



ITS
Institut
Teknologi
Sepuluh Nopember

PROYEK AKHIR - VE 180626

MONITORING PEMANTAUAN KUALITAS UDARA

Amalia Habibah
NRP 10311500010043

Dosen Pembimbing
Ir. Josaphat Pramudijanto, M.Eng.

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



FINAL PROJECT - VE 180626

MONITORING OF AIR QUALITY MONITORING

Amalia Habibah
NRP 10311500010043

Supervisor
Ir. Josaphat Pramudijanto, M.Eng.

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

PERNYATAAN KEASLIAN PROYEK AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Proyek Akhir saya dengan judul “**Monitoring Pemantauan Kualitas Udara**” adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak 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 peraturan yang berlaku.

Surabaya, Januari 2019

Amalia Habibah
NRP 10311500010043

MONITORING PEMANTAUAN KUALITAS UDARA

PROYEK AKHIR

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

Menyetujui :

Dosen Pembimbing



Ir. Josaphat Pramudjanto M.Eng
NIP. 196210051990031003

SURABAYA
MARET, 2019

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

MONITORING PEMANTAUAN KUALITAS UDARA

Nama Mahasiswa : Amalia Habibah
NRP : 10311500010043
Dosen Pembimbing : Ir. Josaphat Pramudijanto, M.Eng
NIP : 19621005 199003 1 003

ABSTRAK

Kualitas udara pada saat ini semakin penting untuk kehidupan mahluk hidup. Hal ini dapat disebabkan karena banyaknya pabrik yang tidak ramah lingkungan. Pabrik yang tidak ramah lingkungan akan menimbulkan gas berbahaya pada daerah sekitar pabrik. Gas tersebut dapat mengakibatkan daerah itu tercemar dan dapat merugikan orang – orang didaerah itu serta sekitarnya. Parameter gas yang dapat dijadikan sebagai tolak ukur pengukuran kualitas udara yaitu dari gas *Oksigen* (O_2), gas *Carbon Monoksida* (CO), dan gas *Carbondioksida* (CO_2).

Berdasarkan kondisi saat ini dengan adanya alat pendeteksi yang bersifat portable ini diharapkan dapat memonitor keadaan pemukiman warga yang berada di sekitaran pabrik. Metode yang digunakan yaitu dengan memonitor kadar gas CO yang terkandung diarea pemukiman warga sekitar pabrik. Mikrokontroler yang digunakan adalah Arduino Mega 2560 dan *Ethernet Shield* yang telah dilengkapi program serta telah dikonfigurasi dengan komponen perangkat keras seperti sensor MQ7 yang digunakan untuk mendeteksi gas CO. Interface alat ini menggunakan *LabVIEW* yang berfungsi untuk menampilkan dan memonitoring data yang telah terbaca oleh sensor gas dan temperatur via WIFI.

Hasil dari Proyek Akhir ini adalah dapat memonitoring kadar gas CO dimana telah dilakukan *kalibrasi* dan perbandingan pengukuran dengan presentase *error* 0-0,2%. Pengaplikasian alat yang telah dirangkai dapat digunakan untuk memonitoring kadar gas CO pada area pabrik secara *real time*.

Kata Kunci : Pencemaran udara, Arduino, Sensor Gas CO MQ-7

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

MONITORING OF AIR QUALITY MONITORING

Student Name : Amalia Habibah
Registration Number : 10311500010043
Supervisor : Ir. Josaphat Pramudijanto, M.Eng
ID : 19621005 199003 1 003

ABSTRACT

Air quality is now increasingly important for the life of living things. This can be caused by the many factories that are not environmentally friendly. Factories that are not environmentally friendly will cause dangerous gas in the area around the factory. The gas can cause the area to be polluted and can harm people in the area and its surroundings. Gas parameters that can be used as a measure of air quality measurement are from Oxygen gas (O₂), Carbon Monoxide (CO) gas, and Carbondioxide (CO₂) gas.

Based on the current conditions with the presence of a portable detector that is expected to be able to monitor the condition of residential areas that are around the factory. The method used is by monitoring CO gas levels contained in residential areas around the factory. The microcontroller used is Arduino Mega 2560 and Ethernet Shield that has been equipped with the program and has been configured with hardware components such as the MQ7 sensor used to detect CO gas. The interface of this tool uses LabVIEW which functions to display and monitor data that has been read by gas sensors and temperature via WIFI.

The results of this Final Project are able to monitor CO gas levels where calibration and comparison of measurements have been carried out with an error percentage of 0-0.2%. The application of assembled tools can be used to monitor CO gas levels in the factory area in real time.

keywords: *Telemetry, Air Quality.*

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

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT yang selalu memberikan rahmat dan hidayah-Nya sehingga Proyek 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 Program Studi Elektronika Industri, Departemen Teknik Elektro Otomasi, Fakultas Vokasi, Institut Teknologi Sepuluh Nopember Surabaya.

Pada kesempatan ini penulis menyampaikan terima kasih atas segala doa, bantuan dan dukungannya yang telah diberikan selama proses pembuatan Proyek Akhir ini kepada :

1. Kedua orang tua atas dukungan baik spiritual maupun material yang tak ternilai harganya.
2. Bapak Ir. Joko Susila, MT. Ketua departemen Teknik Elektro Otomasi, Fakultas Vokasi- ITS Surabaya.
3. Bapak Ir. Josaphat Pramudijanto, M Eng. Sebagai dosen pembimbing yang telah banyak memberikan ilmu, pengarahan dan bimbingan selama penulis mengerjakan Proyek Akhir ini, serta membimbing saya dengan kesabaran yang tiada batasnya.
4. Semua pihak yang telah membantu saya dalam menyelesaikan Proyek Akhir ini yang tidak dapat kami sebutkan satu persatu.

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

Surabaya, Januari 2019

Amalia Habibah

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

DAFTAR ISI

	HALAMAN
HALAMAN JUDUL.....	i
PERNYATAAN KEASLIAN PROYEK AKHIR	v
LEMBAR PENGESAHAN	vii
ABSTRAK	ix
<i>ABSTRACT</i>	xi
KATA PENGANTAR	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR	xvii
DAFTAR TABEL.....	xx
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Permasalahan	2
1.3 Batasan Masalah	2
1.4 Tujuan	2
1.5 Sistematika Laporan	2
1.6 Relevansi	3
BAB II TEORI DASAR.....	5
2.1 Pencemaran Udara di Indonesia.....	5
2.2 Potensio WireWound	6
2.3 Sensor Gas CO	7
2.4 Arduino Atmega 2560	8
2.5 <i>Ethernet Shield</i>	10
2.6 <i>LabVIEW</i>	11
BAB III PERANCANGAN <i>HARDWARE</i> DAN <i>SOFTWARE</i>.....	13
3.1 Diagram Sistem Kerja Alat	13
3.2 Perancangan Perangkat Keras Mekanik.....	14
3.3 Perancangan Perangkat Keras Elektrik	15
3.2.1. Perancangan <i>Power Supply</i>	15
3.2.2. Rangkaian Sensor MQ-7.....	16
3.2.3. Rangkaian Potensio <i>WireWound</i>	18
3.2.4. Rangkaian Relay <i>Exhaust Fan</i>	18
3.2.5. Perancangan Gas CO	19

3.4 Perancangan <i>Software</i>	20
BAB IV PENGUJIAN DAN ANALISA	25
4.1 Pengujian Input / Output Arduino Mega	25
4.2 Kalibrasi Sensor	27
4.3 Kalibrasi dengan Menggunakan <i>LabVIEW</i>	32
4.4 Pengujian Komunikasi <i>Ethernet</i>	33
4.5 Pengujian Jarak.....	35
BAB V PENUTUP	47
5.1 Kesimpulan.....	47
5.2 Saran.....	47
DAFTAR PUSTAKA.....	49
LAMPIRAN A	A-1
LAMPIRAN B.....	B-1
LAMPIRAN C	C-1
DAFTAR RIWAYAT HIDUP	C-2

DAFTAR GAMBAR

HALAMAN

Gambar 2.1 Potensio Wirewound	7
Gambar 2.2 Struktur Sensor MQ-7	8
Gambar 2.3 Sensor MQ-7	8
Gambar 2.4 <i>Board</i> Arduino Atmega 2560	10
Gambar 2.5 <i>Ethernet Shield</i>	11
Gambar 2.6 Tampilan pada <i>LabVIEW</i>	12
Gambar 3.1 Diagram Fungsional	14
Gambar 3.2 Perancangan Alat Tampak Depan dan Samping	14
Gambar 3.3 Perancangan Alat Tampak Atas	15
Gambar 3.4 Perancangan Kotak Sensor	15
Gambar 3.5 Perhitungan Pemakaian Aki	16
Gambar 3.6 Rangkaian Skematik Sensor CO MQ-7	17
Gambar 3.7 Rangkaian Sensor MQ-7	17
Gambar 3.8 Rangkaian Potensio <i>Wirewound</i>	18
Gambar 3.9 Rangkaian Relay <i>Exhaust Fan</i>	19
Gambar 3.10 Rangkaian LCD	19
Gambar 3.11 <i>Setting port</i> pada <i>software</i> arduino IDE	21
Gambar 3.12 <i>Flowchart</i> Pada Rangkaian Arduino	22
Gambar 3.13 <i>Block Diagram</i> Pada <i>LabVIEW</i>	24
Gambar 4.1 Program Pengujian <i>I/O</i> Arduino Mega 2560 saat <i>Active High</i>	26
Gambar 4.2 Skema Pengujian <i>I/O</i> Arduino Mega 2560	26
Gambar 4.3 Program Kalibrasi Sensor CO	28
Gambar 4.4 Hasil Kalibrasi Sensor CO (MQ-7)	29
Gambar 4.5 Rangkaian Potensio dengan Arduino	29
Gambar 4.6 Program Arduino pada potensio	30
Gambar 4.7 Rangkaian Potensio dengan Arduino	31
Gambar 4.8 Program Arduino pada potensio	31
Gambar 4.9 Program Pada Arduino Mega	34
Gambar 4.10 <i>Command Prompt</i> Pada Laptop Dengan <i>IP address</i> Arduino Mega	34
Gambar 4.11 Tampilan Pengujian <i>Interface</i> pada <i>LabVIEW</i>	35
Gambar 4.12 Ilustrasi Pengujian Koneksi Alat Dengan Menggunakan Penghalang	36

Gambar 4.13	<i>Command Prompt</i> Pada Laptop Dengan IP Address Arduino Mega 2560 Pada Jarak 5 Meter Dengan Penghalang	37
Gambar 4.14	<i>Command Prompt</i> Pada Laptop Dengan IP Address Arduino Mega 2560 Pada Jarak 10 Meter Dengan Penghalang	37
Gambar 4.15	<i>Command Prompt</i> Pada Laptop Dengan IP Address Arduino Mega 2560 Pada Jarak 20 Meter Dengan Penghalang	38
Gambar 4.16	Ilustrasi Pengujian Koneksi Alat Dengan Tanpa Menggunakan Penghalang.....	39
Gambar 4.17	<i>Command Prompt</i> Pada Laptop Dengan IP Address Arduino Mega 2560 Pada Jarak 1,5 Meter Tanpa Penghalang	39
Gambar 4.18	<i>Command Prompt</i> Pada Laptop Dengan IP Address Arduino Mega 2560 Pada Jarak 8 Meter Tanpa Penghalang	40
Gambar 4.19	<i>Command Prompt</i> Pada Laptop Dengan IP Address Arduino Mega 2560 Pada Jarak 13 Meter Tanpa Penghalang	41
Gambar 4.20	<i>Command Prompt</i> Pada Laptop Dengan IP Address Arduino Mega 2560 Pada Jarak 50 Meter Tanpa Penghalang	41
Gambar 4.21	Lokasi Alat Pada Departemen Teknik Elektro Otomasi ..	43
Gambar 4.22	Tampilan <i>LabVIEW</i> Hasil Telemetering pada Parkiran Departemen Teknik Elektro Otomasi	44
Gambar 4.23	Tampilan <i>LabVIEW</i> Hasil Telemetering pada Lampu Merah Kertajaya	45

DAFTAR TABEL

	HALAMAN
Tabel 4.1 Hasil Pengukuran Per <i>Pin</i> Saat <i>Active High</i>	26
Tabel 4.2 Hasil Pengukuran Per <i>Pin</i> Saat <i>Active Low</i>	27
Tabel 4.3 Hasil Kalibrasi Sensor CO	28
Tabel 4.4 Hasil Kalibrasi Potensio.....	30
Tabel 4.5 Hasil Kalibrasi Potensio.....	31
Tabel 4.6 Kalibrasi Potensio1 Menggunakan <i>LabView</i>	32
Tabel 4.7 Hasil Kalibrasi Potensio 2 Pada <i>Software LabVIEW</i>	32
Tabel 4.8 Hasil Kalibrasi Sensor CO Pada <i>Software LabVIEW</i>	33
Tabel 4.9 Hasil Pengujian <i>Acces Point</i> Dengan Penghalang	38
Tabel 4.10 Hasil Pengujian <i>Acces Point WiFi</i> dengan Tanpa Penghalang	42
Tabel 4.11 Hasil Pengujian Peletakan Alat Dengan Diberi Jarak.....	42
Tabel 4.12 Data Pengukuran Gas Pada Parkiran Departemen Teknik Elektro Otomasi	43

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

BAB I

PENDAHULUAN

1.1 Latar Belakang

Setiap makhluk hidup membutuhkan udara untuk kelangsungan hidupnya. Karena itu udara memiliki peran penting. Sehingga kita harus memperhatikan kualitas udara yang kita hirup. Jika udara yang kita hirup mengandung gas – gas berbahaya yang melebihi ambang batas, maka efek yang ditimbulkan akan sangat berbahaya, yaitu dapat mengganggu metabolisme kerja dari organ tubuh kita. Efek terburuknya yaitu menyebabkan kematian. Ada beberapa gas yang berpengaruh dalam kehidupan makhluk hidup, antara lain gas O₂, gas CO₂ dan gas CO. Gas O₂ merupakan gas yang sangat diperlukan bagi kehidupan manusia yang kita ketahui sebagai gas yang terdapat dalam sistem peredaran darah. Sementara gas CO₂ merupakan sebuah gas yang dihasilkan dari sebuah proses seperti misalnya proses pembakaran. Proses pembusukan yang terjadi pada bahan-bahan tertentu atau bahkan sampah sehari-hari juga dapat menghasilkan gas karbon dioksida. Yang terakhir adalah gas CO, gas ini juga merupakan gas yang berbahaya jika dihirup oleh seseorang. Gas CO dapat berikatan dengan hemoglobin dalam tubuh sehingga pengikatan oksigen dalam tubuh akan terganggu. Pada gas karbon monoksida dan gas karbon dioksida termasuk gas yang dapat membahayakan kelangsungan hidup makhluk hidup. Jika gas tersebut melebihi ambang batas juga bisa dapat menyebabkan kematian.

Agar dapat mengetahui kualitas udara tersebut layak tidaknya maka diperlukan alat yang dapat mengontrol kondisi udara. Oleh karena itu, pada Proyek Akhir ini mengambil judul “Monitoring Pemantauan Kualitas Udara”. Yakni pembuatan sebuah monitoring kualitas udara menggunakan sensor gas O₂, gas CO₂ dan gas CO dengan melakukan perbandingan nilai agar dapat mengetahui kelayakan kualitas udara tersebut. Perbandingan nilai tersebut nantinya sebagai data persentase kualitas udara. Data tersebut dikirim oleh Arduino menggunakan komunikasi Ethernet yang akan ditampilkan pada *software LabVIEW*. Hal ini selain mempermudah petugas jaga, juga lebih efisien untuk mengetahui kualitas udara dengan jarak jauh melalui *personal computer* pada *software LabVIEW* dengan menggunakan komunikasi Ethernet.

1.2 Permasalahan

Pada Proyek Akhir ini yang menjadi permasalahan utama adalah belum banyaknya mekanisme yang memonitoring secara jarak jauh kualitas udara terutama untuk gas CO.

1.3 Batasan Masalah

Dari perumusan masalah di atas, maka batasan masalah dari Proyek Akhir ini adalah :

1. Sensor yang digunakan sebagai pembaca kualitas udara adalah sensor CO.
2. Pengukuran yang dilakukan hanya terbatas pada pengukuran gas CO.
3. Pembuatan monitoring gas CO menggunakan *Ethernet Shield* dan Arduino
4. Mengimplementasikan pemantauan kualitas udara dengan software LabVIEW.

Dengan adanya batasan masalah ini diharapkan hasil akhir atau tujuan dari Proyek Akhir ini dapat dicapai dengan baik.

1.4 Tujuan

Pembuatan Monitoring Pemantauan Kualitas Udara bertujuan untuk:

1. Merancang dan membuat perangkat lunak untuk monitoring pemantauan kualitas udara.

1.5 Sistematika Laporan

Pembahasan Proyek Akhir ini akan dibagi menjadi lima Bab dengan sistematika sebagai berikut:

Bab I Pendahuluan

Bab ini meliputi latar belakang, permasalahan, batasan masalah, tujuan, metodologi penelitian, sistematika laporan, dan relevansi.

Bab II Teori Dasar

Bab ini menjelaskan tentang tinjauan pustaka, konsep dari pendistribusian air secara umum, Arduino Atmega 2560, *Ethernet Shield* , dan *LabVIEW*.

Bab III Perancangan Sistem

Bab ini membahas perancangan sistem *hardware* maupun *software* pada monitoring pemantauan kualitas udara. Air Dengan Komunikasi *Ethernet* berdasarkan teori penunjang pada Bab II.

Bab IV Pengujian dan Analisa Sistem

Bab ini membahas tentang pengukuran, pengujian, serta analisa terhadap prinsip kerja dan proses dari suatu alat yang dibuat..

Bab V Penutup

Bab ini menjelaskan tentang kesimpulan dari Proyek Akhir dan saran – saran untuk pengembangan alat ini lebih lanjut

1.6 Relevansi

Hasil yang diperoleh dari Proyek Akhir ini diharapkan dengan pengembangan lebih lanjut dapat dijadikan sebagai referensi atau ide dalam kemajuan sistem telemetering kualitas udara. Dengan efisiensi monitoring kualitas udara melalui jarak jauh ini semoga dapat membantu mengetahui kualitas udara tersebut.

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

BAB II

TEORI DASAR

Pada bab ini akan membahas teori dasar dan teori penunjang terkait perangkat dan bahan yang digunakan dalam Proyek Akhir ini.

2.1 Pencemaran Udara di Indonesia [1]

Indonesia adalah negara yang terdapat lebih dari 26,71 juta industri yang beroperasi mulai dari perusahaan industri besar sedang hingga perusahaan industri Mikro. Menurut Badan Pusat Statistik (BPS) terjadi peningkatan jumlah industri selama 10% selama 10 tahun. Maka hal ini akan menyebabkan semakin banyaknya alat-alat industri yang mengeluarkan gas yang dapat mencemarkan lingkungan dan akan semakin parah pula pencemaran udara yang terjadi. Banyaknya perusahaan industri tentunya menimbulkan dampak positif dan negatif di kalangan masyarakat sekitar. Dampak positif dari banyaknya industri ini, tentunya akan meningkatkan kesejahteraan masyarakat dan akan mengurangi angka pengangguran di Indonesia, yang selama ini juga menjadi masalah besar bagi Indonesia. Akan tetapi, Indonesia juga harus memperhatikan dan memikirkan dampak negatif dari banyaknya industri ini, yang akan merugikan masyarakat dan lingkungan yaitu semakin meningkatnya pencemaran udara.

Sumber pencemar udara dapat diklasifikasikan menjadi sumber diam dan sumber bergerak. Sumber diam terdiri dari pembangkit listrik, industri dan rumah tangga. Sedangkan sumber bergerak adalah aktifitas lalu lintas kendaraan bermotor dan transportasi laut. Dari data BPS tahun 1999, di beberapa propinsi terutama di kota-kota besar seperti Medan, Surabaya dan Jakarta, emisi kendaraan bermotor merupakan kontribusi terbesar terhadap konsentrasi NO₂ dan CO di udara yang jumlahnya lebih dari 50%. Penurunan kualitas udara yang terus terjadi selama beberapa tahun terakhir menunjukkan kita bahwa betapa pentingnya digalakkan usaha-usaha pengurangan emisi ini. Baik melalui penyuluhan kepada masyarakat ataupun dengan mengadakan penelitian bagi penerapan teknologi pengurangan emisi. Secara umum, terdapat 2 sumber pencemaran udara, yaitu pencemaran akibat sumber alamiah, seperti letusan gunung berapi, dan yang berasal dari kegiatan manusia, seperti yang berasal dari transportasi, emisi pabrik, dan lain-lain. Di

dunia, dikenal 6 jenis zat pencemar udara utama yang berasal dari kegiatan manusia, yaitu Karbon monoksida (CO), oksida sulfur (SO_x), oksida nitrogen (NO_x), partikulat, hidrokarbon (HC), dan oksida fotokimia, termasuk ozon .

Apalagi kondisi udara di Indonesia berada pada kondisi *under pollution*. Hasil penelitian dari Internasional Energy Agency(IEA) tahun 2016 juga mengungkapkan bahwa lebih dari 60 ribu orang di Indonesia meninggal akibat pencemaran udara. Bahkan kota Jakarta juga disebutkan sebagai kota yang memiliki kualitas udara terburuk di Asia Tenggara. Oleh sebab itu, pemerintah wajib turun tangan dalam menangani masalah ini. Pemerintah juga perlu mengkaji khusus tentang masalah ini karena sangat berpengaruh pada masyarakat Indonesia.

2.2 Potensio WireWound [2]

Potensiometer *WireWound* adalah potensiometer yang menggunakan kawat halus yang dililit pada batang metal. Ketelitian potensiometer tergantung dari ukuran kawat. Kawat yang digunakan biasanya adalah kawat nikelin. Potensio *WireWound* Digunakan pada perangkat yang membutuhkan akurasi pengontrolan yang tinggi

Pada potensiometer ini memiliki perubahan tahanannya sangat halus dengan jumlah putaran sampai sepuluh kali putaran (multi turn). Untuk keperluan sensor posisi potensiometer linier memanfaatkan perubahan resistansi, diperlukan proteksi apabila jangkauan ukurnya melebihi rating, linearitas yang tinggi hasilnya mudah dibaca tetapi hati-hati dengan friksi dan backlash yang ditimbulkan, resolusinya terbatas yaitu 0,2 – 0,5%.

Potensio wirewound juga memiliki *power rating* (@ 70°C) = 2,0 Watts artinya jumlah daya yang dapat disipasikan pada suhu 70°C tanpa menyebabkan kerusakan pada potensiometer *WireWound*. *Noise* 100Ω ENR artinya besarnya kelebihan *noise* ratio dari output potensiometer yang dapat digunakan untuk masukannya. Pada Gambar 2.1 merupakan bentuk dari potensio wirewound.



Gambar 2.1 Potensio Wirewound

2.3 Sensor Gas CO [3]

Gas CO adalah gas yang tak berwarna, tak berbau dan bersifat sangat beracun. Gas CO juga bisa ditemukan pada emisi gas akibat pembakaran yang tidak sempurna. Gas CO juga tak berbau dan tak berwarna, sehingga orang dapat tertidur tanpa curiga bahwa ia sedang mulai keracunan. Kebanyakan gas CO dapat mengakibatkan napas pendek, sakit kepala, mudah lelah, mudah tersinggung, kejang dan koma, yang dapat berakhir dengan kematian apabila konsentrasi gas ini masuk ke dalam tubuh dalam jumlah yang sangat tinggi dan dapat menyebabkan kecatatan pada bayi terutama pada bagian otak.

MQ 7 merupakan sensor gas yang digunakan dalam peralatan untuk mendeteksi gas karbon monoksida (CO) dalam kehidupan sehari-hari, industri, atau mobil. Fitur dari sensor gas MQ7 ini adalah mempunyai sensitivitas yang tinggi terhadap karbon monoksida (CO), stabil, dan berumur panjang. Sensor ini menggunakan catu daya heater : 5 Volt AC/DC dan menggunakan catu daya rangkaian : 5VDC, jarak pengukuran : 20 - 2000ppm untuk ampu mengukur gas karbon monoksida.

Spesifikasi:

- VC/(Tegangan Rangkaian) = $5V \pm 0,1$
- VH (H)/ Tegangan Pemanas (Tinggi) = $5V \pm 0,1$
- VH (L)/ Tegangan Pemanas (Rendah) = $1,4V \pm 0,1$
- RL/Resistansi Beban Dapat disesuaikan
- TH (H) Waktu Pemanasan (Tinggi) = 60 ± 1 seconds
- TH (L) Waktu Pemanasan (Rendah) = 90 ± 1 seconds
- Suhu Penggunaan = $-20^{\circ}\text{C} - 50^{\circ}\text{C}$
- Suhu Penyimpanan = $-20^{\circ}\text{C} - 50^{\circ}\text{C}$
- Kelembapan Relatif = kurang dari 95% RH

- O₂ Konsentrasi Oksigen = 21% (*stand condition* / Konsentrasi Oksigen dapat mempengaruhi sensitivitas).

Sensor MQ-7 dan susunan masing-masing kakinya dapat dilihat pada Gambar 2.2. Sedangkan pada Gambar 2.3 juga akan dijelaskan bentuk fisik dari sensor CO (MQ-7).

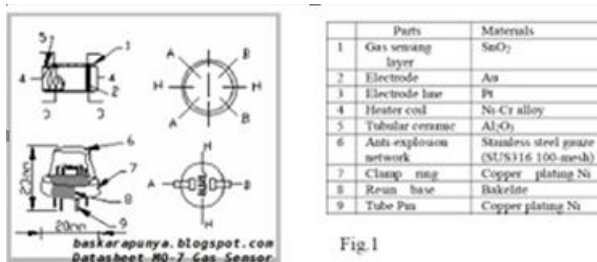


Fig.1

Gambar 2.2 Struktur Sensor MQ-7



Gambar 2.3 Sensor MQ-7

2.4 Arduino Atmega 2560 [4]

Arduino merupakan papan rangkaian elektronik atau kit elektronik yang bersifat *open-source* yang didalamnya terdapat komponen utama yaitu sebuah mikrokontroler dengan jenis *Atmel AVR* dan menggunakan software sendiri untuk pemrograman.

Mikrokontroler sendiri adalah sebuah komputer kecil disuatu sikuit terpadu yang berisi tentang inti prosesor, memori dan input/output yang telah diprogram. Program disimpan dalam bentuk *Ferroelectric RAM*, *Nor Flash*, *OTP ROM* yang disertakan dalam

chip. Mikrokontroler digunakan untuk aplikasi *embedded*, tidak seperti mikroprosesor yang digunakan dalam komputer pribadi. Fungsi dari mikrokontroler adalah untuk mengontrol produk atau perangkat secara otomatis seperti sistem kontrol mesin mobil, mesin kantor, alat-alat listrik, dan sistem *embedded* lainnya.

Arduino Mega 2560 merupakan papan microcontroller berbasis Atmega 2560. Arduino Mega 2560 memiliki 54 pin digital input / output, dimana 15 pin dapat digunakan sebagai output PWM, 16 pin sebagai input analog, dan 4 pin sebagai UART (port serial hardware), 16 MHz kristal osilator, koneksi USB, jack power, header ICSP, dan tombol reset. Ini semua yang diperlukan untuk mendukung microcontroller. Cukup dengan menghubungkannya ke komputer melalui kabel USB atau power dihubungkan dengan adaptor AC – DC atau baterai untuk mulai mengaktifkannya. Arduino Mega 2560 kompatibel dengan sebagian besar shield yang dirancang untuk Arduino Duemilanove atau Arduino Diecimila. Arduino Mega 2560 merupakan versi terbaru yang menggantikan versi Arduino Mega. Spesifikasinya adalah seperti berikut:

- Memiliki kecepatan *clock* mencapai 16MHz.
- Memiliki *Flash Memory* 256 KB, 8 KB diantaranya sebagai *bootloader*.
- Rentang *input* sumber tegangan 7-20 Volt, dan beroperasi pada tegangan 5 Volt.
- Memiliki EEPROM (*Electrically Erasable Programmable Read Only Memory*) sebesar 4KB.
- Memiliki SRAM (*Static Random Access Memory*) sebesar 8KB.
- Memiliki pin I/O digital sebanyak 54 digital Input / Output, 15 buah di antaranya dapat di gunakan sebagai output PWM.
- Arus DC per pin I/O 20 mA dan pada pada pin 3,3 Volt 50 mA.

Letak pin pada *board* Arduino Atmega 2560 dapat dilihat pada Gambar 2.4.

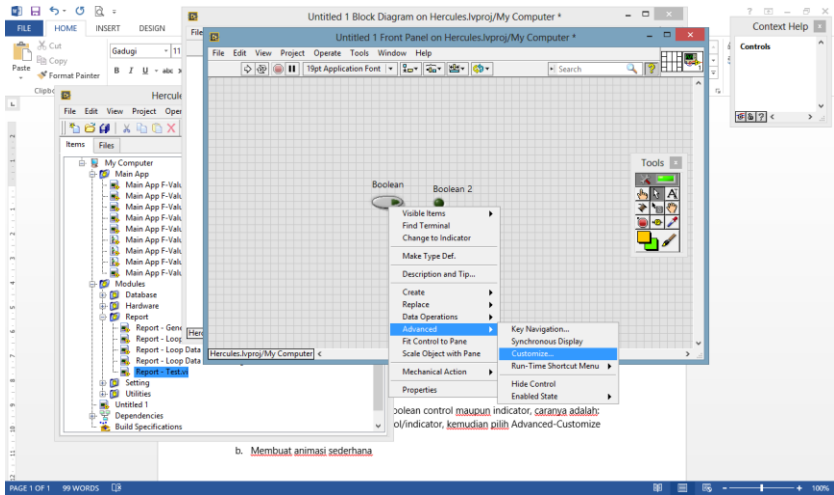


Gambar 2.5 *Ethernet Shield*

2.6 *LabVIEW* [6]

LabVIEW adalah sebuah software pemrograman yang diproduksi oleh *National instruments* dengan konsep yang berbeda. Seperti bahasa pemrograman lainnya yaitu C++, matlab atau *Visual basic*, *LabVIEW* juga mempunyai fungsi dan peranan yang sama, perbedaannya bahwa *LabVIEW* menggunakan Bahasa pemrograman berbasis *grafis* atau blok diagram sementara bahasa pemrograman lainnya menggunakan basis *text*. Program *LabVIEW* dikenal dengan sebutan Vi atau *Virtual instruments* karena penampilan dan operasinya dapat meniru sebuah *instrument*. Pada *LabVIEW*, user pertama-tama membuat *user interface* atau *front panel* dengan menggunakan *control* dan indikator, yang dimaksud dengan *control* adalah *knobs*, *push buttons*, *dials* dan peralatan input lainnya sedangkan yang dimaksud dengan indikator adalah *graphs*, LED dan peralatan display lainnya.

Setelah menyusun *user interface*, lalu user menyusun blok diagram yang berisi kode-kode VIs untuk mengontrol *front panel*. *Software LabVIEW* terdiri dari tiga komponen utama, yaitu : *front panel*, Blok diagram dari Vi, *Control* dan *Functions Pallette*. Gambar 2.6 merupakan tampilan dari *Software LabVIEW*



Gambar 2.6 Tampilan pada *LabVIEW*

BAB III

PERANCANGAN *HARDWARE* DAN *SOFTWARE*

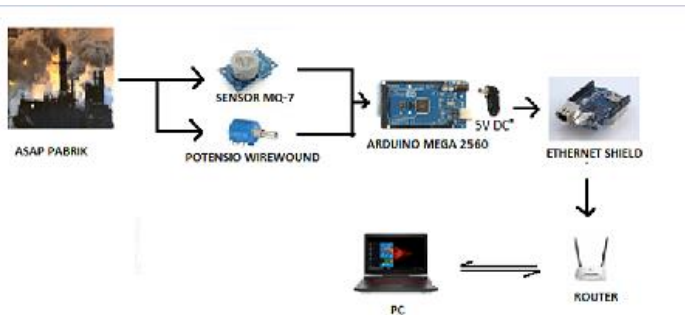
Pada bab ini akan membahas mengenai perancangan dan pembuatan “*Monitoring* pemantauan kualitas udara”, yang terdiri dari beberapa perancangan yaitu:

1. Perancangan Sistem Kerja Alat
 2. Perancangan Perangkat Keras Mekanik
 3. Perancangan Perangkat Keras Elektrik
- Perancangan Perangkat Lunak

3.1 Diagram Sistem Kerja Alat

Pada perancangan sistem kerja alat, *prototype* alat pendeteksi *monitoring* pemantauan kualitas udara akan berisi *power supply*, sensor gas, Arduino Mega, *Ethernet Shield*, potensio *Wire Wound*, dan juga *aces point*. Pada alat ini, pendeteksian difokuskan pada gas *Carbon Monosida* dengan menggunakan Sensor MQ-7 dan potensio *Wire Wound* digunakan sebagai indikator dari gas CO₂ serta O₂. Alat ini akan dimonitor pada satu PC menggunakan *interface* LabVIEW.

Input sistem ini adalah gas polutan pabrik. *Array sensor* akan mendeteksi apakah gas tersebut mengandung polutan seperti CO, O₂, dan CO₂. Untuk gas O₂ dan CO₂ menggunakan pengganti sensor menggunakan indikator potensio *Wire Wound*. Hasil yang didapatkan *array* sensor akan dikirimkan ke Arduino Mega menggunakan komunikasi TTL untuk sensor CO, dan potensio untuk kemudian diproses oleh Arduino Mega. Data hasil pembacaan sensor akan ditampilkan juga di *Liquid crystal Display*. Data hasil pembacaan sensor yang telah diproses akan dikirimkan melalui komunikasi data melalui *wireless*. *Wireless (Transmitter)* akan mengirimkan informasi ini ke komputer. Di komputer, informasi tersebut akan di sajikan dan disimpan dengan aplikasi yang telah dibuat menggunakan LabVIEW. Untuk cara kerja sistem alat ditunjukkan pada Gambar 3.1. Perancangan sistem secara keseluruhan pembuatan alat yang digunakan dapat diperlihatkan pada Gambar 3.1.

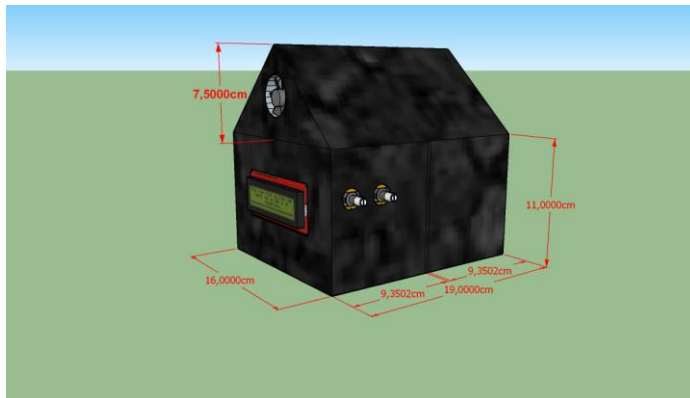


Gambar 3.1 Diagram Fungsional

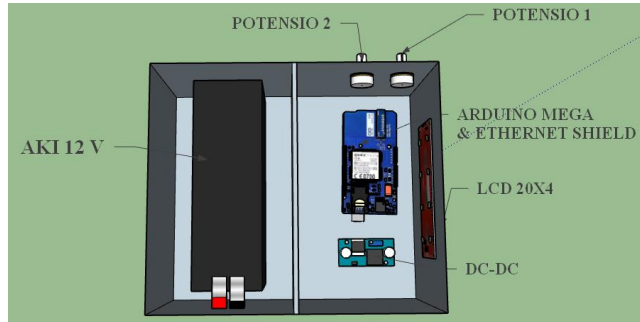
3.2 Perancangan Perangkat Keras Mekanik

Pada perancangan perangkat keras mekanik ini terbagi menjadi 2 bagian, yaitu perancangan pada bagian sistem dan perancangan dari panel kontrol.

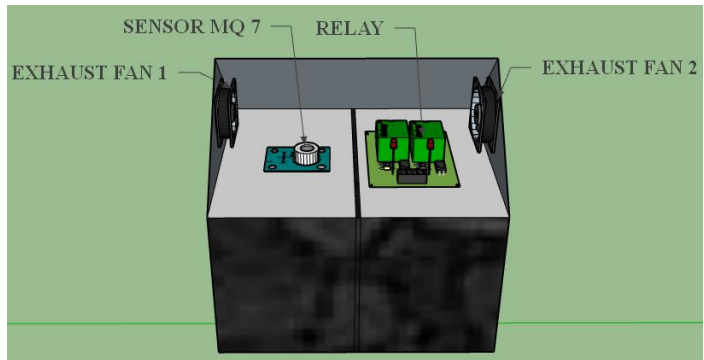
Pada perancangan sistem merupakan bagian dari komponen penyusun dari Proyek Akhir ini, antara lain Arduino Mega, LCD, *Power Supply*, Ethernet Shield serta potensiometer. Seluruh komponen tersebut diletakkan pada akrilik seperti pada Gambar 3.2 untuk tampak depan dan Gambar 3.3 untuk tampak atas. Sedangkan rancangan kotak sensor terdiri dari beberapa komponen yaitu *Exhaust Fan*, Relay dan sensor CO (MQ-7) yang dapat dilihat pada Gambar 3.4



Gambar 3.2 Perancangan Alat Tampak Depan dan Samping



Gambar 3.3 Perancangan Alat Tampak Atas



Gambar 3.4 Perancangan Kotak Sensor

3.3 Perancangan Perangkat Keras Elektrik

Pada perancangan perangkat elektrik (*hardware*) pada Proyek Akhir ini meliputi perancangan rangkaian power supply, rangkaian sensor gas CO, perancangan rangkaian potensio, perancangan rangkaian lcd, dan perancangan rangkaian exhaust.

3.2.1. Perancangan *Power Supply*

Rangkaian *power supply* berfungsi untuk supply arus dan tegangan ke seluruh rangkaian yang ada. Rangkaian *power supply* ini terdiri dari dua keluaran, yaitu 5 Volt, dan 8 Volt dengan sumber dari aki 12 Volt 7,2 Ah. Keluaran 8 Volt digunakan untuk menghidupkan arduino sedangkan 5 Volt digunakan untuk menghidupkan LCD, sensor gas, sensor temperatur, module relay dan exhaust fan.

Regulator tegangan 8 Volt (LM7808CT) digunakan agar keluaran yang dihasilkan tetap 8 Volt walaupun terjadi perubahan pada tegangan masukannya. Regulator tegangan 5 Volt (LM7805CT) digunakan agar keluaran yang dihasilkan tetap 5 Volt walaupun terjadi perubahan pada tegangan masukannya.

Pemakaian daya dari masing-masing komponen yaitu Arduino mega sebesar 0,1 Watt, *Ethernet Shield* 0,095 Watt, sensor MQ-7 0,028 Watt, Thermocouple 0,008 Watt, 2 buah Exhaust Fan 1 Watt, modul relay 0,72 Watt dengan total pemakaian daya sebesar 2,051 Watt. Perhitungan menggunakan rumus $P = V \times I$ dan penjumlahan total daya menggunakan rumus $P = P_1 + P_2 + P_n$. Dari daya total yang didapat maka kita bisa menghitung lama waktu dari pemakaian aki dengan perhitungan. Perhitungan lama pemakaian aki dapat dilihat pada Gambar 3.5.

$$I = P / V \quad \text{keterangan:}$$

$$I = 2,051 \text{ Watt} / 12 \text{ Volt} \quad I = \text{ arus (A)}$$

$$I = 0,171 \text{ A} \quad P = \text{Daya (Watt)}$$

$$V = \text{Tegangan (Volt)}$$

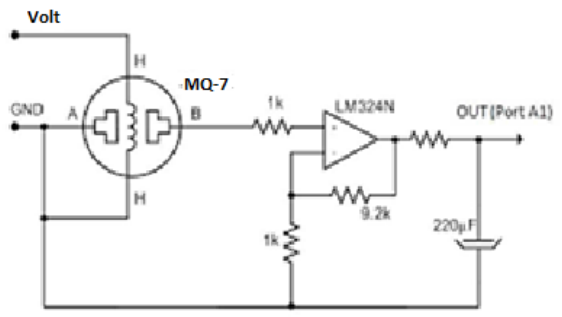
$$\begin{aligned} \text{Waktu pemakaian} &= I_{aki} / 0,171 = \dots \text{ jam} \\ \text{Waktu pemakaian} &= 7,2 / 0,171 = 42,11 \text{ jam} \\ \text{Waktu pemakaian} &= 42,11 \text{ jam} - \text{dieffisiensi aki } 20\% \\ &= 42,11 \text{ jam} - 8,422 \text{ jam} \\ &= 33,688 \text{ jam} \end{aligned}$$

Gambar 3.5 Perhitungan Pemakaian Aki

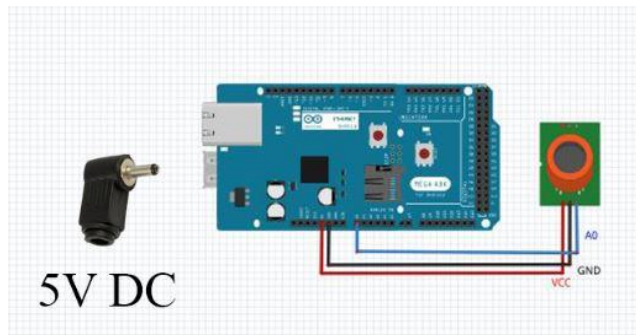
Dapat disimpulkan waktu pemakaian aki sekitar 33,68 jam atau selama 1 hari lebih 9 jam 6 menit.

3.2.2. Rangkaian Sensor MQ-7

Pada rangkaian sensor CO ini yaitu membuat *driver* sensor ter-lebih dahulu. Pada sensor CO sebelum mendapatkan tegangan sebesar 5VDC perlu adanya suatu beban RL. Beban RL yang digunakan yaitu resistor 2K ohm. Untuk rangkaian perancangan sensor CO bisa dilihat pada Gambar 3.6



Gambar 3.6 Rangkaian Skematik Sensor CO MQ-7

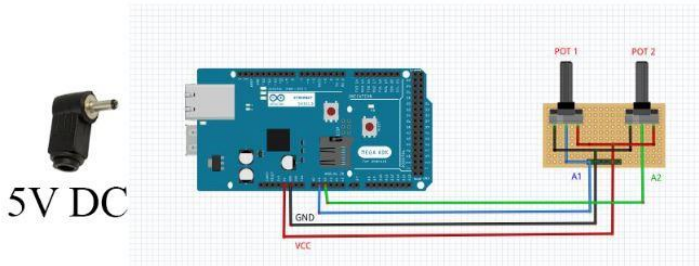


Gambar 3.7 Rangkaian Sensor MQ-7

Pada Gambar 3.7 merupakan perancangan sensor CO pada arduino mega ini terlihat pin Vin dan Vout. Yang mana Vin sebagai *input* sensor yang akan disambungkan ke *power supply*. Ada 2 pin Vin yaitu Vin 1 disambungkan ke *ground power supply* sedangkan VI-2 disambungkan ke *vcc power supply*. Tegangan yang dibutuhkan sensor co dari *power supply* sebesar 5VDC. Pin Vout sendiri sebagai *output* sensor co yang akan disambungkan ke Arduino Mega. pin Vout juga ada 2, yaitu Vout-1 disambungkan ke A1 arduino dan Vout-2 disambungkan ke *ground* arduino.

3.2.3. Rangkaian Potensio *WireWound*

Pada alat ini rangkaian potensio *Wirewound* digunakan sebagai pengganti dari sensor gas CO2 dan gas O2. Potensio *wirewound* memiliki 10 putaran dan memiliki keakuratan data dibandingkan potensio pada umumnya.

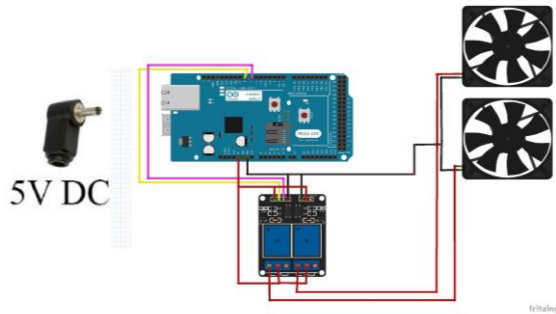


Gambar 3.8 Rangkaian Potensio *Wirewound*

Pada Gambar 3.8 dapat dilihat bahwa Konfigurasi pin potensio *wirewound* dengan *Arduino Mega* yaitu pin 5V arduino dihubungkan dengan pin VCC *wirewound*, pin GND Arduino di hubungkan dengan pin GND *wirewound*, dan pin Vout *wirewound* di hubungkan dengan pin A1 dan A2 analog Arduino.

3.2.4. Rangkaian Relay *Exhaust Fan*

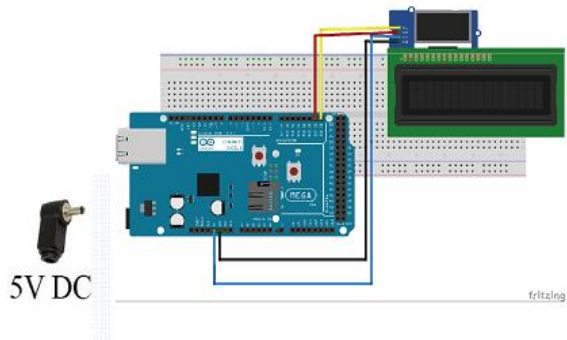
Rangkaian *Exhaust Fan* yang dikoneksikan dengan modul relay dua channel skema konfigurasinya yaitu pin 5V Arduino di hubungkan dengan COM relay 1 dan COM relay 2, pin out NO relay 1 dan relay 2 dihubungkan dengan VCC kedua *exhaust fan*, pin GND Arduino di hubungkan dengan GND Relay dan pin 7 digital Arduino di hubungkan dengan IN1 relay serta pin 8 digital Arduino di hubungkan dengan pin IN2 relay. Dapat dilihat pada Gambar 3.9



Gambar 3.9 Rangkaian Relay *Exhaust Fan*

3.2.5. Perancangan LCD

Pada Proyek Akhir ini, LCD berfungsi sebagai penampil data gas beracun yang telah dideteksi oleh sensor. LCD yang digunakan yaitu LCD berukuran 20x4. LCD dapat menampilkan data jika disambungkan dengan *controller*. *Controller* yang digunakan pada Proyek Akhir ini yaitu Arduino Mega. Untuk menyambungkan LCD dengan arduino, yaitu bisa dengan modul I2C *backpack*. Dapat dilihat pada Gambar 3.10.



Gambar 3.10 Rangkaian LCD

3.4 Perancangan *Software*

Pada perancangan perangkat *software* pada Proyek Akhir ini yang dibahas terdiri dari pemrograman pengukuran debit air sensor *flowmeter* pada software Arduino IDE, dan perancangan *software LabVIEW*.

3.4.1 Pemrograman Pembacaan Sensor

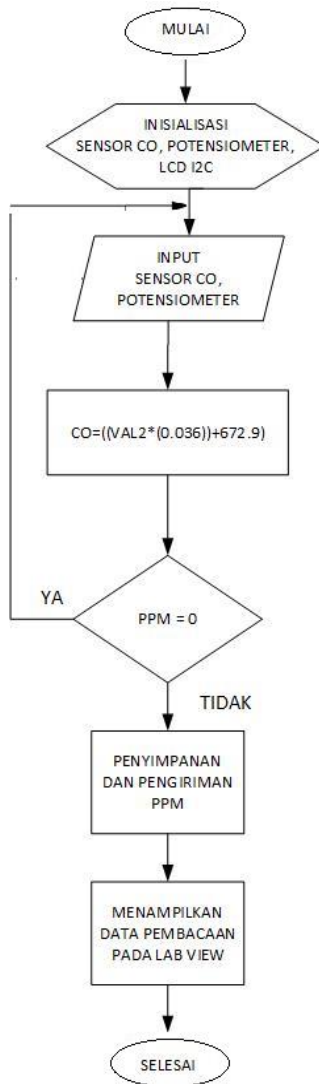
Pada perancangan program software arduino dengan fungsi tersebut dibutuhkan beberapa alur kerja dari pembacaan sensor sensor CO: Untuk memprogram arduino juga harus dilakukan beberapa tahapan sebagai berikut :

1. **Setting Board Arduino.** Dalam pemrograman *software* arduino harus di *setting* terlebih dahulu *board* arduino agar penggunaan arduino cocok. Dalam purwarupa kali ini arduino menggunakan arduino Mega. Untuk *setting board* arduino bisa masuk ke *tools – board* – setelah itu pilihlah *board* arduino yang sesuai.
2. **Setting Serial.** *Serial* ini merupakan kabel arduino yang dihubungkan kepada komputer atau laptop. *Serial* ini mempunyai dua fungsi yang bisa digunakan. Pertama *serial port* digunakan untuk mengunduh program dari arduino yang kedua *serial* digunakan sebagai komunikasi *serial* pada arduino dengan komputer. *Setting serial* bisa masuk *tools – serial* - lalu pilih COM yang sesuai dengan arduino yang terpasang dan dapat dilihat pada Gambar 3.11
3. Apabila *program* tidak dapat diunduh karena *serial port*, maka cek terlebih dahulu *serial* yang benar pada *device manager*. Lalu dalam *software* arduino untuk memilih *serial port*nya samakan dengan *serial port* untuk arduino dalam *device manager* tersebut. Untuk masuk ke *device manager* dapat masuk *start windows* – lalu ketika *device manager* klik dua kali dan masuk ke COM.



Gambar 3.11 Setting port pada software arduino IDE

Pada Proyek Akhir ini *software* yang digunakan adalah program Arduino Mega 2560 untuk membuat dan merencanakan program dalam bahasa C. Pemrograman *software* arduino dirancang dengan menggunakan *software* yang bernama arduino IDE dengan menggunakan bahasa pemrograman C. Arduino sangatlah berbeda sekali dengan ATmega lainnya. Arduino merupakan sebuah kit mikro-kontroler AVR yang dibuat dalam sebuah *board* (papan PCB). Dikem-bangkan di Italia sejak tahun 2005. Dalam 1 *board* sudah terdapat mikrokontroler lengkap dengan *pin/port* untuk koneksi serta sudah dilengkapi dengan *downloader*. Dalam segi bahasa pemrograman, arduino memiliki bahasa pemrograman yang lebih mudah dan sederhana terutama bagi pemula. Alasan bahasa pemrograman arduino lebih mudah dan sederhana adalah karena didalam arduino sudah terdapat beberapa *library* yang dapat digunakan untuk merancang pemrograman yang diinginkan. Berikut ini merupakan *flowchart* yang digunakan pada Proyek Akhir ini. *Flowchart* tersebut terdapat pada Gambar 3.12 yang menjelaskan tentang pembacaan sensor dengan menggunakan arduino hingga dapat ditampilkan pada *LabVIEW*.



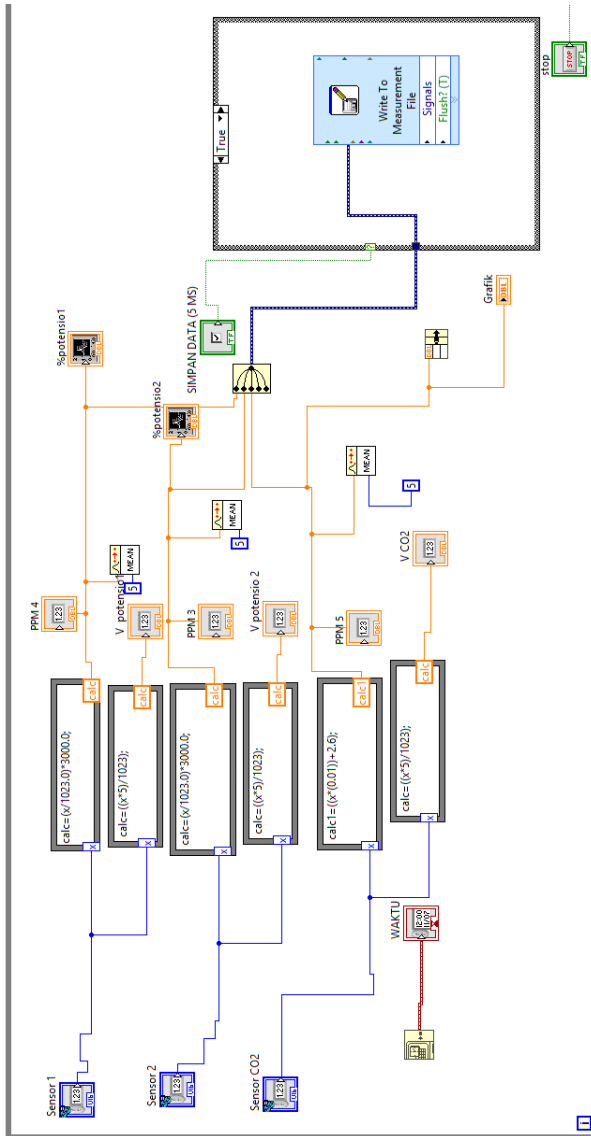
Gambar 3.12 *Flowchart* Pada Rangkaian Arduino

3.4.2 Perancangan Software LabVIEW

Pada perancangan *software LabVIEW* yang pertama dirancang yaitu membuat *block diagram* setelah itu merancang *front panel*. *Block diagram* berisi *source code* yang berfungsi sebagai instruksi untuk *front panel*. Sedangkan pada *front panel* sendiri mengandung *control* dan indikator untuk membangun sebuah VI (*Virtual Instruments*), menjalankan program dan *debug* program.

Pada Gambar 3.13 merupakan perancangan *block diagram*. Pada bagian kiri terdapat *function palatte numeric* dengan tanda nama sensor1, sensor2 dan sensor CO yang berguna untuk menampilkan pembacaan 3 sensor per 200 ms, ditunjukkan pada *function palatte* dengan *icon* jam, yang dikirim melalui *Ethernet*. Pada bagian kanan sensor1, sensor2 dan sensor CO terdapat kotak berwarna abu-abu yang disebut dengan *formula node*, fungsi dari *function palatte* ini adalah untuk menuliskan rumus konversi sinyal analog menjadi besaran PPM (%) dan *function palatte* ini adalah untuk menuliskan rumus konversi sinyal analog menjadi besaran tegangan. Lalu pada bagian sebelah kanan *formula node* disambungkan pada *function palatte numeric* dan grafik sebagai hasil perhitungan PPM dengan tanda nama Sensor1, Sensor2, dan gas CO. Setelah didapatkan nilai perhitungan PPM, disambungkan pada *function palatte for loop* yang didalamnya terdapat *function palatte numeric*, grafik, dan *wait* untuk menampilkan nilai perhitungan dan grafik per 1000 ms. Setelah data sudah didapat, maka data tersebut akan disimpan secara otomatis dalam bentuk microsoft excel.

Pada perancangan *LabVIEW* juga terdapat penambahan pada indikator dengan menggunakan indikator lampu dan bunyi sehingga pada saat Sensor1, Sensor2, ataupun gas CO memiliki kadar gas yang tinggi (500-1000 PPM) maka monitor otomatis berbunyi, dan lampu indikator berubah berwarna merah. Jika Sensor1, Sensor2, ataupun gas CO memiliki kadar siaga maka lampu indikator akan berubah berwarna kuning dan lampu akan berwarna hijau jika kondisi gas CO, Sensor 1 ataupun Sensor 2 dalam keadaan aman.



Gambar 3.13 Block Diagram Pada LabVIEW

BAB IV

PENGUJIAN DAN ANALISA

Dalam membuat suatu sistem, pengujian dan analisa sangat diperlukan. Pengujian dan analisa bertujuan untuk mengetahui system bekerja sesuai rencana atau belum. Dari hasil pengujian dan analisa dapat diketahui kelemahan-kelemahan dari sistem, sehingga dapat dilakukan perbaikan, pengembangan, dan penyempurnaan sistem.

Dalam bab ini dibahas pengujian serta analisa dari perancangan sistem yang telah dibuat pada Proyek Akhir. Pengujian dilakukan secara parsial terlebih dahulu, tujuannya adalah untuk mengetahui kinerja *hardware* setiap unit. Setelah mengetahui respon kinerja *hardware* setiap unit, kemudian akan dilakukan pengujian integrasi sistem secara keseluruhan. Adapun beberapa pengujian yang telah dilakukan antara lain:

1. Pengujian *Input/Output* Arduino Mega
2. Kalibrasi Sensor
3. Kalibrasi menggunakan *LabView*
4. Pengujian *Ethernet Shield*
5. Pengujian *Acces Point*

4.1 Pengujian Input / Output Arduino Mega

Pengujian ini dilakukan pada rangkaian board Arduino Mega 2560. Pengujian dilakukan untuk mengetahui tegangan *output* pada setiap *pin* arduino jika diberi *input high* dan *input low* dengan menggunakan Voltmeter. Cara agar dapat mengetahui tegangan Output saat input high yaitu dengan melakukan:

1. Menyiapkan Arduino Mega 2560 dan multimeter.
2. Memberikan *supply* tegangan pada arduino.
3. Membuat program pada arduino IDE dengan menggunakan logika 1 agar data yang keluar pada pengujian output arduino dalam perintah *high*. Contoh program dapat dilihat pada Gambar 4.1.

```

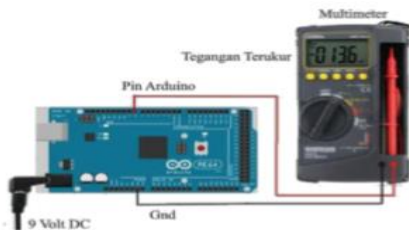
void setup() {
  // put your setup code here, to run once:
  pinMode(A1, OUTPUT);
  pinMode(A3, OUTPUT);
  pinMode(A4, OUTPUT);
  pinMode(A5, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite(A1, HIGH);
  digitalWrite(A3, HIGH);
  digitalWrite(A4, HIGH);
  digitalWrite(A5, HIGH);
  digitalWrite(3, HIGH);
  digitalWrite(4, HIGH);
  digitalWrite(5, HIGH);
}

```

Gambar 4.1 Program Pengujian I/O Arduino Mega 2560 saat *Active High*.

4. Kemudian mengukur tegangan setiap pin pada Arduino dengan multimeter. Skema pengukuran ditunjukkan pada Gambar 4.2.



Gambar 4.2 Skema Pengujian I/O Arduino Mega 2560

Hasil pengukuran pada tiap pin dari *board* Arduino Mega 2560 dapat dilihat pada Tabel 4.1.

Tabel 4.1 Hasil Pengukuran Per *Pin* Saat *Active High*

<i>Pin Analog</i>	Tegangan (Volt)	<i>Pin Digital</i>	Tegangan (Volt)
<i>Pin A0</i>	5,01	<i>Pin 1</i>	5,01
<i>Pin A1</i>	5,01	<i>Pin 3</i>	5,01
<i>Pin A2</i>	5,01	<i>Pin 4</i>	5,01

<i>Pin Analog</i>	Tegangan (Volt)	<i>Pin Digital</i>	Tegangan (Volt)
<i>Pin A3</i>	5,01	<i>Pin 5</i>	5,01
		<i>Pin 7</i>	5,01

Tabel 4.2 Hasil Pengukuran Per *Pin* Saat *Active Low*

<i>Pin Analog</i>	Tegangan (mV)	<i>Pin Digital</i>	Tegangan (mV)
<i>Pin A0</i>	0,8	<i>Pin 1</i>	10,0
<i>Pin A1</i>	0,8	<i>Pin 3</i>	5,9
<i>Pin A2</i>	0,8	<i>Pin 4</i>	1,2
<i>Pin A3</i>	0,8	<i>Pin 5</i>	1,3

Dari hasil Tabel 4.2 dapat disimpulkan bahwa ketika *board* Arduino Mega diberi *input low* pada tiap *pin* maka tegangan yang dihasilkan bernilai rata-rata 1,76 mV yang artinya tegangan yang dikeluarkan sangat kecil dan *pin* tersebut dapat berfungsi sebagai *ground* karena *output board* Arduino Mega bernilai *low* dengan *range* tegangan 0–800 mV.

4.2 Kalibrasi Sensor

Kalibrasi sensor bertujuan agar mengetahui nilai presentasi ke error an suatu sensor dan menguji ke sensitifan sensor tersebut. oleh karena itu, sebelum melakukan pengukuran kualitas udara itu sendiri maka diperlukan kalibrasi untuk menentukan keakuratan sensor.

4.2.1. Kalibrasi Sensor CO

Kalibrasi sensor CO ini bertujuan untuk mengetahui kesensitifan sensor CO dan mengetahui presentase ke erroran sensor tersebut. Sebelum melakukan kalibrasi maka diperlukan adanya pengecekan sensor CO terlebih dahulu. Hal tersebut Hal tersebut bertujuan agar dapat mengetahui apakah sensor CO dapat bekerja dengan baik dan benar dalam mendeteksi gas CO. Sensor CO yang digunakan pada Proyek Akhir ini yaitu MQ-7. Sensor MQ-7 memiliki kepekaan yang baik terhadap gas karbon monoksida. Langkah – langkah yang dilakukan saat melakukan kalibrasi yaitu:

1. Melakukan pemanasan sensor sesuai yang tertera pada data sheet masing- masing sensor. Pada sensor CO sendiri dilakukan pemanasan selama 60 menit agar nilai sensor lebih stabil.

2. Menyiapkan sumber emisi sebagai tolak ukur gas tersebut. Gas emisi yang digunakan kali ini berasal dari Motor Matic berbahan bakar pertamax.
3. Agar sensor mendeteksi sesuai yang kita inginkan, arduino akan diberi sebuah program yang telah dibuat di *software* IDE seperti pada Gambar 4.3.
4. Kemudian dekatkan cerobong gas pada alat dengan sumber emisi yang digunakan.
5. Amatn hasil yang muncul pada lcd alat tersebut
6. Kemudian bandingkan dengan hasil dari alat ukur STAR GAS dengan lcd alat.

```

void loop() {

int sensorValue = analogRead(A0);
int val1 = sensorValue;
Serial.print(val1);
Serial.println();
lcd.clear();
lcd.setCursor(1,3);
lcd.print("COmel: ");
lcd.print(sensorValue);
float voltage = val1 *5/1023.0;
lcd.setCursor(8,1);
lcd.print(" v: ");
lcd.print(voltage, 3);
lcd.println();
Serial.print("\tvv:");
Serial.print(voltage, 3);
Serial.println();
delay(1000);
}

```

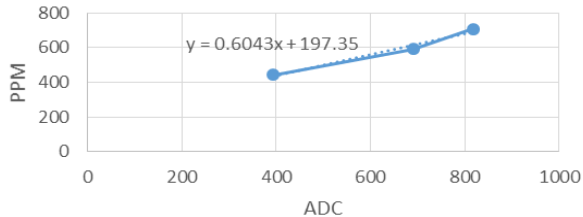
Gambar 4.3 Program Kalibrasi Sensor CO

Hasil kalibrasi sensor CO yang diukur yaitu dari asap motor matic yang berbahan bakar pertamax dapat dilihat pada Tabel 4.3. dan pada hasil kalibrasi terdapat *error* pada pembacaan arduino. *Error* merupakan hasil perbandingan dari nilai pembacaan di stargas dengan pembacaan arduino. Gambar 4.4 merupakan grafik dari hasil kalibrasi untuk mencari nilai slop dan intercept.

Tabel 4.3 Hasil Kalibrasi Sensor CO

No	Temperatur (°C)	Pembacaan di STARGAS (ppm)	Pembacaan pada Arduino (ppm)	Error (%)
1	30,3	394	464,45	0,1788
2	30,9	691	555,69	0,1958

No	Temperatur (°C)	Pembacaan di STARGAS (ppm)	Pembacaan pada Arduino (ppm)	Error (%)
3	30,9	818	624,59	0,2364

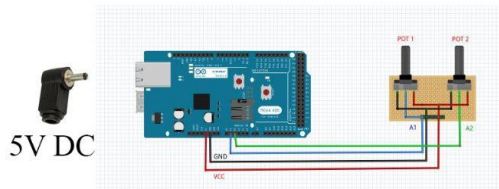


Gambar 4.4 Hasil Kalibrasi Sensor CO (MQ-7)

4.2.2. Kalibrasi Sensor 1

Sensor 1 menggunakan potensio yang menjadi indikator adanya gas CO₂. Kalibrasi potensio 1 ini bertujuan untuk mengetahui apakah potensio sudah dapat bekerja dengan baik atau belum.. Tujuan lain dari melakukan proses kalibrasi ini yaitu melihat keakuratan potensio itu sendiri. Maka langkah – langkah yang dilakukan waktu kalibrasi yaitu

1. Bikin rangkaian seperti Gambar 4.5 yaitu menyambungkan potensio pada port- pot yang ingin digunakan.



Gambar 4.5 Rangkaian Potensio dengan Arduino

2. Agar Potensio 1 dapat digantikan sebagai sensor yang kita inginkan, maka arduino akan diberi sebuah program yang telah dibuat di *software* IDE seperti pada Gambar 4.6.

```

//sensor1
int val = analogRead(A1);
sensor1 = map(val,0,1023,0, 255) *4;
mb.Ireg(sencalc, val);
Serial.print("sensor 1 :");
Serial.print(sensor1);
Serial.print(" ppm ");

```

Gambar 4.6 Program Arduino pada potensio

3. Kemudian amatin hasil yang muncul pada lcd alat.

Tabel 4.4 Hasil Kalibrasi Potensio

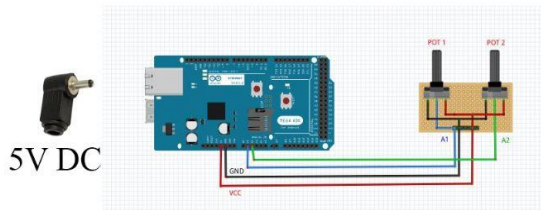
No	Pembacaan pada Arduino (ppm)	Tegangan sensor (Volt)
1	56,83	0,27
2	468,622	2,29
3	880,411	4,31

Pada Tabel 4.4 pembacaan potensio dalam bentuk ppm gas. Pada hasil kalibrasi ini nilai ppm dari 35 sampai dengan 1000 ppm. Hal ini dikarenakan batas tubuh menerima gas kurang dari 1000 ppm. Jika melebihi maka akan dapat menyebabkan gangguan pada tubuh manusia dan bahkan dapat menyebabkan kematian.

4.2.3. Kalibrasi Sensor 2

Sensor 2 menggunakan potensio yang menjadi indikator adanya gas O₂. Kalibrasi potensio 2 ini bertujuan untuk mengetahui apakah potensio sudah dapat bekerja dengan baik atau belum.. Tujuan lain dari melakukan proses kalibrasi ini yaitu melihat keakuratan potensio itu sendiri. Maka langkah – langkah yang dilakukan waktu kalibrasi yaitu

1. Bikin rangkaian seperti Gambar 4.7 yaitu menyambungkan potensio pada *port- port* yang ingin digunakan.



Gambar 4.7 Rangkaian Potensio dengan Arduino

2. Agar Potensio 2 dapat digantikan sebagai sensor yang kita inginkan, maka arduino akan diberi sebuah program yang telah dibuat di *software* IDE seperti pada Gambar 4.8.

```
int val1 = analogRead(A2); //A2
sensor2 = map(val1, 0, 1023, 0, 255) * 4;
mb.Ireg(sencalcl, val1);
Serial.print("sensor 2 :");
Serial.print(sensor2);
Serial.print(" ppm ");
//Serial.println(" ");
```

Gambar 4.8 Program Arduino pada potensio

3. Kemudian amatin hasil yang muncul pada lcd alat. Dan pada Tabel 4.5 merupakan hasil dari kalibrasi potensio

Tabel 4.5 Hasil Kalibrasi Potensio

No	Pembacaan pada Arduino (ppm)	Tegangan sensor (Volt)
1	46,59	0,22
2	398,82	1,95
3	880,411	4,31

Pada Tabel 4.5 pembacaan potensio dalam bentuk ppm gas. Pada hasil kalibrasi ini nilai ppm dari 35 sampai dengan 1000. Hal ini dikarenakan batas tubuh menerima gas kurang dari 1000 ppm.

Jika melebihi maka akan dapat menyebabkan gangguan pada tubuh manusia dan bahkan dapat menyebabkan kematian.

4.3 Kalibrasi dengan menggunakan *LabVIEW*

Kalibrasi sensor menggunakan *LabVIEW* ini bertujuan agar mengetahui kondisi sensor serta membandingkan nilai ppm sensor dari LCD alat dengan yang ada pada tampilan *LabView*. Pada kalibrasi sensor dengan *LabVIEW* ini hanya menggunakan prosedur.

Tabel 4.6 Kalibrasi Potensio1 pada *Software LabVIEW*

No	Tegangan (Volt)	Pembacaan pd Arduino (ppm)	Pembacaan LabView (ppm)
1	0,303	61,818	61,818
2	0,610	124,63	124,63
3	1,45	297,13	297,13
4	1,720	350,968	350,968
5	2,35	479,589	479,589
6	2,66	534,40	534,40
7	3,07	628,152	628,152
8	3,641	742,815	742,815

Tabel 4.7 Hasil kalibrasi Potensio 2 pada *Software LabVIEW*

No	Tegangan (Volt)	Pembacaan pd Arduino (ppm)	Pembacaan LabView (ppm)
1	0,464	94,721	94,721
2	0,77	158,534	158,534
3	1,22	249,267	249,267
4	1,749	356,9	356,9
5	2,29	468,22	468,22
6	2,97	606,217	606,217
7	3,9	796,65	796,65
8	4,59	937,24	937,24

Tabel 4.8 Hasil kalibrasi sensor CO pada *Software LabVIEW*

No	Tegangan pd Analog 1 (Volt)	Pembacaan pd Arduino (ppm)	Pembacaan LabVIEW (ppm)
1	0,332	238,44	238,44
2	0,337	239,04	239,04
3	0,351	240,85	240,85
4	0,347	240,25	240,25
5	0,395	246,29	246,29
6	0,4	246,90	246,90
7	0,41	248,11	248,11
8	0,415	248,72	248,72

Seperti pada Tabel 4.6 sampai dengan Tabel 4.8 dapat disimpulkan bahwa untuk melihat apakah data tersebut sudah dikirim dengan benar ataupun tidak dapat dilihat dari perbedaan angka pada kalibrasi sensor menggunakan *LabVIEW*. Jika nilai dari Lcd dengan hasil yang ditampilkan di *LabVIEW* sama maka data berhasil dikirimkan.

4.4 Pengujian Komunikasi *Ethernet*

Pengujian ini dilakukan pada modul *Ethernet Shield* yang telah dipasang pada bagian atas *board* Arduino Mega. Dalam hal ini pengujian komunikasi dibagi menjadi dua, yaitu pengujian komunikasi *Ethernet* pada *board* Arduino Mega dan pengujian komunikasi *Ethernet* dengan *Software LabVIEW*.

4.4.1. Pengujian pada *Board* Arduino Mega

Sebelum menggunakan komunikasi *Ethernet*, sebaiknya dilakukan pengujian komunikasi *Ethernet* pada *board* Arduino Mega terlebih dahulu. Pengujian ini dilakukan dengan cara menyamakan *setting* IP (*Internet Protocol*) *Address* yang telah diprogram pada *board* Arduino Mega, dapat dilihat pada Gambar 4.9 yang bergaris bawah merah, dengan *setting* IP *Address* pada *Ethernet shield*.

```
30 Program_terbaru1 | Arduino 1.8.6 Hourly Build 2018/01/03 03:33
File Edit Sketch Tools Help

Program_terbaru1

double CG;

void setup() {
  Serial.begin(9600);
  lcd.begin();
  lcd.backlight();
  lcd.clear();
  //lcd.init(); // initialize the lcd
  //lcd.init();
  // Print a message to the LCD.
  // put your setup code here, to run once:
  byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
  byte ip[] = { 192, 168, 0, 200 };
  mb.config(mac, ip);
  Serial.begin(9600);
  mb.addIreg(sencalc);
  mb.addIreg(sencalc1);
  mb.addIreg(sencalc2);

  //set();
}

void loop()
{
  // put your main code here, to run repeatedly
}
```

Gambar 4.9 Program Pada Arduino Mega

```
cmd Select Command Prompt
Microsoft Windows [Version 10.0.17134.523]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\USER>ping 192.168.0.200

Pinging 192.168.0.200 with 32 bytes of data:
Reply from 192.168.0.200: bytes=32 time=3ms TTL=128
Reply from 192.168.0.200: bytes=32 time=1ms TTL=128
Reply from 192.168.0.200: bytes=32 time=2ms TTL=128
Reply from 192.168.0.200: bytes=32 time=2ms TTL=128

Ping statistics for 192.168.0.200:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 3ms, Average = 2ms
```

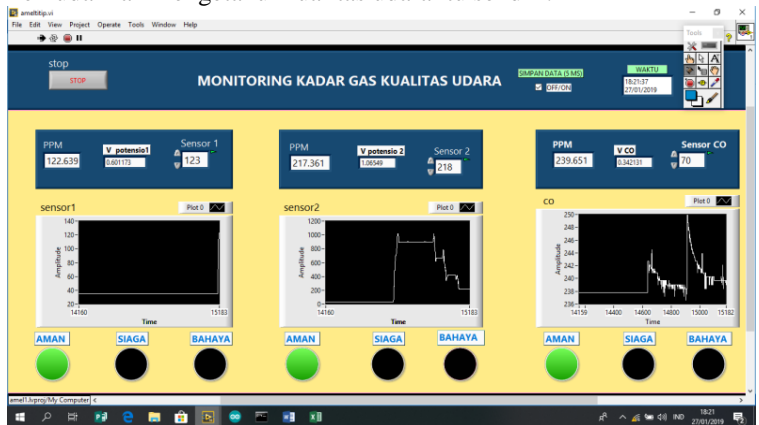
Gambar 4.10 Command Prompt Pada Laptop Dengan IP address Arduino Mega

Dapat dilihat pada 2 Gambar di atas menunjukkan bahwa *Ethernet Shield* yang tersambung pada laptop telah siap digunakan. Ketika mengirim perintah *ping* nomor IP Address pada kolom *command prompt* maka laptop akan memunculkan informasi bahwa nomor IP Address telah bekerja, dapat dilihat pada Gambar 4.10

4.4.2. Pengujian pada LabVIEW

Setelah pengujian komunikasi *Ethernet* pada Arduino Mega, pengujian komunikasi *Ethernet* perlu dilakukan pada *Software LabVIEW* untuk mengetahui apakah *Ethernet Shield* dapat mengirimkan data pada *software LabVIEW* atau tidak.

Pengujian ini dilakukan dengan cara menyambungkan 3 sensor yaitu sensor 1, Sensor CO, dan Sensor 2. Setelah rangkaian sensor tersambung, nyalakan Fan pada alat. Lalu, amati tampilan pada *Software LabVIEW*. Setelah itu data yang di tampilan *LabVIEW* akan secara langsung menyimpan pada excel dan terdapat indikator untuk memudahkan mengetahui kualitas udara itu sendiri.



Gambar 4.11 Tampilan Pengujian *Interface* pada *LabVIEW*

4.5 Pengujian Jarak

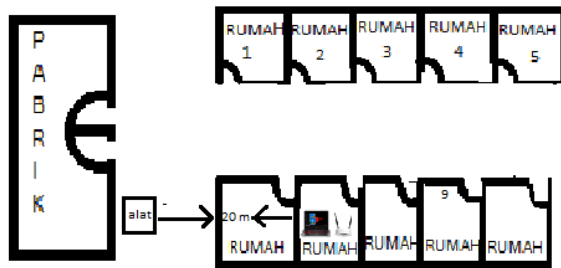
Fungsi telemetering adalah memungkinkan pengukuran jarak jauh dan pelaporan informasi kepada operator sistem. Untuk mengetahui stabilitas koneksi terhadap jarak antara PC dan *access point* maka perlu dilakukan tes ping terhadap IP Address arduino.

4.5.1 Pengujian Koneksi Dengan Penghalang

Untuk mengetahui jarak maksimal yang dapat dicapai dari koneksi *access point* dan PC dengan penghalang, maka dilakukan langkah-langkah pengujian sebagai berikut :

1. Memastikan PC dan *access point* telah terhubung.

2. Meletakkan PC dan *access point* dengan jarak dengan menggunakan penghalang. Pada pengecekan *access point* ini menggunakan jarak 5 meter, 10 meter, 17 meter dan 20 meter.
3. Melakukan pengecekan LabVIEW apakah pada saat jarak tersebut apa masih dapat mengirim data atau tidak. Sketsa pengujian koneksi dapat dilihat pada ilustrasi Gambar 4.12



Gambar 4.12 Ilustrasi Pengujian Koneksi Alat Dengan Menggunakan Penghalang

Pengujian koneksi dengan *command prompt* pada Gambar 4.13 menunjukkan bahwa Ethernet Shield telah tersambung pada laptop siap digunakan. Ketika mengirim perintah ping nomor IP Address pada kolom *command prompt* maka laptop akan memunculkan informasi bahwa nomor IP Address telah bekerja, Gambar 4.13 menunjukkan pengujian komunikasi Ethernet saat jarak PC dan *access point* sejauh 5 meter, pada jarak ini tidak ada loss koneksi yang terdeteksi.

```
C:\Users\USER>ping 192.168.0.200

Pinging 192.168.0.200 with 32 bytes of data:
Reply from 192.168.0.200: bytes=32 time=1ms TTL=128
Reply from 192.168.0.200: bytes=32 time=1ms TTL=128
Reply from 192.168.0.200: bytes=32 time=1ms TTL=128
Reply from 192.168.0.200: bytes=32 time=2ms TTL=128

Ping statistics for 192.168.0.200:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 2ms, Average = 1ms
```

Gambar 4.13 *Command Prompt* Pada Laptop Dengan IP Address Arduino Mega 2560 Pada Jarak 5 Meter Dengan Penghalang

Pengujian koneksi dengan *command prompt* pada Gambar 4.14 menunjukkan bahwa *Ethernet Shield* telah tersambung pada laptop siap digunakan. Ketika mengirim perintah ping nomor IP Address pada kolom *command prompt* maka laptop akan memunculkan informasi bahwa nomor IP Address telah bekerja, Gambar 4.14 menunjukkan pengujian komunikasi *Ethernet* saat jarak PC dan *access point* sejauh 10 meter, pada jarak ini tidak ada *loss* koneksi yang terdeteksi.

```
C:\Users\USER>ping 192.168.0.200

Pinging 192.168.0.200 with 32 bytes of data:
Reply from 192.168.0.200: bytes=32 time=1ms TTL=128
Reply from 192.168.0.200: bytes=32 time=5ms TTL=128
Reply from 192.168.0.200: bytes=32 time=2ms TTL=128
Reply from 192.168.0.200: bytes=32 time=6ms TTL=128

Ping statistics for 192.168.0.200:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 6ms, Average = 3ms
```

Gambar 4.14 *Command Prompt* Pada Laptop Dengan IP Address Arduino Mega 2560 Pada Jarak 10 Meter Dengan Penghalang

Pengujian koneksi dengan *command prompt* pada Gambar 4.15 menunjukkan bahwa *Ethernet Shield* telah tersambung pada laptop siap digunakan. Ketika mengirim perintah ping nomor IP Address pada kolom *command prompt* maka laptop akan memunculkan informasi bahwa nomor IP Address telah bekerja,

Gambar 4.15 menunjukkan pengujian komunikasi *Ethernet* saat jarak PC dan *access point* sejauh 20 meter, pada jarak ini ada *loss* koneksi yang terdeteksi sebanyak 50%.

```
C:\Users\USER>ping 192.168.0.200

Pinging 192.168.0.200 with 32 bytes of data:
Request timed out.
Reply from 192.168.0.150: Destination host unreachable.
Reply from 192.168.0.150: Destination host unreachable.
Request timed out.

Ping statistics for 192.168.0.200:
    Packets: Sent = 4, Received = 2, Lost = 2 (50% loss),
```

Gambar 4.15 *Command Prompt* Pada Laptop Dengan IP Address Arduino Mega 2560 Pada Jarak 20 Meter Dengan Penghalang

Pada langkah – langkah di atas dapat menunjukkan bahwa *Ethernet Shield* yang terhubung pada laptop mengalami gangguan atau tidaknya pada sistem komunikasinya, dicek dari hasil ping pada *command prompt* di laptop pada jarak antara laptop dan *access point* dengan penghalang dinding ruangan.

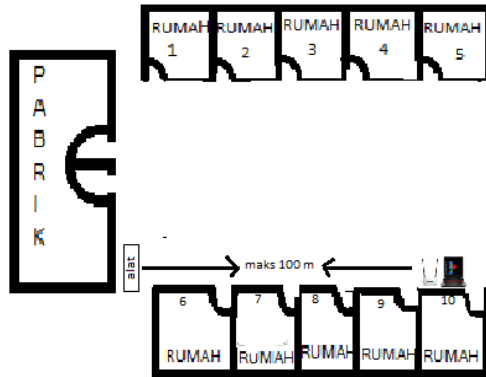
Tabel 4.9 Hasil Pengujian *Acces Point* dengan Penghalang

Jarak (meter)	Koneksi Loss (%)	Time Average
5	0	1ms
10	0	3ms
20	50	1ms

4.5.2 Pengujian Koneksi Tanpa Penghalang

Untuk mengetahui jarak maksimal yang dapat dicapai dari koneksi *access point* dan PC tanpa penghalang, maka dilakukan langkah-langkah pengujian sebagai berikut :

1. Memastikan PC dan *access point* telah terhubung.
2. Meletakkan PC dan *access point* dengan jarak yang di inginkan secara berurutan tanpa penghalang.
3. Melakukan pengecekan *LabVIEW* apakah pada saat jarak tersebut apa masih dapat mengirim data atau tidak.. Sketsa pengujian koneksi dapat dilihat pada ilustrasi Gambar 4.16.



Gambar 4.16 Ilustrasi Pengujian Koneksi Alat Dengan Tanpa Menggunakan Penghalang

Pengujian koneksi dengan *command prompt* pada Gambar 4.17 menunjukkan bahwa *Ethernet Shield* telah tersambung pada laptop siap digunakan. Ketika mengirim perintah ping nomor IP Address pada kolom *command prompt* maka laptop akan memunculkan informasi bahwa nomor IP Address telah bekerja, Gambar 4.17 menunjukkan pengujian komunikasi *Ethernet* saat jarak PC dan *access point* sejauh 1,5 meter, pada jarak ini tidak ada *loss* koneksi yang terdeteksi

```

Command Prompt
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 6ms, Average = 3ms

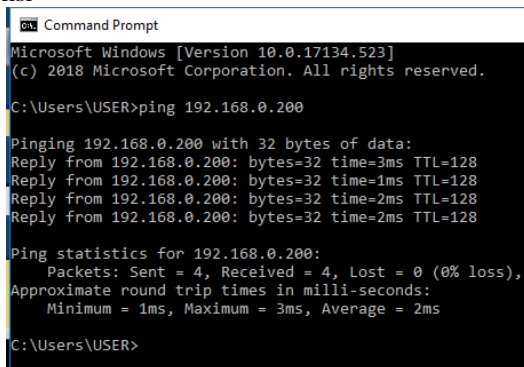
C:\Users\USER>ping 192.168.0.200

Pinging 192.168.0.200 with 32 bytes of data:
Reply from 192.168.0.200: bytes=32 time=1ms TTL=128
Reply from 192.168.0.200: bytes=32 time=1ms TTL=128
Reply from 192.168.0.200: bytes=32 time=1ms TTL=128
Reply from 192.168.0.200: bytes=32 time=2ms TTL=128

Ping statistics for 192.168.0.200:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 2ms, Average = 1ms
  
```

Gambar 4.17 *Command Prompt* Pada Laptop Dengan IP Address Arduino Mega 2560 Pada Jarak 1,5 Meter Tanpa Penghalang

Pengujian koneksi dengan *command prompt* pada Gambar 4.18 menunjukkan bahwa *Ethernet Shield* telah tersambung pada laptop siap digunakan. Ketika mengirim perintah ping nomor IP Address pada kolom *command prompt* maka laptop akan memunculkan informasi bahwa nomor IP Address telah bekerja, Gambar 4.18 menunjukkan pengujian komunikasi *Ethernet* saat jarak PC dan *access point* sejauh 8 meter, pada jarak ini tidak ada *loss* koneksi yang terdeteksi



```
Command Prompt
Microsoft Windows [Version 10.0.17134.523]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\USER>ping 192.168.0.200

Pinging 192.168.0.200 with 32 bytes of data:
Reply from 192.168.0.200: bytes=32 time=3ms TTL=128
Reply from 192.168.0.200: bytes=32 time=1ms TTL=128
Reply from 192.168.0.200: bytes=32 time=2ms TTL=128
Reply from 192.168.0.200: bytes=32 time=2ms TTL=128

Ping statistics for 192.168.0.200:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 3ms, Average = 2ms

C:\Users\USER>
```

Gambar 4.18 *Command Prompt* Pada Laptop Dengan IP Address Arduino Mega 2560 Pada Jarak 8 Meter Tanpa Penghalang

Pengujian koneksi dengan *command prompt* pada Gambar 4.19 menunjukkan bahwa *Ethernet Shield* telah tersambung pada laptop siap digunakan. Ketika mengirim perintah ping nomor IP Address pada kolom *command prompt* maka laptop akan memunculkan informasi bahwa nomor IP Address telah bekerja, Gambar 4.19 menunjukkan pengujian komunikasi *Ethernet* saat jarak PC dan *access point* sejauh 13 meter, pada jarak ini tidak ada *loss* koneksi yang terdeteksi

```
C:\Users\USER>ping 192.168.0.200

Pinging 192.168.0.200 with 32 bytes of data:
Reply from 192.168.0.200: bytes=32 time=1ms TTL=128
Reply from 192.168.0.200: bytes=32 time=5ms TTL=128
Reply from 192.168.0.200: bytes=32 time=2ms TTL=128
Reply from 192.168.0.200: bytes=32 time=6ms TTL=128

Ping statistics for 192.168.0.200:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 6ms, Average = 3ms

C:\Users\USER>
```

Gambar 4.19 *Command Prompt* Pada Laptop Dengan IP Address Arduino Mega 2560 Pada Jarak 13 Meter Tanpa Penghalang

Pengujian koneksi dengan *command prompt* pada Gambar 4.20 menunjukkan bahwa *Ethernet Shield* telah tersambung pada laptop siap digunakan. Ketika mengirim perintah ping nomor IP Address pada kolom *command prompt* maka laptop akan memunculkan informasi bahwa nomor IP Address telah bekerja, Gambar 4.20 menunjukkan pengujian komunikasi *Ethernet* saat jarak PC dan access point sejauh 50 meter, pada jarak ini tidak ada *loss* koneksi yang terdeteksi

```
Select Command Prompt
Microsoft Windows [Version 10.0.17134.523]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\USER>ping 192.168.0.200

Pinging 192.168.0.200 with 32 bytes of data:
Reply from 192.168.0.200: bytes=32 time=44ms TTL=128
Reply from 192.168.0.200: bytes=32 time=3ms TTL=128
Reply from 192.168.0.200: bytes=32 time=4ms TTL=128
Reply from 192.168.0.200: bytes=32 time=546ms TTL=128

Ping statistics for 192.168.0.200:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 3ms, Maximum = 546ms, Average = 149ms
```

Gambar 4.20 *Command Prompt* Pada Laptop Dengan IP Address Arduino Mega 2560 Pada Jarak 50 Meter Tanpa Penghalang

Pada langkah – langkah di atas dapat menunjukkan bahwa *Ethernet Shield* yang terhubung pada laptop mengalami gangguan atau tidaknya pada sistem komunikasinya, dicek dari hasil ping pada *command prompt* di laptop pada jarak antara laptop dan *access point* dengan penghalang dinding ruangan.

Tabel 4.10 Hasil Pengujian *Acces Point WiFi* dengan Tanpa Penghalang

Jarak (Meter)	Koneksi <i>Loss</i> (%)	Time Avarage
1.5	0	1ms
8	0	2ms
13	0	3ms
50	0	149ms
160	100	2ms

4.6 Pengujian Area

Pada pengujian ini bertujuan untuk mengetahui ketelitian monitoring kualitas udara, walaupun alat diletakkan jauh dari sumber gas tersebut. Pada pengujian ini, sumber gas yang digunakan yaitu, asap kendaraan motor untuk sumber gas CO, sensor 1 dan sensor 2. Serta kami juga menggunakan asap rokok sebagai sumber gas CO, sensor 1 dan sensor 2. Hasil pengujiannya bisa dilihat pada Tabel 4.11.

Tabel 4.11 Hasil Pengujian Peletakan Alat dengan diberikan Jarak

No pengujian	Jarak	Potensio 1	Gas CO	Potensio 2
1	< 1 meter	966,158	298,72	908,32
2	< 1 meter	931,261	300,02	739,823
3	1 meter	855,484	286,90	648,094
4	1.5 meter	742,81	256,29	560,352
5	2 meter	.522,463	249,85	356,95
6	3 meter	271,202	246,29	140,587
7	4 meter	-	-	-

Pada hasil Tabel 4.11 dapat diambil kesimpulan jika jarak antara sumber gas dengan alat dapat berpengaruh. Semakin dekat alat pendeteksi dengan sumber gas maka semakin besar ppm yang akan di tampilkan pada layar LCD serta pada tampilan LabView tersebut.

4.7 Pengujian Secara Keseluruhan

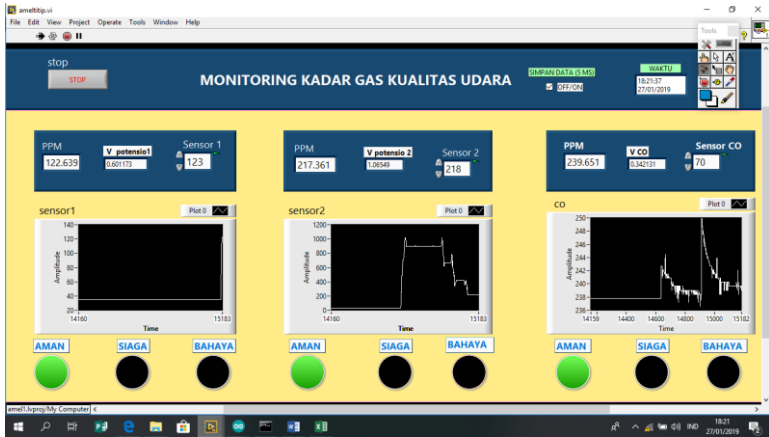
Pengujian secara keseluruhan merupakan gabungan dari beberapa pengujian yang telah dilakukan dan pengujian alat pemantauan kualitas udara pada *prototype* pemantauan kualitas udara. Berikut ini pada Tabel 4.12 merupakan pengambilan data di tempat parkir Departemen Teknik Elektro Otomasi ITS.

Tabel 4.12 Data Pengukuran Gas Pada Parkiran Departemen Teknik Elektro Otomasi

NO	Sensor 1 (PPM)	Sensor 2 (PPM)	CO (PPM)
1	862,4633	863,4604	238,44
2	861,4663	863,4604	239,04
3	860,4692	864,4575	240,85
4	862,4633	864,4575	240,25
5	862,4633	864,4575	246,29
6	861,4663	864,4575	246,90
7	862,4633	863,4604	248,72



Gambar 4.21 Lokasi Alat Pada Departemen Teknik Elektro Otomasi



Gambar 4.22 Tampilan *LabVIEW* Hasil Telemetering pada Parkiran Departemen Teknik Elektro Otomasi

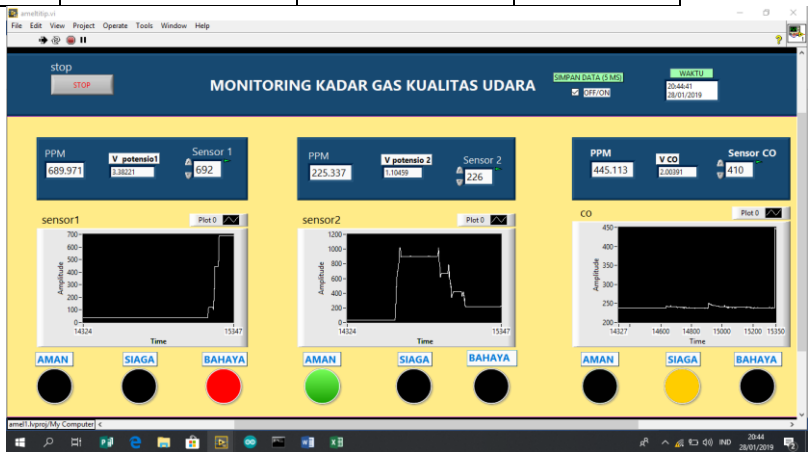
Pengambilan data di Parkiran Departemen Teknik Elektro Otomasi dilakukan pada jam 16.00 dan pengambilan data dilakukan selama 20 menit. Dari data Tabel 4.12 menunjukkan kadar gas CO pada kendaraan motor sebesar 238 – 250 PPM. Kadar tersebut dikatakan aman karena masih banyaknya perpohonan pada lingkungan sekitar parkiran. Akan tetapi, jika udara tersebut terhirup lebih dari 3 jam tidak menggunakan masker atau penutup hidung dapat menyebabkan terjadi iritasi mata dan pusing kepala. Pada Gambar 4.21 menunjukkan data tampilan LCD saat pengambilan.

Pada Gambar 4.22 merupakan hasil tampilan *LabVIEW* hasil pengukuran kadar gas CO pada parkiran Departemen Teknik Elektro Otomasi. Data pengukuran yang ditampilkan *LabVIEW* tersebut akan otomatis tersimpan pada excel. Kadar CO yang terbaca menunjukkan nilai 239,561 PPM. Nilai kadar gas CO tersebut termasuk batas ambang aman yaitu 36-300 PPM sehingga status berwarna hijau. Pada Tabel 4.13 merupakan hasil dari pengambilan data pada lampu merah Kertajaya.

Tabel 4.13 Data Pengukuran Gas Pada Lampu Merah Kertajaya

NO	Sensor 1 (PPM)	Sensor 2 (PPM)	CO (PPM)
1	862,4633	224,3402	429,4012

NO	Sensor 1 (PPM)	Sensor 2 (PPM)	CO (PPM)
2	861,4663	224,3402	430,0055
3	860,4692	224,3402	430,6098
4	862,4633	224,3402	431,2141
5	862.4633	223,3431	432,4227
6	861,4663	223,3431	433,027
7	862,4633	224,3402	432,4227
8	862,4633	224,3402	434,2356
9	861,4663	224,3402	434,2356
10	862,4633	224,3402	434,8399
11	862,4633	223,3431	435,4442
12	861,4663	223,3431	436,6528
13	862,4633	223,3431	437,8614
14	863,4604	223,3431	45511
15	864,4575	223,3431	455,11



Gambar 4.23 Tampilan LabVIEW Hasil Telemetering pada Lampu Merah Kertajaya

Pengambilan data di lampu merah Kertajaya dilakukan pada jam 12.00 dan pengambilan data dilakukan selama 20 menit. Dari data Tabel 4.13 menunjukkan kadar gas CO pada kendaraan motor dan mobil di daerah lampu merah kertajaya sebesar 429 – 450 PPM. Pada

Gambar 4.23 merupakan hasil tampilan LabVIEW hasil pengukuran kadar gas CO pada lampu merah Kertajaya. Data pengukuran yang ditampilkan pada LabVIEW tersebut akan otomatis tersimpan pada excel. Kadar CO yang terbaca menunjukkan nilai 445,11 PPM. Nilai kadar gas CO tersebut termasuk batas ambang siaga yaitu 400-600 PPM sehingga status berwarna kuning. Kadar tersebut dikatakan siaga karena banyaknya polusi dan volume kendaraan yang ada dilingkungan tersebut. Jika udara tersebut terhirup lebih dari 2 Jam tidak menggunakan masker atau penutup hidung dapat menyebabkan terjadi iritasi mata dan sesak nafas.

BAB V

PENUTUP

Dari hasil yang telah didapatkan selama proses pembuatan serta proses analisa data untuk Proyek Akhir ini, maka dapat diambil beberapa kesimpulan dan saran yang berguna untuk perbaikan dan pengembangan agar nantinya bisa bermanfaat.

5.1 Kesimpulan

Dari seluruh tahapan yang sudah dilaksanakan pada penyusunan Proyek Akhir ini, mulai dari studi literatur, perancangan dan pembuatan sampai pada pengujiannya maka dapat disimpulkan bahwa :

1. Dari hasil uji setelah kalibrasi dilakukan pengujian perbandingan pengukuran agar didapat hasil yang akurat. Hasil yang diperoleh yaitu alat ini dapat mendeteksi gas CO dengan *error* 0,17 – 0.23%
2. Pada alat Monitoring pemantauan kualitas udara dapat mengetahui besar PPM pada udara dilingkungan secara *real time* tersebut.
3. Dari hasil pengujian, data dari sensor 1, sensor 2 dan gas CO dapat di telemetring dengan *range* jarak 0 - 150 meter

5.2 Saran

Untuk lebih memperbaiki dan menyempurnakan kinerja dari alat ini, maka perlu disarankan untuk penelitian selanjutnya yaitu diharapkan dapat disempurnakan dengan penambahan sensor gas sesuai pengukur kualitas udara yang ada pada daerah industri, menggunakan perangkat komunikasi data selain *Ethernet Shield* karena seringnya gangguan saat pengiriman data pada *software LabVIEW*, tampilan *software LabVIEW* dibuat lebih baik dan menarik lagi.

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

DAFTAR PUSTAKA

- [1], *Pencemaran Udara di Indonesia*, <http://https://www.dw.com/id/walhi-ketiadaan-data-akurat-perburuk-polusi-di-indonesia/a-43620871>, 12 November 2018.
- [2] Abdul, Kadir, “*Panduan Praktis Mempelajari Aplikasi Mikrokontroler Dan Pemrogramannya Menggunakan Arduino*”, *Andi Offset*, Jogjakarta, 2013
- [3] Artanto Dian, *Interaksi Arduino dan LabVIEW*, Elex Media Komputindo, Jakarta, 2012.
- [4] Agung Bairuni Wardi, “Rancang Bangun Sistem Alat Pendeteksi Gas CO, dan CO₂ Sebagai Informasi Pencemaran Udara Menggunakan Arduino dan Ethernet”, *Tugas Akhir Program Studi D3 Teknik Elektro FTI-ITS*, 2012
- [5] Didin Jalanudin “*Sensor Gas CO, Gas CO₂ dan Gas O₂*” Detik.com, <http://detik.com./macam-macam/sensor/udara/> 12 Mei 2018
- [6] Muhammad Amnuddin “**LCD**” detik.com, <http://detik.com/apa/itu/lcd/> 15 April 2018
- [7] Kusuma Laksana, “Rancang Bangun Alat Pendeteksi Gas Beracun CO Pada Area Aktivitas Gunung Berapi”, *Tugas Akhir*, Program D3 Teknik Elektro FTI-ITS, Surabaya, 2018.
- [8] Mochamad, Fajar, “*Mudah Belajar Mikrokontroler Arduino*”, Penerbit Informatika, Bandung, 2017
- [9] Saludin Muis, “*Prinsip Kerja LCD*”, Graha Ilmu, Yogyakarta, 2013.
- [10] Sri Suryaningsih, “Rancang Bangun Alat Ukur Kadar Gas Co Berbasis Nirkabel Rf Untuk Pemantauan Kondisi Pencemaran Udara”, *Tugas Akhir*, Departemen Fisika FMIPA Universitas Padjadjaran, 2015
- [11], *Pengertian Relay dan Fungsinya*, <https://teknikelektronika.com/pengertian-relay-fungsi-relay/>, 24 Mei 2018.

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

LAMPIRAN A

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <SPI.h>
#include <Modbus.h>
#include <ModbusIP.h>
#include <Ethernet.h>

LiquidCrystal_I2C lcd(0x27, 20, 4);
const int sencalc = 100;
const int sencalc1 = 110;
const int sencalc2 = 120;

long ts;

int Calc;
int Calc1;
int Calc2;

ModbusIP mb;
float sensor1;
float sensor2;
double CO;

void setup() {

Serial.begin(9600);
lcd.begin();
lcd.backlight();
lcd.clear();
//lcd.init();           // initialize the lcd
//lcd.init();
// Print a message to the LCD.
// put your setup code here, to run once:
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
byte ip[] = { 192, 168, 0, 200 };
mb.config(mac, ip);
Serial.begin(9600);
```

```

mb.addIreg(sencalc);
mb.addIreg(sencalc1);
mb.addIreg(sencalc2);

//sei();

}

void loop()

{
// put your main code here, to run repeatedly

mb.task();
if (millis() > ts + 1000) {
  ts = millis();
  cli();

  //sensor1
  int val = analogRead(A1);
  sensor1 = map(val,0,1023,0, 255)*4;
  if(sensor1<35)
  {
    sensor1=35;
  }
  mb.Ireg(sencalc, val);
  Serial.print("sensor 1 :");
  Serial.print(sensor1);
  Serial.print(" ppm ");

  int val1 = analogRead(A2);///A2
  sensor2 = map(val1,0,1023,0, 255)*4;
  if(sensor2<35)
  {
    sensor2=35;
  }
  mb.Ireg(sencalc1, val1);
  Serial.print("sensor 2 :");
  Serial.print(sensor2);

```

```

Serial.print(" ppm ");
//Serial.println(" ");

//sensor3
int val2 = analogRead(A0);
CO = ((val2 * (0.6043)) + 197.35);
int sensorVal = val2;
mb.Ireg(sencalc2, sensorVal);
Serial.print("CO :");
Serial.print(CO);
Serial.print(" ppm ");
Serial.println(" ");

}

sei();

lcd.clear();
lcd.setCursor(0, 0);
lcd.print("sensor1 :");
lcd.setCursor(10, 0);
lcd.print(sensor1);//s1
lcd.setCursor(15, 0);
lcd.print("PPM");
//sensor2
lcd.setCursor(0, 1);
lcd.print("sensor2 :");
lcd.setCursor(10, 1);
lcd.print(sensor2);//s2
lcd.setCursor(15, 1);
lcd.print("PPM");

//sensor3
lcd.setCursor(0, 2);
lcd.print("CO  :");
lcd.setCursor(10, 2);
lcd.print(CO);//co
lcd.setCursor(15, 2);
lcd.print("PPM");

```

```
delay(500);
```

```
}
```

LAMPIRAN B



Description

The relay module is an electrically operated switch that allows you to turn on or off a circuit using voltage and/or current much higher than a microcontroller could handle. There is no connection between the low voltage circuit operated by the microcontroller and the high power circuit. The relay protects each circuit from each other.

The each channel in the module has three connections named NC, COM, and NO. Depending on the input signal trigger mode, the jumper cap can be placed at high

level effective mode which 'closes' the normally open (NO) switch at high level input and at low level effective mode which operates the same but at low level input.

Specifications

- On-board EL517 photoelectric coupler with photoelectric isolating anti-interference ability strong
- On-board 5V, 10A / 250VAC, 10A / 30VDC relays
- Relay long life can absorb 100000 times in a row
- Module can be directly and MCU I/O link, with the output signal indicator
- Module with diode current protection, short response time
- PCB Size: 45.8mm x 32.4mm

Pin Configuration



1. VCC: 5V DC
2. COM: 5V DC
3. IN1: high/low output
4. IN2: high/low output
5. GND: ground

LAMPIRAN C

µA7800 SERIES POSITIVE-VOLTAGE REGULATORS

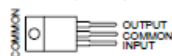
DLV5096J - MAY 1976 - REVISED MAY 2003

- 3-Terminal Regulators
- Output Current up to 1.5 A
- Internal Thermal-Overload Protection
- High Power-Dissipation Capability
- Internal Short-Circuit Current Limiting
- Output Transistor Safe-Area Compensation

KC (TO-220) PACKAGE
(TOP VIEW)



KCS (TO-220) PACKAGE
(TOP VIEW)



KTE PACKAGE
(TOP VIEW)



description/ordering information

This series of fixed-voltage integrated-circuit voltage regulators is designed for a wide range of applications. These applications include on-card regulation for elimination of noise and distribution problems associated with single-point regulation. Each of these regulators can deliver up to 1.5 A of output current. The internal current-limiting and thermal-shutdown features of these regulators essentially make them immune to overload. In addition to use as fixed-voltage regulators, these devices can be used with external components to obtain adjustable output voltages and currents, and also can be used as the power-pass element in precision regulators.

ORDERING INFORMATION

T _J	V _{O(NOM)} (V)	PACKAGE†	ORDERABLE PART NUMBER	TOP-SIDE MARKING	
0°C to 125°C	5	POWER-FLEX (KTE)	Reel of 2000	µA7805KTER	µA7805C
		TO-220 (KC)	Tube of 50	µA7805KIC	µA7805C
		TO-220, short shoulder (KCS)	Tube of 20	µA7805KCS	µA7805C
	8	POWER-FLEX (KTE)	Reel of 2000	µA7808KTER	µA7808C
		TO-220 (KC)	Tube of 50	µA7808KIC	µA7808C
		TO-220, short shoulder (KCS)	Tube of 20	µA7808KCS	µA7808C
	10	POWER-FLEX (KTE)	Reel of 2000	µA7810KTER	µA7810C
		TO-220 (KC)	Tube of 50	µA7810KIC	µA7810C
		TO-220, short shoulder (KCS)	Tube of 20	µA7810KCS	µA7810C
	12	POWER-FLEX (KTE)	Reel of 2000	µA7812KTER	µA7812C
		TO-220 (KC)	Tube of 50	µA7812KIC	µA7812C
		TO-220, short shoulder (KCS)	Tube of 20	µA7812KCS	µA7812C
15	POWER-FLEX (KTE)	Reel of 2000	µA7815KTER	µA7815C	
	TO-220 (KC)	Tube of 50	µA7815KIC	µA7815C	
	TO-220, short shoulder (KCS)	Tube of 20	µA7815KCS	µA7815C	
24	POWER-FLEX (KTE)	Reel of 2000	µA7824KTER	µA7824C	
	TO-220 (KC)	Tube of 50	µA7824KIC	µA7824C	

† Package drawings, standard packing quantities, thermal data, symbolization, and PCB design guidelines are available at www.ti.com/package.



Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

PRODUCTS: SMD information is current as of publication date.
Products conform to specifications per form of data sheets.
Product warranty. Product processing does not necessarily include
testing of all parameters.



TEXAS
INSTRUMENTS

POST OFFICE BOX 655503 • DALLAS, TEXAS 75265

Copyright © 2003, Texas Instruments Incorporated

1

DAFTAR RIWAYAT HIDUP



Nama : Amalia Habibah
TTL : Surabaya, 4
September 1997
Jenis Kelamin : Perempuan
Agama : Islam
Alamat Rumah : Jl. Tanah Merah
III no. 21
Surabaya
Telpn/HP : 085731607992
E-mail :
amaliageon@gm
ail.com

RIWAYAT PENDIDIKAN

1. 2002 – 2003 : TK Panca Dharma
2. 2003 – 2009 : SDN Tanah Kali Kedinding IV Surabaya
3. 2009-2012 : SMP Wachid Hasyim I Surabaya
4. 2012 – 2015 : SMAN IX Surabaya
5. 2015 – 2018 : Departemen Teknik Elektro Otomasi Program Studi Komputer Kontrol - Fakultas Vokasi (FV) Institut Teknologi Sepuluh Nopember (ITS)

PENGALAMAN KERJA

1. Kerja Praktek di PT. PLN (Persero) Area Madiun
2. Kerja Praktek di AIR NAV Surabaya

PENGALAMAN ORGANISASI

1. Sekertaris Departemen PSDM Periode 2017/2018 Himpunan Mahasiswa D3 Teknik Elektro, Institut Teknologi Sepuluh Nopember