


## BAB IV

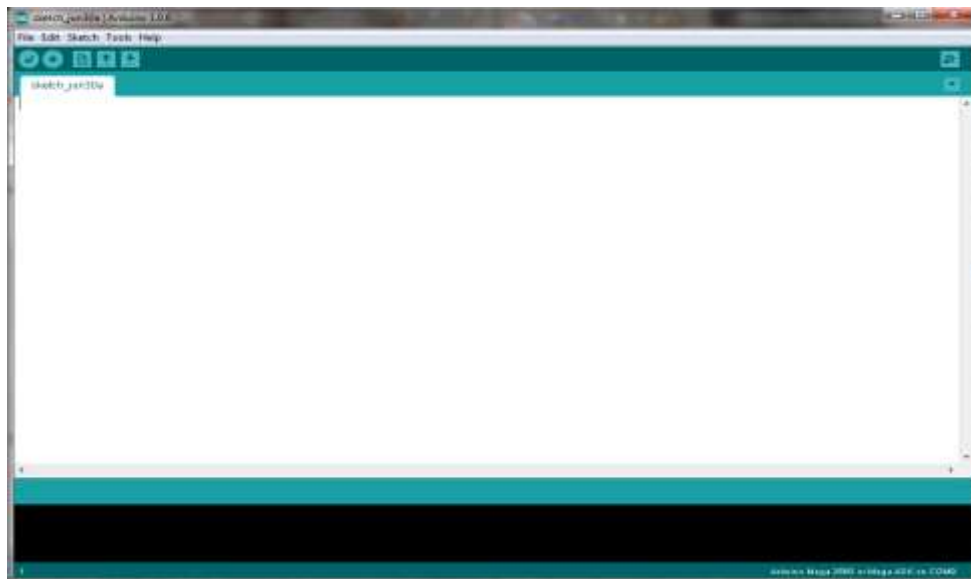
### HASIL DAN UJI COBA

#### IV.1. Software

Software arduino merupakan software yang sangat penting karena merupakan proses penginputan data dari computer kedalam mikrokontroler arduino menggunakan software arduino IDE.

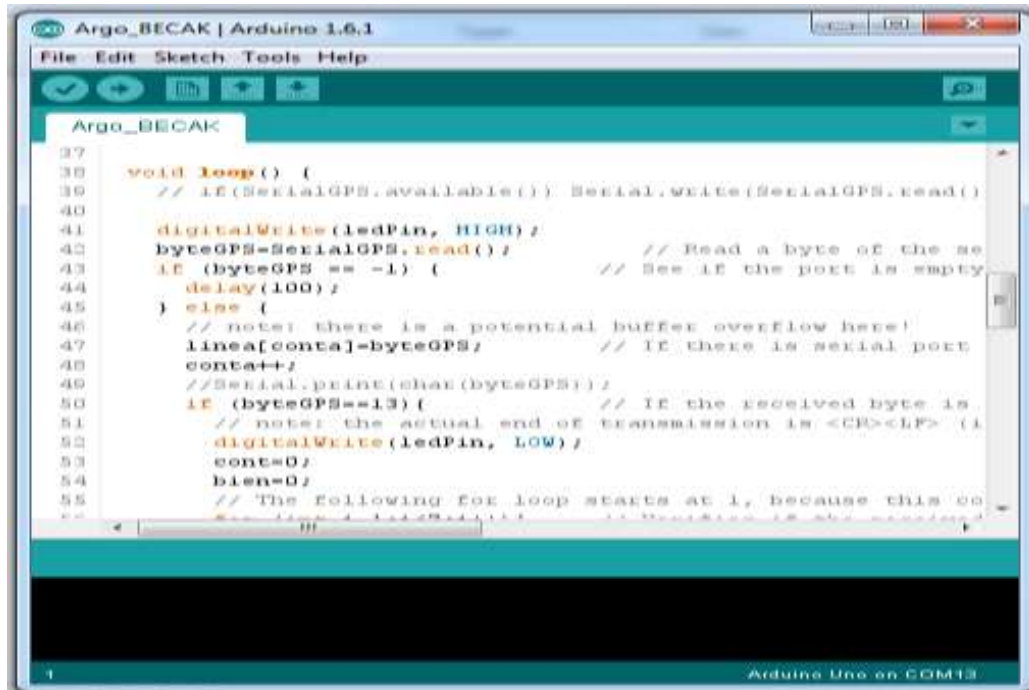
Dalam proses instalasi ini menggunakan aplikasi IDE Arduino Untuk melakukan instalasi inidapat dilakukan dengan beberapa langkah antara lain :

1. Langkah pertama yang dilakukan adalah menjalankan *software* IDE Arduino dengan mengklik *icon*  Setelah program melakukan *load* maka akan terlihat bentuk tampilan seperti gambarIV.1.



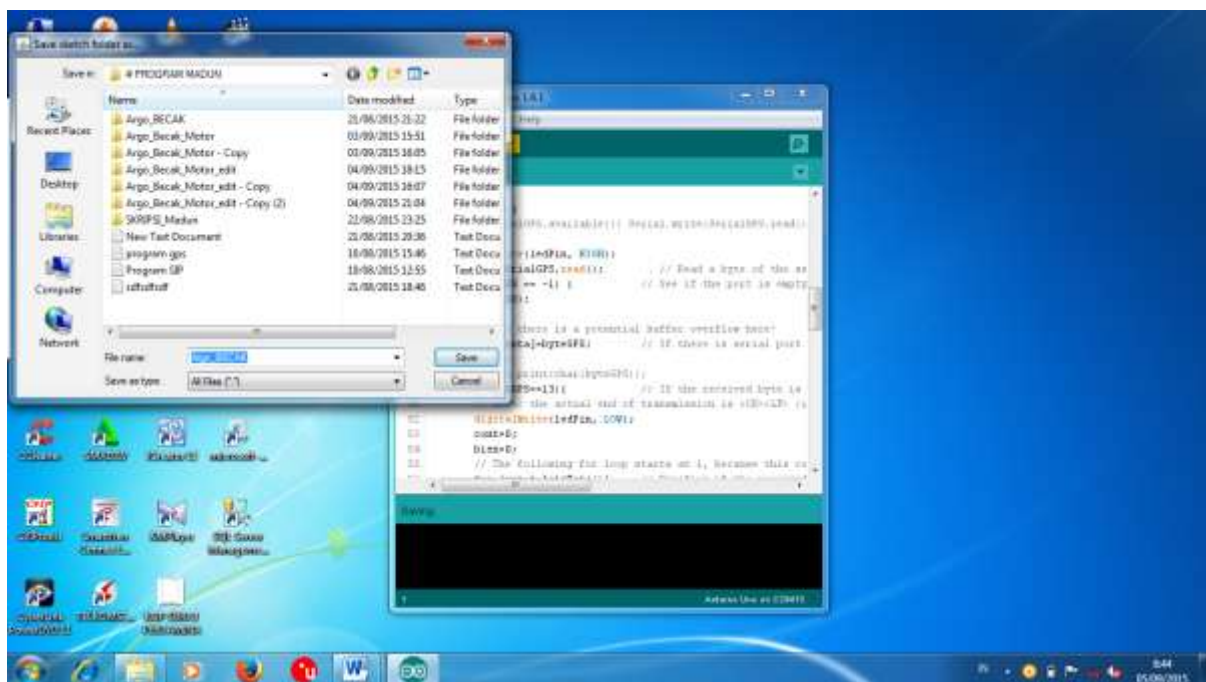
**Gambar IV.1. Tampilan *Software* IDE Arduino.**

2. Selanjutnya untuk memprogram Mikrokontroler Atmega328 Arduino yaitu dengan mengetikkan program sesuai dengan yang dibutuhkan pada alat. Seperti yang terlihat pada gambarIV.2.



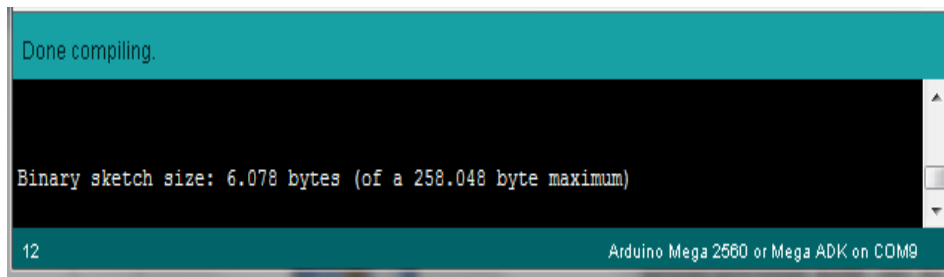
Gambar IV.2. Tampilan Program GPS

- Sebelum melanjutkan tahap instalasi mikrokontroler pada program yang telah selesai, maka terlebih dahulu program tersebut di-Save sebelum di-Compile. Untuk menyimpan Program dapat dilihat pada gambar IV.3.



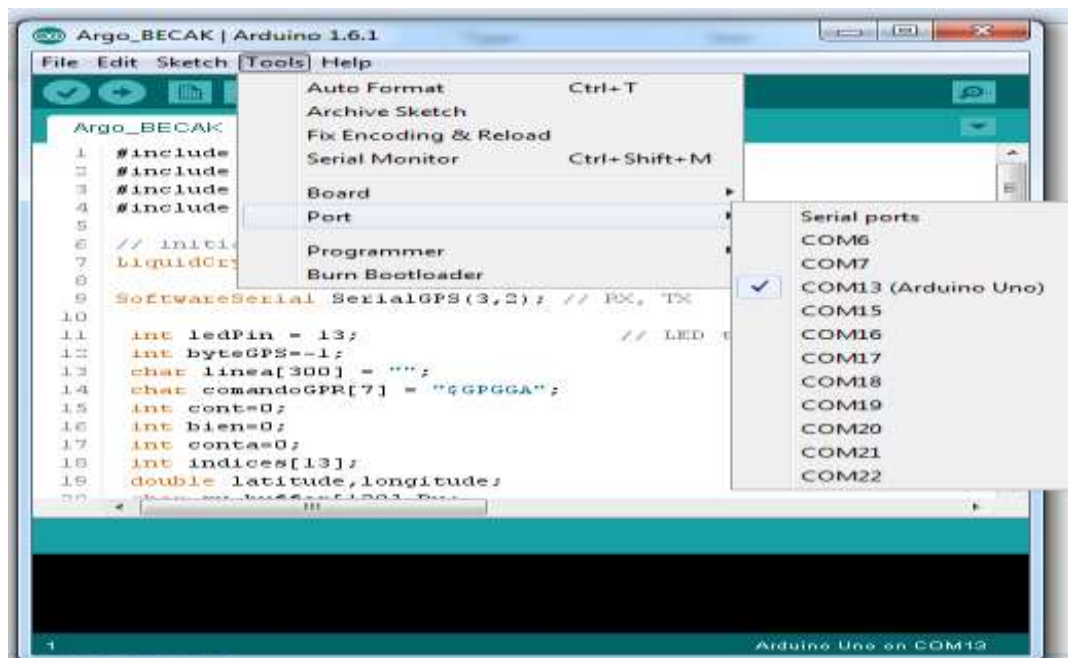
Gambar IV.3. Proses Penyimpanan File.

4. Untuk melanjutkan tahap instalasi mikrokontroler, program terlebih dahulu di-*Verify*, proses ini berfungsi untuk mensetting program kedalam arduino. Dapat dilihat apakah program yang dibuat memiliki kesalahan atau tidak, kalau berhasil maka akan tertulis “*No errors*”. Proses Compile dapat dilihat pada gambar IV.4.



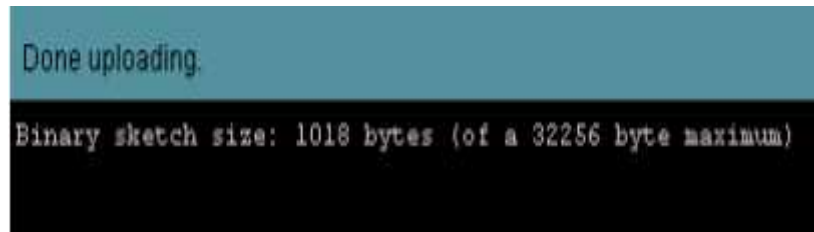
**Gambar IV.4. Proses *Compile***

5. Sebelum mengupload program terlebih dahulu mengecek dan memilih port arduino agar saat mengupload program berhasil. Pengecekan port arduino dapat dilihat pada gambar IV.5.



**Gambar IV.5. Memilih Port Arduino**

6. Untuk mengisi program yang telah di *compile* dari PC/Laptop ke dalam Arduino Atmega328, kemudian mengklik tombol “*Upload*”, proses ini berfungsi untuk memasukkan *sketch* program kedalam Arduino, jika berhasil akan tampil seperti gambar IV.6.



**Gambar IV.6. Proses Upload**

#### **IV.2. Hardware**

Setelah semua rangkaian yang telah selesai dirancang pada Perancangan Argo Becak Motor Berbasis Arduino Dan GPS Dengan Metode Euclidean, kemudian dilakukan penyatuan semua rangkaian yang telah selesai. Berikut adalah gambar hasil dari Perancangan Argo Becak Motor Berbasis Arduino Dan GPS Dengan Metode Euclidean, ditunjukkan oleh gambar IV.7.



**Gambar IV.7. Keseluruhan dari Hardware**

### IV.3. Uji Coba Perangkat

Pengujian perangkat dilakukan guna mendapatkan hasil yang maksimal pada GPS. Ada beberapa pengujian yang akan dilakukan antara lain:

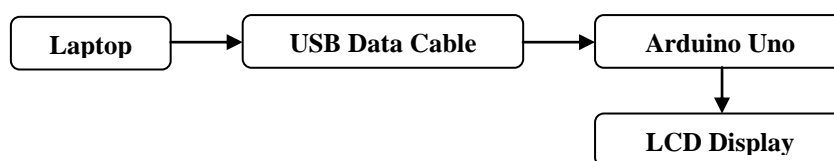
#### IV.3.1 Pengujian Rangkaian Arduino Uno dengan LCD

Untuk mengetahui apakah Arduino Uno ini dapat bekerja dengan baik maka harus menjalankan program Arduino IDE dengan menggunakan bahasa visual C++ pada rangkaian tersebut. Yang harus dilakukan sebelum proses running program adalah mengupload program pada mikrokontroler.

Peralatan :

1. Minimum Sistem Arduino Uno
2. Seperangkat USB Data Cable
3. Software Arduino IDE
4. DC Power Supply
5. Rangkaian LCD pada pin A0-A5

Rangkaian:



**GambarIV. 8. Diagram Blok Pengujian Rangkaian Arduino Uno dan LCD**

Persiapan:

1. Memasang rangkaian seperti yang ditunjukkan pada Gambar 4.8.
2. Mengetik program pengujian menggunakan Software Arduino IDE.
3. Mengupload program dan Menjalankan program.

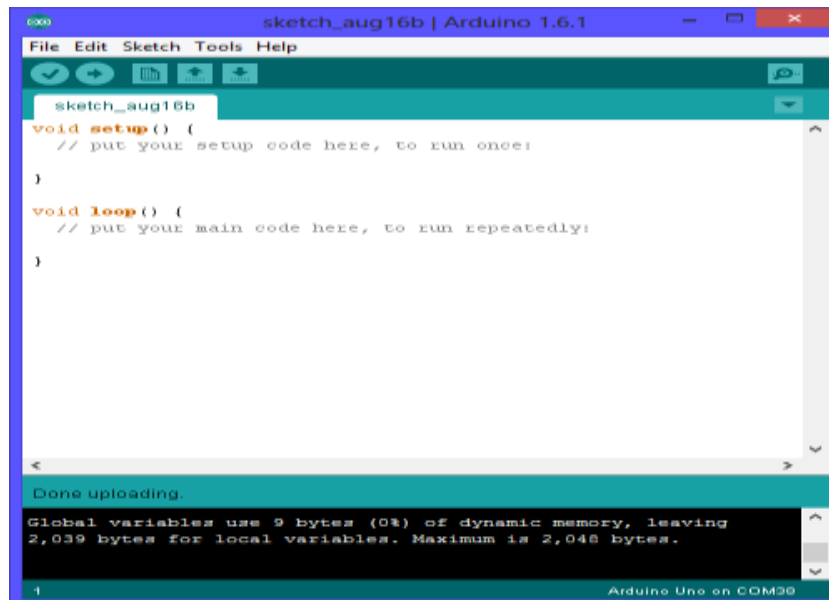
Langkah-langkah yang dilakukan:

1. Double klik aplikasi Arduino yang ada di layar laptop.



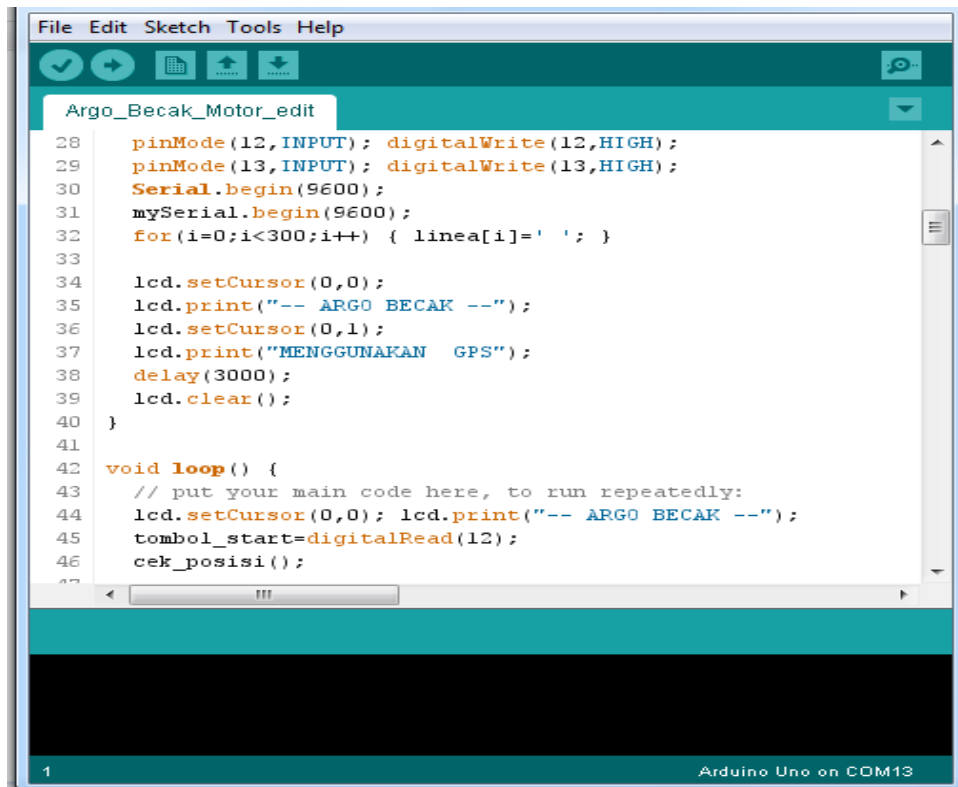
**GambarIV.9.Tampilan Software Arduino IDE**

2. Selanjutnya akan muncul tampilan awal “*sketch\_xxxxx*” secara otomatis. Pada halaman inilah dimulai menuliskan program sesuai yang diinginkan.



**GambarIV.10.Halaman Kerja Arduino IDE**

3. Ketikkan listing program sesuai pada tampilan gambar IV.11.Berikut :



```

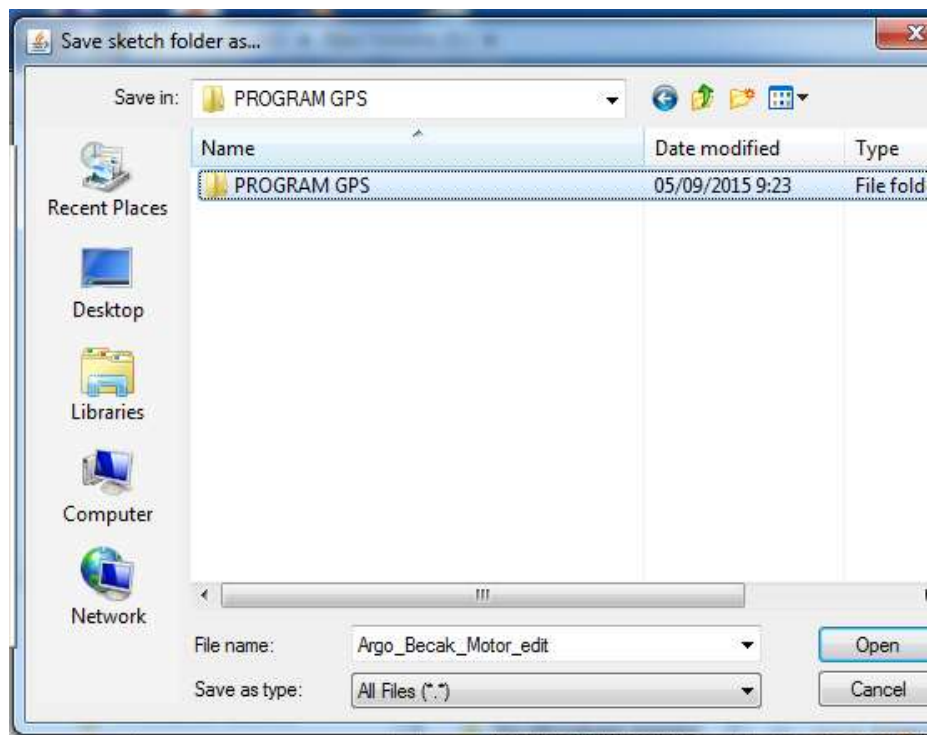
File Edit Sketch Tools Help
Argo_Becak_Motor_edit
28 pinMode(12,INPUT); digitalWrite(12,HIGH);
29 pinMode(13,INPUT); digitalWrite(13,HIGH);
30 Serial.begin(9600);
31 mySerial.begin(9600);
32 for(i=0;i<300;i++) { linea[i]=' '; }
33
34 lcd.setCursor(0,0);
35 lcd.print("-- ARGO BECAK --");
36 lcd.setCursor(0,1);
37 lcd.print("MENGGUNAKAN GPS");
38 delay(3000);
39 lcd.clear();
40 }
41
42 void loop() {
43 // put your main code here, to run repeatedly:
44 lcd.setCursor(0,0); lcd.print("-- ARGO BECAK --");
45 tombol_start=digitalRead(12);
46 cek_posisi();
47

```

1 Arduino Uno on COM13

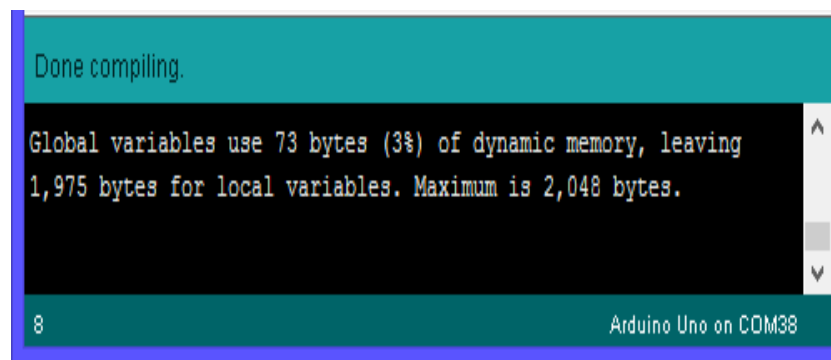
**Gambar IV.11. Listing Program Test LCD**

4. Klik Sketch → Verify. Kemudian akan muncul kotak dialog untuk menyimpan file project yang baru dibuat.



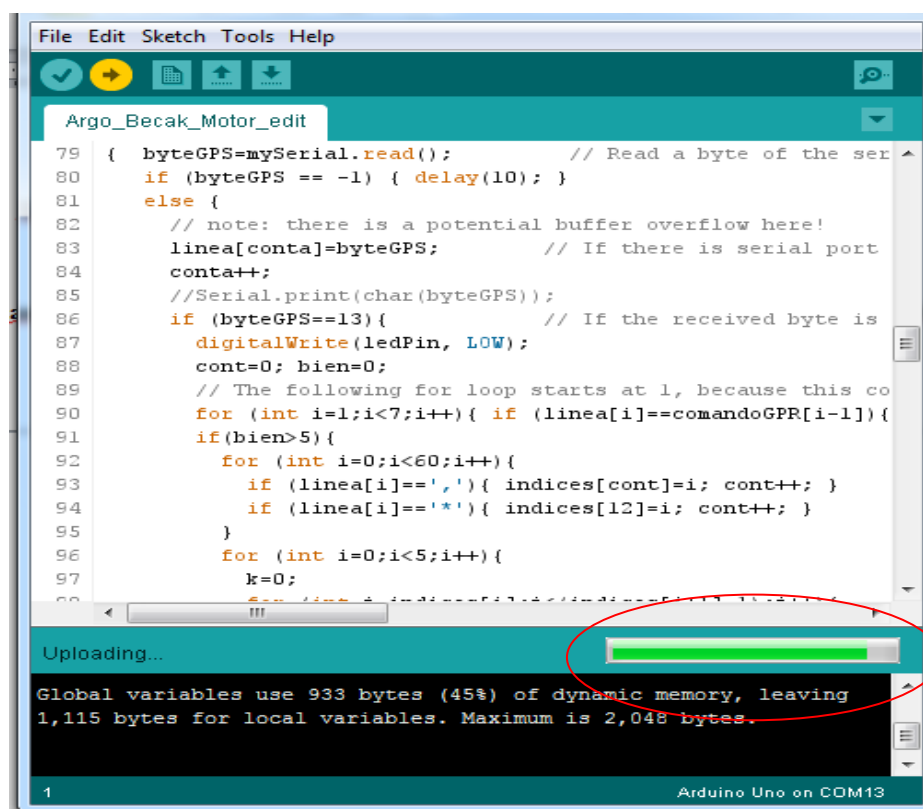
**GambarIV.12.Kotak Dialog menyimpan File Test LCD**

5. Ketika sudah selesai proses Verify dan penyimpanan, akan muncul pada bagian bawah program Arduino IDE seperti pada gambar IV.13.



**Gambar IV.13. Compiling Program Test LCD**

6. Kalau sudah tidak ada error, maka klik ikon → Upload atau Ctrl + U.



**Gambar IV.14. Proses Uploading Program dari Laptop ke Rangkaian Arduino**

Hasil dan analisa :

Pada uji coba rangkaian Arduino Uno terhubung dengan LCD, diperlukan pemanggilan library “`#include <LiquidCrystal.h>`” yang berfungsi untuk menambahkan

fungsi-fungsi program menampilkan karakter pada LCD. Kemudian “*Liquid Crystallcd* (A5,A4,A3,A2,A1,A0);” adalah listing program untuk pengaturan letak pin-pin kaki LCD dihubungkan ke pin-pin Arduino Uno. Penulisan pin-pin ini harus sesuai antara program dengan alat yang telah dipasang. Selanjutnya “*lcd\_begin (16,2);*” yaitu pengaturan jumlah baris dan kolom sesuai LCD yang digunakan. Karena yang digunakan yaitu LCD 16x2 karakter, maka penulisan pada program ini yaitu “*lcd\_begin(16,2);*”. Apa bila menggunakan LCD yang berukuran 16 x 2, maka pada program seharusnya tertulis “*lcd\_begin(16,2);*”.

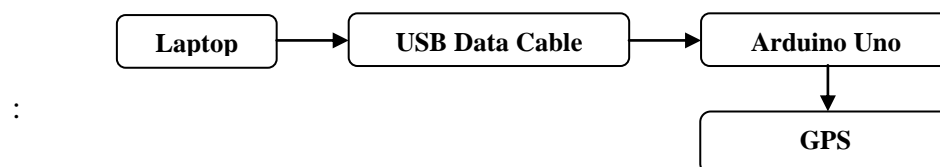
### IV.3.2. Pengujian GPS Dengan Arduino

Pengujian yang berikutnya yaitu Mengkoneksikan GPS dengan arduino. Pengujian ini dilakukan untuk mengetahui apakah GPS yang telah dirancang bekerja sesuai dengan harapan atau tidak.

Peralatan yang dibutuhkan:

1. Minimum Sistem Arduino Uno
2. GPS
3. Laptop
4. Seperangkat USB Data Cable
5. Software Arduino IDE
6. Power Supply

Rangkaian:



**GambarIV.15.**Diagram Blok RangkaianPengujianGPS

Persiapan:

1. Memasang rangkaian seperti yang ditunjukkan pada Gambar IV.15.
2. Mengkoneksikan GPS.
3. Mengetik text dan mengirimkan.

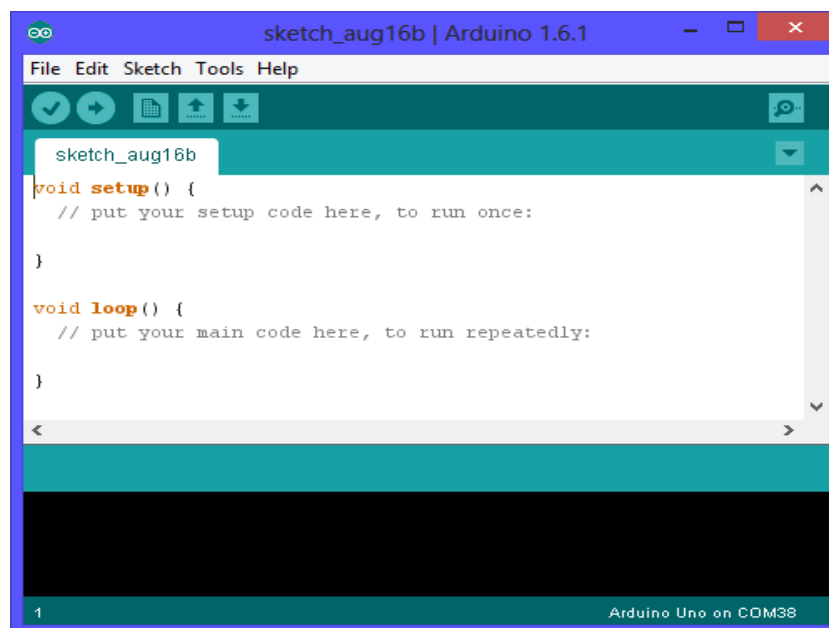
Langkah-langkah yang dilakukan:

1. Double klik aplikasi Arduino yang ada di layar laptop.



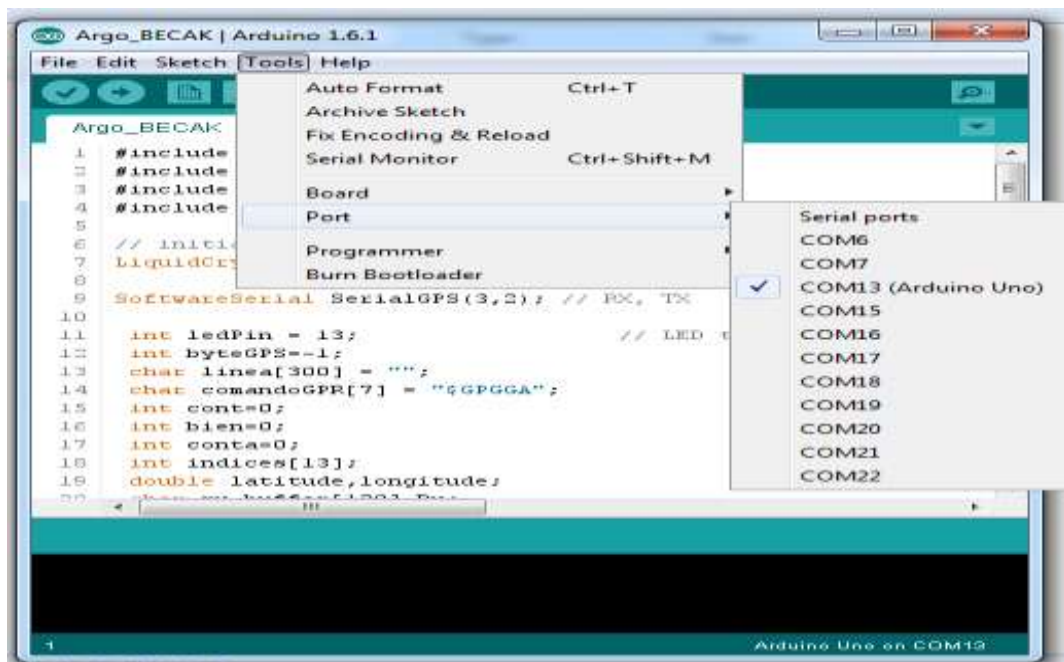
**Gambar IV.16. Tampilan Software Arduino IDE**

2. Selanjutnya akan muncul tampilan awal "*sketch\_xxxxx*" secara otomatis. Pada halaman ini lah dimulai menuliskan program.



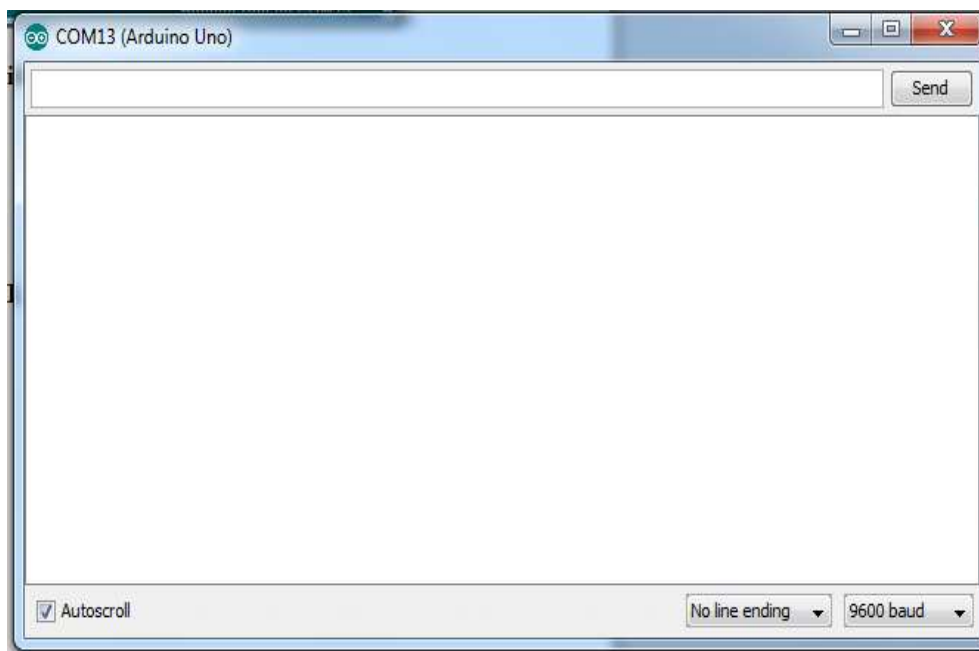
**Gambar IV.17. Halaman Kerja untuk Memulai Menuliskan Program**

3. Arahkan kursor di menu Tools, lalu pilih Port, lalu pilih Com13 (arduino Uno), lalu klik Serial Monitor.



**Gambar IV.18.**Menu pemilihan port dan serial monitor

4. Tampilan serial monitor.



**Gambar IV.19.**Serial Monitor

Hasil dan Analisa :

Pada uji coba rangkaian Arduino Uno terhubung dengan GPS, diperlukan pemanggilan library “*if (byteGPS == -1) { delay(10); }*” yang berfungsi untuk menambahkan fungsi-fungsi program menampilkan karakter pada GPS. Kemudian void cek\_posisi() { byteGPS=mySerial.read(); adalah listing program untuk pengaturan letak posisi GPS dan dihubungkan ke pin-in Arduino Uno. Penulisan pin-in ini harus sesuai antara program dengan alat yang telah dipasang. Selanjutnya “*lcd.setCursor(0,0); lcd.print(nilai\_latitude,4)*” yaitu pengaturan untuk menampilkan nilai latitude ke LCD yang telah di konversikan.

#### IV.4. Prinsip Kerja Akhir

Prinsip kerja Alat secara keseluruhan merupakan gabungan dari serangkaian pengujian yang telah dilakukan.

```

Argo_Becak_Motor_edit
1  #include <LiquidCrystal.h>
2  #include <SoftwareSerial.h>
3
4  SoftwareSerial mySerial(3, 2); // RX,TX
5  LiquidCrystal lcd(A5,A4,A3,A2,A1,A0);
6  char data;
7
8  int ledPin = 13;
9  int byteGPS= -1;
10 char linea[300]= "";
11 char comandoGPR[7] = "$GPRMC";
12 char array_latitude[10],array_longitude[11],array_jam[10];
13 int cont=0;
14 int bien=0;
15 int conta=0;
16 int indices[13];
17 double nilai[10];
18 int i,j,k,flag_cek_posisi;
19 double nilai_latitude,nilai_longitude,nilai_latitude_lama,ni
20 double jarak_tumbuh,panjang_tumbuh,panjang_tumbuh_lama;

```

```

20 double jarak_tempuh, jarak_total, biaya;
21 int tombol_start, tombol_stop, timer;
22 int keluar=0;
23 char buffer[33];
24 void setup() {
25     // put your setup code here, to run once:
26     pinMode(ledPin, OUTPUT); // Initialize LED pin
27     lcd.begin(16, 2);
28     pinMode(12, INPUT); digitalWrite(12, HIGH);
29     pinMode(13, INPUT); digitalWrite(13, HIGH);
30     Serial.begin(9600);
31     mySerial.begin(9600);
32     for(i=0; i<300; i++) { linea[i]=' '; }
33
34     lcd.setCursor(0, 0);
35     lcd.print("-- ARGO BECAK --");
36     lcd.setCursor(0, 1);
37     lcd.print("MENGUNAKAN GPS");
38     delay(3000);
39
39     lcd.clear();
40 }
41
42 void loop() {
43     // put your main code here, to run repeatedly:
44     lcd.setCursor(0, 0); lcd.print("-- ARGO BECAK --");
45     tombol_start=digitalRead(12);
46     cek_posisi();
47
48     if(tombol_start==0) { lcd.clear();
49     timer=30;
50     while(timer>0) {
51         lcd.setCursor(0, 0); lcd.print("WarmingUp GPS...");
52         lcd.setCursor(0, 1);
53         sprintf(buffer, "Waiting for %02d S", timer);
54         lcd.print(buffer);
55         delay(700);
56         timer--;
57     }
58

```

```

82 // note: there is a potential buffer overflow here!
83 linea[conta]=byteGPS; // If there is serial port
84 conta++;
85 //Serial.print(char(byteGPS));
86 if (byteGPS==13){ // If the received byte is
87     digitalWrite(ledPin, LOW);
88     cont=0; bien=0;
89     // The following for loop starts at 1, because this co
90     for (int i=1;i<7;i++){ if (linea[i]==comandoGPR[i-1]){
91     if(bien>5){
92         for (int i=0;i<60;i++){
93             if (linea[i]==' '){ indices[cont]=i; cont++; }
94             if (linea[i]=='*'){ indices[12]=i; cont++; }
95         }
96         for (int i=0;i<5;i++){
97             k=0;
98             for (int j=indices[i];j<(indices[i+1]-1);j++){
99                 if (i==0) { array_jam[k]=linea[j+1]; k++; }
100                else if(i==2) { array_latitude[k]=linea[j+1]; k+
101                else if(i==4) { array_longitude[k]=linea[j+1]; k+
102                else if(i==6) { array_biaya[k]=linea[j+1]; k++; }
103            }
104            konversi_latitude();
105            konversi_longitude();
106            //lcd.setCursor(0,0); lcd.print(array_jam);
107            lcd.setCursor(0,0); lcd.print(nilai_latitude,4);
108            lcd.setCursor(0,1); lcd.print(nilai_longitude,3);
109
110            hasil_latitude=(nilai_latitude - nilai_latitude_lama
111            hasil_longitude=(nilai_longitude - nilai_longitude_l
112            jarak=sqrt(pow(hasil_latitude,2) + pow(hasil_longitu
113            jarak_total=jarak_total+jarak;
114            jarak_tempuh=jarak_total*111.319;
115            if(jarak_tempuh>10900) { jarak_tempuh=0; jarak_total
116            biaya=jarak_tempuh * 5000; // Rp. 5000 adalah biaya
117            lcd.setCursor(8,0); lcd.print(jarak_tempuh,3); lcd.p
118            lcd.setCursor(8,1); lcd.print("Rp."); lcd.print(biay
119
120            nilai_latitude_lama=nilai_latitude;
121            nilai_longitude_lama=nilai_longitude;
122            nilai_latitude_lama=nilai_latitude;

```

**Gambar IV.20.**Inisialisasi Program Arduino, GPS dan LCD