ISSN 2460-1608 (Media Cetak) 2622-3244 (Media Online)

SISTEM KOMPRESI DATA PADA MUATAN ROKET MENGGUNAKAN KODE HUFFMAN

[1]Muhammad Luqman Bukhori, [2]Erwan Eko Prasetiyo

[1]Teknik Dirgantara, STTKD, [2]Aeronautika, STTKD

Abstrak

Muatan roket (payload) merupakan teknologi terapan dengan pemanfaatan kendali data jarak jauh. Teknologi ini mendukung fungsionalitas sistem pemantauan data ke Ground Control Station (GCS). GCS dijalankan oleh operator yang jaraknya jauh dari roket, sehingga operator dapat langsung melihat paket data yang dibutuhkan untuk keperluan pemantauan secara real-time. Walaupun sistem pemantauan ini dapat berjalan secara real-time, terdapat beberapa kendala yang terjadi yaitu kehilangan data akibat adanya keterlambatan data pada saat pengiriman data. Keterlambatan data dapat terjadi dikarenakan jumlah paket data dan transfer data saat pengiriman tidak seimbang. Jumlah paket data ini tidak diimbangi dengan kapasitas jaringan komunikasi data yang ditentukan. Untuk mengatasi hal tersebut terdapat salah satu teknik yang dapat diterapkan yaitu sistem kompresi data. Kompresi data yang diterapkan adalah metode kompresi lossless yang dibangun menggunakan kode Huffman. Hasil penelitian menunjukkan bahwa algoritma ini dapat melakukan kompresi data sebesar 41,25% dari data aslinya. Sistem roket dapat membaca dan mengirim sensor sebanyak 22 baris data dalam waktu 2 detik. Pada GCS, data yang diterima dan diolah dengan algoritma dekompresi Huffman menghasilkan data yang tervalidasi sebesar 19 baris data atau 86% data diterima.

Kata kunci: *Kompresi Data, Muatan Roket, Kode Huffman*

Abstract

Rocket payload is an applied technology with the utilization of remote data control. This technology supports the functionality of the data monitoring system to the Ground Control Station (GCS). GCS a run by operators who are far away from the rocket, so the operator can directly see the data packets needed for real-time monitoring purposes. Although this monitoring system can run in real-time, there is some obstacles that occur, namely data loss due to data delays at the time of data transmission. Data delays can occur due to the number of data packets and data transfers when delivery is unbalanced. The number of data packets is not balanced with the capacity of the specified data communication network. To overcome this there is one technique that can be applied, namely the data compression system. Data compression is a lossless compression method built using Huffman code. The results showed that this algorithm can perform data compression of 41.25% of the original data. The rocket system can read and send sensors as much as 22 lines of data in 2 seconds. At GCS, data received and processed with Huffman's decompression algorithm produces validated data of 19 lines of data or 86% of the data received.

Keywords: *Data Compression, Rocket Payload, Huffman Code*

Pendahuluan

Perkembangan teknologi penerbangan saat ini sangat berkembang pesat, khususnya perkembangan teknologi sistem muatan roket (payload) (Mudarris & Zain, 2020). Teknologi ini merupakan teknologi terapan yang diperuntukkan untuk pemanfaatan kendali data jarak jauh. Data yang dikendalikan ini berupa data-data yang diperlukan seperti posisi roket, GPS, tekanan, kecepatan udara, temperatur dan lain sebagainya. Cara mendapatkan semua data ini, memerlukan sebuah sistem pengiriman data secara nirkabel (wireless) (Yulkifli et al., 2016). Pengiriman data secara nirkabel ini banyak memiliki keunggulan, salah satunya adalah fleksibilitas pada perangkat karena tidak menggunakan kabel. Jika pengiriman data menggunakan kabel, maka perangkat akan mengalami kendala dalam melakukan aktivitas dan biaya yang besar untuk pembelian kabel.

Teknologi nirkabel ini dapat mendukung fungsi sistem pemantauan data dengan menggunakan sensor Inertial Measurement Unit (IMU). Sensor inersia ini dikombinasikan dengan perangkat transceiver

[1]Email Address : *mluqmanbukhori@gmail.com*
Received 3 November 2021, Available Online 30 Desember 2021

Radio Frequency (RF) dalam memadukan jaringan sensor nirkabel yang dapat dipakai pada muatan roket. Untuk menerapkan sensor nirkabel ini, dibutuhkan beberapa hal yang perlu diperhatikan khususnya penempatan posisi transmitter dan receiver, jenis protokol komunikasi data yang digunakan, serta kekuatan sinyal (Sani, 2017). Banyak penelitian yang membahas tentang tranceiver dan kekuatan sinyal (Mudarris & Zain, 2020; Sani, 2017; Yulkifli et al., 2016) namun belum banyak yang lebih dalam membahas tentang protokol komunikasi data pada roket.

Beberapa jenis protokol komunikasi data yang digunakan di antaranya Message Queuing Telemetri Transport (MQTT), Constrained Application Protocol (CoAP), Hypertext Transfer Protocol (HTTP) dan banyak lainnya (Oklilas et al., 2019). Komunikasi data ini sangat penting penggunaannya pada roket dikarenakan paket data yang dikirimkan oleh roket sangat besar. Terkadang jumlah paket yang besar tidak diimbangi dengan jaringan komunikasi data yang besar. Oleh sebab itu banyak kejadian data roket tidak dapat dimonitor secara sempurna oleh Ground Control Station (GCS) dikarenakan data yang berlebihan. GCS ini merupakan sistem pemantauan roket terhadap pembacaan kondisi muatan roket, dimana operator dapat melihat secara real-time data-data roket yang terbaca.

Kompresi data (Rustamaji et al., 2015) merupakan proses redundansi dari suatu informasi dengan cara memadatkan isi data file sehingga ukurannya menjadi lebih kecil dan menjaga kualitas data tetap sesuai dengan aslinya. Teknik kompresi ini dapat dilakukan terhadap data teks/biner, gambar, audio, dan video. Terdapat dua metode utama kompresi yaitu lossless dan lossy. Metode lossy adalah kompresi yang membuat data hilang selama proses kompresi sehingga kualitas data yang dihasilkan lebih rendah dari data aslinya, sedangkan metode lossless tidak menghilangkan data selama proses kompresi sehingga kualitas data sama seperti aslinya.

Pada penelitian ini, roket menerapkan sistem kompresi data lossless untuk media transceiver RF pada muatan roket yang akan dikirim ke GCS untuk sistem pemantauan data secara real-time. Metode ini diharapkan dapat mencegah adanya terjadinya keterlambatan data saat proses pengiriman data berlangsung akibat banyaknya jumlah data yang dikirim oleh muatan roket ke GCS.

Tinjauan Pustaka dan Pengembangan

Data Rate or Bit Rate

Data rate (Besaran Data) adalah ukuran kecepatan bit data dalam proses transmisi yang dihitung dalam satuan bit per detik (bps) (Prasetyo Adi & Kitagawa, 2020). Besaran data sangat terkait erat pada saat melakukan transfer data. Besaran memiliki beberapa definisi sebagai berikut (K = 1024, k = 1000, B = Bytes, dan b = bits). Sebagai contoh satuan 1 KB (1 kilobyte) adalah 1024 Bytes. *Bit Rate* atau sering disebut dengan data transfer, biasa ditulis dalam satuan 1 kbps atau 1000 bps.

Time on Air (ToA)

Time on Air (ToA) adalah satuan waktu yang digunakan oleh Radio Transmisi untuk mengirimkan data dari transmitter (Tx) ke receiver (Rx) (Prasetyo Adi & Kitagawa, 2020).

Bir ErrorRate (BER)

Bit error rate atau bit error rasio adalah angka dari jumlah bit dalam jaringan transmisi. Contoh kasus pada LoRa 915 MHz mempunyai jumlah error bit sebesar 30% dari 10 data bit yang dikirim terdapat 3 data bit yang tidak terkirim (Prasetyo Adi & Kitagawa, 2020). Persamaan yang dapat dipakai untuk menghitung BER dapat dilihat pada persamaan 1.

$$BER = \frac{N\ error}{N\ bits} \quad (1)$$

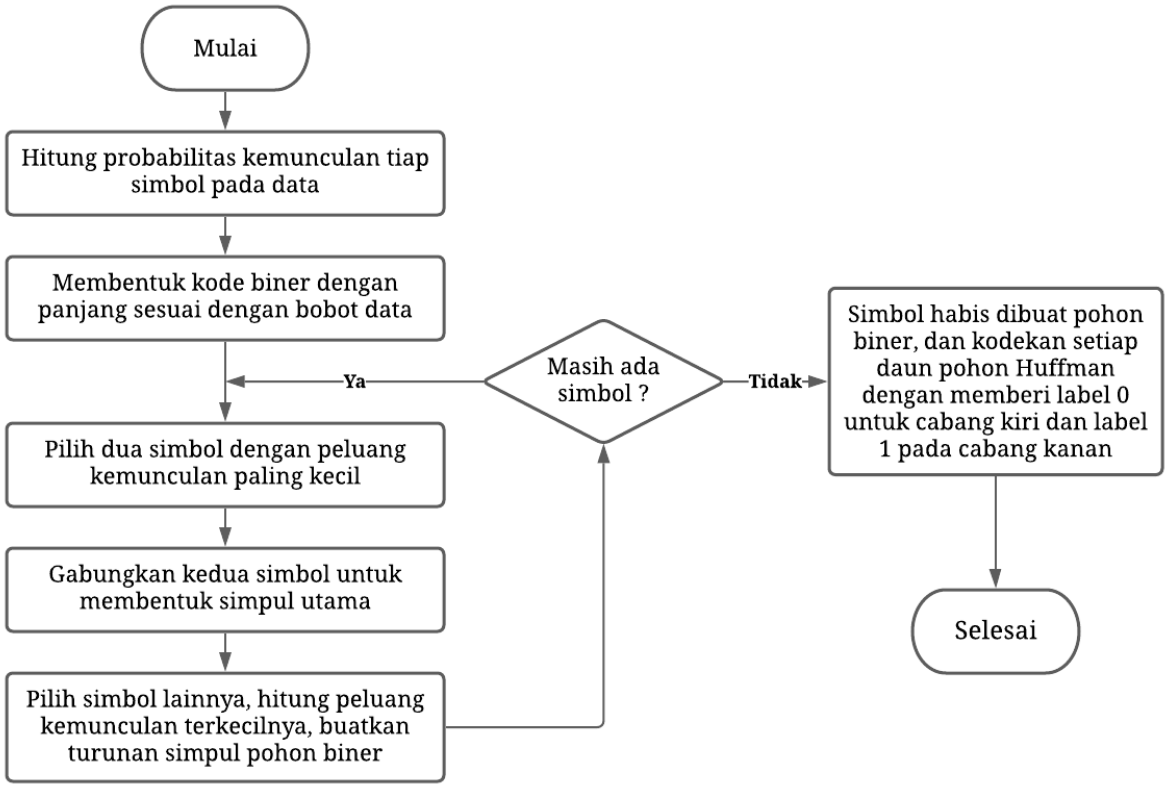
Algoritma Kode Huffman

Kode Huffman ini merupakan main core algoritma yang dipakai oleh beberapa aplikasi seperti ZIP, 7-ZIP, dan WinRAR. Teknik kompresi dengan algoritma ini cukup efisien untuk mempertahankan data atau informasi yang dikirimkan oleh pengirim dengan ukuran yang relatif kecil sehingga mempercepat data terkirim secara real-time. Perancangan sistem pada roket mempunyai format pengiriman data seperti pada Tabel 1. Data yang terbaca oleh raspberry berupa data text/biner sehingga banyak baris data yang diperoleh untuk dikirimkan ke GCS.

Tabel 1 Format pengiriman data

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte …	Byte n
23	Data Sensor (Accelero, Gyro, Magneto)			2C	Data sensor lainnya	2A

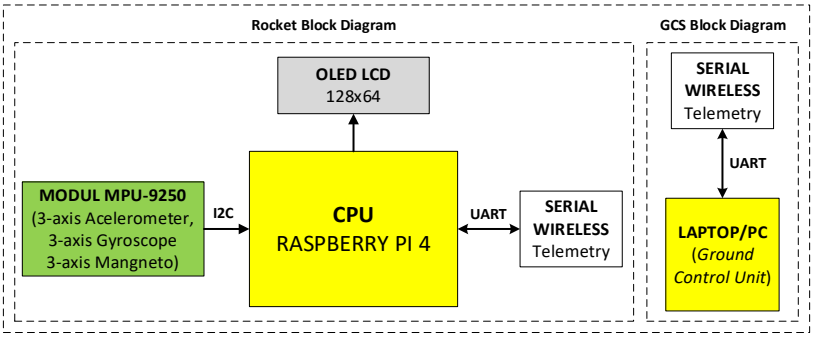
Tabel 1 menunjukkan format pengiriman data yang telah dikemas untuk dikompresi terlebih dahulu sebelum dikirim. Byte pertama yaitu 23 (karakter #), kemudian kapasitas 3 byte untuk masing-masing penampungan data sensor dari accelero, gyro, dan magneto. Data byte 2C merupakan karakter dari "," (koma) digunakan sebagai pemisah antar masing-masing data sensor. Byte ke-n merupakan byte terakhir dengan kode 2A yang merupakan karakter dari "*" (bintang). Alur algoritma yang digunakan untuk dapat membentuk kode Huffman dapat dilihat pada Gambar 1.



Gambar 1 Diagram algoritma kode Huffman

Metode Penelitian

Perancangan sistem yang diinginkan terdapat dua hal yaitu sistem yang berada pada roket dan sistem yang digunakan untuk pemantauan roket. Kedua sistem ini dirancang secara terpisah namun saling keterkaitan seperti yang ditunjukkan pada Gambar 2.

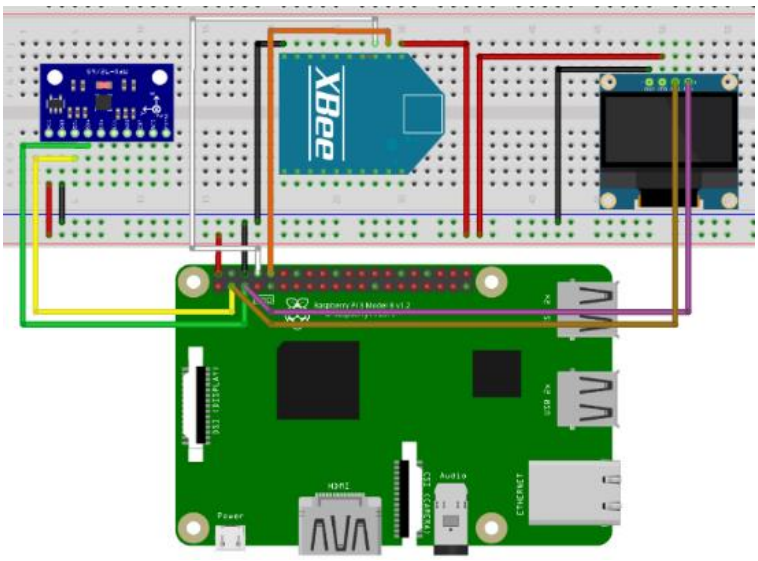


Gambar 2 Diagram blok sistem keseluruhan

Gambar 2 menunjukkan bahwa terdapat dua buah perancangan sistem yaitu untuk sistem roket dan sistem GCS. Kedua sistem dapat terhubung untuk melakukan komunikasi data secara langsung menggunakan bantuan telemetri dari tranceiver RF. Modul transceiver yang digunakan adalah modul 3DR telemetri 915MHz 500mW seperti yang terlihat pada Gambar 3. Modul ini mempunyai frekuensi kerja yang tetap sebesar 915MHz. Sistem yang diterapkan untuk telemetri ini adalah sistem komunikasi full-duplex dengan TDM adaptif.

Gambar 3 Radio telemettri 915 Mhz

Selain sistem komunikasi data, yang terpenting adalah sistem pada pembacaan sensor pada roket. Pembacaan sensor digunakan beberapa perangkat keras seperti yang ditunjukkan pada Gambar 4.

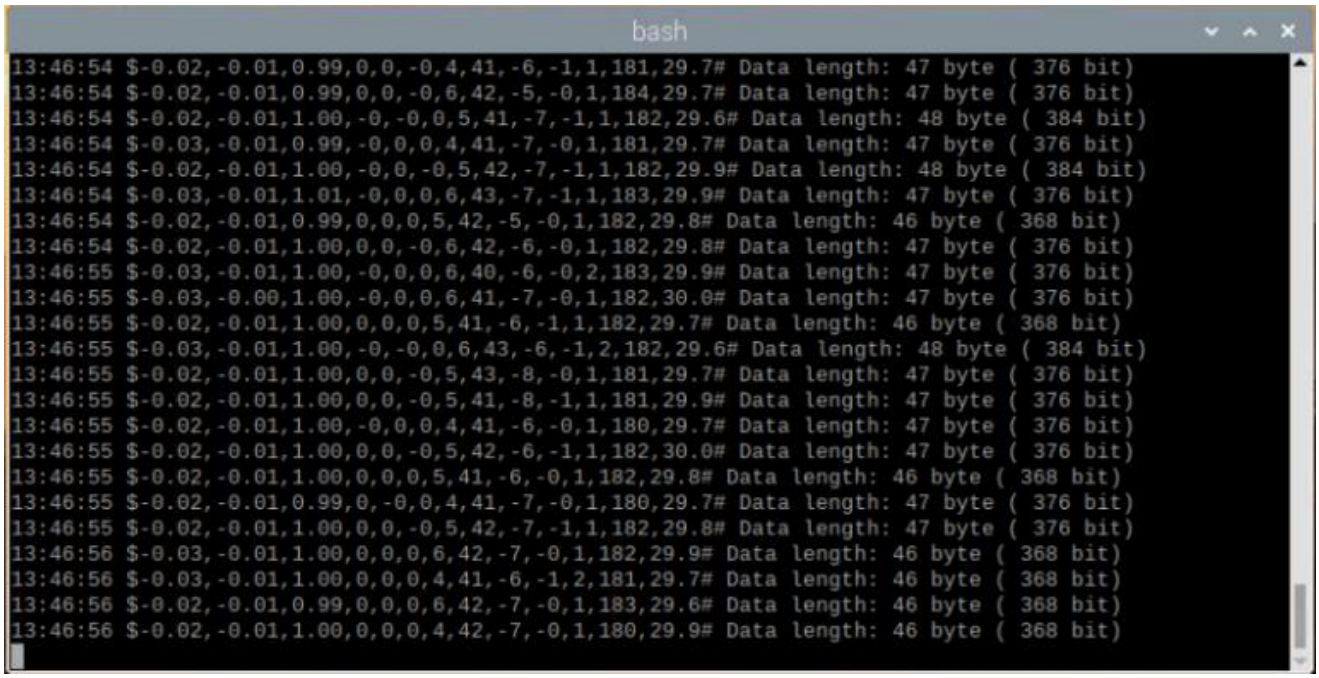


Gambar 4 Skema rangkaian elektronik roket

Komponen utama yang tertera pada Gambar 4 ini meliputi Raspberry Pi 4 type B, sensor IMU/MPU-9250 yang digunakan untuk membaca data temperature; accelerometer; gyroscope; magnetometer; pressure, dan 3DR Telemetri Radio yang merupakan media komunikasi dari roket dengan GCS. Raspberry merupakan sebuah mini PC yang dapat digunakan untuk mengolah data yang besar seperti data sensor, gambar, video, dan suara. Data-data yang telah diolah ini akan dikirim ke GCS melalui radio telemetri sebagai media komunikasi datanya. Sebelum mengirimkan data, sistem akan melakukan kompresi data terlebih dahulu menggunakan teknik kompresi lossless menggunakan metode kode Huffman. Setelah data terkompresi maka data akan dikirimkan ke GCS. Setelah GCS menerima data, maka akan dilakukan dekompresi terlebih dahulu sebelum data ditampilkan ke GUI monitor roket.

Hasil dan Pembahasan

Data pengiriman dari sistem roket yang dikendalikan oleh Raspberry berupa data sensor seperti accelerometer, gyroscope, magnetometer, roll, pitch, yaw, dan temp. Data dikemas satu baris string seperti yang terlihat pada Gambar 5.



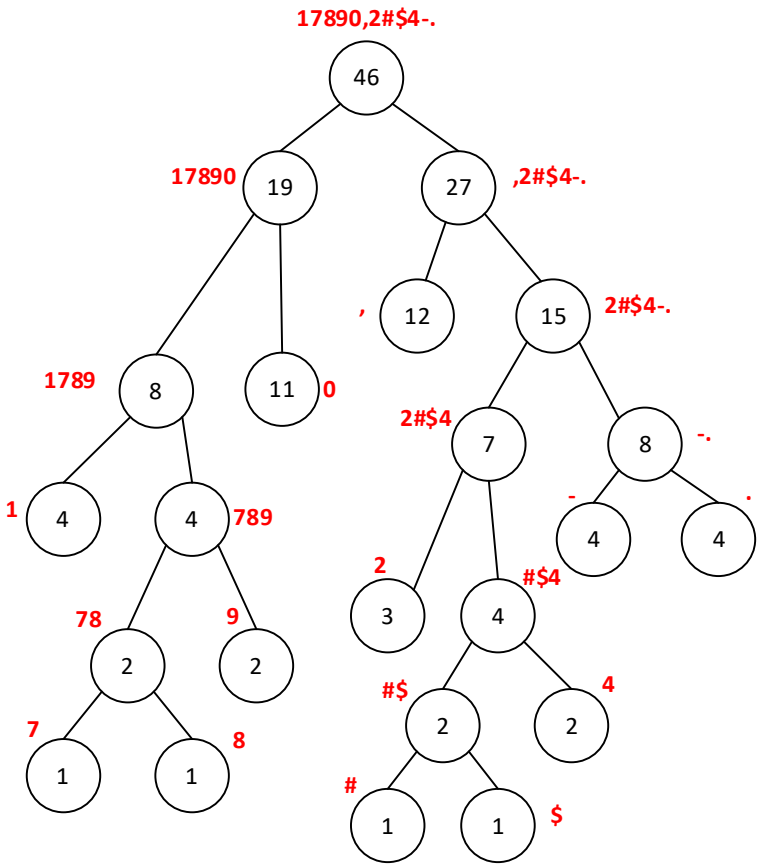
Gambar 5 Hasil Pengemasan Data Sensor dalam Bentuk String

Gambar 5 memperlihatkan bahwa dalam satuan waktu data sensor telah dikemas dalam satuan baris dengan awalan "$" dan akhiran "#", sehingga data dapat dibaca dengan benar nanti di GCS. Setiap baris data yang terbaca mempunyai panjang data yang berbeda-beda seperti yang diperlihatkan pada baris akhir 13:46:56 $-0.02,-0.01,1.00,0,0,0,4,42,-7,-0,1,180,29.9# Data length: 46 byte (368 bit).

Kode 13:46:56 adalah pembacaan waktu saat ini, kemudian $xxx# merupakan data sensor, dan yang terakhir data length: 46 byte (368 bit) merupakan jumlah besaran data yang dikirim melalui tranceiver RF. Panjang data masih tergolong data pendek, namun jika panjang data sampai 100 byte lebih akan memakan waktu. Diperlukan metode kompresi data untuk mengatasi permasalahan ini.. Kompresi data yang diterapkan adalah menggunakan metode loseless dari kode Huffman. Langkah awal adalah melakukan probabilitas terhadap kemunculan karakter yang paling kecil dan memasukkannya ke dalam node pohon Huffman dengan menggunakan algoritma berikut.

```
# frequecy char on data
word_count = []
kata = []
for kalimat in string:
    if kalimat not in word_count:
        freq = string.count(kalimat)
        word_count.append(freq)
        word_count.append(kalimat)
        kata.append(kalimat)
# Level node for Huffman tree probability
nodes = []
while len(word_count) > 0:
    nodes.append(word_count[0:2])
    word_count = word_count[2:]
nodes.sort()
pohon_Huffman = []
pohon_Huffman.append(nodes)
```

Penerapan algoritma tersebut dapat memecah karakter-karakter data dalam satuan node pada pohon Huffman. Hasil yang diperoleh dengan penerapan algoritma ini adalah sebagai berikut.



Gambar 6 Hasil node data pohon Huffman

Gambar 6 menunjukkan bahwa data hasil kompresi untuk nilai 46 byte adalah karakter (17890,2#$4-.). Data ini diperoleh dari pembentukan pohon Huffman dari pencabangan nilai byte terkecil (minheap) menuju ke atas (maxheap). Terdapat 11 level pembentukan pohon Huffman. Dari setiap cabang node diberikan alokasi bit 0 dan 1 dengan cara cabang kiri mendapatkan nilai 0 dan cabang kanan mendapatkan nilai 1. Setelah semua mendapatkan alokasi bit dari keseluruhan cabang, maka dapat dibuatkan binary code untuk setiap karakter dengan algoritma berikut.

```
# Make binary code for every word
binary_kata = []
if len(kata) == 1:
    char_code = [kata[0],"0"]
    binary_kata.append(char_code * len(string))
else:
    for kt in kata:
        charcode = ""
        for node in ceklist:
            if len(node) > 2 and kt in node[1]:
                charcode = charcode + node[2]
        char_code = [kt, charcode]
        binary_kata.append(char_code)
```

Binary code ini digunakan untuk menyatakan data dalam bentuk biner. Kode biner sangat mudah untuk kompresi dan ekstraksi karena konstruksinya yang hanya menggunakan dua kode saja yaitu 0 dan 1. Setiap karakter yang di dapatkan pada pohon Huffman sebelumnya dengan kompresi Huffman yaitu data karakter (17890,2#$4-.) harus disusun menjadi sebuah kode biner dari masing-masing karakter. Hasil kode biner yang didapatkan pada masing-masing karakter dapat dilihat pada Tabel 2.

Tabel 2 Hasil kode biner pohon Huffman

Karakter	Kode Biner
$	110101
-	1110
0	01
.	1111
2	1100
,	10
1	000
4	11011
7	00100
8	00101
9	0011
#	110100

Tabel 2 memperlihatkan deret biner yang dapat dituliskan untuk nilai hasil kompresi pada data asli yang sebelumnya berjumlah 368 bit. Setelah data berhasil dikompresi nilai data yang terkompres sebesar 142 bit. Rasio kompresi yang diperoleh sebesar 38% dari data aslinya yang dapat dilihat pada Tabel 3 perbandingan data asli dengan data kompresi berdasarkan panjang nilai kode binernya.

Tabel 3 Perbandingan data asli dan data kompresi

	Kode Biner	Data
Data Asli	10010010110111000010111011000011001010110010110111000010111011000011000110110011000110111011000011000010110011000010110011000010110011000010110011010010110011010011001010110010110111011110110010110111000010110011000110110011000111100011000010110011001011100110111011100110001	368 bit
Data Kompresi	1101011110011111011100101110011111010001000011110101100110011001101101110110111100101110001001011100110000100000010101101100001111110011110100	142 bit

Berdasarkan Tabel 3, dapat dilihat bahwa data asli yang jumlah data sebelumnya terbaca sebesar 368 bit dapat dikompres menggunakan kode Huffman menjadi data yang besarnya 142 bit saja. Terlihat untuk deret kode biner pada data asli lebih banyak dibandingkan dengan deret kode biner pada data yang telah dikompresi. Dengan demikian kode Huffman ini sangat membantu dalam melakukan pemampatan data atau kompresi data dengan baik sebesar 38% untuk satu baris data yang terbaca.

Percobaan selanjutnya adalah dengan melakukan uji coba untuk melihat berapa banyak jumlah baris data yang dapat dikirimkan oleh sistem roket dalam waktu 1 detik dengan menggunakan data hasil kompresi. Jumlah baris data ini dapat dilihat pada Tabel 4. Terlihat bahwa dalam waktu 1 detik terdapat 11 baris data sensor yang dapat dibaca dan dikirim ke GCS dari sistem roket yang telah dibuat. Pengiriman data secara real-time terhadap setiap data yang dibaca pada sistem roket.

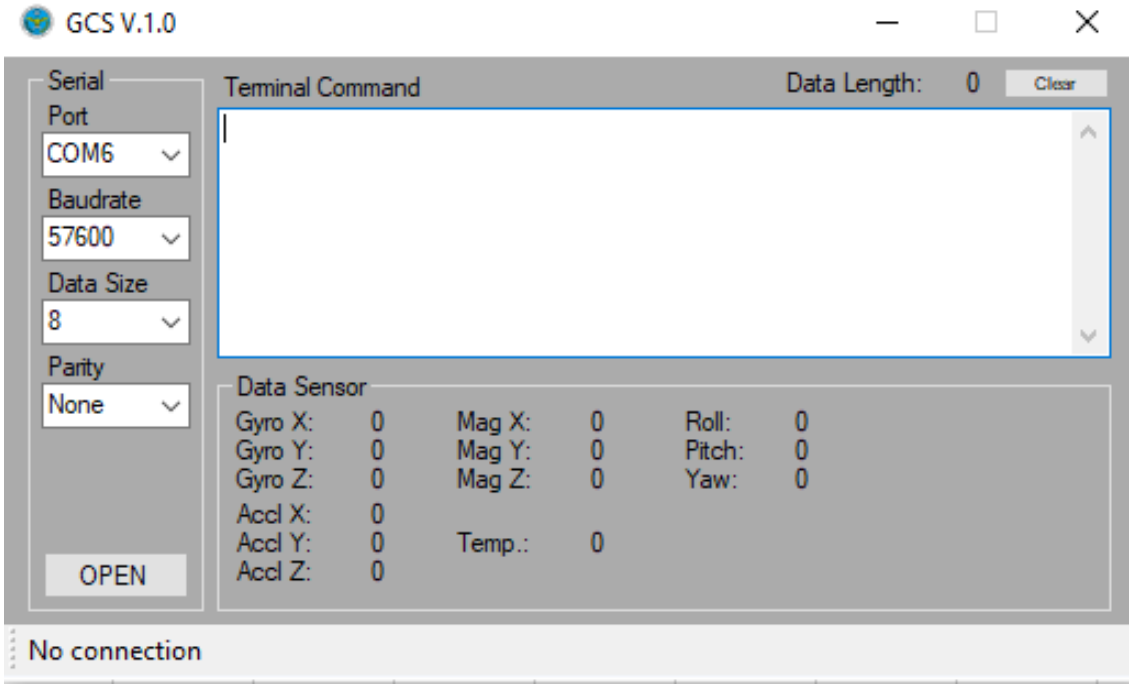
Tabel 4 Jumlah baris data dalam waktu 2 detik

No	Waktu	Baris Data Sensor	Panjang Data		Rasio
			Asli	Kompres	Kompres
1	20.52.02	$0.04,-0.08,1.02,1,1,-0,30,-12,-23,-4,-2,284,34.4#	400 bit	162 bit	40,50%
2	20.52.02	$0.03,-0.02,1.00,-1,0,-0,31,-11,-24,-1,-2,283,34.3#	408 bit	160 bit	39,22%
3	20.52.02	$0.03,-0.02,1.00,-1,-0,0,31,-11,-23,-1,-2,283,34.1#	408 bit	160 bit	39,22%
4	20.52.02	$0.02,-0.05,0.99,0,-0,-0,31,-12,-23,-3,-1,284,34.3#	408 bit	171 bit	41,91%
5	20.52.02	$0.02,-0.04,0.99,-1,-1,-0,33,-11,-22,-2,-1,282,34.2#	416 bit	170 bit	40,87%
6	20.52.02	$0.04,-0.05,0.99,-0,-0,-0,31,-12,-22,-3,-2,284,34.1#	416 bit	174 bit	41,83%
7	20.52.02	$-0.00,-0.02,0.98,-1,-1,-0,30,-12,-23,-1,0,285,34.2#	416 bit	172 bit	41,35%
8	20.52.02	$0.01,-0.02,0.99,0,-1,-0,32,-12,-22,-1,-0,284,34.3#	408 bit	167 bit	40,93%
9	20.52.02	$-0.01,-0.04,0.97,0,-1,-0,32,-11,-23,-2,0,282,34.4#	408 bit	170 bit	41,67%
10	20.52.02	$-0.03,0.01,0.97,-0,-1,-1,30,-11,-23,1,2,287,33.9#	400 bit	164 bit	41,00%
11	20.52.02	$-0.04,0.02,0.97,-1,-1,-0,31,-12,-24,1,3,287,34.3#	400 bit	170 bit	42,50%
12	20.52.03	$-0.03,-0.01,0.97,-0,-0,-0,31,-12,-23,-0,2,287,34.3#	416 bit	172 bit	41,35%
13	20.52.03	$-0.05,0.03,0.98,-1,-0,0,33,-11,-22,2,3,285,34.3#	392 bit	165 bit	42,09%
14	20.52.03	$-0.05,0.03,0.98,-1,-0,0,31,-12,-23,2,3,288,34.3#	392 bit	164 bit	41,84%
15	20.52.03	$-0.06,0.04,0.98,-0,-0,0,31,-10,-23,2,3,285,34.1#	392 bit	167 bit	42,60%
16	20.52.03	$-0.05,0.05,0.97,-1,-0,-0,31,-11,-23,3,3,288,34.0#	400 bit	171 bit	42,75%
17	20.52.03	$-0.04,0.02,0.98,-0,-0,-0,32,-11,-23,1,2,285,34.1#	400 bit	168 bit	42,00%

No	Waktu	Baris Data Sensor	Panjang Data		Rasio
			Asli	Kompres	Kompres
18	20.52.03	$-0.05,0.01,0.98,-0,-0,0,32,-11,-23,1,3,285,34.2#	392 bit	166 bit	42,35%
19	20.52.03	$-0.04,0.05,0.97,-1,-0,0,31,-11,-23,3,3,288,34.2#	392 bit	169 bit	43,11%
20	20.52.03	$-0.04,0.01,0.99,-1,-0,-0,32,-12,-23,0,3,286,34.1#	400 bit	169 bit	42,25%
21	20.52.03	$-0.05,0.01,0.98,0,-0,-0,32,-13,-22,0,3,288,34.2#	392 bit	162 bit	41,33%
22	20.52.03	$-0.04,0.01,0.99,-0,-0,0,32,-12,-24,1,2,286,34.3#	392 bit	165 bit	42,09%
			Rata-rata Kompresi		**41,58%**

Tabel 4 memperlihatkan jumlah banyak baris data yang dapat dibaca secara real-time menggunakan algoritma kompresi Huffman. Terdapat 22 baris data yang terbaca dalam waktu 2 detik. Setiap baris yang terbaca akan dilakukan kompresi data secara langsung untuk dikirimkan ke GCS supaya data dapat terkirim secara real-time. Hasil kompresi dari setiap baris data yang diperoleh, didapatkan rata-rata sebesar 41,58% dari 22 baris data.

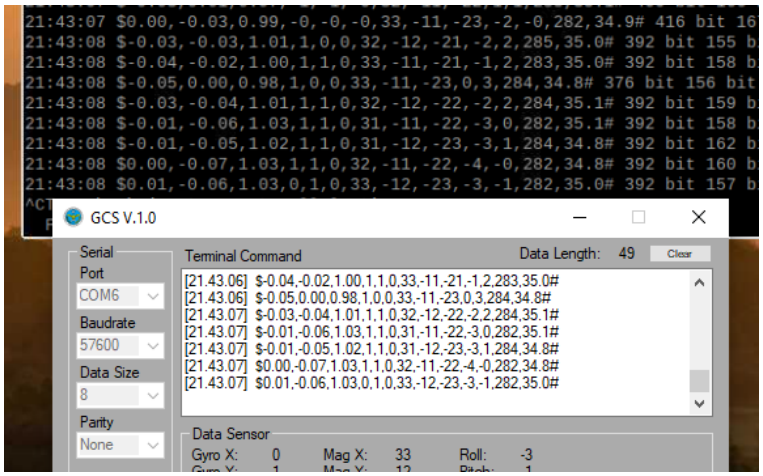
Pengujian data yang diterima oleh GCS dapat dilihat secara langsung menggunakan GUI yang telah dibuat khusus dengan menggunakan algoritma dekompresi Huffman. Tampilan GUI yang dibuat dapat dilihat pada Gambar 7.



Gambar 7 Tampilan GUI pada GCS

Terlihat pada Gambar 7 menunjukkan terdapat beberapa komponen yaitu Serial, Terminal Command, Data Length, dan Data Sensor. Serial digunakan untuk melakukan pengaturan awal dalam melakukan penyambungan Transceiver RF dari roket ke GCS. Terminal Command merupakan tampilan data hasil ekstraksi (dekompresi) data yang telah dikirim oleh roket secara real-time. Data Sensor merupakan tampilan real-time data sensor yang telah berhasil diterima oleh GCS untuk dapat melihat pemantauan nilai sensor-sensor secara langsung.

Setelah dilakukan pengujian pengiriman data dari roket ke sistem GUI pada GCS data yang dikirim dan data yang diterima berhasil diterima dengan baik oleh GCS seperti yang diperlihatkan pada Gambar 8.



Gambar 8 Hasil data kirim dan data terima pada GCS

Terlihat pada Gambar 8, nilai baris data antara yang dikirim dan data yang diterima oleh GCS dengan sebesar 7 baris data berhasil diterima dengan baik, tanpa kehilangan data dengan dibuktikan hasil yang sama. Waktu yang tertera pada sistem roket dan sistem GCS terdapat perbedaan waktu ini dikarenakan masing-masing menggunakan pengaturan waktu pada UTC Operating System (OS) pada masing-masing perangkat. Walaupun terdapat perbedaan waktu namun secara perhitungan masih tetap sama, untuk lebih jelasnya dapat melihat total data pengiriman dan penerimaan pada Tabel 5.

Tabel 5 Validasi data pengiriman dan data penerimaan

Data ke-	Baris Data Sensor Sistem Roket (Sender)	Baris Data Sistem GCS (Receiver)	Validasi
1	$0.02,-0.05,1.01,-1,-0,0,31,-12,-23,-3,-1,284,35.0#	$0.02,-0.05,1.01,-1,-0,0,31,-12,-23,-3,-1,284,35.0#	OK
2	$-0.01,0.00,0.96,-1,-1,-0,32,-12,-22,0,1,286,35.0#	$-0.01,0.00,0.96,-1,-1,-0,32,-12,-22,0,1,286,35.0#	OK
3	$-0.06,0.06,0.96,-1,-0,0,31,-12,-23,3,4,290,35.0#	$-0.06,0.06,0.96,-1,-0,0,31,-12,-23,3,4,290,35.0#	OK
4	$-0.00,-0.02,0.98,-1,-1,-0,31,-12,-23,-1,0,285,34.8#	$-0.00,-0.02,0.98,-1,-1,-0,31,-12,-23,-1,0,285,34.8#	OK
5	$0.01,-0.04,1.00,-0,-0,-0,31,-12,-22,-2,-1,284,35.0#	$-0.00,-0.02,0.98,-1,-1,-0,31,-12,-23,-1,0,285,34.8#	-
6	$0.01,-0.04,1.00,-0,-0,0,32,-12,-23,-3,-1,283,35.0#	$0.01,-0.04,1.00,-0,-0,0,32,-12,-23,-3,-1,283,35.0#	OK
7	$0.03,-0.08,1.03,0,0,0,32,-11,-23,-5,-2,281,35.1#	$0.03,-0.08,1.03,0,0,0,32,-11,-23,-5,-2,281,35.1#	OK
8	$0.02,-0.05,1.00,-0,-0,-0,31,-12,-22,-3,-1,285,35.0#	$0.02,-0.05,1.00,-0,-0,-0,31,-12,-22,-3,-1,285,35.0#	OK
9	$0.01,-0.05,0.99,-0,-0,-1,32,-11,-21,-3,-1,283,34.9#	$0.01,-0.05,0.99,-0,-0,-1,32,-11,-21,-3,-1,283,34.9#	OK
10	$-0.00,-0.02,0.98,-1,-1,-0,33,-11,-22,-1,0,283,34.9#	$0.01,-0.05,0.99,-0,-0,-1,32,-11,-21,-3,-1,283,34.9#	-
11	$0.02,-0.04,1.00,-0,-1,-0,32,-10,-21,-2,-1,281,35.0#	$0.02,-0.04,1.00,-0,-1,-0,32,-10,-21,-2,-1,281,35.0#	OK
12	$-0.00,-0.02,0.99,-1,-1,-0,33,-11,-23,-1,0,283,35.0#	$-0.00,-0.02,0.99,-1,-1,-0,33,-11,-23,-1,0,283,35.0#	OK
13	$-0.03,0.01,0.97,-1,-1,-0,32,-11,-22,1,1,285,35.1#	$-0.03,0.01,0.97,-1,-1,-0,32,-11,-22,1,1,285,35.1#	OK
14	$0.00,-0.03,0.99,-0,-0,-0,33,-11,-23,-2,-0,282,34.9#	$0.00,-0.03,0.99,-0,-0,-0,33,-11,-23,-2,-0,282,34.9#	OK
15	$-0.03,-0.03,1.01,1,0,0,32,-12,-21,-2,2,285,35.0#	$0.00,-0.03,0.99,-0,-0,-0,33,-11,-23,-2,-0,282,34.9#	-

Data ke-	Baris Data Sensor Sistem Roket (Sender)	Baris Data Sistem GCS (Receiver)	Validasi
16	$-0.04,-0.02,1.00,1,1,0,33,-11,-21,-1,2,283,35.0#	$-0.04,-0.02,1.00,1,1,0,33,-11,-21,-1,2,283,35.0#	OK
17	$-0.05,0.00,0.98,1,0,0,33,-11,-23,0,3,284,34.8#	$-0.05,0.00,0.98,1,0,0,33,-11,-23,0,3,284,34.8#	OK
18	$-0.03,-0.04,1.01,1,1,0,32,-12,-22,-2,2,284,35.1#	$-0.03,-0.04,1.01,1,1,0,32,-12,-22,-2,2,284,35.1#	OK
19	$-0.01,-0.06,1.03,1,1,0,31,-11,-22,-3,0,282,35.1#	$-0.01,-0.06,1.03,1,1,0,31,-11,-22,-3,0,282,35.1#	OK
20	$-0.01,-0.05,1.02,1,1,0,31,-12,-23,-3,1,284,34.8#	$-0.01,-0.05,1.02,1,1,0,31,-12,-23,-3,1,284,34.8#	OK
21	$0.00,-0.07,1.03,1,1,0,32,-11,-22,-4,-0,282,34.8#	$0.00,-0.07,1.03,1,1,0,32,-11,-22,-4,-0,282,34.8#	OK
22	$0.01,-0.06,1.03,0,1,0,33,-12,-23,-3,-1,282,35.0#	$0.01,-0.06,1.03,0,1,0,33,-12,-23,-3,-1,282,35.0#	OK
		Total data berhasil diterima	**19**

Tabel 5 menunjukkan bahwa dari 22 data yang dikirim oleh sistem roket ke sistem GCS, hanya 19 data yang berhasil diterima dan valid. Jika dinyatakan dalam persentase keberhasilan diperoleh sebesar 86% data berhasil diterima dan diolah. Jika dianalisis pada data yang error terdapat pada data ke-5, 10, dan 15 di sistem GCS. Nilai yang diterima pada data ke-5, 10, dan 15 ini, sama dengan nilai yang ditampilkan pada data sebelumnya. Hasil analisa didapatkan bahwa algoritma dekompresi Huffman pada sistem GCS mengalami overload buffering data saat melakukan proses pengolahan data. Hal ini disebabkan karena terjadi loncatan data yang tidak diinginkan terhadap jumlah data yang telah dialokasikan sebelumnya dan menjadikan data tidak dapat dibaca dan dianggap data yang rusak. Sehingga, data yang ditampilkan tidak disimpan pada variabel dekompresi dan menampilkan data penyimpanan variabel sebelumnya.

Kesimpulan

Sistem kompresi data dengan menggunakan algoritma kode Huffman pada muatan roket dan sistem GCS telah dapat direalisasikan dengan baik. Hasil yang didapatkan dengan menggunakan algoritma ini, data dapat dikompresi menjadi data yang sederhana dengan rasio kompresi sebesar 41,25%. Jumlah yang dapat dibaca dan dikirimkan pada sistem roket sebesar 22 baris data dalam waktu 2 detik. Hasil pengiriman dilakukan validasi dengan hasil penerimaan yang diolah pada sistem GCS sebesar 19 baris data atau sebesar 86% data valid.

Daftar Pustakas

Mudarris, M., & Zain, S. G. (2020). Implementasi Sensor Inertial Meansurenment Unit (IMU) untuk Monitoring Perilaku Roket. *Avitec*, *2*(1), 55–64. https://doi.org/10.28989/avitec.v2i1.610

Oklilas, A. F., Zulfahmi, R., Ermatita, & Jaya, A. P. (2019). Temperature Monitoring System Based on Protocol Message Queue Telemetry Transport (MQTT). *Proceedings - 1st International Conference on Informatics, Multimedia, Cyber and Information System, ICIMCIS 2019*, 61–66. https://doi.org/10.1109/ICIMCIS48181.2019.8985356

Prasetyo Adi, P. D., & Kitagawa, A. (2020). A study of LoRa performance in monitoring of patient's SPO2 and heart rate based IoT. *International Journal of Advanced Computer Science and Applications*, *1*(2), 238–251. https://doi.org/10.14569/ijacsa.2020.0110232

Rustamaji, H. C., Mariani, M., & Yuwono, B. (2015). Aplikasi Kompresi Data Menggunakan Metode Huffman Statik Pada Perangkat Mobile Berbasis Android. *Telematika*, *11*(1). https://doi.org/10.31315/telematika.v11i1.311

Sani, M. I. (2017). Implementasi Zigbee Transceiver Untuk Akuisisi Data Sensor Inersia Pada Wireless Body Area Network (WBAN). *Jurnal Infotel*, *9*(1), 48. https://doi.org/10.20895/infotel.v9i1.154

Yulkifli, Yohandri, & Affandi, Z. (2016). Pembuatan Sistem Pengiriman Data Menggunakan Telemetri Wireless untuk Detektor Getaran Mesin Dengan Sensor Fluxgate. *Jurnal Ilmiah SETRUM*, *5*(2), 57–61.