entity Relationship Diagram*

ERD merupakan gambaran hubungan antar entitas yang satu dengan yang lain yang saling berhubungan dan berinteraksi. Berikut *ERD* aplikasi GCS ini :



Gambar 4. 5 *Entity Relathionship Diagram*

4.1.4.4. Struktur File

Struktur file merupakan urutan isi atau data-data item yang ada pada file basis data. Rancangan struktur ini dimaksudkan untuk dapat melakukan kegiatan-kegiatan dalam pencarian data untuk mempermudah kerja sistem. Struktur file yang ada pada aplikasi GCS posisi *payload* ini sebagai berikut :

1. File History
    a. Nama file : Data_history
    b. Primary key : ID_History
    c. Jumlah field : 3

Tabel 4. 1 Struktur *file History*

No.	*Field Name*	*Type*	Size	Description
1.	ID_History	Varchar	6	ID *History*
2.	Tanggal	Date	-	Tanggal ambil data
3.	Jumlah_Data	Number		Jumlah data ambil

2. *File* Detail *History*
    a. Nama *file* : Detail_History
    b. Primary key : ID_Detail_History
    c. Jumlah *field* : 5

Tabel 4. 2 Struktur *file* detail_history

No.	*Field Name*	*Type*	Size	Description
1.	ID_Detail_History	Varchar	6	ID detail *history*
2.	ID_History	Varchar	6	Tanggal ambil data
3.	Jam	Number		Jam ambil data
4.	Jumlah_Jam	Number		Jumlah baris ambil
5.	ID_Foto	Varchar	6	ID Foto

3. File Detail *History2*

a. Nama *file* : Detail_History2

b. Primary key : -

c. Jumlah *field* : 13

Tabel 4. 3 Struktur file detail_history2

No.	*Field Name*	*Type*	Size	Description
1.	ID_Detail_History	Varchar	6	ID detail *history*
2.	Acel_X	Number		Nilai acel X
3.	Acel_Y	Number		Nilai acel Y
4.	Acel_Z	Number		Nilai acel Z
5.	Gyro_X	Number		Nilai gyro X
6.	Gyro_Y	Number		Nilai gyro Y
7.	Gyro_Z	Number		Nilai gyro Z
8.	Magn_X	Number		Nilai magn X
9.	Magn_Y	Number		Nilai magn Y
10.	Magn_Z	Number		Nilai magn Z
11.	Latitude	Varchar	10	Nilai *latitude*
12.	Longitude	Varchar	10	Nilai *longitude*
13.	Altitude	Varchar	10	Nilai *altitude*

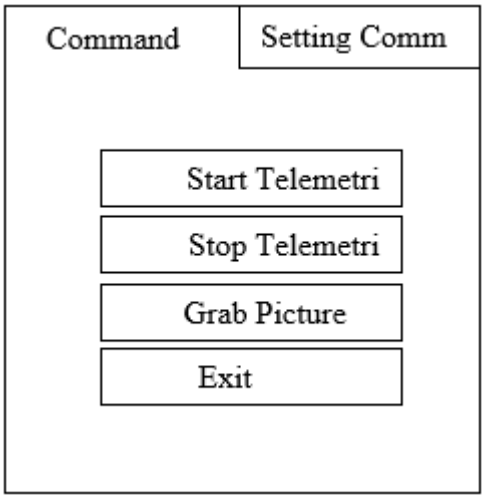
4.2. Perancangan Antarmuka

Sub bab perancangan antarmuka membahas mengenai tata desain perancangan *input* dan *output* yang akan digunakan pada sistem informasi GCS posisi global *payload*. Hanya ada satu *user* yang mengoperasikan sistem informasi GCS ini, berikut perancangan antarmukanya :

4.2.1. Perancangan *Input*

Untuk inputan dari user terdiri dari 2 tab pane yang berisi tombol-tombol perintah untuk *payload* dan inputan untuk pengaturan protokol komunikasi antar *ground control station* dan *payload*.

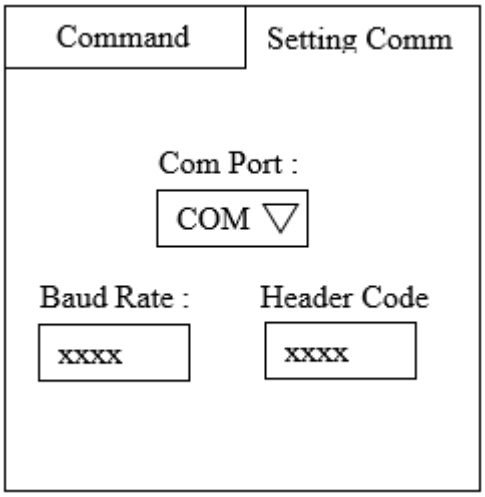
Berikut rancangan 2 tab pane tersebut :



Gambar 4. 6 Rancangan tampilan tab pane tombol-tombol perintah

Pada tab *command* diatas berikut penjelasan fungsi-fungsi tiap tombol inputan :

1. Start *Telemetri* : tombol perintah untuk menginstruksi payload mengambil dan mengirim data sensor-sensor yang ada, seperti sensor *accelerometer, gyroscope,* dan gps.
2. Stop *Telemetri* : tombol perintah untuk menginstruksi payload berhenti mengirimkan data.
3. *Grab Picture* : tombol perintah untuk menginstruksi payload mengambil dan mengirimkan foto udara dari ketinggian ±900 mter dpal.
4. *Exit* : tombol untuk menghentikan program.



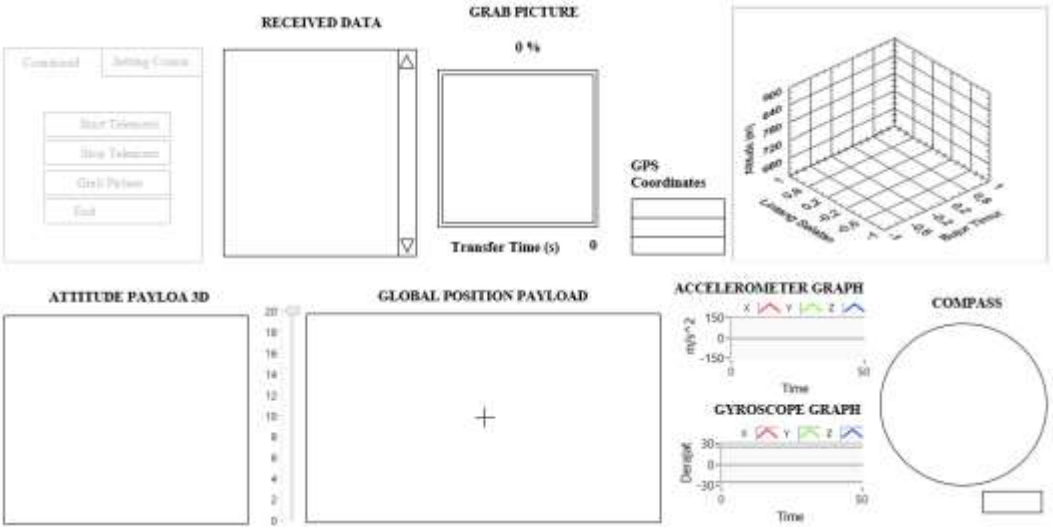
Gambar 4. 7 Rancangan tampilan tab pane *setting* komunikasi

Pada tab *Setting Comm* diatas berikut penjelasan dari fungsi masing-masing inputan :

1. *Com Port* : *combo box* untuk memilih pilihan *port* komunikasi yang digunakan.
2. *Header Code* : *input*an teks berupa *header code* atau *header* untuk data tiap tim yang telah ditentukan sebelumnya, guna membedakan data suatu tim dengan tim yang lainnya.
3. *Baud Rate* : *input*an berupa kecepatan *baud rate* atau kecepatan transfer data yang digunakan. Semakin tinggi *baud rate* maka semakin tinggi pula kecepatan transfer datanya.

4.2.2. Perancangan *Output*

Untuk *output* atau hasil dari proses sistem informasi terdiri dari tampilan data yang diterima dari *payload*, foto udara, grafik trayektori, sikap *payload* dalam bentuk 3D, tampilan visual posisi global yang merupakan hasil dari penelitian ini, tampilan sensor grafik *accelero*, *gyro* dan arah mata angin:



Gambar 4. 8 Rancangan tampilan output

Berikut masing-masing fungsi dari output diatas :

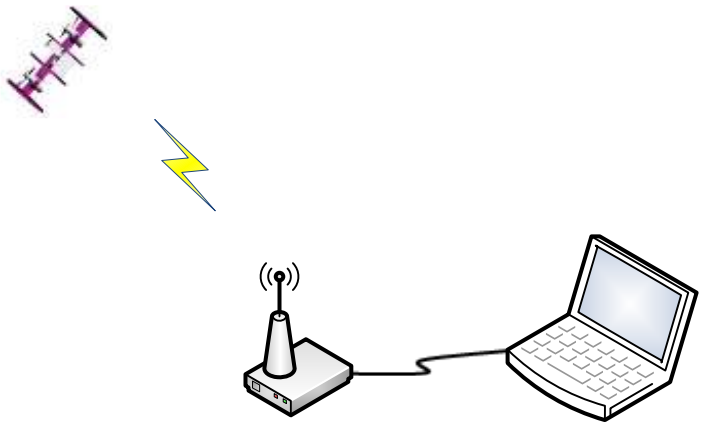
a. *Received Data* : berisi data-data mentah yang diterima dari *payload*.
b. *Grab Picture* : berisi data gambar/foto udara yang merupakan hasil dari pengolahan data foto yang sebelumnya dalam bentuk ASCII..
c. *GPS Coordinates* : berisi data posisi global *payload* lintang, bujur dan ketinggian diatas permukaan laut.
d. *Trajectory* : berisi data pergerakan roket dan atau *payload* dari titik peluncuran roket.
e. *Attitude Payload* 3D : berupa tampilan grafik sikap roket dan atau *payload.*
f. *Global Position Payload* : berupa tampilan visual objek bergerak *payload* dan posisi globalnya di permukaan bumi secara *real-time* dalam bentuk map.
g. *Accelerometer Graph* : berupa tampilan grafik garis perubahan sensor *accelerometer*.
h. *Gyroscope Graph* : berupa tampilan grafik garis perubahan sudut dari sensor *gyroscope*.

i. *Compass* : berupa tampilan arah mata angin dari sensor kompas.

4.3. Perancangan Arsitektur Jaringan

Arsitektur jaringan antar komponen sistem informasi ini terdiri dari :

1. Laptop/komputer : berfungsi sebagai *interface* yang menjembatani antara user dengan *payload*, sebagai objek display dari data yang dikirim *payload*.
2. *Modul receiver* : berupa rangkaian elektronika termasuk catu daya dan radio frekwensi, sebagai alat penerima data dan pengkonversi data agar dimengerti komputer.
3. *Payload* : berupa rangkaian elektronik, sensor-sensor termasuk GPS *receiver* dan kamera termasuk catu daya dan radio frekuensi.



Gambar 4. 9 Rancangan arsitektur jaringan

4.4. Implementasi

Kegiatan implementasi dilakukan dengan menerapkan rancangan sistem yang diusulkan yang sebelumnya telah dibuat. Penerapan sistem yang diusulkan ini

tentu saja terdapat perbaikan atau penambahan komponen baru yang dibutuhkan guna guna memberikan solusi terhadap masalah yang ada.

Implementasi adalah proses penerapan rancangan sistem informasi yang diusulkan berupa penerapan program aplikasi yang telah dibuat.

Implementasi merupakan tahap lanjut dalam pengembangan perangkat lunak, berupa kumpulan elemen-elemen yang telah dirancang kemudian diterjemahkan kedalam bahasa pemrograman untuk menghasilkan suatu produk perangkat lunak sesuai kebtuhan yang telah dianalisis sebelumnya.

4.4.1. Impelementasi Perangkat Lunak

Perangkat lunak pendukung minimum yang digunakan dalam pembuatan sistem informasi ini adalah sebagai berikut :

1. Sistem Operasi : Windows 7 profesional, berdasarkan percobaan penulis menjalankan program dibeberapa jenis sistem operasi.
2. LabVIEW & toolkit : sebagai perangkat lunak untuk pengembangan aplikasi berupa bahasa pemrograman visual.

4.4.2. Implementasi Perangkat Keras

Kebutuhan perangkat keras minimal yang dibutuhkan untuk penerapan sistem ini adalah sebagai berikut :

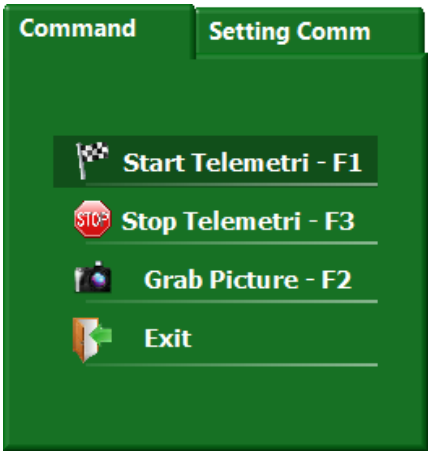
1. Processor : Intel dual core, berdasarkan percobaan penulis menjalankan program pada spesifikasi komputer dibawah komputer penulis.

2. RAM : 1GB DDR3, berdasarkan kebutuhan minimal dari perangkat lunak LabVIEW.
3. HardDrive : 320GB, untuk menyimpan data-data dan tools pendukung lainnya.
4. Perangkat antar muka diantaranya LCD/monitor, keyboard dan mouse.

4.4.3. Impelementasi Antarmuka

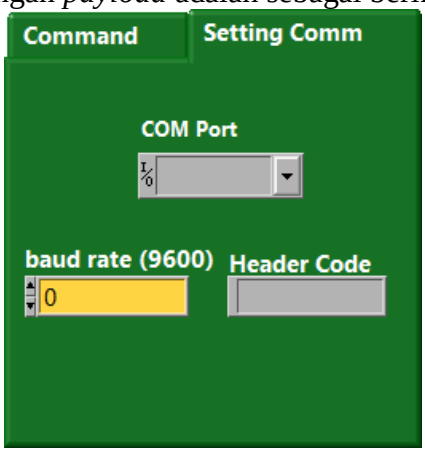
Setelah dilakukan perancangan antarmuka sebelumnya, maka selanjutnya dilakukan implementasi antarmuka, berdasakran penyesuaian terhadap rancangan antarmuka sebelumnya. Implementasi perangkat lunak pada penelitian ini adalah sebagai berikut:

Implementasi tab tombol-tombol perintah yang digunakan untuk memberikan perintah pada *payload*



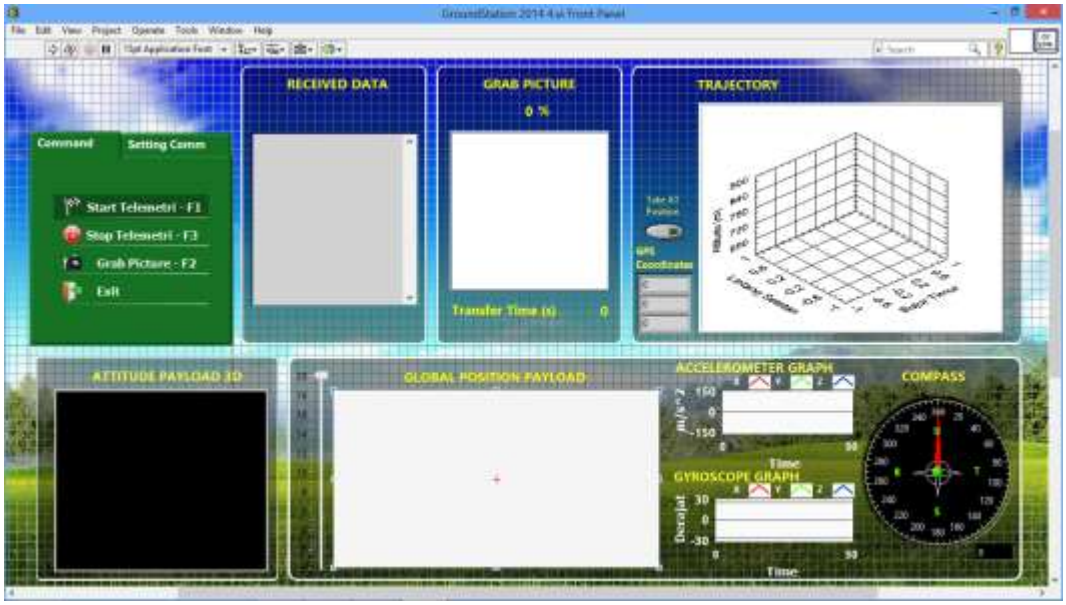
Gambar 4. 10 Implementasi tab tombol-tombol perintah

Implementasi tab masukkan pengaturan yang digunakan untuk berkomunikasi antara laptop/PC dengan *payload* adalah sebagai berikut :



Gambar 4. 11 Implementasi tab *setting comm*

Impelementasi *output* dari hasil pengolahan data *payload* termasuk data GPS beserta visualisasinya



Gambar 4. 12 Implementasi tampilan output

4.4.4. Penggunaan Program

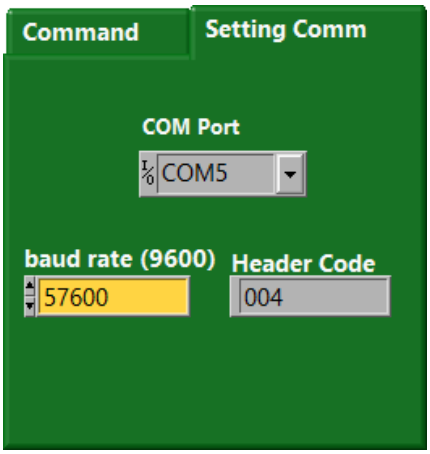
Berikut cara penggunaan program perangkat lunak antarmuka program yang telah dirancang sebelumnya, serta bagaimana penggunaan program aplikasi sistem informasi posisi global *payload.*

1. Buka aplikasi SI GCS



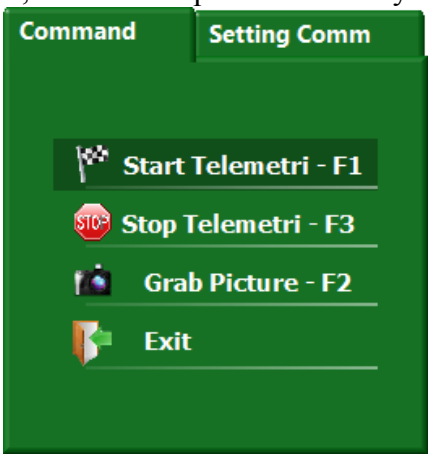
Gambar 4. 13 *Icon* program SI GCS

2. Pada tab "*Setting Comm"*, atur *port* komunikasi yang digunakan, *baudrate* atau kecepatan transfer data *bit persecond* serta *header code* yang digunakan untuk memfilter data berdasarkan identitas tertentu.



Gambar 4. 14 Pengaturan *setting* komunikasi

3. Pada tab "*Command*", klik tombol perintah sesuai yang dibutuhkan



Gambar 4. 15 Tombol-tombol perintah

4. Maka akan tampil data yang diterima beserta penyajiannya dalam beberapa bentuk, juga termasuk tampilan visualisasi posisi global *payload* berupa objek visual bergerak dimana *payload* tersebut berada.



Gambar 4. 16 Tampilan *output* keseluruhan

4.5. Pengujian

Dalam Jurnal Dwikatmono (2014 : 102) menurut Bambang Hariyanto, pengujian perangkat lunak adalah proses pemeriksaan atau evaluasi sisem atau komponen sistem secara manual atau otomatis untuk memverifikasi apakah sistem memenuhi kebutuhan-kebutuhan yang dispesifikasikan atau mengidentifikasi perbedaan-perbedaan antara hasil yang diharapkan dengan hasil yang terjadi. Awalnya, pengujian diartikan sebagai aktivitas yang hanya dilakukan setelah pengkodean (kode program selesai). Namun, pengujian sebenarnya dilakukan dalam skala yang lebih luas. Pengujian dapat dilakukan begitu spesifikasi

kebutuhan telah dapat didefinisikan. Evaluasi terhadap spesifikasi dan perancangan juga merupakan teknik dalam pengujian.

Teknik pengujian perangkat lunak yang penulis gunakan dalam penelitian ini adalah teknik pengujian *black box*. Menurut Al-Bahra bin Ladjamudin (2005:379), metode *black box* ini berfokus pada keperluan fungsional dari *software* yang memungkinkan untuk membuat himpunan kondisi input yang akan melatih seluruh syarat-syarat fungsional suatu program. Uji coba *black box* merupakan pendekatan yang melengkapi untuk menemukan kesalahan yang terjadi.

4.5.1. Rencana Pengujian

Pengujian perangkat lunak aplikasi pada Sistem Informasi *Ground Control Station* Posisi Global *payload* menggunakan teknik pengujian *black box,* dimana teknik pengujian ini berfokus pada keperluan fungsional, kinerja perangkat lunak, serta kesesuaian antara input yang diberikan dan output yang dihasilkan. Rencana pengujian aplikasi ini adalah sebagai berikut :

1. Pengujian unit, yaitu pengujian yang difokuskan pada suatu unit program secara tersendiri.
2. Pengujian penerimaan, yaitu pengujian yang meyakinkan bahwa sistem telah sesuai dengan kebutuhan dan menjawab masalah yang ada.

4.5.2. Kasus dan Hasil Pengujian

Berdasarkan rencana pengujian, maka studi kasus pengujian perlu dilakukan untuk mengetahui bagaimana hasilnya. Ada beberapa poin pengujian yang dilakukan penulis dalam menguji perangkat lunak Sistem Informasi GCS Posisi Global *payload* ini, yaitu sebagai berikut:

1. Pengecekkan fungsionalitas tombol "*Start Telemetri*"

   Pengujian ini berupa uji fungsional untuk pemberian perintah jarak jauh (*telecommand*) apakah *payload* dapat melakukan tugas sesuai dengan yang diperintahkan pengguna dari Sistem Informasi GCS ini, yaitu melakukan pengukuran jarak jauh data sensor dan GPS (*telemetri*). Berikut kasus dan hasil pengujian "*Start Telemetri*" :

Tabel 4. 4 Pengujian *Start Telemetri*

Kasus dan Hasil Uji			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik tombol "*start telemetri*"	Data perintah terkirim dan *payload* mengirimkan data-data pengukuran sensor serta data GPS	Aplikasi menerima data-data pengukuran sensor dan GPS	Diterima

2. Pengecekkan fungsionalitas tombol "*Stop Telemetri*"

   Pengujian ini berupa uji fungsional untuk pemberian perintah jarak jauh (*telecommand*) apakah *payload* dapat melakukan tugas sesuai dengan yang diperintahkan pengguna dari Sistem Informasi GCS ini, yaitu berhenti mengirimkan data-data pengukuran kemudian berada pada mode *standby*.

   Berikut kasus dan hasil pengujian "*Stop Telemetri*" :

Tabel 4. 5 Pengujian *Stop Telemetri*

Kasus dan Hasil Uji			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik tombol "*start telemetri*"	Data perintah terkirim dan *payload* mengirimkan data-data pengukuran sensor serta data GPS	Aplikasi menerima data-data pengukuran sensor dan GPS	Diterima

3. Uji lapangan kesesuaian antara posisi *hardware payload* pada faktanya, dan posisi objek *payload* yang divisualisasikan Sistem Informasi GCS Posisi Global.

Tabel 4. 6 Uji Lapangan

Kasus dan Hasil Uji			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Payload dibawa berjalan pada tempat terbuka	Aplikasi memvisualisasikan di mana *payload* yang dibawa berjalan tersebut berada, serta memvisualisasikan pergerakannya secara *real-time*	Aplikasi memvisualisasikan pergerakan *payload* secara *real-time*	Diterima

4.5.3. Kesimpulan Hasil Pengujian

Berdasarkan hasil pengujian dengan kasus dan hasil uji diatas, maka dapat ditarik kesimpulan bahwa perangkat lunak bebas dari kesalahan sintaks, bebas dari kesalahan *error-runtime* serta secara fungsional aplikasi dapat mengeluarkan hasil yang diharapkan sebelumnya.