

PERANCANGAN ALAT KENDALI LAMPU MENGGUNAKAN SENSOR SUARA FC-04 BERBASIS ARDUINO UNO

TUGAS AKHIR

Diajukan untuk Memenuhi Salah Satu Syarat Kelulusan Program Pendidikan
Sarjana

Oleh:
Hendri Satriadi
2016130020



JURUSAN INFORMATIKA
SEKOLAH TINGGI MANAJEMEN INFORMATIKA & KOMPUTER-LIKMI
BANDUNG
2021

**PERANCANGAN ALAT KENDALI LAMPU MENGGUNAKAN
SENSOR SUARA FC-04 BERBASIS ARDUINO UNO**

Oleh:
Hendri Satriadi
2016130020

Bandung, 18 Januari 2021

Menyetujui,

Sudimanto, S.T., M.Kom.

Pembimbing

Dhanny Setiawan, S.T., M.T.

Ketua Jurusan

JURUSAN INFORMATIKA
SEKOLAH TINGGI MANAJEMEN INFORMATIKA & KOMPUTER-LIKMI
BANDUNG
2021

ABSTRAK

Perkembangan teknologi di dalam otomatisasi sangat cepat dapat memungkinkan manusia untuk melakukan serba cepat dan praktik. Salah satu contohnya adalah sistem otomatisasi menyalakan dan mematikan lampu dengan tepukan tangan. Maka dari itu dibutuhkan alat untuk mendeteksi tepukan tangan yaitu dengan sensor suara FC-04. Sensor ini berguna untuk mendeteksi suara tepukan tangan. Alat kendali lampu ini menggunakan sensor analog seperti sensor suara FC-04 dan sensor cahaya LDR yang dirancang menggunakan arduino UNO, dan relay, selain itu juga menggunakan diagram-diagram seperti *deployment* diagram, *statechart* diagram, dan diagram skematik. Proses kerja alat ini sensor suara FC-04 suara berupa tepukan tangan. Setelah itu sensor suara FC-04 mengirimkan data dalam bentuk analog kepada arduino. Data analog akan dikonversi menjadi sinyal digital oleh ADC kemudian diteruskan kepada mikrokontroler. Hasil data digital yang berbentuk biner 0 dan 1. Kemudian mikrokontroler mengelola data biner tersebut dan sensor cahaya LDR akan mendeteksi cahaya pada lampu yang menentukan apakah lampu menyala atau mati. Pada pengujiannya jarak tepukan tangan sangat mempengaruhi hasil dalam menyalakan dan mematikan lampu, semakin jauh tepukan tangan maka semakin lampu tidak merespon perintah untuk menyalakan dan mematikan lampu. Alat ini dapat bermanfaat untuk mempermudah pengguna dalam menyalakan dan mematikan lampu tanpa harus menekan saklar.

KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Kuasa karena atas hikmat dan anugerahNya, tugas akhir ini dapat terselesaikan dengan baik.

Penulisan tugas akhir ini diajukan untuk memenuhi salah satu syarat kelulusan pada jenjang pendidikan Strata 1 jurusan Sistem Informasi dengan bidang keahlian Manajemen Bisnis di Sekolah Tinggi Manajemen Informatika dan Komputer LIKMI Bandung.

Dalam penyusunannya, penulis menerima bantuan dari berbagai pihak berupa bimbingan, saran, dorongan, dan nasihat. Melalui kesempatan ini, penulis ingin menyampaikan rasa terima kasih kepada semua pihak yang telah membantu dalam penyelesaian tugas akhir ini, terutama kepada:

1. Bapak Sudimanto, S.T.,M.Kom. selaku dosen pembimbing yang telah meluangkan waktu untuk memberikan bimbingan, nasihat serta arahan terbaik dalam penyelesaian tugas akhir ini.
2. Bapak Dhanny Setiawan, S.T.,M.T. selaku dosen wali dan Bapak Dr. Djajasukma Tjahjadi, S.E., M.T selaku Wakil Ketua I STMIK LIKMI yang telah memperkenankan penulis menempuh tugas akhir di semester ini.
3. Keluarga penulis yang telah mendukung dalam doa, moril, dan materil sehingga penulis mampu menyelesaikan tugas akhir ini.
4. Sobat karib penulis, yaitu Yolanda Oktaviani yang selalu setia dan dapat diandalkan saat penulis membutuhkan bantuan serta senantiasa memberikan wejangan, semangat, doa dan menjadi sumber inspirasi bagi penulis dalam penyelesaian tugas akhir ini.
5. STMIK LIKMI yang telah memberi kesempatan kepada penulis untuk menempuh pendidikan sarjana, serta seluruh dosen STMIK LIKMI yang telah memberikan ilmu pengetahuan untuk kelancaran penyusunan tugas akhir ini.

6. Teman-teman persekutuan rohani (*Home Made* CLCC Bandung) yang tak pernah lelah mendoakan dan terus menjaga pertumbuhan iman penulis akan pengharapan dalam kasih Kristus.
7. Semua staff perpustakaan STMIK LIKMI yang telah membantu proses peminjaman buku kepada penulis dalam pengumpulan materi tugas akhir.

Penulis menyadari bahwa tugas akhir ini masih terdapat kekurangan. Oleh karena itu, dengan segala kerendahan hati penulis memohon saran dan kritik yang membangun untuk dijadikan bahan masukan guna memperbaiki kekurangan penelitian ini. Semoga tugas akhir ini dapat bermanfaat bagi pihak-pihak yang tertarik pada penelitian ini dan semua rekan-rekan mahasiswa/i lainnya.

Bandung, 18 Januari 2021

Penulis

DAFTAR ISI

ABSTRAK.....	i
KATA PENGANTAR	ii
DAFTAR ISI.....	iv
DAFTAR GAMBAR	vi
DAFTAR TABEL	vii
DAFTAR SIMBOL	viii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan	3
1.4 Batasan Masalah	3
1.5 Kegunaan Hasil.....	3
1.6 Metodologi Penelitian	4
1.7 Sistematika Penulisan.....	4
BAB II LANDASAN TEORI.....	6
2.1 Bunyi	6
2.2 Sensor	9
2.2.1 Sensor Suara FC-04	10
2.2.2 Sensor Cahaya LDR	12
2.3 <i>Analog to Digital Converter (ADC)</i>	12
2.4 Mikrokontroler.....	13
2.4.1 Mikrokontroler ATmega328P	15
2.4.2 Arduino	17
2.4.3 Arduino Uno	18
2.5 Arduino IDE	20
2.6 <i>Sound Meter</i>	22

2.7	Bahasa C.....	23
2.8	Rekayasa Perangkat Lunak.....	24
2.9	<i>Unified Modeling Language</i>	24
2.9.1	<i>Deployment Diagram</i>	25
2.9.2	<i>Statechart Diagram</i>	27
BAB III ANALISIS DAN PERANCANGAN		30
3.1	Gambaran Umum.....	30
3.2	Spesifikasi Kebutuhan.....	31
3.2.1	Kebutuhan Perangkat Keras	31
3.2.2	Kebutuhan Perangkat Lunak.....	31
3.3	<i>Deployment Diagram</i>	32
3.4	<i>Statechart Diagram</i>	33
3.5	Skematik Perancangan Alat Kendali Lampu.....	34
BAB IV IMPLEMENTASI DAN PENGUJIAN		35
4.1	Implementasi Perangkat Keras	35
4.2	Pengujian dan Analisa Perangkat Keras	37
4.2.1	Pengujian Sistem dan Analisa Sensor Suara FC-04	37
4.2.2	Pengujian Sistem dan Analisa Sensor Cahaya LDR	38
4.2.3	Pengujian Sistem dan Analisa Alat Kendali Lampu	38
BAB V KESIMPULAN DAN SARAN		43
5.1	Kesimpulan.....	43
5.2	Saran.....	43
DAFTAR PUSTAKA.....		44
LAMPIRAN.....		47

DAFTAR GAMBAR

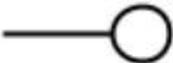
Gambar 2.1 Perambatan sinyal audio (Gelombang suara).....	7
Gambar 2.2 Grafik Gelombang Suara.....	8
Gambar 2.3 Prinsip Kerja Kondensor <i>Microphone</i>	10
Gambar 2.4 Struktur Piezoelectric <i>Microphone</i>	11
Gambar 2.5 <i>Magnetic Microphones</i>	11
Gambar 2.6 Sensor Cahaya LDR.....	12
Gambar 2.7 Konfigurasi Pin Mikrokontroler ATmega328P.....	15
Gambar 2.8 Blok Diagram ATmega328P.....	17
Gambar 2.9 Arduino UNO.....	18
Gambar 2.10 <i>Sound Meter</i>	22
Gambar 2.11 <i>Manifestation Deployment Diagram</i>	25
Gambar 2.12 <i>Target Deployment</i>	26
Gambar 2.13 Node.....	26
Gambar 2.14 Jalur Komunikasi.....	27
Gambar 2.15 <i>Simple State</i>	28
Gambar 3.1 <i>Deployment Diagram</i> Alat Kendali Lampu.....	31
Gambar 3.2 <i>Statechart Diagram</i> Alat Kendali Lampu.....	32
Gambar 3.3 Skematik Diagram Alat Kendali Lampu.....	33
Gambar 4.1 Papan Arduino UNO.....	34
Gambar 4.2 Sensor Suara FC-04.....	34
Gambar 4.3 Sensor Cahaya LDR.....	35
Gambar 4.4 Relay.....	35
Gambar 4.5 Alat Kendali Lampu Berbasis Arduino.....	36

DAFTAR TABEL

Tabel 2.1 Tabel Spesifikasi Arduino UNO.....	18
Tabel 4.1 Pembacaan Sensor Suara FC-04.....	36
Tabel 4.2 Pembacaan Sensor Cahaya LDR.....	37
Tabel 4.3 Pengujian dan Analisa Alat Kendali Lampu.....	37

DAFTAR SIMBOL

Deployment Diagram

No.	Simbol	Keterangan
1.	Node 	Mendeskrripsikan sumber daya komputasi, minimal memiliki <i>memory</i> dan kemampuan untuk memproses.
2.	Komponen 	Komponen dari sistem.
3.	Link 	Relasi antar <i>node</i> .
4.	<i>Provided Interface</i> 	Mendeskrripsikan keterikatan kebutuhan antara komponen dengan <i>node</i> .
5.	<i>Required Interface</i> 	Mendeskrripsikan keterikatan kebutuhan antara komponen dengan <i>node</i> .

Statechart Diagram

No.	Simbol	Keterangan
1.	State 	Kondisi selama objek menjalankan beberapa aktivitas.
2.	<i>Start</i> 	Bagaimana objek diawali.
3.	<i>End</i> 	Bagaiman objek berakhir.
4.	<i>Transisi</i> 	Sebuah kejadian yang memicu sebuah state objek dengan memperbaharui satu atau lebih nilai atributnya.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Saat ini, kemajuan teknologi berkembang sangat pesat diikuti dengan kebutuhan manusia yang ingin membuat segala sesuatunya serba cepat dan praktis. Perkembangan teknologi seperti ini membuat manusia melakukan usaha untuk mendapatkan kemudahan dan kenyamanan dalam melakukan aktivitas sehari-hari dengan mengembangkannya. Sistem otomatisasi merupakan sistem yang bekerja secara otomatis dan praktis sehingga sistem ini tidak perlu pengawasan manusia. Sistem otomatisasi dapat ditemukan dalam berbagai kehidupan sehari-hari antara lain sistem otomatisasi pada rumah (*Home Automation*), perkantoran, bangunan, sekolah dan yang lainnya.

Salah satu sistem otomatisasi yang akan diteliti adalah sistem yang dapat diterapkan di rumah-rumah yaitu sistem yang dapat menyalakan dan mematikan lampu dengan tepukan tangan. Melalui pengembangan sistem ini diharapkan penghuni rumah memiliki pengalaman yang nyaman, menyenangkan, dan memudahkan penghuni rumah. Dalam beberapa hal konsep rumah seperti ini sangat membantu khususnya bagi para orang tua yang sudah lanjut usia dan susah untuk berjalan, sehingga diharapkan akan dapat memberikan kemudahan dalam melakukan aktivitas kehidupan sehari-hari khususnya ketika ingin menyalakan/mematikan lampu.

Maka dari itu penulis membuat sebuah sistem tersebut, yaitu sistem yang dapat menyalakan dan mematikan lampu dengan perintah suara yang memiliki fungsi seperti saklar. Dengan demikian jika penghuni rumah ingin menyalakan dan mematikan lampu tidak harus lewat saklar tetapi bisa dengan menggunakan tepukan tangan. Sistem yang serupa tersebut telah dirancang dan diteliti oleh Sinta, et.al (2018) dalam jurnal yang berjudul "*Sistem Buka Tutup Pintu Otomatis Berbasis Suara Manusia*". Hasil dari pengujian pada alat yang dilakukan oleh orang yang berwenang dalam jurnal tersebut adalah (suara telah terekam pada program) besar presentase tingkat keberhasilannya adalah 95% untuk kata buka dan 90% untuk kata

tutup, sedangkan untuk pemberian perintah oleh orang lain rata – rata presentase keberhasilannya adalah 13% untuk kata buka dan 4% untuk kata tutup. Sistem yang serupa juga telah dirancang dan diteliti oleh Kinasih, et.al (2018) dalam jurnal yang berjudul “*Pengontrolan Ayunan Bayi Otomatis Dengan Mendeteksi Sensor Suara Menggunakan Mikrokontroler Arduino*”. Hasil dari pengujian pada alat yang dilakukan oleh orang yang berwenang dalam jurnal tersebut adalah sistem akan mulai bekerja jika sensor suara dapat dengan baik mendeteksi input suara yang nantinya akan mengaktifkan penanda bahwa terdeteksi input suara dan mengayun secara otomatis, sensor suara akan merespon suara dengan jarak tertentu sesuai dengan apa yang telah disetting sebelumnya di sensor tersebut, ketika sensor telah mendeteksi suara tangisan bayi, maka ayunan akan bergerak/bergoyang selama 1 menit sesuai durasi yang sudah diprogram ke arduino.

Berdasarkan latar belakang di atas, penulis tertarik untuk membuat alat pengendali lampu tersebut dan memilih judul “**PERANCANGAN ALAT KENDALI LAMPU MENGGUNAKAN SENSOR SUARA FC-04 BERBASIS ARDUINO UNO.**”

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, maka rumusan masalah yang didapatkan yaitu:

1. Bagaimana cara membuat alat yang dapat mengendalikan nyala dan mati lampu menggunakan tepukan tangan?
2. Bagaimana pengaruh tepukan tangan agar dapat menyalakan dan mematikan lampu?
3. Bagaimana pengaruh *noise* agar lampu tidak menyala dan mati ketika terdapat banyak tepukan tangan?

1.3 Tujuan

Adapun tujuan penelitian yang hendak dicapai dalam penulisan tugas akhir ini adalah:

1. Untuk memenuhi salah satu syarat kelulusan program pendidikan sarjana di Sekolah Tinggi Manajemen Informatika dan Komputer (STMIK) LIKMI Bandung, di bidang Informatika.
2. Untuk mempermudah pengguna dalam menyalakan dan mematikan lampu tanpa harus menekan saklar.

1.4 Batasan Masalah

Adapun ruang lingkup permasalahan pembahasan laporan Tugas Akhir ini akan dibatasi pada :

1. Sensor yang digunakan untuk mendeteksi suara tepukan tangan adalah sensor suara FC-04.
2. Sensor cahaya digunakan untuk mendeteksi cahaya lampu adalah sensor cahaya LDR (*Light Dependent Resistor*).
3. Suara yang digunakan untuk menyalakan lampu yaitu suara yang berasal dari tepukan tangan dengan ketentuan satu kali tepukan untuk menyalakan dan dua kali tepukan untuk mematikan lampu.

1.5 Kegunaan Hasil

Kegunaan hasil yang diharapkan dalam pembuatan permodelan dalam penelitian ini adalah:

1. Merancang dan membuat sebuah *prototype* untuk menyalakan/mematikan lampu menggunakan perintah suara melalui tepukan tangan yang fungsinya sama seperti saklar *on/off*.
2. Untuk mempermudah pengguna agar tidak perlu menyalakan/mematikan lampu melalui saklar.

1.6 Metodologi Penelitian

Tugas akhir ini dibuat dengan tujuan untuk mendapatkan hasil yang efektif dengan cara yang praktis dan berdasarkan dengan realita yang dihasilkan dari mengumpulkan, menyajikan, serta menganalisis data-data tersebut. Pengumpulan data yang dilakukan oleh penulis yaitu:

1. Studi Pustaka

Studi pustaka ini dilakukan dengan cara membaca buku, memahami, dan mempelajari buku dan *e-book* yang berhubungan dengan permodelan yang dibuat, sebagai landasan teori dari penulisan tugas akhir ini.

2. Perancangan Alat

Mengaplikasikan teori yang didapat dari studi pustaka, sehingga dapat tersusun sebuah sistem, untuk bagian perangkat keras maupun perangkat lunak.

3. Pengujian Alat

Menguji dan menganalisis hasil perancangan, dan menyimpulkan apakah alat tersebut sudah berfungsi sesuai dengan yang direncanakan.

1.7 Sistematika Penulisan

Untuk memudahkan pembahasan dalam tugas akhir ini, penulis menyusun penelitian dengan sistematika:

BAB I PENDAHULUAN

Bab ini menguraikan tentang latar belakang, rumusan masalah, tujuan, batasan masalah, kegunaan hasil, metodologi penelitian, dan sistematika penulisan.

BAB II LANDASAN TEORI

Bab ini berisi tentang sumber teori resmi yang berhubungan dengan pemodelan yang dirancang seperti bunyi, sensor, sensor suara FC-04, *analog to digital converter*, mikrokontroler, mikrokontroler Atmega328P, arduino, arduino uno, arduino IDE,

bahasa c, rekayasa perangkat lunak, *unified modeling language*, *deployment diagram*, *class diagram*, dan *statechart diagram* .

BAB III METODOLOGI PENELITIAN

Bab ini berisi tentang gambaran umum rancangan perangkat keras dan perangkat lunak, tahapan-tahapan pemodelan yang dirancang seperti *deployment diagram*, *statechart diagram*, dan diagram skematik.

BAB IV IMPLEMENTASI DAN PENGUJIAN

Bab ini berisi tentang implementasi dan pengujian dari alat yang sudah dirancang.

BAB V KESIMPULAN DAN SARAN

Bab ini berisi tentang kesimpulan yang dapat diambil dari hasil penelitian serta perancangan dan saran untuk pengembangan pemodelan lebih lanjut.

BAB II

LANDASAN TEORI

2.1 Bunyi

Suara atau bunyi dihasilkan oleh suatu benda yang bergetar. Asriwati (2017:67) menegaskan, "*pada hakekatnya suara dan bunyi adalah sama. Hanya saja kata suara dipakai untuk makhluk hidup, sedangkan bunyi di pakai untuk benda mati*".

Menurut Asriwati (2017:67) pada buku "*Fisika Kesehatan dalam keperawatan*" suara yaitu:

"Suara adalah fenomena fisik yang dihasilkan oleh getaran benda, getaran suatu benda yang berupa sinyal analog dengan amplitudo yang berubah secara kontinyu terhadap waktu".

Sehingga dapat disimpulkan bahwa, suara adalah fenomena fisik yang dihasilkan oleh getaran melalui benda oleh makhluk hidup atau benda mati yang berubah secara kontinyu terhadap waktu dan dapat didengar. Gelombang suara atau bunyi bervariasi dan memiliki media perantara seperti udara. Bunyi atau suara dapat diciptakan oleh getaran dari suatu obyek atau benda, yang menyebabkan udara disekelilingnya bergetar.

Berdasarkan frekuensinya, bunyi dapat dibedakan beberapa jenis antara lain:

1. Bunyi infrasonik

Bunyi infrasonik (*infrasound*) adalah bunyi atau suara yang memiliki frekuensi kurang dari 20 Hz yang memiliki pengertian bahwa 20 getaran dalam satuan detik. Bunyi infrasonik tidak dapat didengar oleh pendengaran telinga manusia dikarenakan frekuensi yang terlalu rendah, tetapi beberapa hewan dapat mendengar bunyi infrasonik seperti anjing, jangkrik, hiu dan hewan-hewan lainnya.

2. Bunyi audiosonik

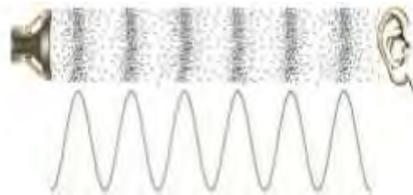
Bunyi audiosonik adalah bunyi atau suara yang memiliki frekuensi antara 20 Hz sampai 20.000 Hz. Pada frekuensi tersebut bunyi audiosonik dapat didengar oleh pendengaran telinga manusia. Getaran udara yang melalui media perantara udara yang kemudian

menyebabkan gendang telinga manusia bergetar dan oleh otak dipahami atau dikenal sebagai suara.

3. Bunyi ultrasonik

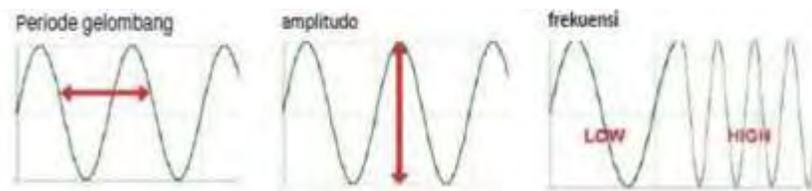
Bunyi ultrasonik adalah bunyi atau suara yang memiliki frekuensi lebih dari 20.000 Hz. Bunyi ultrasonik tidak dapat di dengar oleh manusia, tetapi beberapa hewan dapat mendengar bunyi ultrasonik seperti kelelawar, lumba-lumba, dan anjing.

Gelombang suara yang bergetar melalui udara memiliki cara kerja yang sama seperti cara kerja gelombang air melalui air. Dalam hal tersebut, karena gelombang air dapat dilihat dengan mudah membuat para peneliti menggunakan gelombang air sebagai analogi untuk mengilustrasikan bagaimana gelombang suara merambat menuju telinga manusia. Perambatan gelombang suara dapat dilihat pada gambar 2.1.



Gambar 2.1
Perambatan sinyal audio (Gelombang suara)
(Sumber: Rusdi dan Yani, 2018:28)

Titik-titik hitam pada gambar 2.1 yang dapat dilihat menunjukkan molekul udara. Sebagaimana getaran yang menyebabkan molekul disekitarnya mengalami suatu getaran dalam pola atau bentuk tertentu yang dapat digambarkan dengan bentuk gelombang sesuai gambar 2.1. Getaran udara tersebut menyebabkan gendang telinga bergetar dengan pola yang sama. Gelombang suara dapat juga ditentukan dalam suatu grafik standar dengan garis x dan y seperti pada gambar 2.2.



Gambar 2.2
Grafik Gelombang Suara

(Sumber: Rusdi dan Yani, 2018:28)

Gambar 2.2 tersebut memungkinkan untuk menggambarkan gelombang dengan dengan garis x dan y yang kemudian menghasilkan kurva yang dikenal sebagai bentuk gelombang dalam sudut pandang matematis. Gelombang memiliki tiga sifat meliputi:

1. Periode Gelombang

Periode gelombang adalah waktu yang diperlukan untuk menempuh satu gelombang dalam satuan detik.

2. Amplitudo Gelombang

Amplitudo gelombang adalah kuat lemahnya bunyi yang dipengaruhi oleh suatu gelombang berupa simpangan maksimum. Semakin besar getaran dari sumber, semakin kuat juga bunyi di dengar. Kuat lemahnya suatu bunyi bergantung pada besar dan kecilnya amplitudo.

3. Frekuensi Gelombang

Frekuensi gelombang adalah banyak gelombang yang terjadi dalam satu detik, diukur dengan satuan Hertz (Hz). Getaran gelombang suara/bunyi semakin cepat, maka frekuensi semakin tinggi.

Suara atau bunyi merambat melalui suatu perantara yaitu udara, air dan material lainnya. Satu satunya suara tidak dapat merambat adalah ruang hampa udara yang berada luar angkasa.

2.2 Sensor

Sensor menurut para ahli sebagai berikut:

“Sensor adalah komponen yang dapat digunakan untuk mengkonversi suatu besaran tertentu menjadi satuan analog sehingga dapat dibaca oleh suatu rangkaian elektronik” (Ekojono et al., 2017:14).

Pengertian sensor menurut Syam dalam buku yang berjudul *“Dasar-Dasar Teknik Sensor”* adalah:

“Sensor adalah elemen sistem yang secara efektif berhubungan dengan proses dimana suatu variabel sedang diukur dan menghasilkan suatu keluaran dalam bentuk tertentu tergantung pada variabel inputnya, dan dapat digunakan oleh bagian sistem pengukuran yang lain untuk mengenali nilai variabel tersebut” (Syam, 2013:13).

Kesimpulan yang dapat ditarik berdasarkan definisi sensor dari beberapa sumber, sensor adalah elemen sistem yang secara aktif atau alat untuk melacak, mendeteksi, dan mengkonversi suatu objek yang di baca dengan rangkaian elektronik.

Adapun beberapa macam sensor yang sering digunakan untuk mengukur :

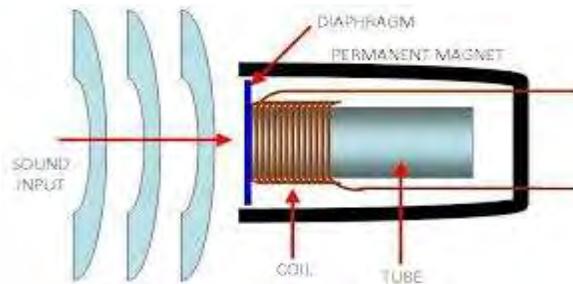
1. Temperatur
2. *Strain*
3. Suara
4. Getaran
5. Posisi dan *Displacement* (perpindahan)
6. Tekanan
7. Gaya
8. Cahaya

Menurut Aryanti dalam jurnal *“Implementasi Sensor Suara Sebagai Pengendali Gerakan Robot Penari Humanoid dengan ATMEGA 8535”* Sensor suara adalah sebuah alat yang mampu mengubah gelombang Sinusioda, suara menjadi gelombang sinus energi listrik (Aryanti et al., 2016:2).

2.2.1 Sensor Suara FC-04

Menurut Kharisma, Oktaf Brilian dan Hanif Burhanuddin Putra Utama (2018:115) sensor suara FC-04 adalah teknologi yang mampu mengidentifikasi suara dengan keluarannya berupa bilangan biner (1 0) atau bernilai *high* dan *low*. Sensor suara FC-04 terdiri dari *microphone*. *Microphone* adalah sensor yang digunakan untuk mengukur suara, tapi terdapat beberapa tipe dari sensor suara mikrofon, yaitu:

1. Kondensor Mikrofon



Gambar 2.3
Prinsip kerja Kondensor *Microphone*
(Sumber : <https://www.mediacollege.com/audio/microphones/condenser.html>)

Mikrofon kondensor adalah sensor suara yang paling umum digunakan. Jenis sensor ini disebut juga *prepolarized* (yang berarti bahwa sumber daya termasuk dalam mikrofon) atau eksternal terpolarisasi. Eksternal mikrofon kondensor terpolarisasi membutuhkan sumber daya tambahan, yang menambah biaya untuk proyek-proyek. Mikrofon *Prepolarized* lebih disukai di lingkungan lembab di mana komponen *power supply* bisa rusak, dan mikrofon kondensor eksternal terpolarisasi lebih disukai di lingkungan suhu tinggi.

2. *Piezoelectric Microphones*

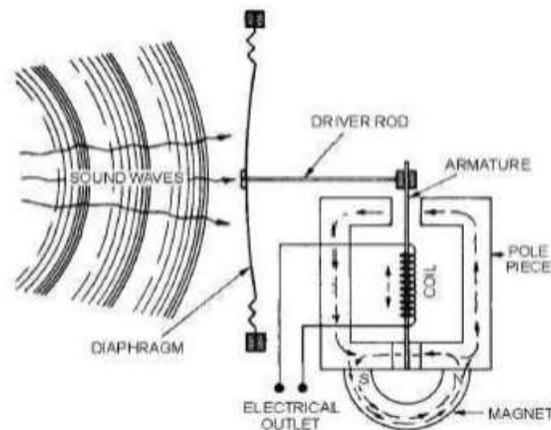
Robust Mikrofon piezoelektrik digunakan untuk aplikasi *shock* dan pengukuran suara dengan tekanan ledakan. Jenis sensor mikrofon ini tahan lama dapat mengukur tinggi amplitudo (desibel) rentang tekanan. Kerugian jenis sensor ini adalah tingkat kebisingan yang tinggi dapat diukur oleh system sensor ini.



Gambar 2.4
Struktur Piezoelectric *Microphone*
(Sumber : <http://hyperphysics.phy-astr.gsu.edu/hbase/Audio/mic3.html>)

3. Dinamis / *Magnetic Mikrofon / Dynamic / Magnetic Microphones*

Mikrofon magnet menggunakan gerakan magnetis menginduksi muatan listrik dengan cara yang membuat mereka tahan terhadap air, tapi jelas mikrofon ini tidak berguna dalam lingkungan yang sangat magnetik.



Gambar 2.5
Magnetic Microphones
(Sumber : <http://electriciantraining.tpub.com/14184/css/Figure-1-36-Magnetic-Microphone-Action-53.htm>)

Aplikasi jenis sensor ini banyak diaplikasikan untuk bidang musik. Mikrofon dinamis (*mics*) terdiri dari kumparan suara melekat pada diafragma ringan yang tergantung di sebuah medan magnet. Ketika suara menyebabkan diafragma bergetar, kumparan bergerak

dalam medan magnet, dan akibatnya tegangan listrik bolak kecil dihasilkan yang sebanding dengan suara diterima.

2.2.2 Sensor Cahaya LDR

Sensor cahaya LDR menurut Sujarwata dalam buku "Belajar Mikrokontroler BS2SX Teori, Penerapan dan Contoh pemrograman PBasic" yaitu:

"Sensor cahaya LDR (Light Dependent Resistor) adalah suatu bentuk komponen sensor yang mempunyai perubahan resistansi yang besarnya tergantung pada intensitas cahaya yang mengenainya" (Sujarwata, 2012:36).

Nilai resistansi LDR akan berubah-ubah sesuai dengan intensitas cahaya yang diterima. Jika LDR tidak terkena cahaya nilai maka nilai tahanan akan menjadi besar sekitar $10M\Omega$ dan jika terkena cahaya nilai tahanan akan menjadi kecil sekitar $1k\Omega$ (Novianti et al., 2012:2).

Sensor cahaya LDR ini memiliki cara kerja yaitu mengubah sebuah energi foton menjadi elektron dan kebanyakan 1 foton dapat berubah menjadi 1 elektron. Adapun beberapa komponen yang dimiliki oleh sensor cahaya seperti LDR (*Light Dependent Resistor*), *Photodiode*, dan *Photo Transistor*.



Gambar 2.6
Sensor Cahaya LDR

(Sumber : <http://www.jogjarobotika.com/photodiode-ldr-ir-led/700-sensor-cahaya-ldr-light-dependent-resistor-5mm.html>)

2.3 Analog to Digital Converter (ADC)

ADC (*Analog to Digital Converter*) adalah sebuah alat yang mengubah sinyal/besaran analog menjadi satuan digital sesuai ketelitian dari perangkat ADC (banyak bit dari ADC).(Hardana, 2018:117)

Pada proses pengubah sinyal analog menjadi sinyal digital, ADC memiliki beberapa langkah seperti:

1. Sampling

Sampling merupakan pengambilan data dalam satuan detik tertentu. Semakin banyak titik sampling dalam waktu yang bersamaan semakin teliti data digital yang dihasilkan.

2. Kuantisasi

Kuantisasi merupakan proses pemetaan input menjadi besaran digital.

3. Pengkodean

Pengkodean merupakan proses untuk menghasilkan nilai digital dari proses kuantisasi (proses sebelumnya).

ADC memiliki 2 karakter prinsip yaitu kecepatan sampling dan resolusi. Kecepatan sampling suatu ADC menyatakan seberapa sering sinyal analog dikonversikan ke bentuk sinyal digital pada selang waktu tertentu. Kecepatan sampling biasanya dinyatakan dalam *sample per second* (SPS). Resolusi ADC menentukan ketelitian nilai hasil konversi ADC. Sebagai contoh: ADC 8 bit akan memiliki output 8 bit data digital yang memiliki pengertian bahwa sinyal input dapat dinyatakan dalam 255 ($2^n - 1$) nilai diskrit. ADC 12 bit memiliki 12 bit output data digital, ini berarti sinyal input dapat dinyatakan dalam 4096 nilai diskrit.

2.4 Mikrokontroler

Mikrokontroler dikembangkan pertama kali oleh Texas Instrument pada tahun 1971 yang memiliki seri TMS-1000. Mikrokontroler tersebut merupakan mikrokontroler berbentuk 4bit yang memiliki sebuah *chip* dan dilengkapi dengan RAM (*Random Access Memory*) dan ROM (*Read Only Memory*). Saat ini banyak ditemui tipe-tipe mikrokontroler dimulai dari 8 bit sampai dengan 64 bit yang dikembangkan oleh *vendor* dengan berbagai fitur-fitur agar memudahkan pengguna untuk merancang sebuah sistem. Dengan mikrokontroler tersebut dapat membuat sebuah sistem untuk keperluan sehari hari, hobi, belajar dan lain lain.

“Mikrokontroler adalah sebuah sistem mikroprosesor lengkap yang terkandung di dalam sebuah chip”(Sujarwata, 2012:3).

“Mikrokontroler merupakan chip mikrokomputer secara fisik berupa sebuah IC (*Integrated Circuit*)” (Dharmawan, 2017:1).

Berdasarkan definisi yang dikemukakan dapat disimpulkan bahwa mikrokontroller adalah sebuah sistem komputer yang dibuat kedalam sebuah *chip* tunggal berukuran sangat kecil berupa sebuah IC. Pada umumnya mikrokontroler digunakan dalam ruang lingkup sistem yang lebih murah, kecil, efisien dan tidak membutuhkan perhitungan yang sangat kompleks seperti dalam aplikasi di Komputer (*Personal Computer*). Mikrokontroler sering ditemukan dalam peralatan-peralatan dalam kehidupan sehari-hari seperti *microwave, oven, keyboard, CD Player, VCR, Remote Control*, robot dan lainnya. Mikrokontroller memiliki bagian-bagian utama yaitu seperti CPU (*Central Processing Unit*), ROM (*Read Only Memory*), port I/O (*Input Output*) dan RAM (*Random Access Memory*). Selain bagian-bagian utama tersebut, terdapat beberapa perangkat keras (*hardware*) yang dapat digunakan untuk banyak keperluan yang sangat berguna seperti melakukan interupsi perintah, melakukan komunikasi serial, dan melakukan pencacahan untuk hitungan. Mikrokontroler tertentu bahkan menyatakan memiliki bagian yang mempunyai fungsi seperti *USB Controller, CAN (Controller Area Network)*, ADC (*Analog to Digital Converter*), dan lainnya.

Mikrokontroler bekerja berdasarkan program (perangkat lunak) yang ditanamkan di dalamnya dan program tersebut dibuat sesuai dengan aplikasi yang diinginkan. Aplikasi mikrokontroler normalnya terkait pembacaan data dari luar atau pengontrolan peralatan diluarnya.

Mikrokontroler AVR (*Advanced Versatile RISC*) merupakan mikrokontroler yang memiliki arsitektur berbasis RISC (*Reduced Instruction Set Computing*) 8 bit. AVR dalam satu siklus tertentu menjalankan sebuah instruksi tunggal dan memiliki struktur masukan/keluaran yang sangat lengkap sehingga pemakaian komponen dari luar (*external*) dapat dikurangi sebaik mungkin. Mikrokontroler AVR menggunakan bentuk arsitektur *Harvard* yang memiliki

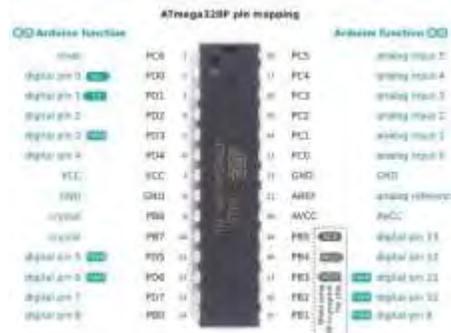
fungsi untuk melakukan pemisahan antara memori untuk data dan memori untuk kode program sehingga mikrokontroler dapat bekerja tanpa adanya masalah.

2.4.1 Mikrokontroler ATmega328P

Mikrokontroler ATmega328P adalah *chip* mikrokontroler 8-bit berbasis AVR-RISC buatan Atmel dan arsitektur RISC (*Reduce Instruction Set Computer*) yang membuat proses pengekseskuan data-data lebih cepat dibandingkan dengan arsitektur CISC (*Completed Instruction Set Computer*).

Mikrokontroler Atmega328P ini memiliki fitur-fitur sebagai berikut:

1. Memiliki besar 2KB (*Kilo Bytes*) SRAM (*Static Random Access Memory*).
2. Memiliki EEPROM (*Electrically Erasable Programmable Read Only Memory*) untuk tempat penyimpanan sebesar 1KB data semi permanen karena EEPROM ini tetap dapat menyimpan data meskipun tidak ada daya listrik pada mikrokontroler.
3. Dalam satu siklus *clock* dapat melakukan eksekusi sebanyak 130 macam instruksi.
4. Memiliki pin I/O digital sebanyak 14 buah pin 6 diantaranya merupakan PWM (*Pulse Width Modulation*) output.
5. Dengan *clock* 16 MHz kecepatan mencapai 16 MIPS.
6. 32 KB *Flash memory* dan pada arduino memiliki *bootloader* yang menggunakan 2 KB dari flash memori sebagai *bootloader*.
7. 32 x 8-bit register serba guna.



Gambar 2.7

Konfigurasi pin mikrokontroler Atmega328P

(Sumber : <http://www.labelektronika.com/2017/02/arduino-uno-mikrokontroler-atmega-328.html>)

ATMega328P juga mempunyai 3 jenis memori yang masing-masing mempunyai fungsi yang berbeda yaitu memori program, memori data, dan memori EEPROM. Fungsi dari masing-masing memori tersebut adalah :

1. Memori Program

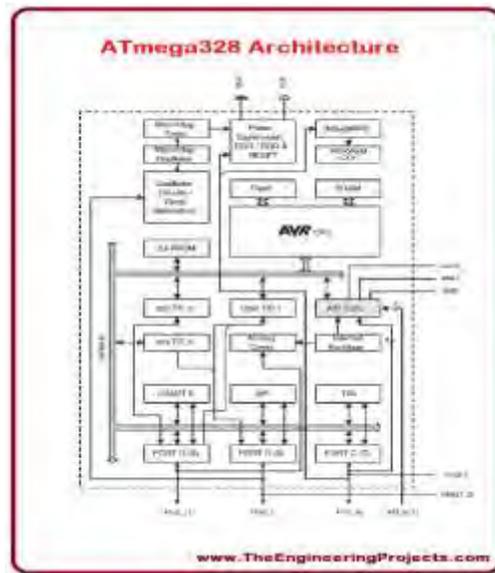
Berkapasitas 8 Kb yang dipetakan pada alamat 0x0000 – 0x3FFF dengan lebar data dari masing-masing alamat sebesar 32 bit. Memori program dibagi menjadi 2 bagian yaitu bagian untuk program boot dan bagian untuk program aplikasi.

2. Memori Data

Memori data dibagi menjadi 3 bagian register yaitu 32 register serbaguna, 64 register I/O yang jika menggunakan instruksi LD atau ST maka dapat diakses sebagai bagian dari RAM dan bisa juga diakses sebagai I/O, dan 2048 byte register SRAM.

3. Memori EEPROM

Memori *electronics erasable programmable read only memory* (EEPROM). EEPROM sebesar 1 Kb dan terpisah dari memori program dan memori data. Memori EEPROM memiliki waktu esekusi lebih lama dari mengakses data dari SRAM, hal ini terjadi karena untuk mengakses memori EEPROM adalah seperti melakukan akses data external. Memori EEPROM sendiri keterbatasan seperti hanya dapat diakses melalui register-register I/O seperti alamat EEPROM, register EEPROM data, dan register EEPROM *control*.



Gambar 2.8
Blok Diagram ATmega 328P

(Sumber : <https://www.theengineeringprojects.com/2017/08/introduction-to-atmega328.html>)

2.4.2 Arduino

Arduino menurut para ahli yaitu:

“Arduino adalah suatu open-source platform elektronik yang berbasis kemudahan penggunaan (easy to use) baik hardware maupun software” (Ahyadi, 2018:1).

“Arduino merupakan perangkat keras sekaligus perangkat lunak yang memungkinkan siapa saja melakukan pembuatan prototype suatu rangkaian elektronika yang berbasis mikrokontroler dengan mudah dan cepat” (Kadir, 2016:2).

“Arduino is an open source physical computing platform based on a simple input/output (I/O) board and a development environment that implements the Processing language(www.processing.org)” (Banzi, 2011:1).

Arduino dapat diartikan suatu rangkaian elektronika berupa perangkat keras dan perangkat lunak berbasis *open source platform* elektronik yang mempunyai masukan/keluaran berbasis mikrokontroler.

2.4.3 Arduino Uno

Arduino Uno adalah Arduino *Board* yang menggunakan mikrokontroler ATmega328 (Parinduri et al., 2017:32). Arduino UNO memiliki 14 pin digital masukan/keluaran (6 di antaranya dapat digunakan sebagai keluaran PWM/*Pulse Width Modulation*), 6 masukan analog, *power jack*, osilator Kristal 16 MHz, koneksi USB, *ICSP header*, dan tombol reset. Arduino UNO dapat memuat semua yang diperlukan untuk membantu mikrokontroler menjalankan fungsinya dan mudah menghubungkannya ke sebuah komputer dengan menggunakan sebuah kabel USB.



Gambar 2.9
Arduino UNO

(Sumber : <https://store.Arduino.cc/usa/Arduino-uno-rev3>)

Tabel 2.1
Tabel Spesifikasi Arduino UNO

Paramater	Spesifikasi
Tegangan Operasi	5V
Tegangan Input (Rekomendasi)	7-12V
Tegangan Input (Limit)	6-20V
Pin Digital I/O	14(6 PinOutput PWM)
Pin Digital PWM I/O	6
Pin Analog Input	6
Arus DC tiap Pin I/O	20 mA
Arus DC Pin 3.3V	50 mA
Flash Memory	32 KB (ATmega328P) dimana 0.5 KB digunakan untuk bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
LED_BUILTIN	13
Panjang	68,6 mm
Lebar	53,4 mm
Berat	25 g

(Sumber : <https://store.Arduino.cc/usa/Arduino-uno-rev3>)

Kelebihan dari Arduino UNO adalah:

1. Board Arduino juga tidak membutuhkan *downloader* untuk *download* program yang telah dibuat dari komputer ke mikrokontroler.
2. Memiliki sebuah modul yang bisa dipakai dan modul dapat langsung dipasang pada *board* Arduino.
3. Pengembangan *project* mikrokontroler akan menjadi lebih mudah dan menyenangkan karena menghubungkan *board* Arduino ke komputer atau laptop melalui kabel USB.
4. Didukung oleh Arduino IDE dengan bahasa pemrograman dengan *library* yang lengkap.

Berikut merupakan penjelasan beberapa bagian-bagian penting yang berada pada papan (*board*) Arduino UNO :

1. Mikrokontroler Atmega328P adalah “otak” dari papan Arduino. Bagian ini adalah sebuah IC (*Integrated Circuit*) yang dipasangkan ke kepala soket (*header socket*) sehingga mikrokontroler dapat dilepas dari papan Arduino UNO.
2. Pin digital adalah pin untuk menerima atau mengirim isyarat digital. Isyarat 1 (sering dinyatakan dengan *HIGH*) direpresentasikan dalam bentuk tegangan 5V dan isyarat 0 (sering dinyatakan dengan *LOW*) diwujudkan dalam bentuk tegangan 0V. Nomor untuk pin digital berupa 0 hingga 13. Beberapa pin digital, yang dinamakan pin PWM dapat digunakan sebagai keluaran analog. Pin PWM ditandai dengan simbol “~”. Ada 6 pin PWM, yaitu pin 2, 5, 6, 9, 10, dan 11.
3. Pin analog adalah pin yang dipakai untuk menerima nilai analog. Jika dinyatakan dalam tegangan, nilai analog akan berkisar antara 0V hingga 5V. Nilai seperti 1.0 atau 2.5 dimungkinkan di pin ini.
4. Pin sumber tegangan adalah pin yang memberikan catu daya kepada pin-pin lain yang membutuhkannya.
5. LED yang tersedia berjumlah 4. Fungsi masing-masing adalah seperti:
 - a. ON akan menyala kalau papan Arduino diberi sumber tegangan.

- b. RX dan TX menyatakan data sedang dikirim dan diterima oleh papan Arduino.
 - c. L adalah LED yang terhubung ke pin 13.
6. Tombol reset akan membuat sketch dijalankan ulang atau seperti *reboot* pada komputer. Ketika adanya instruksi yang diberikan oleh Arduino tidak normal maka tombol reset yang ditekan akan membuat sistem melakukan reset dan kemudian arduino akan aktif kembali.
7. Konektor USB (*Universal Serial Bus*) berfungsi sebagai penghubung mikrokontroler ke PC (*Personal Computer*) yang memiliki fungsi sebagai pemasok tegangan bagi papan Arduino. Konektor catu daya berfungsi sebagai penghubung antara mikrokontroler dan sumber tegangan eksternal. Hal tersebut diperlukan jika konektor USB tidak dihubungkan ke PC. Adaptor AC-ke-DC atau baterai dapat dihubungkan ke konektor USB. Konektor USB dapat menerima tegangan sebesar 7V hingga 12V.

2.5 Arduino IDE

Arduino IDE (*Integrated Development Enviroment*) adalah *software* pendukung mikrokontroller arduino yang memiliki fungsi untuk membuat sebuah program, meng-compile menjadi kode dalam bentuk biner dan mengunggah ke dalam mikrokontroler. Melalui software inilah dapat melakukan pengembangan arduino. Bahasa pemrograman yang digunakan adalah bahasa pemrograman C. Bahasa C menghasilkan file kode objek yang sangat kecil dan dieksekusi dengan sangat cepat, karena itu bahasa C sering digunakan dalam pada sistem operasi dan pemrograman mikrokontroler. Arduino menyediakan library yang menggunakan bahasa C++ yang sederhana dan target yang sudah ditetapkan dalam suatu fungsi untuk memudahkan pengguna dalam membuat program. Bahasa C adalah bahasa yang dapat digunakan di beberapa *platffom (multi-platform)* seperti windows, unix dan linux tanpa mengalami mengubah *source code* sebelumnya, karena itu pada umumnya perangkat lunak Arduino pun dapat dijalankan pada beberapa sistem operasi. Pemrograman untuk mikrokontroler arduino menggunakan bahasa C yang sering disebut dengan *sketches*. Sketsa-sketsa ini ditulis di dalam teks *editor* dan disimpan dengan file yang berformat .ino.

Software Arduino IDE memiliki beberapa menu dengan fungsi yang berbeda-beda. Beberapa menu yang ada pada software Arduino IDE adalah *File*, *Edit*, *Sketch*, *Tools* dan *Help*. Fungsi dari setiap menu tersebut adalah :

1. *File*:

Menu file mempunyai *submenu* seperti membuat *sketch* baru, menyimpan dan membuka *sketch*, membuka *preferences*, pilihan untuk keluar dari program dan beberapa pilihan lainnya.

2. *Edit*

Menu edit mempunyai pilihan seperti *copy*, *paste*, *select* dan pilihan lainnya.

3. *Sketch*

Menu *sketch* mempunyai pilihan *submenu* seperti *verify* yang berguna untuk memverifikasi *sketch* (Program) yang telah dibuat, *upload* adalah *menu* yang berfungsi untuk mengunggah *sketch* yang telah dibuat dan dikompilasi ke arduino, *Include Library* adalah *menu* yang digunakan untuk memilih *library* arduino yang ingin digunakan, *manage library* adalah *menu* yang digunakan untuk memperbaharui *library* dan untuk mengunduh *library*. Pembaruan *library* dapat dilakukan secara *offline* dengan *file* yang berekstensi .Zip.

4. *Tools*

Menu tools mempunyai beberapa pilihan *submenu*. *Submenu* yang biasa digunakan adalah *submenu* untuk memilih papan arduino yang akan digunakan, pemilihan *port COM* di mana arduino yang telah dipilih terhubung dengan komputer, *submenu programmer* yang berfungsi untuk mengunggah *sketch* kedalam arduino.

5. *Help*

Pada menu *help* terdapat *submenu* untuk mencari informasi. Dan terdapat juga tombol serial monitor yang terdapat diujung sebelah kanan dapat digunakan untuk melihat data karakter, angka, ataupun teks yang dikirimkan oleh arduino ke komputer.

2.6 Sound Meter

Sound Meter merupakan sebuah aplikasi *platform android* untuk menghitung atau menunjukkan nilai kebisingan/*desibel* dengan satuan (dB) pada suatu lingkungan dan menampilkan nilai yang diukur dalam berbagai bentuk. Fitur yang terdapat pada aplikasi *sound meter* yaitu :

1. Menunjukkan *desibel* dengan pengukur.
2. Menampilkan referensi tingkat kebisingan saat ini.
3. Menampilkan nilai *desibel* terendah, tertinggi dan rata-rata.
4. Menampilkan nilai *desibel* dengan grafik.
5. Dapat mengkalibrasi *desibel* untuk setiap perangkat.

Gambar 2.10 merupakan gambar aplikasi *sound meter* pada *handphone android*



Gambar 2.10
Sound Meter

(Sumber : <https://store.Arduino.cc/usa/Arduino-uno-rev3>)

2.7 Bahasa C

Bahasa C merupakan perkembangan bahasa dari bahasa BCPL (*Basic Combined Programming Language*) yang dikembangkan oleh Martin Richards pada tahun 1967. Kemudian bahasa pemrograman ini membuat Ken Thompson mengembangkan bahasa yang disebut bahasa B pada tahun 1970. Setelah pengembangan bahasa B tersebut lahirlah Bahasa C yang dikembangkan oleh Dennis Ritchie sekitar tahun 1970-an di *Bell Telephone Laboratories Inc.* Bahasa C digunakan pertama kali di *Computer Digital Equipment Corporation* PD-11 yang menggunakan operasi UNIX. Saat ini, bahasa C sangat berperan besar dalam kemajuan perkembangan perangkat lunak atau software karena digunakan dalam pemrograman mikrokontroler untuk menghasilkan kode objek yang sangat kecil dan dapat dieksekusi dengan cepat serta merupakan sebuah mutiplatform karena dapat diterapkan pada Windows, Unix serta Linux.

Kelebihan bahasa C menurut Fauzan dan Adiputri (2019:5) :

6. Bahasa C tersedia hampir di semua jenis komputer.
7. Kode bahasa C sifatnya *portable* dan fleksibel untuk semua jenis komputer.
8. C adalah bahasa yang terstruktur.
9. Proses *executable* program bahasa C lebih cepat.
10. Dukungan pustaka yang banyak.
11. Bahasa termasuk bahasa tingkat menengah.
12. Bahasa C hanya menyediakan sedikit kata-kata kunci, hanya terdapat 32 kata kunci.

Kekurangan bahasa C menurut Fauzan dan Adiputri (2019:5):

1. Banyaknya operator serta fleksibilitas penulis program yang kadang-kadang membingungkan pemakai.
2. Bagi pemula pada umumnya akan kesulitan menggunakan *pointer*.

2.8 Rekayasa Perangkat Lunak

Pengertian rekayasa perangkat lunak dalam buku "*Software Engineering*" adalah sebagai berikut (Yurindra, 2017:2):

"Rekayasa perangkat lunak adalah suatu disiplin ilmu yang membahas semua aspek produksi perangkat lunak, mulai dari tahap awal yaitu analisa kebutuhan pengguna, menentukan spesifikasi dari kebutuhan pengguna desain, pengkodean, pengujian, sampai pemeliharaan sistem setelah digunakan ".

Pengertian rekayasa perangkat lunak dalam buku "*Rekayasa Perangkat Lunak*" adalah sebagai berikut (Wicaksono, 2017:29):

"Rekayasa perangkat lunak adalah sebuah disiplin ilmu yang mencakup segala hal yang berhubungan dengan proses pengembangan perangkat lunak sejak dari tahap rancangan hingga tahapan implementasi serta pasca implementasi sehingga siklus hidup perangkat lunak dapat berlangsung secara efisien dan terukur".

Menurut Roger Pressman dalam bukunya "*Software Engineering*", perangkat lunak adalah:

"Software is : (1) instructions (computer programs) that when executed provide desired function and performance, (2) data structures that enable the programs to adequately manipulate information, and (3) documents that describe the operation and use of the programs." (Pressman, 2010:4).

Berdasarkan dari beberapa pengertian para ahli diatas bahwa rekayasa perangkat lunak adalah disiplin ilmu dalam pembuatan, pengembangan, penggunaan, pengujian dan pemeliharaan sistem perangkat lunak untuk memperoleh perangkat lunak yang terukur, bebas dari kesalahan, dan bekerja secara efisien.

2.9 Unified Modeling Language

Unified Modeling Language atau UML pertama kali dikembangkan oleh Grady Booch, Jim Rumbaugh, dan Ivars Jacobson pada pertengahan tahun 1990. UML adalah himpunan struktur dan teknik untuk pemodelan dan desain program berorientasi objek (OOP) serta aplikasinya. UML merupakan singkatan "*Unified Modelling Language*" yaitu suatu metode pemodelan secara visual untuk sarana perancangan sistem berorientasi objek (Haqi dan Setiawan, 2019:75).

2.9.1 *Deployment Diagram*

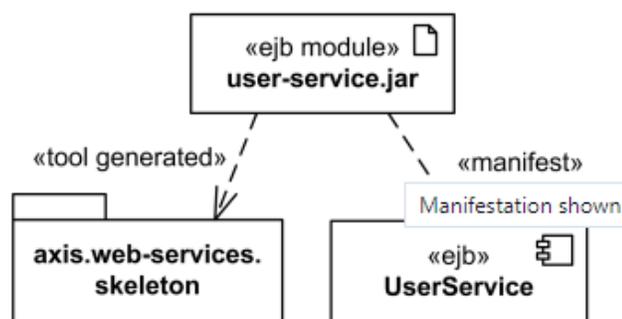
Menurut Nugroho (2010:227) “*deployment* diagram merupakan suatu model objek yang mendeskripsikan sebaran fisik/perangkat lunak pada terminologi bagaimana fungsionalitas ditebarkan di antara simpul-simpul komputasional”. *Deployment* diagram menunjukkan konfigurasi komponen-komponen (*hardware*) dalam proses eksekusi perintah aplikasi. Diagram *deployment* juga dapat digunakan untuk memodelkan hal-hal berikut (Dalis, 2017:3):

1. Sistem tambahan (*embedded sistem*) yang menggambarkan rancangan *device*, *node*, *hardware*.
2. Sistem *client* atau *server*.
3. Sistem terdistribusi murni.
4. Rekayasa ulang aplikasi.

Hal-hal yang perlu diperhatikan pada *deployment* diagram seperti:

1. *Manifestation*

Manifestation merupakan hubungan *abstraction* yang mepresentasikan *physical rendering* (hasil implementasi) satu atau lebih model element dari *artifact* atau pemanfaatan model element dari konstruksi *artifact* atau generasi *artifact*.



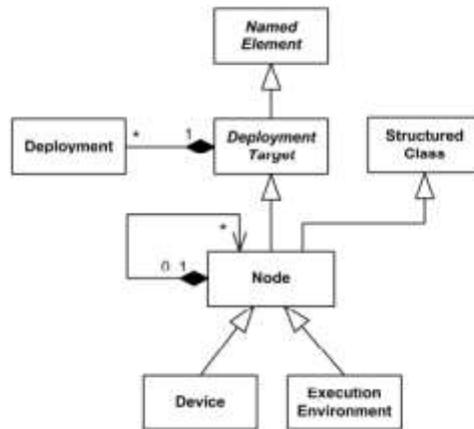
Gambar 2.11

Manifestation Deployment Diagram

(Sumber : <https://www.uml-diagrams.org/deployment-diagrams.html>)

2. Target *deployment*

Pada gambar 2.15 *artifact* diarahkan kepada target *deployment*. Target *deployment* merupakan lokasi *artifact* yang telah diarahkan. Target *deployment* memiliki kumpulan *deployment* yang mengarahkannya. Target *deployment* tidak memiliki notasi sendiri tetapi melihat notasi untuk sub kelasnya.



Gambar 2.12
Target *Deployment*

(Sumber : <https://www.uml-diagrams.org/deployment-diagrams.html>)

3. *Node*



Gambar 2.13
Node

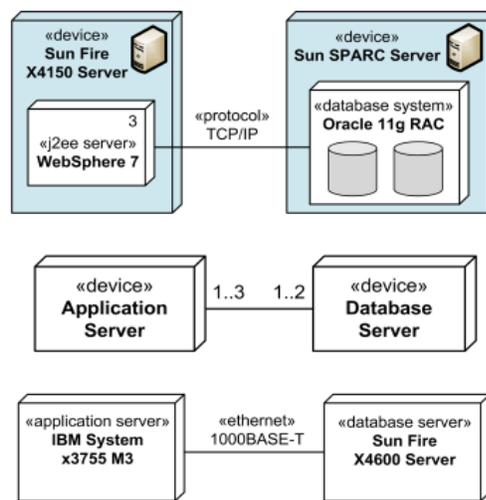
(Sumber : <https://www.uml-diagrams.org/deployment-diagrams.html>)

Node adalah target *deployment* yang merepresentasikan sumber daya komputasi (*represents computational resource*) dimana *artifact* bisa menggunakan untuk mengeksekusi. *Node* diasosiasikan dengan *deployment* dari *artifact* dan dapat diasosiasikan secara tidak langsung dengan elemen yang dapat dikemas (*packageable elements*) yang melibatkan ke dalam *manifestation* dari *artifact* yang dikerahkan ke *node*.

Node dapat saling berhubungan dengan jalur komunikasi (*communication path*), dapat digambarkan dengan bentuk kubus tiga dimensi, dan dapat digambarkan dalam bentuk visual atau gabungan antara *node* dan visual.

4. Jalur komunikasi.

Jalur komunikasi atau bisa disebut *communication path* merupakan hubungan antara dua target *deployment*, melalui pertukaran sinyal dan pesan. Jalur komunikasi digambarkan dengan bentuk garis utuh yang menghubungkan antara dua *node*.



Gambar 2.14
Jalur Komunikasi

(Sumber : <https://www.uml-diagrams.org/deployment-diagrams.html>)

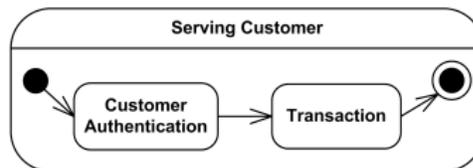
2.9.2 Statechart Diagram

Menurut Isa dan Hartawan (2017:141) *statechart* diagram adalah menggabungkan semua *state* (kondisi) yang dimiliki dari suatu objek dari suatu *class* dan keadaan yang menyebabkan *state* berubah. Pada umumnya *statechart* diagram menggambarkan sebuah kelas-kelas tertentu (satu *class* dapat memiliki lebih dari satu *statechart* diagram).

Kondisi-kondisi yang perlu diperhatikan pada *statechart* diagram seperti :

1. *Simple State*

Simple State adalah State yang tidak memiliki substates dan memiliki *compartments* seperti *name compartment*, *internal activities compartment*, dan *internal transitions compartment*. *Name compartment* memiliki fungsi sebagai pemberi keterangan (*string*) jika *state* tidak memiliki nama maka akan disebut sebagai *anonymous*. *Internal activities compartment* yang memiliki fungsi untuk menyimpan daftar aktifitas internal, status, dan perilaku yang dilakukan pada *state*, label aktivitas mengidentifikasi keadaan di mana perilaku yang ditentukan oleh ekspresi aktivitas akan dipanggil. Ekspresi perilaku dapat menggunakan atribut dan ujung asosiasi yang berada dalam cakupan entitas pemilik. *Internal transitions compartment* memiliki fungsi sebagai daftar transisi internal, dimana setiap item memiliki bentuk sebagai pemicu dan setiap nama acara dapat muncul lebih dari satu kali.



Gambar 2.15
Simple State

(Sumber : <https://www.uml-diagrams.org/state-machine-diagrams.html>)

2. *Composite State*

Composite state merupakan *state* yang memiliki *substates*. *Substates* bisa beruntutan (*sequential*) atau bersamaan (*concurrent*). Seperti *simple state*, *composite state* memiliki *compartments* seperti *name compartment*, *internal activities compartment*, *internal transitions compartment*. Perbedaannya *composite state* memiliki *decomposition compartment*, *decomposition compartment* memiliki fungsi sebagai penunjuk struktur komposisi sebuah *state*.

3. *Submachine State*

Submachine state merupakan state yang berfungsi menentukan penyisipan dari *submachine state*. *State* diagram yang memiliki *submachine state* disebut dengan *containing state machine*. *Submachine state* secara semantik setara atau sama dengan *composite state* yang memiliki *compartments* seperti *name compartment*, *internal activities compartment*, *internal transitions compartment*, dan *decomposition compartment*.

BAB III

ANALISIS DAN PERANCANGAN

3.1 Gambaran Umum

Alat yang dirancang ini merupakan alat yang digunakan untuk menyalakan dan mematikan lampu dengan tepukan tangan tanpa perlu menekan saklar. Alat ini bertujuan untuk memudahkan manusia dalam menyalakan dan mematikan tanpa harus menekan saklar lampu yang pada umumnya terletak pada dinding ruangan kamar.

Perancangan alat menyalakan dan mematikan lampu ini menggunakan sensor suara FC-04 sebagai alat untuk mendeteksi suara tepukan tangan, mikrokontroler Atmega328P sebagai pusat kendali, untuk mengolah data masukan berupa data analog yang kemudian diubah menjadi data digital. Kemudian sensor cahaya LDR berguna untuk mengetahui apakah lampu dalam keadaan menyala atau mati dan adapun relay sebagai alat yang memiliki prinsip kerja seperti saklar untuk membuka dan menutup arus listrik yang akan diteruskan kepada lampu.

Cara kerja alat ini adalah sensor suara FC-04 menangkap suatu bunyi atau suara berupa tepukan tangan. Setelah itu sensor suara FC-04 mengirimkan data dalam bentuk analog kepada arduino. Data analog akan dikonversi menjadi sinyal digital oleh ADC kemudian diteruskan kepada mikrokontroler. Hasil data digital yang berbentuk biner 0 dan 1. Kemudian mikrokontroler mengelola data biner tersebut dan sensor cahaya LDR akan mendeteksi cahaya pada lampu yang menentukan apakah lampu menyala atau mati.

1. Jika kondisi lampu mati dan sensor suara FC-04 tidak mendeteksi suara tepukan tangan maka mikrokontroler tidak memberi intruksi kepada relay dan lampu tetap mati.
2. Jika kondisi lampu mati dan sensor suara FC-04 mendeteksi satu kali tepukan tangan maka mikrokontroler tidak memberi intruksi kepada relay dan lampu tetap mati.
3. Jika kondisi lampu mati dan sensor suara FC-04 mendeteksi dua kali tepukan tangan maka mikrokontroler memberi intruksi kepada relay agar arus listrik dapat mengalir dan lampu menyala, setelah itu sensor suara FC-04 diberi jeda (*delay*) selama 5 detik agar tidak

mendeteksi suara. Jeda waktu atau *delay* ini berfungsi agar tepukan yang lebih dari dua kali tidak terdeteksi langsung, agar mikrokontroler tidak memberi intruksi kepada relay yang menyebabkan lampu dapat menyala dan mati dalam waktu cepat.

4. Jika kondisi lampu menyala dan sensor suara FC-04 tidak mendeteksi suara tepukan tangan maka mikrokontroler tidak memberi intruksi kepada relay dan lampu tetap menyala.
5. Jika kondisi lampu menyala dan sensor suara FC-04 mendeteksi satu kali tepukan tangan maka mikrokontroler memberi intruksi kepada agar arus listrik tidak dapat mengalir dan lampu mati, setelah itu sensor suara FC-04 diberi jeda (*delay*) selama 5 detik agar tidak mendeteksi suara. Jeda waktu atau *delay* ini berfungsi agar tepukan yang lebih dari satu kali tidak terdeteksi langsung, agar mikrokontroler tidak memberi intruksi kepada relay yang menyebabkan lampu dapat menyala dan mati dalam waktu cepat.

Kemudian menggunakan aplikasi *sound meter* pada *smartphone* untuk mendeteksi tingkat kebisingan pada lingkungan sekitar alat kendali lampu.

3.2 Spesifikasi Kebutuhan

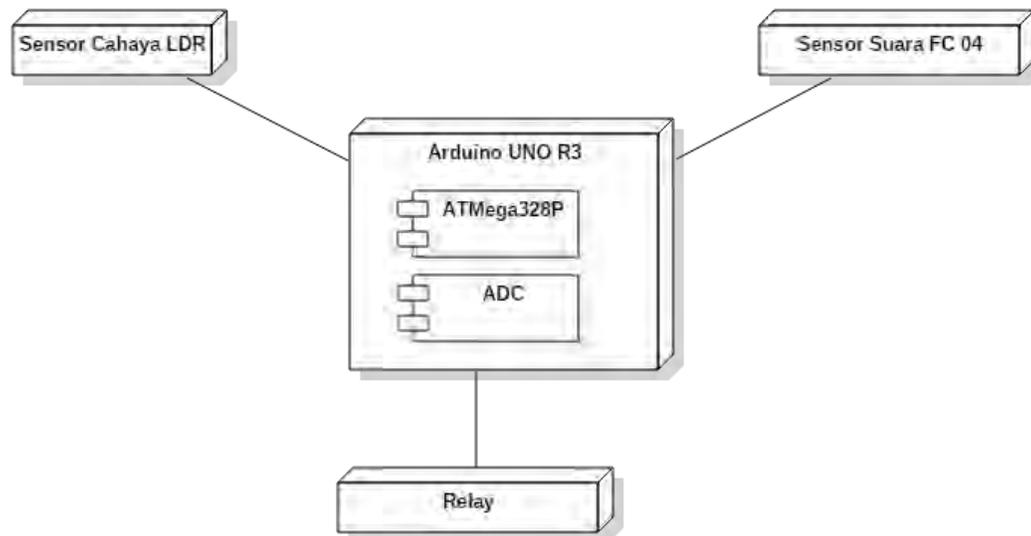
3.2.1 Kebutuhan Perangkat Keras

1. *Board* Arduino UNO : sebagai mikrokontroler.
2. Sensor suara FC-04 : sebagai alat pendeteksi suara tepukan tangan.
3. *Relay* : sebagai alat untuk memutuskan dan menghubungkan arus.
4. Sensor cahaya LDR : sebagai alat pendeteksi cahaya pada lampu.

3.2.2 Kebutuhan Perangkat Lunak

1. *Arduino IDE* : perangkat lunak untuk membuat program yang akan diunggah ke papan Arduino UNO.
2. *Windows 10* : sebagai sistem operasi dari laptop.
3. *Sound meter* : sebagai perangkat lunak untuk mengukur tingkat kebisingan.

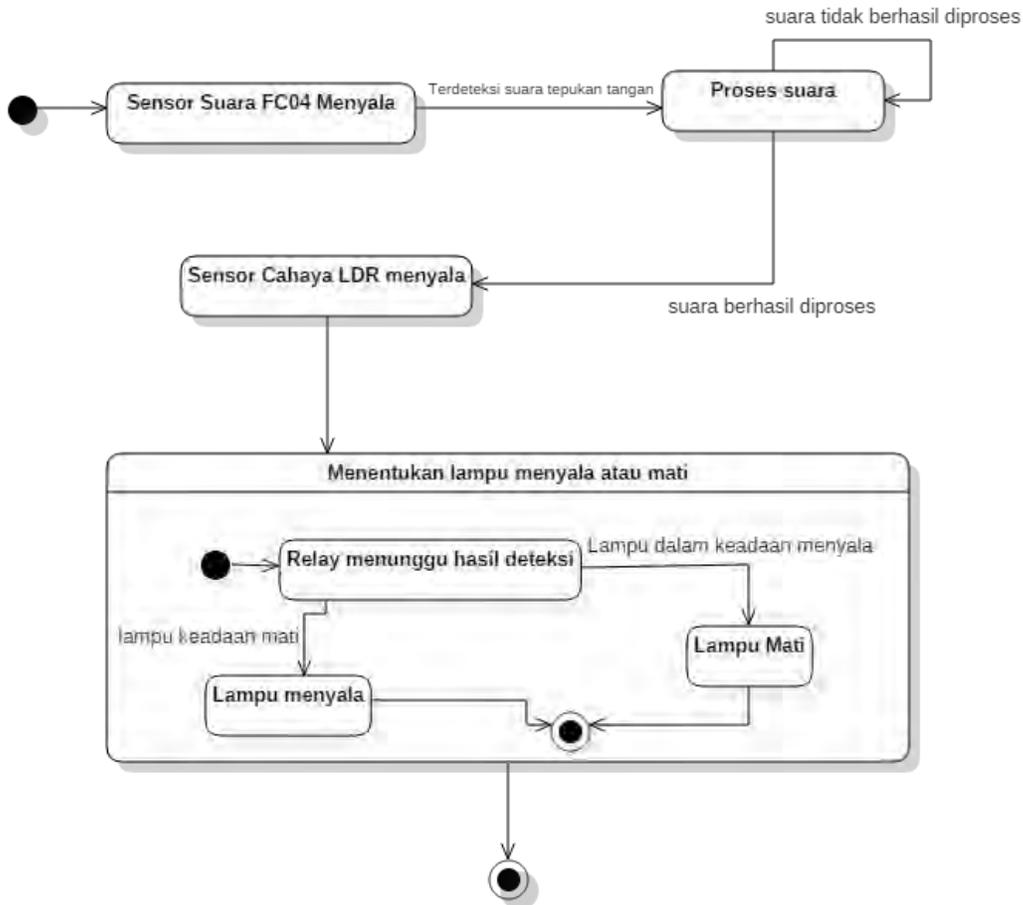
3.3 Deployment Diagram



Gambar 3.1
Deployment Diagram Alat Kendali Lampu

Deployment diagram alat kendali lampu pada gambar 3.1 terdiri dari *node* sensor cahaya LDR dan *node* sensor suara FC-04 yang terhubung dengan *node* arduino R3 yang memiliki 2 *component* di dalamnya yaitu ATMega328P dan *analog to digital converter* (ADC). Kemudian *node* arduino uno R3 terhubung dengan *node* relay

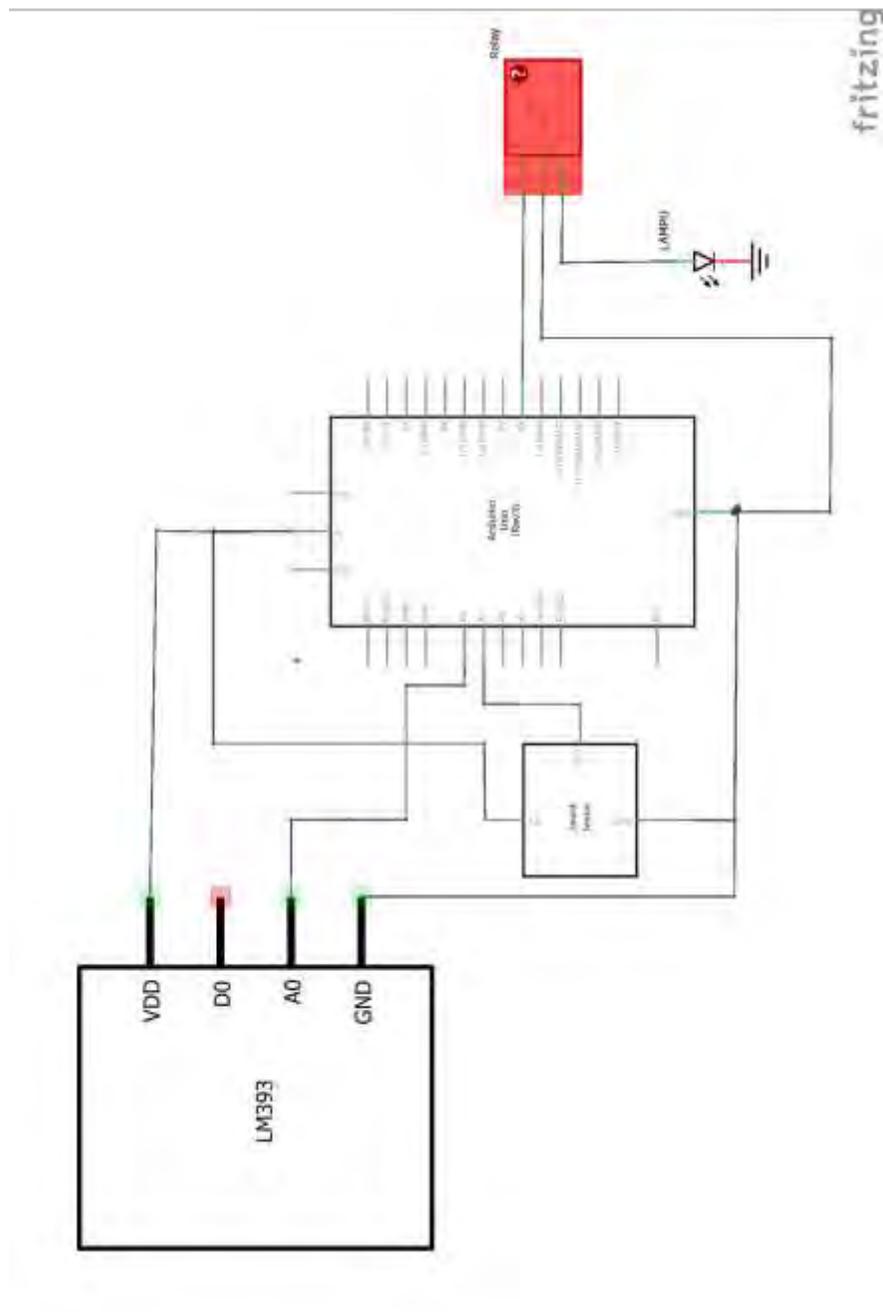
3.4 Statechart Diagram



Gambar 3.2
Statechart Diagram Alat Kendali Lampu

Sistem melakukan inialisasi, kemudian sensor suara FC-04 menyala dan menunggu tepukan tangan. Setelah itu terdeteksi suara tepukan tangan yang kemudian di proses suara tersebut. Jika suara tepukan tangan tidak berhasil diproses maka melakukan proses ulang dan ketika suara tepukan tangan berhasil diproses maka sensor cahaya LDR menyala dan menunggu hasil deteksi lampu. Jika hasil deteksi lampu dalam keadaan mati maka relay memberi intruksi untuk menyalakan lampu dan jika hasil deteksi lampu dalam keadaan menyala maka relay memberi intruksi untuk mematikan lampu. Sistem berhenti.

3.5 Skematik Perancangan Alat Kendali Lampu



Gambar 3.3
Skematik Diagram Alat Kendali Lampu

BAB IV

IMPLEMENTASI DAN PENGUJIAN

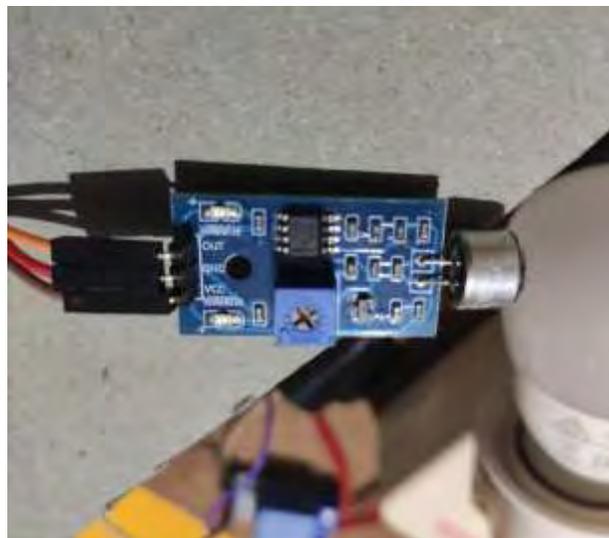
4.1 Implementasi Perangkat Keras

Pada gambar 4.1 merupakan hasil foto papan Arduino UNO yang sudah ditempatkan di dalam kotak.



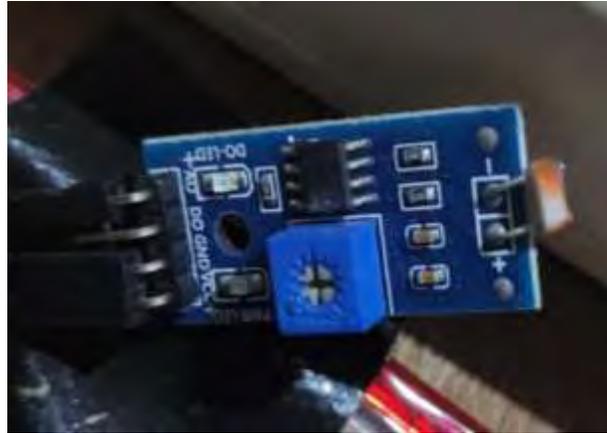
Gambar 4.1
Papan Arduino UNO

Pada gambar 4.2 merupakan hasil foto Sensor Suara FC-04 yang ditempatkan diatas kotak dan telah dihubungkan pada papan Arduino UNO.



Gambar 4.2
Sensor Suara FC-04

Pada gambar 4.3 merupakan hasil foto Sensor Cahaya LDR yang ditempatkan disamping *stop* kontak, dibawah lampu, dan telah dihubungkan pada papan Arduino UNO.



Gambar 4.3
Sensor Cahaya LDR

Pada gambar 4.4 merupakan hasil foto Relay yang ditempatkan disamping *stop* kontak, yang telah dihubungkan pada papan Arduino UNO dan *stop* kontak.



Gambar 4.4
Relay

Pada gambar 4.5 merupakan hasil akhir dari alat kendali lampu berbasis Arduino. Perancangan ini menggunakan papan Arduino UNO, sebuah sensor suara FC-04, sebuah sensor cahaya LDR, relay, kabel *jumper*, *stop* kontak, *fitting* lampu, lampu dan bantuan *Breadboard* Arduino. Mikrokontroler yang digunakan pada pengujian ini adalah Mikrokontroler ATmega 328P.



Gambar 4.5
Alat Kendali Lampu Berbasis Arduino

4.2 Pengujian dan Analisa Perangkat Keras

4.2.1 Pengujian Sistem dan Analisa Sensor Suara FC-04

Tabel 4.1
Pembacaan Sensor Suara FC-04

Deteksi Suara Tepukan Tangan	Respon Sistem	Keterangan	Hasil Pengujian
Terdeteksi suara tepukan tangan	Memberi intruksi kepada relay	Sensor Suara FC-04 mendeteksi adanya suara tepukan tangan	Sesuai
Tidak terdeteksi suara tepukan tangan	Tidak memberi intruksi kepada relay	Sensor Suara FC-04 tidak mendeteksi adanya suara tepukan tangan	Sesuai

Berdasarkan pengujian pada tabel 4.1 sistem akan melakukan pembacaan sensor suara FC-04 yang terhubung dengan mikrokontroler. Ketika sensor suara FC-04 mendeteksi

adanya suara tepukan tangan maka sistem akan memberi intruksi kepada relay melalui mikrokontroler dan ketika sensor suara FC-04 tidak mendeteksi adanya suara tepukan tangan maka sistem tidak akan memberi intruksi kepada relay.

4.2.2 Pengujian Sistem dan Analisa Sensor Cahaya LDR

Tabel 4.2
Pembacaan Sensor Cahaya LDR

Deteksi Cahaya pada Lampu	Respon Sistem	Keterangan	Hasil Pengujian
Terdeteksi cahaya pada lampu	Memberi pemberitahuan kepada mikrokontroler	Sensor Cahaya LDR mendeteksi adanya cahaya pada lampu	Sesuai
Tidak terdeteksi cahaya pada lampu	Memberi pemberitahuan kepada mikrokontroler	Sensor Cahaya LDR tidak mendeteksi adanya cahaya pada lampu	Sesuai

Berdasarkan pengujian pada tabel 4.2 sistem akan melakukan pembacaan sensor cahaya LDR yang dihubungkan dengan mikrokontroler. Ketika sensor cahaya LDR mendeteksi adanya cahaya pada lampu maka sistem akan memberi pemberitahuan kepada mikrokontroler bahwa lampu dalam kondisi menyala dan Ketika sensor cahaya LDR tidak mendeteksi adanya cahaya pada lampu maka sistem akan memberi pemberitahuan kepada mikrokontroler bahwa lampu dalam kondisi mati.

4.2.3 Pengujian Sistem dan Analisa Alat Kendali Lampu

Tabel 4.3
Pengujian dan Analisa Alat Kendali Lampu

Aksi	Respon Sistem	Keterangan	Hasil Pengujian
Sistem dihubungkan dengan baterai 9 Volt	Lampu LED indikator pada sensor (FC-04,LDR) dan relay menyala. Sensor (FC-04,LDR) aktif	Inisialisasi masukan	Sesuai

Aksi	Respon Sistem	Keterangan	Hasil Pengujian
	Memberi pemberitahuan kepada mikrokontroler bahwa lampu menyala	Sensor Cahaya LDR mendeteksi adanya cahaya pada lampu	Sesuai
Memberikan satu kali tepukan tangan dengan jarak 3 meter	Sistem memberikan intruksi kepada relay, memberikan jeda waktu selama 5 detik pada sensor suara FC-04 dan lampu mati	Sensor Suara FC-04 mendeteksi adanya tepukan tangan	Sesuai
	Sistem memberi pemberitahuan kepada mikrokontroler bahwa lampu mati	Sensor cahaya LDR tidak mendeteksi adanya cahaya pada lampu	Sesuai
Memberikan satu kali atau lebih tepukan tangan dengan jarak 3 meter (Saat jeda waktu 5 detik)	Sistem tidak memberikan perintah maupun intruksi	Sensor Suara FC-04 tidak mendeteksi adanya tepukan tangan	Sesuai
Memberikan satu kali tepukan tangan dengan jarak 3 meter (setelah jeda waktu 5 detik)	Sistem tidak memberikan intruksi kepada relay dan lampu tetap dalam kondisi mati	Sensor Suara FC-04 mendeteksi adanya tepukan tangan	Sesuai
Memberikan dua kali tepukan tangan dengan jarak 3 meter (setelah jeda waktu 5 detik)	Sistem memberikan intruksi kepada relay, memberikan jeda waktu selama 5 detik pada sensor suara FC-04 dan lampu menyala	Sensor Suara FC-04 mendeteksi adanya tepukan tangan	Sesuai
	Sistem memberi pemberitahuan kepada mikrokontroler bahwa lampu menyala	Sensor cahaya LDR mendeteksi adanya cahaya pada lampu	Sesuai

Aksi	Respon Sistem	Keterangan	Hasil Pengujian
Memberikan satu kali atau lebih tepukan tangan dengan jarak 3 meter (Saat jeda waktu 5 detik)	Sistem tidak memberikan perintah maupun intruksi	Sensor Suara FC-04 tidak mendeteksi adanya tepukan tangan	Sesuai
Memberikan satu kali tepukan tangan dengan jarak 3 meter dan memberikan <i>noise</i> dibawah 70 dB	Sistem memberikan intruksi kepada relay, memberikan jeda waktu selama 5 detik pada sensor suara FC-04 dan lampu mati	Sensor Suara FC-04 mendeteksi adanya tepukan tangan	Sesuai
	Sistem memberi pemberitahuan kepada mikrokontroler bahwa lampu mati	Sensor cahaya LDR tidak mendeteksi adanya cahaya pada lampu	Sesuai
Memberikan satu kali atau lebih tepukan tangan dengan jarak 3 meter dan memberikan <i>noise</i> dibawah 70 dB (Saat jeda waktu 5 detik)	Sistem tidak memberikan perintah maupun intruksi	Sensor Suara FC-04 tidak mendeteksi adanya tepukan tangan	Sesuai
Memberikan satu kali tepukan tangan dengan jarak 3 meter dan memberikan <i>noise</i> dibawah 70 dB (setelah jeda waktu 5 detik)	Sistem tidak memberikan intruksi kepada relay dan lampu tetap dalam kondisi mati	Sensor Suara FC-04 mendeteksi adanya tepukan tangan	Sesuai
Memberikan dua kali tepukan tangan dengan jarak 3 meter dan memberikan <i>noise</i> dibawah 70 dB (setelah jeda waktu 5 detik)	Sistem memberikan intruksi kepada relay, memberikan jeda waktu selama 5 detik pada sensor suara FC-04 dan lampu menyala	Sensor Suara FC-04 mendeteksi adanya tepukan tangan	Sesuai
	Sistem memberi pemberitahuan kepada mikrokontroler bahwa lampu menyala	Sensor cahaya LDR mendeteksi adanya cahaya pada lampu	Sesuai

Aksi	Respon Sistem	Keterangan	Hasil Pengujian
Memberikan satu kali atau lebih tepukan tangan dengan jarak 3 meter dan memberikan <i>noise</i> dibawah 70 dB (Saat jeda waktu 5 detik)	Sistem tidak memberikan perintah maupun intruksi	Sensor Suara FC-04 tidak mendeteksi adanya tepukan tangan	Sesuai

Berdasarkan pengujian tabel 4.3, ketika sistem dihubungkan dengan baterai 9V, Lampu LED indikator pada sensor suara FC-04, sensor cahaya LDR, dan relay menyala menandakan bahwa inisialisasi masukan. Ketika proses inisialisasi masukan telah selesai maka sensor suara FC-04, sensor cahaya LDR, dan relay aktif. Ketika sensor cahaya mendeteksi adanya cahaya maka sistem akan memberitahukan kepada mikrokontroler bahwa lampu menyala. Saat sensor suara FC-04 mendeteksi adanya tepukan tangan dengan jarak 3 meter yang sebanyak satu kali maka sistem memberikan intruksi kepada relay, memberikan jeda waktu selama 5 detik pada sensor suara FC-04 agar tidak menerima perintah apapun, sistem memberi pemberitahuan kepada mikrokontroler bahwa lampu mati dan lampu akan mati. Ketika adanya tepukan tangan dengan jarak 3 meter sebanyak satu kali atau lebih dibawah jeda waktu 5 detik maka sistem tidak memberikan perintah maupun intruksi dan sensor suara FC-04 tidak mendeteksi adanya tepukan tangan. Saat sensor suara FC-04 mendeteksi adanya tepukan tangan dengan jarak 3 meter sebanyak dua kali maka sistem memberikan intruksi kepada relay, memberikan jeda waktu selama 5 detik pada sensor suara FC-04 agar tidak menerima perintah apapun, sistem memberi pemberitahuan kepada mikrokontroler bahwa lampu menyala dan lampu akan menyala. Ketika adanya tepukan tangan dengan jarak 3 meter sebanyak satu kali atau lebih dibawah jeda waktu 5 detik maka sistem tidak memberikan perintah maupun intruksi dan sensor suara FC-04 tidak mendeteksi adanya tepukan tangan.

Saat sensor suara FC-04 mendeteksi adanya tepukan tangan dengan jarak 3 meter sebanyak satu kali dan memberikan *noise* dibawah 70 dB maka sistem memberikan intruksi

kepada relay, memberikan jeda waktu selama 5 detik pada sensor suara FC-04 agar tidak menerima perintah apapun, sistem memberi pemberitahuan kepada mikrokontroler bahwa lampu mati dan lampu akan mati. Ketika adanya tepukan tangan dengan jarak 3 meter sebanyak satu kali atau lebih dibawah jeda waktu 5 detik dan memberikan *noise* dibawah 70 dB maka sistem tidak memberikan perintah maupun intruksi dan sensor suara FC-04 tidak mendeteksi adanya tepukan tangan. Saat sensor suara FC-04 mendeteksi adanya tepukan tangan dengan jarak 3 meter sebanyak dua kali dan memberikan *noise* dibawah 70 dB maka sistem memberikan intruksi kepada relay, memberikan jeda waktu selama 5 detik pada sensor suara FC-04 agar tidak menerima perintah apapun, sistem memberi pemberitahuan kepada mikrokontroler bahwa lampu menyala dan lampu akan menyala. Ketika adanya tepukan tangan dengan jarak 3 meter sebanyak satu kali atau lebih dibawah jeda waktu 5 detik dan memberikan *noise* dibawah 70 dB maka sistem tidak memberikan perintah maupun intruksi dan sensor suara FC-04 tidak mendeteksi adanya tepukan tangan.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan penyusunan yang sudah disusun dan dilakukan pengujiannya, maka mendapatkan beberapa kesimpulan, sebagai berikut :

1. Alat kendali lampu dibuat dengan menggunakan sensor suara FC-04 untuk mendeteksi tepukan tangan dan sensor cahaya LDR untuk mendeteksi cahaya pada lampu.
2. Jarak tepukan tangan sangat mempengaruhi hasil dalam menyalakan dan mematikan lampu, semakin jauh tepukan tangan maka semakin lampu tidak merespon perintah untuk menyalakan dan mematikan lampu. Dengan jarak dibawah 3 meter maka hasil yang didapatkan semakin akurat untuk menyalakan dan mematikan lampu.
3. Suara yang tidak diinginkan atau *noise* sangat berpengaruh terhadap hasil. Ketika melakukan pengujian dengan memberikan *noise* diatas 70dB pada alat kendali lampu, lampu tidak akan merespon tepukan tangan.

5.2 Saran

Pada perancangan alat kendali lampu ini masih memiliki banyak kekurangan dan masih jauh dari kesempurnaan. Alat kendali lampu yang dirancang masih membutuhkan penyempurnaan dan dikarenakan beberapa hal penulis memiliki keterbatasan untuk penambahannya, sehingga beberapa saran yang mungkin akan lebih membantu untuk penelitian selanjutnya, sebagai berikut :

1. Pada perancangannya menambahkan filter suara agar suara *noise* tidak diterima oleh alat kendali lampu dan dapat membedakan suara tepukan tangan dengan suara lainnya.
2. Menambahkan suara lain dalam menyalakan dan mematikan lampu. Bukan hanya tepukan tangan.

DAFTAR PUSTAKA

- A.S., Rosa dan M, Shalahuddin, 2014, "*Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*", Bandung: Informatika Bandung.
- Ahmed, Syed Naeem, 2007, "*Physics and Engineering of Radiation Detection*", London: Academic Press Inc.
- Ahyadi, Zaiyan. 2018, "*Belajar Antarmuka Arduino Secara Cepat Dari Contoh*", Yogyakarta: Poliban Press.
- Aryanti dan Ikhtison Mekongga, Hari Ramadhan, 2016, "*Implementasi Sensor Suara Sebagai Pengendali Gerakan Robot Penari Humanoid dengan ATMEGA 8535*", *Jurnal JUPITER*, Vol. 8, No. 1
- Ariyanti, Slamet , dkk, 2018, "*Sistem Buka Tutup Pintu Otomatis Berbasis Suara Manusia*", ELINVOS.
- Asriwati, 2017, "*Fisika Kesehatan dalam Keperawatan*", Yogyakarta: DeepPublish.
- Banzi, Massimo, 2011, "*Getting Started with Arduino Second Edition*", Amerika Serikat: O'Reilly Media.
- Dalis, Sopiyan, 2017, "*Rancang Bangun Sistem Informasi Lembaga Penelitian Dan Pengabdian Masyarakat Berbasis Web*", *Paradigma*, Vol. XX, No. XX.
- Dharmawan, Hari Arief, 2017, "*Mikrokontroler Konsep Dasar dan Praktis*", Malang: UBMedia.
- Ekojono dan Andriani Parastiwi, Cahya Rahmad, Anugran Nur Rahmanto, 2018, "*Pemrograman Spreadsheet Untuk Pemodelan Kontrol Rangkaian Elektronika*", Malang: Polinema Press.
- Fauzan, Lalita, Lalita Chandiany Adiputri, 2019, "*Tutorial Membuat Prototipe Prediksi Ketinggian Air(PKA) Untuk Pendeteksi Banjir Peringatan Dini Berbasis IOT*", Bandung: Kreatif Industri Nusantara.
- Hardana, 2018, "*Dasar-Dasar Mikrokontroler Arsitektur ARM ST Microelectronics*", Jakarta: PT. Mitra Sinergi Optima.
- Haqi, Heri, Heri Satria Setiawan, 2019, "*Aplikasi Absensi Dosen dengan Java dan Smartphone sebagai Barcode Reader*", Jakarta: PT Elex Media Komputindo Kelompok Gramedia.
- Isa, Indra Griha Tofik dan George Pri Hartawan, 2017, "*Perancangan Aplikasi Koperasi Simpan Pinjam Berbasis Web (Studi Kasus Koperasi Mitra Setia)*", *Jurnal Ilmiah Ekonomi*, Vol. 5 Edisi 10.
- Kadir, Abdul, 2016, "*Simulasi Arduino*", Jakarta: PT Elex Media Komputindo Kelompok Gramedia.
- Kharisma, Oktaf Brilian dan Hanif Burhanuddin Putra Utama, 2018, "*Pengembangan Sistem Pengaman Pintu Laboratorium Robotika Uin Sultan Syarif Kasim Berdasarkan Siulan Berbasis Sensor Fc-04 Dan Mikrokontroler Atmega 328*", *Jurnal Sains dan Teknologi*, Vol. 7 No. 1,
- Kinasih, Syarli, dkk, 2018, "*Pengontrolan Ayunan Bayi Otomatis Dengan Mendeteksi Sensor Suara Menggunakan Mikrokontroler Arduino*", *Jurnal Ilmiah Ilmu Komputer*, Vol. 4, No. 2.

- Kroenke, David M, 2004, "*Database Processing Dasar-dasar, Desain & Implementasi*", Jakarta: Erlangga.
- Malik, Moh. Ibnu, et.al, 2009, "*Aneka Proyek Mikrokontroler PIC16F84/A*", Jakarta: PT Elex Media Komputindo Kelompok Gramedia.
- Novianti Keyza, Chairisni Lubis, dan Tony, 2012, "*Perancangan Prototipe Sistem Penerangan Otomatis Ruang Berjendela Berdasarkan Intensitas Cahaya*", Seminar Nasional Teknologi Informasi 2012.
- Nugroho, Adi, 2010, "*Rekayasa Perangkat Lunak Berbasis Objek dengan Metode USDP*". Yogyakarta: Andi
- Parinduri, Ikhsan dan Helmi Fauzi Siregar, Iskandar, 2017, "*Pengontrolan Suhu Kelembaban Kumbung Jamur Tiram Putih*", Green Press.
- Pressman, Roger, 2010, "*Software Engineering : a practitioner's approach*", New York: McGraw-Hill Education.
- Rao, Josyula R. dan Berk Sunar, 2005, "*Cryptographic Hardware and Embedded Systems - CHES 2005*", Edinburgh: Springer Science & Business Media.
- Rinas, I Wayan, 2013, "*Simulasi Penggunaan Filter Pasif, Filter Aktif dan Filter Hybrid Shunt untuk Meredam Meningkatnya Distorsi Harmonisa yang Disebabkan Oleh Munculnya Gangguan Resonansi*", *Jurnal Elektro*. Vol. 12, No. 2.
- Rusdi, Muhammad dan Achmad Yani, 2018. "*Sistem Kendali Peralatan Elektronik Melalui Media Bluetooth Menggunakan Voice Recognition*", *Journal of Electrical Technology*, Vol. 3, No. 1.
- Saleh, Muhamad dan Munnik Haryanti, 2017, "*Rancang Bangun Sistem Keamanan Rumah Menggunakan Relay*", *Jurnal Teknologi Elektro, Universitas Mercu Buana* Vol. 8 No. 3.
- Smith, Steven W, 1999, "*The Scientist and Engineer's Guide to Digital Signal Processing*". San Diego: California Technical Publishing
- Sujarwata, 2012, "*Belajar Mikrokontroler BS2SX Teori, Penerapan dan Contoh pemrograman PBasic*". Yogyakarta: Deepublish.
- Syam, Rafiuddin, 2013, "*Seri Buku Ajar Dasar Dasar Teknik Sensor*", Makassar: Fakultas Teknik Universitas Hasanuddin.
- Whitten, Jeffrey L. dan Lonnie D. Bentley, 2007, "*System Analysis & Design Methods Seventh Edition*", New York: McGraw-Hill.
- Wicaksono, Soetam Rizky, 2015, "*Rekayasa Perangkat Lunak. Seribu Bintang*".
- Yurindra, 2017, "*Software Engineering*", Yogyakarta : Deepublish Publisher.
- <http://electriciantraining.tpub.com/14184/css/Figure-1-36-Magnetic-Microphone-Action-53.htm>. Diakses tanggal 10 Oktober 2019. Jam 20.45 WIB.
- <http://hyperphysics.phy-astr.gsu.edu/hbase/Audio/mic3.html>. Diakses tanggal 9 Oktober 2019. Jam 13.45 WIB.
- <http://www.jogjarobotika.com/photodiode-phototransistor-ldr-ir-led/700-sensor-cahaya-ldr-light-dependent-resistor-5mm.html>. Diakses tanggal 7 Maret 2020. Jam 15.00 WIB.

<https://store.Arduino.cc/usa/Arduino-uno-rev3>. Diakses tanggal 7 Oktober 2019. Jam 17.30 WIB.

<https://www.alldatasheet.com>. Diakses tanggal 10 Oktober 2019. Jam 02.40 WIB.

<https://www.elechouse.com/elechouse/images/product/simpleVR/simpleVR-4.jpg>. Diakses tanggal 14 Februari 2020. Jam 12.24 WIB.

<https://www.labelektronika.com/2017/02/arduino-uno-mikrokontroler-atmega-328.html>. Diakses tanggal 5 Oktober 2019. Jam 11.42 WIB.

<https://www.mediacollege.com/audio/microphones/condenser.html>. Diakses tanggal 9 Oktober 2019. Jam 11.11 WIB.

<https://www.norwegiancreations.com/2016/03/arduino-tutorial-simple-high-pass-band-pass-and-band-stop-filtering/>. Diakses tanggal 12 Februari 2020. Jam 07.00 WIB.

<https://www.theengineeringprojects.com/2017/08/introduction-to-atmega328.html>. Diakses tanggal 5 Oktober 2019. Jam 21.10 WIB.

<https://www.uml-diagrams.org/class-diagrams-overview.html>. Diakses tanggal 19 Oktober 2020. Jam 20.20 WIB.

<https://www.uml-diagrams.org/deployment-diagrams.html>. Diakses tanggal 19 Oktober 2020. Jam 21.30 WIB.

<https://www.uml-diagrams.org/state-machine-diagrams.html>. Diakses tanggal 19 Oktober 2020. Jam 18.00 WIB.

LAMPIRAN

Listing Program Alat Kendali Lampu

```
#define on 1
#define off 0

const int suara = A1;
const int relay = 8;
const int ldr = A0;

int bacaSuara;
int batasSuara = 50;

int bacaLdr;
int batasLDR = 150;

int i = 0;
int x = 0;
int tepukan = 0;
long batasWaktu = 0;

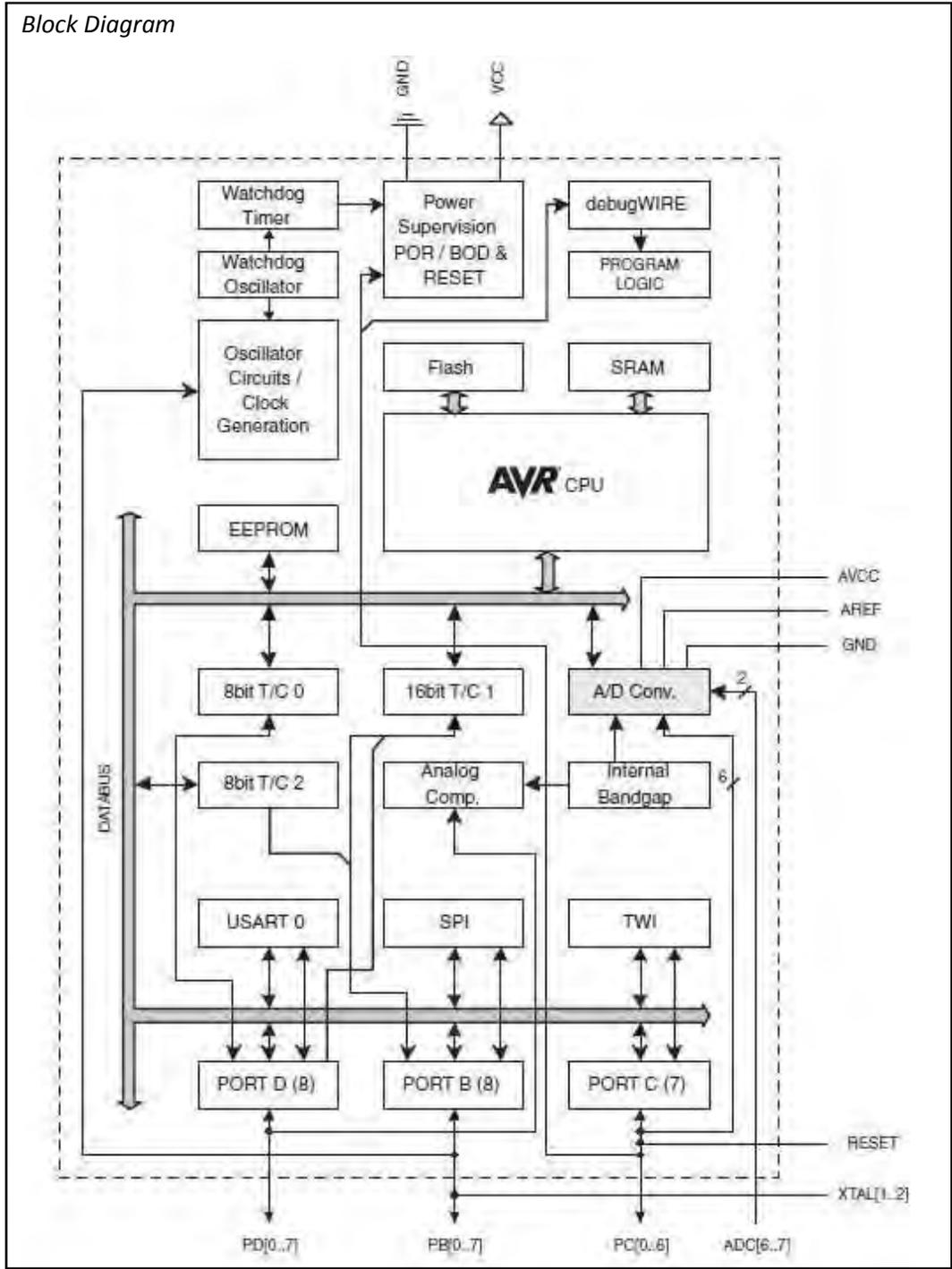
void setup() {
  Serial.begin(9600);

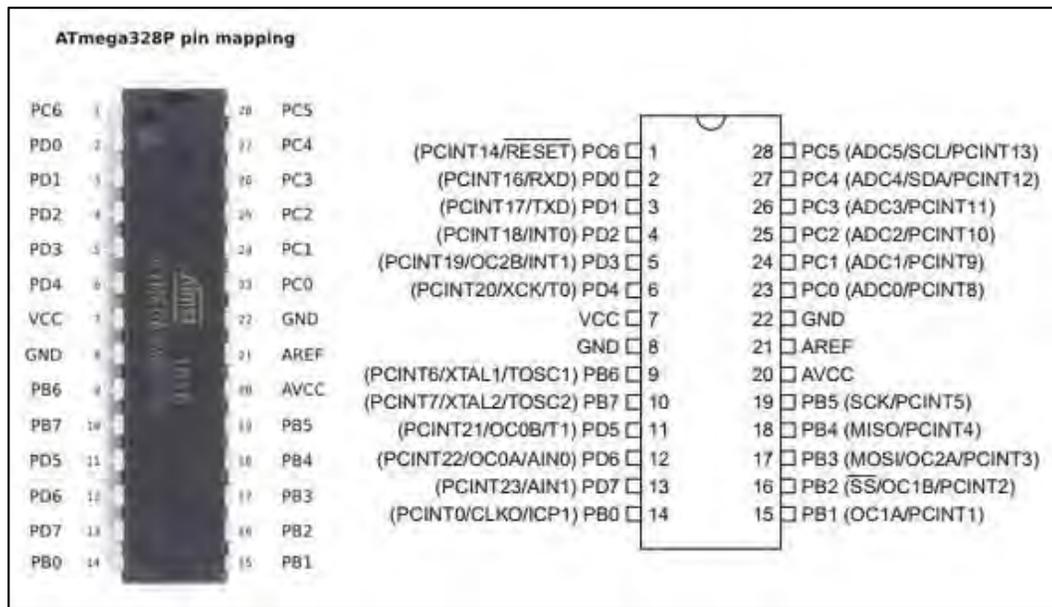
  pinMode(suara, INPUT);
  pinMode(ldr, INPUT);
  pinMode(relay, OUTPUT);
```

```
}  
  
void delaySensorSuara(int d) {  
    bacaSuara = analogRead(suara);  
    delay(d * 1000);  
}  
  
void loop() {  
    bacaSuara = analogRead(suara);  
    bacaLdr = analogRead(ldr);  
    unsigned long t = millis();  
  
    // LAMPU KEADAAN MATI  
    if (bacaSuara <= batasSuara && bacaLdr >= 200) {  
        batasWaktu = t+2000;  
        tepukan++;  
        delay(200);  
    }  
    if(x==1 && t>=batasWaktu){  
        tepukan=0;  
    }  
    else if(tepukan==2){  
        i++;  
        tepukan=0;  
    }  
  
    // LAMPU KONDISI NYALA
```

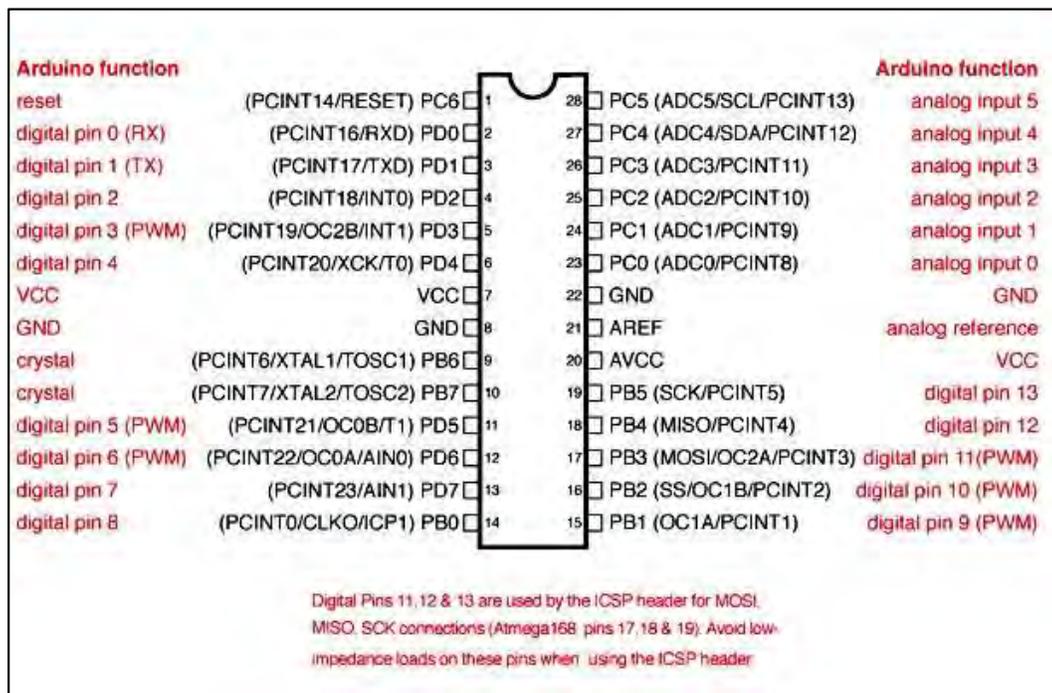
```
if (bacaSuara <= batasSuara && bacaLdr < 200){  
    i++;  
}  
  
Serial.println(bacaLdr);  
bacaSuara = i % 2;  
if (bacaLdr >= 200 && bacaSuara == 0) {  
    digitalWrite(relay, off);  
    delaySensorSuara(5);  
}  
else if (bacaLdr < 200 && bacaSuara == 1) {  
    digitalWrite(relay, on);  
    delaySensorSuara(5);  
}  
}
```

Datasheet ATmega 328p

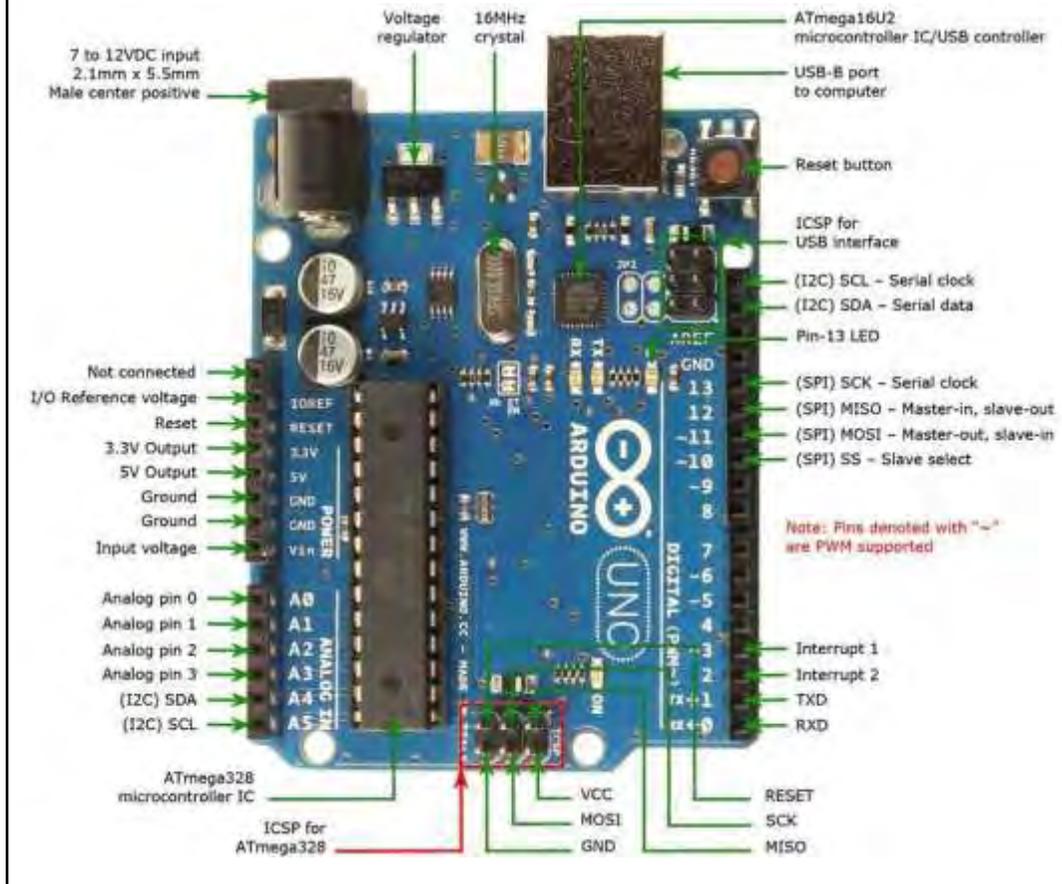




Datasheet Arduino UNO



Konfigurasi pin





Data Sheet

Light dependent resistors

NORP12 RS stock number 651-507
NSL19-M51 RS stock number 596-141

Two cadmium sulphide (cdS) photoconductive cells with spectral responses similar to that of the human eye. The cell resistance falls with increasing light intensity. Applications include smoke detection, automatic lighting control, batch counting and burglar alarm systems.

Guide to source illuminations

Light source	Illumination (Lux)
Moonlight	0.1
60W bulb at 1m	50
1W MES bulb at 0.1m	100
Fluorescent lighting	500
Bright sunlight	30,000

Circuit symbol



Light memory characteristics

Light dependent resistors have a particular property in that they remember the lighting conditions in which they have been stored. This memory effect can be minimised by storing the LDRs in light prior to use. Light storage reduces equilibrium time to reach steady resistance values.

NORP12 (RS stock no. 651-507)

Absolute maximum ratings

Voltage, ac or dc peak	320V
Current	75mA
Power dissipation at 30°C	250mW
Operating temperature range	-60°C to +75°C

Electrical characteristics

$T_A = 25^\circ\text{C}$, 2854°K tungsten light source

Parameter	Conditions	Min.	Typ.	Max.	Units
Cell resistance	1000 lux	-	400	-	Ω
	10 lux	-	9	-	k Ω
Dark resistance	-	1.0	-	-	M Ω
Dark capacitance	-	-	3.5	-	pF
Rise time 1	1000 lux	-	2.8	-	ms
	10 lux	-	1.8	-	ms
Fall time 2	1000 lux	-	48	-	ms
	10 lux	-	120	-	ms

1. Dark to 110% R_L

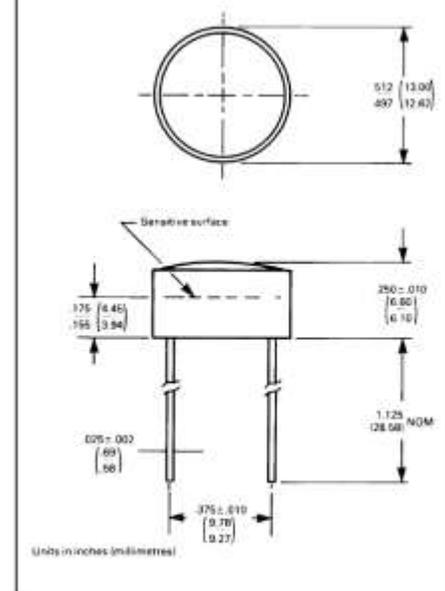
2. To $10 \times R_L$

R_L = photocell resistance under given illumination.

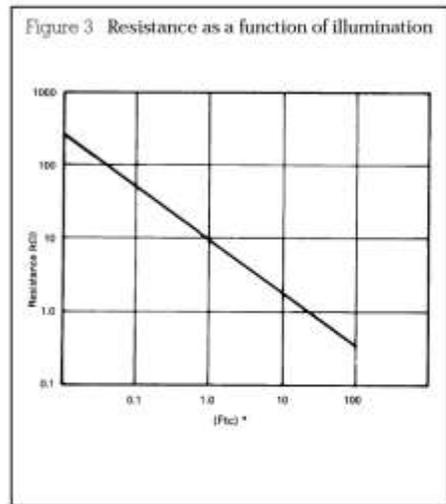
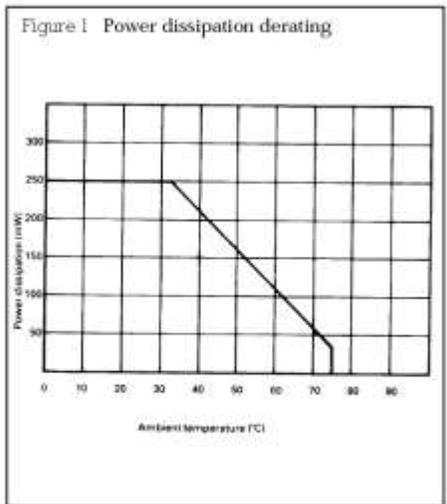
Features

- Wide spectral response
- Low cost
- Wide ambient temperature range.

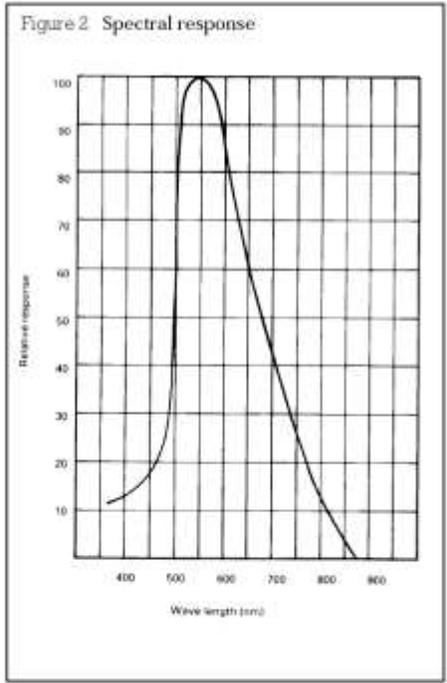
Dimensions



232-3816



*1f/c=10.764 lumens



Absolute maximum ratings

Voltage, ac or dc peak _____ 100V
 Current _____ 5mA
 Power dissipation at 25°C _____ 50mW*
 Operating temperature range _____ -25°C +75°C

*Derate linearly from 50mW at 25°C to 0W at 75°C.

Electrical characteristics

Parameter	Conditions	Min.	Typ.	Max.	Units
Cell resistance	10 lux	20	-	100	Ω
	100 lux	-	5	-	Ω
Dark resistance	10 lux after 10 sec	20	-	-	M Ω
Spectral response	-	-	550	-	nm
Rise time	10hr	-	45	-	ms
Fall time	10hr	-	55	-	ms

Figure 4 Resistance as a function illumination

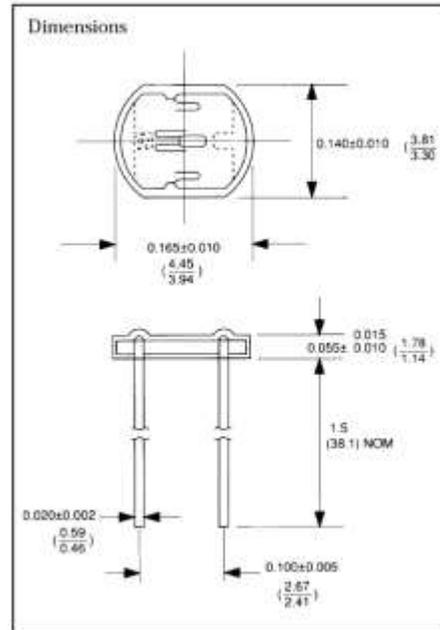
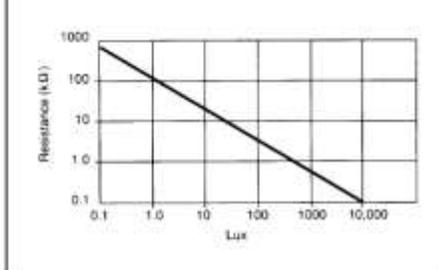
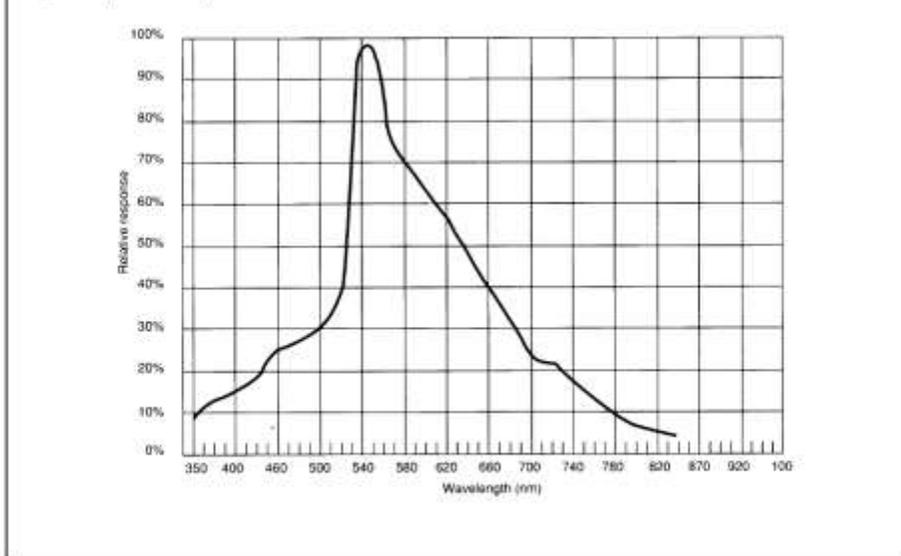


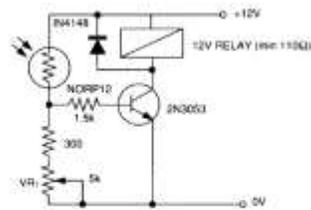
Figure 5 Spectral response



232-3816

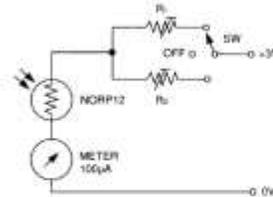
Typical application circuits

Figure 6 Sensitive light operated relay



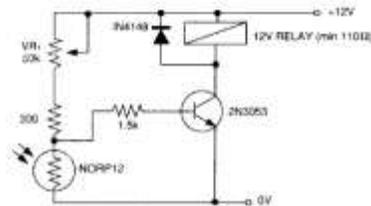
Relay energised when light level increases above the level set by VR₁.

Figure 9 Logarithmic law photographic light meter



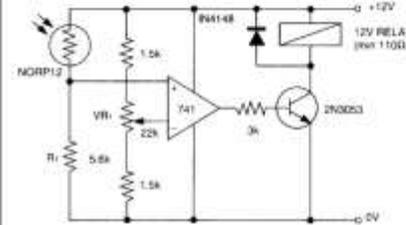
Typical value $R_1 = 100k\Omega$
 $R_1 = 200k\Omega$ preset to give two overlapping ranges.
 (Calibration should be made against an accurate meter.)

Figure 7 Light interruption detector



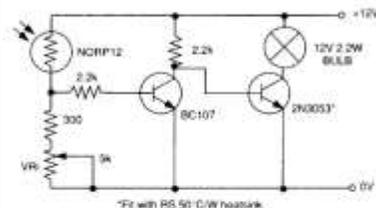
As Figure 6 relay energised when light level drops below the level set by VR₁.

Figure 10 Extremely sensitive light operated relay



(Relay energised when light exceeds preset level.)
 Incorporates a balancing bridge and op-amp. R₁ and NORP12 may be interchanged for the reverse function.

Figure 8 Automatic light circuit



Adjust turn-on point with VR₁.

The information provided in RS technical literature is believed to be accurate and reliable; however, RS Components assumes no responsibility for inaccuracies or omissions, or for the use of this information, and all use of such information shall be entirely at the user's own risk. No responsibility is assumed by RS Components for any infringements of patents or other rights of third parties which may result from its use. Specifications shown in RS Components technical literature are subject to change without notice.

RS Components, PO Box 99, Corby, Northants, NN17 9RS

 An Electrocomponents Company

Telephone: 01536 201234

© RS Components 1997

SONGLE RELAY

	RELAY ISO9002	SRD
---	---------------	-----



1. MAIN FEATURES

- Switching capacity available by 10A in spite of small size design for high density P.C. board mounting technique.
- UL, CUL, TUV recognized.
- Selection of plastic material for high temperature and better chemical solution performance.
 - Sealed types available.
- Simple relay magnetic circuit to meet low cost of mass production.

2. APPLICATIONS

- Domestic appliance, office machine, audio, equipment, automobile, etc.
(Remote control TV receiver, monitor display, audio equipment high rushing current use application.)

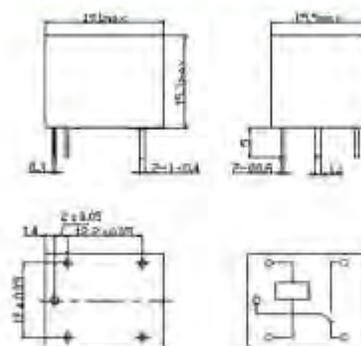
3. ORDERING INFORMATION

SRD	XXVDC	S	L	C
Model of relay	Nominal coil voltage	Structure	Coil	Contact form
SRD	03 05 06 09 12 24 48VDC	S-Sealed type	L:0.36W	A-1 form A
		F-Fix free type	F:0.45W	B-1 form B
				C-1 form C

4. RATING

CCC	FILE NUMBER: CQC03001003729	7A/240VDC
CCC	FILE NUMBER: CQC03001003731	10A/250VDC
UL/CUL	FILE NUMBER: E167996	10A/125VAC 28VDC
TUV	FILE NUMBER: R50056114	10A/250VAC 30VDC

5. DIMENSION (unit:mm) DRILLING (unit:mm) WIRING DIAGRAM



6. COIL DATA CHART (AT20 ° C)

Coil Sensitivity	Coil Voltage Code	Nominal Voltage (VDC)	Nominal Current (mA)	Coil Resistance (Ω) □	Power Consumption (W)	Pull-In Voltage (VDC)	Drop-Out Voltage (VDC)	Max. Allowable Voltage (VDC)
SRD (High Sensitivity)	03	03	120	25	abt. 0.36W	75% Max.	10% Min.	120%
	05	05	71.4	70				
	06	06	80	100				
	09	09	40	225				
	12	12	30	400				
	24	24	15	1600				
SRD (Standard)	03	03	150	20	abt. 0.45W	75% Max.	10% Min.	110%
	05	05	89.3	55				
	06	06	75	80				
	09	09	50	180				
	12	12	37.5	320				
	24	24	18.7	1280				
	48	48	10	4500	abt. 0.31W			

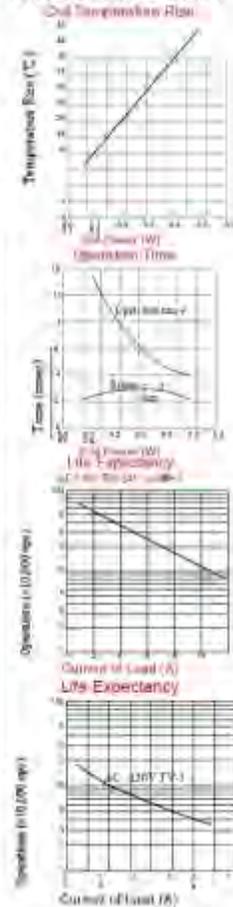
7. CONTACT RATING

Item	Type	SRD
	FORM C	FORM A
Contact Capacity	7A	10A 30VDC
Resistive Load (cosφ=1)	30VDC	16A 240VAC
Inductive Load (cosφ=0.4 L/R=7msec)	10A 125VAC	5A 120VAC
	250VAC	5A 28VDC
	3A 120VAC	
	3A 28VDC	
Max. Allowable Voltage	250VAC/110VDC	250VAC/110VDC
Max. Allowable Power	800VAC/240W	1200VA/300W
Contact Material	AgCdO	AgCdO

8. PERFORMANCE (at initial value)

Item	Type	SRD
Contact Resistance		100mΩ Max.
Operation Time		10msec Max.
Release Time		5msec Max.
Dielectric Strength	Between coil & contact	1500VAC 50/60HZ (1 minute)
	Between contacts	1000VAC 50/60HZ (1 minute)
	Insulation Resistance	100 MΩ Min. (500VDC)
Max. ON/OFF Switching		
Mechanically		300 operation/min
Electrically		30 operation/min
Ambient Temperature		-25°C to +70°C
Operating Humidity		45 to 85% RH
Vibration		
Endurance		10 to 55Hz Double Amplitude 1.5mm
Error Operation		10 to 55Hz Double Amplitude 1.5mm
Shock		
Endurance		100G Min.
Error Operation		10G Min.
Life Expectancy		
Mechanically		10 ⁷ operations Min. (no load)
Electrically		10 ⁵ operations Min. (at rated coil voltage)
Weight		abt. 10grs.

9. REFERENCE DATA





Author: gr4y1

Hi, I'm Brad. My interests spread over a large area and I tend to get carried away when something new peaks my interest. I picked up my basic electronics knowledge in bed. Say what? I was laid up after surgery (years ago) and my wife got me a Radio Shack Electronics Learning Lab to pass the time. I still have it and I HIGHLY recommend it to anyone interested in learning about electronics. *Please let me know if I'm taking the time and there's something better out there too! :) Please do not be shy about letting me know if you spot any mistakes or omissions in any of my Instructables. Your feedback is appreciated! And here are a couple of "rules" I've learned along the way that help me with my DIY projects. #1. (And this is the number one rule!) Finish what you start #2. K.I.S.S. (Keep it Simple Stupid) #3. The enemy of good is better (think about that one)

Intro: Simple FC-04 Sound Sensor Demo

The FC-04 sound sensor module, in my opinion, is a very sensitive detection module for the price. Although this sensor does not provide the ability to identify specific sounds or the frequency of the sounds detected it does what it is supposed to -- it detects sound.

After almost a full day of playing with this sensor I did manage to come up with a simple script that breaks a couple of LEDs when sound is detected. More importantly, to me anyway, is that after a day of "playing" with this thing I learned that stupid little inanimate objects can still "get" me off.

Had I understood the true value of the on-board potentiometer (POT) my day, and this Instructable, would have been a lot shorter. The key to using this sound sensor is setting right there on the board, the POT. Everything I saw said the POT was for adjusting "sensitivity". Initially, I assumed that the POT adjusted the "sensitivity" of the microphone pickup, as in a GAIN control. But nope, just to mess with my head, that POT adjusts the "sensitivity" of the voltage trigger point (VTP). Meaning that it adjusts the level of voltage required to trigger the OUTPUT.

The key to using this sensor is that POT and finding the sweet spot for the VTP. Turning that POT a tiny, tiny, tiny bit in either direction will leave the output pin in either a constantly HIGH or LOW state. This will be covered in more detail in the Setup step of this Instructable.

And if you want to shorten your setup experience even further I would suggest you also pay close attention to any delay times in a sketch that may affect the rate at which you are sampling the output value from the sensor. You want to read that output value as many times as you can, as fast as you can. Slowing things down with delays can cause you to miss a significant amount of changes in the output value. There is a debugging section in the attached sketch which is commented out. It is commented out because it introduces a half of a second delay in the sketch so you can almost see the values as they flash by on the serial monitor. But, just that 1/2 of a second delay has a negative impact on the entire sketch. So if you do use the debugging section REMEMBER to comment those lines back out when you are done.

And with all of that out of the way let's get this thing working.



Step 1: Setup Items you will need

- one (1) Arduino board
- one (1) FC-04 Sound Sensor Module (or similar 3 pin sensor)
- two (2) LEDs
- two (2) 150 ohm resistors (color bands = Brown/Green/Brown)
- one (1) breadboard and jumper wires
- a small screwdriver (or something) to adjust the potentiometer

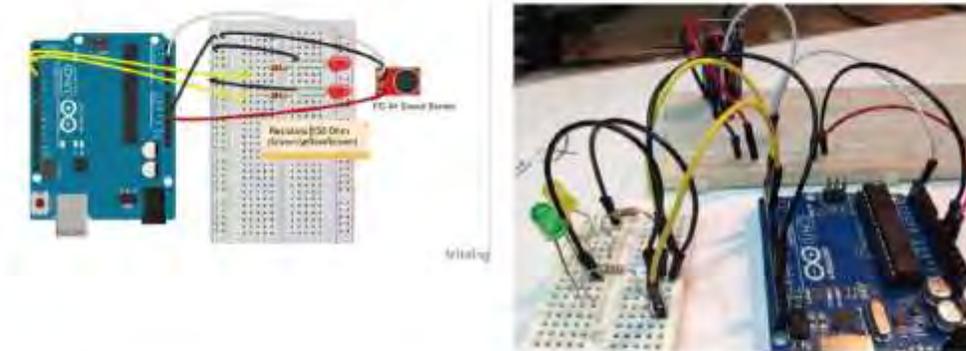
Setup Instructions

HINT: When placing the sound sensor on the breadboard do so in a manner that makes it easy for you to get to the on-board POT.

1. 5v out from Arduino to positive rail on breadboard
2. GND from Arduino to ground rail on breadboard
3. connect 5v positive to VCC on the sensor module
4. connect ground to GND on the sensor module
5. connect analog pin A4 (A4) on Arduino to OUT on the sensor module
6. place the resistors and LEDs on your breadboard

© 2014 gr4y1. All rights reserved. This work is derived from Instructables.com.

- 6a. connect the positive leg of one LED to one leg of a resistor
 - 6b. connect the positive leg of the other LED to one leg of the other resistor
 7. connect digital pin 42 to a resistor on the opposite leg from the LED
 8. connect digital pin 44 to the other resistor in the same manner
 9. connect the negative legs of both LEDs to ground
- And I believe that does it for setting this up. Shall we move on to some code for this? And guess what is coming up in the next step!



Step 2: Code

The sketch below is not commented, because the attached file is HEAVILY commented. You can automatically load the attached file to the Arduino interface simply by double clicking on it.

```

// FC-04 Sound Sensor
const int SenOut = A4;
const int LED1 = 2;
const int LED2 = 4;
// sensorValue = 0;
void setup()
{
  pinMode(LED1, OUTPUT);
  pinMode(LED2, OUTPUT);
  pinMode(SenOut, INPUT);
  Serial.begin(9600);
  void loop()
  {
    sensorValue = analogRead(SenOut);
    if (sensorValue > 850)
    {
      digitalWrite(LED1, HIGH);
      digitalWrite(LED2, LOW);
    }
    else if (sensorValue < 950)
    {
      digitalWrite(LED2, HIGH);
      digitalWrite(LED1, LOW);
    }
    delay(50);
  }
}

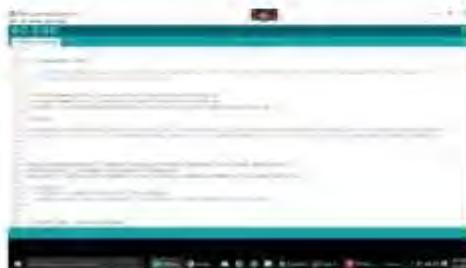
```

```

}
// Debugging below (in comment to use)
// Serial.println(sensorValue);
// Serial.println();
// delay(100);
}

```

Again, for a ton of information regarding the above sketch (and more) please refer to the attached .ino file



File Downloads



FCP4SoundSensor.ino (7 KB)

NOTE: When saving, if you get .ino as the file ext, rename it to FCP4SoundSensor.ino)

Step 3: Specifications and a Treat

SPECIFICATIONS:

Detection Ability - detects sound waves, but does not recognize particular sounds nor the frequency of a sound.

Working Voltage - 3.3 - 5 V

Output - single channel Digital - (output remains HIGH (5v) until its pulsed LOW when sound is detected)

Note: Even though the output is digital, either HIGH or LOW, this little guy responds fast enough so that you can still use different voltage levels to make things happen in your sketch. What I'm trying to say here is that when the sense picks up a sound vibration it will begin to bring it's output low. But, those sound vibrations move so fast and the guy responds so fast, that it may be in the process of pulling the output LOW when the vibration(s) stop and the sensor runs the output HIGH again. So the voltage may have only dropped from HIGH (5v) to 3.5v before it was brought back to 5v again. And we can take advantage of that in our sketches.

Adjustable Sensitivity - via onboard POT (potentiometer) - This adjusts the sensitivity OUTPUT trigger. Adjusting the POT is high and the OUTPUT will always remain HIGH, to low and OUTPUT will remain LOW.

Pins on Module - three (3), VCC / GND / OUT

On-Board LEDs: Red LED, power indicator Green LED, on when sound is detected

Dimensions: 1.73 x 0.83 x 0.25 (inches)

AND YOUR TREAT

Of course I couldn't leave this alone. Above is a pic of my wiring for ten (10) LEDs responding to the sound sensor. The LEDs are wired into five groups, with two LEDs in each group. I included a video of these LEDs in action. Hope you like Tim Turner (Rollin on the River).

As always, thank you for reading this and if you found any errors or omissions please do not be shy about telling me. It's much rather to something than know as you floating around to mess someone up.