



TUGAS AKHIR - TE 145561

**PEMBUATAN PENDETEKSI GAS BERACUN *PORTABLE*
UNTUK PENGGALI SUMUR BERBASIS MIKROKONTROLER**

Ninik Vatmawati
NRP 2213039026
Masaji Alif Rahman
NRP 2213039032

Dosen Pembimbing
Ir. Josaphat Pramudijanto, M.Eng.
Subadi, S.Pd

PROGRAM STUDI D3 TEKNIK ELEKTRO
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2016



FINAL PROJECT - TE 145561

***PORTABLE DETECTOR OF POISON GAS TO WELL DIGGER
BASED BY MICROCONTROLLER***

Ninik Vatmawati
NRP 2213039026
Masaji Alif Rahman
NRP 2213039032

Supervisor
Ir. Josaphat Pramudijanto, M.Eng.
Subadi, S.Pd
ELECTRICAL ENGINEERING D3 STUDY PROGRAM
Faculty of Industrial Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2016

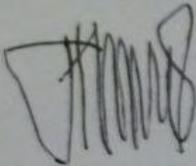
PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul "**Pembuatan Pendeteksi Gas Beracun Portable Untuk Penggali Sumur Berbasis Mikrokontroler**" 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.

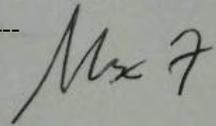
Surabaya, 1 Juni 2016



Ninik Vatmawati

2213039026

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



Masaji Alif Rahman

2213039032

**PEMBUATAN PENDETEKSI GAS BERACUN *PORTABLE* UNTUK
PENGGALI SUMUR BERBASIS MIKROKONTROLER**

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Ahli Madya Teknik**

Pada

**Bidang Studi Elektro Industri
Program Studi D3 Teknik Elektro
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember**

Menyetujui :

Dosen Pembimbing I

Dosen Pembimbing II



Ir. Josaphat Pramudijanto, M.Eng
NIP. 196210051990031003

Subadi, S.Pd

NIP. 196309231986031

**SURABAYA
JUNI, 2016**

**LEMBAR PERNYATAAN
PERSETUJUAN PUBLIKASI KARYA ILMIAH
UNTUK KEPENTINGAN AKADEMIS**

Sebagai mahasiswa Institut Teknologi Sepuluh Nopember Surabaya, yang bertanda tangan di bawah ini saya :

Nama : Ninik Vahmawati
Nrp. : 2213 039 026
Jurusan / Fak. : D3 Teknik Elektro / FTI
Alamat kontak : Ngelom Megare RT/Rw 01/01 NO. 6LT, Sepanjang - Sidoarjo
a. Email : ninik.vahmawati.nv@gmail.com
b. Telp/HP : -1089620825600

Menyatakan bahwa semua data yang saya *upload* di Digital Library ITS merupakan hasil final (revisi terakhir) dari karya ilmiah saya yang sudah disahkan oleh dosen penguji. Apabila dikemudian hari ditemukan ada ketidaksesuaian dengan kenyataan, maka saya bersedia menerima sanksi.

Demi perkembangan ilmu pengetahuan, saya menyetujui untuk memberikan **Hak Bebas Royalti Non-Eksklusif (Non-Exclusive Royalti-Free Right)** kepada Institut Teknologi Sepuluh Nopember Surabaya atas karya ilmiah saya yang berjudul :

Pembuatan Alat Pendeteksi Gas Beracun Portable Untuk Pengawasan Sumur Berbasis Mikrokontroler

Dengan Hak Bebas Royalti Non-Eksklusif ini, Institut Teknologi Sepuluh Nopember Surabaya berhak menyimpan, mengalih-media/format-kan, mengelolanya dalam bentuk pangkalan data (*database*), mendistribusikannya, dan menampilkan/mempublikasikannya di internet atau media lain untuk kepentingan akademis tanpa meminta ijin dari saya selama tetap mencantumkan nama saya sebagai penulis/pencipta. Saya bersedia menanggung secara pribadi, segala bentuk tuntutan hukum yang timbul atas pelanggaran Hak Cipta dalam karya Ilmiah saya ini tanpa melibatkan pihak Institut Teknologi Sepuluh Nopember Surabaya.

Demikian pernyataan ini saya buat dengan sebenarnya.



Dibuat di : Surabaya
Pada tanggal : 29-06-2016
Yang menyatakan,

Ninik Vahmawati

Nrp. 2213 039 026

KETERANGAN :

Tanda tangan pembimbing wajib dibubuhi stempel jurusan.

Form dicetak dan diserahkan di bagian Pengadaan saat mengumpulkan hard copy TA/Tesis/Disertasi.

PEMBUATAN PENDETEKSI GAS BERACUN *PORTABLE* UNTUK PENGGALI SUMUR BERBASIS MIKROKONTROLER

Nama 1 : Ninik Vatmawati
NRP : 2213039026
Nama 2 : Masaji Alif Rahman
NRP : 2213039032
Pembimbing 1 : Ir. Josaphat Pramudijanto, M.Eng
NIP : 196210051990031003
Pembimbing 2 : Subadi S.Pd
NIP : 196309231986031

ABSTRAK

Pembuatan sumur timba memiliki resiko atau bahaya yang sangat tinggi. Hal ini dapat mengakibatkan penggali sumur keracunan udara yang terdapat di dalam tanah karena di dalam tanah yang akan digali memiliki beberapa gas beracun diantaranya gas CO, CO₂, CH₄ dan H₂S. Dan yang paling berbahaya adalah gas CO dan CH₄, karena jika gas tersebut terhirup melebihi ambang batas, maka akan menyebabkan sulit bernafas, tidak sadarkan diri, hingga kematian. Dari permasalahan tersebut maka dibuatlah alat pendeteksi gas beracun *portable* untuk penggali sumur berbasis mikrokontroler yang dapat mendeteksi adanya kadar gas beracun yang ditempatkan pada badan penggali sumur. Jika gas beracun terdeteksi oleh sensor gas MQ-7 untuk pendeteksi gas CO dan sensor gas MQ-4 untuk pendeteksi gas CH₄, maka *buzzer* dan led menyala sebagai indikator, kemudian RF 433 MHz mengirimkan data ke LCD (*Liquid Crystal Display*) yang ada diatas sumur sebagai media penampil informasi. Hasil akhir dari alat ini yaitu ketika ppm mencapai angka lebih dari 500 atau melebihi ambang batas, maka indikator *buzzer* akan berbunyi, led akan menyala dan pada layar LCD akan menampilkan gas beracun yang terdeteksi.

Kata Kunci : Gas Karbon Monoksida , Gas Metana , Sumur Beracun.

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

**PORTABLE DETECTOR OF POISON GAS TO WELL DIGGER
BASED BY MICROCONTROLLER**

Name 1 : Ninik Vatmawati
Registration Number : 2213039026
Name 2 : Masaji Alif Rahman
Registration Number : 2213039032
Supervisor 1 : Ir. Josaphat Pramudijanto, M.Eng
ID : 196210051990031003
Supervisor 2 : Subadi S.Pd
ID : 196309231986031

ABSTRACT

Dipper well construction risk or danger is very high. This can result in well-digging poisoning the air contained in the ground as in the soil to be excavated have some toxic gases including CO gas, CO₂, CH₄ and H₂S. And the most dangerous is the gas CO and CH₄, because if the gas is inhaled exceeds the threshold, it will cause difficult breathing, unconsciousness, and death. Of the problems it is made portable toxic gas detection equipment for digging wells based microcontroller that can detect the presence of high levels of toxic gas that is placed on the body-digging. If a toxic gas is detected by a gas sensor MQ-7 for the detection of CO gas and gas sensors MQ-4 for detecting CH₄, then the buzzer and the LED lights up as an indicator, then RF 433 MHz transmit data to LCD (Liquid Crystal Display) that are above the well as a media viewer information. The final result of this tool is when ppm reached more than 500, or exceeds the threshold, the indicator buzzer will sound, and the LED will light up on the LCD screen will display the toxic gases were detected.

Keywords : Gas Carbon Monoxide , methane , Toxic wells.

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

KATA PENGANTAR

Dengan mengucapkan puji syukur kehadirat Allah SWT yang telah melimpahkan rahmat, hidayahserta karunia-Nya sehingga penulis dapat menyelesaikan Tugas Akhir dengan judul :

“PEMBUATAN PENDETEKSI GAS BERACUN *PORTABLE* UNTUK PENGGALI SUMUR BERBASIS MIKROKONTROLER”

Tugas Akhir ini merupakan sebagian syarat untuk menyelesaikan studi dan memperoleh gelar Ahli Madya Teknik di Jurusan D3 Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember.

Dengan selesainya Tugas Akhir ini penulis menyampaikan terima kasih sebesar-besarnya kepada:

1. Kedua orang tua atas limpahan doa, kasih sayang dan teladan hidup bagi penulis.
2. Bapak Ir. Josaphat Pramudijanto, M.Eng selaku dosen pembimbing 1.
3. Bapak Subadi S.Pd selaku dosen pembimbing 2.
4. Seluruh staf pengajar dan administrasi Jurusan D3 Teknik Elektro FTI-ITS.
5. Semua pihak yang telah banyak membantu untuk menyelesaikan Tugas Akhir ini yang tidak dapat penulis sebutkan satu persatu.

Harapan kami sebagai penulis adalah semoga terselesaikannya Tugas Akhir ini dapat bermanfaat bagi kami serta pembaca. Sadar atas keterbatasan yang dimiliki oleh penulis karena hasil dari Tugas Akhir ini jauh dari kesempurnaan. Demikian penulis sudah berusaha semaksimal mungkin dan pintu maaf serta saran kritik yang membangun penulis harapkan.

Surabaya, 1 Juni 2016

Penulis

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

DAFTAR ISI

	HALAMAN
HALAMAN JUDUL.....	i
TUGAS AKHIR.....	v
HALAMAN PENGESAHAN.....	vii
<i>ABSTRACT</i>	xi
KATA PENGANTAR	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR	xvii
DAFTAR TABEL.....	xix
BAB I.....	1
1.1 Latar Belakang.....	1
1.2 Permasalahan	2
1.3 Batasan Masalah	2
1.4 Tujuan	3
1.5 Sistematika Laporan	3
1.6 Relevansi	4
BAB II.....	5
2.1 Sumur Galian.....	5
2.1.1 Karbon Monoksida (CO).....	5
2.1.2 Karbon Dioksida (CO ₂).....	6
2.1.3 Hidrogen Sulfida (H ₂ S)	6
2.1.4 Gas Metana (CH ₄)	7
2.1.5 Gas Yang Dideteksi.....	7
2.2 Arduino Uno R3[2].....	7
2.2.1 Daya (<i>Power</i>) dan Memori.....	9
2.2.2 <i>Input & Output</i>	10
2.2.3 Komunikasi	10
2.3 Arduino Nano	11
2.3.1 Daya (<i>Power</i>) dan Memori.....	12
2.3.2 <i>Input & Output</i>	12
2.3.3 Komunikasi	13
2.4 <i>Software</i> Arduino.....	14
2.5 Sensor Gas Monoksida (MQ-7).....	15
2.6 Sensor Gas Metana (MQ-4).....	16

2.7	<i>Wireless</i> Modul RF 433 mHz.....	17
2.8	Modul <i>Relay</i>	18
2.9	<i>Liquid Crystal Display</i> (LCD).....	19
2.10	<i>Exhaust Fan</i>	19
2.11	LED (<i>Light Emitting Diode</i>)	20
2.12	<i>Buzzer</i>	21
2.13	Baterai.....	21
BAB III.....		23
3.1	Perancangan Perangkat Keras	23
3.2	Spesifikasi Alat	24
3.3	Perancangan <i>Hardware</i> Pendeteksi Gas Beracun di Sumur.....	25
3.3.1	Perancangan <i>Hardware</i> Kontrol	25
3.4	Arduino	26
3.5	Perancangan Perangkat Lunak	27
3.5.1	AntarMuka UART TTL.....	27
3.5.2	Program Arduino IDE.....	27
3.5.3	AntarMuka Muka Sensor MQ-4 dan MQ-7	29
3.5.4	AntarMuka pada <i>Hardware</i> Kontrol Pemantau	30
3.5.5	Perancangan Mikrokontroler dan <i>Relay</i>	31
3.6	<i>Flowchart</i> Program Mikrokontroler	32
3.7	Pembacaan Nilai Sensor Gas Metana (MQ-4)	32
3.8	Pembacaan Nilai Sensor Gas Karbon Monoksida (MQ-7).....	33
3.8.1	Menampilkan Data Pada LCD	34
3.9	Inisialisasi <i>Wireless</i> RF 433 mHz	35
BAB IV.....		37
4.1	<i>Pengujian Liquid Crystal Display</i> (LCD)	37
4.2	Pengujian Modul Sensor MQ-7 (CO)	38
4.3	Pengujian Modul Sensor MQ-4 (CH ₄).....	42
4.4	Pengujian <i>Wireless</i> RF 433 mHz.....	45
4.5	Pengambilan Data Pada Sumur	47
4.6	Pengujian Keseluruhan.....	49
BAB V PENUTUP		53
5.1	Kesimpulan	53
5.2	Saran	53
DAFTAR PUSTAKA.....		55
LAMPIRAN.....		A1

DAFTAR TABEL

HALAMAN

Tabel 2.1 Konsentrasi Karbon Monoksida.....	6
Tabel 2.2 Spesifikasi Arduino R3	9
Tabel 2.3 Spesifikasi Arduino Nano	11
Tabel 2.4 Komponen Sensor MQ – 7.....	16
Tabel 2.5 Spesifikasi <i>Transmitter</i> RF 433 mHz.....	17
Tabel 2.6 Spesifikasi <i>Receiver</i> RF 433 mHz.....	18
Tabel 3 .1 Spesifikasi Alat	24
Tabel 4.1 Pengujian Alat Pemanas (<i>Heater</i>) Sensor MQ-7	40
Tabel 4.2 Hasil Pembacaan Alat ECOM dan data ADC sensor MQ7.....	40
Tabel 4.3 Hasil Pembacaan Sensor Gas CO	41
Tabel 4.4 Pengujian Alat Pemanas (<i>Heater</i>) Sensor MQ-4	43
Tabel 4.5 Hasil Pembacaan Sensor Gas CH ₄	44
Tabel 4.6 Hasil Pembacaan Sensor Gas CH ₄	44
Tabel 4.7 Jarak Sinyal <i>Wireless</i>	47
Tabel 4.8 Tabel <i>Delay</i> Terhadap Jarak Pengiriman	47
Tabel 4.9 Pengambilan Data Pada Sumur	48
Tabel 4.10 Pengujian Keseluruhan	51

DAFTAR GAMBAR

HALAMAN

Gambar 2.1 Board Arduino Uno R3	8
Gambar 2.2 Arduino Nano	11
Gambar 2.3 Tampilan IDE Arduino Dengan Sebuah <i>Sketch</i>	15
Gambar 2.4 Struktur Karbon Monoksida Sensor (MQ – 7)	15
Gambar 2. 5 Struktur Gas Metana Sensor (MQ-4)	16
Gambar 2.6 <i>Transmitter dan Receiver</i> RF 433 mHz.....	17
Gambar 2.7 <i>Modul Relay 2 channel 5V</i>	19
Gambar 2.8 <i>Liquid Crystal Display</i> (LCD).....	19
Gambar 2.9 <i>Exhaust Fan</i>	20
Gambar 2.10 LED (<i>Light Emitting Diode</i>).....	20
Gambar 2.11 <i>Buzzer 5 Volt</i>	21
Gambar 3.1 Sketsa Alat	24
Gambar 3. 2 Rancangan <i>Hardware</i> Kontrol <i>Wearable</i>	25
Gambar 3.3 Rancangan <i>Hardware</i> Kontrol Pemantau.....	26
Gambar 3.4 Tampilan Awal <i>Software</i> Arduino.....	28
Gambar 3.5 Perancangan Sensor Ke Mikrokontroler (<i>hardware wearable</i>)	29
Gambar 3.6 AntarMuka pada <i>Hardware</i> Kontrol Pemantau.....	30
Gambar 3.7 Perancangan <i>Relay</i> ke Mikrokontroler	31
Gambar 3.8 <i>Flowchart</i> Mikrokontroler	32
Gambar 3.9 Kode Program Inisialisasi	33
Gambar 3.10 Kode Progam Pembacaan Sensor CH ₄	33
Gambar 3.11 Kode Program Inisialisasi	34
Gambar 3.12 Kode Program Pembacaan Sensor Gas MQ-7	34
Gambar 3.13 Kode Program Menampilkan Hasil Pembacaan Sensor	35
Gambar 3.14 <i>Inisialisasi</i> RF 433 mHz.....	35
Gambar 3.15 Komunikasi <i>Serial</i>	36
Gambar 4.1 Rangkaian Pengujian LCD	37
Gambar 4.2 Program Pengujian <i>Liquid Crystal Display</i>	38
Gambar 4.3 Hasil Pengujian	38
Gambar 4.4 Rangkaian Pengujian Sensor MQ-7	39
Gambar 4.5 Program Pembacaan Sensor CO.....	39

Gambar 4.6 Alat ECOM	40
Gambar 4.7 Rangkaian Pengujian Senor MQ-4.....	42
Gambar 4.8 Program Pembacaan Sensor CH ₄	43
Gambar 4.12 Pengujian <i>Wireless</i> Rf 433 mHz	45
Gambar 4.9 Pemrograman Pengiriman data <i>Wireless</i>	46
Gambar 4.10 Pemrograman Penerima Data <i>Wireless</i>	46
Gambar 4.11 Data Yang Muncul Pada Layar LCD	47
Gambar 4.13 Pengambilan Data Pada Sumur	48
Gambar 4.14 Rangkaian Pengujian Keseluruhan	49
Gambar 4.15 Pendeteksi Awal Gas Beracun	50
Gambar 4.16 Terdeteksinya Gas CO	50
Gambar 4.17 Terdeteksinya Gas CH ₄	51
Gambar 4.18 Terdeteksinya Gas CO dan CH ₄	51

Halaman ini sengaja dikosongkan

BAB I

PENDAHULUAN

1.1 Latar Belakang

Di daerah pedesaan penggunaan sumur timba masih banyak digunakan, bahkan mayoritas perumahan di pedesaan masih banyak menggunakan sumur timba tersebut karena lebih menghemat biaya dan juga air di pedesaan masih tergolong bersih dibandingkan di perkotaan. Akan tetapi dalam pembuatan sumur timba memiliki resiko atau bahaya yang sangat tinggi. Pada saat melakukan penggalian, penggali sumur menggali tanah dengan peralatan yang masih manual seperti cangkul dan linggis setelah mencapai kedalaman beberapa meter penggali sumur akan masuk kedalam tanah untuk mengangkat material tanpa menggunakan bantuan mesin, tak dipungkiri banyak musibah yang terjadi pada saat orang melakukan penggalian sumur bahkan yang tragis sering juga berakibat kejadian dengan korban lebih dari satu orang, seperti diberitakan pada surat kabar dan internet di Banjarnegara 11 Oktober 2015 terjebak gas beracun tiga penggali sumur tewas, Banyuwangi 6 Februari 2016 tiga orang penggali sumur tewas karena gas beracun, Jakarta Utara 27 Oktober 2015 dua orang penggali sumur tewas usai menghirup gas beracun dan masih banyak lagi link berita dapat dilihat pada lampiran B.

Di dalam tanah mempunyai beberapa kandungan gas berbahaya seperti CO, CO₂, CH₄ dan H₂S. Gas yang paling berbahaya di dalam sumur yaitu Gas Metana (CH₄) dan Karbon Monoksida (CO). Gas Metana (CH₄) merupakan gas yang lebih berat dari udara sehingga cenderung berada pada daerah yang rendah seperti halnya di dalam tanah atau galian sumur. Gas Metana (CH₄) ini tidak berwarna, dan mudah terbakar. Jika Gas Metana (CH₄) dihirup terlalu lama dalam konsentrasi yang tinggi, maka akan menyebabkan kelumpuhan pada indera penciuman bahkan kematian. Karbon Monoksida (CO) merupakan gas yang tidak berwarna, tidak berbau dan tidak berasa. Sifat Karbon Monoksida (CO) yang tidak berbau menyulitkan orang untuk mendeteksi keberadaan gas tersebut. Kadar Karbon Monoksida (CO) yang banyak terdapat di dalam sumur dapat menyebabkan orang yang menghirupnya pingsan dan jika tidak segera di tolong maka akan menyebabkan kematian.

Dari berbagai permasalahan yang ditimbulkan, maka dibuat inovasi yaitu pembuatan pendeteksi gas beracun *portable* untuk penggali sumur berbasis mikrokontroler yang bertujuan untuk mencegah terlambatnya pertolongan jika sewaktu-waktu penggali sumur menghirup gas beracun. Alat ini menggunakan sensor gas MQ-7 untuk mendeteksi gas Karbon Monoksida (CO) dan sensor gas MQ-4 untuk mendeteksi gas Metana (CH₄), Arduino sebagai mikrokontroler, RF 433 mHz sebagai media komunikasi dan LCD sebagai media penampil informasi.

1.2 Permasalahan

Pada saat pembuatan sumur timba, hal tersebut memiliki resiko atau bahaya yang sangat tinggi. Saat melakukan penggalian, penggali sumur menggali tanah dengan peralatan yang masih manual seperti cangkul dan linggis setelah mencapai kedalaman beberapa meter penggali sumur tersebut akan masuk kedalam tanah untuk mengangkat material tanpa menggunakan bantuan mesin, Sehingga dapat mengakibatkan orang tersebut keracunan gas beracun yang ada didalam tanah.

Dinilai sangat meresahkan karena dapat menimbulkan korban jiwa yang dikarenakan kurangnya keamanan atau pengetahuan mengenai keselamatan dan kesehatan kerja (K3) saat melakukan proses penggalian sumur dan juga belum tersedianya alat yang dapat menunjukkan adanya kadar gas beracun di dalam sumur sehingga dapat mencegah adanya penggali sumur yang keracunan gas beracun. Hal ini mendorong agar alat pendeteksi gas beracun *portable* di dalam sumur ini bisa dipasang pada badan penggali sumur.

1.3 Batasan Masalah

Batasan masalah dalam perancangan purwa rupa pembuatan pendeteksi gas beracun *portable* untuk penggali sumur berbasis mikrokontroler :

1. Senyawa gas yang dideteksi adalah senyawa gas Karbon Monoksida (CO) dan senyawa gas Metana (CH₄)
2. *Controller* yang digunakan adalah Arduino Uno R3 dan Arduino Nano
3. Tegangan input memakai baterai Li-po
4. Pengiriman data ke sebuah LCD dengan menggunakan RF 433 mHz

5. Hasil pendeteksi gas akan di tampilkan pada Liquid Crystal Display (*LCD*)

1.4 Tujuan

Memberikan solusi kepada para penggali sumur agar dapat memantau gas beracun yang ada di dalam sumur sehingga dapat mencegah adanya korban jiwa saat proses penggalian sumur.

Membuat sebuah alat yang dapat mendeteksi kadar Gas Karbon Monoksida (CO) dan Gas Metana (CH_4) di dalam sumur yang langsung dikirimkan ke sebuah Liquid Crystal Display (*LCD*) untuk mengetahui adanya kadar gas beracun di dalam sumur galian.

1.5 Sistematika Laporan

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

Bab I Pendahuluan

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

Bab II Teori Dasar

Bab ini menjelaskan tentang tinjauan pustaka dan teori-teori penunjang yang dijadikan prinsip dasar dan mendukung dalam perencanaan dan pembuatan alat.

Bab III Perancangan Sistem

Bab ini membahas perencaan dan pembuatan alat yang meliputi perencaan dan pembuatan perangkat keras (*Hardware*) dan perangkat lunak (*Software*). Pada bab ini dijelaskan juga secara flowchart cara kerja dari alat *monitoring* kadar *CO*.

Bab IV Pengujian dan Analisa

Bab ini membahas hasil simulasi dan implementasi serta analisis dari pengujian alat tersebut yang meliputi kalibrasi, pengukuran, pengujian, dan anilisa alat.

Bab V Penutup

Bab ini berisi kesimpulan dan saran dari hasil pembahasan yang telah diperoleh.

1.6 Relevansi

Hasil yang diperoleh dari alat monitoring kadar CO dan CH₄ diharapkan mempermudah *monitoring* gas karbon monoksida dan gas metana, pengembangan metode pengawasan, dan perbandingan metode *monitoring* yang tepat untuk gas CO dan CH₄.

BAB II

TEORI DASAR

2.1 Sumur Galian[1]

Sumur Galian adalah sarana menampung air tanah untuk memperoleh air tak dipungkiri banyak musibah yang terjadi pada saat orang melakukan penggalian sumur bahkan yang tragis sering juga berakibat kejadian dengan korban lebih dari satu orang. Apabila dilihat lebih jauh lagi, korban pertama yang mengalami permasalahan dalam sumur umumnya karena mengalami sesak pernafasan. Kondisi ini terjadi karena minimnya udara segar dalam lubang sumur yang sempit. Tetapi dalam banyak kasus, penyebab kematian lebih banyak karena keracunan gas beracun yang ada di dalam lubang sumur. Faktor keracunan gas ini dapat terjadi karena udara dalam sumur sangat minim secara alami, karena di dalamnya terdapat sumber alam atau gas hasil dekomposisi bahan organik dari tanah di sekitar sumur. Biasanya hal ini terjadi untuk daerah yang kaya akan sumber gas alam atau juga karena tanah di situ sebelumnya merupakan daerah bekas timbunan sampah. Proses kematian yang terjadi karena sesak nafas ini sering juga disebut sebagai keracunan gas beracun. Paru-paru yang dalam proses bernafas seharusnya memperoleh pasokan oksigen (O_2) tetapi akan menyerap gas lain dan yang paling berbahaya adalah Metana (CH_4) atau Karbon Monoksida (CO). Gas Metana (CH_4) dapat berada dalam lubang sumur sebagai gas alam yang mungkin terjadi karena terjadinya dekomposisi alami dari dalam tanah atau gas lainnya yang sering di temukan di dalam sumur yaitu CO_2 dan H_2S .

2.1.1 Karbon Monoksida (CO)[2]

Karbon Monoksida adalah gas yang terdiri dari satu atom karbon (C) dan satu atom oksigen (O). Gas ini tidak berbau, tidak berwarna, tidak berasa dan tidak mengiritasi. Namun Karbon Monoksida ini mudah terbakar dan sangat beracun apabila terhirup oleh manusia dan memasuki sistem peredaran darah. Karbon Monoksida terjadi akibat proses pembakaran yang tidak sempurna akibat kurangnya oksigen. Hal ini bisa terjadi pada kendaraan bermotor, alat pemanas, tungku kayu, bahkan asap rokok. Gas CO muncul sebagai salah satu dari gas buangan mesin pompa air akibat pembakaran bahan bakar yang tidak sempurna. Apabila mesin

pompa air dimasukkan dan diaktifkan di dalam lubang sumur, tentu saja akan otomatis akan mengurangi jumlah udara segar dalam sumur untuk proses pembakaran itu. Pada waktu bersamaan hasil pembakaran akan menghasilkan gas buangan seperti CO₂, gas hidrokarbon dan terutama CO, dan akibatnya adalah gas-gas inilah yang akan terhirup oleh orang yang ada di dalam lubang sumur. Berikut Tabel 2.1 yang menunjukkan Konsentrasi Karbon Monoksida (CO) dari konsentrasi yang rendah hingga melebihi ambang batas.

Tabel 2.1 Konsentrasi Karbon Monoksida

Konsentrasi Co (ppm)	Konsentrasi COHb dalam darah (%)	Gangguan pada tubuh
3	0,98	Tidak ada
5	1,3	Belum begitu terasa
10	2,1	Sistem syaraf sentral
20	3,7	Panca indera
40	6,9	Fungsi jantung
60	10,1	Sakit kepala
80	1,33	Sulit bernapas
100	1,65	Pingsan - Kematian

2.1.2 Karbon Dioksida (CO₂)[2]

Gas karbon dioksida (CO₂) adalah gas tidak berwarna, tak berbau, tak terbakar, tidak reaktif dan mempunyai berat jenis 1,53. Pada konsentrasi rendah tidak bersifat racun, tetapi konsentrasi antara 3 - 5 % mengaktifkan saluran pernafasan, dan sakit kepala. Pada konsentrasi antara 8 – 15 % menimbulkan sakit kepala, pening, muntah-muntah, bahkan dapat mengakibatkan meninggal bila korban tidak mendapat cukup oksigen. Konsentrasi yang lebih tinggi secara cepat menyebabkan koma dan kematian. Konsentrasi maksimum di udara yang diizinkan sebesar 5.000 ppm CO₂. Di udara normal konsentrasinya 0,03 %.

2.1.3 Hidrogen Sulfida (H₂S)[2]

Gas Hidrogen Sulfide (H₂S) mempunyai sifat asam, merupakan gas tidak berwarna, berbau busuk seperti bau telur busuk, berat jenis kira-kira 1,19, termasuk gas beracun, mudah terbakar dan dapat membentuk campuran yang eksplosif dengan udara. Dampaknya bagi kesehatan manusia, pada konsentrasi rendah dapat mengiritasi mata dan saluran

mata kemudian untuk konsentrasi tinggi dapat mengakibatkan susah bernafas dan sampai pingsan.

2.1.4 Gas Metana (CH₄)[2]

Metana adalah ikatan C-H yang ekuivalen. Struktur elektroniknya dapat dijelaskan dengan 4 ikatan orbital molekul yang dihasilkan dari orbital valensi C dan H yang saling melengkapi. Pada suhu ruangan dan tekanan standar, Metana adalah gas yang tidak berwarna dan tidak berbau. Sebagai gas, Metana hanya mudah terbakar bila konsentrasinya mencapai 5-15% di udara. Metana terdapat pada sampah-sampah organik setelah dilakukan perombakan oleh bakteri sehingga dapat terurai di dalam tanah. Metana yang berbentuk cair tidak akan terbakar kecuali diberi tekanan tinggi (4-5 atmosfer). Efek akut dari terpapar oleh gas Metana adalah kekurangan oksigen, yaitu < 16%. Masalah kesehatan akan timbul bila terpapar Metana dalam konsentrasi tinggi. Napas menjadi cepat, denyut nadi meningkat, koordinasi otot menurun, emosi meningkat, mual, muntah, kehilangan kesadaran, gagal napas, dan kematian.

2.1.5 Gas Yang Dideteksi[2]

Dari semua penjelasan macam-macam gas beracun yang ada, kami menyimpulkan untuk mendeteksi Karbon Monoksida (CO) dan Gas Metana (CH₄), karena sifat kedua gas tersebut yang sangat membahayakan. Efek dari kedua gas beracun ini juga dapat menyebabkan efek berantai yang dapat berujung kematian.

2.2 Arduino Uno R3[2]

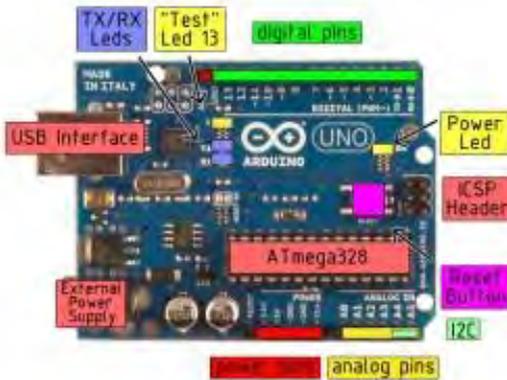
Arduino UNO R3 yang ditunjukkan pada Gambar 2.1 adalah sebuah *board* mikrokontroler yang didasarkan pada ATmega328. Arduino UNO R3 mempunyai 14 pin digital *input* atau *output*, 6 *input* analog, Sebuah osilator Kristal 16 MHz, sebuah koneksi USB, sebuah *power jack*, sebuah *ISCP header*, dan sebuah tombol *reset*. Arduino UNO R3 memuat semua yang dibutuhkan untuk menunjang mikrokontroler, mudah menghubungkan ke sebuah komputer dengan sebuah kabel USB atau mensuplainya dengan sebuah adaptor AC ke DC atau menggunakan baterai untuk memulainya.

Arduino memiliki kelebihan tersendiri disbanding board mikrokontroler yang lain selain bersifat *open source*, arduino juga mempunyai bahasa pemrogramannya sendiri yang berupa bahasa C. Selain itu dalam *board* arduino sendiri sudah terdapat *loader* yang berupa USB

sehingga memudahkan kita ketika kita memprogram mikrokontroler didalam arduino. Sedangkan pada kebanyakan *board* mikrokontroler yang lain yang masih membutuhkan rangkaian *loader* terpisah untuk memasukkan program ketika kita memprogram mikrokontroler. *Port* USB tersebut selain untuk *loader* ketika memprogram, bisa juga difungsikan sebagai port komunikasi serial.

Arduino menyediakan 20 pin I/O, yang terdiri dari 6 pin *input* analog dan 14 pin digital *input/output*. Untuk 6 pin analog sendiri bisa juga difungsikan sebagai *output* digital jika diperlukan *output* digital tambahan selain 14 pin yang sudah tersedia. Untuk mengubah pin analog menjadi digital cukup mengubah konfigurasi pin pada program. Dalam *board* kita bisa lihat pin digital diberi keterangan 0-13, jadi untuk menggunakan pin analog menjadi *output* digital, pin analog yang pada keterangan *board* 0-5 kita ubah menjadi pin 14-19. dengan kata lain pin analog 0-5 berfungsi juga sebagai pin *output* digital 14-16 dan pada Tabel 2.2 menjelaskan Spesifikasi Arduino UNO R3.

Sifat *open source* arduino juga banyak memberikan keuntungan tersendiri untuk kita dalam menggunakan *board* ini, karena dengan sifat *open source* komponen yang kita pakai tidak hanya tergantung pada satu merek, namun memungkinkan kita bisa memakai semua komponen yang ada dipasaran. Bahasa pemrograman arduino merupakan bahasa C yang sudah disederhanakan *syntax* bahasa pemrogramannya sehingga mempermudah kita dalam mempelajari dan mendalami mikrokontroler.



Gambar 2.1 Board Arduino Uno R3

Tabel 2.2 Spesifikasi Arduino R3

Mikrokontroler	ATmega 328
Tegangan Pengoperasian	5 V
Batas tegangan yang disarankan	7 - 12 V
Batas tegangan Input	6 – 20V
Jumlah <i>pin I/O digital</i>	14 pin digital
Jumlah <i>pin input Analog</i>	6 pin
Arus DC tiap pin I/O	40 mA
Arus DC untuk pin 3,3 V	50 mA
<i>Memory Flash</i>	32 KB (ATmega 328) sekitar 0,5KB digunakan oleh bootloader
SRAM	2 KB (ATmega 328)
EPROM	1 KB (ATmega 328)
<i>ClocksPEED</i>	16 MHz

2.2.1 Daya (*Power*) dan Memori[2]

Arduino dapat diberikan *power* melalui koneksi USB atau *power supply*. *Powernya* diseleksi secara otomatis. *Power supply* dapat menggunakan adaptor DC atau baterai. Adaptor dapat dikoneksikan dengan mencolok *jack* adaptor pada koneksi port *input supply*. *Board* arduino dapat dioperasikan menggunakan *supply* dari luar sebesar 6 - 20 volt. Jika *supply* kurang dari 7V, kadangkala pin 5V akan menyuplai kurang dari 5 Volt dan *board* bisa menjadi tidak stabil. Jika menggunakan lebih dari 12 V, tegangan di regulator bisa menjadi sangat panas dan menyebabkan kerusakan pada *board*. Rekomendasi tegangan ada pada 7 sampai 12 Volt.

Penjelasan pada pin daya (*power*) adalah sebagai berikut :

- a. **Vin.** Tegangan *input* ke *board* arduino ketika menggunakan tegangan dari luar (seperti yang disebutkan 5 Volt dari koneksi USB atau tegangan yang diregulasikan). Pengguna dapat memberikan tegangan melalui pin ini, atau jika tegangan suplai menggunakan *power jack*, aksesnya menggunakan pin ini.
- b. **5V.** Regulasi *power supply* digunakan untuk *power* mikrokontroler dan komponen lainnya pada *board*. 5V dapat melalui *Vin* menggunakan regulator pada *board*, atau *supply* oleh USB atau *supply* regulasi 5V lainnya.

- c. **3V3.** Suplai 3.3Volt didapat oleh FTDI *chip* yang ada di *board*. Arus maximumnya adalah 50mA
- d. **Pin Ground** berfungsi sebagai jalur *ground* pada Arduino.
- e. **Memori.** ATmega328 memiliki 32 KB *flash* memori untuk menyimpan kode, juga 2 KB yang digunakan untuk *bootloader*. ATmega328 memiliki 2 KB untuk SRAM dan 1 KB untuk EEPROM.

2.2.2 Input & Output[2]

Setiap 14 pin digital pada arduino dapat digunakan sebagai *input* atau *output*, menggunakan fungsi *pinMode()*, *digitalWrite()*, dan *digitalRead()*. *Input/output* dioperasikan pada 5 Volt. Setiap pin dapat menghasilkan atau menerima maximum 40 mA dan memiliki internal *pull-up* resistor (*disconnected* oleh *default*) 20-50K Ohm.

Beberapa pin memiliki fungsi sebagai berikut :

- a. **Serial** : 0 (RX) dan 1 (TX). Digunakan untuk menerima (RX) dan mengirim (TX) TTL data serial. Pin ini terhubung pada pin yang koresponding dari USB ke TTL chip serial.
- b. **Interrupt eksternal** : 2 dan 3. Pin ini dapat dikonfigurasi untuk *trigger* sebuah interap pada *low value*, *rising* atau *falling edge*, atau perubahan nilai.
- c. **PWM** : 3, 5, 6, 9, 10, dan 11. Mendukung 8-bit *output* PWM dengan fungsi *analogWrite()*.
- d. **SPI** : 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Pin ini mensupport komunikasi SPI, yang mana masih mendukung *hardware*, yang tidak termasuk pada bahasa arduino.
- e. **LED** : 13. Ini adalah dibuat untuk koneksi LED ke digital pin 13. Ketika pin bernilai HIGH, LED hidup, ketika pin LOW, LED mati.

2.2.3 Komunikasi[2]

Arduino UNO R3 memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, Arduino lain, atau mikrokontroler lain. ATmega328 ini menyediakan UART TTL (5V) komunikasi serial, yang tersedia pada pin digital 0 (RX) dan 1 (TX). *Firmware* Arduino menggunakan USB *driver* standar COM, dan tidak ada *driver* eksternal yang dibutuhkan. Namun, pada Windows, file. Ini diperlukan. Perangkat lunak Arduino termasuk monitor serial yang memungkinkan data sederhana yang akan dikirim ke *board* Arduino. RX dan TX LED di *board* akan berkedip ketika data

sedang dikirim melalui chip USB-to-serial dan koneksi USB ke komputer.

2.3 Arduino Nano[3]

Arduino Nano yang ditunjukkan pada Gambar 2.2 adalah salah satu papan pengembangan mikrokontroler yang berukuran kecil, lengkap dan mendukung penggunaan *breadboard*. Arduino Nano diciptakan dengan basis mikrokontroler ATmega328 (untuk Arduino Nano versi 3.x) atau ATmega 168 (untuk Arduino versi 2.x). Arduino Nano kurang lebih memiliki fungsi yang sama dengan Arduino *Duemilanove*, tetapi dalam paket yang berbeda. Arduino Nano tidak menyertakan colokan DC berjenis *Barrel Jack*, dan dihubungkan ke komputer menggunakan *port* USB Mini-B. Arduino Nano dirancang dan diproduksi oleh perusahaan *Gravitech* yang yang spesifikasinya dijelaskan pada Tabel 2.3.



Tampak Depan



Tampak Belakang

Gambar 2.2 Arduino Nano

Tabel 2.3 Spesifikasi Arduino Nano

Mikrokontroler	Atmel ATmega168 atau ATmega328
Tegangan Operasi	5V
<i>Input Voltage</i> (disarankan)	7-12V
<i>Input Voltage</i> (limit)	6-20V
<i>Pin Digital I/O</i>	14 (6 pin digunakan sebagai <i>output</i> PWM)
<i>Pin Input Analog</i>	8
Arus DC per pin I/O	40 mA
<i>Flash Memory</i>	16KB (ATmega168) atau 32KB (ATmega328) 2KB digunakan oleh Bootloader
SRAM	1 KB (ATmega168) atau 2 KB (ATmega328)

EEPROM	512 byte (ATmega168) atau 1KB (ATmega328)
Clock Speed	16 MHz
Ukuran	1.85cm x 4.3cm

2.3.1 Daya (*Power*) dan Memori[3]

Arduino Nano dapat diaktifkan melalui koneksi USB Mini-B, atau melalui catu daya *eksternal* dengan tegangan belum teregulasi antara 6-20 Volt yang dihubungkan melalui pin 30 atau pin VIN, atau melalui catu daya *eksternal* dengan tegangan teregulasi 5 volt melalui pin 27 atau pin 5V. Sumber daya akan secara otomatis dipilih dari sumber tegangan yang lebih tinggi. Chip FTDI FT232L pada Arduino Nano akan aktif apabila memperoleh daya melalui USB, ketika Arduino Nano diberikan daya dari luar (Non-USB) maka Chip FTDI tidak aktif dan pin 3.3V pun tidak tersedia (tidak mengeluarkan tegangan), sedangkan LED TX dan RX pun berkedip apabila pin digital 0 dan 1 berada pada posisi *HIGH*.

ATmega168 memiliki 16 KB *flash memory* untuk menyimpan kode (2 KB digunakan untuk bootloader) Sedangkan ATmega328 memiliki *flash memory* sebesar 32 KB, (juga dengan 2 KB digunakan untuk bootloader). ATmega168 memiliki 1 KB *memory* pada SRAM dan 512 byte pada EEPROM (yang dapat dibaca dan ditulis dengan perpustakaan EEPROM) sedangkan ATmega328 memiliki 2 KB *memory* pada SRAM dan 1 KB pada EEPROM.

2.3.2 Input & Output[3]

Masing-masing dari 14 pin digital pada Arduino Nano dapat digunakan sebagai input atau output, dengan menggunakan fungsi *pinMode()*, *digitalWrite()*, dan *digitalRead()*. Semua pin beroperasi pada tegangan 5 volt. Setiap pin dapat memberikan atau menerima arus maksimum 40 mA dan memiliki resistor pull-up internal (yang terputus secara default) sebesar 20-50 KOhm. Selain itu beberapa pin memiliki fungsi khusus, yaitu:

- a. **Serial** : 0 (RX) dan 1 (TX). Digunakan untuk menerima (RX) dan mengirimkan (TX) TTL data serial. Pin ini terhubung ke pin yang sesuai dari chip FTDI USB-to-TTL Serial.
- b. **External Interrupt** (Interupsi Eksternal): Pin 2 dan pin 3 ini dapat dikonfigurasi untuk memicu sebuah interupsi pada nilai yang rendah, meningkat atau menurun, atau perubahan nilai.

- c. **PWM** : Pin 3, 5, 6, 9, 10, dan 11. Menyediakan output PWM 8-bit dengan fungsi *analogWrite()*. Jika pada jenis papan berukuran lebih besar (misal: Arduino Uno R3), pin PWM ini diberi simbol tilde atau “~” sedangkan pada Arduino Nano diberi tanda titik atau strip.
- d. **SPI** : Pin 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Pin ini mendukung komunikasi SPI. Sebenarnya komunikasi SPI ini tersedia pada *hardware*, tapi untuk saat belum didukung dalam bahasa Arduino.
- e. **LED** : Pin 13. Tersedia secara built-in pada papan Arduino Nano. LED terhubung ke pin digital 13. Ketika pin diset bernilai *HIGH*, maka LED menyala, dan ketika pin diset bernilai *LOW*, maka LED padam.

Arduino Nano memiliki 8 pin sebagai input analog, diberi label A0 sampai dengan A7, yang masing-masing menyediakan resolusi 10 bit (yaitu 1024 nilai yang berbeda). Secara default pin ini dapat diukur/diatur dari mulai Ground sampai dengan 5 Volt, juga memungkinkan untuk mengubah titik jangkauan tertinggi atau terendah mereka menggunakan fungsi *analogReference*. Pin Analog 6 dan 7 tidak dapat digunakan sebagai pin digital. Selain itu juga, beberapa pin memiliki fungsi yang dikhususkan, yaitu:

- f. **I2C** : Pin A4 (SDA) dan pin A5 (SCL). Yang mendukung komunikasi I2C (TWI) menggunakan perpustakaan *Wire*.

Masih ada beberapa pin lainnya pada Arduino Nano, yaitu:

- g. **AREF** : Referensi tegangan untuk input analog. Digunakan dengan fungsi *analog reference*.
- h. **RESET** : Jalur LOW ini digunakan untuk me-reset (menghidupkan ulang) mikrokontroler. Biasanya digunakan untuk menambahkan tombol *reset* pada shield yang menghalangi papan utama Arduino.

2.3.3 Komunikasi[3]

Arduino Nano memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, dengan Arduino lain, atau dengan mikrokontroler lainnya. Atmega168 dan Atmega328 menyediakan komunikasi serial UART TTL (5 Volt), yang tersedia pada pin *digital 0 (RX)* dan pin 1 (TX). Sebuah chip FTDI FT232RL yang terdapat pada papan Arduino Nano digunakan sebagai media komunikasi serial melalui USB dan driver FTDI (tersedia pada software Arduino IDE) yang akan menyediakan *COM Port*

Virtual (pada Device komputer) untuk berkomunikasi dengan perangkat lunak pada komputer. Perangkat lunak Arduino termasuk didalamnya serial monitor memungkinkan data tekstual sederhana dikirim dari papan Arduino. LED RX dan TX yang tersedia pada papan akan berkedip ketika data sedang dikirim atau diterima melalui chip FTDI dan koneksi USB yang terhubung melalui USB komputer (tetapi tidak untuk komunikasi serial pada pin 0 dan 1). Sebuah perpustakaan *SoftwareSerial* memungkinkan komunikasi serial pada beberapa pin digital Nano. Atmega168 dan Atmega328 juga mendukung komunikasi I2C (TWI) dan SPI. Perangkat lunak Arduino termasuk perpustakaan *Wire* digunakan untuk menyederhanakan penggunaan bus I2C. Untuk komunikasi SPI, silakan lihat datasheet Atmega168 atau Atmega328.

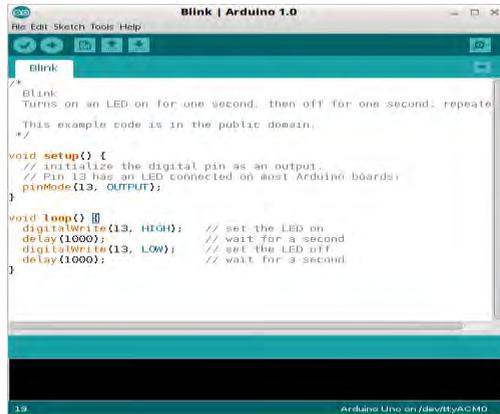
2.4 *Software Arduino*[4]

Arduino Uno R3 dapat diprogram dengan perangkat lunak Arduino. Pada ATmega328 di Arduino terdapat *bootloader* yang memungkinkan Anda untuk meng-*upload* kode baru untuk itu tanpa menggunakan *programmer hardware eksternal*. IDE Arduino adalah *software* yang sangat canggih ditulis dengan menggunakan Java.

IDE Arduino terdiri dari:

1. Editor program, sebuah window yang memungkinkan pengguna menulis dan mengedit program dalam bahasa *Processing*.
2. *Compiler*, sebuah modul yang mengubah kode program (bahasa *Processing*) menjadi kode biner. Bagaimanapun sebuah mikrokontroler tidak akan bisa memahami bahasa *Processing*. Yang bisa dipahami oleh mikrokontroler adalah kode biner. Itulah sebabnya *compiler* diperlukan dalam hal ini.
3. *Uploader*, sebuah modul yang memuat kode biner dari komputer ke dalam *memory* didalam papan Arduino.

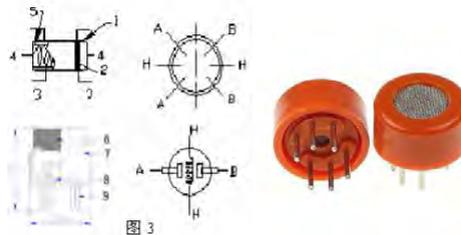
Sebuah kode program Arduino umumnya disebut dengan istilah *sketch*. Kata “*sketch*” digunakan secara bergantian dengan “kode program” dimana keduanya memiliki arti yang sama. Berikut adalah Gambar 2.3 Tampilan IDE Arduino dengan Sebuah *Sketch*.



Gambar 2.3 Tampilan IDE Arduino Dengan Sebuah *Sketch*

2.5 Sensor Gas Monoksida (MQ-7)[5]

Sensor gas MQ-7 pada Gambar 2.4 adalah sebuah sensor gas CO (Karbon Monoksida) yang cukup mudah penggunaannya. Sensor ini sangat cocok untuk mendeteksi gas CO dengan jangkauan pendeteksiannya mulai dari 10 sampai 10.000 ppm (*Part per Million*). Kemasan sensor MQ-7 tersedia dalam dua macam yaitu dari bahan metal dan plastic dan komponen lainnya yang ada pada sensor MQ-7 dijelaskan pada Tabel 2.4, Sensor ini memiliki sensitivitas yang tinggi dan waktu respon yang cepat. *Output* sensor berupa resistansi analog. Struktur dan konfigurasi sensor gas MQ-7 Pertama adalah material sensor yaitu *tin dioxide* (SnO_2). MQ-7 memiliki 6 pin, 4 pin yang digunakan untuk mengambil sinyal, dan 2 pin digunakan untuk memberikan pemanasan material sensor.



Gambar 2.4 Struktur Karbon Monoksida Sensor (MQ – 7)

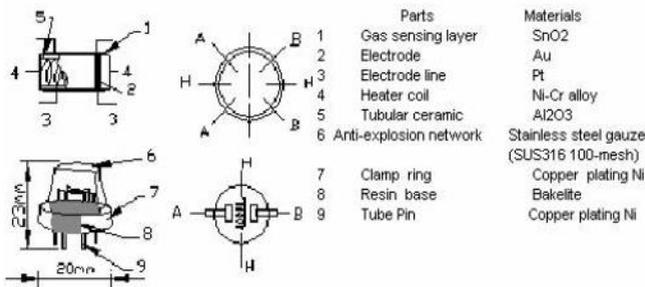
Tabel 2.4 Komponen Sensor MQ – 7

No	Parts	Materials
1	<i>Gas sensing layer</i>	snO ₂
2	<i>Electrode</i>	Au
3	<i>Electrode line</i>	Pt
4	<i>Heater coil</i>	Ni-cralloy
5	<i>Tubular ceramic</i>	Al ₂ O ₃
6	<i>Anti-explosion network</i>	Stainless steel gauze (sus316 100- mesh)
7	<i>Clamp ring</i>	Copper plating Ni
8	<i>Resin base</i>	Bakelite
9	<i>Tube pin</i>	Copper

2.6 Sensor Gas Metana (MQ-4)[6]

Sensor gas MQ-4 pada Gambar 2.5 memiliki sensitivitas tinggi untuk Metana, juga untuk Propane dan Butana. Sensor ini dapat digunakan untuk mendeteksi gas yang mudah terbakar terutama Metana. Konfigurasi karakter :

1. Sensitivitas yang baik untuk berbagai gas yang mudah terbakar
2. Sensitivitas tinggi untuk gas alam
3. Cocok untuk pemakaian jangka panjang dan biaya rendah
4. Drive sirkuit Sederhana.



Gambar 2.5 Struktur Gas Metana Sensor (MQ-4)

2.7 Wireless Modul RF 433 mHz[7]

Wireless modul RF 433 mHz yang ditunjukkan pada Gambar 2.6 ini harganya cukup terjangkau sehingga cocok digunakan untuk komunikasi menggunakan frekuensi radio (RF). Hal yang perlu diperhatikan jika menggunakan modul ini yaitu kerentanannya terhadap *noise* yang dapat mengganggu komunikasi. Untuk tegangan, modul ini cukup *fleksibel* dan dapat bekerja pada rentang tegangan 3-12V sehingga dapat dengan mudah digunakan pada tegangan 5V sesuai dengan yang digunakan oleh arduino pada umumnya untuk spesifikasi detailnya ada pada Tabel 2.5. Dan dapat menggunakan tegangan yang lebih tinggi jika ingin meningkatkan jangkauan modul ini, tentunya dengan tambahan beberapa komponen diskrit agar tidak merusak arduino yang digunakan.



Gambar 2.6 Transmitter dan Receiver RF 433 mHz

Tabel 2.5 Spesifikasi Transmitter RF 433 mHz

Transmitter	
Kerja Tegangan	3V – 12V. Penggunaan daya 12V
Kerja saat	max Kurang dari 40mA max, dan min 9mA
Resonansi Modus	(SAW)
Modus Modulation	ASK
Kerja Frekuensi	Eve 315MHz Atau 433MHz
Transmisi Daya	25mW (315MHz di 12V)
Frekuensi Eror	150 kHz (max)
Velocity	kurang dari 10kbps

Tabel 2.6 Spesifikasi *Receiver* RF 433 mHz

Receiver	
Bekerja Tegangan	5.0VDC + 0.5V
Bekerja Saat Ini	≤5.5mA max
Metode Kerja	OOK / ASK
Frekuensi Kerja	315MHz-433.92MHZ
<i>Bandwidh</i>	2MHz
<i>Sensitivitas</i>	excel -100dBm (50Ω)
Mengirimkan Kecepatan	<9.6Kbps (di 315MHz dan -95dBm)

2.8 Modul *Relay*[7]

Modul *Relay* 5V dengan 2 *channel* output pada Gambar 2.7 dapat digunakan sebagai saklar elektronik untuk mengendalikan perangkat listrik yang memerlukan tegangan dan arus yang besar. *Kompatible* dengan semua mikrokontroler (khususnya Arduino, 8051, 8535, AVR, PIC, DSP, ARM, ARM, MSP430, TTL logic). *Relay* 2 Channel ini memerlukan arus sebesar sekurang-kurangnya 15-20 mA untuk mengontrol masing-masing *channel*. Disertai dengan *relay high-current* sehingga dapat menghubungkan perangkat dengan AC250V 10A. Jika menggunakan mikrokontroler dengan tegangan kerja 3.3V, anda tetap dapat menggunakan *Relay 2 channel* ini dengan cara :

1. Lepas jumper *JD-VCC*
2. Hubungkan *JD-VCC* dengan *external power* 5V lainnya.

Spesifikasi :

1. *Number of Relays* : 2
2. *Control signal* : TTL level (*ACTIVE LOW*)
3. *Rated load* : 10A 250VAC, 10A 30VDC, 10A 125VAC, 10A 28VDC
4. *Contact action time* : 10ms/5ms
5. *Indicator LED* for each *channel*

6. *Relay size : 51 x 41 mm*



Gambar 2.7 Modul Relay 2 channel 5V

2.9 *Liquid Crystal Display (LCD)*[8]

Liquid Crystal Display (LCD) pada Gambar 2.8 adalah suatu *display* dari bahan cairan kristal yang pengoperasiannya menggunakan sistem dot matriks. *LCD* banyak digunakan sebagai *display* dari alat-alat elektronik seperti kalkulator, *multitester digital*, jam digital, dan sebagainya. *LCD* yang dipakai pada pembuatan sistem monitor dan pengontrol gas CO dan CH₄ ini adalah 2 baris x 16.



Gambar 2.8 *Liquid Crystal Display (LCD)*

2.10 *Exhaust Fan*[8]

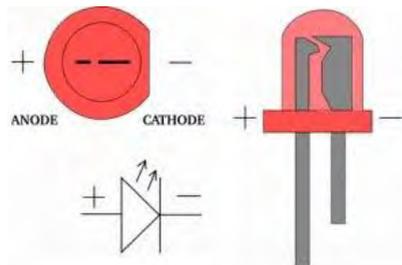
Exhaust fan pada Gambar 2.9 berfungsi untuk menghisap udara di dalam ruang untuk dibuang ke luar, dan pada saat bersamaan menarik udara segar di luar ke dalam ruangan. Selain itu *exhaust fan* juga bisa mengatur *volume* udara yang akan disirkulasikan pada ruang. Supaya tetap sehat ruang butuh sirkulasi udara agar selalu ada pergantian udara dalam ruangan dengan udara segar dari luar ruangan. *Exhaust fan* merupakan salah satu jenis kipas angin yang difungsikan untuk sirkulasi udara dalam ruang. Oleh karena itu, peletakkannya diantara *indoor* dan *outdoor*.



Gambar 2.9 Exhaust Fan

2.11 LED (*Light Emitting Diode*)[8]

Light Emitting Diode (LED) pada Gambar 2.10 adalah komponen elektronika yang dapat memancarkan cahaya monokromatik ketika diberikan tegangan maju. LED merupakan keluarga Dioda yang terbuat dari bahan semikonduktor. Warna-warna Cahaya yang dipancarkan oleh LED tergantung pada jenis bahan semikonduktor yang dipergunakannya. LED juga dapat memancarkan sinar inframerah yang tidak tampak oleh mata seperti yang sering kita jumpai pada *Remote Control TV* ataupun *Remote Control* perangkat elektronik lainnya. Bentuk LED mirip dengan sebuah bohlam (bola lampu) yang kecil dan dapat dipasangkan dengan mudah ke dalam berbagai perangkat elektronika. Berbeda dengan Lampu Pijar, LED tidak memerlukan pembakaran filamen sehingga tidak menimbulkan panas dalam menghasilkan cahaya.



Gambar 2.10 LED (*Light Emitting Diode*)

2.12 Buzzer[8]

Buzzer pada Gambar 2.11 menggunakan lilitan dan kontaktor. Saat lilitan dialiri arus, maka akan timbul medan magnet. Medan magnet akan menarik kontaktor, sehingga kontaktor mengenai lempengan metal yang lain, sehingga menimbulkan suara. Pada saat yang sama kontaktor tidak menempel pada metal *contact*, sehingga arus yang mengalir ke lilitan terputus. Dengan terputusnya arus, maka lilitan tidak menghasilkan medan magnet, sehingga kontaktor kembali ke metal *contact*. Arus kembali mengalir ke lilitan, sehingga timbul medan magnet, dan kontaktor kembali tertarik ke lempengan metal. Hal ini terjadi terus menerus selama arus mengalir ke lilitan.



Gambar 2.11 Buzzer 5 Volt

2.13 Baterai[9]

Baterai listrik adalah alat yang terdiri dari 2 atau lebih sel elektrokimia yang mengubah energi kimia yang tersimpan menjadi energi listrik. Tiap sel memiliki kutub positif (katoda) dan kutub negatif (anoda). Kutub yang bertanda positif menandakan bahwa memiliki energi potensial yang lebih tinggi daripada kutub bertanda negatif. Kutub bertanda negatif adalah sumber elektron yang ketika disambungkan dengan rangkaian *eksternal* akan mengalir dan memberikan energi ke peralatan eksternal. Ketika baterai dihubungkan dengan rangkaian eksternal, elektrolit dapat berpindah sebagai ion didalamnya, sehingga terjadi reaksi kimia pada kedua kutubnya. Perpindahan ion dalam baterai akan mengalirkan arus listrik keluar dari baterai sehingga menghasilkan kerja. Meski sebutan *baterai* secara teknis adalah alat dengan beberapa sel, sel tunggal juga umumnya disebut baterai. Baterai primer (satu kali penggunaan) hanya digunakan sekali dan dibuang, material elektrodanya tidak dapat berkebalikan arah ketika dilepaskan. Penggunaannya umumnya adalah baterai alkaline digunakan untuk senter dan berbagai alat *portable*

lainnya. Baterai sekunder (Baterai dapat diisi ulang) dapat digunakan dan diisi ulang beberapa kali, komposisi awal elektroda dapat dikembalikan dengan arus berkebalikan.

BAB III PERANCANGAN SISTEM

Pada bab ini akan dibahas mengenai perancangan perangkat keras dan lunak yang dilakukan dengan metode penelitian berdasarkan pada studi kepustakaan berupa data-data literatur dari masing-masing komponen, informasi dari internet, dan konsep-konsep teoretis dari buku-buku penunjang. Dari data-data yang diperoleh maka dilakukan perencanaan rangkaian perangkat keras. Dalam perangkat keras ini, penulis akan melakukan pengujian perangkat keras dengan program-program yang telah dibuat, pembuatan perangkat lunak adalah tahap selanjutnya. Terakhir adalah penggabungan perangkat keras dengan kerja perangkat lunak yang telah selesai dibuat. Perencanaan ini diperlukan sebelum proses pembuatan sistem tersebut, Perancangan ini berguna agar pengerjaan tahapan selanjutnya berjalan dengan lancar. Tahapan-tahapannya meliputi tahap pembuatan perangkat keras, perangkat lunak dan menggabungkan keduanya.

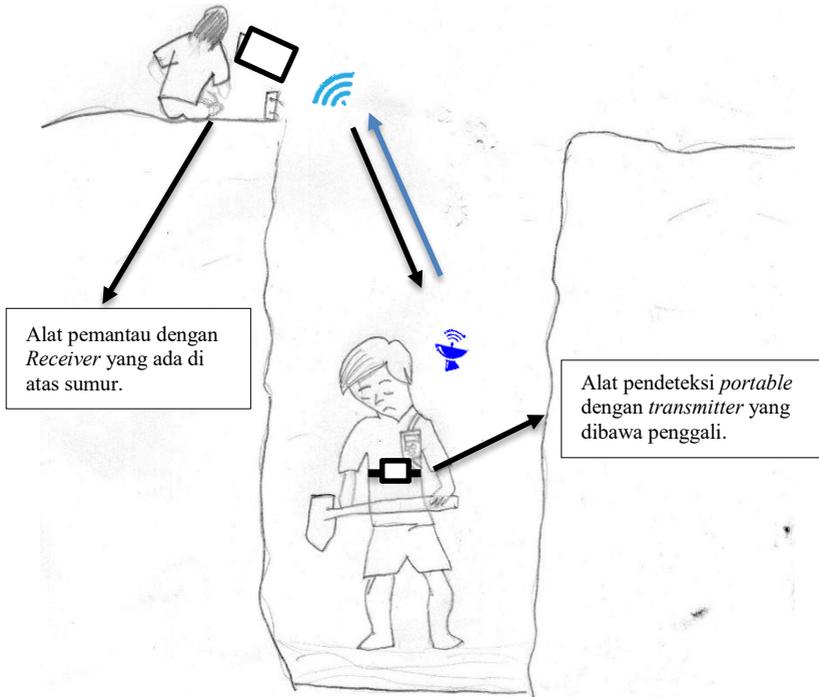
Untuk memudahkan dalam pembahasan bab ini akan dibagi menjadi dua bagian yaitu :

1. Perancangan *hardware* (perangkat keras) yang terdiri dari pembahasan Arduino Uno R3 dan arduino Nano, perancangan perangkat keras kontrol, konfigurasi *gas sensor* dengan Arduino Nano, rangkaian *wireless* ke Arduino Uno.
2. Perancangan *software* (perangkat lunak) meliputi perancangan pembuatan program pembacaan data sensor ke LCD, perancangan program pengiriman data menggunakan RF 433 mHz.

3.1 Perancangan Perangkat Keras

Input sistem ini adalah gas beracun CO dan CH₄. *Array sensor* akan mendeteksi apakah gas tersebut mengandung gas beracun seperti CH₄ dan CO. Hasil yang didapatkan *array sensor* akan dikirimkan ke Arduino menggunakan komunikasi TTL untuk sensor CH₄ dan CO untuk kemudian diproses oleh Arduino. Data hasil pembacaan sensor akan diproses dan dikirimkan melalui komunikasi data melalui RF 433 mHz. *Wireless (Transmitter)* akan mengirimkan informasi ini ke *wireless (receiver)* yang terhubung pada LCD. Di LCD, informasi tersebut akan di tampilkan berapa ppm gas yang terdeteksi. Berikut Gambar 3.1 Sketsa

alat yang di pakai oleh penggali sumur dan alat pemantau yang ada di atas sumur.



Gambar 3.1 Sketsa Alat

3.2 Spesifikasi Alat

Pada rangkaian sistem monitoring gas beracun pada galian sumur, spesifikasi alatnya yaitu pada Tabel 3.1.

Tabel 3 .1 Spesifikasi Alat

No	Komponen	Tipe	Tegangan Yang Digunakan
1	Sensor gas CO	MQ-7	5 V

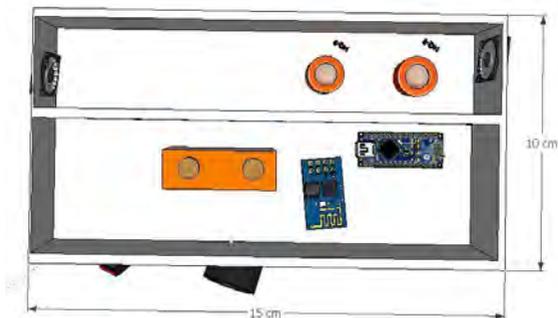
No	Komponen	Tipe	Tegangan Yang Digunakan
2	Sensor gas CH ₄	MQ-4	5 V
3	Arduino Uno R3	Atmega328	7 – 12 V
4	Arduino Nano	Atmega328	7 – 12 V
5	Module <i>Relay</i>	SONGLE	5V
6	LCD	2 baris x 16	5 V
7	<i>Exhaust Fan</i>	M25E-6	5 V
8	<i>Wireless</i>	RF 433 mHz	5 V

3.3 Perancangan *Hardware* Pendeteksi Gas Beracun di Sumur

Perancangan pembuatan *hardware monitoring* gas beracun di dalam sumur ini akan dibagi menjadi 2 bagian, yaitu pembuatan *hardware kontrol wearable* dan *hardware kontrol pemantau*. Untuk penjelasan lebih jelasnya akan dibahas pada subbab berikut ini.

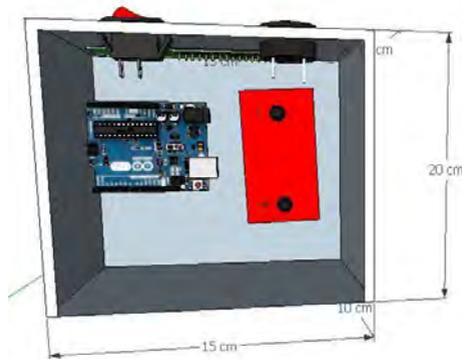
3.3.1 Perancangan *Hardware Kontrol*

Perancangan *hardware kontrol* ini sebagai tempat atau wadah dari beberapa rangkaian kontroler yang disusun dalam satu kesatuan, agar *interface* dari rangkaian tersebut terlihat lebih minimalis dan rapi. Di dalam *hardware kontrol wearable* terdiri dari *fan*, sensor MQ-7, MQ-4, arduino nano, *buzzer*, led, baterai dan *wireless*. Untuk *Hardware kontrol pemantau* terdiri dari *wireless*, arduino Uno R3, *buzzer*, led dan LCD.



Gambar 3 .2 Rancangan *Hardware Kontrol Wearable*

Gambar 3.2. menjelaskan Rancangan *hardware* kontrol *wearable* yang dibawa oleh penggali sumur yang terdiri dari Fan, Sensor MQ-7 dan MQ-4, Arduino Nano, *Buzzer*, Led, *Wireless*, Baterai dan juga tombol *on/off*. Dimana sebelum menggali sumur *power* di on kan kemudian jika terdeteksi gas CO atau gas CH₄ maka akan diproses oleh arduino nano dan menyalakan *buzzer* dan led, lalu hasilnya akan dikirim ke pemantau yang ada diatas dengan *wireless*.



Gambar 3.3 Rancangan *Hardware* Kontrol Pemantau

Gambar 3.3. di atas menjelaskan Rancangan *hardware* kontrol pemantau yang di letakkan di atas dekat galian sumur, rangkaian ini terdiri dari *Wireless*.Arduino Uno R3, *Buzzer*, LCD,baterai daan juga tombol *on/off*. Dimana saat melakukan penggalian tombol *power* di on kan kemudian jika *wireless* menerima data dari *wireless* pengirim maka akan langsung diproses oleh arduino Uno R3 dan menyalakan *buzzer* dan ditampilkan pada LCD.

3.4 Arduino

Pengontrol yang digunakan pada alat ini adalah Arduino Atmega328, dipilihnya chip ini karena fitur – fitur yang dimiliki cukup lengkap dan simple, selain itu juga memiliki kecepatan yang lebih baik yaitu satu siklus mesin untuk satu instruksi dengan kecepatan hingga 16MHz serta banyak tersedia dipasaran. Untuk dapat bekerja, mikrokontroler perlu mendapat tegangan kerja sebesar 9 Volt dengan arus 40 mA dan *ground* serta *clock*, dengan *clock* yang digunakan pada perancangan ini sebesar 16 MHz.

3.5 Perancangan Perangkat Lunak

Perangkat lunak yang digunakan dalam pemrograman sensor adalah Arduino IDE.

3.5.1 Antarmuka UART TTL

Antarmuka UART TTL selain digunakan untuk komunikasi serial juga digunakan untuk mengubah alamat *I2C* pada pengerjaan Tugas Akhir kali ini. Karena penulis menggunakan dua modul sensor maka alamat dari tiap sensor harus dibedakan untuk bisa berkomunikasi, berikut parameter yang digunakan untuk mengubah alamat UART TTL :

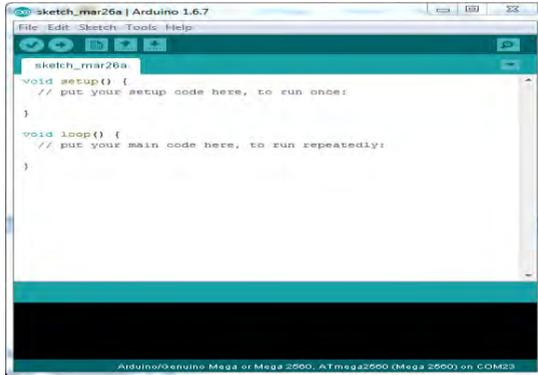
1. *Baudrate 38400 bps*
2. *8 bit data*
3. *1 bit stop*
4. *Tanpa bit parity*
5. *Tanpa flow control*

Untuk mengubah alamat UART TTL pin TX Arduino harus terhubung ke pin RX modul gas sensor dan pin RX Arduino terhubung dengan pin TX dari modul gas sensor.

Kode program antarmuka UART untuk mengganti alamat dari pin RX (0) dan TX (1) menjadi pin 6 dan 7. Sebelum memasukkan program tersebut kita diharuskan memasukkan *library Software serial*, *library* tersebut berguna untuk merekayasa alamat agar data tidak saling bertabrakan pada Arduino.

3.5.2 Program Arduino IDE

Pada tahap perancangan program di *software* Arduino, program yang di buat meliputi program mengukur Sensor gas MQ-4 dan MQ-7, dan program pengiriman data menggunakan RF 433 mHz..

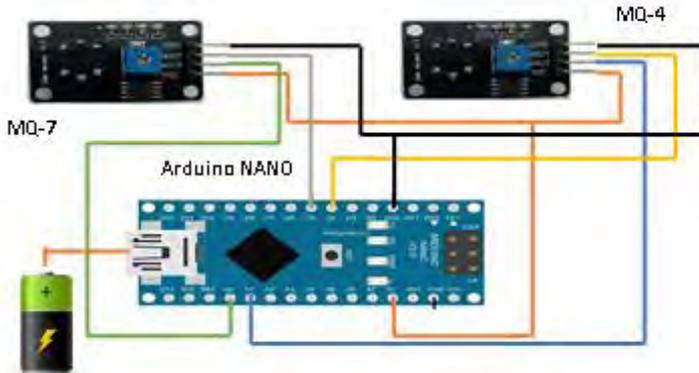


Gambar 3.4 Tampilan Awal *Software* Arduino

Dari Gambar 3.4 dapat dilihat bahwa tampilan awal *software* Arduino masih belum terisi program. Sebelum menjelaskan tentang bagaimana cara memprogram dengan *software* Arduino maka terlebih dahulu dijelaskan beberapa simbol yang sering digunakan, yaitu :

1.  Merupakan simbol *Verify*. Simbol ini berfungsi memferivikasi program, apakah program yang dibuat sudah benar atau masih terdapat eror.
2.  Merupakan simbol *Upload*. Simbol ini berfungsi mengupload program dari *software* ke Arduino.
3.  Merupakan simbol *New*. Simbol ini berfungsi untuk membuat sebuah projek baru.
4.  Merupakan simbol *Open*. Simbol ini berfungsi untuk membuka sebuah file atau projek program yang sudah pernah dibuat.
5.  Merupakan simbol *Save*. Simbol ini berfungsi menyimpan *project* program yang sudah dibuat.

3.5.3 AntarMuka Muka Sensor MQ-4 dan MQ-7



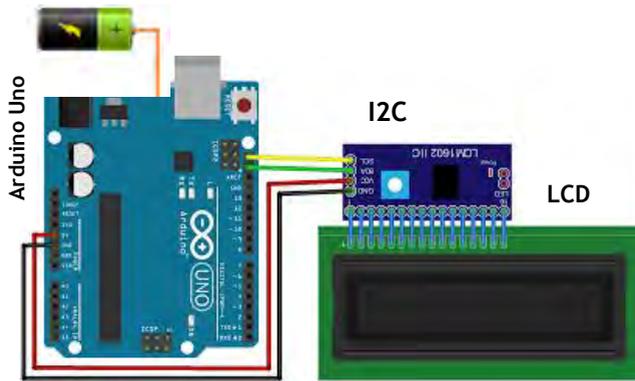
Gambar 3.5 Perancangan Sensor Ke Mikrokontroler (*hardware wearable*)

Keterangan :

-  = *VCC*
-  = *Ground*
-  = *Analog Out MQ-7*
-  = *Analog Out MQ-4*
-  = *Digital Out MQ-4*
-  = *Digital Out MQ-7*

Gambar 3.5 di atas menjelaskan modul sensor MQ-7 memiliki antarmuka UART TTL yang akan menghubungkan pin *Aout* dan *Dout* sebagai pin keluaran sensor. Pin *Aout* dihubungkan ke A0 pada arduino Nano sedangkan pin *Dout* dihubungkan ke D5 arduino Nano. Untuk sensor MQ-4 juga memiliki pin *Aout* dan *Dout*. Kedua pin ini dihubungkan pada A1 untuk pin *Aout* sensor dan *Dout* dihubungkan D4 arduino Nano. Data alamat disimpan di *EEPROM* sehingga tidak akan hilang saat tidak ada daya. Rangkaian ini di pasang di alat pendeteksi gas beracun *portable* yang di bawa oleh penggali sumur.

3.5.4 Antarmuka pada *Hardware* Kontrol Pemantau



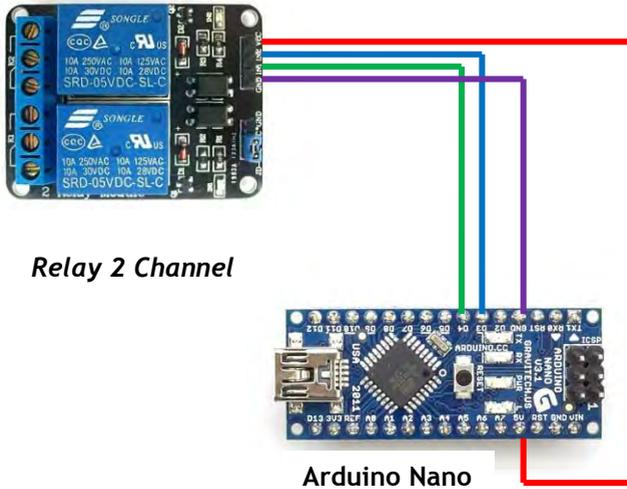
Gambar 3.6 Antarmuka pada *Hardware* Kontrol Pemantau

Keterangan :

-  = *Vcc*
-  = *Ground*
-  = *Sda*
-  = *Scl*

Gambar 3.6 di atas menjelaskan arduino Uno yang dipasangkan dengan *I2C* dan juga *LCD*, rangkaian ini akan di letakkan pada alat pemantau yang ada di atas sumur guna *memonitoring* adanya kadar gas beracun yang ada di dalam sumur. Dari skema Gambar 3.6 *LCD I2C* telah dihubungkan dengan arduino Uno , pin *SCL* yaitu kabel berwarna kuning pada *I2C* masuk ke pin *SDA* kabel berwarna hijau pada *I2C* masuk ke pin *SDA* arduino Uno pada bagian atas setelah pin *SCL* dan yang terakhir pin *VCC* kabel berwarna merah masuk ke pin *5V* serta pin *GND* kabel berwarna hitam masuk ke pin *GND* arduino Uno.

3.5.5 Perancangan Mikrokontroler dan *Relay*



Gambar 3.7 Perancangan *Relay* ke Mikrokontroler

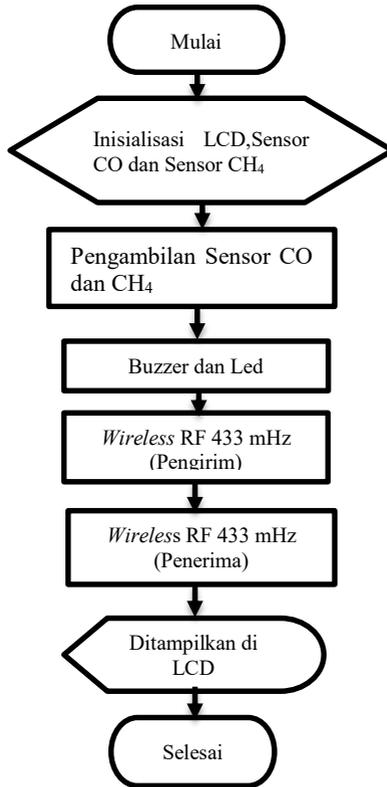
Keterangan :

-  = *Vcc*
-  = *Ground*
-  = *In 1*
-  = *In 2*

Rangkaian Gambar 3.7 menjelaskan arduino nano yang di pasang dengan *relay 2 channel* yang berfungsi untuk mengatur jalannya *exhaust fan* agar berjalan dengan baik, rangkaian ini akan di letakkan pada alat pendeteksi gas beracun *portable*. *Relay 2 channel* di hubungkan dengan arduino Nano yaitu kabel warna merah *VCC* dihubungkan ke pin 5V, kabel warna ungu ke *ground*, kabel warna biru *In 1* dihubungkan ke pin 3 dan warna hijau *In 2* dihubungkan ke pin 4.

3.6 Flowchart Program Mikrokontroler

Pada Gambar 3.8 merupakan Gambaran *flowchart* dari program yang terdapat pada mikrokontroler.



Gambar 3.8 *Flowchart* Mikrokontroler

3.7 Pembacaan Nilai Sensor Gas Metana (MQ-4)

Gambar 3.9 adalah cuplikan program inisialisasi dari sensor gas Metana (MQ-4), alur program pertama yang akan dijalankan adalah inisialisasi dari *aeey sensor* dan *delay* atau jeda selama 10 detik untuk memanaskan *heater* dari masing – masing yang ditampilkan pada *liquid crystal display* (LCD).

```

const int gasPin1 = A1;
const int gasPin2 = A0;

void setup()
{
  Serial.begin(9600);
}

```

Gambar 3.9 Kode Program Inisialisasi

Kode program pembacaan sensor CO ditampilkan pada Gambar 3.10, setelah proses jeda ini selesai selanjutnya LCD akan di kosongkan untuk kemudian berjalan kode program pembacaan sensor CH₄ menggunakan komunikasi serial UART TTL.

```

void loop()
{
  Serial.print("CO : ");
  Serial.println(analogRead(gasPin1));
  Serial.print("CH4 : ");
  Serial.println(analogRead(gasPin2));
  delay(300);
}

```

Gambar 3.10 Kode Progam Pembacaan Sensor CH₄

3.8 Pembacaan Nilai Sensor Gas Karbon Monoksida (MQ-7)

Gambar 3.11 adalah cuplikan program inisialisasi dari sensor gas Karbon Monoksida (MQ-7), Alur program pertama yang akan dijalankan adalah inisialisasi dari *aeey sensor* dan *delay* atau jeda selama 60 detik untuk memanaskan *heater* dari masing – masing yang ditampilkan pada *liquid crystal display*. Pada sensor gas MQ-7 akan menggunakan variable sensor sebagai pembacaan serialnya. Karena diawal sudah digunakan *software.serial* yang memungkinkan pin RX dan TX dirubah penempatannya. Lalu kita atur *beudrate* menjadi 38400 karena itu sudah menjadi parameter untuk komunikasi serial UART TTL.

```

const int gasPin1 = A1;
const int gasPin2 = A0;

void setup()
{
  Serial.begin(9600);
}

```

Gambar 3.11 Kode Program Inisialisasi

Kode program pembacaan sensor CO ditampilkan pada Gambar 3.12, setelah proses jeda ini selesai selanjutnya LCD akan di kosongkan untuk kemudian berjalan kode program pembacaan sensor CO menggunakan komunikasi serial UART TTL.

```

void loop()
{
  Serial.print("CO : ");
  Serial.println(analogRead(gasPin1));
  Serial.print("CH4 : ");
  Serial.println(analogRead(gasPin2));
  delay(300);
}

```

Gambar 3.12 Kode Program Pembacaan Sensor Gas MQ-7

3.8.1 Menampilkan Data Pada LCD

Pada Gambar 3.13 menampilkan kode program untuk menampilkan hasil pembacaan sensor pada *LCD*, setelah semua sensor telah dibaca maka langkah selanjutnya adalah menampilkan data pada LCD.

```

void loop()
{
    nilai = baca();
    nilai2 = baca2();
    lcd.home();
    sprintf(kata, "CO = %d  ", nilai);
    lcd.print(kata);
    lcd.setCursor(11, 0);
    lcd.print("ppm ");
    lcd.setCursor(0, 1);
    sprintf(kata2, "CH4 = %d  ", nilai2);
    lcd.setCursor(11, 1);
    lcd.print("ppm ");
    delay(1000);
}

```

Gambar 3.13 Kode Program Menampilkan Hasil Pembacaan Sensor

3.9 Inisialisasi *Wireless* RF 433 mHz

Untuk memulai komunikasi *serial* pada arduino, terlebih dahulu lakukan inisialisasi serial yang akan digunakan dan kecepatan yang diinginkan. Pastikan bahwa kecepatan yang digunakan sama dengan kecepatan perangkat yang akan terhubung dengan arduino. Caranya melakukan inisialisasi *serial* port adalah dengan memanggil fungsi *Serial.begin()* dengan memasukkan parameter kecepatan komunikasi serial yang diinginkan pada fungsi *setup()* seperti Gambar 3.14.

```

void setup()
{
    Serial.begin(9600);    // inisialisasi komunikasi serial
    dengan kecepatan 9600bps
}

```

Gambar 3.14 Inisialisasi RF 433 mHz

Setelah proses *inisialisasi* selesai, dapat melakukan komunikasi serial dengan menggunakan beberapa fungsi yang tersedia dan memanggil fungsi yang akan digunakan di dalam fungsi *loop()* pada sketch. Berikut ini adalah contoh memanfaatkan fungsi *Serial.available()* untuk memeriksa apakah ada data yang diterima oleh arduino yang ditunjukkan pada Gambar 3.15.

```

void loop()
{
  if (Serial.available() > 0)    // Serial.available() adalah
  fungsi untuk memeriksa apakah ada data yang diterima
  {
    char s = Serial.read();      // Serial.read() adalah fungsi
    untuk membaca karakter yang diterima
    Serial.print(s);            // Serial.print() adalah
    fungsi yang digunakan untuk mengirimkan karakter/kalimat (string)
                                // jika ingin menambahkan
                                baris baru, maka gunakan fungsi Serial.println()
  }
}

```

Gambar 3.15 Komunikasi *Serial*

Untuk berinteraksi dengan arduino, dapat menggunakan fitur *Serial Monitor* pada IDE arduino.

BAB IV

PENGUJIAN DAN ANALISA

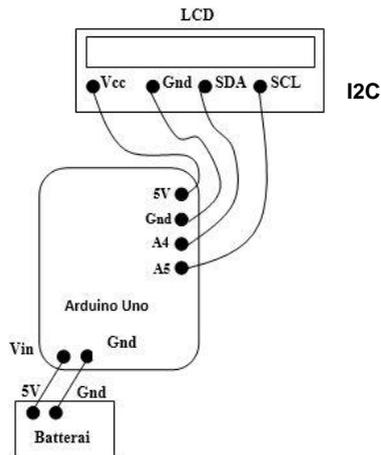
Pengujian sistem yang dilakukan merupakan pengujian terhadap perangkat keras dan perangkat lunak dari sistem secara keseluruhan yang telah selesai dibuat untuk mengetahui komponen-komponen sistem apakah berjalan dengan baik dan sesuai yang diharapkan. Terdapat beberapa pengujian sistem, antara lain:

1. Pengujian *Liquid Crystal Display* (LCD)
2. Pengujian modul sensor CO
3. Pengujian modul sensor CH₄
4. Pengujian *Wireless RF* 433 mHz

4.1 Pengujian *Liquid Crystal Display* (LCD)

Pengujian LCD ini bertujuan agar dapat diketahui *liquid crystal display* yang digunakan bisa berfungsi dengan baik. Pada alat ini kita menggunakan *multiplexer I2C* yang dihubungkan dengan LCD.

Pin SCL pada *multiplexer* dihubungkan ke pin A5 pada Arduino, sedangkan pin SDA pada *multiplexer* dihubungkan dengan pin A4 pada Arduino Seperti Rangkaian pada Gambar 4.1.



Gambar 4.1 Rangkaian Pengujian LCD

Untuk pengujiannya kita masukkan kode program seperti Gambar 4.2 berikut.

```
LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);  
void setup()  
{  
  lcd.begin(16,2);           // initialize the lcd  
  lcd.home ();              // go home  
  lcd.print("TESTING LCD");  
}
```

Gambar 4.2 Program Pengujian *Liquid Crystal Display*

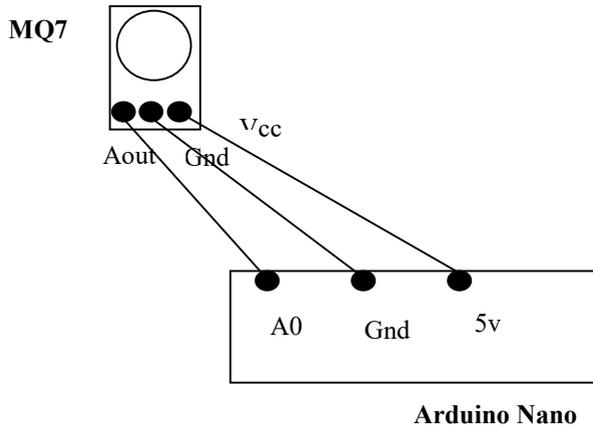
Maka hasil dari kode program tersebut kita bisa lihat pada Gambar 4.3, dengan melihat bahwa LCD yang digunakan bisa berjalan dengan baik.



Gambar 4.3 Hasil Pengujian

4.2 Pengujian Modul Sensor MQ-7 (CO)

Sensor MQ-7 merupakan sensor yang memiliki kepekaan yang baik terhadap gas Karbon Monoksida dan hasil kalibrasinya pun cukup stabil. Dalam penggunaannya, sensor ini membutuhkan waktu kondisi kerja atau pemanasan dari alat pemanas (*heater*). Cara pengujian alat pemanas (*Heater*) yaitu hubungkan pin sensor gas MQ-7 ke arduino Nano seperti Gambar 4.4, kemudian sambungkan arduino Nano ke laptop, buka *serial monitor* pada pemrograman arduino sehingga kita dapat melihat keluaran nilai sensor gas MQ-7, nyalakan *stopwatch* lalu tunggu hingga angka pada *serial monitor* stabil jika sudah stabil maka pemanasan sensor gas dianggap sudah selesai. Waktu pemanasan rata-rata yang diperlukan agar komponen sensitif dapat bekerja berdasarkan pengujian yang telah dilakukan ditunjukkan pada Tabel 4.1.



Gambar 4.4 Rangkaian Pengujian Sensor MQ-7

Setelah dirangkai seperti gambar di atas maka di buatlah program di *software* IDE Seperti Gambar 4.5.

```

Station_FIX
void loop()
{
  MQ7GetData(); //Mendapatkan data MQ7
  MQ4GetData(); //Mendapatkan data MQ4

  Serial.print("MQ7: ");
  Serial.println(dataPFM[0]);
  Serial.print("MQ7: ");
  Serial.println(analogRead(1));
  Serial.print("MQ4: ");
  Serial.println(dataPFM[1]);
  Serial.print("MQ4: ");
  Serial.println(analogRead(0));

  sprintf(buff, "CO : %d      ", dataPFM[0]); //analogRead(1));
  send(buff);
  //delay(500);
  //Serial.println(buff);
  sprintf(buff, "CH4 : %d      ", dataPFM[1]); //analogRead(0));
  send(buff);
  //delay(500);
  //Serial.println(buff);
}

```

Gambar 4.5 Program Pembacaan Sensor CO

Tabel 4.1 Pengujian Alat Pemanas (*Heater*) Sensor MQ-7

No. Pengujian	Waktu Pemanasan
Pengujian 1	154 detik
Pengujian 2	70 detik
Pengujian 3	63 detik
Pengujian 4	67 detik
Pengujian 5	60 detik
Waktu Pemanasan Rata – Rata	82,8 detik

Sensor gas MQ-7 sudah dilengkapi dengan ADC resolusi 10 bit dalam modulnya, sehingga data yang dikirimkan ke mikrokontroler Arduino sudah dalam bentuk digital. Untuk mengetahui apakah kalibrasi sensor MQ-7 sudah benar atau sama dengan hasil kalibrasi alat ukur gas CO lainnya, maka perlu dibandingkan dengan alat ukur gas CO lainnya. Pada Gambar 4.6 adalah alat ukur yang dipakai yaitu ECOM yang mampu mengukur kadar gas CO dengan teliti. Dan Tabel 4.2 adalah data hasil kalibrasi sensor pada asap kendaraan bermotor (knalpot) di Balai Hiperkes & keselamatan kerja Jatim dan alat kalibrasi ECOM, data pengujian dapat dilihat pada lampiran B-19.



Gambar 4.6 Alat ECOM

Tabel 4.2 Hasil Pembacaan Alat ECOM dan data ADC sensor MQ-7

Pengukuran	Pembacaan ECOM (PPM)	Pembacaan Alat Pendeteksi Gas Beracun	Error
1	5309	471	6,065%

Pengukuran	Pembacaan ECOM (PPM)	Pembacaan Alat Pendeteksi Gas Beracun	Eror
2	3458	376	6,159%
3	3352	371	5,698%
4	2834	244	6,492%

Dari data Tabel 4.2 dimasukkan ke *Microsoft excel* untuk mencari linieritasnya. Dengan rumus *intercept* dan *slope* bisa ditemukan bahwa untuk mengkalibrasi sensor CO dengan persamaan dibawah ini :

$$x = 0,077011y + 77,61299$$

dimana hal ini.....(4.1)

x = Nilai hasil kalibrasi (ppm)

y = Data ADC yang dibaca

Pengujian kedua pada Tabel 4.3 menggunakan sumber asap pembakaran kertas yaitu dengan cara mendekatkan alat pendeteksi gas beracun *portable* pada asap yang keluar pada pembakaran kertas hingga muncul kadar gas beracun pada LCD alat pemantau.

Tabel 4.3 Hasil Pembacaan Sensor Gas CO

PPM	ADC
0	62
2	70
5	80
5	79
8	92
9	95
21	100
23	105

Dari data Tabel 4.3 dimasukkan ke *Microsoft excel* untuk mencari linieritasnya. Dengan rumus *intercept* dan *slope* bisa ditemukan bahwa untuk mengkalibrasi sensor CO dengan persamaan dibawah ini :

$$x = 0,586299y - 41,5845$$

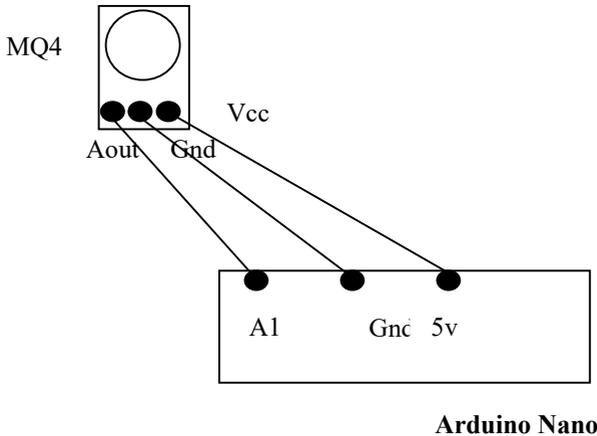
dimana hal ini.....(4.2)

x = Nilai hasil kalibrasi (ppm)

y = Data ADC yang dibaca

4.3 Pengujian Modul Sensor MQ-4 (CH₄)

Berdasarkan pengujian, sensor MQ-4 memiliki kepekaan yang baik terhadap gas *metane*. Sensor MQ-4 juga membutuhkan waktu kondisi kerja atau pemanasan dari alat pemanas (*heater*) dalam penggunaannya. Cara pengujian alat pemanas (*heater*) yaitu hubungkan pin sensor gas MQ-4 ke arduino Nano seperti Gambar 4.7, kemudian sambungkan arduino Nano ke laptop, buka *serial monitor* pada pemrograman arduino sehingga kita dapat melihat keluaran nilai sensor gas MQ-4, nyalakan *stopwatch* lalu tunggu hingga angka pada *serial monitor* stabil jika sudah stabil maka pemanasan sensor gas dianggap sudah selesai. Waktu pemanasan alat pemanas (*heater*) rata-rata yang diperlukan agar komponen sensitif dapat bekerja berdasarkan pengujian yang telah dilakukan ditunjukkan pada Tabel 4.4.



Gambar 4.7 Rangkaian Pengujian Senor MQ-4

Setelah dirangkai seperti gambar di atas maka di buatlah program di *software* IDE Seperti Gambar 4.8.

```

Station_FIX

void loop()
{
  MQ7GetData(); //Mendapatkan data MQ7
  MQ4GetData(); //Mendapatkan data MQ4

  Serial.print("MQ7: ");
  Serial.println(dataPPM[0]);
  Serial.print("MQ7: ");
  Serial.println(analogRead(1));
  Serial.print("MQ4: ");
  Serial.println(dataPPM[1]);
  Serial.print("MQ4: ");
  Serial.println(analogRead(0));

  sprintf(buff, "CO : %d      ", dataPPM[0]); //analogRead(1);
  send(buff);
  //delay(500);
  //Serial.println(buff);
  sprintf(buff, "CH4 : %d      ", dataPPM[1]); //analogRead(0);
  send(buff);
  //delay(500);
  //Serial.println(buff);
}

```

Gambar 4.8 Program Pembacaan Sensor CH₄

Tabel 4.4 Pengujian Alat Pemanas (*Heater*) Sensor MQ-4

No. Pengujian	Waktu Pemanasan
Pengujian 1	157 detik
Pengujian 2	74 detik
Pengujian 3	63 detik
Pengujian 4	52 detik
Pengujian 5	46 detik
Waktu pemanasan rata-rata	78,4 detik

Dari percobaan yang telah dilakukan, dapat diketahui bahwa semakin lama interval atau jarak penggunaan dari sensor, maka membutuhkan waktu pemanasan yang semakin lama pula. Pengujian 1 dilakukan dalam rentang waktu sekitar 3 hari dari percobaan sebelumnya. Sementara untuk percobaan 2 hingga percobaan 5 dilakukan dalam rentang waktu 60 menit. Dan dari percobaan ini, dapat diketahui bahwa waktu rata-rata pemanasan *heater* adalah 78,4 detik.

Pada Tabel 4.5 berikut ini adalah data pengujian sensor gas CH₄, saat pengambilan data sumber yang digunakan yaitu gas pada korek api yaitu dengan cara nyalakan gas pada korek api lalu dekatkan pada *exhaust*

fan alat pendeteksi gas beracun *portable* maka pada layar LCD alat pemantau akan muncul kadar gas beracun yang terdeteksi.

Tabel 4.5 Hasil Pembacaan Sensor Gas CH₄

PPM	ADC
100	45
150	50
251	60
353	70
504	85
606	95

Dari data Tabel 4.5 dimasukkan ke *Microsoft excel* untuk mencari linieritasnya. Dengan rumus *intercept* dan *slope* bisa ditemukan bahwa untuk mengkalibrasi sensor CH₄ dengan persamaan dibawah ini :

$$x = 10,12129y - 355,854$$

dimana hal ini(4.3)

x = Nilai hasil kalibrasi (ppm)

y = Data ADC yang dibaca

Pada Tabel 4.6 berikut ini adalah hasil data dari pembacaan sensor gas CH₄, Sumber yang dipakai untuk pengambilan data yaitu menggunakan Kompor Gas. Cara pengujiannya yaitu nyalakan kompor gas dan dekatkan alat pendeteksi gas beracun *portable*, maka *exhaust fan* akan menghisap masuk gas yang ada pada kompor gas dan kadar gas beracun akan muncul pada layar LCD alat pemantau.

Tabel 4.6 Hasil Pembacaan Sensor Gas CH₄

PPM	ADC
26	240
35	256
44	267
51	274
72	292
83	302
86	310
74	298
64	293
12	214
13	216

PPM	ADC
16	218
23	230
51	281
58	288
50	273
33	246
23	230

Dari data Tabel 4.6 dimasukkan ke *Microsoft excel* untuk mencari linieritasnya. Dengan rumus *intercept* dan *slope* bisa ditemukan bahwa untuk mengkalibrasi sensor CH₄ dengan persamaan dibawah ini :

$$x = 0,735326y - 147,924$$

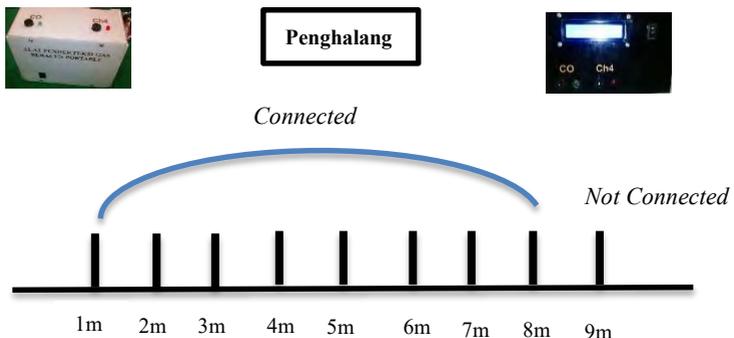
dimana hal ini.....(4.4)

x = Nilai hasil kalibrasi (ppm)

y = Data ADC yang dibaca

4.4 Pengujian *Wireless RF 433 mHz*

Cara pengujian pada Gambar 4.12 *wireless RF 433 mHz* yaitu alat pendeteksi gas beracun *portable* dipisah dengan alat pendeteksi gas beracun pemantau dengan pembatas tembok yang mempunyai lebar sekitar 12 cm dan di lakukan dengan jarak sejauh 1-9 meter, kemudian dibuat program *wireless* di *software IDE arduino*. Untuk pemrograman *Wireless RF 433 mHz* ada 2 pemrograman yaitu pemrograman untuk rangkaian pengirim yang ditunjukkan pada Gambar 4.9 dan pemrograman rangkaian penerima yang ditunjukkan pada Gambar 4.10.



Gambar 4.9 Pengujian *Wireless Rf 433 mHz*

```

    sprintf(buff, "MQ7 : %d", dataPPM[0]); //analogRead(0);
    send(buff);
    delay(2000);

    sprintf(buff, "MQ4 : %d", dataPPM[1]); //analogRead(1);
    send(buff);
    delay(2000);
}

void send (char *pesan)
{
    vw_send((uint8_t *)pesan, strlen(pesan));
    vw_wait_tx();
}

void MQ7GetData(void)
{
    for(int i=0; i<3; i++)
        dataPPM[0] += MQGetGasPercentage(MQRead(MQ7_PIN), GAS_CO);
    dataPPM[0]/=3;
}

void MQ4GetData(void)
{
    for(int i=0; i<3; i++)
        dataPPM[1] += MQGetGasPercentage(MQRead(MQ4_PIN), GAS_CH4);
    dataPPM[1]/=3;
}

```

Gambar 4.10 Pemrograman Pengiriman data *Wireless*.

```

void loop()
{
    uint8_t message[VW_MAX_MESSAGE_LEN];
    uint8_t messageLength = VW_MAX_MESSAGE_LEN;
    buff = "";

    if (vw_get_message(message, &messageLength))
    {
        Serial.print("Received: ");
        for (int i = 0; i < messageLength; i++)
        {
            buff += (char)message[i];
            //Serial.write(message[i]);
        }
        Serial.println(buff);
    }
}

```

Gambar 4.11 Pemrograman Penerima Data *Wireless*

Setelah data dikirim maka data akan muncul pada layar LCD yang ditunjukkan pada Gambar 4.11.



Gambar 4.12 Data Yang Muncul Pada Layar LCD

Dan diperoleh data pada Tabel 4.7 batas tidak tersambung Rf 433 mHz mencapai 9 meter. Saat pengiriuman data dilakukan akan memiliki *delay* beberapa detik seperti yang ditunjukkan pada Tabel 4.8.

Tabel 4.7 Jarak Sinyal *Wireless*

No	Jarak	Status
1.	1 meter	Tersambung
2.	3 meter	Tersambung
3.	5 meter	Tersambung
4.	7 meter	Tersambung
5.	9 meter	Tidak Tersambung

Tabel 4.8 Tabel *Delay* Terhadap Jarak Pengiriman

No.	Jarak	<i>Delay</i> (s)
1	1 meter	7
2	3 meter	9
3	5 meter	11
4.	7 meter	15
5.	9 meter	Tidak Tersambung

4.5 Pengambilan Data Pada Sumur

Pengambilan data pada sumur dilakukan untuk mengetahui didalam sumur terdapat gas beracun CO dan CH₄ atau tidak, pengujian pertama dilakukan di Perumahan Wisma permai tengah 1 AA 1B Surabaya, pengujian kedua dilalukan di Rumah Ibu Juwita Ngelom Megare Rt/Rw 01/01 No. 619 dan pengujian ketiga di Rumah Bapak Agus Ngelom Megare Rt/Rw 01/01 No. 616. Pengambilan data pada sumur dilakukan

dengan cara memasukkan alat pendeteksi gas beracun *portable* ke dalam sumur dengan menggunakan tali rafia hingga mencapai kedalaman yang diinginkan seperti Gambar 4.13 menunjukkan pengambilan data didalam sumur. Jika terdeteksi adanya gas beracun maka kadar gas akan muncul pada layar LCD alat pemantau dan data yang telah diperoleh ditunjukkan pada Tabel 4.9.

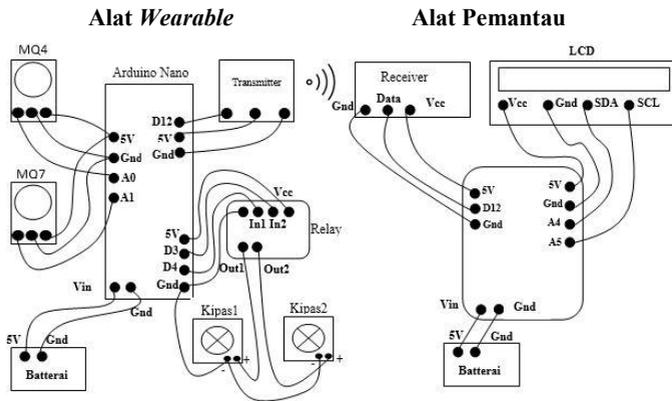


Gambar 4.13 Pengambilan Data Pada Sumur

Tabel 4.9 Pengambilan Data Pada Sumur

Sumur ke	Jarak	Gas yang terdeteksi (PPM)	
		CO	CH ₄
1	2 meter	0	0
2	5 meter	1	0
3	7 meter	1	0

4.6 Pengujian Keseluruhan



Gambar 4.14 Rangkaian Pengujian Keseluruhan

Setelah di rangkai seperti Gambar 4.14 selanjutnya unggah program terlebih dahulu yang dibuat di *software* IDE. Untuk pemrograman arduino Nano dapat dilihat di lampiran A-1 sampai A-4 dan untuk pemrograman arduino Uno dapat dilihat pada lampiran A-5 sampai A-9.

Pada pengujian ini dilakukan untuk mengetahui pendeteksian gas beracun secara keseluruhan dengan cara saat alat pendeteksi gas beracun *portable* di *on* kan dan juga alat pemantau di *on* kan, maka *exhaust fan* dari alat *portable* akan menghirup gas yang ada disekitar, lalu gas beracun akan dideteksi oleh sensor gas MQ-7 untuk gas CO dan sensor gas MQ-4 untuk gas CH₄, kemudian data dikirim ke alat pemantau utuk ditampilkan pada layar LCD. Jika terdeteksi adanya gas berbahaya maka indikator *buzzer* dan led akan menyala ketika melewati ambang batas sekitar 500 ppm , untuk led warna hijau menandakan adanya gas beracun CO dan led warna merah menandakan adanya gas beracun CH₄. Pengujian keseluruhan menggunakan sumber gas korek api, pengujian dilakukan di perumahan Wisma Permai dengan jarak 1 sampai 9 meter dengan pembatas tembok yang mempunyai lebar sekitar 12 cm.

Pada pengujian ini dilakukan 4 kali percobaan saat alat pendeteksi gas beracun ini tidak mendeteksi gas beracun hingga terdeteksinya 2 macam gas beracun yaitu CO dan CH₄. Gambar 4.15 merupakan pendeteksian awal tidak ada gas beracun yang terdeteksi. Gambar 4.16

merupakan saat terdeteksinya gas CO dan led menyala berwarna hijau. Gambar 4.17 merupakan Gambar saat terdeteksinya gas CH₄ dan led menyala berwarna merah. Gambar 4.18 merupakan terdeteksinya 2 gas beracun yaitu CO dan CH₄. Dan Tabel 4.10 merupakan data yang diperoleh saat pengujian keseluruhan.



Gambar 4.15 Pendeteksi Awal Gas Beracun



Gambar 4.16 Terdeteksinya Gas CO



Gambar 4.17 Terdeteksinya Gas CH₄



Gambar 4.18 Terdeteksinya Gas CO dan CH₄

Tabel 4.10 Pengujian Keseluruhan

NO.	Gas Yang Terdeteksi	PPM		Jarak (m)	Delay (s)
		CO	CH ₄		
1.	-	0	0	-	-
2.	CO	511	0	3	9
3.	CH ₄	0	506	5	11
4.	CO dan CH ₄	502	505	7	15

LAMPIRAN

Lampiran A

Listing Program Arduino Nano

```
#include <VirtualWire.h>
#include <math.h>

#define MQ7_PIN          1
#define MQ4_PIN          0

#define RL_VALUE          5
#define RO_CLEAN_AIR_FACTOR    9.83 //Nilai RO pada Udara Bersih

#define CALIBRATION_SAMPLE_TIMES    20 //Banyaknya pengambilan sampling
#define CALIBRATION_SAMPLE_INTERVAL 50

#define READ_SAMPLE_TIMES    5
#define READ_SAMPLE_INTERVAL  2

#define GAS_SMOKE            0
#define GAS_CO                1
#define GAS_CH4               2

float COCurve[3] = {2.3, 0.72, -0.34};
float SmokeCurve[3] = {2.3, 0.53, -0.44};
float CH4Curve[3] = {2.7, 0.82, -0.55};

float Ro1 = 10; //Inisialisasi nilai R0 pada board 10 kOhm
float Ro2 = 10; //Inisialisasi nilai R0 pada board 10 kOhm
unsigned long data;

unsigned int dataPPM[2] = {0, 0};
char buff[32], buff2[32];

void setup()
{
  Serial.begin(9600);
```

```

Ro1 = MQCalibration(MQ7_PIN);
Ro2 = MQCalibration(MQ4_PIN);

vw_set_tx_pin(12);
vw_setup(2000); // Bits per sec
vw_set_ptt_inverted(true);
pinMode(3,OUTPUT);
pinMode(4,OUTPUT);
}

void loop()
{
  MQ7GetData(); //Mendapatkan data MQ7
  MQ4GetData(); //Mendapatkan data MQ4

  Serial.print("MQ7: ");
  Serial.println(analogRead(0));
  Serial.print("MQ4: ");
  Serial.println(analogRead(1));

  sprintf(buff, "MQ7 : %d      ", analogRead(0)); //dataPPM[0];
  send(buff);
  //delay(500);
  //Serial.println(buff);
  sprintf(buff, "MQ4 : %d      ", analogRead(1)); //dataPPM[1];
  send(buff);
  //delay(500);
  //Serial.println(buff);
  digitalWrite(3,LOW);
  delay(1000);
  digitalWrite(3,HIGH);
  delay(2000);
  digitalWrite(4,LOW);
  delay(1000);
  digitalWrite(4,HIGH);
  delay(2000);
}

void send (char *pesan)

```

```

{
  vw_send((uint8_t *)pesan, strlen(pesan));
  vw_wait_tx();
}

void MQ7GetData(void)
{
  for(int i=0; i<3; i++)
    dataPPM[0] += MQGetGasPercentage(MQRead(MQ7_PIN),
    GAS_CO);
  dataPPM[0]/=3;
}

void MQ4GetData(void)
{
  for(int i=0; i<3; i++)
    dataPPM[1] += MQGetGasPercentage(MQRead(MQ4_PIN),
    GAS_CH4);
  dataPPM[1]/=3;
}

// Mendapatkan kalkulasi nilai resistansi sensor
float MQResistanceCalculation(int raw_adc)
{
  return (((float)RL_VALUE * (1023 - raw_adc)/raw_adc));
}

//Melakukan kalibrasi sensor untuk mendapatkan nilai Ro yang sesuai
dengan lingkungan
float MQCalibration(int mq_pin)
{
  int i;
  float val = 0;

  for(i = 0; i<CALIBRATION_SAMPLE_TIMES; i++)
  {
    val += MQResistanceCalculation(analogRead(mq_pin));
    delay(CALIBRATION_SAMPLE_INTERVAL);
  }
}

```

```

    }

    val = val/CALIBRATION_SAMPLE_TIMES;
    val = val/RO_CLEAN_AIR_FACTOR;
    return val;
}

//Melakukan pembacaan nilai ADC sensor untuk mendapatkan nilai
resistansi
float MQRead(int mq_pin){
    int i;
    float rs=0;
    for(i=0; i<READ_SAMPLE_TIMES; i++)
    {
        rs += MQResistanceCalculation(analogRead(mq_pin));
        delay(READ_SAMPLE_INTERVAL);
    }
    rs = rs/READ_SAMPLE_TIMES;
    return rs;
}

//Mendapatkan nilai prosenstase dalam ppm
double MQGetGasPercentage(float rs_ro_ratio, int gas_id)
{
    if(gas_id == GAS_SMOKE)
        return MQGetPercentage(rs_ro_ratio, SmokeCurve);
    else if(gas_id == GAS_CO)
        return MQGetPercentage(rs_ro_ratio, COCurve);
    else if(gas_id == GAS_CH4)
        return MQGetPercentage(rs_ro_ratio, CH4Curve);

    return 0;
}

int MQGetPercentage(float rs_ro_ratio, float *pcurve)
{
    return (pow(10,(((log(rs_ro_ratio)-pcurve[1])/pcurve[2]) +
pcurve[0])));
}

```

Listing Program Arduino Uno

```
#include <VirtualWire.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

#define batas0 500
#define batas1 400

LiquidCrystal_I2C lcd(0x27,16,2);
String buff;
boolean alarm=false;
int mode;
int found, data0, data1;
void setup()
{
  pinMode(11, OUTPUT);
  pinMode(10, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(8, OUTPUT);

  lcd.init();
  //lcd.setBacklightPin(3, POSITIVE);
  lcd.setBacklight(HIGH);

  Serial.begin(9600);
  vw_set_rx_pin(12);
  vw_setup(2000); // Bits per sec
  vw_set_ptt_inverted(true);
  vw_rx_start(); // Start the receiver PLL running

  lcd.setCursor(0,0);
  lcd.print(" Warming Up ");
  lcd.setCursor(0,1);
  lcd.print(" Sensors ");
  delay(1000);
  lcd.clear();
}

void loop()
```

```

{
uint8_t message[VW_MAX_MESSAGE_LEN];
uint8_t messageLength = VW_MAX_MESSAGE_LEN;
buff = "";
if (vw_get_message(message, &messageLength)) // Non-blocking
{
Serial.print("Received: ");
for (int i = 0; i < messageLength; i++)
{
buff += (char)message[i];
//Serial.write(message[i]);
}
Serial.println(buff);

// int found, data0, data1;
String temp;

found = findStr("MQ4 : ", buff);
if(found > -1)
{
data0 = atoi(buff.substring(found+6).c_str());
}
else if(findStr("MQ7 : ", buff) > -1)
{
data1 = atoi(buff.substring(found+6).c_str());

if(data1 > batas1)
{
digitalWrite(11, HIGH);
delay(50);
digitalWrite(9, HIGH);
delay(50);
digitalWrite(11, LOW);
delay(100);
digitalWrite(9, LOW);
delay(100);
digitalWrite(11, HIGH);
delay(50);
digitalWrite(9, HIGH);
}
}
}

```

```

delay(50);
digitalWrite(11, LOW);
delay(100);
digitalWrite(9, LOW);
delay(100);
}
if(data0 > batas1)
{
digitalWrite(10, HIGH);
delay(50);
digitalWrite(8, HIGH);
delay(50);
digitalWrite(10, LOW);
delay(100);
digitalWrite(8, LOW);
delay(100);
digitalWrite(10, HIGH);
delay(50);
digitalWrite(8, HIGH);
delay(50);
digitalWrite(10, LOW);
delay(100);
digitalWrite(8, LOW);
delay(100);
}
if(data0 > batas0 && data1 > batas0)
{
mode=1;
alarm=true;
}
else if(data1 > batas0 && data0 < batas0 )
{
mode=2;
alarm=true;
}
else if(data0 > batas0 && data1 < batas0)
{
mode=3;
alarm=true;
}

```

```

}
else if(data1<batas0 && data0<batas0)
{
  alarm=false;
}

```

```

if(alarm==true)
{
  if(mode==2)
  {
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print(" ---ALARM ON--- ");
    lcd.setCursor(0,1);
    lcd.print(" CO BERBAHAYA! ");
    digitalWrite(11, HIGH);
    digitalWrite(9, HIGH);
    delay(3000);
  }
  if(mode==3)
  {
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print(" ---ALARM ON--- ");
    lcd.setCursor(0,1);
    lcd.print(" CH4 BERBAHAYA! ");
    digitalWrite(10, HIGH);
    digitalWrite(8, HIGH);
    delay(3000);
  }
  if(mode==1)
  {
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print(" ---ALARM ON--- ");
    lcd.setCursor(0,1);
    lcd.print("C0&CH4 BERBAHAYA!");
  }
}

```

```

    digitalWrite(11, HIGH);
    digitalWrite(10, HIGH);
    digitalWrite(9, HIGH);
    digitalWrite(8, HIGH);
    delay(3000);
  }
}
else
{
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("MQ4 :");
  lcd.print(data0);
  lcd.setCursor(0,1);
  lcd.print("MQ7 :");
  lcd.print(data1);
  digitalWrite(11, LOW);
  digitalWrite(9, LOW);
  digitalWrite(10, LOW);
  digitalWrite(8, LOW);
}
Serial.print(data0);
Serial.println(data1);
Serial.println(mode);
}
}
}

```

```

int findStr(String needle, String haystack)
{
  int foundpos = -1;
  for (int i = 0; i <= haystack.length() - needle.length(); i++)
  {
    if (haystack.substring(i,needle.length()+i) == needle) foundpos = i;
  }
  return foundpos;
}

```


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

Lampiran B

Link Berita Penggali Sumur Korban Gas Beracun

<http://jambi.tribunnews.com/2015/11/12/sumur-maut-diduga-ada-gas-beracun>

https://www.facebook.com/permalink.php?story_fbid=897954833550499&id=188275254518464

<http://suarakampar.com/berita/berita-diduga-hirup-gas-beracun-penggali-sumur-tewas-dan-satu-kritis-di-salo-.html>

<http://www.terasjatim.com/diduga-karena-gas-beracun-3-penggali-sumur-tewas-di-dasar-sumur/>

<http://www.merdeka.com/peristiwa/terjebak-gas-beracun-tiga-penggali-sumur-tewas.html>

<http://www.beritasatu.tv/news/tiga-orang-penggali-sumur-tewas-akibat-gas-beracun/>

<http://news.okezone.com/read/2015/10/27/338/1238596/dua-penggali-sumur-tewas-usai-hirup-gas-beracun>

<http://www.mandaoexpress.co/berita-4368-sumur-di-tumatangtang-diduga-keluarkan-gas-beracun.>

Data Sheet

Electric parameter measurement circuit is shown as Fig.2

E. Sensitivity characteristic curve

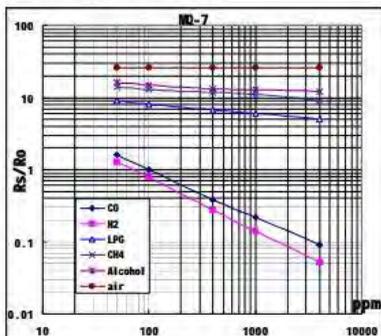


Fig.3 sensitivity characteristics of the MQ-7

Fig.3 is shows the typical sensitivity characteristics of the MQ-7 for several gases.

in their: Temp: 20°C ,

Humidity: 65% ,

O₂ concentration 21%

RL=10k Ω

Ro: sensor resistance at 100ppm

CO in the clean air.

Rs: sensor resistance at various concentrations of gases.

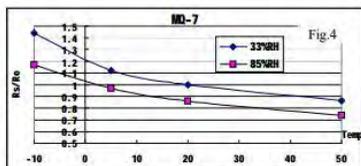


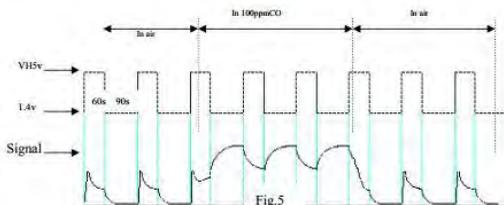
Fig.4 shows the typical dependence of the MQ-7 on temperature and humidity.
 Ro: sensor resistance at 100ppm CO in air at 33%RH and 20degree.
 Rs: sensor resistance at 100ppm CO at different temperatures and humidities.

OPERATION PRINCIPLE

The surface resistance of the sensor R_s is obtained through effected voltage signal output of the load resistance R_L which series-wound. The relationship between them is described:

$$R_s R_L = (V_c - V_{RL}) / V_{RL}$$

Fig. 5 shows alterable situation of R_L signal output measured by using Fig. 2 circuit output



signal when the sensor is shifted from clean air to carbon monoxide (CO), output signal measurement is made within one or two complete heating period (2.5 minute from high voltage to low voltage).

Sensitive layer of MQ-7 gas sensitive components is made of SnO_2 with stability, So, it has excellent long term stability. Its service life can reach 5 years under using condition.

SENSITIVITY ADJUSTMENT

Resistance value of MQ-7 is difference to various kinds and various concentration gases. So, When using this components, sensitivity adjustment is very necessary. we recommend that you calibrate the detector for 200ppm CO in air and use value of Load resistance that (R_L) about 10 K Ω (5K Ω to 47 K Ω).

When accurately measuring, the proper alarm point for the gas detector should be determined after considering the temperature and humidity influence. The sensitivity adjusting program:

- Connect the sensor to the application circuit.
- Turn on the power, keep preheating through electricity over 48 hours.
- Adjust the load resistance R_L until you get a signal value which is respond to a certain carbon monoxide concentration at the end point of 90 seconds.
- Adjust the another load resistance R_L until you get a signal value which is respond to a CO concentration at the end point of 60 seconds.

[Supplying special IC solutions, More detailed technical information, please contact us.](#)

Datasheet MQ-4

SPECIFICATIONS

A. Standard work condition

Symbol	Parameter name	Technical condition	Remarks
Vc	Circuit voltage	5V±0.1	AC OR DC
V _H	Heating voltage	5V±0.1	AC OR DC
P _L	Load resistance	20K Ω	
R _H	Heater resistance	33 Ω ± 5%	Room Tem
P _H	Heating consumption	less than 750mw	

B. Environment condition

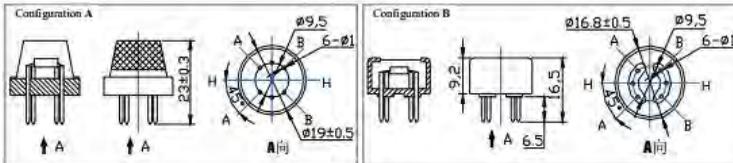
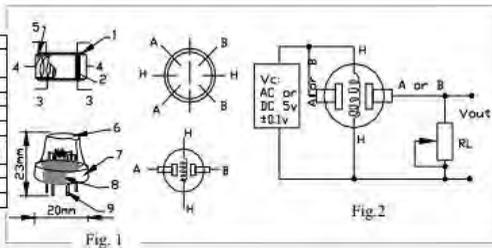
Symbol	Parameter name	Technical condition	Remarks
Tao	Using Tem	-10°C-50°C	
Tas	Storage Tem	-20°C-70°C	
R _H	Related humidity	less than 95%Rh	
O ₂	Oxygen concentration	21%(standard condition)Oxygen concentration can affect sensitivity	minimum value is over 2%

C. Sensitivity characteristic

Symbol	Parameter name	Technical parameter	Remark 2
R _s	Sensing Resistance	10K Ω - 60K Ω (1000ppm CH ₄)	Detecting concentration scope: 200-10000ppm CH ₄ , natural gas
α (1000ppm/ 5000ppm CH ₄)	Concentration slope rate	≤0.6	
Standard detecting condition	Temp: 20°C ± 2°C Humidity: 65%±5%	Vc: 5V±0.1 Vh: 5V±0.1	
Preheat time	Over 24 hour		

D. Strucyre and configuration, basic measuring circuit

Parts	Materials
1 Gas sensing layer	SnO ₂
2 Electrode	Au
3 Electrode line	Pt
4 Heater coil	Ni-Cr alloy
5 Tubular ceramic	Al ₂ O ₃
6 Anti-explosion network	Stainless steel gauze (SUS316 100-mesh)
7 Clamp ring	Copper plating Ni
8 Resin base	Bakelite
9 Tube Pin	Copper plating Ni



Structure and configuration of MQ-4 gas sensor is shown as Fig. 1 (Configuration A or B), sensor composed by micro Al_2O_3 ceramic tube, Tin Dioxide (SnO_2) sensitive layer, measuring electrode and heater are fixed into a crust made by plastic and stainless steel net. The heater provides necessary work conditions for work of sensitive components. The enveloped MQ-4 have 6 pin, 4 of them are used to fetch signals, and other 2 are used for providing heating current.

Electric parameter measurement circuit is shown as Fig.2

E. Sensitivity characteristic curve

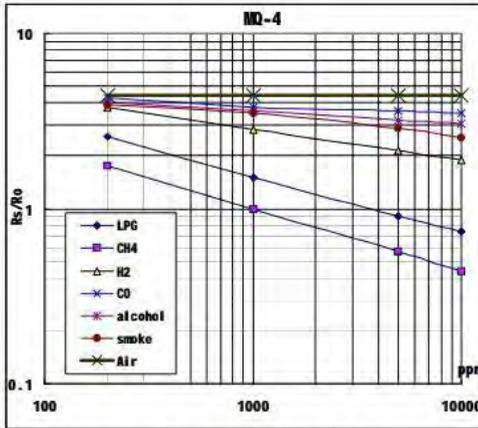


Fig.2 sensitivity characteristics of the MQ-4

Fig.3 shows the typical sensitivity characteristics of the MQ-4 for several gases. in their: Temp: 20°C, Humidity: 65%, O_2 concentration 21%, $R_L=20k\ \Omega$. R_o : sensor resistance at 1000ppm of CH_4 in the clean air. R_s : sensor resistance at various concentrations of gases.

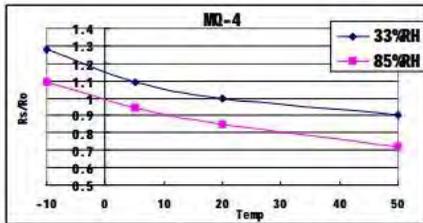


Fig.4 is shows the typical dependence of the MQ-4 on temperature and humidity. R_o : sensor resistance at 1000ppm of CH_4 in air at 33%RH and 20 degree. R_s : sensor resistance at 1000ppm of CH_4 in air at different temperatures and humidities.

SENSITIVITY ADJUSTMENT

Resistance value of MQ-4 is difference to various kinds and various concentration gases. So, When using this components, sensitivity adjustment is very necessary. we recommend that you calibrate the detector for 5000ppm of CH_4 concentration in air and use value of Load resistance (R_L) about $20K\ \Omega$ ($10K\ \Omega$ to $47K\ \Omega$).

When accurately measuring, the proper alarm point for the gas detector should be determined after considering the temperature and humidity influence.

Datasheet Relay

2 CHANNEL 5V 10A RELAY MODULE



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

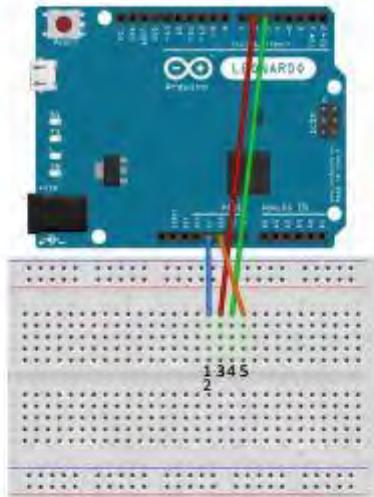
- On-board EL817 photoelectric coupler with photoelectric isolating anti-interference ability strong
- On-board 5V, 10A / 250VAC, 10A / 30VDC relays
- Relay long life can absorb 100000 times in a row
- Module can be directly and MCU I/O link, with the output signal indicator
- Module with diode current protection, short response time
- PCB Size: 45.8mm x 32.4mm

Pin Configuration

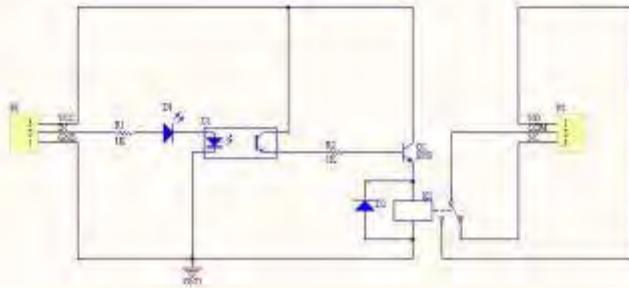


1. **VCC**: 5V DC
2. **COM**: 5V DC
3. **IN1**: high/low output
4. **IN2**: high/low output
5. **GND**: ground

Wiring Diagram



Schematic Diagram



Sample Sketch

```
void setup(){
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
}

void loop(){
  digitalWrite(5, LOW);
  digitalWrite(6, HIGH);
  delay(4000);
  digitalWrite(5, HIGH);
  digitalWrite(6, LOW);
  delay(4000);
}
```

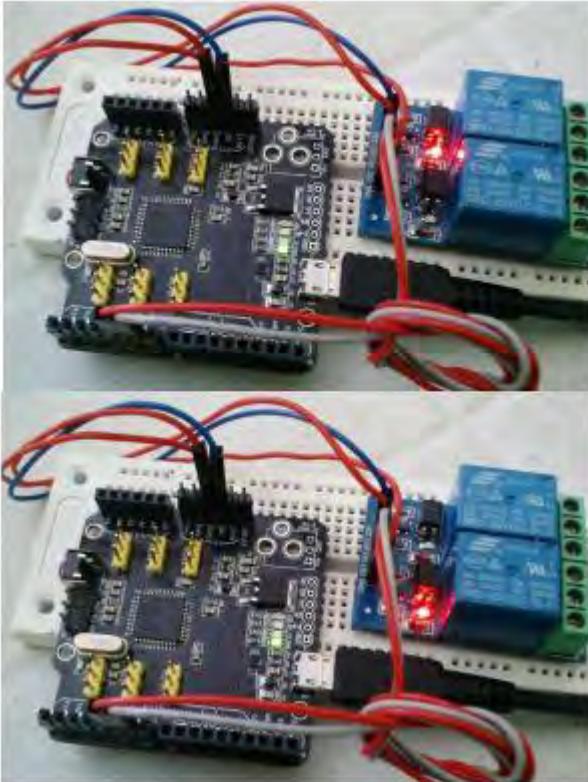
How to Test

The components to be used are:

- Microcontroller (any compatible arduino)
 - 2 channel 5V 10A relay module
 - Pin connectors
 - Breadboard
 - USB cable
1. Connect the components based on the figure shown in the wiring diagram using pin connectors. VCC and COM pin is connected to the 5V power supply, GND pin is connected to the GND, IN1 and IN2 pins are connected to the digital I/O pin. Pin number will be based on the actual program code.
 2. After hardware connection, insert the sample sketch into the Arduino IDE.
 3. Using a USB cable, connect the ports from the microcontroller to the computer.
 4. Upload the program.

Testing Results

The figures below shows an alternate switching of the two relays every 4 seconds. A tick sound and a red LED would be observed.



Datasheet Arduino Uno

Technical Specification

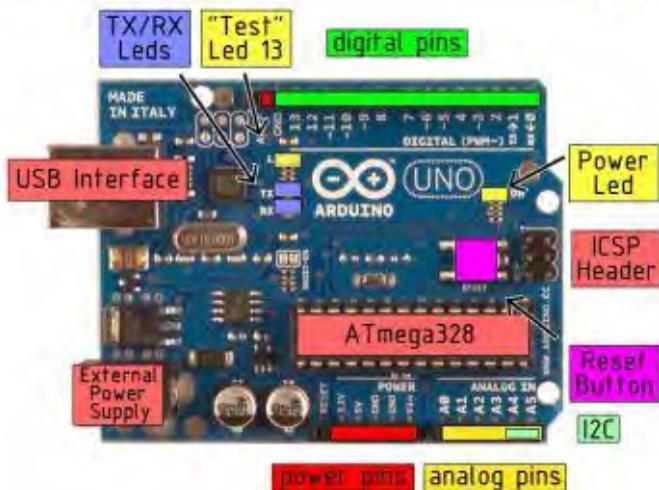


EAGLE files: [arduino-drumlinzov-uno-design.zip](#) Schematic: [arduino-uno-schematic.pdf](#)

Summary

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB of which 0.5 KB used by bootloader
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

the board



Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Uno is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see [this forum thread](#) for details.

USB Overcurrent Protection

The Arduino Uno has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics

The maximum length and width of the Uno PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.



How to use Arduino



Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the [Arduino programming language](#) (based on [Wiring](#)) and the Arduino development environment (based on [Processing](#)). Arduino projects can be stand-alone or they can communicate with software on running on a computer (e.g. Flash, Processing, MaxMSP).

Arduino is a cross-platform program. You'll have to follow different instructions for your personal OS. Check on the [Arduino site](#) for the latest instructions. <http://arduino.cc/en/Guide/HomePage>

Linux Install

Windows Install

Mac Install

Once you have downloaded/unzipped the arduino IDE, you can Plug the Arduino to your PC via USB cable.

Blink led

Now you're actually ready to "burn" your first program on the arduino board. To select "blink led", the physical translation of the well known programming "hello world", select

**File>Sketchbook>
Arduino-0017>Examples>
Digital>Blink**

Once you have your skech you'll see something very close to the screenshot on the right.

In **Tools>Board** select

Now you have to go to **Tools>SerialPort** and select the right serial port, the one arduino is attached to.



Done compiling

Press Compile button
(to check for errors)



Upload

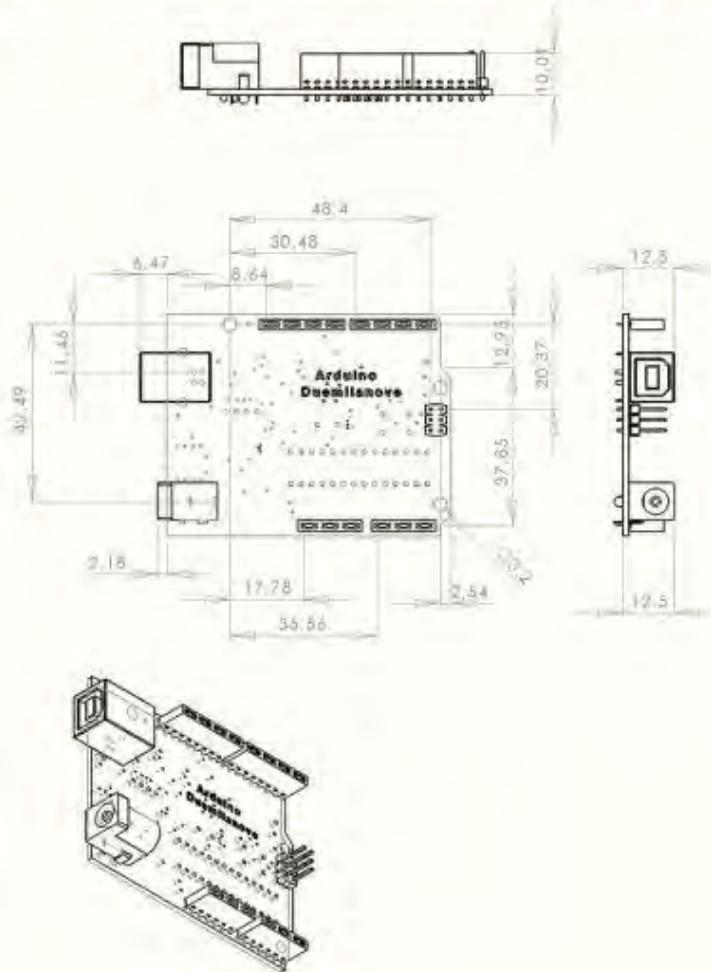


TX RX Flashing



Blinking Led

Dimensioned Drawing



Datasheet Arduino Nano



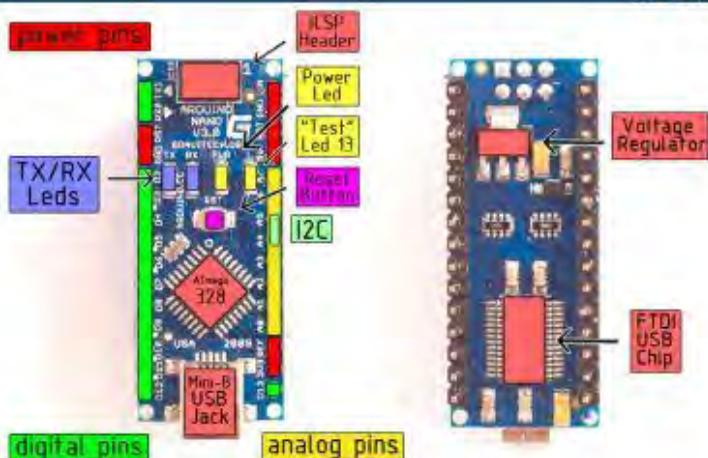
Arduino Nano 3.0 (ATmega328): [schematic](#), [Eagle files](#).

Arduino Nano 2.3 (ATmega168): [manual](#) (pdf), [Eagle files](#). Note: since the free version of Eagle does not handle more than 2 layers, and this version of the Nano is 4 layers, it is published here unrouted, so users can open and use it in the free version of Eagle.

Summary

Microcontroller	Atmel ATmega168 or ATmega328
Operating Voltage (logic level)	5 V
Input Voltage (recommended)	7-12 V
Input Voltage (limits)	6-20 V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	8
DC Current per I/O Pin	40 mA
Flash Memory	16 KB (ATmega168) or 32 KB (ATmega328) of which 2 KB used by bootloader
SRAM	1 KB (ATmega168) or 2 KB (ATmega328)
EEPROM	512 bytes (ATmega168) or 1 KB (ATmega328)
Clock Speed	16 MHz
Dimensions	0.73" x 1.70"

the board



Communication

The Arduino Nano has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega168 and ATmega328 provide UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An FTDI FT232RL on the board channels this serial communication over USB and the [FTDI drivers](#) (included with the Arduino software) provide a virtual com port to software on the computer. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the FTDI chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A [SoftwareSerial library](#) allows for serial communication on any of the Nano's digital pins.

The ATmega168 and ATmega328 also support I2C (TWI) and SPI communication. The Arduino software includes a [Wire library](#) to simplify use of the I2C bus; see the [documentation](#) for details. To use the SPI communication, please see the ATmega168 or ATmega328 datasheet.

Programming

The Arduino Nano can be programmed with the Arduino software ([download](#)). Select "Arduino Diecimila, Duemilanove, or Nano w/ ATmega168" or "Arduino Duemilanove or Nano w/ ATmega328" from the **Tools > Board** menu (according to the microcontroller on your board). For details, see the [reference](#) and [tutorials](#).

The ATmega168 or ATmega328 on the Arduino Nano comes preburned with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol ([reference](#), [C header files](#)).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see [these instructions](#) for details.

Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Nano is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the FT232RL is connected to the reset line of the ATmega168 or ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Nano is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Nano. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.



Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the [Arduino programming language](#) (based on [Wiring](#)) and the Arduino development environment (based on [Processing](#)). Arduino projects can be stand-alone or they can communicate with software on running on a computer (e.g. Flash, Processing, MaxMSP).

Arduino is a cross-platform program. You'll have to follow different instructions for your personal OS. Check on the [Arduino site](#) for the latest instructions. <http://arduino.cc/en/Guide/HomePage>

Linux Install

Windows Install

Mac Install

Once you have downloaded/unzipped the arduino IDE, you'll need to install the FTDI Drivers to let your PC talk to the board. First **Plug the Arduino to your PC via USB cable.**

Blink led

Now you're actually ready to "burn" your first program on the arduino board. To select "blink led", the physical translation of the well known programming "hello world", select

**File>Sketchbook>
Arduino-0017>Examples>
Digital>Blink**

Once you have your sketch you'll see something very close to the screenshot on the right.

In **Tools>Board** select Arduino NANO and with the AtMEGA you're using (probably 328)

Now you have to go to **Tools>SerialPort** and select the right serial port, the one arduino is attached to.



Done compiling



Upload

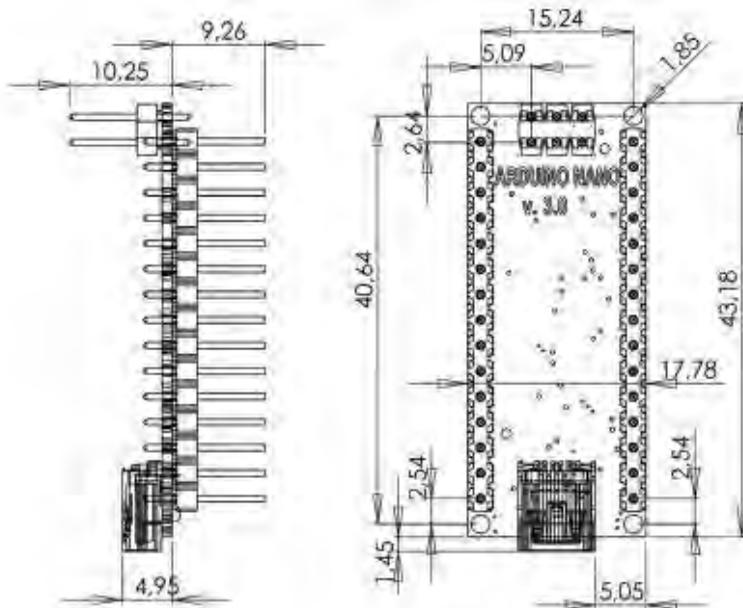


TX RX Flashing



Blinking Led!

Press Compile button
(to check for errors)





PEMERINTAH PROVINSI JAWA TIMUR
DINAS TENAGA KERJA, TRANSMIGRASI DAN KEPENDUDUKAN
UNIT PELAKSANA TEKNIS KESELAMATAN DAN KESEHATAN KERJA
(UPT K3)



Jl. Dukuhtel Mestajiral 121 Telepon 8280440, 8294490, Fax, 8294277 Surabaya 60234
Website : www.uptk3surbaya.dinasakertransdtk.jatimprov.go.id
Email : admn@uptk3surbaya.dinasakertransdtk.jatimprov.go.id

LHU ini merupakan hasil pada tesasi dan saat pengukuran
LAPORAN HASIL PENGUJIAN
No. LAB. 061/V/2016

Form: S.10-041
Terbitan / Revisi: 4 / 0

- I Nama Pengguna Jasa : MAHASISWA ITS
II Alamat : Kampus ITS – Sukolilo, Surabaya
III Jenis Pengukuran : Kadar CO₂ di Udara Lingkungan Kerja
IV Tanggal Pengukuran : 27 Mei 2016
V Alat yang digunakan : Gas Emission Monitoring Merk ECOM Type J2KN
VI Hasil Pengukuran :

No	Lokasi Pengukuran	Jam (WIB)	Alat 1	Alat 2
			CO (ppm)	CO ₂ (%)
1	Percobaan I	11,00	5309	-
2	Percobaan II	13,00	-	5,2
3	Percobaan III	13,05	3458	-
4	Percobaan IV	15,10	-	6,1
5	Percobaan V	13,15	3352	-
6	Percobaan VI	13,20	-	5,8
7	Percobaan VII	13,25	2834	-
8	Percobaan VIII	13,30	-	5,6

Catatan :13.25

- Sumber Emisi berasal dari motor Mahasiswa (Vario berbalikan bakar Pertamina)
- Pengukuran dilakukan dengan cara simulasi selintas (menit)
- Pengukuran bertujuan untuk menguji alat yang dibawa Mahasiswa agar digunakan sebagai data dalam ujian akhir Mahasiswa.
- Pengukuran dilaksanakan pada 2 alat yang berbeda, pada alat 1 digunakan untuk mengukur CO sedangkan pada alat kedua digunakan untuk mengukur CO₂.



Mengetahui,
Kepala UPT K3 SURAHAYA
Kategori Pelayanan Teknis

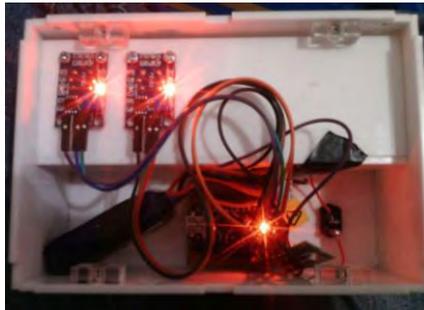
Dra. KIRIWI WINARNI, MM.
NIP. 1961111019860320017

Surabaya, 30 Mei 2016
MANAJER TEKNIK LINGKUNGAN

Drs. SITI SUMARTINI R.
NIP. 196505271992032008

Lampiran C
Tampilan Alat





BAB V

PENUTUP

5.1 Kesimpulan

Dari hasil penelitian dan pembahasan mengenai pembuatan gas beracun *portable* untuk penggali sumur berbasis mikrokontroler ini dapat disimpulkan bahwa:

1. Dari hasil uji coba , Sensor Gas CO (MQ-7) membutuhkan kondisi kerja atau pemanasan dengan waktu rata-rata sekitar 82,8 detik. Sedangkan sensor gas CH₄ (MQ-4) membutuhkan kondisi kerja atau pemanasan dengan waktu rata-rata sekitar 78,4 detik.
2. Dari hasil pengujian Sensor CO (MQ-7) dan Sensor Gas CH₄ (MQ-4) ketika ppm mencapai angka 400, maka *buzzer* akan berbunyi dan led akan berkedip. Jika ppm melebihi ambang batas yaitu mencapai angka lebih dari 500, maka *buzzer* akan berbunyi nyaring dan led akan menyala tanpa berkedip.
3. RF 433 mHz mampu mengirim data berupa sinyal tetapi memiliki *delay* dengan rata-rata 7-20 detik.

5.2 Saran

Untuk pengembangan dan penyempurnaan pembuatan alat pendeteksi gas beracun *portable* untuk penggali sumur berbasis mikrokontroler, maka bisa diberikan beberapa saran yang harapannya bisa diaplikasikan kelak di kemudian hari sebagai berikut :

1. Dalam perancangan modul sensor dapat ditambahkan jenis sensor gas yang lain sehingga jenis gas yang terdeteksi dapat lebih banyak.
2. Lakukan perbandingan dengan *gas detector* yang sudah ada agar dapat menjadi perbandingan atau acuan nilai dari hasil pengukuran gas.

DAFTAR PUSTAKA

- [1] Marwati, "Kualitas Air Sumur Gali Ditinjau dari Kondisi Lingkungan Fisik Dan Perilaku Masyarakat Di Wilayah Puskesmas 1 Denpasar Selatan". *Tesis*, Jurusan Ilmu Lingkungan Universitas Udayana, Denpasar, 2008.
- [2] Wildan Aryo Wicaksono, Indra Astriawan Suseno. "Perancangan Sistem Monitoring Tingkat Pencemaran Udara Dengan Komunikasi SMS Menggunakan Modem GSM". *Tugas Akhir*, D3 Teknik Elektro ITS, Surabaya, 2015.
- [3] Nur Ahmad Effendi, "Purwarupa Sistem Peringatan Dini Nirkabel Pada Jembatan Antar Pulau Oleh Angin Berbasis Arduino Nano". *Tugas Akhir*, D3 Elektronika dan Instrumentasi UGM, Yogyakarta, 2014.
- [4] Yuwono Marta Dinata, "Arduino Itu Mudah", PT Elex Media Komputindo, Jakarta, 2015.
- [5] Ya'kut, Haris Aydin, "Rancang Bangun Sistem Pengukur Gas Karbonmonoksida (CO) Menggunakan sensor MQ-7 Berbasis Mikrokontroler Atmega 16A". *Tugas Akhir*, Teknik Elektro Universitas Brawijaya, Malang, 2010.
- [6] Muntini, Melania Suweni, "Monitoring Produksi Gas Metana Secara Waktu Nyata pada Digester Anaerob". *Prosiding 7th Basic Science National Seminar Vol 2 hal II-329*. Teknik Elektro Universitas Brawijaya, Malang, 2010.
- [7] Heriadi, D. "Jaringan Internet Wi-Fi". C.V Andi Offset. Yogyakarta, 2005.
- [8] Pradigdy, Christian Andrian, "Rancang Bangun Monitor Suhu Gas Metane dan Karbon Dioksida pada Biogas". *Tugas Akhir*, Teknik Elektro STIKOM, Surabaya, 2013.

DAFTAR RIWAYAT HIDUP



Nama : Ninik Vatmawati
TTL : Sidoarjo, 22 Januari 1995
Jenis Kelamin : Perempuan
Agama : Islam
Alamat : Ngelom Megare RT/RW 01/01
No.617, Sepanjang-Sidoarjo
Telp/HP : 089620825600
E-mail : ninik.vatmawati.nv@gmail.com

RIWAYAT PENDIDIKAN

1. 2001 - 2007 : SD Negeri NGELOM
2. 2007 – 2010 : SMP Negeri 2 Taman
3. 2010 – 2013 : SMK Muhammadiyah 2 Surabaya
4. 2013 – 2016 : D3 Teknik Elektro, Program Studi Teknik Elektro Industri - FTI Institut Teknologi Sepuluh Nopember (ITS)

PENGALAMAN KERJA

1. Kerja Praktek di PT.SIER
2. Kerja Praktek di PLN Rayon Sidoarjo

PENGALAMAN ORGANISASI

1. Staff Departemen KOMINFO Periode 2014/2015 HIMA D3TEKTRO, FTI – ITS

Halaman ini sengaja dikosongkan

DAFTAR RIWAYAT HIDUP



Nama : Masaji Alif Rahman
TTL : Kediri, 29 Maret 1995
Jenis Kelamin : Laki-Laki
Agama : Islam
Alamat : Suwaloh RT/RW 01/05
Sambirejo, Pare-Kediri
Telp/HP : 085635158727
E-mail : masajialif29@gmail.com

RIWAYAT PENDIDIKAN

1. 2001 - 2007 : SD Islam Al-Fath Pare
2. 2007 – 2010 : SMP Islam Al-Fath Pare
3. 2010 – 2013 : SMA Negeri 2 Pare
4. 2013 – 2016 : D3 Teknik Elektro, Program Studi Teknik Elektro Industri - FTI Institut Teknologi Sepuluh Nopember (ITS)

PENGALAMAN KERJA

1. Kerja Praktek di Hotel Santika
2. Kerja Praktek di PLTA Siman

PENGALAMAN ORGANISASI