

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Penelitian-Penelitian Sebelumnya

Pada sub bab ini, akan dibahas langkah-langkah awal yang menjadi dasar bagi penelitian ini. Proses perencanaan penelitian dimulai dengan merumuskan pertanyaan penelitian yang relevan dan signifikan. Melalui eksplorasi literatur yang mendalam, studi ini mengidentifikasi kerangka teoritis yang relevan dan penelitian sebelumnya yang telah dilakukan dalam bidang yang sama.

Pada penelitian yang dilakukan oleh [6] yang berjudul “Pengembangan Aplikasi Ground Control Station (GCS) untuk Pengawasan dan Pengendalian UAV” bertujuan untuk Pada penelitian ini bertujuan untuk mengembangkan aplikasi GCS (*Ground Control Station*) yang mampu mempermudah pengguna dalam hal mengontrol dan memonitoring kondisi UAV secara real time dengan menampilkan data keadaan dan parameter penerbangan UAC sesuai dengan kebutuhan pengguna. Penelitian ini memiliki beberapa komponen *hardware* seperti *Flight Controller* 3DR Mini, *Telemetry* 433MHz, dan GPS.

Pada penelitian yang dilakukan oleh [2] yang berjudul “Rancang Bangun Sistem Monitoring Muatan Balon Atmosfer pada Kompetisi Muatan Balon Atmosfer Tahun 2019 Lembaga Penerbangan dan Antariksa Nasional (LAPAN) Kabupaten Garut” penelitian ini membuat suatu rancangan system *monitoring* yang dapat digunakan untuk menampilkan dan menyimpan data profil *vertical* atmosfer secara real time bagi pengguna. Pada muatan balon atmosfer mampu memonitoring beberapa parameter seperti suhu, kelembaban, tekanan udara, ketinggian. Beberapa komponen pendukung yang ada pada muatan nya yaitu seperti *telemetry* 433MHz, sensor GPS Ublox Neo-7m dan BME280.

Pada penelitian yang dilakukan oleh [7] yang berjudul “Pembangunan Aplikasi Ground Control Station pada Unmanned Aerial Vehicle Berbasis IoT” penelitian ini bertujuan untuk memudahkan operator atau pengawas dalam memantau kondisi UAV dari jarak yang jauh dan mengevaluasi hasil pengujian terbang dari UAV. Adapun komponen *hardware* yang dipakai dalam penelitian ini seperti BMP180, HMC5883L, NEO6M V2, dan MPU5060.

Pada penelitian yang dilakukan oleh [8] yang berjudul “Sistem Telemetry pada Ground Control Station (GCS) Muatan Balon Atmosfer (Payload) Menggunakan Delphi 7” Penelitian ini lebih membahas mengenai penggunaan pada system. Perancangan dan realisasi sistem telemetry menggunakan aplikasi Delphi7 yang berfungsi sebagai penampung data dan menampilkan informasi data yang didapatkan seperti pressure, Temperature, Humidity, Lat, Long, dan Altitude. Yang dikirim oleh alat ukur muatan balon atmosfer. Lalu Data tersebut akan disimpan ke dalam file berformat .txt

Pada penelitian yang dilakukan oleh [9] yang berjudul “ Rancang Bangun Payload dan Interface Monitoring Roket dalam Visualisasi 3D dan Pengambilan Foto” ini bertujuan untuk menganalisa kinerja pada *payload* dan juga pengambilan foto saat di udara beserta interface Nya, Adapun komponen yang terdapat pada *payload* seperti Xbee-pro, sensor GY801, dan kamera VC0706. Hasil dari penelitian ini adalah perangkat lunak yang digunakan mampu melakukan pengamatan perilaku roket dan menampilkan hasil foto yang dikirim oleh roket/*payload*.

## 2.2 Sistem Monitoring

*Monitoring* merujuk pada serangkaian kegiatan yang meliputi pengumpulan data, peminjaman ulang, pelaporan, dan tindakan atau informasi yang ada kaitannya dengan suatu pengerjaan yang sedang berlangsung. Selain itu, sistem merujuk pada sekelompok faktor yang saling terkait yang mempunyai tujuan tertentu secara bersama-sama. Jadi, dapat disimpulkan *sistem monitoring* merupakan kumpulan siklus kegiatan yang meliputi serangkaian kegiatan yang terkait dengan informasi mengenai pengerjaan yang sedang berlangsung [2].

Menurut [10] pemantauan ialah proses untuk mengetahui status, kemajuan, dan kecenderungan suatu kegiatan atau proses. Untuk melakukan ini, digunakan indikator tertentu untuk memastikan kesesuaian rencana dan protokol yang telah ditetapkan.

### 2.3 Algoritma dan Metode

Pada tahap ini, akan menjelaskan metode yang digunakan untuk membantu penulis dalam menyelesaikan perancangan alat *monitoring vertical* atmosfer, serta mendeteksi kesalahan pada data. Berikut metode nya:

#### 2.3.1 Metode CRC

Menurut [11] *Cyclic Redundancy Check* (CRC) adalah Teknik deteksi kesalahan system penyimpanan data dan jaringan komputer. Metode ini menghitung dan menambah *checksum* atau *check bit* ke data yang ingin dikirimkan. Nilai CRC ini digunakan untuk mengidentifikasi data *error* yang diterima ialah jenis CRC yang biasa digunakan termasuk CRC-8, CRC-10, CRC-12, CRC-16, dan CRC-32. Untuk komputasinya mirip dengan kode siklik lainnya, karena CRC didasari pada gambar 2.1.

Code	Generator Polynomial $g(x)$	Jumlah CRC bit	Penggunaan
CRC-4	$x^4 + x + 1$	4	ITU G.704, IEEE 802.11ah
CRC-8-CCITT	$x^8 + x^2 + x + 1$	8	ATM Header Error Control
CRC-10	$x^{10} + x^9 + x^5 + x^4 + x + 1$	10	ATM Adaptation Layer
CRC-12	$x^{12} + x^{11} + x^3 + x^2 + x + 1$	12	Telecom system
CRC-16-CCITT	$x^{16} + x^{12} + x^5 + 1$	16	HDLC, Bluetooth, XMODEM
CRC-16-ANSI	$x^{16} + x^{15} + x^2 + 1$	16	USB, Modbus
CRC-24	$x^{24} + x^{10} + x^9 + x^6 + x^4 + x^3 + x + 1$	24	Bluetooth
CRC-32 IEEE 802	$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$	32	<i>Local Area Network</i> (LAN)

Gambar 2. 1 Kode Siklik

Pada penelitian ini metode CRC16 IBM adalah algoritma yang dikembangkan oleh *International Business Machines Corporation* (IBM). CRC16 ini digunakan dalam berbagai aplikasi, termasuk komunikasi data, deteksi kesalahan, dan validasi data.

CRC-16-IBM	Bisync, Modbus, USB, ANSI X3.28 <sup>2</sup> , SIA DC-07, many others; also known as CRC-16 and CRC-16-ANSI	0x8005	0xA001	0x4003	0xC002
		$x^{16} + x^{15} + x^2 + 1$			

*Gambar 2. 2 CRC 16 IBM*

Pada gambar 2.2 merupakan kode polynomial yang dipakai pada penelitian ini. Polynomial “A001” adalah polynomial yang digunakan pada penelitian ini untuk perhitungan checksum dalam berbagai protokol komunikasi serta protokol seperti Modbus untuk deteksi kesalahan dalam transmisi data. Dalam bentuk heksadesimal, polynomial “A001” direpresentasikan sebagai “0xA001”. Sedangkan, dalam bentuk biner, polynomial ini memiliki pola “1010 0000 0000 0001”. Untuk implementasi CRC16 pada penelitian ini, nilai CRC dihitung dan dibandingkan dengan data perhitungan manual dan perhitungan digital. Berikut merupakan tabel 2.1 yang menjelaskan tentang implementasi berdasarkan satu data dari *payload*:

*Tabel 2. 1 implementasi CRC16*

<b>Data Payload: ,25:1:28,799.7,-6.888277,107.618420,927.4,25.2,70.0,1.0,</b>	
Perhitungan Kalkulator CRC16	
Algorithm	Result
CRC-16/MODBUS	0xEF19
Perhitungan Manual CRC16	





0x7e65=	0	1	1	1	1	1	1	0	0	1	1	0	0	1	0	1			
0x3a=									0	0	1	1	1	0	1	0			
XOR=	0	1	1	1	1	1	1	0	0	1	0	1	1	1	1	1	0x7E5F		
Hasil XOR (0x7e65 dan 0x3a)	0	1	1	1	1	1	1	0	0	1	0	1	1	1	1	1	Update Register		
0 Filling MSB	0	0	1	1	1	1	1	1	0	0	1	0	1	1	1	1			
Hasil	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	CRC Poly		
Hasil	1	0	0	1	1	1	1	1	0	0	1	0	1	1	1	0	Update Register	Shift 1	0x9F2E
0 Filling MSB	0	1	0	0	1	1	1	1	1	0	0	1	0	1	1	1		Shift 2	0x4F97
0 Filling MSB	0	0	1	0	0	1	1	1	1	1	0	0	1	0	1	1			
Hasil	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	CRC Poly		
Hasil	1	0	0	0	0	1	1	1	1	0	0	1	0	1	0	0	Update Register	Shift 3	0x87CA
0 Filling MSB	0	1	0	0	0	0	1	1	1	1	0	0	1	0	1	1		Shift 4	0x43E5
0 Filling MSB	0	0	1	0	0	0	0	1	1	1	1	1	0	0	1	0			
Hasil	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	CRC Poly		
Hasil	1	0	0	0	0	0	0	1	1	1	1	0	0	1	1	0	Update Register	Shift 5	0x81F3
0 Filling MSB	0	1	0	0	0	0	0	0	1	1	1	1	1	0	0	1			
Hasil	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	CRC Poly		
Hasil	1	1	1	0	0	0	0	0	1	1	1	1	1	0	0	0	Update Register	Shift 6	0xE0F8
0 Filling MSB	0	1	1	1	0	0	0	0	0	1	1	1	1	1	0	0		Shift 7	0x707C
0 Filling MSB	0	0	1	1	1	0	0	0	0	0	1	1	1	1	1	0		Shift 8	0x383E

0x383E=	0	0	1	1	1	0	0	0	0	0	1	1	1	1	1	0				
0x31=											0	0	1	1	0	0	1			
XOR=	0	0	1	1	1	0	0	0	0	0	0	0	0	1	1	1	0x380F			
Hasil XOR	0	0	1	1	1	0	0	0	0	0	0	1	1	1	1	1	Update Register			
0 Filling MSB	0	0	0	1	1	1	0	0	0	0	0	0	0	1	1	1				
Hasil	1	0	1	1	1	1	0	0	0	0	0	0	0	1	1	0	CRC Poly			
Hasil	1	0	1	1	1	1	0	0	0	0	0	0	0	1	1	0	Update Register	Shift 1	0xbc06	
0 Filling MSB	0	1	0	1	1	1	1	0	0	0	0	0	0	1	1	1		Shift 2	0x5e03	
0 Filling MSB	0	0	1	0	1	1	1	1	0	0	0	0	0	0	0	1				
Hasil	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	CRC Poly			
Hasil	1	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	Update Register	Shift 3	0x8F00	
0 Filling MSB	0	1	0	0	0	1	1	1	1	0	0	0	0	0	0	0		Shift 4	0x4780	
0 Filling MSB	0	0	1	0	0	0	1	1	1	1	0	0	0	0	0	0		Shift 5	0x23c0	
0 Filling MSB	0	0	0	1	0	0	0	1	1	1	1	0	0	0	0	0		Shift 6	0x11e0	
0 Filling MSB	0	0	0	0	1	0	0	0	1	1	1	1	0	0	0	0		Shift 7	0x8F0	
0 Filling MSB	0	0	0	0	0	1	0	0	0	1	1	1	1	0	0	0		Shift 8	0x478	

0x478=	0	0	0	0	0	1	0	0	0	1	1	1	1	0	0	0				
0x3a=										0	0	1	1	1	0	1	0			
XOR=	0	0	0	0	0	1	0	0	0	1	0	0	0	0	1	0	0x442			
Hasil XOR	0	0	0	0	0	1	0	0	0	1	0	0	0	0	1	0	Update Register			
0 Filling MSB	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	1		Shift 1	0x221	
0 Filling MSB	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0				
Hasil	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	CRC Poly			
Hasil	1	0	1	0	0	0	0	1	0	0	0	1	0	0	0	1	Update Register	Shift 2	0xA111	
0 Filling MSB	0	1	0	1	0	0	0	0	1	0	0	0	1	0	0	0				
Hasil	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	CRC Poly			
Hasil	1	1	1	1	0	0	0	0	1	0	0	0	1	0	0	1	Update Register	Shift 3	0xf089	
0 Filling MSB	0	1	1	1	1	0	0	0	0	1	0	0	0	1	0	0				
Hasil	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	CRC Poly			
Hasil	1	1	0	1	1	0	0	0	0	1	0	0	0	1	0	1	Update Register	Shift 4	0xd845	
0 Filling MSB	0	1	1	0	1	1	0	0	0	0	1	0	0	0	1	0				
Hasil	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	CRC Poly			
Hasil	1	1	0	0	1	1	0	0	0	0	1	0	0	0	1	1	Update Register	Shift 5	0xcc23	
0 Filling MSB	0	1	1	0	0	1	1	0	0	0	0	1	0	0	0	1				
Hasil	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	CRC Poly			
Hasil	1	1	0	0	0	1	1	0	0	0	0	1	0	0	0	0	Update Register	Shift 6	0xc610	
0 Filling MSB	0	1	1	0	0	0	1	1	0	0	0	0	1	0	0	0		Shift 7	0x6308	
0 Filling MSB	0	0	1	1	0	0	0	1	1	0	0	0	0	1	0	0		Shift 8	0x3184	

0x3184=	0	0	1	1	0	0	0	1	0	0	0	0	1	0	0				
0x32=									0	0	1	1	0	0	1	0			
XOR=	0	0	1	1	0	0	0	1	1	0	1	1	0	1	1	0	0x31b6		
	0	0	1	1	0	0	0	1	1	0	1	1	0	1	1	0	Update Register		
0 Filling MSB	0	0	0	1	1	0	0	0	1	1	0	1	1	0	1	1		Shift 1	0x18db
0 Filling MSB	0	0	0	0	1	1	0	0	0	1	1	0	1	1	0	1			
	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	CRC Poly		
Hasil	1	0	1	0	1	1	0	0	0	1	1	0	1	1	0	0	Update Register	Shift 2	0xac6c
0 Filling MSB	0	1	0	1	0	1	1	0	0	0	1	1	0	1	1	0		Shift 3	0x5636
0 Filling MSB	0	0	1	0	1	0	1	1	0	0	0	1	1	0	1	1		Shift 4	0x2b1b
0 Filling MSB	0	0	0	1	0	1	0	1	1	0	0	0	1	1	0	1			
	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	CRC Poly		
Hasil	1	0	1	1	0	1	0	1	1	0	0	1	1	0	0	0	Update Register	Shift 5	0xb58c
0 Filling MSB	0	1	0	1	1	0	1	0	1	1	0	0	0	1	1	0		Shift 6	0x5ac6
0 Filling MSB	0	0	1	0	1	1	0	1	0	1	1	0	0	0	1	1		Shift 7	0x2d63
0 Filling MSB	0	0	0	1	0	1	1	0	1	0	1	1	0	0	0	1			
	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	CRC Poly		
Hasil	1	0	1	1	0	1	1	0	1	0	1	1	0	0	0	1	Update Register	Shift 8	0xb6b0

0xb6b0=	1	0	1	1	0	1	1	0	1	0	1	0	1	1	0	0	0	0			
0x36=																					
XOR=	1	0	1	1	0	1	1	0	1	0	1	0	0	1	0	0	0	0	0xb688		
	1	0	1	1	0	1	1	0	1	0	0	0	1	0	0	0	0	0	Update Register		
0 Filling MSB	0	1	0	1	1	0	1	1	0	1	0	0	0	1	0	0	0	0		Shift 1	0x54bb
0 Filling MSB	0	0	1	0	1	1	0	1	1	0	1	0	0	0	1	0	0	0		Shift 2	0x2da2
0 Filling MSB	0	0	0	1	0	1	1	0	1	1	0	1	0	0	0	0	1	0		Shift 3	0x16d1
0 Filling MSB	0	0	0	0	1	0	1	1	0	1	1	0	1	0	0	0	0	0			
	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	CRC Poly		
Hasil	1	0	1	0	1	0	1	1	0	1	1	0	1	0	0	1	0	0	Update Register	Shift 4	0xab69
0 Filling MSB	0	1	0	1	0	1	0	1	1	0	1	1	0	1	0	0	0	0			
	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	CRC Poly		
Hasil	1	1	1	1	0	1	0	1	1	0	1	1	0	1	0	1	0	1	Update Register	Shift 5	0xf5b5
0 Filling MSB	0	1	1	1	1	0	1	0	1	1	0	1	1	0	1	0	1	0			
	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	CRC Poly		
Hasil	1	1	0	1	1	0	1	0	1	1	0	1	1	0	1	1	0	1	Update Register	Shift 6	0xdadb
0 Filling MSB	0	1	1	0	1	1	0	1	0	1	1	0	1	1	0	1	0	1			
	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	CRC Poly		
Hasil	1	1	0	0	1	1	0	1	0	1	1	0	1	1	0	1	0	0	Update Register	Shift 7	0xcd6c
0 Filling MSB	0	1	1	0	0	1	1	0	1	0	1	1	0	1	1	0	1	0		Shift 8	0x66b6

0x66b6=	0	1	1	0	0	1	1	0	1	0	1	1	0	1	1	0	0	0			
0x2c=																					
XOR=	0	1	1	0	0	1	1	0	1	0	0	1	1	0	1	0	0	0	0x669a		
	0	1	1	0	0	1	1	0	1	0	0	1	1	0	1	0	0	0	Update Register		
0 Filling MSB	0	0	1	1	0	0	1	1	0	1	0	0	1	1	0	1	0	1		Shift 1	0x334d
0 Filling MSB	0	0	0	1	1	0	0	1	1	0	1	0	0	1	1	0	0	0			
	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	CRC Poly		
Hasil	1	0	1	1	1	0	0	1	1	0	1	0	0	1	1	1	0	1	Update Register	Shift 2	0xb9a7
0 Filling MSB	0	1	0	1	1	1	0	0	1	1	0	1	0	0	1	1	0	1			
	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	CRC Poly		
Hasil	1	1	1	1	1	1	0	0	1	1	0	1	0	0	1	0	0	1	Update Register	Shift 3	0xfcd2
0 Filling MSB	0	1	1	1	1	1	1	0	0	1	1	0	1	0	0	1	0	0		Shift 4	0x7e69
0 Filling MSB	0	0	1	1	1	1	1	1	0	0	0	1	1	0	1	0	0	0			
	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	CRC Poly		
Hasil	1	0	0	1	1	1	1	1	0	0	1	1	0	1	0	1	0	1	Update Register	Shift 5	0x9f35
0 Filling MSB	0	1	0	0	1	1	1	1	1	0	0	1	1	0	1	0	1	0			
	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	CRC Poly		
Hasil	1	1	1	0	1	1	1	1	1	0	0	1	1	0	1	1	0	1	Update Register	Shift 6	0xef9b
0 Filling MSB	0	1	1	1	0	1	1	1	1	1	0	0	1	1	0	1	0	1			
	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	CRC Poly		
Hasil	1	1	0	1	0	1	1	1	1	1	0	0	1	1	0	0	0	1	Update Register	Shift 7	0xd7cc
0 Filling MSB	0	1	1	0	1	0	1	1	1	1	1	0	0	1	1	0	0	0		Shift 8	0x6be6



0x6be6=	0	1	1	0	1	0	1	1	1	1	1	0	0	1	1	0			
0x37=												0	0	1	1	0			
XOR=	0	1	1	0	1	0	1	1	1	1	0	1	0	0	0	1	0x6bd1		
	0	1	1	0	1	0	1	1	1	1	0	1	0	0	0	1	Update Register		
0 Filling MSB	0	0	1	1	0	1	0	1	1	1	1	0	1	0	0	0	1		
	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	CRC Poly		
Hasil	1	0	0	1	0	1	0	1	1	1	0	1	0	0	1	Update Register	Shift 1	0x95e9	
0 Filling MSB	0	1	0	0	1	0	1	0	1	1	1	1	0	1	0	0	1		
	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	CRC Poly		
Hasil	1	1	1	0	1	0	1	0	1	1	1	0	1	0	1	Update Register	Shift 2	0xeaf5	
0 Filling MSB	0	1	1	1	0	1	0	1	0	1	1	1	1	0	1	0	1		
	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	CRC Poly		
Hasil	1	1	0	1	0	1	0	1	0	1	1	1	0	1	1	Update Register	Shift 3	0xd57b	
0 Filling MSB	0	1	1	0	1	0	1	0	1	0	1	1	1	1	0	1	1		
	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	CRC Poly		
Hasil	1	1	0	0	1	0	1	0	1	0	1	1	1	0	0	Update Register	Shift 4	0xcabc	
0 Filling MSB	0	1	1	0	0	1	0	1	0	1	1	1	1	1	0	0	0	Shift 5	0x655e
	0	0	1	1	0	0	1	0	1	0	1	1	1	1	1	0	Shift 6	0x32af	
0 Filling MSB	0	0	0	1	1	0	0	1	0	1	0	1	0	1	1	1	1		
	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	CRC Poly		
Hasil	1	0	1	1	1	0	0	1	0	1	0	1	1	0	0	Update Register	Shift 7	0xb956	
0 Filling MSB	0	1	0	1	1	1	0	0	1	0	1	0	1	0	1	1	0	Shift 8	0x5cab
0x5cab=	0	1	0	1	1	1	0	0	1	0	1	0	1	0	1	1			
0x39=																			
XOR=	0	1	0	1	1	1	0	0	1	0	0	1	0	0	1	0	0x5c32		
	0	1	0	1	1	1	0	0	1	0	0	1	0	0	1	0	Update Register		
0 Filling MSB	0	0	1	0	1	1	1	0	0	1	0	0	1	0	0	1	0	Shift 1	0x2e49
	0	0	0	1	0	1	1	1	0	0	1	0	0	1	0	0	1		
0 Filling MSB	0	0	0	1	0	1	1	1	0	0	1	0	0	1	0	0	0		
	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	CRC Poly		
Hasil	1	0	1	1	0	1	1	1	0	0	1	0	0	1	0	Update Register	Shift 2	0xb725	
0 Filling MSB	0	1	0	1	1	0	1	1	1	0	0	1	0	0	1	0	1		
	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	CRC Poly		
Hasil	1	1	1	1	1	0	1	1	1	0	0	1	0	0	1	Update Register	Shift 3	0xfb93	
0 Filling MSB	0	1	1	1	1	1	0	1	1	1	0	0	1	0	0	1	0		
	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	CRC Poly		
Hasil	1	1	0	1	1	1	0	1	1	1	0	0	1	0	0	Update Register	Shift 4	0xddc8	
0 Filling MSB	0	1	1	0	1	1	1	0	1	1	1	0	0	1	0	0	0	Shift 5	0x6ee4
	0	0	1	1	0	1	1	1	0	1	1	1	0	0	1	0	0	Shift 6	0x3772
0 Filling MSB	0	0	0	1	1	0	1	1	1	0	1	1	1	0	0	1	0	Shift 7	0x1b99
	0	0	0	1	1	0	1	1	1	0	1	1	1	0	0	1			
0 Filling MSB	0	0	0	0	1	1	0	1	1	1	0	1	1	1	0	0	0		
	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	CRC Poly		
Hasil	1	0	1	0	1	1	0	1	1	1	0	1	1	1	0	1	Update Register	Shift 8	0xaddd



0xaddd=	1	0	1	0	1	1	0	1	1	1	0	1	1	1	0	1					
0x39=											0	0	1	1	1	0	0	1			
XOR=	1	0	1	0	1	1	0	1	1	1	1	0	0	1	0	0			0xade4		
	1	0	1	0	1	1	0	1	1	1	1	0	0	1	0	0			Update Register		
																			0		
0 Filling MSB	0	1	0	1	0	1	1	0	1	1	1	0	0	1	0					Shift 1	0x56f2
																			0		
0 Filling MSB	0	0	1	0	1	0	1	1	0	1	1	1	1	0	0	1				Shift 2	0x2b79
																			1		
0 Filling MSB	0	0	0	1	0	1	0	1	1	0	1	1	1	1	0	0					
	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1			CRC Poly		
Hasil	1	0	1	1	0	1	0	1	1	0	1	1	1	1	0	1			Update Register	Shift 3	0xb5bd
																			1		
0 Filling MSB	0	1	0	1	1	0	1	0	1	1	0	1	1	1	1	0					
	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1			CRC Poly		
Hasil	1	1	1	1	1	0	1	0	1	1	0	1	1	1	1	1			Update Register	Shift 4	0xfadf
																			1		
0 Filling MSB	0	1	1	1	1	1	0	1	0	1	1	0	1	1	1	1					
	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1			CRC Poly		
Hasil	1	1	0	1	1	1	0	1	0	1	1	0	1	1	1	0			Update Register	Shift 5	0xdd6e
																			0		
0 Filling MSB	0	1	1	0	1	1	1	0	1	0	1	1	0	1	1	1				Shift 6	0x6eb7
																			1		
0 Filling MSB	0	0	1	1	0	1	1	1	0	1	0	1	1	0	1	1					
	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1			CRC Poly		
Hasil	1	0	0	1	0	1	1	1	0	1	0	1	1	0	1	0			Update Register	Shift 7	0x975a
																			0		
0 Filling MSB	0	1	0	0	1	0	1	1	1	0	1	0	1	1	0	1				Shift 8	0x4bad

Setelah mendapatkan hasil dari bit diatas dengan perhitungan diulang selama 8 kali maka bit 0x39 memiliki nilai Hex 0x4bad. Begitupun untuk pengujian bitbit selanjutnya. Berikut merupakan hasil pengujian berikutnya:

### Pembuktian Metode

- Byte: 0x39, Bit: 1, CRC: 0x4bad
- Byte: 0x2e, Bit: 0, CRC: 0xa10a
- Byte: 0x37, Bit: 0, CRC: 0xd160
- Byte: 0x2c, Bit: 0, CRC: 0xf5d0
- Byte: 0x2d, Bit: 0, CRC: 0x8134
- Byte: 0x36, Bit: 0, CRC: 0xc100
- Byte: 0x2e, Bit: 1, CRC: 0x1c41
- Byte: 0x38, Bit: 1, CRC: 0xe2dd
- Byte: 0x38, Bit: 1, CRC: 0x8b23
- Byte: 0x38, Bit: 1, CRC: 0xbcb
- Byte: 0x32, Bit: 1, CRC: 0x42cb
- Byte: 0x37, Bit: 0, CRC: 0x4142
- Byte: 0x37, Bit: 0, CRC: 0xe780
- Byte: 0x2c, Bit: 1, CRC: 0x7de7
- Byte: 0x31, Bit: 0, CRC: 0x9efc
- Byte: 0x30, Bit: 0, CRC: 0x559e
- Byte: 0x37, Bit: 1, CRC: 0x7e95

**Pembuktian Metode**

- Byte: 0x2e, Bit: 0, CRC: 0x733e
- Byte: 0x36, Bit: 0, CRC: 0xc672
- Byte: 0x31, Bit: 1, CRC: 0xf187
- Byte: 0x38, Bit: 0, CRC: 0xb0b0
- Byte: 0x34, Bit: 0, CRC: 0x63b0
- Byte: 0x32, Bit: 1, CRC: 0x61e3
- Byte: 0x30, Bit: 0, CRC: 0x9d20
- Byte: 0x2c, Bit: 1, CRC: 0x59d
- Byte: 0x39, Bit: 0, CRC: 0xbb04
- Byte: 0x32, Bit: 1, CRC: 0x163b
- Byte: 0x37, Bit: 0, CRC: 0x516
- Byte: 0x2e, Bit: 0, CRC: 0xd204
- Byte: 0x34, Bit: 0, CRC: 0x14d2
- Byte: 0x2c, Bit: 1, CRC: 0x8095
- Byte: 0x32, Bit: 1, CRC: 0xbac1
- Byte: 0x35, Bit: 1, CRC: 0x87bb
- Byte: 0x2e, Bit: 1, CRC: 0x6f47
- Byte: 0x32, Bit: 0, CRC: 0xe7ae
- Byte: 0x2c, Bit: 1, CRC: 0x6167
- Byte: 0x37, Bit: 1, CRC: 0x3c61
- Byte: 0x30, Bit: 1, CRC: 0xfcfd
- Byte: 0x2e, Bit: 1, CRC: 0x9dbd
- Byte: 0x30, Bit: 1, CRC: 0x655d
- Byte: 0x2c, Bit: 1, CRC: 0x24a5
- Byte: 0x31, Bit: 1, CRC: 0xaf25
- Byte: 0x2e, Bit: 0, CRC: 0xc7ee
- Byte: 0x30, Bit: 1, CRC: **0x5847**

0x5847	0	1	0	1	1	0	0	0	0	1	0	0	0	1	1	1				
0x2c =										0	0	1	0	1	1	0	0			
XOR =	0	1	0	1	1	0	0	0	0	0	1	0	1	0	1	1			0x586B	
Hasil XOR (0x5847 dan 0x2c) =	0	1	0	1	1	0	0	0	0	1	1	0	1	0	1	1	Update Register			
0 Filling MSB	0	0	1	0	1	1	0	0	0	0	1	1	0	1	0	1				
Hasil	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	CRC Poly			
Hasil	1	0	0	0	1	1	0	0	0	0	1	1	0	1	0	0	Update Register	Shift 1	0x8c34	
0 Filling MSB	0	1	0	0	0	1	1	0	0	0	0	1	1	0	1	0				
0 Filling MSB	0	0	1	0	0	0	1	1	0	0	0	0	1	1	0	1				
0 Filling MSB	0	0	1	0	0	0	1	1	0	0	0	0	1	1	0	1				
0 Filling MSB	0	0	1	0	0	0	1	1	0	0	0	0	1	1	0	1				
0 Filling MSB	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	CRC Poly			
Hasil	1	0	1	1	0	0	0	1	1	0	0	0	0	1	1	1	Update Register	Shift 4	0xb187	
0 Filling MSB	0	1	0	1	1	0	0	0	1	1	0	0	0	0	1	1				
0 Filling MSB	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	CRC Poly			
Hasil	1	1	1	1	1	0	0	0	1	1	0	0	0	0	1	0	Update Register	Shift 5	0xf8c2	
0 Filling MSB	0	1	1	1	1	1	0	0	0	1	1	0	0	0	0	1				
0 Filling MSB	0	0	1	1	1	1	1	0	0	0	1	1	0	0	0	0				
0 Filling MSB	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	CRC Poly			
Hasil	1	0	0	1	1	1	1	0	0	0	1	1	0	0	0	1	Update Register	Shift 7	0x9e31	
0 Filling MSB	0	1	0	0	1	1	1	1	0	0	0	1	1	0	0	0				
0 Filling MSB	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	CRC Poly			
Hasil	1	1	1	0	1	1	1	1	0	0	0	1	1	0	0	1	Update Register	Shift 8	0xef19	

**2.3.2 Analisa Metode CRC**

Implementasi metode CRC16 pada tabel 2.1 untuk perhitungan manual dan menggunakan kalkulator CRC. Berikut, prosedur menghitung nilai CRC:

1. Isi sebuah register 16-bit dengan FFFF hex.
2. ExOR 8-bit byte pertama data dengan low-order byte Register 16 bit CRC, simpan hasil di register CRC.
3. Geser kekanan 1 bit CRC register, isi MSB dengan 0. Ambil dan cek LSB apakah 1 atau 0.

4. (Jika LSB= 0): ulangi Langkah 3(geser lagi 1 bit). (Jika LSB= 1): ExOR CRC register dengan 0xA001 (1010 0000 0000 0001).
5. Ulangi Langkah 3 dan 4 sampai 8 kali pergeseran dilakukan. Setelah, sebuah byte siap akan diproses.
6. Ulangi Langkah 2 sampai 5 untuk byte berikutnya dari data. Lanjutkan sampai semua byte data diproses.
7. Hasil akhir isi CRC register adalah nilai CRC.

Berdasarkan prosedur yang dilakukan menggunakan polinomial  $X^{16} +$

$X^{15} + X^2 + 1$  untuk mendapatkan nilai CRC-16 seperti pada tabel 2.1 dengan melakukan pergeseran sebanyak 8 kali pada perhitungan CRC-16 adalah metode yang menghitung checksum untuk data berukuran 16bit (2 byte). Data yang dihitung menggunakan CRC-16 adalah data waktu, altitude, lat, long, tekanan, suhu, kelembapan, dan kecepatan. Setiap bit data diambil satu persatu dan dimasukkan ke dalam perhitungan. Karena setiap byte data terdiri dari 8 bit, pergeseran bit dilakukan sebanyak 8 kali untuk setiap byte data. Total 2 byte data akan menghasilkan 16 bit checksum yang sesuai dengan panjang data yang akan diperiksa integritasnya.

## 2.4 Perangkat Keras

Berikut merupakan komponen–komponen yang digunakan untuk merancang alat *monitoring vertical* atmosfer:

### 2.4.1 BME280

Pada Gambar 2.3 merupakan penerus dari beberapa sensor seperti BMP180. Komunikasi pada sensor ini menggunakan protokol yaitu I2C atau SPI yang berfungsi untuk bertukar data dengan mikrokontroler. [12] Serta ini memiliki pengatur level tegangan I2C dan regulator 3.3V. Sehingga dapat beroperasi pada perangkat mikro 3.3V atau 5V Arduino. [12] Sensor ini memiliki kemampuan beroperasi dalam rentang suhu yang luas, mulai dari  $-40^{\circ} \text{C} \sim 85^{\circ} \text{C}$  sehingga cocok

untuk berbagai kondisi lingkungan. Kelembaban pada sensor ini dapat diukur dalam rentang 0-100% RH dan dapat mengukur tekanan barometric dalam rentang 300Pa ~ 1100 hPa.



*Gambar 2. 3 Sensor BME280*

Menurut [13] modul ini dapat memberikan output seperti suhu, kelembaban, tekanan barometric dan ketinggian. Lalu menurut modul ini dapat mengukur output yang telah di jelaskan sebelumnya. Sehingga dapat digunakan pada aplikasi seperti navigasi *indoor / outdoor*, *monitoring* cuaca, home automation, serta wearable device

#### **2.4.2 AT Mega 328p**

Mikrokontroler AT Mega 328p merupakan mikro dari Atmel yang memiliki arsitektur RISC yang mana setiap proses eksekusi data lebih cepat dari arsitektur CISC. [14] Menurut gambar 2.4 ini adalah seri mikro CMOS 8-bit dan memiliki 8 Kbyte Flash yang dapat diprogram ulang melalui *Serial Peripheral Interface (SPI)*. Mikro ini berbasis RISC yang disebut AVR (*Alf and Vegaard's Risc Processor*). AVR memiliki reputasi dalam mengeksekusi program yang lebih cepat dibandingkan mikrokontroler lain, dengan Sebagian besar, instruksi yang diproses dalam 1 siklus clock dan memiliki kecepatan proses mencapai hamper 1 MIPS per MHz. hal ini dapat mengurangi konsumsi daya [15].





*Gambar 2. 4 Mikrokontroler AT Mega 328p*

Mikrokontroler ini memberikan keunggulan dalam kecepatan eksekusi data, karena menggunakan arsitektur RISC. Dibandingkan, dengan arsitektur CISC. Atmega328p memiliki beberapa fitur sebagai berikut:

- 1) Mempunyai 130 instruksi yang hampir keseluruhan dieksekusi dalam satu siklus clock.
- 2) Flash memory 32 kb.
- 3) Kecepatan eksekusi mencapai 16 MIPS dengan clock 16MHz
- 4) EEPROM (*Electrically Erasable Programmable Read Only Memory*) sebesar 1 Kb sebagai media penyimpanan data semi permanen. Karena hal ini dapat menyimpan data meskipun catu daya dimatikan.
- 5) SRAM (*Static Random AccessMemory*) sebesar 2 Kb.
- 6) Mempunyai 23 pin I/O digital.

#### **2.4.3 Modul GPS receiver U-blox Neo-6m**

Menurut [16] GPS Neo 6M merupakan penerima GPS stand-alone yang begitu kuat. *Time to First Fix* (TIFF) adalah waktu pada modul GPS dalam mengakses data satelit seperti almanac dan ephemeris. TIFF maksimum pada GPS adalah 27 detik. Adapun spesifikasi dari GPS ini, seperti tegangan maksimum yang beroperasi pada GPS adalah 3.6V, konsumsi arus max 67 mA, tipe *receiver* (RX) 50 channel, update rate 5Hz, akurasi kecepatan 0,1 m/s, dan komunikasi interface UART [17]. Menurut [18] modul ini memiliki kemampuan dalam mengidentifikasi koordinat lokasi dan kemudian mengirimkan data tersebut melalui saluran lokasi.



Gambar 2. 5 Module GPS Neo 6M

Menurut [19] modul GPS Neo 6M dapat melacak sampai 22 satelit dan memiliki sensitivitas hingga  $-161$  dB dengan arus suplai  $45$  mA. Dengan akurasi posisi horizontal sebesar  $2,5$  meter, dapat memperbarui lokasi setiap  $5$  detik. Selain itu *mode Power Save* (PSM) dapat mengurangi pemakaian daya menjadi  $11$  mA, yang membuatnya ideal untuk diimplementasikan pada jam tangan GPS yang memiliki keterbatasan daya. Berdasarkan gambar 2.5 memiliki baud rate default  $9600$ , modul ini mendukung baud rate mulai dari  $4800$ bps ~  $230400$ bps.

#### 2.4.5 Radio 3DR 433MHz

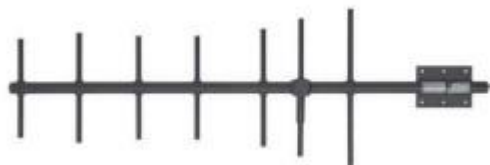
Menurut [20][21] Radio 3DR 433 MHz adalah sistem komunikasi nirkabel half-duplex dua arah. Radio ini merupakan modem jarak menengah yang mempunyai 2 tipe yaitu frekuensi  $915$  MHz dan juga  $433$  MHz yang dapat berkomunikasi sekitar jarak  $1$  mil [2]. Frekuensi  $433$  MHz dipilih karena tidak terpengaruh oleh refleksi dari dinding, struktur logam, dan air. Pada gambar 2.6 ini memiliki 2 modul yaitu *transmitter* (Tx) dan *receiver* (Rx) dan beroperasi pada catu daya  $5$  V, dengan menggunakan alat config telemetri yang bisa diatur sesuai kebutuhan pemakaian. Seperti mengubah baud rate, kecepatan angin, Tx power [2].



Gambar 2. 6 Radio 3DR 433MHz

#### 2.4.6 Antena Yagi

Menurut [22] antena yagi adalah antena *dipole* yang terdapat tambahan parasitic elements berupa *reflector* dan *director* sehingga menghasilkan sebuah gain ke arah tertentu. Komponen atau elemen utama yang terdapat pada antena ini memiliki fungsi seperti elemen dipole sebagai pemancar radiasi utama. Kemudian elemen *reflector* sebagai pemantul radiasi agar pola pancaran mengarah ke depan. Sedangkan komponen penyearah atau *director* digunakan sebagai penyerah pola pancaran dan penyempit sudut pancaran. Banyaknya *director* pada antena, pola radiasi akan semakin sempit dan jauh.



Gambar 2. 7 Antenna Yagi

Berdasarkan gambar 2.7 ini, Panjang gelombang ( $\lambda$ ) dihitung menggunakan rumus dasar  $\lambda = c/f$ .  $\lambda$  merupakan panjang gelombang yang menentukan dimensi setiap elemen antena yagi. Kemudian,  $c$  adalah kecepatan cahaya ( $3 \times 10^8$  m/s), dan  $f$  adalah frekuensi operasi pada antena.

## 2.5 Perangkat Lunak




Perangkat lunak yang digunakan adalah perangkat lunak yang digunakan untuk membuat program untuk fungsi–fungsi yang ada pada alat *monitoring* ini. Berikut merupakan perangkat lunak yang digunakan:

### 2.5.1 Arduino IDE




Menurut [23][24] Arduino adalah program yang tersedia oleh pengembang Arduino untuk menulis daftar program yang sudah disediakan. Dalam desain perangkat lunak, *software* IDE digunakan untuk menulis program, mengkompilasi kode biner, dan mengupload ke dalam memory mikrokontroler.

IDE merupakan aplikasi yang mengembangkan fungsi-fungsi mikro melalui sintaks berbasis Bahasa C. dan memiliki tab fungsi yang dapat diproses oleh code, pemantauan, dan konfigurasi board secara efisien. Arduino IDE menjadi alat yang sangat berguna bagi pengembang dalam mengembangkan program untuk mikro Arduino. Pada tabel 2.1 merupakan penjelasan mengenai fungsi tab pada *software* Arduino. Berikut adalah tab fungsi:

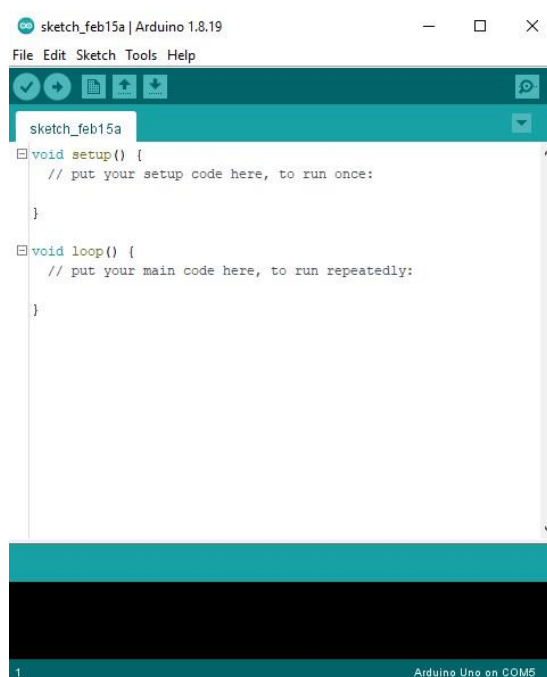
Tabel 2. 2 Tab fungsi pada Arduino IDE

Simbol	Nama	Keterangan
	<i>Verify</i>	Berfungsi untuk mengkompilasi kode program dan mengecek kesalahan sintaksis dan kesalahan lain nya.
	<i>Upload</i>	Berfungsi untuk mengupload sintaks program ke papan Arduino. Setelah kode, telah di kompilasi dan tidak terdapat kesalahan lanjut dengan prose mengunggah sintaks arduino nya
	<i>New</i>	Berfungsi untuk membuat Sketch baru



	<i>Open</i>	Berfungsi untuk membuka sketch yang pernah di buat dan membuka kembali untuk dilakukan editing atau sekedar upload ulang ke Arduino
	<i>Save</i>	Berfungsi untuk menyimpan Sketch yang telah dibuat.
	<i>Serial Monitor</i>	Berfungsi untuk memantau dan berkomunikasi dengan papan Arduino melalui koneksi serial. Dalam tab ini, pengguna dapat melihat data yang dikirim dan diterima oleh papan Arduino melalui koneksi serial, serta mengirimkan perintah ke papan Arduino.

Ketika Arduino IDE pertama kali dijalankan, maka akan menampilkan tampilan seperti berikut:

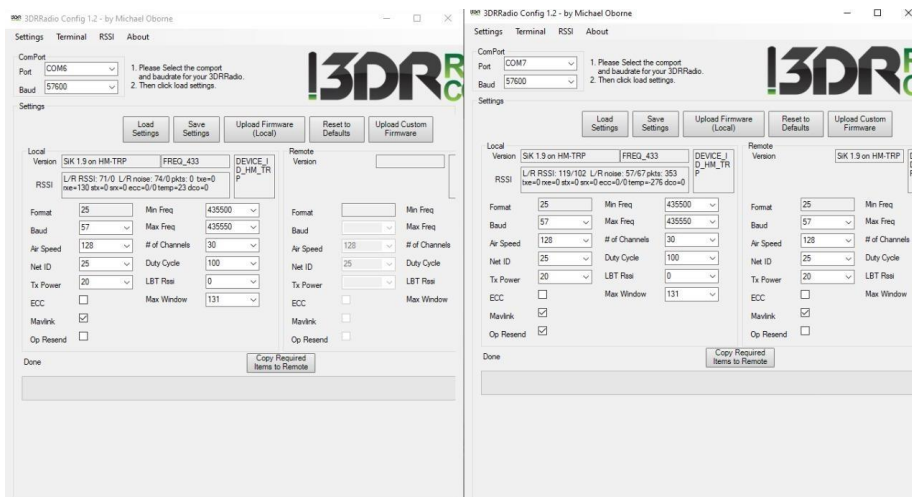


*Gambar 2. 8 Tampilan awal Arduino IDE*

Pada gambar 2.8 merupakan tampilan awal di saat Arduino dibuka.

## 2.5.2 3DR Radio Config

Aplikasi 3DR sangat penting untuk mengoptimalkan jaringan nirkabel pada drone, pesawat tanpa awak, dan kendaraan lainnya. Berdasarkan gambar 2.9 ini fleksibilitas bagi pengguna untuk menyesuaikan frekuensi dan kekuatan sinyal agar jaringan nirkabel berjalan dengan baik dan stabil. Selain itu, aplikasi ini memungkinkan *monitoring* sinyal secara real time dan penyesuaian parameter jaringan jika diperlukan. Dengan pengaturan optimal dan pemantauan terusmenerus, jaringan nirkabel dapat berfungsi dengan baik. Berikut tampilan dari *software* 3DR Radio Config:



Gambar 2. 9 3DR config Radio