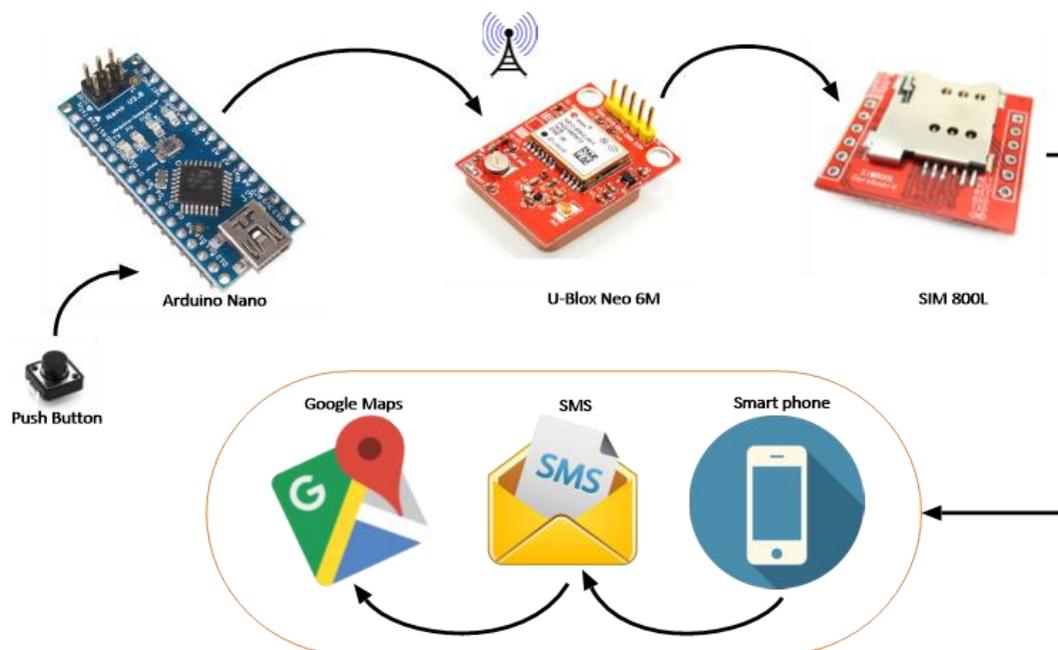


## BAB 5 PERANCANGAN DAN IMPLEMENTASI

Pada bab 5 ini akan membahas tentang perancangan dan implementasi dari Pengembangan Sistem *Tracking Lokasi Low power Sleep* pada *Wearable device*. Tahap perancangan sistem ini merupakan tahap pembuatan berdasarkan analisis yang telah dibuat, dan dilakukan sebelumnya. Dengan perancangan sistem ini diharapkan pengguna bisa dapat lebih mengerti tentang kegunaan sistem.

### 5.1 Diagram Blok Sistem

Pada tahap perancangan diagram blok sistem ini akan menjelaskan mengenai langkah-langkah yang akan dilakukan dalam proses pembuatan Pengembangan Sistem *Tracking Lokasi Low power Sleep* pada *Wearable device*. Dimulai dari perancangan perangkat keras sistem.



**Gambar 5.1 Diagram Blok Sistem**

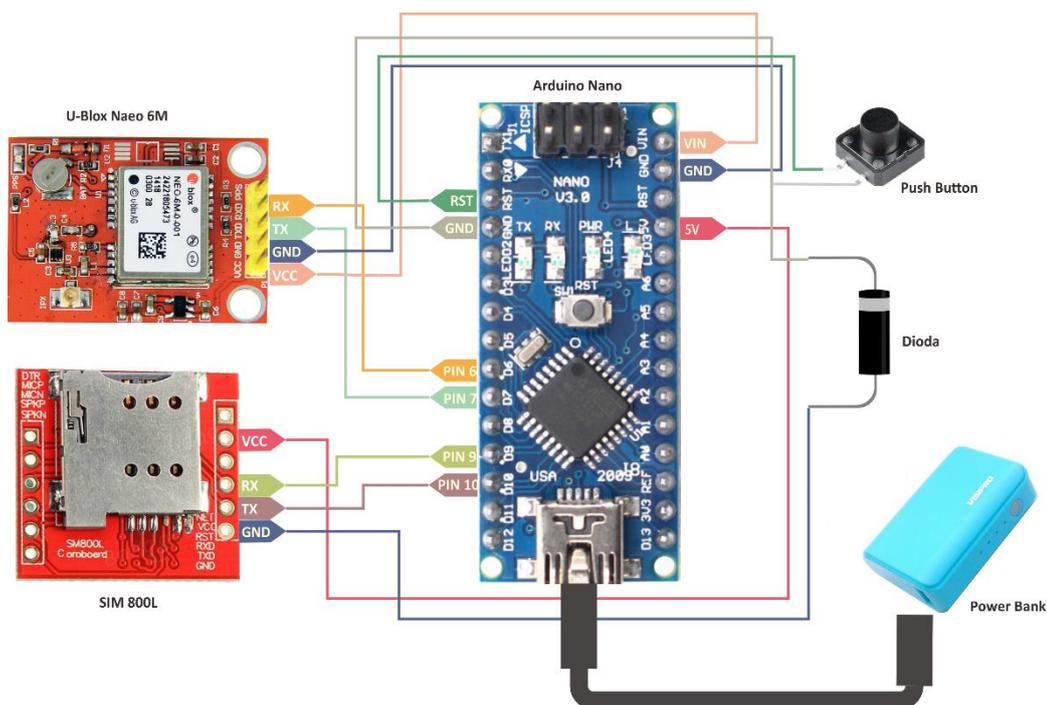
Gambaran umum tentang Pengembangan sistem *tracking lokasi low power sleep* pada *Wearable device* dapat dilihat pada Gambar 5.1. Pada sistem ini menggunakan mikro kontroler Arduino Nano yang akan mengatur keseluruhan sistem pada Pengembangan Sistem *Tracking Lokasi Low Power Sleep* pada *Wearable Device*. Komponen U-Blox Neo 6M yang berfungsi sebagai penangkap sinyal GPS untuk mendapatkan titik koordinat bagi pengguna sistem. Sedangkan komponen SIM 800L yang memiliki fungsi sebagai penangkap sinyal GPRS untuk mengirimkan data titik koordinat kepada pelacak melalui SMS yang berisikan data titik koordinat pengguna sistem. Terdapat dua fitur yang disediakan sistem ketika ingin melakukan pelacakan kepada sistem *tracking lokasi low power sleep* pada *Wearable device* yaitu:

1. Bagi Pengguna Sistem.  
Ketika pengguna sistem merasakan dirinya tersesat atau hilang, maka pengguna dapat menggunakan fitur *Push Button* pada sistem, yang memiliki fungsi mengirimkan data titik koordinat pengguna kepada admin yang telah ditentukan.
2. Bagi Pelacak sistem.  
Ketika pengguna sistem tidak merasa dirinya tersesat atau hilang, tetapi dianggap bahwa dirinya telah hilang menurut orang lain, maka pelacak sistem dapat menggunakan fitur sms pada sistem, dimana sistem dapat mengirimkan data titik koordinat pengguna sistem kepada pelacak.

## 5.2 Perancangan Sistem

Pada sub-bab ini menjelaskan tentang perancangan sistem pada bagian perangkat keras dan perangkat lunak.

### 5.2.1 Perancangan Perangkat Keras



**Gambar 5.2 Skematik Keseluruhan Sistem**

Pada Gambar 5.2 terdapat skematik perangkat keras keseluruhan sistem yang terdiri dari beberapa komponen untuk membuat sistem *tracking* lokasi *low power* pada *Wearable device*. Perangkat keras yang dibutuhkan terdiri dari perangkat Arduino Nano, SIM 800L, U-Blox Neo 6M, *Push Button*, dan Dioda. Seluruh komponen ini akan di implementasikan pada *prototype* yang akan dijadikan *Wearable device*, skematik perangkat keras pada , bahwa Arduino Nano akan disambungkan dengan komponen U-Blox Neo 6M, SIM 800L, dan *Push*

*Button*. Untuk menggabungkan beberapa komponen agar dapat bekerja sesuai sistem, dibutuhkan komponen tambahan pada SIM 800L agar arus yang dibutuhkan komponen SIM 800L tidak melebihi 4,2V. Jika arus yang melewati komponen SIM 800L lebih dari 4,2V maka SIM 800L akan rusak, untuk itu dioda diperlukan untuk menjaga arus yang melewati SIM 800L agar arusnya tidak berlebihan. Sedangkan untuk fungsi dari *push button* adalah untuk memberikan *interrupt* kepada sistem agar sistem terbangun dari metode *sleep power*, dan dapat mengirimkan lokasi titik koordinat pengguna sistem kepada nomor telepon admin. Berikut pin-pin yang digunakan pada skematik keseluruhan sistem:

**Tabel 5.1 Pin skematik keseluruhan sistem**

No	Arduino Nano	SIM 800L	U-Blox Neo 6M
1	PIN 6	-	-PIN RX
2	PIN 7	-	-PIN TX
3	PIN 9	-PIN RX	-
4	PIN 10	-PIN TX	-
5	PIN 5V	-PIN VCC	-
6	PIN VIN	-	-PIN VCC
7	PIN RST	-	-Terhubung dengan kaki <i>Push Button +</i>
8	PIN GND	-	-Terhubung dengan kaki <i>Push Button -</i> -PIN GND

*Power bank* dipakai sebagai sumber daya utama pada sistem *tracking* lokasi *Wearable device* ini. Pada sistem ini *power bank* akan secara langsung disambungkan dengan Arduino Nano untuk mendapatkan daya yang cukup untuk keseluruhan komponen sistem, dan *power bank* yang dipilih pada sistem ini, dikarenakan memiliki kriteria yang dapat memberikan daya sebesar 5V dan memiliki arus sebesar 1 Ampere yang akan digunakan sistem untuk menjalankan proses keseluruhan, selain itu dapat menjaga ketahanan daya pada sistem karena *power bank* ini mampu mendukung sistem kerja *low power*.



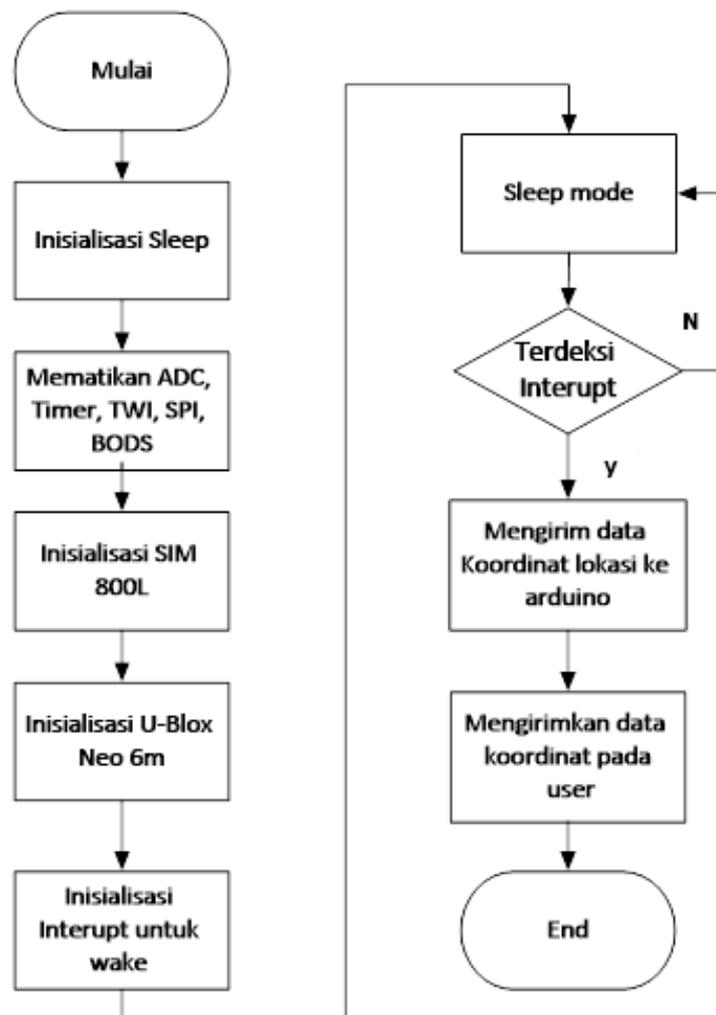
**Gambar 5.3 Spesifikasi Power bank**

Pada Gambar 5.3 dapat dilihat spesifikasi *power bank* yang digunakan pada sistem *tracking* lokasi *low power* pada *Wearable device*. *Power bank* ini memiliki kapasitas sebesar 5200mAh, memiliki input arus DC sebesar 5V, output arus DC sebesar 5V, dan memiliki daya sebesar 1 Ampere untuk input maupun outputnya. Penulis memilih *power bank* yang memiliki daya output sebesar 1 Ampere untuk membantu dalam proses penghematan daya atau membantu dalam proses *low power* pada sistem, karena jika menggunakan *power bank* yang memiliki 2 Ampere maka pemakaian sumber daya pada sistem akan lebih banyak.

### **5.2.2 Perancangan Perangkat Lunak**

Pada sub-bab ini menjelaskan tentang perancangan perangkat lunak sistem pada bagian setiap perangkat keras yang akan digunakan pada sistem dan perangkat lunak sebagai program utama dalam pembuatan sistem. Pada perancangan Pengembangan Sistem *Tracking Lokasi Low Power Sleep* pada *Wearable device* ini memiliki beberapa proses secara umum yang dapat dijabarkan melalui *flowchart*.

### 5.2.3 Perancangan Diagram Perangkat Lunak



**Gambar 5.4 Flowchart Perancangan Sistem Secara Keseluruhan**

Pada Gambar 5.4 menjelaskan bahwa perancangan sistem secara keseluruhan pada pengembangan sistem *tracking* lokasi *low power sleep* pada *Wearable device* ini membutuhkan beberapa inisialisasi terlebih dahulu seperti inisialisasi untuk *sleep power mode* yang dilakukan dengan cara memprogram Arduino Nano menggunakan software Arduino IDE. Perintah yang diberikan oleh Arduino Nano berupa “AT+CSCLK=2” kepada komponen SIM 800L untuk melakukan perintah *sleep power mode* yang berfungsi untuk mematikan fitur ADC, timer, TWI, SPI, dan BODS.

Selain itu sistem ini membutuhkan inisialisasi pada SIM 800L dengan cara mendeklarasikan pin RX dan TX dengan serial pin 9 dan 10 pada Arduino Nano agar dapat saling berkomunikasi. Sedangkan untuk inisialisasi pada U-Blox Neo 6M dengan cara mendeklarasikan pin RX dan TX kepada serial pin 6 dan 7 pada Arduino Nano agar dapat saling berkomunikasi.

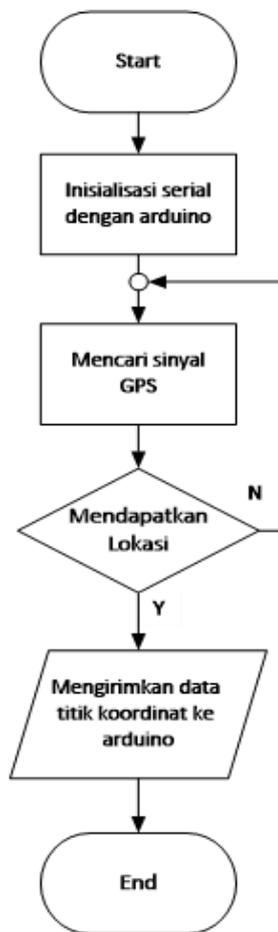
Komponen SIM 800L ini berfungsi untuk mendapatkan sinyal GPRS yang akan digunakan untuk mengirimkan data titik koordinat dari sistem kepada pelacak, sedangkan komponen U-Blox Neo 6M ini berfungsi untuk mendapatkan sinyal GPS yang akan digunakan untuk mendapatkan titik koordinat bagi pengguna sistem ini.

Inisialisasi input string dengan menetapkan kata kunci "LOC" sebagai Interupt yang akan diproses pada Arduino Nano ketika sistem mendapatkan SMS dari seorang pelacak. Selain itu Arduino Nano akan memerintahkan SIM 800L untuk melakukan pengiriman data dengan perintah "AT+CMGF=1" untuk merespon SMS yang masuk pada sistem. Sedangkan untuk interupt berupa *push button* dideklarasikan dengan inisialisai void kirim sms awal. Kedua interupt ini bertujuan untuk membangunkan sistem dari *sleep power mode* agar sistem dapat *melakukan tracking* lokasi.

Sistem ini memiliki fitur *sleep power mode* yang berfungsi untuk menghemat daya yang digunakan ketika sistem berjalan. *Sleep power mode* akan bekerja ketika Arduino Nano memberikan perintah kepada SIM 800L dengan "AT+CSCLK=2". Kondisi ini berlaku ketika tidak terdapat adanya *interupt* berupa SMS yang masuk pada sistem ataupun *push button* yang ditekan, sehingga sistem akan masuk pada *sleep power mode*, dan SIM 800L akan bangun dari *sleep power* sampai terdeteksi adanya interupt yang masuk pada sistem. Agar sistem dapat bekerja kembali secara normal, maka Arduino Nano akan memberikan perintah kepada SIM 800L dengan "AT+CSCLK=0". Kondisi ini berlaku ketika sistem mendapatkan *interupt* berupa SMS ataupun *push button* yang ditekan oleh pengguna.

Sistem ini akan mengirimkan data lokasi titik koordinat pada Arduino Nano ketika komponen U-Blox Neo 6M telah berhasil mendapatkan sinyal GPS untuk menentukan lokasi titik koordinat seorang pengguna sistem, kemudian Arduino Nano akan memerintahkan SIM 800L untuk melakukan pengiriman data titik koordinat yang telah didapatkan kepada pelacak dengan perintah "AT+CMGS".

### 5.2.3.1 Perancangan Diagram U-Blox Neo 6M



**Gambar 5.5 Flowchart Sistem Kerja U-Blox Neo 6M**

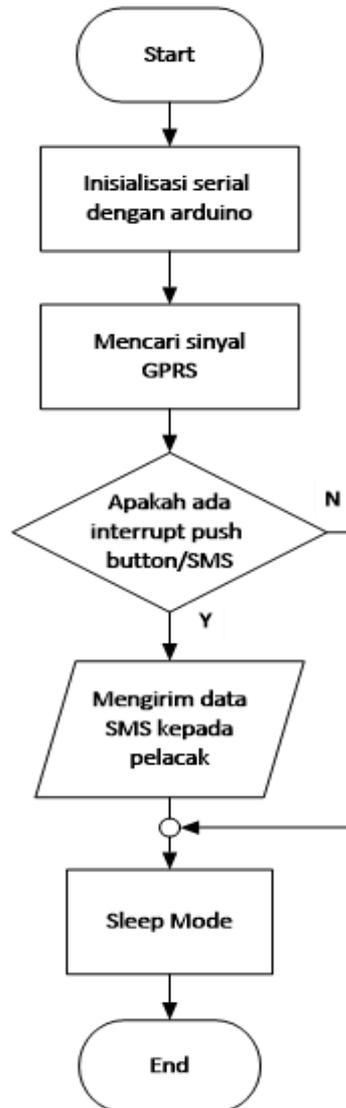
Pada Gambar 5.5 diagram U-Blox Neo 6M akan melakukan inisialisasi terlebih dahulu pada U-Blox Neo 6M dengan cara mendeklarasikan pin RX dan TX kepada serial pin 6 dan 7 pada Arduino Nano agar dapat saling berkomunikasi.

Kemudian U-Blox Neo 6M akan melakukan pencarian sinyal GPS yang bertujuan untuk mendapatkan lokasi titik koordinat pengguna sistem, dimana hasil *tracking* lokasi tersebut berupa titik *longitude* dan titik *latitude*. Untuk dapat mengetahui jika U-Blox Neo 6M telah mendapatkan sinyal GPS ataupun tidak, perlu adanya pendeklarasian pada bagian `GPS.begin`. Pada fungsi `GPS.begin` ini terdapat dua kondisi yang telah di tentukan yaitu ketika U-Blox Neo 6M telah mendapatkan sinyal GPS untuk melakukan *tracking* lokasi, setelah itu hasil *tracking* lokasi tersebut akan dikirimkan kepada Arduino Nano. Sedangkan ketika U-Blox Neo 6M belum mendapatkan sinyal GPS untuk melakukan *tracking* lokasi maka U-Blox Neo 6M akan terus mencari sinyal GPS sampai dapat.

Data titik koordinat yang telah didapatkan dari komponen U-Blox Neo 6M ini akan diproses pada fungsi void `gpsdump "TinyGPS & gps"`, kemudian setelah itu data lokasi titik koordinat tersebut akan dikirimkan kepada Arduino Nano untuk

dilanjutkan prosesnya pada komponen SIM 800L melalui perintah “AT+CMGS” dari Arduino Nano untuk melakukan pengiriman data lokasi titik koordinat pengguna yang telah didapat dari komponen U-Blox Neo 6M. Sedangkan jika komponen U-Blox Neo 6M belum mendapatkan sinyal GPS, maka U-Blox Neo 6M tidak akan mengirimkan data titik koordinat kepada Arduino Nano sampai mendapatkan sinyal GPS untuk melakukan *tracking* lokasi.

### 5.2.3.2 Perancangan Diagram SIM 800L



**Gambar 5.6 Flowchart Sistem Kerja SIM 800L**

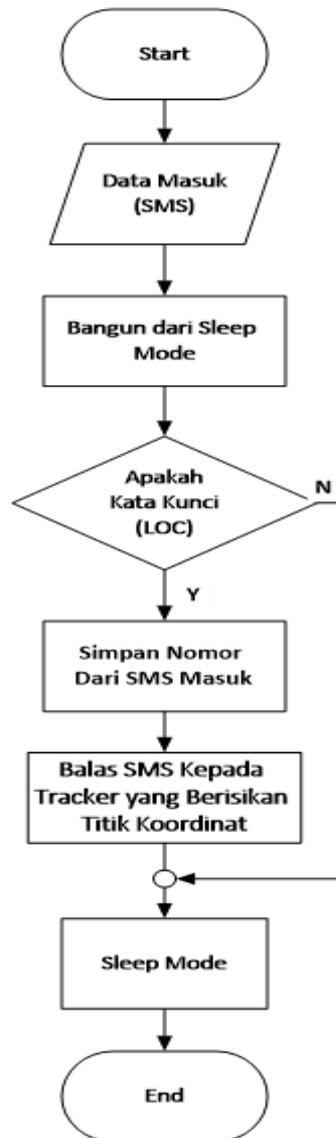
Pada Gambar 5.6 diagram SIM 800L dapat dilihat bahwa komponen ini membutuhkan inisialisasi terlebih dahulu dengan cara mendeklarasikan pin RX dan TX dengan serial pin 9 dan 10 pada Arduino Nano agar dapat saling berkomunikasi.

Kemudian SIM 800L ini akan melakukan proses pencarian sinyal GPRS dari SIM *card provider* yang dipasangkan pada komponen SIM 800L, dimana proses ini dapat dilihat pada fungsi “void sim800 setup()”.

*Interrupt* yang dapat di deteksi oleh sistem ini berupa SMS yang masuk pada sistem ataupun *push button* yang ditekan. *Interrupt* ini diberikan karena pada SIM 800L ini diterapkan metode *sleep power* untuk melakukan penghematan daya yang akan dipakai ketika sistem sedang bekerja. Sistem *sleep power mode* ini akan bekerja ketika Arduino Nano memberikan perintah kepada SIM 800L dengan “AT+CSCLK=2”. Kondisi ini terjadi ketika sistem tidak menerima adanya *interrupt* berupa SMS yang masuk pada sistem ataupun *push button* yang ditekan, sehingga sistem akan masuk pada *sleep power mode*, dan SIM 800L akan bangun dari *sleep power* sampai terdeteksi adanya *interrupt* yang masuk pada sistem. Agar sistem dapat bekerja kembali secara normal, maka Arduino Nano akan memberikan perintah kepada SIM 800L dengan “AT+CSCLK=0”. Kondisi ini berlaku ketika sistem mendapatkan *interrupt* berupa SMS ataupun *push button* yang ditekan oleh pengguna.

Setelah SIM 800L telah mendapatkan sinyal GPRS, maka komponen SIM 800L telah siap untuk melakukan pengiriman data lokasi titik koordinat yang telah didapatkan dari komponen U-Blox Neo 6M kepada pelacak. Kemudian Arduino Nano akan memerintahkan SIM 800L untuk melakukan pengiriman data titik koordinat yang telah didapatkan kepada pelacak dengan perintah “AT+CMGS”. Setelah selesai melakukan proses pengiriman data lokasi titik koordinat pengguna sistem kepada pelacak dan tidak ada perintah lain ataupun *interrupt* yang masuk kepada sistem selama 10 detik, maka SIM 800L akan secara langsung memasuki *sleep power mode* melalui perintah yang diberikan dari Arduino Nano kepada SIM 800L dengan “AT+CSCLK=2”.

### 5.2.3.3 Perancangan Diagram Perintah SMS Tracking



**Gambar 5.7 Flow Chart Sistem Kerja Perintah SMS**

Pada Gambar 5.7 dapat dilihat bahwa ketika pelacak ingin melakukan perintah untuk melakukan *tracking* lokasi, pelacak harus melakukan pengiriman SMS kepada nomor yang terdapat pada sistem. Kemudian sistem akan membaca bahwa terdapat *interrupt* yang masuk kepada sistem berupa SMS.

Setelah itu SIM 800L akan bangun dari *sleep power mode* karena diperintah oleh Arduino Nano berupa "AT+CSCLK=0", kondisi ini terjadi karena terdeteksi adanya data masuk kepada sistem berupa SMS dari pelacak.

Sistem akan melakukan pengecekan pada isi dari SMS yang masuk, apakah isi SMS yang dikirimkan pada sistem sudah sesuai dengan kata kunci yang di tentukan, untuk melakukan pengecekan isi pesan yang masuk kepada sistem, maka Arduino Nano memberi perintah ke SIM 800L dengan AT COMMAND "CMT".

Jika isi SMS sesuai dengan kata kunci yang telah di tentukan yaitu “LOC”, setelah itu Arduino Nano akan memberikan perintah kepada SIM 800L untuk melakukan pembacaan nomor telepon yang masuk dan menyimpannya untuk sementara waktu, kondisi ini terdapat pada fungsi “string simpan nomor”. Kemudian U-Blox Neo 6M akan memberikan hasil data *tracking* lokasi pengguna sistem, dan akan mengirimkan hasil *tracking* lokasi tersebut kepada pelacak melalui komponen SIM 800L dengan perintah “AT+CMGS” berupa SMS data link titik koordinat lokasi pengguna sistem.

Setelah sistem melakukan seluruh proses ataupun perintah, maka komponen SIM 800L akan kembali kepada *sleep power mode* dengan perintah yang diberikan dari Arduino Nano melalui “AT+CSCLK=2” dan SIM 800L akan kembali bangun dari *sleep power mode* ketika terdeteksi adanya *interrupt* yang masuk kepada sistem berupa SMS ataupun *push button* yang ditekan.

### 5.3 Implementasi Sistem

Pada sub bab implementasi sistem ini akan dijelaskan tentang penerapan perancangan perangkat lunak dan perangkat keras.

#### 5.3.1 Implementasi Perangkat Keras

Pada implementasi perangkat keras akan dijelaskan proses merealisasikan hasil perancangan agar menjadi sebuah sistem yang layak untuk digunakan. Sistem ini menggunakan beberapa komponen yang memiliki fungsi masing-masing, yaitu Arduino Nano yang berfungsi sebagai mikrokontroler yang mengatur input ataupun output dari keseluruhan sistem, SIM 800L digunakan untuk melakukan pencarian sinyal GPRS yang bertujuan untuk melakukan pengiriman SMS kepada user, sedangkan U-Blox Neo 6M digunakan untuk melakukan pencarian sinyal GPS yang berfungsi untuk mendapatkan titik koordinat bagi pengguna sistem, dan *Power bank* yang digunakan sebagai sumber daya bagi keseluruhan sistem.



Gambar 5.8 Implementasi *Prototype* Tampak Atas

Pada Gambar 5.8 dapat dilihat bentuk implementasi *prototype* sistem tampak atas, *prototype* alat ini dapat dibawa dengan mudah ketika berpergian oleh pengguna sistem, sehingga memudahkan pelacak untuk melakukan *tracking* lokasi kepada pengguna karena alat ini bersifat *Wearable device*.



**Gambar 5.9 Implementasi *Prototype* Tampak Samping**

Pada Gambar 5.9 dapat dilihat bentuk implementasi *prototype* sistem tampak samping, peletakan komponen pada casing *prototype* ini sudah didesain agar memudahkan pengguna untuk melihat apakah SIM 800L dan U-Blox Neo 6M telah mendapatkan sinyal untuk melakukan komunikasi, selain itu desain ini memudahkan pengguna sistem ketika ingin mengganti SIM *card* provider pada sistem.

### 5.3.1.1 Implementasi Desain Perangkat Elektronik

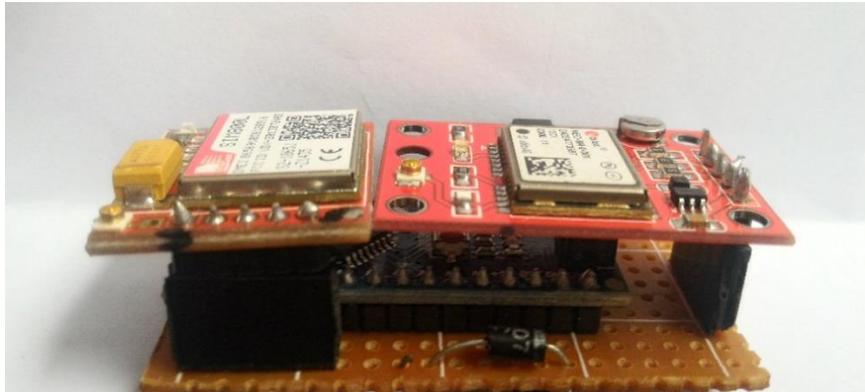
Implementasi desain perangkat keras sistem ini dibuat berdasarkan pada sub bab 2 dan sub bab 3, dengan membuat desain sekecil mungkin pada perangkat keras dengan tujuan agar dapat memudahkan bagi pengguna sistem untuk dibawa berpergian dan untuk mempermudah *tracking* lokasi ketika sistem dibawa pengguna ketika berpergian karena perangkat keras dipakai oleh pengguna sistem ketika berpergian.



**Gambar 5.10 Desain *Hardware* Tampak Atas**

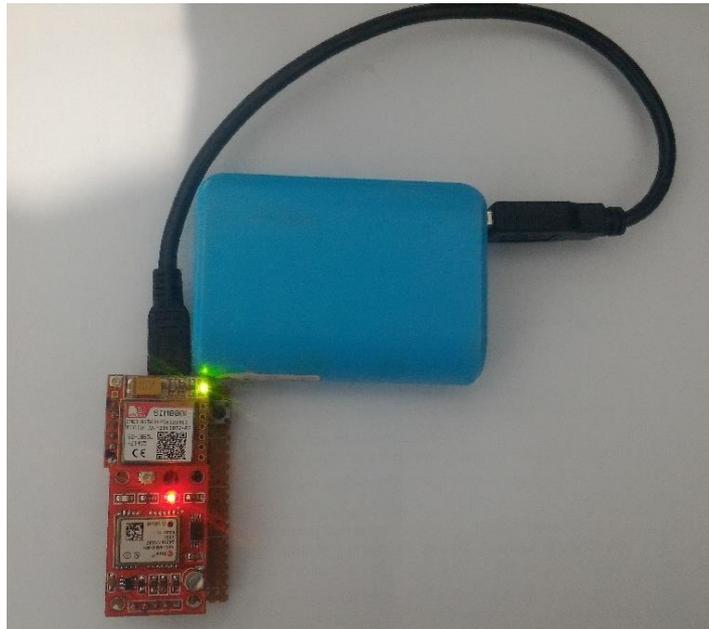
Pada Gambar 5.10 dapat dilihat bahwa implementasi perangkat keras keseluruhan dari tampak atas yang dibutuhkan pada sistem *tracking* lokasi *low*

*power* sistem pada *Wearable device*. Komponen yang digunakan pada sistem ini yaitu Arduino Nano, U-Blox Neo 6M, SIM 800L, Dioda, dan *Push Button*.



**Gambar 5.11 Desain *Hardware* Tampak Samping**

Pada Gambar 5.11 dapat dilihat keseluruhan komponen tampak samping yang akan dibutuhkan pada pengembangan sistem *tracking* lokasi *low power* sistem pada *Wearable device*. Komponen-komponen yang digunakan pada sistem ini disusun secara bertumpuk untuk menghemat tempat agar sistem dapat mudah dibawa oleh pengguna sistem ketika berpergian.



**Gambar 5.12 *Power bank* Terhubung Dengan Sistem**

Pada Gambar 5.12 dapat dilihat bahwa *Power bank* dengan keseluruhan komponen pada sistem ini tersambung secara langsung dengan Arduino Nano dengan tujuan sebagai sumber daya utama pada sistem *tracking* lokasi ini, dan *Power bank* akan memberikan daya untuk menghidupkan seluruh komponen yang ada pada sistem *tracking* lokasi ini. Daya yang akan diberikan *Power bank* kepada komponen yang ada pada sistem sebesar 1 Ampere dengan arus sebesar 5V.

### 5.3.2 Implementasi Perangkat Lunak

Pada implementasi perangkat lunak akan dijelaskan proses merealisasikan hasil perancangan perangkat lunak pada sub bab sebelumnya.

#### 5.3.2.1 Implementasi Perangkat Lunak Pada Arduino Nano

**Tabel 5.2 Source Code Pemanggilan Library GPS dan SoftwareSerial**

1	#include <TinyGPS.h>
2	#include <SoftwareSerial.h>
3	SoftwareSerial mySerial(9, 10);
4	SoftwareSerial GPS(6, 7);
5	
6	TinyGPS gps;

Pada Tabel 5.2 *source code* pada baris 1 sampai 4 ini memiliki fungsi sebagai inisialisasi library yang akan digunakan pada sistem. Untuk melakukan komunikasi dengan komponen U-Blox Neo, membutuhkan *library* TinyGPS.h, sedangkan untuk komponen SIM 800L membutuhkan library SoftwareSerial.h. Komponen U-Blox Neo 6M terdapat pin RX, dan TX yang akan disambungkan dengan arduino nano pada port 6 dan 7. Sedangkan untuk komponen SIM 800L pin RX dan TX akan disambungkan dengan arduino nano pada port 9 dan 10.

**Tabel 5.3 Source Code Penyimpanan Nomor Telpn Admin**

1	char phone_no[] = "081230291548";
2	
3	void gpsdump(TinyGPS &gps);
4	void printFloat(double f, int digits = 2);
5	const byte interrupt = 2;
6	char incomingByte;
7	String inputString;
8	int led = 8;
9	bool state = true;
10	String pelaku;
11	String inchar;
12	char digit1;
13	char digit2;
14	char digit3;
15	char digit4;
16	char digit5;
17	char digit6;
18	char digit7;
19	char digit8;
20	char digit9;
21	char digit10;
22	char digit11;
23	char digit12;

Pada Tabel 5.3 *source code* baris 1 sampai 11 ini menjelaskan tentang bagaimana sistem menginisialisasikan nomor telepon admin untuk menentukan proses pengiriman data titik koordinat melalui SMS ketika *Push Button* pada sistem di tekan. Sedangkan untuk baris 12 sampai 23 memiliki fungsi sebagai pembacaan nomor telepon admin sebanyak 12 digit.

**Tabel 5.4 Source Code Pengiriman SMS dan Telepon**

```

1 void setup()
2 {
3   pinMode(led, OUTPUT);
4   //Serial.begin(9600);
5   delay(1000);
6
7   KirimSmsAwal();
8
9   mySerial.begin(9600);
10  mySerial.print("AT+CLIP=1\r");
11  delay(100);
12  delay(1000);
13  mySerial.println("AT+CMGDA=\"DEL ALL\"");
14  delay(1000);
15  sim800setup();
16 }

```

Pada Tabel 5.4 *source code* ini memiliki fungsi inialisasi void setup untuk komponen SIM 800L, dan memberikan perintah AT *command* kepada SIM 800L agar dapat membaca SMS masuk dan mengirimkan SMS ketika SIM 800L telah mendapatkan sinyal GPRS.

Perintah AT+CLIP merupakan perintah untuk pembacaan adanya telepon yang masuk pada SIM 800L, sedangkan untuk perintah AT+CMGDA="DEL ALL" untuk melakukan penghapusan seluruh SMS yang masuk pada sistem.

**Tabel 5.5 Source Code Pembacaan Nomor Telepon dan SMS**

```

1 void loop()
2 {
3   if (isproses) {
4     if (mySerial.available()) {
5       delay(100);
6       String SimpanNomor = "";
7
8       while (mySerial.available()) {
9         incomingByte = mySerial.read();
10        inputString += incomingByte;
11      }
12
13      delay(10);
14      inchar = inputString;
15      inputString.toUpperCase();
16      Serial.println(inputString);
17      if (inputString.indexOf("CMT") >= 0) {
18        Serial.println("=====");
19        Serial.println(inchar);
20        Serial.println("=====");
21      SimpanNomor = String(inputString.charAt(72)) +
22      String(inputString.charAt(73)) +
23      String(inputString.charAt(74))
24      + String(inputString.charAt(75)) +
25      String(inputString.charAt(76)) +
26      String(inputString.charAt(77)) +
27      String(inputString.charAt(78))
28      + String(inputString.charAt(79)) +
29      String(inputString.charAt(80)) +

```

29	String(inputString.charAt(81))	+
30	String(inputString.charAt(82)) +	
31	String(inputString.charAt(83))	+
32	String(inputString.charAt(84))	+
33	String(inputString.charAt(85));	
34	digit1 = '0';	
35	digit2 = SimpanNomor.charAt(3); delay(100);	
36	digit3 = SimpanNomor.charAt(4); delay(100);	
37	digit4 = SimpanNomor.charAt(5); delay(100);	
38	digit5 = SimpanNomor.charAt(6); delay(100);	
39	digit6 = SimpanNomor.charAt(7); delay(100);	
40	digit7 = SimpanNomor.charAt(8); delay(100);	
41	digit8 = SimpanNomor.charAt(9); delay(100);	
42	digit9 = SimpanNomor.charAt(10); delay(100);	
43	digit10 = SimpanNomor.charAt(11); delay(100);	
44	digit11 = SimpanNomor.charAt(12); delay(100);	
45	digit12 = SimpanNomor.charAt(13); delay(100);	
46	Serial.println("=====");	
47		
48	mySerial.println("AT;");	
49	delay(500);	
50	mySerial.println("AT;");	
51	delay(500);	
52	mySerial.println("AT;");	
53	delay(500);	
54	mySerial.println("AT;");	
55	delay(500);	
56	mySerial.println("AT;");	
57	delay(500);	
58	mySerial.println("AT;");	
59	delay(500);	
60	isproses = false;	
61	mySerial.println("AT;");	
62	delay(500);	
63	mySerial.println("AT+CSCLK=0;");	
64	delay(1000);	
65	mySerial.println("ATZ;");	
66	delay(500);	
67	delay(100);	

Pada *source code* Tabel 5.5 menjelaskan bagaimana cara sistem akan melakukan respon untuk melakukan pengiriman data SMS kepada pelacak dengan cara menyimpan nomor telepon yang masuk kepada sistem, dan melakukan pembacaan isi pesan dari SMS yang masuk kepada sistem.

Pada *source code* baris 14 sampai 47 menjelaskan tentang bagaimana sistem akan melakukan pembacaan isi dari SMS yang dikirimkan pelacak, selanjutnya akan membaca nomor telepon pelacak dan menyimpan sementara untuk melakukan pengiriman data. *Source code* pada baris 49 sampai 68 berfungsi sebagai perintah yang diberikan untuk membangunkan SIM 800L dari *sleep power mode* dengan tujuan untuk melakukan pengiriman data SMS, setelah SIM 800L selesai melakukan pengiriman maka memori SIM 800L di reset ulang secara keseluruhan.

Perintah AT berfungsi sebagai perintah untuk melakukan aktifitas seperti stimulus. Perintah AT+CSCLK=0 ini berfungsi untuk membangunkan SIM 800L dari keadaan *sleep power mode*, kemudian perintah ATZ untuk mereset memori keseluruhan pada SIM 800L, dan CMT berfungsi untuk mendeteksi isi dari SMS yang di terima oleh SIM 800L.

**Tabel 5.6 Source Code Inisialisasi Kata Kunci**

1	if (inputString.indexOf("LOC") >= 0) {
2	panggilmySerial();
3	} else {
4	mySerial.println("AT+CMGF=1");
5	delay(2000);
6	mySerial.print("AT+CMGS=\"");
7	mySerial.print(digit1);    delay(100);
8	mySerial.print(digit2);    delay(100);
9	mySerial.print(digit3);    delay(100);
10	mySerial.print(digit4);    delay(100);
11	mySerial.print(digit5);    delay(100);
12	mySerial.print(digit6);    delay(100);
13	mySerial.print(digit7);    delay(100);
14	mySerial.print(digit8);    delay(100);
15	mySerial.print(digit9);    delay(100);
16	mySerial.print(digit10);   delay(100);
17	mySerial.print(digit11);   delay(100);
18	mySerial.print(digit12);   delay(100);
19	mySerial.write(0x22);
20	mySerial.write(0x0D);
21	mySerial.write(0x0A);
22	delay(2000);
23	mySerial.print("Keyword yang dimasukan salah");
24	delay(500);
25	mySerial.println(char(26));
26	delay(2000);
27	mySerial.begin(9600);
28	delay(1000);
29	}
30	sim800setup();
31	
32	delay(100);
33	inchar = "";
34	delay(10);
35	inputString = "";
36	delay(10);
37	SimpanNomor = "";
38	delay(20);
39	}
40	}
41	
42	}

Pada Tabel 5.6 *source code* ini menjelaskan tentang bagaimana sistem mampu menentukan kata kunci yang akan digunakan dalam melakukan *tracking* lokasi, kata kunci yang di tetapkan yaitu "LOC". Saat sistem membaca isi dari SMS yang di kirimkan, tetapi tidak sesuai dengan kata kunci maka sistem akan tetap merespon SMS dengan berisikan kata-kata "Keyword yang dimasukan salah".

Pada *source code* baris 1 menjelaskan bahwa sistem di berikan ketentuan untuk pembacaan kata "LOC", sedangkan untuk baris 3 sampai 42 adalah perintah untuk mendeteksi adanya pesan yang masuk pada sistem serta membaca nomor telepon yang mengirimkan pesan, dan membaca kata pada isi pesan tidak sesuai dengan "LOC" maka sistem akan melakukan respon melalui SMS kepada pelacak dengan memberitahukan bahwa "Keyword yang dimasukan salah".

Perintah AT yang digunakan pada *source code* ini adalah AT+CMGF=1 yang memiliki fungsi sebagai pendeteksi adanya pesan yang masuk pada sistem, dan melakukan pembacaan isi pesan yang diterima, sedangkan untuk AT+CMGS berfungsi untuk melakukan pengiriman SMS.

**Tabel 5.7 Source Code Sleep Power Mode**

1	if (!isproses) {
2	digitalWrite(led, LOW);
3	mySerial.println("AT+CSCLK=2");
4	isproses = true;
5	}
6	}

Pada *source code* Tabel 5.7 memiliki fungsi untuk melakukan perintah *sleep power mode* pada komponen SIM 800L ketika sedang tidak ada proses apapun yang dilakukan, dengan tujuan penghematan daya yang digunakan ketika sistem sedang berjalan.

Pertintah AT yang digunakan adalah AT+CSCLK=2 perintah ini berfungsi untuk melakukan *sleep power mode* pada komponen SIM 800L.

**Tabel 5.8 Source Code Untuk Menerima Data GPS**

1	void panggilmySerial() {
2	GPS.begin(9600);
3	delay(1000);
4	bool newdata = false;
5	unsigned long start = millis();
6	while (millis() - start < 5000)
7	{
8	if (GPS.available())
9	{
10	char c = GPS.read();
11	if (gps.encode(c))
12	{
13	newdata = true;
14	break;
15	}
16	}
17	}
18	}

Pada *source code* Tabel 5.8 memiliki fungsi untuk melakukan perintah pembacaan data yang didapatkan dari U-Blox Neo 6M ketika sudah mendapatkan sinyal GPS untuk melakukan *tracking* lokasi, dan mendapatkan data berupa titik koordinat pengguna sistem yang nantinya akan dikirimkan kepada pelacak.

**Tabel 5.9 Source Code Merespon SMS yang Masuk**

1	if (newdata) {
2	mySerial.println("AT+CMGF=1");
3	delay(2000);
4	mySerial.print("AT+CMGS=\"");
5	mySerial.print(digit1); delay(100);
6	mySerial.print(digit2); delay(100);
7	mySerial.print(digit3); delay(100);
8	mySerial.print(digit4); delay(100);
9	mySerial.print(digit5); delay(100);
10	mySerial.print(digit6); delay(100);
11	mySerial.print(digit7); delay(100);
12	mySerial.print(digit8); delay(100);
13	mySerial.print(digit9); delay(100);
14	mySerial.print(digit10); delay(100);
15	mySerial.print(digit11); delay(100);
16	mySerial.print(digit12); delay(100);
17	mySerial.write(0x22);
18	mySerial.write(0x0D);
19	mySerial.write(0x0A);
20	delay(2000);
21	gpsdump(gps);
22	delay(500);
23	mySerial.println(char(26));
24	delay(2000);
25	mySerial.begin(9600);
26	delay(1000);
27	}
28	else {
29	mySerial.println("AT+CMGF=1");
30	delay(2000);
31	mySerial.print("AT+CMGS=\"");
32	mySerial.print(digit1); delay(100);
33	mySerial.print(digit2); delay(100);
34	mySerial.print(digit3); delay(100);
35	mySerial.print(digit4); delay(100);
36	mySerial.print(digit5); delay(100);
37	mySerial.print(digit6); delay(100);
38	mySerial.print(digit7); delay(100);
39	mySerial.print(digit8); delay(100);
40	mySerial.print(digit9); delay(100);
41	mySerial.print(digit10); delay(100);
42	mySerial.print(digit11); delay(100);
43	mySerial.print(digit12); delay(100);
44	mySerial.write(0x22);
45	mySerial.write(0x0D);
46	mySerial.write(0x0A);
47	delay(2000);
48	mySerial.print("Belum Mendapatkan Sinyal GPS");
49	delay(500);
50	mySerial.println(char(26));
51	delay(2000);
52	mySerial.begin(9600);
53	delay(1000);
54	}
55	}

Pada *source code* Tabel 5.9 memiliki 2 fungsi sebagai perintah yang akan dijalankan ketika sistem sedang bekerja, perintah 1 ketika sistem menerima SMS dari pelacak ketika keseluruhan sistem telah siap, dan isi dari pesan yang dikirimkan sesuai dengan kata kunci yang di berikan maka sistem akan melakukan pengiriman hasil link data titik koordinat. Sedangkan perintah 2 ketika sistem menerima SMS dari pelacak ketika sistem belum siap secara keseluruhan, maka sistem akan memberitahukan pelacak melalui balasan SMS yang berisikan “Belum mendapatkan sinyal GPS”.

Perintah AT yang digunakan pada *source code* ini adalah AT+CMGF=1 yang memiliki fungsi sebagai pendeteksi adanya pesan yang masuk pada sistem, dan melakukan pembacaan isi pesan yang diterima, sedangkan untuk AT+CMGS berfungsi untuk melakukan perintah pengiriman SMS.

**Tabel 5.10 Source Code Tracking Lokasi**

```

1 void gpsdump(TinyGPS &gps)
2 {
3     long lat, lon;
4     float flat, flon;
5     gps.f_get_position(&flat, &flon);
6     mySerial.print("Anda    Dapat    Mencari    Saya    di:
7 https://www.google.co.id/maps/place/"); printFloat(flat, 5);
8 mySerial.print(","); printFloat(flou, 5);
9 }
10 void printFloat(double number, int digits)
11 {
12     if (number < 0.0)
13     {
14         mySerial.print('-');
15         number = -number;
16     }
17     double rounding = 0.5;
18     for (uint8_t i = 0; i < digits; ++i)
19         rounding /= 10.0;
20     number += rounding;
21     unsigned long int_part = (unsigned long)number;
22     double remainder = number - (double)int_part;
23     mySerial.print(int_part);
24
25     if (digits > 0)
26         mySerial.print(".");
27     while (digits-- > 0)
28     {
29         remainder *= 10.0;
30         int toPrint = int(remainder);
31         mySerial.print(toPrint);
32         remainder -= toPrint;
33     }
34 }

```

Pada *source code* Tabel 5.10 memiliki fungsi sebagai perintah untuk mendapatkan titik koordinat pengguna dengan menggunakan komponen U-Blox Neo 6M. Titik longitude dan titik latitude dari pengguna sistem akan didapatkan

ketika komponen U-Blox Neo 6M telah mendapatkan sinyal GPS, dan hasil *tracking* lokasi tersebut akan diolah menjadi sebuah SMS. Pesan yang akan dikirimkan berisikan sebuah yang dapat langsung tersambung dengan aplikasi Google Maps.

**Tabel 5.11 Source Code Pengiriman Hasil Tracking Lokasi dan Setup SIM 800L**

```

1 void sim800setup() {
2   mySerial.println("AT+CMGF=1");
3   delay(1000);
4   mySerial.println("AT+CNMI=1,2,0,0,0");
5   delay(1000);
6   mySerial.println("AT+CMGL=\"REC UNREAD\"");
7   delay(500);
8 }
9
10 void KirimSmsAwal() {
11   GPS.begin(9600);
12   delay(1000);
13   bool newdata = false;
14   unsigned long start = millis();
15   while (millis() - start < 5000)
16   {
17     if (GPS.available())
18     {
19       char c = GPS.read();
20       if (gps.encode(c))
21       {
22         newdata = true;
23         break;
24       }
25     }
26   }
27   if (newdata) {
28     mySerial.begin(9600); delay(15);
29     mySerial.println("AT+CMGF=1");
30     delay(2000);
31     mySerial.print("AT+CMGS=\"");
32     mySerial.print(phone_no); delay(100);
33     mySerial.write(0x22);
34     mySerial.write(0x0D);
35     mySerial.write(0x0A);
36     delay(2000);
37     gpsdump(gps);
38     delay(500);
39     mySerial.println(char(26));
40     delay(2000);
41     mySerial.begin(9600);
42     delay(1000);
43   }
44 }

```

Pada *source code* Tabel 5.11 memiliki fungsi sebagai perintah untuk menyiapkan komponen SIM 800L untuk menerima data dari komponen U-Blox Neo 6M. SIM 800L akan melakukan pembacaan informasi berupa nomor telepon, waktu, dan tanggal ketika ada pesan yang masuk pada sisem, dan melakukan pengiriman data SMS yang berisikan link data titik koordinat pengguna sistem.

Perintah AT yang digunakan pada *source code* ini adalah AT+CMGF=1 yang memiliki fungsi sebagai pendeteksi adanya pesan yang masuk pada sistem, dan melakukan pembacaan isi pesan yang diterima, untuk AT+CMGS berfungsi untuk melakukan perintah pengiriman SMS, sedangkan untuk AT+CNMI memiliki fungsi perintah untuk melakukan penyimpanan SMS yang masuk kepada sistem, dan AT+CMGL "REC UNREAD" berfungsi untuk mengetahui info SMS yang masuk ke sistem, seperti pembacaan nomor telepon yang masuk, waktu dan tanggal SMS di terima oleh sistem.