



PROYEK AKHIR - VE 180626

**PERANCANGAN *PROTOTYPE* DAN *MONITORING*
BEBAN TRUK JASA EKSPEDISI MENGGUNAKAN *WEB***

BAMBANG WISNU WARDANA KUSUMA
NRP 10311600000028

Dosen Pembimbing
Slamet Budiprayitno
Lucky Putri Rahayu

Departemen Teknik Elektro Otomasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember
Surabaya
2019

-----Halaman ini sengaja dikosongkan-----



FINAL PROJECT - VE 180626

***PROTOTYPE DESIGN AND LOADING MONITORING OF
EXPEDITION SERVICES USING WEB***

*Bambang Wisnu Wardhana Kusuma
NRP 10311600000028*

*Supervisor
Slamet Budiprayitno
Lucky Putri Rahayu*

*Electrical and Automation Engineering Department
Vocational Faculty
Sepuluh Nopember Institute of Technology
Surabaya 2019*

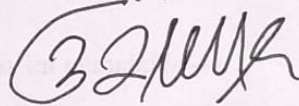
-----Halaman ini sengaja dikosongkan-----

PERNYATAAN KEASLIAN PROYEK AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Proyek Akhir saya dengan judul “*PERANCANGAN PROTOTYPE DAN MONITORING BEBAN TRUK JASA EKSPEDISI MENGGUNAKAN WEB*” adalah benar-benar hasil karya intelektual sendiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya orang lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai dengan peraturan yang berlaku.

Surabaya, Juli 2019



Bambang Wisnu Wardhana Kusuma
NRP 10311600000028

-----Halaman ini sengaja dikosongkan-----

**PERANCANGAN *PROTOTYPE* DAN *MONITORING* BEBAN
TRUK JASA EKSPEDISI MENGGUNAKAN *WEB***

PROYEK AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan
Memperoleh Gelar Ahli Madya Teknik
Pada
Departemen Teknik Elektro Otomasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember

Menyetujui:

Pembimbing 2



Lucky Putri Rahayu, S.Si., M.Si.

NPP. 1991201712058

Pembimbing 1



Slamet Budiprayitno, ST., MT.

NIP. 197811132010121003

**SURABAYA
JULI, 2019**

-----Halaman ini sengaja dikosongkan-----

PERANCANGAN PROTOTYPE DAN MONITORING BEBAN TRUK JASA EKSPEDISI MENGGUNAKAN WEB

Nama : Bambang Wisnu Wardhana Kusuma
NRP : 10311600000028
Pembimbing I : Slamet Budiprayetno. ST., MT.
NIP : 196608111992031004
Pembimbing II : Lucky Putri Rahayu, S.Si., M.Si.
NPP : 1991201712058

ABSTRAK

Pengiriman barang menggunakan truk pada jasa ekspedisi seringkali terjadi kendala yang mengambat proses pengiriman barang sampai tujuan. Seperti pencurian barang yang dilakukan oleh bajing loncat atau kecurangan yang dilakukan oleh sopir itu sendiri. Sehingga membuat pemilik jasa ekspedisi maupun pengguna jasa mengalami kerugian. Berdasarkan permasalahan yang terjadi di lapangan, ditemukan solusi dengan membuat alat *Monitoring* beban dan lokasi truk yang dapat diakses melalui *Web*.

Cara kerja alat *Monitoring* ini yaitu dengan mengaktifkan alat dan disambungkan dengan koneksi internet untuk pengiriman data ke *database* yang ditampilkan pada *Web*. Data yang diambil pada truk adalah beban yang ada di bak truk dan lokasi truk. Fungsi dari alat ini adalah untuk memantau beban truk dan lokasi truk, sehingga pemilik jasa dapat menganalisa waktu dan tempat berkurangnya beban truk melalui *Web*.

Alat *Monitoring* beban dan lokasi truk menggunakan *Web* yang terkoneksi Internet mampu menampilkan data yang dimonitoring dan diakses dimanapun. Sensor berat yang digunakan memiliki persentase *error*, pada pengujian beban 1.000 Gram memiliki persentase *error* pada kisaran 0,2% - 2,1%, pada pengujian beban 2.000 Gram memiliki persentase *error* pada kisaran 0,1% - 9,5%, pada pengujian beban 3.000 Gram memiliki persentase *error* pada kisaran 0,03% - 2,3%, dan pada pengujian beban 4.000 Gram memiliki persentase *error* pada kisaran 0,2% - 7,75%, rata-rata kesalahan pembacaan GPS U-Blox Neo 6 yang dibandingkan dengan pembacaan Garmin GPS di tiga tempat berbeda yaitu di Jalan Wisma Permai II rata-rata erornya sebesar 3,77 Meter, di Taman Eldas Teknik Elektro Otomasi rata-rata erornya sebesar 3,71 Meter, dan di Jalan Gang Makam Blok H rata-rata erornya sebesar 6,678 Meter.

Kata Kunci : Sensor berat *Loadcell*, Sensor Posisi GPS, *Web*.

-----Halaman ini sengaja dikosongkan-----

PROTOTYPE DESIGN AND LOADING MONITORING OF EXPEDITION SERVICES USING WEB

Nama : Bambang Wisnu Wardhana Kusuma
NRP : 10311600000028
Pembimbing I : Slamet Budiprayetno, ST., MT.
NIP : 196608111992031004
Pembimbing II : Lucky Putri Rahayu, S.Si., M.Si.
NPP : 1991201712058

ABSTRACT

Delivery of goods using a truck while waiting for arrival. Such as theft of goods carried out by jumping squirts or cheating carried out by the driver itself. Improve business owners. Based on the problems that occur in the field, a solution is found by creating a Monitoring tool for the cost and location of trucks that can be accessed through a website.

How this Monitoring tool works is by activating the tool and connecting with an internet connection to send data to the required database on the website. Data taken on trucks is the load that is in the tailgate and the location of the truck. The function of this tool is to move truck loads and truck locations, so that service owners can analyze the time and reduce truck space through the website.

Truck load and location monitoring tools use an internet-connected web that can display data that is monitored and can be accessed anywhere. The weight sensor used has a percentage of errors, in a load test of 1,000 Gram has a percentage of errors in the range of 0.2% - 2.1%, in the load test 2,000 Gram has a percentage of errors in the range of 0.1% - 9.5%, in testing 3,000 Gram load has a percentage of errors in the range of 0.03% - 2.3%, and in the calculation of the load of 4,000 Gram has a percentage of errors in the range 0.2% - 7.75%, the average error reading GPS U-Blox Neo 6 compared to Garmin GPS readings in three different places, namely Jalan Wisma Permai II, the average error was 3.77 meters, in the Eldas Park Electrical Engineering Automation the average error was 3.71 meters, and on Jalan Gang Makam Blok H average the error rate is 6,678 meters.

Keywords: *heavy sensors, GPS position sensors, Web.*

-----Halaman ini sengaja dikosongkan-----

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT yang selalu memberikan rahmat dan hidayah-Nya sehingga Tugas Akhir ini dapat terselesaikan dengan baik. Shalawat serta salam semoga selalu dilimpahkan kepada Rasulullah Muhammad SAW, keluarga, sahabat, dan umat muslim yang senantiasa meneladani beliau.

Proyek Akhir ini disusun untuk memenuhi sebagian persyaratan guna menyelesaikan pendidikan Diploma 3 pada Bidang Studi Komputer Kontrol, Departemen Teknik Elektro Otomasi, Fakultas Vokasi, Institut Teknologi Sepuluh Nopember Surabaya. Dengan terselesaikannya Tugas Akhir ini tidak akan terwujud tanpa bantuan, dukungan, doa, serta dorongan semangat dari semua pihak yang terkait. Oleh karena itu, penulis mengucapkan terimakasih kepada:

1. Kedua orang tua dan keluarga yang senantiasa mendoakan dan memberikan dukungan dengan tulus tiada henti.
2. Bapak Slamet Budiprayitno, ST., MT. selaku dosen pembimbing pertama.
3. Ibu Lucky Putri Rahayu, S.Si., M.Si. selaku dosen pembimbing kedua.
4. Thoriqul Fahri, teman yang menemani begadang dan membantu saya baik pikiran maupun tenaga.
5. Teman-teman FOXVIRE D3 Teknik Elektro Otomasi yang selalu memberikan doa, semangat serta dukungan.
6. Semua pihak yang telah membantu baik secara langsung maupun tidak langsung dalam proses penyelesaian Proyek Akhir ini.

Penulis menyadari dan memohon maaf atas segala kekurangan pada Proyek Akhir ini. Akhir kata, semoga Proyek Akhir ini dapat bermanfaat dalam pengembangan ilmu di kemudian hari.

Surabaya, 20 Juli 2019

Penulis

-----Halaman ini sengaja dikosongkan-----

DAFTAR ISI

HALAMAN JUDUL 1	i
HALAMAN JUDUL 2	iii
PERNYATAAN KEASLIAN.....	v
LEMBAR PENGESAHAN	vii
ABSTRAK	ix
ABSTRACT	xi
KATA PENGANTAR.....	xiii
DAFTAR ISI	xv
DAFTAR GAMBAR.....	xvii
DAFTAR TABEL	xix
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah.....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan dan Manfaat	2
1.5 Sistematika Penulisan Laporan	2
BAB II TEORI DASAR	5
2.1 Loadcell	5
2.1.1 Sambungan/Pengawatan.....	9
2.1.2 Data Kalibrasi.....	10
2.1.3 Output Loadcell.....	10
2.2 <i>Global Positioning System</i> (GPS)	12
2.3 Teori <i>Euclidean Distance</i>	13
2.3.1 pada 1 Dimensi.....	14
2.3.2 pada 2 Dimensi.....	14
2.3.3 Penggunaan Teorema Euclid	15
2.3 Arduino Mega 2560	16
2.4 Liquid Cristal Display (LCD) 16x2.....	17
2.5 NodeMCU	19
2.6 GPS U-Blox Neo 6M	20
2.7 Arduino IDE	21
2.8 Atom Editor.....	22
BAB III PERANCANGAN SISTEM DAN IMPLEMENTASI	25
3.1 Perancangan <i>Hardware</i>	26
3.1.1 Perancangan Bak Truk Timbang	26
3.1.2 Perancangan Rangkaian <i>Shield</i> Arduino Mega 2560	28
3.1.3 Perancangan Rangkaian Sensor <i>Loadcell</i>	28

3.1.4 Perancangan Rangkaian I2C dan LCD <i>Display</i>	29
3.1.5 Perancangan Rangkaian GPS Neo 6 U-blox	30
3.1.6 Perancangan Rangkaian Koneksi Arduino dengan NodeMCU	31
3.2 Perancangan <i>Software</i>	32
3.2.1 Perancangan Program Arduino	32
3.2.2 Pemrograman NodeMCU	37
3.2.3 Pembuatan Desain Web	41
BAB IV PENGUJIAN DAN ANALISIS DATA	45
4.1 Pengujian Pembacaan Sensor <i>Loadcell</i>	45
4.2 Pengujian Pembacaan Sensor GPS	48
4.2.1 Pengujian Pertama Sampai Ketiga Sensor GPS	50
4.2.2 Pengujian Keempat Sampai Keenam Sensor GPS	50
4.2.3 Pengujian Ketujuh Sampai Kesembilan Sensor GPS	50
4.3 Pengujian Pengiriman Data dari Arduino ke NodeMCU	51
4.4 Pengujian Pengiriman Data Oleh NodeMCU ke <i>Firebase</i>	53
4.5 Pengujian <i>Monitoring</i> di <i>Web</i>	53
BAB V KESIMPULAN DAN SARAN	57
5.1 Kesimpulan	57
5.2 Saran	58
DAFTAR PUSTAKA	59
LAMPIRAN A	A-1
LAMPIRAN B	B-1
LAMPIRAN C	C-1
BIODATA PENULIS	D-1

DAFTAR GAMBAR

Gambar 2. 1	Rangkaian Jembatan Wheatstone	6
Gambar 2. 2	Rangkaian Resistansi Strain Gauge Saat Tidak Ada Beban	7
Gambar 2. 3	Rangkaian Resistansi Strain Gauge Saat Diberi Beban	8
Gambar 2. 4	Cara Kerja GPS [6].....	13
Gambar 2. 5	Kordinat Titik (1,2) Dan Titik (5,5) [7]	14
Gambar 2. 6	Jarak Antara Gedung Sate Bandung Dan Masjid Raya Lembang Pada Google Maps [7]	15
Gambar 2. 7	Perhitungan Jarak Menggunakan Excel [7]	16
Gambar 2. 8	Arduino Mega 2560.....	17
Gambar 2. 9	<i>Liquid Cristal Display</i> (LCD) 16x2.....	19
Gambar 2. 10	<i>Board</i> ESP8266 NodeMCU.....	20
Gambar 2. 11	GPS U-Blox Neo 6	21
Gambar 2. 12	Arduino IDE	22
Gambar 2. 13	Tampilan Atom Editor.....	23
Gambar 3. 1	Blok DiaGram Sistem.....	25
Gambar 3. 2	Desain Bak Truk Tampak Samping.....	26
Gambar 3. 3	Desain Bak Truk Tampak Atas.....	27
Gambar 3. 4	Desain Bak Truk Tampak Belakang.....	27
Gambar 3. 5	Perancangan Rangkaian <i>Loadcell</i>	29
Gambar 3. 6	Rangkaian I2C LCD 16x2	30
Gambar 3. 7	Perancangan Rangkaian GPS Neo 6 U-blox.....	31
Gambar 3. 8	Deklarasi <i>Variable</i>	32
Gambar 3. 9	<i>Void Setup</i>	33
Gambar 3. 10	<i>Void Loop</i> Untuk Sensor GPS	34
Gambar 3. 11	<i>Void Loop</i> Untuk Sensor <i>Loadcell</i>	35
Gambar 3. 12	<i>Flowchart</i> Pengiriman data dari Arduino Mega 2560 ke NodeMCU	36
Gambar 3. 13	Deklarasi Awal	37
Gambar 3. 14	<i>Void Setup</i> Program NodeMCU	38
Gambar 3. 15	<i>Void Loop</i> Program NodeMCU	39
Gambar 3. 16	<i>Flowchart</i> Penerimaan Data oleh NodeMCU dari Arduino Mega 2560	40
Gambar 3. 17	<i>Flowchart</i> Pengiriman Data Pada <i>Database</i> Melalui NodeMCU	41
Gambar 3. 18	Pembuatan Desain <i>Web DasBoard</i>	42

Gambar 3. 19 Pembuatan Desain *Web History Page*42

Gambar 3. 20 *Flowchart* Menampilkan Data Pada Web Dari Firebase..
.....43

Gambar 4. 1 Data Pembacaan Pada *Serial Monitor* Arduino 52

Gambar 4. 2 Data Penerimaan Pada *Serial Monitor* NodeMCU 52

Gambar 4. 3 Hasil Pengiriman Data Dari Nodemcu Ke *Firebase* 53

Gambar 4. 4 Tampilan *Dashboard Truck 1* 54

Gambar 4. 5 Tampilan *Dashboard Truck 2* 55

Gambar 4. 6 Tampilan *Dashboard Truck 3* 56

DAFTAR TABEL

Tabel 2. 1 Spesifikasi <i>Loadcell</i> 50 Kg	5
Tabel 2. 2 Spesifikasi Arduino Mega 2560.....	17
Tabel 2. 3 Perbandingan Pin Arduino dengan NodeMCU	20
Tabel 3. 1 Perancangan Rangkaian <i>Shield</i> Arduino Mega 2560	28
Tabel 3. 2 Perancangan Rangkaian Sensor <i>Loadcell</i>	29
Tabel 3. 3 Perancangan I2C dan LCD Display	30
Tabel 3. 4 Konfigurasi Pin Arduino dengan <i>Logic Shifter</i>	31
Tabel 3. 5 Konfigurasi Pin NodeMCU dengan <i>Logic Shifter</i>	32
Tabel 4. 1 Hasil Pengujian Beban 1000 Gram	45
Tabel 4. 2 Hasil Pengujian Beban 2.000 Gram	46
Tabel 4. 3 Hasil Pengujian Beban 3.000 Gram	47
Tabel 4. 4 Hasil Pengujian Beban 4.000 Gram	48
Tabel 4. 5 Tabel Perbandingan Data Pembacaan GPS U-Blox Neo 6 Dengan Data Pembacaan Garmin GPS	49

-----Halaman ini sengaja dikosongkan-----

BAB I

PENDAHULUAN

1.1 Latar Belakang

Jasa pengiriman barang merupakan suatu bentuk pelayanan publik yang menawarkan kemudahan dalam proses mengirim suatu barang dari satu kota ke kota lainnya dengan aman dan dapat dipertanggung jawabkan oleh pihak jasa tersebut. Pengiriman barang dapat berupa dokumen, logistik, produk elektronik dan lain-lain. Alat transportasi yang digunakan untuk mengirim barang dapat melalui jalur darat, laut maupun udara. Jarak menjadi bagian penting untuk menentukan alat transportasi apa yang akan digunakan untuk mengangkut barang-barang kiriman tersebut.

Belakangan ini marak terjadi pencurian muatan pada truk yang mengakibatkan berkurangnya muatan tersebut sebelum sampai tujuan. Ketika memulai aksi nekat tersebut, yang bertugas membuka penutup atau terpal melakukan pekerjaannya, kemudian yang bertugas menjarah mulai mengambil sesuatu yang bisa dijual kemudian dibuang ke jalan. Terkadang supir juga menjadi korban akibat mengetahui aksi bajing loncat, namun terkadang juga menghiraukannya karena takut digebuki oleh preman yang menjadi otak kejahatan tersebut. Ini dapat juga digunakan untuk mengantisipasi karyawan ekspedisi yang curang, dengan kata lain barang tersebut sebenarnya hilang bukan karena dicuri bajing loncat melainkan tindakan dari karyawan yang sengaja mencuri barang yang akan dikirimkan ke pihak pelanggan karena merasa tidak puas bekerja di ekspedisi tersebut atau alasan lain [1].

Permasalahan tersebut mengakibatkan kerugian yang besar bagi pemilik jasa maupun konsumen yang menggunakan jasa ekspedisi. Dari permasalahan tersebut, maka kami merancang suatu sistem alat pendeteksi pencurian muatan di truk ekspedisi. Sistem mendeteksi pencurian ini dibuat untuk mendapatkan data berupa koordinat posisi keberadaan truk saat barang tersebut dicuri yang ditandai dengan berkurangnya berat muatan yang ditimbang setiap saat dengan sensor berat yg ditempatkan dibawah bak truk, kemudian mengirimkan data (data dari *GPS* dan *Loadcell*) tersebut ke *database* dan ditampilkan di *Web*.

1.2 Perumusan Masalah

1. pengiriman barang pada jasa ekspedisi seringkali terjadi pencurian oleh bajing loncat membutuhkan *Monitoring* yang dilakukan secara *realtime*.
2. Pemilik jasa dan konsumen seringkali dirugikan akibat pencurian yang dilakukan oleh bajing loncat.
3. Pemilik jasa kesulitan mengetahui lokasi hilangnya barang.

1.3 Batasan Masalah

Dalam pembuatan rancang bangun ini terdapat batasan masalah, yaitu : Menggunakan *firebase* sebagai *database* yang tidak bisa merecord data yang banyak.

1.4 Tujuan dan Manfaat

Tujuan dari Proyek Akhir ini adalah:

1. Merancang alat pendeteksi beban dan lokasi
2. Menyajikan informasi mengenai berat beban truk, koordinat GPS, dan status aktif alat menggunakan *Web* yang digunakan untuk *Monitoring*.

Manfaat pembuatan dan perancangan Proyek Akhir:

1. Jasa ekspedisi dapat me*Monitoring* dan menganalisa dimana tempat dan kapan waktu terjadinya pencurian barang. Sehingga dapat menghindari rute jalan yang rawan pencurian barang.

1.5 Sistematika Penulisan Laporan

Untuk pembahasan lebih lanjut, laporan Proyek Akhir ini disusun dengan sistematika sebagai berikut:

Bab 1 PENDAHULUAN

Membahas tentang latar belakang, perumusan masalah, tujuan dan manfaat, batasan masalah, sistematika laporan, metodologi penelitian, dan sistematika laporan

Bab II TEORI PENUNJANG

Bab ini akan menjelaskan tentang teori yang berisi teori-teori dasar yang akan dijadikan sebagai landasan dan mendukung dalam perencanaan dan pembuatan alat yang dibuat.

Bab III PERANCANGAN SISTEM

Membahas perencanaan dan pembuatan yang meliputi pembuatan *hardware* mekanik dan elektrik. Selain itu juga dilakukannya perancangan *software* yang meliputi program dari alat yang akan dibuat.

Bab IV PENGUJIAN DAN ANALISA DATA

Membahas pengujian alat dan pengambilan data dari pengujian yang didapatkan serta menganalisa hasil kerja dan performa alat.

Bab V PENUTUP

Berisi penutup yang menjelaskan tentang kesimpulan yang didapat dari Proyek Akhir dan saran terkait evaluasi Proyek Akhir dan pengembangan alat ini lebih lanjut.

-----Halaman ini sengaja dikosongkan-----

BAB II TEORI DASAR

Bab ini membahas mengenai teori dasar dari peralatan yang digunakan dalam Proyek Akhir yang berjudul Sistem *Monitoring Dan Data Logging* Pada Jasa Ekspedisi Pengiriman Barang Berbasis *Web*. Materi ini digunakan sebagai dasar materi untuk pembuatan alat yang dibuat masing-masing mahasiswa untuk pembuatan keseluruhan alat ini.

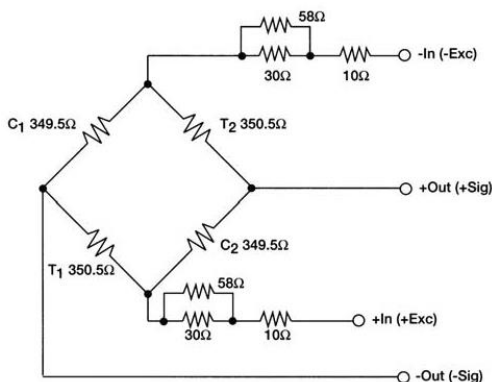
2.1 Loadcell

Loadcell merupakan sensor berat, apabila *Loadcell* diberi beban pada inti besinya maka nilai resitansi di *Strain Gauge* akan berubah. Umumnya *Loadcell* terdiri dari 4 buah kabel, dimana dua kabel sebagai eksitasi dan dua kabel lainnya sebagai sinyal keluaran. *Loadcell* adalah alat elektromekanik yang biasa disebut *Transducer*, yaitu gaya yang bekerja berdasarkan prinsip deformasi sebuah material akibat adanya tegangan mekanis yang bekerja, kemudian merubah gaya mekanik menjadi sinyal listrik. Untuk menentukan tegangan mekanis didasarkan pada hasil penemuan Robert Hooke, bahwa hubungan antara tegangan mekanis dan deformasi yang diakibatkan disebut regangan. Regangan ini terjadi pada lapisan kulit dari material sehingga menungknikan untuk diukur menggaunakan sensor regangan atau *Strain Gauge* [2].

Tabel 2. 1 Spesifikasi *Loadcell* 50 Kg

Spesifikasi	
<i>Capacity</i>	50 Kg
<i>Rated output (MV/V)</i>	2.0±0.15
<i>Zero balance(%RO)</i>	1.0
<i>Input resistance(Ω)</i>	402±6
<i>Output resistance(Ω)</i>	350±3
<i>Insulation resistance(MΩ<50V>)</i>	5000
<i>Recommended Excitation voltage(V)</i>	10~15
<i>Safe overload(%RO)</i>	150
<i>Load cell material</i>	Aluminium
<i>Plafform size</i>	350x350mm
<i>Method of connecting wire</i>	Red(+), Black(-), Green(+), White(-)

Jembatan Wheatstone yang tersusun seperti Gambar 2.1 merupakan diagram sederhana *Loadcell*. Resistor yang bertanda T1 dan T2 merupakan Strain Gauge yang menerima gaya tarik (*Tension*) saat *Loadcell* menerima beban. Sedangkan Resistor yang bertanda C1 dan C2 adalah Strain Gauge yang menerima gaya tekan (*Compression*) ketika *Loadcell* dibebani. Titik +In dan -In mengacu pada +Excitation(+Exc) dan -Excitation(-Exc). Melalui titik/terminal inilah tegangan sumber diberikan oleh Indikator timbangan digital. Pada umumnya, tegangan *Excitation* bernilai 10VDC dan 15VDC bergantung pada indikator dan *Loadcell* yang dipakai. Titik +Out dan -Out mengacu pada +Signal(+Sig) dan -Signal(-Sig). Sinyal yang diperoleh *Loadcell* dikirim ke Indikator melalui signal input untuk selanjutnya diproses sebagai nilai berat dan ditampilkan di layar digital indikator. Ketika *Loadcell* menerima beban, Strain Gauge C1 dan C2 mengalami gaya tekan. Kawatnya memendek dan diameternya membesar, sehingga nilai resistansi C1 dan C2 membesar. Sebaliknya, Strain Gauge T1 dan T2 mengalami gaya tarik, kawatnya memanjang dan diameternya mengecil sehingga nilai resistansi nya membesar. Perubahan nilai resistansi ini menyebabkan arus yang melewati C1 dan C2 lebih besar dibanding arus yang lewat pada t1 dan T2. Dan terjadilah beda potensial pada titik output atau signal *Loadcell*.

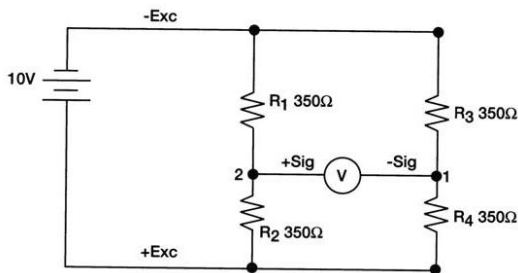


Gambar 2. 1 Rangkaian Jembatan Wheatstone [3]

Lihat arus yang mengalir pada *Loadcell*. Arus listrik di *supply* indicator melalui titik -In dan mengalir melalui C1, -Out dan kembali lagi ke Indikator. Dari indikator, arus mengalir melalui +Out, melewati

C2 dan kembali ke Indikator dititik +In. Untuk mengetahui total arus yang mengalir, kita perlu mengukur arus internal pada rangkaian pembaca signal di Indikator. Tetapi karena Impedansi internal indicator sangatlah tinggi, arus yang mengalir menjadi sangat kecil dan kita bisa mengabaikannya. Terdapat beda potensial antara -In dan +In, sehingga ada juga arus yang mengalir melewati -In, melalui T2 dan C2 kembali ke +In. Arus yang mengalir pada rangkaian sebagian besar berada pada sisi parallel ini. Resistor yang terpasang seri berfungsi sebagai kompensasi *Loadcell* terhadap temperatur, Zero dan linearitas.

Gantilah Ampermeter dengan Voltmeter sebagai pengganti display Indikator, sambungkan pada titik +Sig dan -Sig seperti pada Gambar 2.2, yang melambangkan signal positif dan negatif. Baterai bertegangan 10V melambangkan supply tegangan dari indicator yang akan membuat *Loadcell* bekerja. Resistor yang ada melambangkan Strain Gauge sebagai pengganti *Loadcell*. Resistansi semua Strain Gauge tetap sama selama tidak ada beban yang diterima *Loadcell*. Tegangan drop pada titik 1 dan 2 bisa kita hitung menggunakan Hukum Ohm. Setiap cabang mempunyai resistansi $350\Omega + 350\Omega = 700\Omega$. Arus yang mengalir tiap cabang adalah tegangan dibagi resistansi setiap cabang.



Gambar 2. 2 Rangkaian Resistansi Strain Gauge Saat Tidak Ada Beban
[3]

Resistansi semua Strain Gauge tetap sama selama tidak ada beban yang diterima *Loadcell*. Tegangan drop pada titik 1 dan 2 bisa kita hitung menggunakan Hukum Ohm. Setiap cabang mempunyai resistansi $350\Omega + 350\Omega = 700\Omega$. Arus yang mengalir tiap cabang adalah tegangan dibagi resistansi setiap cabang.

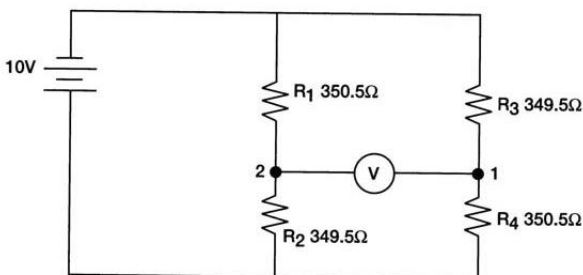
$$\begin{aligned}
 IR + R2 &= \frac{ER1 + R2}{\frac{R1 + R2}{10V}} & \dots\dots\dots (1) \\
 &= \frac{700\Omega}{10V} \\
 &= 14.3mA
 \end{aligned}$$

$$\begin{aligned}
 IR3 + R4 &= \frac{ER3 + R4}{\frac{R3 + R4}{10V}} & \dots\dots\dots (2) \\
 &= \frac{700\Omega}{10V} \\
 &= 14.3mA
 \end{aligned}$$

Untuk menghitung tegangan pada titik 1, gunakan Hukum Ohm

$$\begin{aligned}
 ER3 &= IR3 \times R3 & \dots\dots\dots (3) \\
 &= 14.3mA \times 350\Omega \\
 &= 5V
 \end{aligned}$$

Tegangan pada titik 2 juga 5 Volt karena semua resistornya sama. Tidak ada beda potensial antara titik 1 dan 2, dan inilah kondisi dimana Indikator kita menunjukkan Nola tau Zero. Lalu, berikan beban pada *Loadcell* sehingga R1 dan R4 mengalami gaya tarik dan resistannya membesar, sedangkan R2 dan R3 mengalami gaya tekan sehingga resistannya mengecil, seperti terlihat pada Gambar 2.3.



Gambar 2. 3 Rangkaian Resistansi Strain Gauge Saat Diberi Beban [3]

Catatan: Resistansi total setiap cabang tetap 700Ω sehingga arus yang mengalir disetiap cabang tetap 14.3mA

Dalam kondisi demikian, terjadi beda potensial antara titik 1 dan 2 dan tertampil pada voltmeter/*indicator*. Mari kita hitung besarnya beda potensial tersebut. Untuk mengukur tegangan di titik1, ukurlah terlebih dahulu tegangan *drop* pada R3. Sebagaimana kita ketahui, arus yang melewati R3 adalah 14.3mA.

$$\begin{aligned} ER3 &= IR3 (R3) && \dots\dots\dots (4) \\ &= 0.0143A (349.5\Omega) \\ &= 4.9979V \end{aligned}$$

Dan untuk mengetahui tegangan dititik 2, hitunglah dahulu tegangan drop pada R1. Ingat, arus yang melewati R1 adalah 14.3mA.

$$\begin{aligned} ER1 &= IR1 (R1) && \dots\dots\dots (5) \\ &= 0.0143A (350.5\Omega) \\ &= 5.0122V \end{aligned}$$

Beda potensial pada titik 1 dan 2 adalah selisih ER3 dan ER1 yaitu 0.143V atau 14.3mV. Disini terlihat rangkaian menjadi tidak seimbang dan terjadi beda potensial pada rangkaian sebesar 14.3mV. Indikator dikalibrasi sedemikian rupa sehingga sedikit perubahan pada milivolt akan diterjemahkan perubahan pembacaan pada pengukuran berat. Seperti yang pernah kita bahas, semestinya Indikator akan memakan arus, tetapi karena tingginya resistan internal Indikator, kita bisa mengabaikannya dan hal ini tidak mempengaruhi kinerja *Loadcell*.

2.1.1 Sambungan/Pengawatan

Pada umumnya, kabel pada *Loadcell* berjumlah empat atau enam kabel. Untuk enam kabel *Loadcell*, disamping mempunyai – dan + Signal maupun – dan + *Excitation* juga memiliki jalur - dan + sense. Jalur sense ini tersambung pada jalur sense Indikator yang berfungsi meMonitor tegangan *actual* pada *Loadcell*, dan mengirim balik ke Indikator untuk dianalisa apakah perlu menambah atau menguatkan signal yang dikirim balik sebagai kompensasi daya pada *Loadcell*. Untuk membantu agar pemasangannya tepat, kabel *Loadcell* memiliki kode warna tertentu. *Datasheet* kalibrasi setiap *Loadcell* akan menyertakan juga kode warna untuk penyambungan *Loadcell*.

2.1.2 Data Kalibrasi

Setiap *Loadcell* dilengkapi dengan data kalibrasi atau sertifikat kalibrasi sebagai informasi tentang *Loadcell* yang bersangkutan. Setiap *Datasheet* harus cocok dengan nomor seri, nomor model dan kapasitas. Informasi yang lain berupa karakteristik dalam mV/V, tegangan Excitasi, *non-linearity*, *hysteresis*, *zero balance*, *input resistance*, *output resistance*, efek temperature pada output dan *zero balance*, *insulation resistance* dan *cable length*. Kode warna untuk penyambungan juga disertakan.

2.1.3 Output Loadcell

Hasil pengukuran *Loadcell* selain ditentukan oleh besarnya beban, juga ditentukan oleh besarnya tegangan Eksitasi, dan karakteristik (mV/V) *Loadcell* itu sendiri. Salah satu karakteristik *Loadcell* yaitu 3 mV/V. Yang berarti setiap satu volt tegangan Excitasi, pada saat *Loadcell* dibebani maksimal akan mengeluarkan signal sebesar 3 mV. Jika beban 100 Kg diberikan pada *Loadcell* kapasitas 100 Kg dengan tegangan Excitasi 10 V, maka signal yang terkirim dari *Loadcell* tersebut adalah sebesar 30 mV. Demikian juga apabila dibebani 50 Kg dengan tegangan Excitasi tetap 10 V, karena 50 Kg adalah setengah dari 100 Kg maka keluaran *Loadcell* menjadi 15 mV. [3]

Loadcell memiliki bermacam-macam karakteristik yang bisa diukur, tergantung pada jenis logam yang dipakai, bentuk *Loadcell*, dan ketahanan dari lingkungan sekitar. Untuk memilih *Loadcell* yang sesuai dengan kebutuhan anda, penting untuk mengetahui definisi dari parameter berikut :

Calibration: membandingkan *output (signal)* *Loadcell* dengan standard

Combined Error: Penyimpangan maksimum, jika ditarik garis lurus diukur pada saat tanpa beban sampai ketika diberikan beban maksimal dan sebaliknya saat beban maksimal sampai pada keadaan tanpa beban. Pengukuran dinyatakan dalam persen terhadap kapasitas maksimal. Biasanya disebut juga *Non-linearity* dan *hysteresis*.

Creep: Perubahan sinyal keluaran *Loadcell* selama pembebanan tidak berubah, dan tidak ada perubahan lingkungan sekitar.

Creep Recovery: Perubahan pengukuran kondisi tanpa beban, setelah beberapa waktu diberikan beban dan kemudian beban dihilangkan.

Drift: Perubahan nilai pengukuran saat diberikan beban konstan.

Eccentric Load: Pembebanan pada area timbangan tapi tidak tepat di titik antar *Loadcell*.

Error: Perbedaan pengukuran dengan beban yang sesungguhnya.

Excitation: Tegangan input yang diberikan agar *Loadcell* bekerja. Pada umumnya *Loadcell* membutuhkan tegangan *Excitation* 10 VDC, tetapi ada juga yang memerlukan 15 VDC, 20 VDC, dan 25 VDC dan ada yang bisa bekerja pada arus AC dan DC.

Hysteresis: Penyimpangan maksimum hasil pengukuran dengan beban yang sama. Satu pengukuran dari nol sampai maksimum, pengukuran yang lain dari maksimum sampai nol. Pengukuran histerisis dinyatakan dalam persen terhadap kapasitas maksimum (% FS). Biasanya Histerisis selalu bernilai 0.02% FS, 0.03% FS dan 0.05% FS

Input Bridge Resistance: Resistansi *Input* daripada *Loadcell*. Diukur dengan Ohm Meter antara dua titik *input* atau *Excitasi*. Biasanya selalu lebih besar dari resistansi *Output*/sinyal karena adanya *resistor* kompensasi pada jalur *Excitasi*.

Insulation Resistance: Pengukuran resistansi antara sirkuit *Loadcell* dengan strukturnya. Pengukuran dilakukan dengan tegangan DC.

Non-Linearity: Penyimpangan maksimum pada grafik hasil kalibrasi terhadap garis lurus (Ideal) antara tanpa beban dan beban penuh. Dinyatakan dengan persentase terhadap pengukuran pada kapasitas maksimum, hanya diukur dari nol sampai maksimum. Umumnya *Non-linearity* sebesar 0.025 FS dan 0.035 FS.

Output: Sinyal yang dihasilkan oleh *Loadcell* dimana *Output* berbanding lurus dengan eksitasi dan beban yang diterapkan. Sinyal *Output Loadcell* dalam satuan mv/V atau V/A.

Output Bridge Resistance: Hambatan keluaran dari *Cell*. *Output Bridge Resistance* adalah 3500, 4800, 7000, 7500 dan 10000.

Output Rate: Tingkat perbandingan antara *Output* tanpa beban dengan *Output* saat ada beban.

Reapitibility: Selisih bacaan *Output* maksimum *Loadcell* untuk beban yang sama dan waktu yang identik.

Resolution: Perubahan terkecil di *input* mekanis yang menghasilkan perubahan yang terdeteksi dalam sinyal *Output*.

Safe Overload Rating: Beban maksimum (%) kapasitas beban, yang dapat diterapkan tanpa menghasilkan pergeseran permanen dalam karakteristik kinerja luar yang spesifik. [4]

2.2 *Global Positioning System (GPS)*

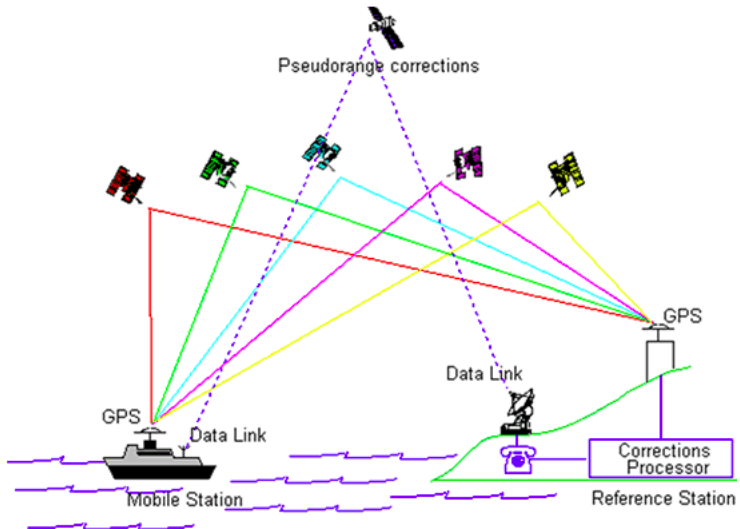
Global Positioning System (GPS) adalah sistem untuk menentukan letak di permukaan bumi dengan bantuan penyelarasan (*synchronization*) sinyal satelit. Sistem ini menggunakan 24 satelit yang mengirimkan sinyal gelombang mikro ke Bumi. Sinyal ini diterima oleh alat penerima di permukaan, dan digunakan untuk menentukan letak, kecepatan, arah, dan waktu. Sistem yang serupa dengan GPS antara lain GLONASS Rusia, Galileo Uni Eropa, IRNSS India.

Sistem ini dikembangkan oleh Departemen Pertahanan Amerika Serikat, dengan nama lengkapnya adalah NAVSTAR GPS (kesalahan umum adalah bahwa NAVSTAR adalah sebuah singkatan, ini adalah salah, NAVSTAR adalah nama yang diberikan oleh John Walsh, seorang penentu kebijakan penting dalam program GPS). Kumpulan satelit ini diurus oleh 50th *Space Wing* Angkatan Udara Amerika Serikat.

Sistem ini menggunakan sejumlah satelit yang berada di orbit bumi, yang memancarkan sinyalnya ke bumi dan ditangkap oleh sebuah alat penerima sinyal tersebut atau *GPS Tracker*. Selain satelit terdapat 2 sistem lain yang saling berhubungan, sehingga jadilah 3 bagian penting dalam sistem GPS. Ketiga bagian tersebut terdiri dari: *GPS Control Segment* (Bagian Kontrol), *GPS Space Segment* (bagian angkasa), dan *GPS User Segment* (bagian pengguna).

Karena GPS bekerja mengandalkan satelit, maka penggunaannya disarankan ditempat yang terbuka. Penggunaan di dalam ruangan, atau

ditempat yang menghalangi arah satelit (diangkasa), maka GPS tidak akan bekerja secara optimal dan akurat. Setiap daerah di atas permukaan bumi ini minimal terjangkau oleh 3-4 satelit. Pada dasarnya, setiap GPS terbaru bisa menerima sampai 12 *channel* satelit sekaligus. Kondisi langit yang cerah dan bebas dari halangan membuat satelit dapat dengan mudah diterima oleh GPS, maka akurasi yang diberikan juga akan semakin tinggi. Dengan mentransmisikan sinyal dari satelit ke perangkat GPS (*portable* murni atau *smartphone* yang sudah memiliki GPS). GPS akan membutuhkan transmisi dari 3 satelit untuk mendapatkan informasi dua dimensi (lintang dan bujur), dan 4 satelit untuk 3 dimensi (lintang, bujur, dan ketinggian). [5]



Gambar 2. 4 Cara Kerja GPS [9]

2.3 Teori *Euclidean Distance*

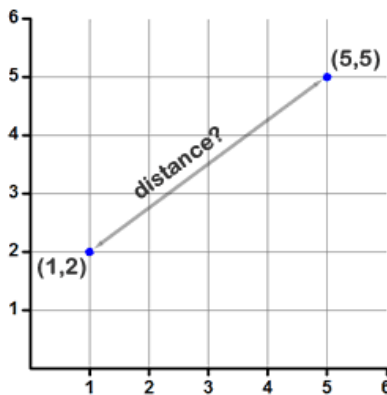
Euclidean distance adalah perhitungan jarak dari 2 buah titik dalam *Euclidean space*. *Euclidean space* diperkenalkan oleh Euclid, seorang matematikawan dari Yunani sekitar tahun 300 B.C.E. untuk mempelajari hubungan antara sudut dan jarak. *Euclidean* ini berkaitan dengan Teorema *Phytagoras* dan biasanya diterapkan pada 1, 2 dan 3

dimensi. Tapi juga sederhana jika diterapkan pada dimensi yang lebih tinggi.

2.3.1 Pada 1 Dimensi

Menghitung jarak *Euclidean* 1 dimensi. Titik pertama adalah 4, titik kedua adalah -10. Caranya adalah kurangkan -10 dengan 4. sehingga menghasilkan -14. Cari nilai absolut dari nilai -14 dengan cara memangkatkannya sehingga mendapat nilai 196. Kemudian diakarkan sehingga mendapatkan nilai 14. Sehingga jarak *euclidean* dari 2 titik tersebut adalah 14.

2.3.2 Pada 2 Dimensi



Gambar 2. 5 Kordinat Titik (1,2) Dan Titik (5,5) [6]

Caranya hampir sama. Pada Gambar 2.4 titik pertama mempunyai kordinat (1,2). Titik kedua ada di kordinat (5,5). Caranya adalah kurangkan setiap kordinat titik kedua dengan titik yang pertama. Yaitu, $(5-1, 5-2)$ sehingga menjadi $(4,3)$. Kemudian pangkatkan masing-masing sehingga memperoleh $(16,9)$. Kemudian tambahkan semuanya sehingga memperoleh nilai $16+9 = 25$. Hasil ini kemudian diakarkan menjadi 5. Sehingga jarak *euclidean*nya adalah 5. Berikut adalah Rumus Euclid :

$$d = \sqrt{(x1 - x2)^2 + (y1 - y2)^2} \dots\dots\dots (6)$$

Sehingga dari Formula Rumus Euclid dapat diimplementasi menjadi :

$$jarak = \sqrt{(lat1 - lat2)^2 + (long1 - long2)^2} \dots\dots\dots (7)$$

Hasil perhitungan Jarak Gambar 2.4 masih dalam satuan *decimal degree* (sesuai dengan format *longitude latitude* yang dipakai) sehingga untuk menyesuaikannya perlu dikalikan dengan 111.319 km (1 derajat bumi = 111.319 km) [6]

2.3.3 Penggunaan Teorema Euclid

Pada uji kali ini menggunakan titik kordinat Gedung Sate Bandung dan titik kordinat Masjid Raya Lembang sebagai berikut :

1. Titik Pertama Gedung Sate Bandung
Latitude : -6.901361
Longitude : 107.618633
2. Titik Kedua Masjid Raya Lembang
Latitude : -6.811771
Longitude : 107.618279

Jarak aktual yang diperoleh dari Google Maps adalah 10.05 Km.



Gambar 2. 6 Jarak Antara Gedung Sate Bandung Dan Masjid Raya Lembang Pada Google Maps

Pada Gambar 2.5 adalah tampilan jarak antara Gedung Sate Bandung dan Masjid Raya Lembang menggunakan Google Maps berikut rumus Excelnya supaya lebih mudah membuktikannya.

$$jarak = ((SQRT((B6 - B7)^2 + (C6 - C7)^2 * 111.319)) \dots (8)$$

Perhitungan Jarak melalui koordinat			
	Latitude	Longitude	
Gedung Sate	-6.901361	107.618633	
Mesjid Lembang	-6.811771	107.618279	
Euclide =	9.97	Km	=((SQRT((C5-C6)^2+(D5-D6)^2)*111.319))

Gambar 2. 7 Perhitungan Jarak Menggunakan Excel

Hasil perhitungan menggunakan Excel diperoleh Jarak = 9.97 Km

2.3 Arduino Mega 2560

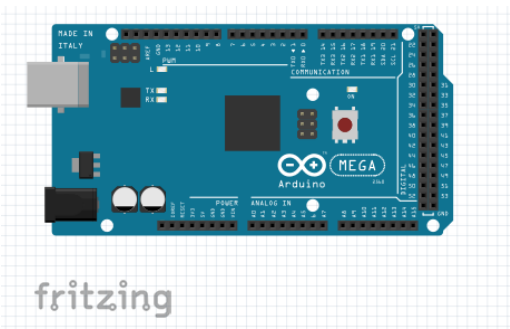
Arduino merupakan perangkat elektronik atau papan rangkaian elektronik *open – source* yang di dalamnya terdapat komponen utama, yaitu sebuah *chip* mikrokontroler dengan jenis AVR dari perusahaan Atmel. Mikrokontroler itu sendiri adalah *chip* atau IC (*Integrated circuit*) yang bisa diprogram menggunakan komputer. Tujuan menanamkan program pada mikrokontroler adalah agar rangkaian elektronik dapat membaca *input*, memproses *input* tersebut dan kemudian menghasilkan *output* sesuai yang diinginkan. Jadi mikrokontroler bertugas sebagai otak yang mengendalikan *input*, proses, dan *output* sebuah rangkaian elektronik [7].

Tipe Arduino yang akan digunakan pada penelitian kali ini yaitu Arduino Mega 2560. Arduino Mega 2560 adalah mikrokontroler berbasis atmega2560 dengan *Clock Speed* 16Mhz dan *Flash Memory* 256KB. Tegangan operasi untuk arduino jenis ini yaitu 5 V. Sedangkan tegangan *input* yang direkomendasikan yakni 7 – 12 V. Arduino ini memiliki 54 pin digital *input/output* pada pin 22-53 dengan 15 pin diantaranya merupakan pin PWM pada pin 0-13, 16 pin analog *input* pada pin A0 – A15, sambungan USB, sambungan catu daya tambahan dan tombol pengaturan ulang. Pada Tabel 2.1 merupakan spesifikasi dari Arduino

Mega 2560

Tabel 2. 2 Spesifikasi Arduino Mega 2560

Spesifikasi	Keterangan
<i>Chip</i> Mikrokontroler	Atmega2560
Tegangan Operasi	5 V
Tegangan <i>Input</i> (Rekomendasi)	7 V-12 V
Tegangan <i>Input</i> (Limit)	6 V- 20 V
<i>Pin</i> Digital I/O	54, (15 buah diantaranya dapat digunakan sebagai <i>Output</i> PWM)
<i>Pin</i> Analog <i>Input</i>	16 (A0 – A.15)
Arus DC per <i>Pin</i> I/O	40 ma
Arus DC <i>Pin</i> 3,3V	50 ma
Memori <i>Flash</i>	256 KB, 8 KB telah digunakan untuk <i>Bootloader</i>
SRAM	8 Kb
EEPROM	4 Kb
<i>Clock Speed</i>	16 mhz



Gambar 2. 8 Arduino Mega 2560

2.4 Liquid Cristal Display (LCD) 16x2

LCD (*Liquid Cristal Display*) adalah salah satu komponen elektronika yang berfungsi untuk menampilkan suatu *output* berupa nilai atau teks pada mikrokontroler. Sumber cahaya pada LCD adalah lampu neon berwarna putih di bagian belakang susunan cairan kristal. Titik cahaya yang jumlahnya puluhan ribu hingga jutaan ini yang akan

membentuk suatu tampilan. Kutub kristal cair yang dilewati arus listrik akan berubah karena pengaruh polarisasi medan magnetik yang timbul, oleh karena itu hanya beberapa warna yang akan diteruskan sedangkan warna lainnya akan disaring. LCD yang dipakai merupakan tipe berkarakter 20x4 baris, yang dapat menampilkan 20 karakter dengan 4 baris [7].

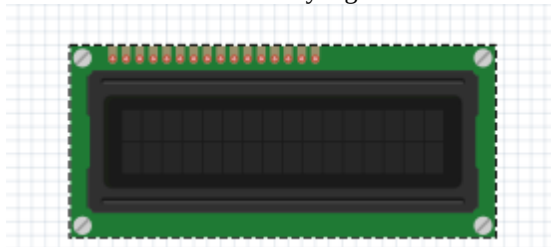
Selain menggunakan LCD 20x4, pada Proyek Akhir ini juga menggunakan modul I2C LCD. Yang dimaksud dengan modul I2C adalah modul LCD yang dikendalikan secara *Serial* sinkron dengan protokol I2C (Inter Integrated Circuit) atau TWI (*Two Wire Interface*). I2C hanya membutuhkan 4 port yaitu port SCL (*Serial Clock*), SDA (*Serial Data*) yang membawa informasi data antara I2C dengan mengontrolnya, VCC dan *ground*.

Pada aplikasinya tidak semua pin pada LCD 20x4 terpakai, melainkan hanya pin-pin yang dihubungkan dengan mikrokontroler yang digunakan saja. Pada aplikasi umumnya RW diberi logika rendah “0”. Bus data terdiri dari 4-bit atau 8-bit. Jika jalur data 4-bit maka yang digunakan ialah DB4 sampai dengan DB7. *Interface* LCD merupakan sebuah paralel bus, dimana hal ini sangat memudahkan dan sangat cepat dalam pembacaan dan penulisan data dari atau ke LCD. Kode ASCII yang ditampilkan sepanjang 8-bit dikirim ke LCD secara 4-bit atau 8 bit pada satu waktu. Jika mode 4-bit yang digunakan, maka 2 nibble data dikirim untuk membuat sepenuhnya 8-bit (pertama dikirim 4-bit MSB lalu 4-bit LSB dengan pulsa *clock* EN setiap *nibble*-nya). Jalur kontrol EN digunakan untuk memberitahu LCD bahwa mikrokontroler mengirimkan data ke LCD. Untuk mengirim data ke LCD program harus men-set EN ke kondisi *high* “1” dan kemudian mengatur dua jalur kontrol lainnya (RS dan R/W) atau juga mengirimkan data ke jalur data *bus*.

Saat jalur lainnya sudah siap, EN harus diatur ke “0” dan tunggu beberapa saat (tergantung pada *Datasheet* LCD), dan set EN kembali ke *high* “1”. Ketika jalur RS berada dalam kondisi *low* “0”, data yang dikirimkan ke LCD dianggap sebagai sebuah perintah atau instruksi khusus (seperti bersihkan layar, posisi kursor dan lain-lain). Ketika RS dalam kondisi *high* atau “1”, data yang dikirimkan adalah data ASCII yang akan ditampilkan di layar. Misal, untuk menampilkan huruf “A” pada layar maka RS harus di-set ke “1”. Jalur kontrol R/W harus berada dalam kondisi *low* (0) saat informasi pada data bus akan dituliskan ke LCD. Apabila R/W berada dalam kondisi *high* “1”, maka program akan

melakukan *query* (pembacaan) data dari LCD. Instruksi pembacaan hanya satu, yaitu *Get LCD status* (membaca status LCD), lainnya merupakan instruksi penulisan. Jadi hampir setiap aplikasi yang menggunakan LCD, R/W selalu di-set ke “0”. Jalur data dapat terdiri 4 atau 8 jalur (tergantung mode yang dipilih pengguna), DB0, DB1, DB2, DB3, DB4, DB5, DB6 dan DB7. Mengirim data secara paralel baik 4-bit atau 8-bit merupakan 2 mode operasi primer.

Mode 8-bit sangat baik digunakan ketika kecepatan menjadi keutamaan dalam sebuah aplikasi dan setidaknya minimal tersedia 11 pin I/O (3 pin untuk kontrol, 8 pin untuk data). Sedangkan mode 4 bit minimal hanya membutuhkan 7-bit (3 pin untuk kontrol, 4 pin untuk data). Bit RS digunakan untuk memilih apakah data atau instruksi yang akan ditransfer antara mikrokontroler dan LCD. Jika bit ini di-set (RS = 1), maka *byte* pada posisi kursor LCD saat itu dapat dibaca atau ditulis. Jika bit ini di-*reset* (RS = 0), merupakan instruksi yang dikirim ke LCD atau status eksekusi dari instruksi terakhir yang di-baca.



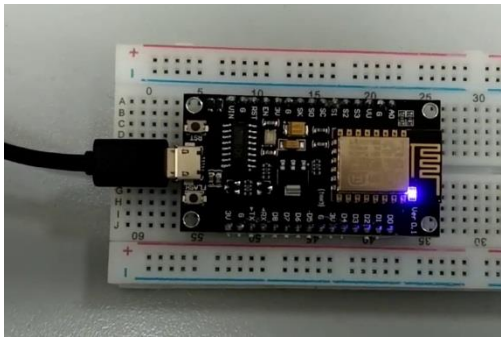
Gambar 2. 9 *Liquid Cristal Display (LCD) 16x2*

2.5 NodeMCU

NodeMCU ESP8266 merupakan sebuah alat *platform wifi database* yang mempunyai sifat *open source*. *Board* elektronik ini telah memiliki *firmware* dan *hardware* yang memiliki fitur wifi. Firmware ini berjalan pada ESP8266 dan *hardware* berupa modul ESP-12. NodeMCU ini dapat diprogram menggunakan bahasa *Arduino (C/C++)*, *Lua* dan *Micropyton*. NodeMCU ini mempunyai spesifikasi yang lebih baik dibanding arduino uno yaitu memori sebesar 4 MB dengan kemampuan RAM 128 KB dan kecepatan *clock* 80 MHz. Selain hal itu fitur adanya *Wifi* yang bisa terintegrasi menambah nilai sendiri pada board ini yang nantinya bisa mengirimkan data secara jarak jauh [8].

Tabel 2. 3 Perbandingan Pin Arduino dengan NodeMCU

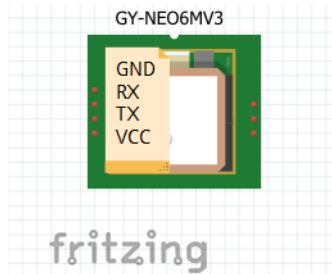
Item	Arduino Uno	NodeMCU
Pin <i>Digital</i> I/O	14 Pin I/O	12/9 Pin I/O
Pin PWM	6	11/9
<i>Analog Input</i>	6	1
<i>Flash Memory</i>	32 KB	4 MB termasuk EEPROM
RAM	2 KB	128 KB
Eeprom	1 KB	-
Kecepatan <i>Clock</i>	16 MHZ	80 MHZ



Gambar 2. 10 Board ESP8266 NodeMCU

2.6 GPS U-Blox Neo 6M

Pada penelitian kali ini modul GPS yang digunakan adalah berjenis U-Blox Neo 6M, jenis GPS ini cukup dapat diandalkan karena memiliki keakuratan yang cukup baik dan juga beberapa fitur yang cukup menguntungkan di antaranya terdapat baterai cadangan data, *built-in* elektronik kompas, dan *built-in* antenna keramik untuk menangkap sinyal dengan kuat. Kemudian untuk dapat mengkomunikasikan GPS ini dengan Arduino diperlukan sebuah *library* yang bernama “TinyGPS++.h”. Bentuk dari modul GPS uBlox Neo 6M dapat dilihat pada Gambar 2.4 [9].

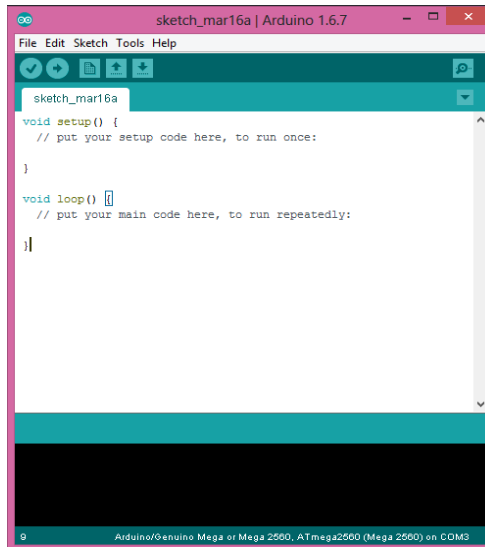


Gambar 2. 11 GPS U-Blox Neo 6

2.7 Arduino IDE

Arduino IDE adalah perangkat lunak yang digunakan untuk memprogram, *Monitoring* dan *debugging* mikrokontroler Arduino. IDE itu merupakan kependekan dari *Integrated Development Environment*, atau secara bahasa mudahnya merupakan lingkungan terintegrasi yang digunakan untuk melakukan pengembangan. Disebut sebagai lingkungan karena melalui *software* inilah Arduino dilakukan pemrograman untuk melakukan fungsi-fungsi yang diberikan melalui sintaks pemrograman. Arduino menggunakan bahasa pemrograman sendiri yang menyerupai bahasa C.

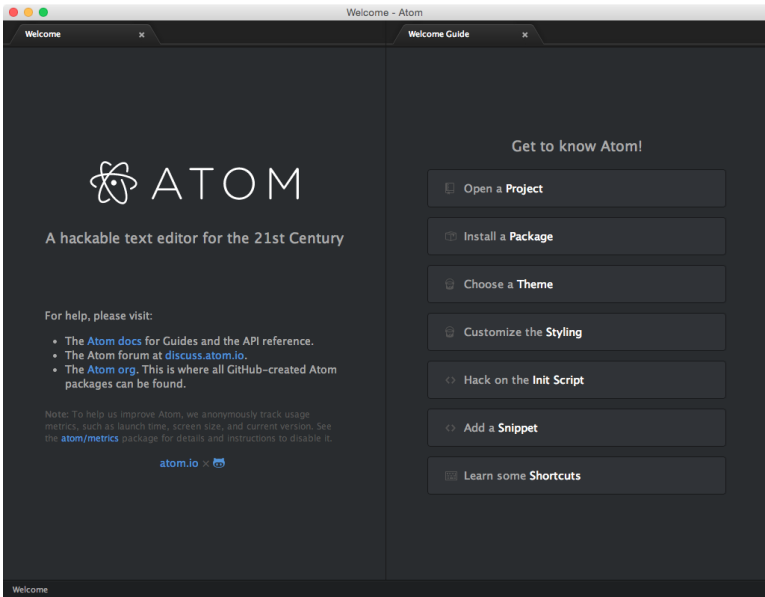
Bahasa pemrograman Arduino (*Sketch*) sudah dilakukan perubahan untuk memudahkan pemula dalam melakukan pemrograman dari bahasa aslinya. Sebelum dijual ke pasaran, IC mikrokontroler Arduino telah ditanamkan suatu program bernama *Bootlader* yang berfungsi sebagai penengah antara *compiler* Arduino dengan mikrokontroler. Pada Gambar 2.2 merupakan tampilan awal untuk membuat program pada *software* Arduino IDE [7].



Gambar 2. 12 Arduino IDE

2.8 Atom Editor

Atom merupakan *text editor* yang disediakan dengan gratis atau *freeware* yang bertujuan sebagai media untuk membuat sebuah *code* atau program dengan berbagai macam bahasa pemrograman, seperti: HTML, PHP, dan C++. Atom memiliki beberapa fitur-fitur yang disediakan terhadap *user* untuk dapat memudahkan dalam membangun sebuah sistem, seperti *auto complete text*, *themes*, *cross platform*, dan *multiple panes* [10].



Gambar 2. 13 Tampilan Atom Editor

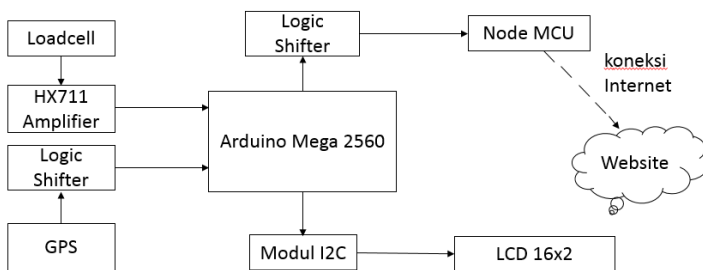
-----Halaman ini sengaja dikosongkan-----

BAB III

PERANCANGAN SISTEM DAN IMPLEMENTASI

Pada bab ini membahas tentang tahapan yang dilakukan terhadap perancangan dan pembuatan Proyek Akhir yang berjudul *Sistem Monitoring Dan Data Logging Pada Jasa Ekspedisi Pengiriman Barang Berbasis Web*. Pada tahap perancangan ini, hal-hal yang perlu untuk dirancang adalah sebagai berikut : Pada *hardware* yang dilakukan adalah perancangan *Shield* Arduino Mega, perancangan rangkaian NodeMCU, perancangan bak truk timbang, perancangan rangkaian LCD dan modul I2C, perancangan rangkaian Sensor *Loadcell*, dan perancangan rangkaian *Logic Shifter*.

Pada *software* yang dilakukan adalah pemrograman pada *software* Arduino IDE yang meliputi : programming sensor *Loadcell* dan beberapa komponen tambahan lainnya. Selanjutnya adalah pemrograman *Web*, dalam melakukan pemrograman *Web* ini *software* yang digunakan adalah *Atom editor* berfungsi untuk membuat desain *Web*, meliputi konten dan tampilan pada *Web*. Kemudian yang terakhir adalah merancang komunikasi data antara alat dengan *Web*.



Gambar 3. 1 Blok DiaGram Sistem

Berikut penjelasan mengenai blok diagram pada Gambar 3.1 yaitu Arduino Mega 2560 sebagai mikrokontroler. Sensor *Loadcell* mendeteksi beban yang ditimbang di bak truk, besaran tersebut perlu dirubah menjadi data digital karna *output* pada *Loadcell* sangat kecil yaitu 0-20 mV sehingga perlu komponen ADC HX711 sebelum masuk ke mikrokontroler. Modul GPS mendeteksi posisi truk untuk mengetahui

lokasi saat barang hilang, tegangan *input* yang disarankan untuk modul GPS adalah 3.3V sehingga perlu komponen *Logic Shifter* untuk menurunkan tegangan dari 5V ke 3.3V. selain itu *Logic Shifter* berfungsi untuk menghubungkan mikrokontroller dengan ESP 8266-12. Dimana mikrokontroller Arduino Mega 2560 tegangan *Output* dan *input*nya sebesar 5Vdc, sedangkan ESP 8266-12 (NodeMCU) hanya 3,3Vdc. dan hasil pembacaan beban dari sensor *Loadcell* kepada esp8266-12 yang kemudian akan dikirim ke *database* sehingga dapat diakses secara *online* melalui *Web* yang telah dibuat.

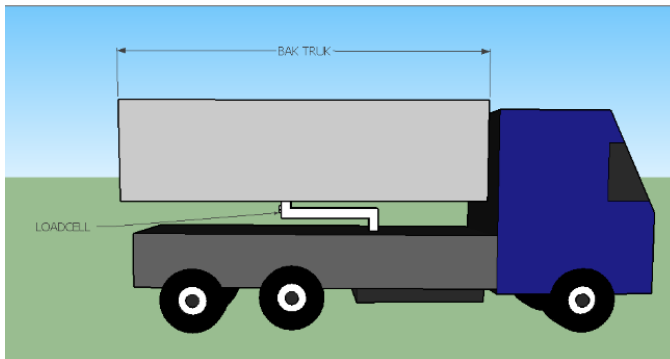
3.1 Perancangan *Hardware*

Pada perancangan *hardware* dibagi menjadi beberapa sub bab yang akan dijelaskan per sub bab nya, antara lain :

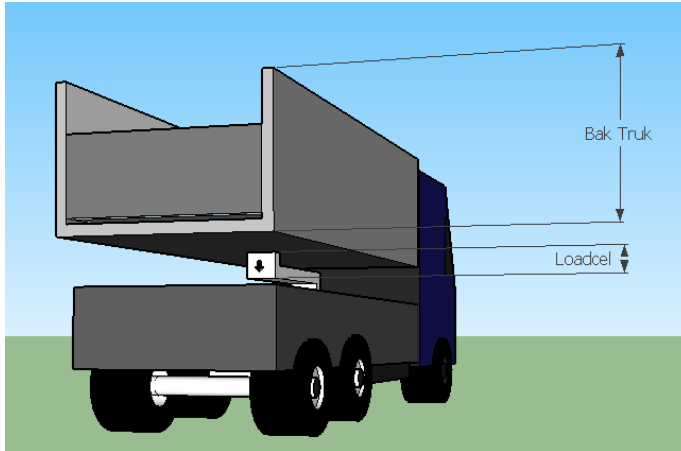
1. Perancangan Bak Truk Timbang
2. Rangkaian *Shield* Arduino Mega 2560
3. Perancangan Rangkaian Sensor *Loadcell*
4. Perancangan I2C dan LCD *Display*
5. Rangkaian GPS Neo 6 U-blox
6. Rangkaian Koneksi Arduino dengan NodeMCU

3.1.1 Perancangan Bak Truk Timbang

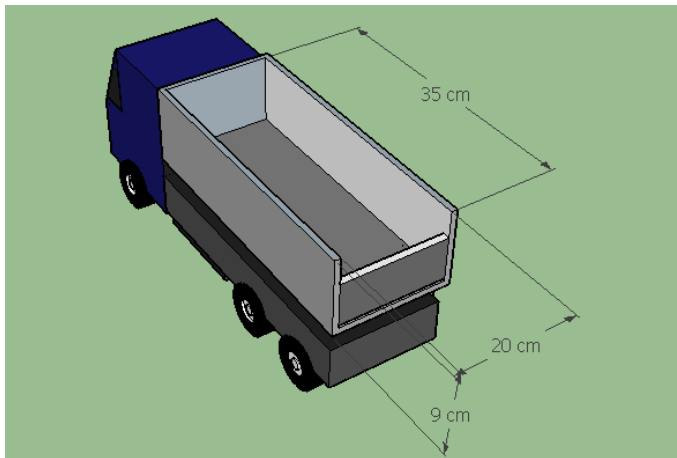
Desain Perancangan Bak Truk Timbang. *Loadcell* diletakkan di bawah bak truk agar dapat melakukan penimbangan berat setiap saat.



Gambar 3. 2 Desain Bak Truk Tampak Samping



Gambar 3. 4 Desain Bak Truk Tampak Belakang



Gambar 3. 3 Desain Bak Truk Tampak Atas

Pada penelitian ini hanya menggunakan satu sensor *Loadcell* dan diletakkan di bawah bak truk di titik berat bak truk agar pembacaan beban oleh *Loadcell* dapat terbaca sama di setiap sudut bak truk.

3.1.2 Perancangan Rangkaian *Shield* Arduino Mega 2560

Pada perancangan rangkaian kontroler ini menggunakan Arduino Mega 2560 untuk menerima masukan berat beban dari sensor *Loadcell*, menerima data posisi dari GPS Neo 6 U-Blox, kemudian data-data dari Arduino Mega 2560 dikirim ke ESP 8266-12 (NodeMCU), serta menampilkan data *Loadcell* pada LCD display dan menjalankan program untuk komunikasi alat dengan PC via koneksi internet. Berikut ini pada Tabel 3.1 merupakan tabel untuk penggunaan pin pada *Shield* Arduino yang akan dibuat :

Tabel 3. 1 Perancangan Rangkaian *Shield* Arduino Mega 2560

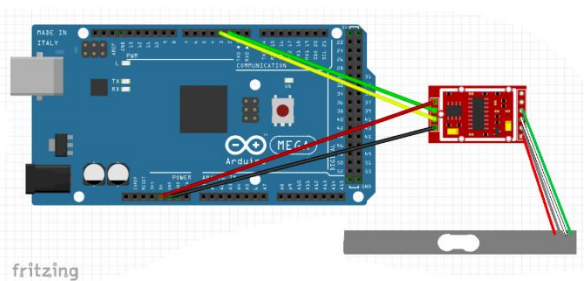
No.	Modul	Pin Analog/Digital
1	GPS Neo 6 U-blox	VCC : 3.3Vdc GND : GND RX : RX3 TX : TX3
2	HX-711	VCC : 5Vdc GND : GND DOUT : Digital 6 CLK : Digital 5
3	I2C	VCC : 5Vdc GND : GND SDA : Digital 20 SCL : Digital 21
4	<i>Logic Shifter</i>	AVCC : 5Vdc ASCL : RX1 pin (digital 19) ASDA : TX1 pin (digital 18) AGND : GND

3.1.3 Perancangan Rangkaian Sensor *Loadcell*

Sensor *Loadcell* merupakan sensor yang berfungsi untuk mengukur beban, dalam Proyek Akhir ini sensor *Loadcell* berfungsi untuk mengukur beban yang diangkut pada bak truk. Keluaran dari sensor ini hanya sebesar 0-20mVdc sehingga diperlukan penguat dalam Proyek Akhir ini tipe *amplifier* yang digunakan adalah modul HX711 yang merupakan *Analog to Digital Converter* sekaligus *Amplifier* yang memang biasa digunakan sebagai penguat sinyal *output* dari *Loadcell*. Berikut adalah tabel koneksi sensor *Loadcell* dengan modul HX711, dijelaskan pada Tabel 3.2.

Tabel 3. 2 Perancangan Rangkaian Sensor *Loadcell*

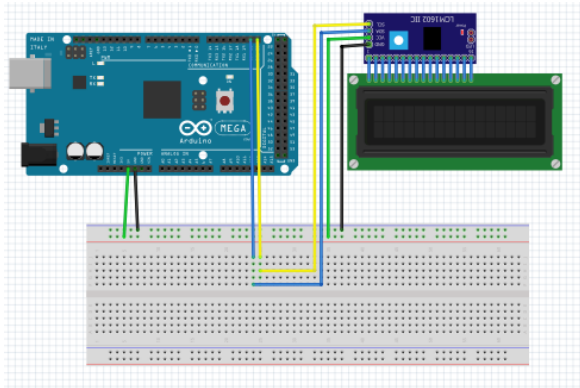
No	Sensor <i>Loadcell</i>	Modul HX-711
1	<i>Red cable (Loadcell power)</i>	E +
2	<i>Black cable (Loadcell gnd)</i>	E -
3	<i>White cable (gnd output)</i>	A -
4	<i>Green cable (positive output)</i>	A +



Gambar 3. 5 Perancangan Rangkaian *Loadcell*

3.1.4 Perancangan Rangkaian I2C dan LCD Display

LCD (*Liquid Cristal Display*) adalah salah satu komponen elektronika yang berfungsi untuk menampilkan suatu *Output* berupa nilai atau teks pada mikrokontroller. Dalam hal ini LCD berfungsi untuk menampilkan data sensor *Loadcell* secara langsung sehingga sopir dapat memantau secara langsung data sensor *Loadcell*, data yang ditampilkan adalah beban yang ada di bak truk. Pada Proyek Akhir ini LCD display 16x2 dihubungkan dengan modul I2C dimana modul I2C ini dapat memperkecil penggunaan pin pada mikrokontroller Arduino Mega 2560. Modul ini menggunakan *protocol* I2C (*Inter Intergrated Circuit*) atau TWI (*Two Wire Interface*). Sehingga LCD yang awalnya membutuhkan banyak pin pada Arduino. Menjadi lebih hemat 4 pin saja, yaitu SCL (*Serial Clock*), SDA (*Serial Data*), VCC, dan ground. Dimana pin SDA dan SCL ini lah yang membawa informasi data antara I2C dan mikrokontroler Arduino Mega2560. Berikut adalah rangkaian yang digunakan serta konfigurasi pin I2C ke Arduino mega 2560 seperti pada Gambar 3.6.



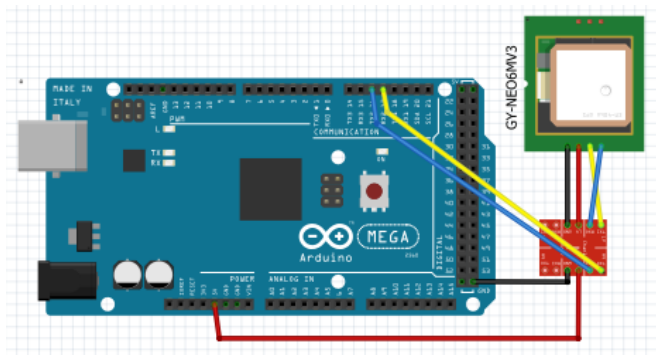
Gambar 3. 6 Rangkaian I2C LCD 16x2

Tabel 3. 3 Perancangan I2C dan LCD Display

No	Arduino	I2C
1	5 Vdc	VCC
2	GND	GND
3	Digital 20	SDA
4	Digital 21	SCL

3.1.5 Perancangan Rangkaian GPS Neo 6 U-blox

Global Positioning System (GPS) adalah sistem untuk menentukan letak di permukaan bumi dengan bantuan penyelarasan (*synchronization*) sinyal satelit. Sistem ini menggunakan 24 satelit yang mengirimkan sinyal gelombang mikro ke Bumi. Sinyal ini diterima oleh alat penerima di permukaan, dan digunakan untuk menentukan letak, kecepatan, arah, dan waktu.



Gambar 3. 7 Perancangan Rangkaian GPS Neo 6 U-blox

3.1.6 Perancangan Rangkaian Koneksi Arduino dengan NodeMCU

Koneksi antara Arduino Mega 2560 dengan ESP 8266-12 (NodeMCU) diperlukan karna Arduino Mega 2560 belum memiliki modul Esp8266 dalam buildnya sehingga untuk mengirim data dari Arduino mega ke jaringan perlu perangkat tambahan dalam Proyek Akhir ini yang digunakan adalah NodeMCU. Untuk menghubungkan kedua *device* ini komunikasi yang digunakan adalah *Serial* komunikasi dari pin RX dan TX tiap *device*. Namun tegangan pada pin IO Arduino mega adalah sebesar 5 Vdc. Sedangkan pada NodeMCU adalah 3.3Vdc. Sehingga diperlukan alat penghubung untuk mengkonversikan tegangan 5Vdc dari Arduino Mega 2560 ke tegangan 3.3Vdc dari NodeMCU. Oleh karena itu digunakanlah *Logic Shifter* untuk mengkonversi besaran tersebut.

Tabel 3. 4 Konfigurasi Pin Arduino dengan *Logic Shifter*

No.	Arduino Mega 2560	Logic Shifter
1	+5VDC	AVCC
2	GND	AGND
3	RX2	ASCL
4	TX2	ASDA

Tabel 3. 5 Konfigurasi Pin NodeMCU dengan *Logic Shifter*

No.	NodeMCU	Logic Shifter
1	+3,3 Vdc	BVCC
2	GND	BGND
3	TX	BSCL
4	RX	BSDA

3.2 Perancangan Software

Pada perancangan *Software* untuk alat ini dibagi menjadi dua yaitu perancangan *Software* untuk alat dan komunikasi data menggunakan ESP8266-12E NodeMCU serta perancangan *Web* menggunakan *Software* Atom dan Arduino IDE

3.2.1 Perancangan Program Arduino

Perancangan program pada Arduino ini berkaitan dengan pembuatan program untuk sub-sub *system* yang berupa sensor, aktuator, beberapa modul tambahan serta untuk pengiriman data oleh modul ESP8266-12 dari alat untuk ditampilkan pada aplikasi *Web* melalui koneksi *Internet*. Langkah awal dalam suatu pemrograman adalah deklarasi *variable* dan tipe data yang digunakan. *Variable* dan tipe data ini digunakan untuk mempermudah proses pemrograman. Berikut pada Gambar 3.8 merupakan deklarasi *variable* dan tipe data yang akan digunakan.

```
// I2C LCD16X2
LiquidCrystal_I2C lcd (0x27, 16, 2)
// GPS U-BLOX NEO6
TinyGPSPlus gps;
//LOADCELL
#define DOUT 6
#define CLK 5
HX711 scale(6, 5);
float calibration_factor = 89.9;
float units;
```

Gambar 3. 8 Deklarasi *Variable*

Pada Gambar 3.8 pendefinisian pin DOUT dihubungkan dengan pin 6 Arduino Mega 2560 dan pin SCK dihubungkan pada pin 5 Arduino Mega 2560 untuk pendefinisian kalibrasi faktor sensor *Loadcell* sebesar 89 yang digunakan untuk membaca beban dengan satuan Gram. Selanjutnya adalah pengaturan pada *void setup*, *void setup* digunakan untuk menginisialisasi pin digital yang kita pakai sebagai *input* atau *output*, inisialisasi komunikasi serial, LCD *Display*, serta beberapa perangkat modul pendukung seperti yang perlu diinisialisasikan yang terdapat pada Gambar 3.9.

```
void setup() {  
    Serial.begin(9600);  
    Serial1.begin(9600);  
    Serial2.begin(9600);  
    Serial3.begin(9600);  
    lcd.begin();  
    lcd.setCursor(0, 0);  
    lcd.print("Baca Loadcell");  
    Serial.begin(9600);  
    scale.set_scale(calibration_factor);  
    scale.tare();  
    Serial.println("Readings:");  
}
```

Gambar 3. 9 Void Setup

Selanjutnya adalah pembuatan program pada *void Loop*, dimana program pada *void Loop* ini yang berfungsi untuk mengolah data *input* dan *Output* dari Arduino Mega 2560 program ini akan diulang terus menerus oleh Arduino dan berfungsi sebagai *list control* utama pada pemrograman Arduino Mega 2560. Untuk lebih jelasnya dapat dilihat pada Gambar 3.10 dan Gambar 3.11 berikut:

```

void bacaGPS() {
    if (Serial3.available() > 0) {
        gps.encode(Serial3.read());
    }

    if (gps.location.isUpdated()) {
        Serial.print("Latitude= ");
        Serial.println(gps.location.lat(), 7);
        Serial.print(" Longitude= ");
        Serial.println(gps.location.lng(), 7);
        double latFloat = gps.location.lat();
        double longFloat = gps.location.lng();

        bacaLoad();
    }
}

```

Gambar 3. 10 *Void Loop* Untuk Sensor GPS

Program *void Loop* untuk sensor GPS digunakan untuk membaca berulang-ulang dan ditampilkan dalam bentuk angka koordinat. Pada Gambar 3.10 program dimulai saat *Serial* tiga terdeteksi dan ketika ada data baru terdeteksi maka Arduino akan menampilkan data *latitude* dan data *longitude*.

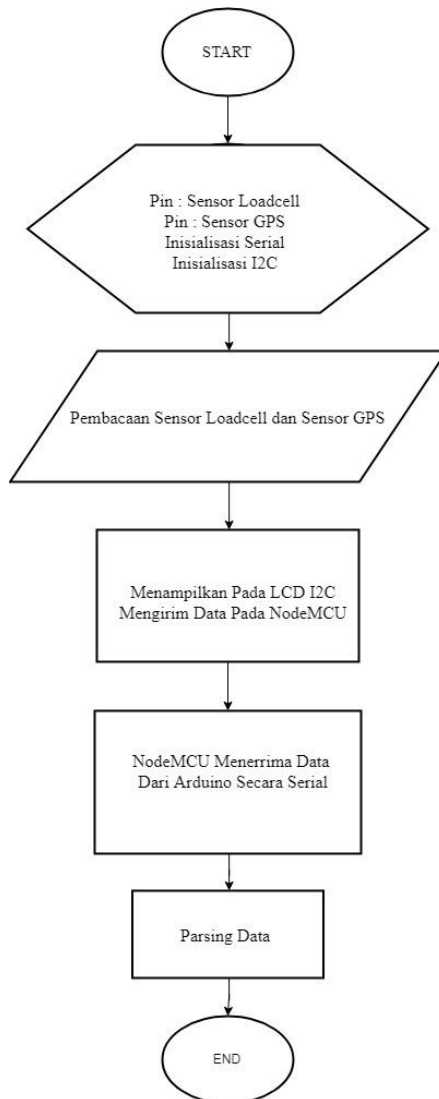
```

void bacaLoad() {
    Serial.print("Reading: ");
    units = abs(scale.get_units());
    Serial.print(units, 1);
    Serial.print(" gram");
    Serial.println();
    double latFloat = gps.location.lat();
    double longFloat = gps.location.lng();
    Serial1.println(latFloat, 7);
    Serial1.println(longFloat, 7);
    Serial1.println(units, 1);
    Serial2.println(latFloat, 7);
    Serial2.println(longFloat, 7);
    Serial2.println(units, 1);
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("berat=");
    lcd.print(units);
    lcd.print("gram");
    delay(1000);
}

```

Gambar 3. 11 *Void Loop Untuk Sensor Loadcell*

Pada Gambar 3.11 adalah *void loop* sensor *Loadcell* pertama Arduino akan menampilkan kata “*Reading*” setelah itu melakukan pembacaan sensor dan *delay* selama 2 detik untuk melakukan *tare* yaitu kalibrasi awal mengenolkan pembacaan sebelum diberi beban. Ketika data terbaca maka Arduino akan menampilkan data pada LCD serta mengirim data ke NodeMCU melalui komunikasi *Serial* satu dan dua setiap satu detik.



Gambar 3. 12 *Flowchart* Pengiriman data dari Arduino Mega 2560 ke NodeMCU

Pada Gambar 3.12 Pengiriman data dari Arduino Mega 2560 ke NodeMCU menggunakan komunikasi *Serial* RX1TX1 dan *Serial* RX3TX3. Data yang dikirim adalah data asil pembacaan oleh sensor *Loadcell* dan sensor GPS.

3.2.2 Pemrograman NodeMCU

NodeMCU sebagai subsistem IoT berperan penting dalam melakukan pengolahan data untuk pengiriman data hasil pembacaan sensor pada *database*. Dalam pemrograman NodeMCU menggunakan *software* Arduino IDE. Terdapat dua proses yang dilakukan oleh NodeMCU yaitu menerima data pembacaan sensor dari Arduino Mega 2560 dan mengirimkan data pembacaan sensor pada *database*.

```
#include <ESP8266WiFi.h>
#include <FirebaseArduino.h>

#define FIREBASE_HOST "trucknet-ioter.firebaseio.com"
#define FIREBASE_AUTH "sY20kUU59BLr9pScNDR5fyrG6EZkAF4z5uvo10c6"
#define WIFI_SSID "koneksi"
#define WIFI_PASSWORD "internet"

String dataArray[3];
String temp;
int selesaiBaca,i;
```

Gambar 3. 13 Deklarasi Awal

Deklarasi awal digunakan untuk mendeskripsikan variabel-variabel yang akan digunakan dalam program utama dan juga untuk menambahkan file-file program yang dibutuhkan untuk menjalankan program utama. Pada Gambar 3.13 digunakan untuk menginisialisasi alamat *database* yang digunakan, sambungan internet yang digunakan, menentukan jumlah data *array* yang ditampilkan pada Gambar 3.13 data *array* yang ditampilkan berjumlah 3 data *array* lalu data yang terbaca dikirim ke *Firebase* untuk ditampilkan pada *Web*.

```

void setup() {
  Serial.begin(9600);
  // connect to wifi.
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("connecting");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
  Serial.println();
  Serial.print("connected: ");
  Serial.println(WiFi.localIP());

  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
}

```

Gambar 3. 14 *Void Setup* Program NodeMCU

Pada Gambar 3.14 *Void Setup* digunakan untuk menginisiasi variabel, mengatur mode pin pada *board*, mengatur *timer*, mengatur *baudrate* *Serial port* dan sebagainya. Bagian setup hanya dijalankan sekali saja yaitu pada saat awal program dijalankan atau ketika program direset. Pada Gambar 3.13 program dijalankan sekali untuk menghubungkan alat dengan internet, ketika alat terhubung dengan internet NodeMCU akan menjeda selama 500 Milidetik lalu menampilkan kata ("*connected*") setelah itu menampilkan alamat IP internet yang terhubung.

```

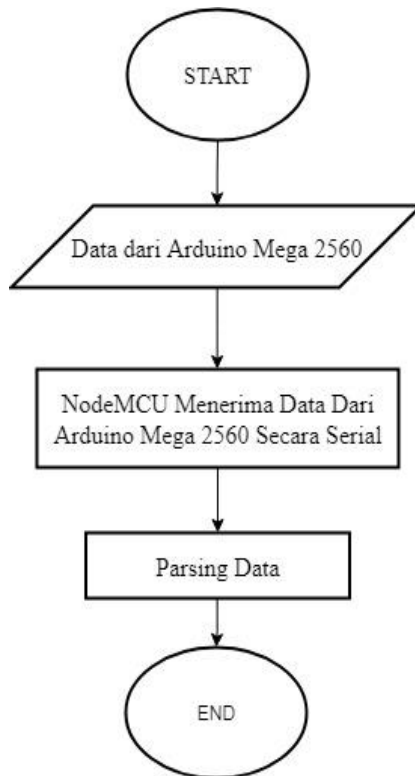
void loop() {
    while(Serial.available() > 0){
        for(i=0;i<3;i++){
            digitalWrite(D0,1);
            temp=Serial.readStringUntil('\n');
            dataArray[i]=temp;
            //i++;
            selesaiBaca=1;
            temp=" \b";
        }
        if (selesaiBaca==1){
            Serial.println("Data Terbaca :");
            Serial.println(dataArray[0]);
            Serial.println(dataArray[1]);
            Serial.println(dataArray[2]);

            Firebase.setString("Latitude3",String(dataArray[0]));
            Firebase.setString("Longitude3",String(dataArray[1]));
            Firebase.setString("Load3",String(dataArray[2]));
            selesaiBaca=0;
            digitalWrite(D0,0);
        }
    }
}

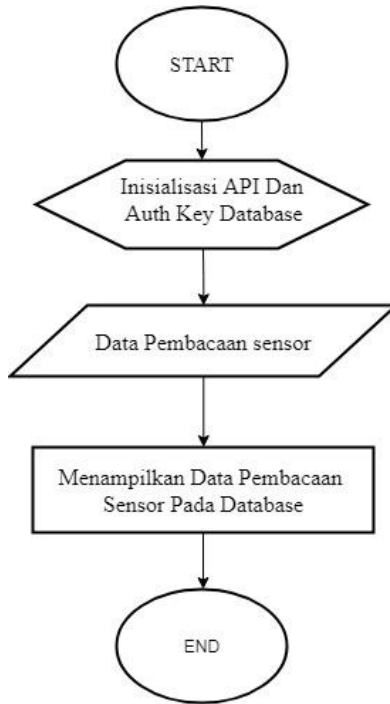
```

Gambar 3. 15 Void Loop Program NodeMCU

Loop adalah fungsi utama program yang akan dijalankan berulang-ulang. dimana program pada *void Loop* ini yang berfungsi untuk mengolah data *input* dan *Output* dari Arduino Mega 2560 program ini akan diulang terus menerus oleh Arduino dan berfungsi sebagai list *control* utama pada pemrograman Arduino Mega 2560. Pada Gambar 3.15 NodeMCU akan memulai program ketika ada data baru yang dikirim melalui komunikasi serial. Selanjutnya jika ada data baru yang terkirim ke NodeMCU, NodeMCU akan menampilkan kata “Data Terbaca” dan data *array* selanjutnya data *array* dikirim ke *firebase* menggunakan komunikasi internet.



Gambar 3. 16 *Flowchart* Penerimaan Data oleh NodeMCU dari Arduino Mega 2560



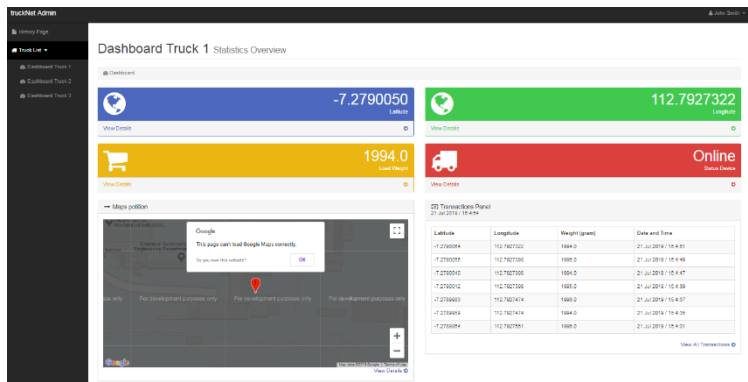
Gambar 3. 17 Flowchart Pengiriman Data Pada Database Melalui NodeMCU

Untuk dapat mengirim data pembacaan sensor pada *database* terlebih dahulu mengubah *rule* yang ada pada *database* untuk *write* menjadi *true*. Kemudian harus menginisialisasi API dan *auth key* yang didapatkan pada proyek yang dibuat pada *firebase*.

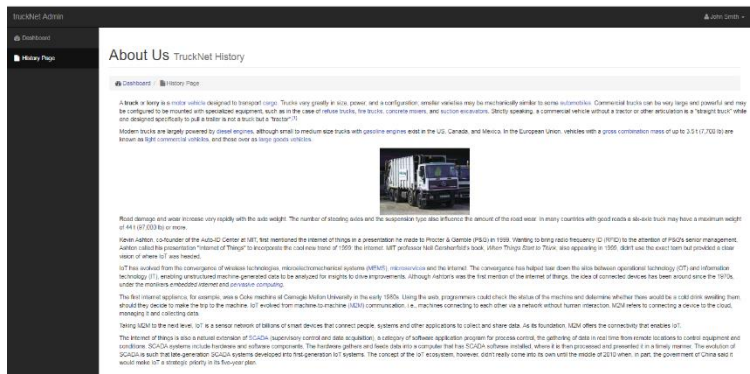
3.2.3 Pembuatan Desain Web

Berikutnya adalah pembuatan desain Website, dalam pembuatan desain Web ini *software* yang digunakan adalah Atom, *software* ini memiliki fungsi untuk membuat desain Web dengan berbagai jenis bahasa pemrograman. Bahasa pemrograman yang digunakan dalam Proyek Akhir ini adalah CSS (*Cascading Style Sheet*), HTML (*Hypertext Markup Language*), dan PHP (PHP: *Hypertext Preprocessor*). Dalam

Proyek Akhir ini ada beberapa halaman yang dibuat yaitu *Home page* dan *History page*. Seperti Gambar 3.18 dan Gambar 3.19.



Gambar 3. 18 Pembuatan Desain Web DasBoard



Gambar 3. 19 Pembuatan Desain Web History Page



Gambar 3. 20 Flowchart Menampilkan Data Pada Web Dari Firebase

Database yang digunakan adalah *Firebase* sedangkan struktur *database* pada *Firebase* adalah NoSQL. Untuk mampu membaca *database* pada *Firebase* perlu dilakukan inisialisasi untuk mengonfigurasi *Firebase* JavaScript SDK untuk menggunakan *Authentication*, *cloud storage*, *realtime database* dan *cloud firestore*. Setelah dilakukan inisialisasi maka untuk menampilkan data pada *web* yang ada di *database* perlu dilakukan pemrograman menggunakan JavaScript.

Pemrograman dilakukan menggunakan *software* Atom Editor. Untuk dapat membaca data pada *database* harus mendapat refrensi dari *database* Data *Firebase* diambil dengan menambahkan *listener* asinkron ke *firebase.database.Reference*. *Listener* dipicu sekali untuk status awal

data, dan dipicu kembali setiap kali data berubah. Untuk membaca data di lokasi dan mendeteksi perubahan, digunakan metode *on()* atau *once()* dari *firebase.database.Reference* untuk mengamati peristiwa.

BAB IV

PENGUJIAN DAN ANALISIS DATA

Pada bab ini dilakukan pengujian dan analisa terkait dengan sistem yang telah dirancang dan dibuat. Adapun pada pengujian dan analisa data yang akan dilakukan yaitu:

1. Pengujian Pembacaan Sensor *Loadcell*
2. Pengujian Pembacaan Sensor GPS
3. Pengujian Pengiriman Data dari Arduino ke NodeMcu
4. Pengujian Pengiriman Data Oleh NodeMCU ke *Firebase*
5. Pengujian *Monitoring* dengan *Web*

Dari pengujian diharapkan mampu berjalan sebagaimana perancangan yang telah dibuat agar bisa digunakan untuk memonitoring beban truk.

4.1 Pengujian Pembacaan Sensor *Loadcell*

Pengujian sensor Beban (*Loadcell*) dilakukan dengan memberikan beban pada sensor, dalam penelitian ini, *Loadcell* yang digunakan adalah *Loadcell* dengan kapasitas beban maksimal 50 Kg. Pada pengujian kali ini digunakan empat jenis beban, yaitu 1.000 Gram, 2.000 Gram, 3.000 Gram, dan 4.000 Gram. Dibawah ini merupakan tabel-tabel hasil pengujian keempat beban tersebut :

Tabel 4. 1 Hasil Pengujian Beban 1000 Gram

No.	Tegangan Output <i>Loadcell</i> (mV/V)	Berat Beban Sesungguhnya (Gram)	Berat Beban Terbaca (Gram)	Persentase Kesalahan (%)
1	0.2	1.000	1.012	1,2
2	0.2	1.000	1.013	1,3
3	0.2	1.000	1.011	1,1
4	0.2	1.000	997	0,3
5	0.2	1.000	1.006	0,6
6	0.2	1.000	1.002	0,2
7	0.2	1.000	1.021	2,1
8	0.2	1.000	1.003	0,3
9	0.2	1.000	1.005	0,5
10	0.2	1.000	1.012	1,2

Dari hasil pengujian *Loadcell* beban 1.000 Gram tegangan output yang terukur menggunakan Avometer adalah 0,2 mV/V, seperti yang tertera pada Tabel 4.1, dapat dihitung rata – rata persentase kesalahannya sebagai berikut :

$$\begin{aligned}
 \text{rata – rata kesalahan} &= \frac{\text{jumlah \% kesalahan}}{\text{frekuensi pengujian}} \\
 &= \frac{8,8\%}{10} \\
 &= 0,88\%
 \end{aligned}$$

Persentase kesalahan pada pembacaan berat beban yang dideteksi *Loadcell* apabila dibandingkan dengan berat beban asli, pada beban Gram adalah sebesar 0,88%.

Tabel 4. 2 Hasil Pengujian Beban 2.000 Gram

No.	Tegangan Output Loadcell (mV/V)	Berat Beban Sesungguhnya (Gram)	Berat Beban Terbaca (Gram)	Persentase Kesalahan (%)
1	0.4	2.000	2.012	0,6
2	0.4	2.000	2.013	6,5
3	0.4	2.000	2.011	5,5
4	0.4	2.000	1.999	0,5
5	0.4	2.000	2.019	9,5
6	0.4	2.000	2.001	0,5
7	0.4	2.000	1.998	0.1
8	0.4	2.000	2.003	1,5
9	0.4	2.000	2.017	8,5
10	0.4	2.000	2.004	0,2

Dari hasil pengujian *Loadcell* beban 2.000 Gram tegangan output yang terukur menggunakan Avometer adalah 0,4 mV/V, seperti yang tertera pada Tabel 4.2, dapat dihitung rata – rata persentase kesalahannya sebagai berikut :

$$\begin{aligned}
 \text{rata – rata kesalahan} &= \frac{\text{jumlah \% kesalahan}}{\text{frekuensi pengujian}}
 \end{aligned}$$

$$= \frac{33,4\%}{10}$$

$$= 3,34\%$$

Persentase kesalahan pada pembacaan berat beban yang dideteksi *Loadcell* apabila dibandingkan dengan berat beban asli, pada beban Gram adalah sebesar 3,34%.

Tabel 4. 3 Hasil Pengujian Beban 3.000 Gram

No.	Tegangan Output Loadcell (mV/V)	Berat Beban Sesungguhnya (Gram)	Berat Beban Terbaca (Gram)	Persentase Kesalahan (%)
1	0,6	3.000	2.996	0,13
2	0,6	3.000	2.980	0,6
3	0,6	3.000	2.993	2,3
4	0,6	3.000	3.004	1,3
5	0,6	3.000	2.980	0,6
6	0,6	3.000	3.003	1,1
7	0,6	3.000	2.983	0,56
8	0,6	3.000	2.974	0,86
9	0,6	3.000	3.001	0,03
10	0,6	3.000	2.975	0,83

Dari hasil pengujian *Loadcell* beban 3.000 Gram tegangan output yang terukur menggunakan Avometer adalah 0,6 mV/V, seperti yang tertera pada Tabel 4.3, dapat dihitung rata – rata persentase kesalahannya sebagai berikut :

$$\text{rata – rata kesalahan} = \frac{\text{jumlah \% kesalahan}}{\text{frekuensi pengujian}}$$

$$= \frac{8,1\%}{10}$$

$$= 0,831\%$$

Persentase kesalahan pada pembacaan berat beban yang dideteksi *Loadcell* apabila dibandingkan dengan berat beban asli, pada beban Gram adalah sebesar 0,831%.

Tabel 4. 4 Hasil Pengujian Beban 4.000 Gram

No.	Tegangan Output Loadcell (mV/V)	Berat Beban Sesungguhnya (Gram)	Berat Beban Terbaca (Gram)	Persentase Kesalahan (%)
1	0.8	4.000	3.957	1,07
2	0.8	4.000	3.946	1,35
3	0.8	4.000	3.950	1,25
4	0.8	4.000	3.969	7,75
5	0.8	4.000	3.958	1,05
6	0.8	4.000	4.018	0,45
7	0.8	4.000	4.041	1,02
8	0.8	4.000	4.061	1,52
9	0.8	4.000	4.065	1,62
10	0.8	4.000	3.992	0,2

Dari hasil pengujian *Loadcell* beban 2000 Gram tegangan output yang terukur menggunakan Avometer adalah 0,8 mV/V, seperti yang tertera pada Tabel 4.4, dapat dihitung rata – rata persentase kesalahannya sebagai berikut :

$$\begin{aligned}
 \text{rata – rata kesalahan} &= \frac{\text{jumlah \% kesalahan}}{\text{frekuensi pengujian}} \\
 &= \frac{17,28\%}{10} \\
 &= 1,728\%
 \end{aligned}$$

Persentase kesalahan pada pembacaan berat beban yang dideteksi *Loadcell* apabila dibandingkan dengan berat beban asli, pada beban Gram adalah sebesar 1,728%.

4.2 Pengujian Pembacaan Sensor GPS

Pengujian GPS dilakukan untuk memastikan bahwa GPS dapat bekerja dengan baik dengan membandingkan data pembacaan GPS U-Blox Neo 6 dengan data pembacaan Garmin GPS. Pengujian dilakukan ditiga tempat berbeda yaitu di Jalan Wisma Permai II, di Taman Eldas Teknik Elektro Otomasi, dan di Jalan Gang Makam Blok H.

Tabel 4. 5 Tabel Perbandingan Data Pembacaan GPS U-Blox Neo 6 Dengan Data Pembacaan Garmin GPS

No	GPS U-Blox Neo 6		Garmin GPS		Kesalahan Error (Meter)
	<i>Latitude</i>	<i>Longitude</i>	<i>Latitude</i>	<i>Longitude</i>	
1	-7.27500	112.79153	-7.27500	112.79150	3,339
2	-7.27500	112.79153	-7.27500	112.79150	3,339
3	-7.27499	112.79153	-7.27500	112.79150	4,532
4	-7.27890	112.79220	-7.27892	112.79219	3,339
5	-7.27889	112.79220	-7.27892	112.79219	4,452
6	-7.27899	112.79219	-7.27892	112.79219	3,339
7	-7.29318	112.79730	-7.29319	112.79726	5,565
8	-7.29318	112.79732	-7.29319	112.79726	7,792
9	-7.29318	112.79731	-7.29319	112.79726	6,679

Pada Tabel 4.5 pengujian dilakukan di tiga tempat berbeda. Pengujian pertama sampai ketiga dilakukan di Jalan Wisma Permai II, pengujian keempat sampai keenam dilakukan di Taman Eldas Teknik Elektro Otomasi, dan pengujian ketujuh sampai kesembilan dilakukan di Jalan Gang Makam Blok H. Berdasar data pada Tabel 4.5 jarak kesalahan eror titik lokasi yang dibaca oleh GPS U-Blox Neo 6 yang dibandingkan dengan titik sebenarnya yaitu pembacaan titik lokasi oleh Garmin GPS dapat dicari dengan menggunakan rumus *Euclide Distance*.

Perhitungan jarak pembacaan GPS U-Blox Neo 6 dan Garmin GPS menggunakan rumus *Euclide Distance* di Jalan Wisma Permai II

$$\begin{aligned}
 &= \sqrt{(x1 - x2)^2 + (y1 - y2)^2} \\
 &= \sqrt{(lat1 - lat2)^2 + (long1 - long2)^2} \\
 &= \sqrt{((-7.27890) - (-7.27500))^2 + (112.79153 - 112.79150)^2} \\
 &= \sqrt{(0)^2 + (0,00003)^2} \\
 &= 0,00003^\circ
 \end{aligned}$$

Hasil perhitungan tersebut masih dalam satuan *decimal degree* (sesuai dengan format *longitude* dan *latitude* yang dipakai) sehingga untuk menyesuaikannya perlu dikalikan dengan 111.319 km (1 derajat bumi = 111.319 km)

$$\begin{aligned}
 &= 0,00003^\circ \times 113.319 \text{ KM} \\
 &= 0,00339957 \text{ KM} \\
 &= 3,339 \text{ Meter}
 \end{aligned}$$

4.2.1 Pengujian Pertama Sampai Ketiga Sensor GPS

Berdasar data kesalahan eror pada Tabel 4.5 pengujian pertama sampai ketiga yang dilakukan di Jalan Wisma Permai II nilai eror terbesar yaitu 4,532 Meter dan rata-rata kesalahan erornya sebesar :

$$\begin{aligned}
 \text{rata - rata kesalahan} &= \frac{\text{jumlah kesalahan}}{\text{frekuensi pengujian}} \\
 &= \frac{3,339 + 4,532 + 3,339}{3} \\
 &= \frac{11,21}{3} \\
 &= 3,77 \text{ Meter}
 \end{aligned}$$

4.2.2 Pengujian Keempat Sampai Keenam Sensor GPS

Berdasar data kesalahan eror pada Tabel 4.5 pengujian keempat sampai keenam yang dilakukan di Taman Eldas Teknik Elektro Otomasi nilai eror terbesar yaitu 4,452 Meter dan rata-rata kesalahan erornya sebesar :

$$\begin{aligned}
 \text{rata - rata kesalahan} &= \frac{\text{jumlah kesalahan}}{\text{frekuensi pengujian}} \\
 &= \frac{3,339 + 4,452 + 3,339}{3} \\
 &= \frac{11,13}{3} \\
 &= 3,71 \text{ Meter}
 \end{aligned}$$

4.2.3 Pengujian Ketujuh Sampai Kesembilan Sensor GPS

Berdasar data kesalahan eror pada Tabel 4.5 pengujian ketujuh sampai kesembilan yang dilakukan di Jalan Gang Makam Blok H nilai

eror terbesar yaitu 7,792 Meter dan rata-rata kesalahan erornya sebesar:

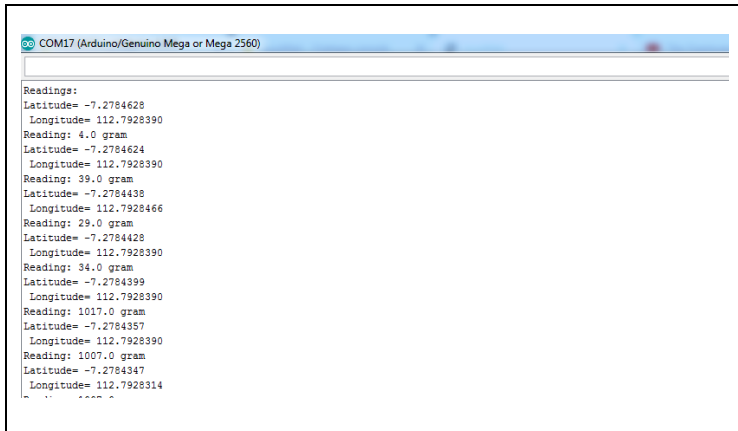
$$\begin{aligned}
 \text{rata - rata kesalahan} &= \frac{\text{jumlah kesalahan}}{\text{frekuensi pengujian}} \\
 &= \frac{5,565 + 7,792 + 6,679}{3} \\
 &= \frac{20,036}{3} \\
 &= 6,678 \text{ Meter}
 \end{aligned}$$

Jadi rata-rata kesalahan pembacaan GPS U-Blox Neo 6 yang dibandingkan dengan pembacaan Garmin GPS di tiga tempat berbeda yaitu di Jalan Wisma Permai II rata-rata erornya sebesar 3,77 Meter, di Taman Eldas Teknik Elektro Otomasi rata-rata erornya sebesar 3,71 Meter, dan di Jalan Gang Makam Blok H rata-rata erornya sebesar 6,678 Meter.

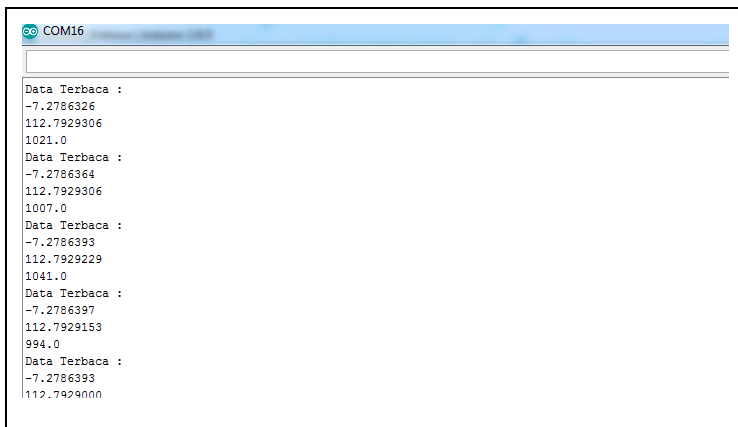
Kesalahan eror pembacaan yang terjadi disebabkan adanya penghalang seperti pohon dan bangunan yang tinggi sehingga menghalangi komunikasi antara GPS dengan satelit yang dapat mengurangi akurasi pembacaan titik lokasi. Selain itu kualitas dari GPS U-blox Neo 6M yang digunakan juga memengaruhi akurasi pembacaan titik lokasi.

4.3 Pengujian Pengiriman Data dari Arduino ke NodeMCU

Pengujian ini dilakukan untuk memastikan bahwa data yang di peroleh oleh Arduino terkirim ke NodeMCU secara serial. Berikut adalah data pengiriman oleh Arduino Mega 2560 dan penerimaan data oleh NodeMCU pada Gambar 4.1 dan Gambar 4.2.



Gambar 4. 1 Data Pembacaan Pada *Serial Monitor* Arduino

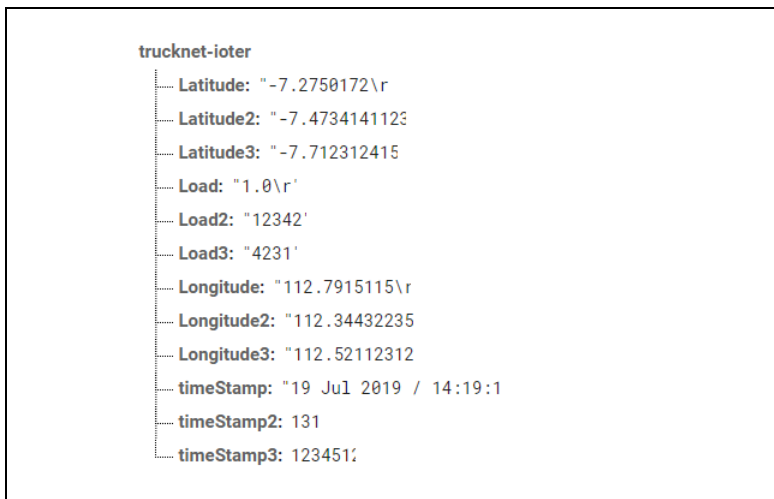


Gambar 4. 2 Data Penerimaan Pada *Serial Monitor* NodeMCU

Data yang dikirim oleh Arduino Mega 2560 ke NodeMCU adalah data hasil hasil Pembacaan Sensor GPS yang berisi data *Latitude* dan *Longitude* sedangkan pembacaan oleh Sensor *Loadcell* berupa data berat. Berdasarkan gambar 4.1 dan gambar 4.2 data-data pembacaan oleh Arduino Mega 2560 berhasil dikirim dan diterima oleh NodeMCU.

4.4 Pengujian Pengiriman Data Oleh NodeMCU ke *Firebase*

Pengujian ini dilakukan untuk memastikan bahwa NodeMCU dapat melakukan pengiriman data ke *database Firebase* dengan komunikasi internet. Pengiriman data dari NodeMCU ke *Firebase* bertujuan untuk menyimpan data pada *database firebase* untuk ditampilkan di *Web*. Pada Gambar 4.3 adalah hasil pengiriman NodeMCU ke *Firebase*

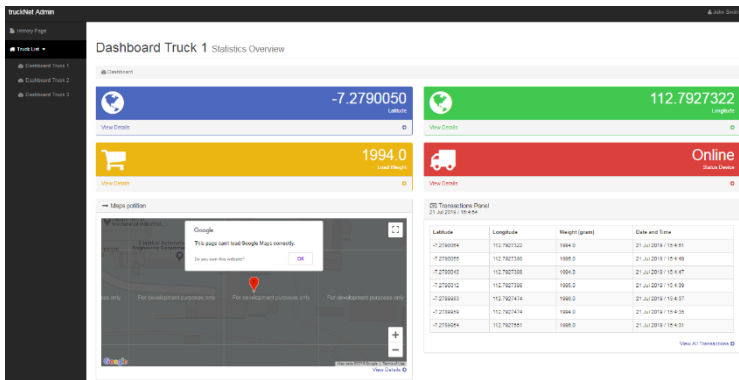


Gambar 4. 3 Hasil Pengiriman Data Dari Nodemcu Ke *Firebase*

Berdasarkan Gambar 4.3 data-data yang dikirim melalui NodeMCU berhasil ditampilkan pada *database Firebase*.

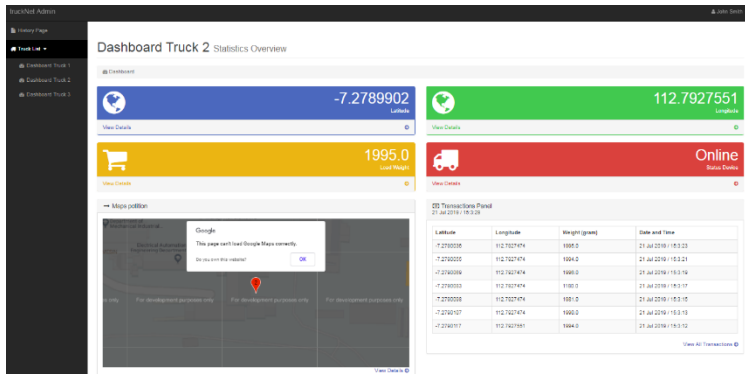
4.5 Pengujian *Monitoring di Web*

Pengujian ini dilakukan bertujuan me*Monitoring* beban truk agar bisa menganalisis waktu dan lokasi jika ada barang yang hilang yang ditandai dengan berkurangnya beban truk. Pada penelitian ini menggunakan 3 NodeMCU yang di analogikan sebagai 3 truk. Setiap NodeMCU mengirim data ke *Dashboard* masing-masing. NodeMCU 1 mengirim data ke *Dashboard Truck 1*, NodeMCU 2 mengirim data ke *Dashboard Truck 2*, dan NodeMCU 3 mengirim data ke *Dashboard Truck 3*. Seperti yang ditampilkan pada Gambar 4.4, Gambar 4.5, dan Gambar 4.6.



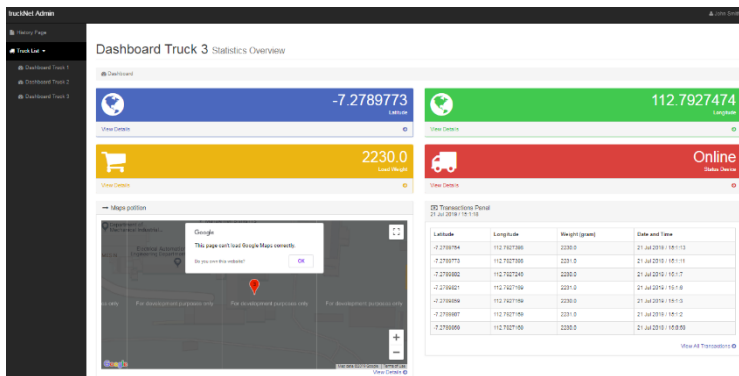
Gambar 4. 4 Tampilan *Dashboard Truck 1*

Pada Gambar 4.4 adalah tampilan dashboard Truck 1 yang menampilkan data-data yang dikirim oleh NodeMCU, pada kotak warna biru menampilkan data *latitude*, pada kotak warna hijau menampilkan data *longitude*, pada kotak warna oranye menampilkan data beban, dan pada kotak warna merah menampilkan status alat nyala atau mati. Selain itu juga ditampilkan maps yang menunjukkan lokasi sesuai koordinat *latitude* dan *longitude*. Serta tampilan tabel yang menampilkan data *latitude*, *longitude*, berat, dan waktu.



Gambar 4. 5 Tampilan *Dashboard Truck 2*

Pada Gambar 4.5 adalah tampilan dashboard Truck 2 yang menampilkan data-data yang dikirim oleh NodeMCU, pada kotak warna biru menampilkan data *latitude*, pada kotak warna hijau menampilkan data *longitude*, pada kotak warna oranye menampilkan data beban, dan pada kotak warna merah menampilkan status alat nyala atau mati. Selain itu juga ditampilkan maps yang menunjukkan lokasi sesuai koordinat *latitude* dan *longitude*. Serta tampilan tabel yang menampilkan data *latitude*, *longitude*, berat, dan waktu.



Gambar 4. 6 Tampilan *Dashboard Truck 3*

Pada Gambar 4.5 adalah tampilan dashboard Truck 2 yang menampilkan data-data yang dikirim oleh NodeMCU, pada kotak warna biru menampilkan data *latitude*, pada kotak warna hijau menampilkan data *longitude*, pada kotak warna oranye menampilkan data beban, dan pada kotak warna merah menampilkan status alat nyala atau mati. Selain itu juga ditampilkan maps yang menunjukkan lokasi sesuai koordinat *latitude* dan *longitude*. Serta tampilan tabel yang menampilkan data *latitude*, *longitude*, berat, dan waktu.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan pengujian dan analisis data, maka dapat ditarik kesimpulan:

1. *System Monitoring* beban dan lokasi truk diakses dengan menggunakan *Webs* yang memanfaatkan koneksi internet. Alat yang menggunakan Arduino Mega 2560 dan NodeMCU ESP8266 serta sensor yang digunakan adalah sensor *Loadcell* dan sensor dan sensor GPS U-blox Neo 6M. Didapatkan kesimpulan dari pengujian sensor *Loadcell* nilai yang terbaca oleh sensor pada pengujian beban 1.000 Gram memiliki persentase *error* pada kisaran 0,2% - 2,1%, pada pengujian beban 2.000 Gram memiliki persentase *error* pada kisaran 0,1% - 9,5%, pada pengujian beban 3.000 Gram memiliki persentase *error* pada kisaran 0,03% - 2,3%, dan pada pengujian beban 4.000 Gram memiliki persentase *error* pada kisaran 0,2% - 7,75%, untuk pengujian sensor GPS U-blox Neo 6M rata-rata kesalahan pembacaan GPS U-Blox Neo 6 yang dibandingkan dengan pembacaan Garmin GPS di tiga tempat berbeda yaitu di Jalan Wisma Permai II rata-rata erornya sebesar 3,77 Meter, di Taman Eldas Teknik Elektro Otomasi rata-rata erornya sebesar 3,71 Meter, dan di Jalan Gang Makam Blok H rata-rata erornya sebesar 6,678 Meter.
2. Pengujian alat keseluruhan yang dilakukan, informasi yang disajikan pada *Web* meliputi data beban yang ada pada truk lokasi truk yang berupa data *latitude* dan *longitude*, status alat aktif atau nonaktif terhubung dengan jaringan internet, tampilan maps, dan tampilan tabel. Alat membaca perubahan data dan semua perubahan data tersebut dikirim dan ditampilkan pada *Web*.

5.2 Saran

Beberapa saran yang penulis dapat berikan untuk pengembangan Proyek Akhir :

1. Menggunakan *database* lain yang dapat menyimpan data banyak.
2. Menggunakan sensor GPS yang memiliki pembacaan lokasi lebih akurat.
3. Perancangan Mekanik bak truk menggunakan bahan plat seng besi agar lebih kuat.
4. Menggunakan sensor *Loadcell* berjumlah 4 yang diletakan di setiap sudut bak truk agar lebih imbang.

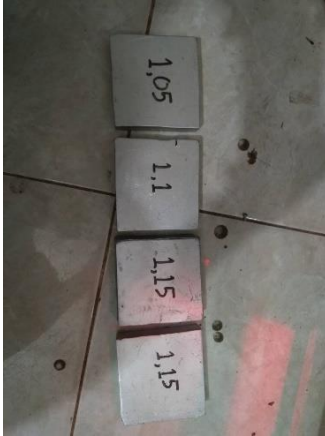
DAFTAR PUSTAKA

- [1] R. W. Agil Jeffry Aditya, "RANCANG BANGUN ALAT MENDETEKSI PENCURIAN MUATAN DI TRUK EKSPEDISI," *Tugas Akhir, Program D3 Teknik Elektro FV-ITS*, 2011.
- [2] W. Erlangga, "Rancang Bangun Timbangan Digital Dengan Pemilihan Jenis Buah," *Skripsi, Jurusan Teknik Elektro, Universitas Negeri Malang, Malang.*, 2011.
- [3] Anonim, "Teori Kelistrikan Loadcell," <http://www.rajaloadcell.com/article/teori-kelistrikan-load-cell-125>, Jakarta, 2015.
- [4] M. A. Suci, "PERANCANGAN TIMBANGAN DIGITAL BUAH DENGAN KELUARAN MASSA DAN HARGA MENGGUNAKAN SENSOR LOAD CELL BERBASIS ATMEGA 328P," *Skripsi*, 2017.
- [5] B. Parkinson, "Global Positioning System: Theory and Applications," in *chap. 1: Introduction and Heritage of NAVSTAR, the Global Positioning System*, Washington, D.C., American Institute of Aeronautics and Astronautics, 1996.
- [6] A. Adiwilaga, "Teori Pengukuran Jarak," <https://blogs.itb.ac.id/anugraha/2014/09/10/teori-pengukuran-jarak/>, Bandung, 2014.
- [7] A. Kadir, *Buku Pintar Pemrograman Arduino*, 2015: MediaKom, Yogyakarta.
- [8] J. Kale, *NodeMCU a Complete Guide*, Denamrk: Emereo Pty Limited , 2018.
- [9] Ajie, "Menangani GPS Ublox Neo Dengan Arduino," <http://saptaji.com/2016/08/27/menangani-gps-ublox-neo-dan-arduino/>, 2016.
- [10] Yasha, "Panduan Dasar Text Editor Atom, Sublime, Notepad++," PT DEWAWEB ID, 2018.
- [11] A. Herjuna, "Trainer Iot Berbasis Esp8266 Sebagai Media Pembelajaran Mata Kuliah Komunikasi Data Dan

- Interface," *TUGAS AKHIR SKRIPSI, STUDI PENDIDIKAN TEKNIK ELEKTRONIKA UNY*, 2018.
- [12] H. Yuliansyah, "Uji Kinerja Pengiriman Data Secara Wireless Menggunakan Modul ESP8266 Berbasis Rest Architecture," *ELECTRICIAN – Jurnal Rekayasa dan Teknologi Elektro*, 2016.
- [13] F. A. Muhammad, "SISTEM KONTROL KECEPATAN MOTOR DC D-6759 BERBASIS ARDUINO MEGA 2560," *JURNAL UNIVERSITAS BRAWIJAYA*, 2011.
- [14] J. Blum, *Exploring Arduino: Tools and Technique for Engineering Wizzardry*, Kanada: Wiley, 2013.

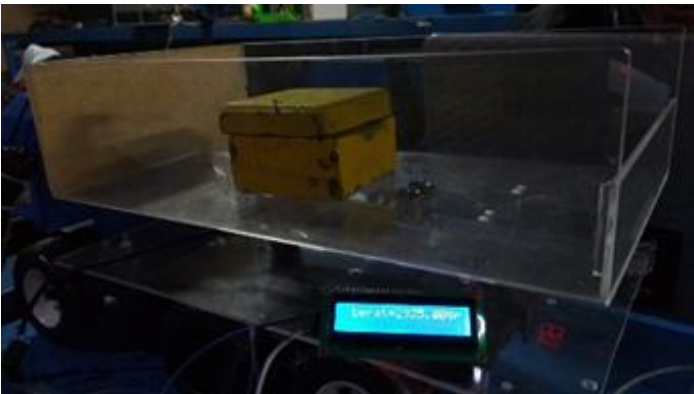
LAMPIRAN A

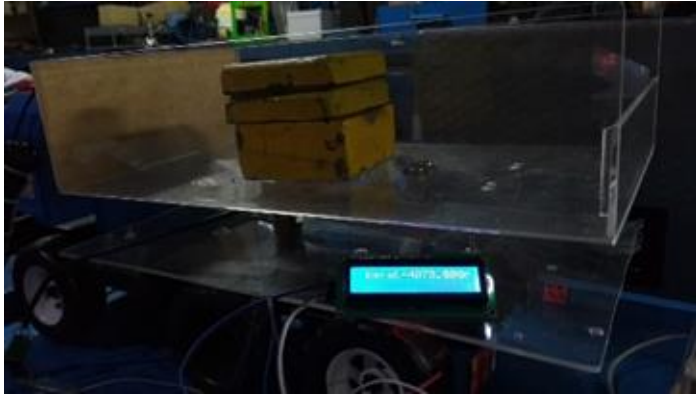
A. Beban-beban yang digunakan untuk pengujian



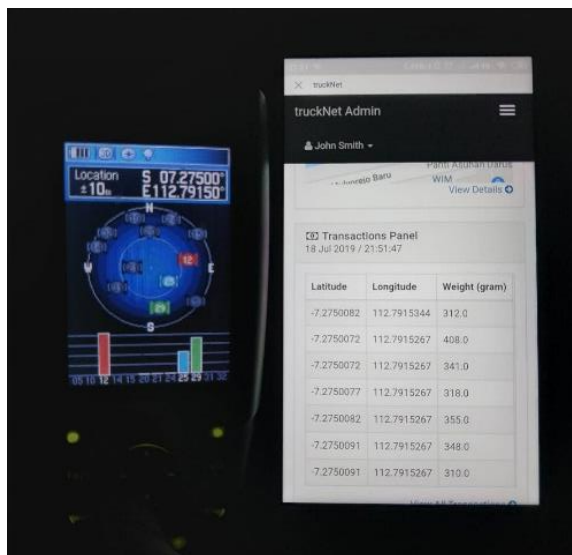
B. Tampilan LCD Display Selama Pengujian *Loadcell*

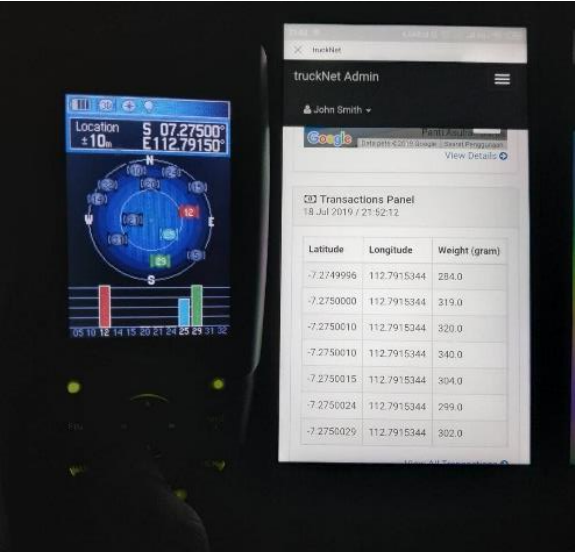


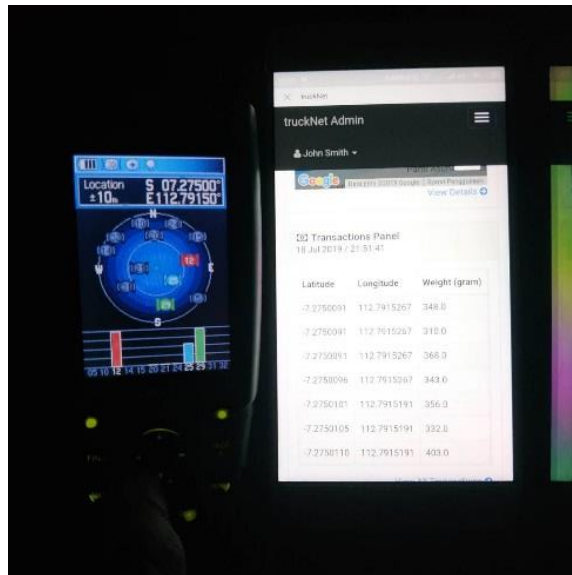




C. Pengujian GPS U-Blox Neo 6 Dengan Garmin GPS







-----Halaman ini sengaja dikosongkan-----

LAMPIRAN B

A. *Listing* Program Arduino

```
#include "HX711.h"
#include <TinyGPS++.h>
#include <LiquidCrystal_I2C.h>
#include <SoftwareSerial.h>

// I2C LCD16X2
LiquidCrystal_I2C lcd (0x27, 16, 2);
// GPS U-BLOX NEO6
TinyGPSPlus gps;
//LOADCELL
#define DOUT 6
#define CLK 5
HX711 scale(6, 5);
float calibration_factor = 89.9;
float units;

void bacaGPS() {
  if (Serial3.available() > 0) {
    gps.encode(Serial3.read());
  }

  if (gps.location.isUpdated()) {
    Serial.print("Latitude= ");
    Serial.println(gps.location.lat(), 7);
    Serial.print(" Longitude= ");
    Serial.println(gps.location.lng(), 7);
    double latFloat = gps.location.lat();
    double longFloat = gps.location.lng();

    bacaLoad();
  }
}

void bacaLoad() {
  Serial.print("Reading: ");
  units = abs(scale.get_units());
```

```

Serial.print(units, 1);
Serial.print(" Gram");
Serial.println();
double latFloat = gps.location.lat();
double longFloat = gps.location.lng();
Serial.println(latFloat, 7);
Serial.println(longFloat, 7);
Serial.println(units, 1);
Serial2.println(latFloat, 7);
Serial2.println(longFloat, 7);
Serial2.println(units, 1);
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("berat=");
lcd.print(units);
lcd.print("Gram");
delay(1000);
}

```

```

void setup() {
  Serial.begin(9600);
  Serial1.begin(9600);
  Serial2.begin(9600);
  Serial3.begin(9600);
  lcd.begin();
  lcd.setCursor(0, 0);
  lcd.print("Baca Loadcell");
  Serial.begin(9600);
  scale.set_scale(calibration_factor);
  scale.tare();
  Serial.println("Readings:");
}

```

```

void Loop() {
  bacaGPS();
  //bacaLoad();
}

```


B. Listing Program NodeMCU

```
#include <ESP8266WiFi.h>
#include <FirebaseArduino.h>

#define FIREBASE_HOST "trucknet-ioter.firebaseio.com"
#define FIREBASE_AUTH
"sY20kUU59BLr9pScNDR5fyrG6EZkAF4z5uvol0c6"
#define WIFI_SSID "koneksi"
#define WIFI_PASSWORD "internet"

String dataArray [3];
String temp;
int selesaiBaca,i;
//////////////////SETUP//////////////////
void setup() {
  Serial.begin(9600);
  // connect to wifi.
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("connecting");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
  Serial.println();
  Serial.print("connected: ");
  Serial.println(WiFi.localIP());
  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
}

////////////////// LOOP ////////////////////
void Loop() {
  while(Serial.available()> 0){
    for(i=0;i<3;i++){
```

```

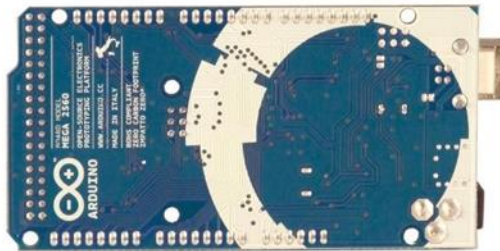
digitalWrite(D0,1);
temp=Serial.readStringUntil('\n');
dataArray [i]=temp;
selesaiBaca=1;
temp=" \b";
}
}
if (selesaiBaca==1){
Serial.println("Data Terbaca :");
Serial.println(dataArray [0]);
Serial.println(dataArray [1]);
Serial.println(dataArray [2]);

Firestore.setString("Latitude3",String(dataArray [0]));
Firestore.setString("Longitude3",String(dataArray [1]));
Firestore.setString("Load3",String(dataArray [2]));
selesaiBaca=0;
digitalWrite(D0,0);
}

}

```

LAMPIRAN C

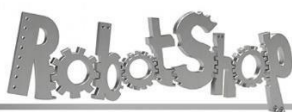


Overview

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 ([datasheet](#)). It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

Schematic & Reference Design

EAGLE files: [arduino-mega2560-reference-design.zip](#)



www.robotshop.com

La robotique à votre service! - Robotics at your service!



Schematic: [arduino-mega2560-schematic.pdf](#)

Summary

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

Power

The Arduino Mega can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The Mega2560 differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

BIODATA PENULIS



Nama : Bambang Wisnu
Wardhana Kusuma
TTL : Ngawi, 17 Mei 1998
Jenis Kelamin : Laki-Laki
Agama : Islam
Alamat Rumah : 02/4 Bandem, Kendal,
Ngawi, Jawa Timur
Nomor HP : 083857265709
E-mail : bambangwwk@gmail.com

RIWAYAT PENDIDIKAN

- 2004 - 2010 : MI Negeri 1 Ngawi
- 2010 - 2013 : MTs Negeri 13 Ngawi
- 2013 - 2016 : SMA Negeri 1 Magetan
- 2016 - sekarang : Bidang Studi Komputer Kontrol,
Departemen Teknik Elektro Otomasi
Fakultas Vokasi Institut Teknologi Sepuluh
Nopember

PENGALAMAN KERJA

- Magang Industri di PT Virama Karya Jawa Timur (2018)