

# **Prototipe Sistem Peringatan Dini Kebakaran Hutan Menggunakan Sensor Suhu dan Arduino**

## **Tugas Akhir**

Disampaikan Sebagai Bagian Dari Persyaratan Kelulusan Diploma 3  
Program Studi Teknik Komputer

### **Oleh :**

13314020 Jenius Kalpin  
13314021 Tia Monica Silalahi  
13314029 Ayu Putri Hijau



**Institut Teknologi Del**  
**2016/2017**



**Lembar Pengesahan Tugas Akhir  
Institut Teknologi Del**

**Prototipe Sistem Peringatan Dini Kebakaran  
Hutan Menggunakan Sensor Suhu dan Arduino**

institut teknologi

Oleh:

13314020 Jenius Kalpin  
13314021 Tia Monica Silalahi  
13314029 Ayu Putri Hijau

Sitoluama, 31 Agustus 2017

Pembimbing

Maisevli Harika, M.T, M.Eng

Dinyatakan memenuhi syarat dan karenanya disetujui dan disahkan sebagai  
**Laporan Tugas Akhir Diploma 3  
Program Studi Teknik Komputer  
Institut Teknologi Del**

## Prakata

Puji dan syukur dipanjatkan kepada Tuhan Yang Maha Esa atas berkat-Nya yang menyertai penulis selama proses pengerjaan Tugas Akhir yang berjudul “*Prototipe Sistem Peringatan Dini Kebakaran Hutan Menggunakan Sensor Suhu dan Arduino*” sampai Tugas Akhir ini selesai dengan lancar dan tepat waktu.

Penulis juga menyampaikan ucapan terima kasih kepada:

1. Bapak Maisevli Harika, M.T, M.Eng selaku dosen pembimbing yang telah memberikan masukan, serta arahan dan memberikan banyak waktu untuk membimbing.
2. Koordinator Tugas Akhir 2016 yang memberikan arahan tahapan pengerjaan Tugas Akhir.
3. Rekan-rekan mahasiswa dan non-mahasiswa yang telah berbagi pengetahuan dan pengalaman.

Semoga laporan dari tugas akhir ini bisa bermanfaat bagi semua pihak. Laporan Tugas Akhir ini juga tidak lepas dari kesalahan, oleh karena itu penulis mengharapkan saran dan kritik yang membangun untuk menjadi referensi perbaikan di masa mendatang.

Sitoluama, 31 Agustus 2017

Jenius Kalpin 13314020

Tia Monica Silalahi 13314021

Ayu Putri Hijau 13314029

## Abstrak

Kebakaran hutan merupakan salah satu masalah yang mengancam kelestarian lingkungan hutan. Pada umumnya kebakaran hutan disebabkan oleh keserakahan dan kelalaian manusia yang dapat mengakibatkan kerusakan lingkungan. Karena alasan-alasan ini sehingga sangat diperlukan sebuah sistem peringatan dini kebakaran hutan untuk memonitoring terjadinya kebakaran hutan. Akan tetapi, luasnya hutan menjadi salah satu masalah dalam peringatan dini kebakaran hutan. Oleh karena itu, solusi yang diajukan menggunakan sensor suhu **LM35** sebagai pendeteksi kebakaran penanganan bencana kebakaran hutan dan dilengkapi dengan *router* sebagai transmisi datanya.

Tugas Akhir ini menerapkan sistem yang diajukan dalam bentuk prototipe. Penggunaan teknologi sensor *network* ditujukan untuk menghindari kebakaran hutan yang lebih parah. Sehingga kita dapat melakukan penanganan sebelum terjadinya kebakaran hutan secara meluas.

Hasil pengujian yang telah dilakukan yaitu bahwa sistem ini mampu bekerja dengan baik pada lahan 1x1 meter dimana jarak antar sensor yaitu 50 cm. Ketika minimal 2 sensor yang berdekatan mendeteksi suhu  $35^{\circ}\text{C} < x < 75^{\circ}\text{C}$  maka led kuning pada *server* akan menyala. Dan ketika minimal 2 sensor berdekatan mendeteksi suhu  $> 75^{\circ}\text{C}$  maka led merah pada *server* akan menyala dan buzzer akan mengeluarkan bunyi peringatan. Sensor suhu **LM35** mendeteksi suhu mulai dari  $-55^{\circ}\text{C}$  sampai dengan  $150^{\circ}\text{C}$ , sedangkan suhu yang dianggap bahaya adalah  $> 75^{\circ}\text{C}$ .

**Kata kunci:** kebakaran hutan, prototipe, sensor suhu, router

# DAFTAR ISI

Prakata .....	2
Abstrak .....	3
Bab I   Pendahuluan .....	7
1.1   Latar Belakang .....	7
1.2   Tujuan.....	8
1.3   Lingkup .....	8
1.4   Metodologi Penelitian .....	8
1.5   Sistematika Penyajian.....	10
Bab II   Tinjauan Pustaka .....	12
2.1   Sistem Peringatan Dini Kebakaran .....	12
2.2   Jaringan LAN .....	13
2.2.2   Topologi LAN.....	14
2.3   Arduino Uno .....	16
2.4   Arduino <i>Ethernet Shield</i> .....	17
2.5 <i>Fuzzy Logic</i> .....	19
2.6   Sensor Suhu .....	20
2.7   LED ( <i>Light Emmiting Diode</i> ) .....	21
2.8 <i>Buzzer</i> .....	22
2.9 <i>Router</i> .....	22
2.10 <i>Software</i> Arduino IDE .....	23
Bab III   Analisis dan Perancangan Sistem.....	25
3.1   Deskripsi Sistem.....	25
3.2   Kebutuhan <i>Software</i> .....	27
3.2.1 <i>Windows</i> .....	27
3.2.2 <i>Software</i> Arduino .....	27
3.3   Kebutuhan <i>Hardware</i> .....	28
3.3.1   Arduino Uno .....	28
3.3.2   Sensor Suhu LM35 .....	28
3.3.4 <i>Buzzer</i> .....	28
3.3.5 <i>Ethernet Shield</i> .....	28
3.3.7   Laptop .....	28
3.4   Rancangan Sistem .....	28
3.4.1   Rancangan Pemasangan <i>Hardware</i> .....	29
3.4.2 <i>Flowchart</i> Peringatan Kebakaran.....	29
Bab IV   Implementasi dan Pengujian.....	31
4.1   Implementasi .....	31
4.1.1   Rangkaian Sensor Suhu LM35 dengan Arduino .....	31
4.1.2   Konfigurasi <i>Router</i> .....	32
4.1.3   Konfigurasi Arduino <i>Ethernet Shield</i> .....	33
4.1.4   Rangkaian LED dan <i>Buzzer</i> pada <i>Server</i> .....	33
4.2   Pengujian.....	34
4.2.1   Cara Pengujian Sensor Suhu LM35 .....	34
4.2.2   Cara Pengujian <i>Router</i> .....	36
4.2.3   Cara Pengujian LED dan <i>Buzzer</i> .....	37
Bab V   Kesimpulan dan Saran.....	39
5.1   Kesimpulan.....	39
5.2   Saran.....	39
Daftar Pustaka .....	40
Lampiran .....	42

## DAFTAR GAMBAR

Gambar 1. Metodologi Penelitian.....	9
Gambar 2. Topologi Bus [6].....	14
Gambar 3. Topologi <i>Ring</i> [6] .....	15
Gambar 4. Topologi <i>Star</i> [6] .....	15
Gambar 5. Topologi <i>Tree</i> [6].....	16
Gambar 6. <i>Pin Out</i> Arduino Uno.....	17
Gambar 7. <i>Ethernet Shield</i> .....	18
Gambar 8. Penggabungan Arduino dan <i>Ethernet Shield</i> .....	18
Gambar 9. Sensor Suhu LM35 .....	20
Gambar 10. LED .....	21
Gambar 11. <i>Buzzer</i> .....	22
Gambar 12. Tampilan Awal <i>Software</i> Arduino IDE .....	23
Gambar 13. <i>Software</i> Arduino IDE .....	24
Gambar 14. Diagram Blok Sistem.....	26
Gambar 15. Rancangan Peletakan <i>Hardware</i> .....	27
Gambar 16. Rancangan Pemasangan <i>Hardware</i> .....	29
Gambar 17. <i>Flowchart</i> Peringatan Kebakaran .....	30
Gambar 18. Rangkaian Sensor Suhu LM35 dengan Arduino.....	31
Gambar 19. Halaman Utama <i>Web</i> untuk Konfigurasi <i>Router</i> .....	32
Gambar 20. Halaman <i>Login</i> .....	32
Gambar 21. <i>LAN Interface Setting</i> .....	33
Gambar 22. Rangkaian LED dan <i>Buzzer</i> .....	34
Gambar 23. Pengujian Sensor Suhu LM35 di PCB .....	35
Gambar 24. Hasil Kode Program untuk Sensor Pada <i>Serial Monitor</i> .....	35
Gambar 25. Pengujian LED dan <i>Buzzer</i> di PCB .....	37
Gambar 26. Hasil Kode Program <i>Server</i> Pada <i>Serial Monitor</i> .....	38

## DAFTAR TABEL

Tabel 1. Syarat Dinyatakan Kebakaran oleh Sistem.....	25
---	----





# Bab I

## Pendahuluan

Pada bab ini dijelaskan mengenai latar belakang, tujuan pemilihan topik, lingkup, pendekatan yang dilakukan, dan sistematika penyajiannya.

### 1.1 Latar Belakang

Beberapa faktor penyebab kebakaran di Indonesia antara lain, pembakaran hutan yang bertujuan untuk membuka lahan baru, sambaran petir pada hutan yang kering karena musim kemarau yang panjang, ataupun akibat dari kecerobohan manusia seperti membuang puntung rokok secara sembarang di sekitar pinggiran hutan.

Berdasarkan data dan informasi bencana Indonesia, pada tahun-tahun belakangan ini sering terjadi kebakaran hutan terutama di provinsi Sumatera Utara. Kabut asap yang diakibatkan dari hasil kebakaran hutan sudah sangat mengkhawatirkan. Untuk mengatasi permasalahan ini diperlukan perhatian khusus dan penanganan serius dari berbagai pihak. Selain itu dapat digunakan teknologi sensor *network* untuk mengetahui terjadinya kebakaran dan mencegah meluasnya kebakaran. Kita dapat membangun sebuah *detector* kebakaran hutan untuk menangani masalah ini.

Penelitian tentang *detector* bencana kebakaran hutan telah banyak dilakukan sebelumnya. Salah satu contoh dari penelitian tentang deteksi tersebut adalah Prototipe Sistem Telemetri Berbasis Sensor Suhu dan Sensor Asap untuk Pemantau Kebakaran Lahan oleh Boni Pahlanop Lapanporo[1]. Penelitian ini berhasil merancang alat untuk mendeteksi kenaikan suhu dan keberadaan asap. Contoh penelitian lainnya adalah Rancang Bangun Sistem Pendeteksi Dini Kebakaran Lahan Gambut Jenis Kayuan Dengan Memanfaatkan Karakteristik Panas Yang Ditimbulkannya oleh Gilang Baskara dan Rahyul Amri[2]. Penelitian ini merancang sistem deteksi kebakaran menggunakan sensor panas dengan memanfaatkan karakteristik panas yang ditimbulkan oleh kebakaran lahan gambut. Sistem pendeteksi panas menggunakan sensor suhu **LM35**, dimana sensor ini akan dipasang pada batang aluminium yang ditancapkan ke dalam lahan gambut.

Pada Tugas Akhir ini kami ingin membangun suatu prototipe berbasis **Arduino Uno** dan sensor suhu **LM35** untuk meminimalisir bahkan mencegah terjadinya kebakaran hutan. Alasan kami ingin membuat prototipe detektor kebakaran ini yaitu untuk membantu pemerintah dalam penanganan situasi kebakaran. Dengan demikian petugas kehutanan

dapat mengatasi kebakaran hutan dengan segera, sehingga dapat mencegah potensi meluasnya kebakaran tersebut.

Konsep dari prototipe sistem yang akan dibangun yaitu meletakkan sensor pada daerah monitoring (lahan prototipe berukuran 1x1 meter). *Ethernet Shield* yang digunakan dalam prototipe ini adalah sebagai pengirim dan penerima data. Setiap sensor suhu mendeteksi ukuran suhu dan **Arduino Uno** akan mengirim data suhu yang dideteksi oleh sensor ke *server* melalui *Ethernet Shield*. Perbedaan prototipe sistem yang akan dibuat pada Tugas Akhir ini dibandingkan dengan penelitian terkait adalah sistem ini menggunakan metode *fuzzy logic* dimana metode tersebut digunakan sebagai syarat terjadinya kebakaran hutan.

## 1.2 Tujuan

Tujuan dari pengerjaan Tugas Akhir ini adalah untuk merancang dan membangun sebuah prototipe peringatan dini kebakaran menggunakan **Arduino Uno** dan sensor suhu **LM35** yang dapat membantu dalam mencegah penyebaran kebakaran hutan.

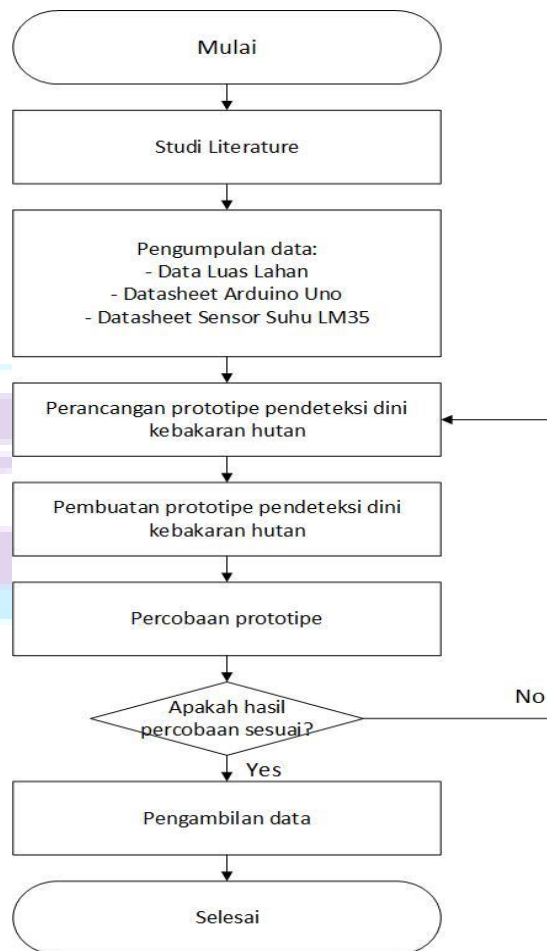
## 1.3 Lingkup

Batasan dan ruang lingkup dalam pengerjaan Tugas Akhir ini adalah:

- Pengerjaan tugas akhir ini difokuskan untuk membuat prototipe alat peringatan kebakaran hutan untuk mengukur suhu pada lahan prototipe.
- Prototipe lahan yang digunakan adalah 1 m<sup>2</sup>.
- Jaringan yang digunakan untuk komunikasi data adalah *Ethernet*.

## 1.4 Metodologi Penelitian

Adapun metodologi penelitian yang kami gunakan dapat dilihat pada Gambar 1.



**Gambar 1. Metodologi Penelitian**

### 1. Studi Literatur

Pada tahap ini kami mencari jurnal atau penelitian orang lain tentang *detector* kebakaran dan berdasarkan penelitian tersebut kami melakukan analisis dimana kami menganalisis kekurangan dan kelebihan dari alat-alat yang digunakan pada sistem yang ada pada penelitian tersebut. Berdasarkan hal tersebut kami akan merancang sebuah sistem *detector* dimana sistem yang dirancang tersebut akan menutupi kekurangan pada sistem atau penelitian tersebut.

### 2. Pengumpulan data

Pada tahap ini dilakukan pengumpulan data dengan cara mengumpulkan dan mendapatkan sejumlah informasi melalui data yang meliputi luas lahan, *datasheet* **Arduino Uno** dan *datasheet* sensor suhu **LM35**.

### 3. Perancangan

Pada tahap ini kami merancang dan membuat desain prototipe sistem yang akan diterapkan. Sebelum melakukan perancangan kami menentukan alat-alat yang akan kami gunakan berdasarkan analisis alat-alat yang digunakan pada penelitian yang telah diteliti.

### 4. Pembuatan prototipe

Pada tahap ini kami membuat sebuah prototipe sistem berdasarkan hasil dari rancangan yang telah kami lakukan sebelumnya.

### 5. Pengujian

Pada tahap ini dilakukan pengujian terhadap prototipe sistem, dimana setelah melakukan pengujian terhadap prototipe sistem kami akan melakukan perbandingan dengan sistem yang ada pada penelitian yang telah dianalisis sebelumnya untuk membuktikan bahwa sistem yang dirancang sekarang adalah sistem yang lebih baik dari pada sistem yang sebelumnya.

### 6. Pengambilan data

Pada tahap ini dilakukan pengambilan data dari setiap sensor dimana data-data tersebut akan menjadi acuan dalam menentukan kemungkinan terjadinya kebakaran hutan.

## 1.5 Sistematika Penyajian

Adapun sistematika penulisan laporan tugas akhir ini adalah sebagai berikut:

#### 1. Bab I. Pendahuluan

Pada Bab ini dijelaskan tentang latar belakang, tujuan, lingkup kajian, pendekatan dan sistematika penyajian laporan.

#### 2. Bab II. Tinjauan Pustaka

Pada Bab ini, dijelaskan tentang teori yang pada dasarnya mencakup rangkuman informasi yang dikumpulkan yang relevan dengan permasalahan yang menjadi objek kajian. Objek kajian akan digunakan untuk memperluas informasi dasar sebagai argumentasi dasar dalam mengemukakan pendapat dan pandangan.

#### 3. Bab III. Analisis dan Perancangan

Pada Bab ini, dijelaskan tentang hasil analisis kebutuhan *hardware* dan kebutuhan *software* yang digunakan oleh sistem. Berdasarkan hasil analisis, dapat dirancang sistem target yang akan dibuat sesuai dengan pandangan dari studi literatur.

#### 4. Bab IV. Implementasi dan Pengujian

Pada Bab ini dijelaskan mengenai implementasi dan pengujian sistem yang telah dibangun.

## 5. Bab V. Kesimpulan dan Saran

Pada Bab ini dijelaskan tentang kesimpulan dan saran dari hasil yang dicapai selama pengerjaan tugas akhir.



## Bab II

### Tinjauan Pustaka

Pada Bab ini, dijelaskan tentang teori yang pada dasarnya mencakup rangkuman informasi yang dikumpulkan yang relevan dengan permasalahan yang menjadi objek kajian. Objek kajian akan digunakan untuk memperluas informasi dasar sebagai argumentasi dasar dalam mengemukakan pendapat dan pandangan.

#### 2.1 Sistem Peringatan Dini Kebakaran Hutan

Penelitian tentang sistem peringatan dini telah banyak dilakukan sebelumnya. Salah satu contoh dari penelitian tentang sistem peringatan dini adalah Pengukur Kelembaban Tanah dan Suhu Udara sebagai Pendeteksi Dini Kebakaran Hutan Melalui *Wireless Sensor Network* (WSN) *Software* oleh Ferly Octavianes, dkk[3]. Penelitian ini bertujuan untuk meminimalisir keterlambatan penanganan kebakaran hutan. Sistem monitoring ini dilakukan dengan mengimplementasikan sensor kelembaban tanah, sensor suhu dan sensor kelembaban udara dengan menggunakan KYL-500S sebagai pengirim datanya dan data-data yang diterima sensor akan ditampilkan melalui *user interface*. Sistem monitoring ini terdiri dari beberapa *node* dimana setiap nodenya terdapat beberapa sensor. Hasil dari pembacaan sensor seperti sensor kelembaban tanah, sensor suhu dan sensor kelembaban udara akan dikirimkan menggunakan media *wireless* dan akan ditampilkan pada sebuah grafik pada *server*.

Contoh penelitian lainnya adalah Sistem Pendeteksi Suhu dan Asap Pada Ruang Tertutup Memanfaatkan Sensor LM35 dan Sensor AF30 oleh Ilona Usuman dan Hasmi Ardhi[4]. Tujuan penelitian ini adalah mendeteksi lebih dini terjadinya kebakaran dengan mendeteksi kenaikan suhu yang disebabkan oleh keberadaan api. Cara kerja sistem ini yaitu ketika sistem mendeteksi panas dan asap maka pompa air, *buzzer* dan kipas yang terhubung ke *relay* akan bekerja, selanjutnya SMS akan dikirim ke pengguna. Sistem ini menggunakan kotak yang menyerupai sebuah ruangan sehingga pompa air mampu menjalankan tugasnya hingga hampir keseluruhan kotak. Dari hasil pengujian, sistem akan mengirim SMS kepada *user* setiap terjadi kebakaran dan mengirim SMS setiap empat menit jika ruangan dalam kondisi aman. Peringatan dini bekerja menyalakan *buzzer*, kipas, pompa air, dan sms jika suhu mencapai 55 derajat dan sensor asap mencapai 120 mV atau bahkan lebih.

## 2.2 Jaringan LAN

LAN (*Local Area Network*) adalah suatu jaringan yang mencakup area yang terbatas, misalnya suatu gedung dalam suatu kantor mempunyai komputer dimana jaringan komputer tersebut secara fisik berdekatan satu sama lainnya[5].

Keuntungan menggunakan jaringan LAN:

1. Pertukaran *file*
2. Pemakaian *printer* dapat dilakukan oleh semua klien.
3. *File* data dapat disimpan pada *server*, sehingga dapat diakses semua klien
4. *File* data yang keluar dari *server* ataupun masuk ke *server* dapat dikendalikan.
5. Proses *back up* data menjadi lebih mudah
6. Resiko data terkena *virus* komputer menjadi sangat kecil
7. Komunikasi dalam suatu gedung dapat dilakukan dengan menggunakan *chat*.

### 2.2.1 Media Transmisi

Media transmisi adalah medium fisik yang menjadi penghubung antara pengirim dan penerima informasi. Dikarenakan jarak yang jauh, maka informasi tersebut dimanipulasi menjadi kode dan kode tersebut akan diubah kembali menjadi data dengan cara apapun. Jenis media transmisi ada dua yaitu *Guided* dan *Unguided*[6].

#### 1. Media Transmisi *Guided*

Media transmisi *guided* menyediakan jalur transmisi sinyal yang terbatas secara fisik. Sinyal-sinyal yang melewati media transmisi *guided* akan diarahkan dan dibatasi oleh batas fisik media. Berikut adalah jenis dari media transmisi *guided*:

##### a. Kabel *Twisted-Pair*

Kabel *twisted-pair* terdiri atas dua jenis yaitu *shielded twisted pair* dan *unshielded twisted pair*. Kabel *shielded twisted pair* sering diterapkan untuk topologi *token ring*. Contoh dari kabel *unshielded twisted* adalah kabel RJ-45. Kabel *twisted-pair* terdiri atas dua pasang kawat yang terpilin. *Twisted-pair* lebih tipis, mudah putus, dan mengalami gangguan lain sewaktu kabel terpuntir atau kusut. Keunggulan dari kabel *twisted pair* adalah jika sebagian kabel *twisted pair* rusak, tidak seluruh jaringan berhenti.

##### b. Kabel Koaksial

Kabel koaksial terdiri atas konduktor silindris melingkar yang mengelilingi sebuah kabel tembaga inti yang konduktif. Keunggulan dari kabel koaksial adalah dapat dijalankan tanpa banyak *repeater* sebagai penguat untuk komunikasi jarak jauh diantara *node network*.



### c. Kabel Fiber-Optik

Kabel fiber optik merupakan media *networking* yang mampu digunakan untuk transmisi modulasi. Kabel fiber optik dapat mentransmisikan puluhan juta bit digital perdetik pada link kabel optik yang beroperasi dalam sebuah jaringan komersial. Berikut adalah keuntungan kabel fiber optik:

- Jaringan *fiber optic* beroperasi pada kecepatan tinggi
- *Fiber optic* mampu membawa paket-paket dengan kapasitas yang besar
- Sinyal-sinyal yang ditransmisikan lebih jauh tanpa memerlukan *refresh*
- Daya tahan terhadap imas elektromagnetik lebih kuat
- Biaya untuk perawatan kabel fiber optik lebih murah

## 2. Media Transmisi *Unguided*

Media transmisi *unguided* mentransmisikan gelombang elektromagnetik tanpa menggunakan konduktor fisik seperti kabel atau serat optik. Media transmisi *unguided* terdiri atas empat bagian yaitu:

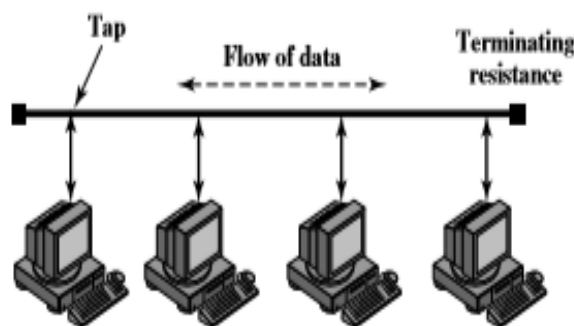
- a. Gelombang Mikro Terrestrial
- b. Gelombang Mikro Satelit
- c. Radio Broadcast
- d. Infra Merah

### 2.2.2 Topologi LAN

Topologi fisik jaringan adalah cara yang digunakan untuk menghubungkan *workstation-workstation* di dalam LAN tersebut[7]. Macam-macam topologi jaringan fisik, antara lain:

#### 1. Topologi Bus atau *Linear*

Topologi bus merupakan topologi yang banyak digunakan untuk penggunaan kabel koaksial. Karakteristik dari topologi bus adalah satu kabel dimana kedua ujungnya ditutup dan pada sepanjang kabel terdapat *node-node*. Topologi ini termasuk topologi paling *prevealent* karena sederhana dalam instalasi.

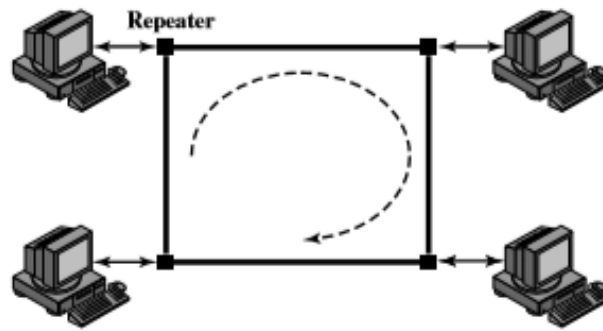


Gambar 2. Topologi Bus [8]



## 2. Topologi *Ring*

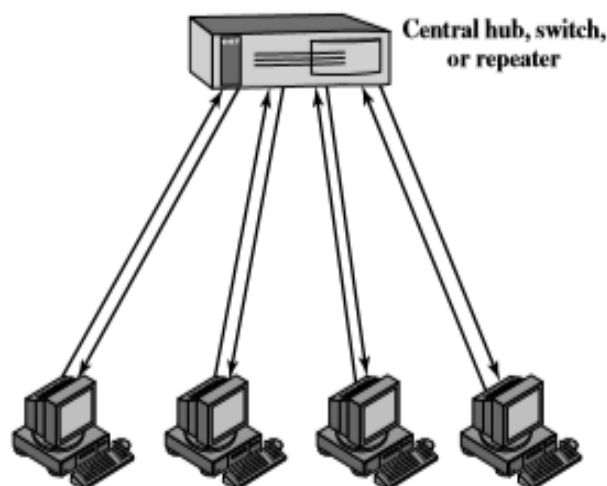
Topologi *ring* adalah yang memanfaatkan fasilitas *fiber optic* pada umumnya. Karakteristik topologi ini adalah berisi *node-node* pada lingkaran tertutup, sederhana dalam *layout*, dan memperkecil adanya *collision* karena signal hanya mengalir satu arah.



Gambar 3. Topologi *Ring* [8]

## 3. Topologi *Star*

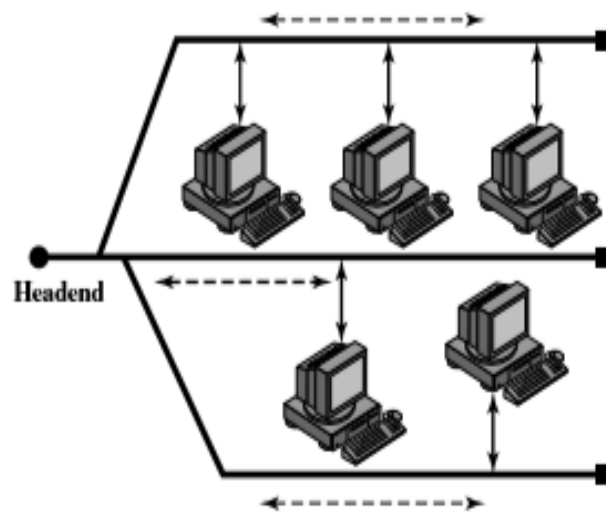
Topologi *star* merupakan topologi yang banyak digunakan karena kemudahan untuk menambah, mengurangi, atau mendeteksi kerusakan jaringannya. Karakteristik topologi ini yaitu setiap *node* berkomunikasi langsung dengan *central node*, *traffic* data mengalir dari *node* ke *central node* dan kembali lagi, mudah dikembangkan karena setiap *node* hanya memiliki kabel yang langsung terhubung ke *central node*, keunggulan jika satu kabel *node* terputus maka yang lainnya tidak akan terganggu.



Gambar 4. Topologi *Star* [8]

#### 4. Topologi *Tree*

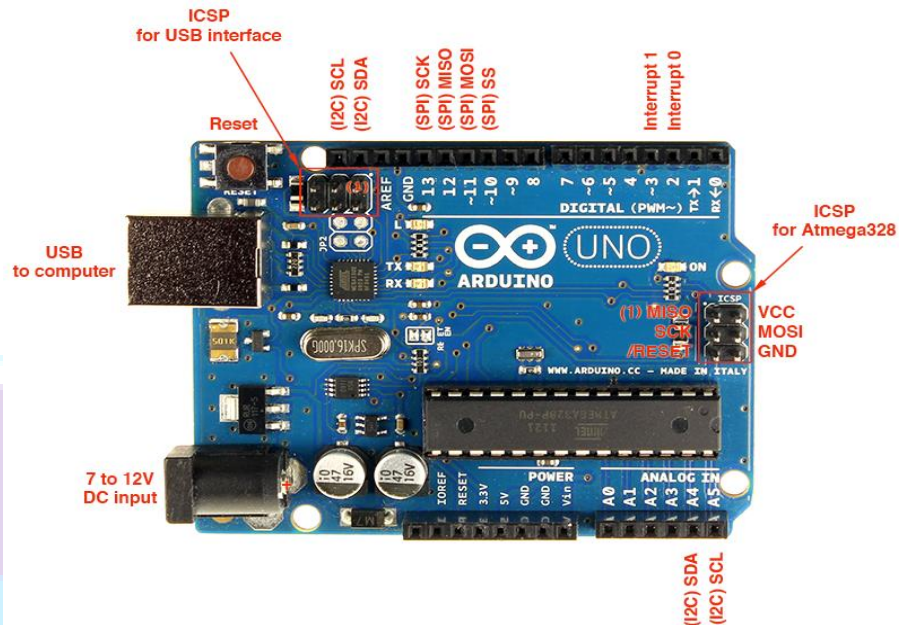
Merupakan topologi gabungan dari ketiga topologi yaitu topologi *star*, topologi *ring*, dan topologi *bus*. Topologi ini memungkinkan untuk mengembangkan jaringan yang sudah ada dan mengkonfigurasinya sesuai dengan kebutuhan. Keunggulan dari topologi ini adalah proses konfigurasi jaringan dilakukan dari titik ke titik pada masing-masing segmen dan topologi ini didukung oleh banyak perangkat keras dan perangkat lunak.



Gambar 5. Topologi *Tree* [8]

### 2.3 Arduino Uno

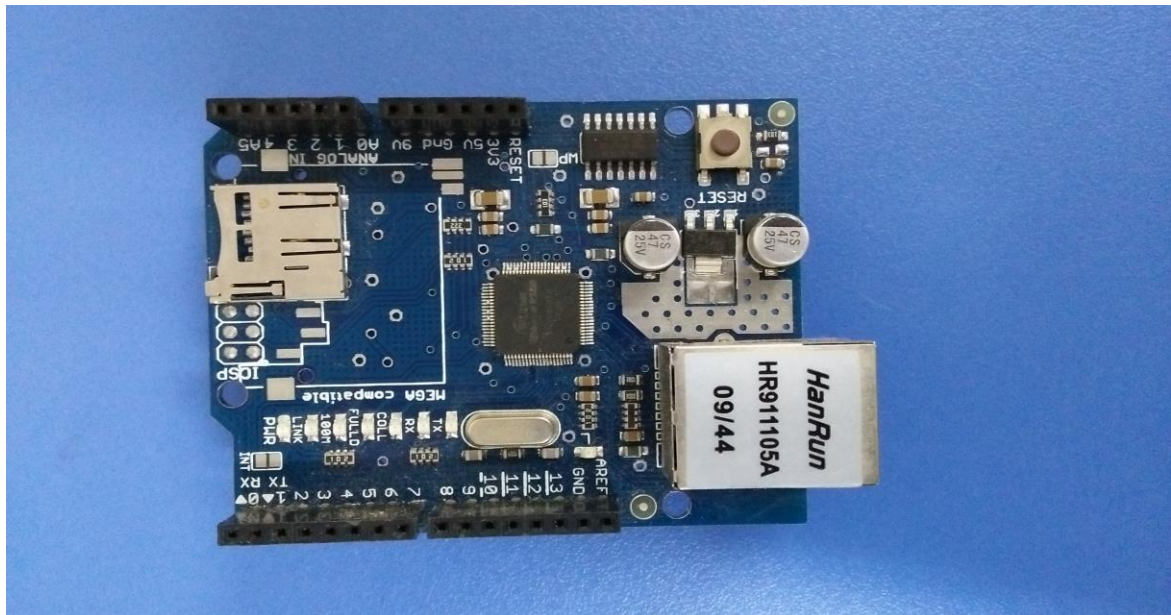
**Arduino Uno** merupakan *board* mikrokontroler berbasis chip **ATmega328**. *Board* ini berfungsi sebagai arena *prototyping* sirkuit mikrokontroler sehingga kita lebih mudah untuk merangkai rangkaian elektronika. *Board* ini memiliki 14 digital *input/output*, 6 pin *input* analog dengan menggunakan crystal 16MHz, kabel USB, jack listrik, *header* ICSP dan tombol reset. Hal-hal tersebut sangat diperlukan untuk mendukung rangkaian mikrokontroler yang kita buat. *Pin out* pada Arduino Uno dapat dilihat pada Gambar 6 berikut.



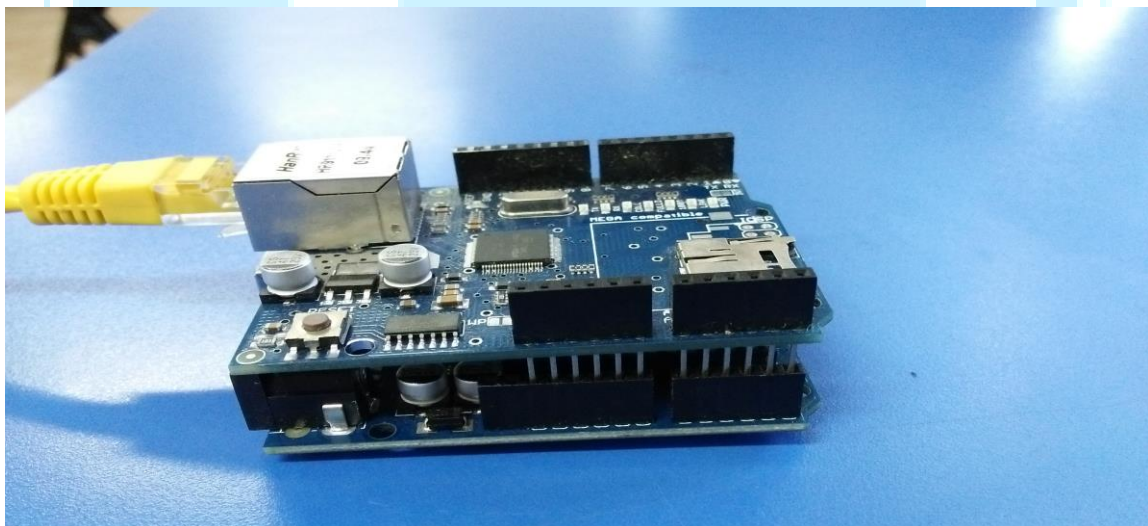
Gambar 6. Pin Out Arduino Uno

## 2.4 Arduino Ethernet Shield

**Arduino Ethernet Shield** adalah suatu perangkat yang menghubungkan *board* Arduino dengan jaringan internet. *Ethernet shield* berbasis chip W5100 Wiznet Ethernet. Penggunaan *Ethernet shield* mengacu pada *library Ethernet shield* yang berisi perintah atau instruksi agar *board* Arduino dapat terhubung ke jaringan dengan menggunakan *Ethernet shield*. Untuk pemasangan *Ethernet shield* dengan *board* Arduino yaitu dengan memasang *board* Arduino di atas modul *Ethernet shield* dan menghubungkannya ke jaringan Internet dengan menggunakan kabel RJ-45. *Ethernet shield* mempunyai koneksi port RJ-45 yang standar, dengan trafo garis yang terintegrasi dan *power over Ethernet* yang aktif.



**Gambar 7. Ethernet Shield**



**Gambar 8. Penggabungan Arduino dan Ethernet Shield**

Syarat agar suatu perangkat juga dapat terhubung ke jaringan internet yaitu perangkat tersebut harus memiliki IP address. IP address tersebut akan didefinisikan pada program *Ethernet shield* sehingga perangkat tersebut akan dikenali oleh perangkat lainnya dalam jaringan. Berikut ini *requirement* penggunaan modul *Ethernet shield*:

- Bekerja dengan board Arduino melalui port SPI.
- Board Arduino beroperasi dengan tegangan 5V.
- Ethernet Controller W5100 dengan internal buffer sebesar 16K.
- Connection speed mulai dari 10Mb sampai 100Mb.



## 2.5 Fuzzy Logic

*Fuzzy logic* pertama kali diperkenalkan oleh Prof. Lotfi Zadeh melalui papernya pada tahun 1965 mengenai himpunan teori fuzzy. *Fuzzy logic* dikembangkan di Amerika, akan tetapi *fuzzy logic* lebih populer dan banyak diaplikasikan oleh praktisi Jepang dengan mengadaptasikannya ke bidang kontrol. Hal ini dikarenakan kultur orang Barat cenderung memandang suatu hal sebagai hitam-putih, ya-tidak, atau yang setara dengan logika biner Aristotele sedangkan kultur Timur cenderung “abu-abu” atau “samar-samar” dalam memandang suatu persoalan.

Dalam pengertiannya, *fuzzy logic* merupakan suatu logika yang memiliki nilai kekaburan antara benar atau salah. *Fuzzy logic* merupakan pengembangan dari logika Boolean yang mengenalkan konsep kebenaran sebagian. Dalam logika klasik, segala sesuatu diekspresikan dalam *binary* yaitu 0 atau 1, ya atau tidak, dan hitam atau putih. Tidak ada nilai lain di antaranya.

Dengan adanya *fuzzy logic*, segala sesuatu dapat diukur tingkat kebenarannya di antara hitam atau putih (abu-abu). Saat ini konsep *fuzzy logic* sudah banyak diimplementasikan pada produk-produk kebutuhan rumah tangga seperti *vacuum cleaner*, *microwave* dan kamera video. Ada 3 proses utama jika ingin mengimplementasikan konsep *fuzzy logic* pada suatu perangkat yaitu[9]:

1. *Fuzzification*, yaitu suatu proses untuk mengubah suatu masukan dari bentuk tegas (*crisp*) menjadi bentuk variable linguistik (fuzzy) yang biasanya disajikan dalam bentuk himpunan fuzzy dengan suatu fungsi keanggotaannya masing-masing.
2. *Interference System*, yaitu acuan untuk menjelaskan suatu hubungan antara variable-variabel *input* dan *output* dimana variabel yang diproses dan yang dihasilkan berbentuk fuzzy.
3. *Defuzzification*, yaitu proses pengubahan variabel berbentuk fuzzy menjadi data-data pasti (*crisp*) yang dapat dikirimkan ke peralatan pengendalian.

Sistem inferensi *fuzzy* dapat mengambil *input fuzzy* ataupun *crisp*, tetapi *output* yang dihasilkan hampir selalu menghasilkan himpunan *fuzzy*. Terdapat 2 metode inferensi yang paling dikenal yaitu:

1. Metode Mamdani

Metode inferensi Mamdani diusulkan sebagai usaha awal untuk mengendalikan mesin uap dan kombinasi *boiler* dengan sebuah himpunan aturan kendali linguistik yang diperoleh berdasarkan pengalaman operator manusia.

## 2. Metode Tagaki Sugeno

Metode Tagaki Sugeno diusulkan oleh Tagaki Sugeno dalam usaha membangun pendekatan sistematis untuk men-*generate* aturan *fuzzy* dari *dataset input-output* yang diberikan. Secara umum bentuk metode *fuzzy* Tagaki Sugeno adalah[10]:

$$\text{IF}(x_1 \text{ is } A_1) \bullet (x_2 \text{ is } A_2) \bullet \dots \bullet (x_n \text{ is } A_n) \text{ THEN } z = k \quad (2.1)$$

Keterangan:

$A_i$  adalah himpunan *fuzzy* ke- $i$  sebagai *premise*, dan  $k$  adalah suatu konstanta yang dinyatakan sebagai kesimpulan.

### 2.6 Sensor Suhu

Sensor peringatan dini kebakaran hutan ini menggunakan sensor suhu **LM35**. Sensor suhu **LM35** adalah salah satu jenis sensor yang mengubah besaran suhu menjadi besaran listrik dalam bentuk tegangan[11]. Sensor suhu **LM35** memiliki keakuratan yang tinggi dalam membaca data suhu dan perancangan yang lebih mudah jika dibandingkan dengan sensor suhu yang lain.

Sensor ini dapat mencapai tegangan 30 volt, akan tetapi tegangan yang dapat diberikan ke sensor adalah sebesar 5 volt, sehingga dapat digunakan dengan catu daya tunggal dengan ketentuan bahwa **LM35** hanya membutuhkan arus sebesar 60  $\mu\text{A}$ . Hal ini berarti **LM35** memiliki kemampuan pemanasan sendiri (*self-heating*) dari sensor yang dapat menyebabkan kesalahan pembacaan yang rendah yaitu kurang dari 0.5°C pada suhu 25°C. Suhu minimum yang dapat dideteksi oleh sensor suhu **LM35** adalah -55°C dan suhu maksimumnya adalah 150°C [12].



Gambar 9. Sensor Suhu LM35

Dari Gambar 9 diatas dapat diketahui bahwa sensor suhu IC **LM35** pada dasarnya memiliki 3 pin diantaranya yaitu, pin 1 berfungsi sebagai sumber tegangan kerja dari **LM35**(Vs), pin 2 atau tengah digunakan sebagai tegangan keluaran(Vout) dengan jangkauan kerja dari 0 Volt sampai dengan 1,5Volt dengan tegangan operasi sensor suhu **LM35** yang dapat digunakan antara 4Volt sampai 30Volt. Keluaran sensor ini naik sebesar 10mV setiap derajat *celcius* sehingga diperoleh persamaan sebagai berikut [13]:

$$V_{LM35} = \text{Suhu} * 10 \text{ mV} \quad (2.2)$$

Keterangan:

VLM35 adalah tegangan keluaran dari sensor suhu LM35 dan suhu adalah besarnya data suhu yang terukur oleh sensor suhu LM35. *Output* berupa tegangan dari sensor ini akan langsung dimasukkan ke Arduino.

## 2.7 LED (*Light Emmiting Diode*)

*Light Emmiting Diode* atau lebih dikenal dengan LED merupakan rangkaian eletronika yang memancarkan cahaya monokromatik saat diberikan tegangan[14]. Disebut monokromatik karena dioda LED hanya memancarkan satu warna saja. Warna dari cahaya LED tergantung pada panjang gelombang serta bahan semikonduktor yang digunakan. LED juga dapat memancarkan sinar inframerah yang tidak dapat dilihat dengan mata. Contoh penerapannya dapat kita lihat pada *Remote Control* TV, AC, CD/DVD.



Gambar 10. LED

LED memiliki ukuran sepanjang 5mm. Bentuk dari LED ini sendiri mirip dengan bohlam atau bola lampu yang kecil. Dengan bentuknya yang kecil, LED dapat dipasangkan ke berbagai rangkaian eletronika. Untuk mengetahui polaritas LED dapat dilihat dari terminal anoda(+) dan katoda(+) pada LED. Ciri-ciri terminal anoda pada LED adalah kaki LED yang lebih panjang sedangkan terminal katoda LED adalah kaki LED yang

lebih pendek. LED hanya akan menghasilkan cahaya apabila dialiri sebuah tegangan maju dari anoda menuju ke katoda. Saat menghasilkan cahaya, LED tidak menimbulkan panas. Hal ini dikarenakan LED tidak memerlukan pembakaran filamen.

Meskipun LED mempunyai bentuk yang kecil, tetapi LED memiliki banyak sekali pemanfaatannya. Saat ini LED banyak digunakan dalam perangkat elektronik seperti lampu penerangan pada LCD TV. Untuk mengetahui polaritas LED dapat dilihat dari terminal anoda(+) dan katoda(+) pada LED. Ciri-ciri anoda pada LED adalah kaki LED yang lebih panjang sedangkan terminal katoda LED adalah kaki LED yang lebih pendek. LED hanya akan menghasilkan cahaya apabila dialiri sebuah tegangan maju dari anoda menuju ke katoda.

## 2.8 Buzzer

*Buzzer* merupakan rangkaian elektronika yang berfungsi untuk mengubah getaran listrik menjadi getaran suara. Cara kerja *buzzer* hampir sama dengan *loud speaker*. *Buzzer* terdiri dari kumparan yang terpasang pada *diafragma* dimana kumparan tersebut akan dialiri arus sehingga menjadi elektromagnet.

Kumparan tadi akan tertarik kedalam atau keluar berdasarkan arah arus dan polaritas magnetnya. Karena kumparan dipasang pada *diafragma* maka setiap gerakan kumparan akan menggerakkan *diafragma* secara bolak-balik sehingga membuat udara bergetar yang akan menghasilkan suara. Pada Tugas Akhir ini, *buzzer* digunakan sebagai alarm untuk menandakan bahwa telah terjadi suatu kesalahan (*warning*).



Gambar 11. *Buzzer*

## 2.9 Router

**Router** merupakan perangkat keras komputer yang dapat digunakan sebagai perantara untuk menghubungkan jaringan yang sama atau jaringan yang berbeda. Proses pengiriman paket data melalui jaringan atau internet agar sampai ke tujuan disebut proses *routing*.

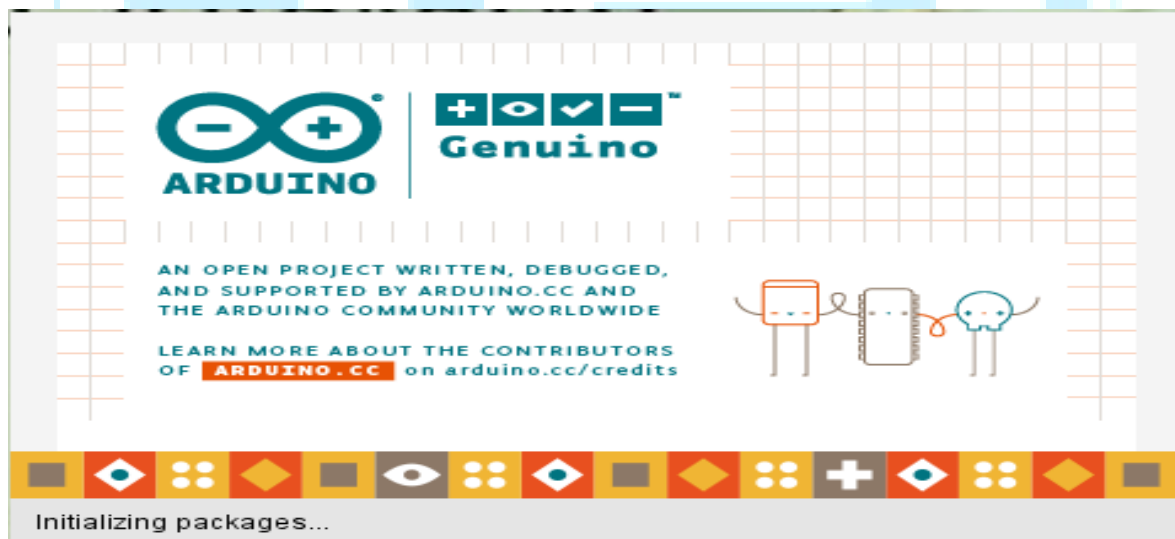


Dengan teknik *routing*, *router* dapat mengetahui arah rute perjalanan paket data menuju *destination*, apakah berada pada jaringan yang sama atau tidak. Jika pengiriman paket data mengarah kepada jaringan yang berbeda, maka *router* akan meneruskannya pada jaringan tersebut. Sebaliknya, apabila paket yang dituju berada dalam jaringan yang sama, maka *router* akan menghalangi paket keluar serta meneruskan paket tersebut dengan *routing* di jaringan yang sama sampai terkirim ke tujuan.

Proses *routing* terjadi pada lapisan ketiga dalam OSI (*Open System Interconnection*) Layer. Fungsi *router* pada umumnya adalah sebagai penghubung antar dua atau lebih jaringan untuk meneruskan pengiriman data dari satu jaringan ke jaringan lainnya. Fungsi utama *router* yaitu sebagai pembagi atau penyalur IP *address*, baik secara statis ataupun DHCP (*Dynamic Host Configuration Protocol*) kepada semua komputer yang terhubung dengan *router* tersebut. Dengan adanya IP *address* yang dibagikan oleh *router*, maka setiap komputer yang saling terhubung dapat saling berkomunikasi, baik dengan menggunakan LAN atau *wireless*.

## 2.10 Software Arduino IDE

Software Arduino IDE digunakan sebagai *editor* teks untuk membuat, membuka, mengedit, dan juga memvalidasi kode program untuk di-*upload* ke papan Arduino. *File* kode program yang di *upload* pada *software* ini menggunakan ekstensi *.ino*.

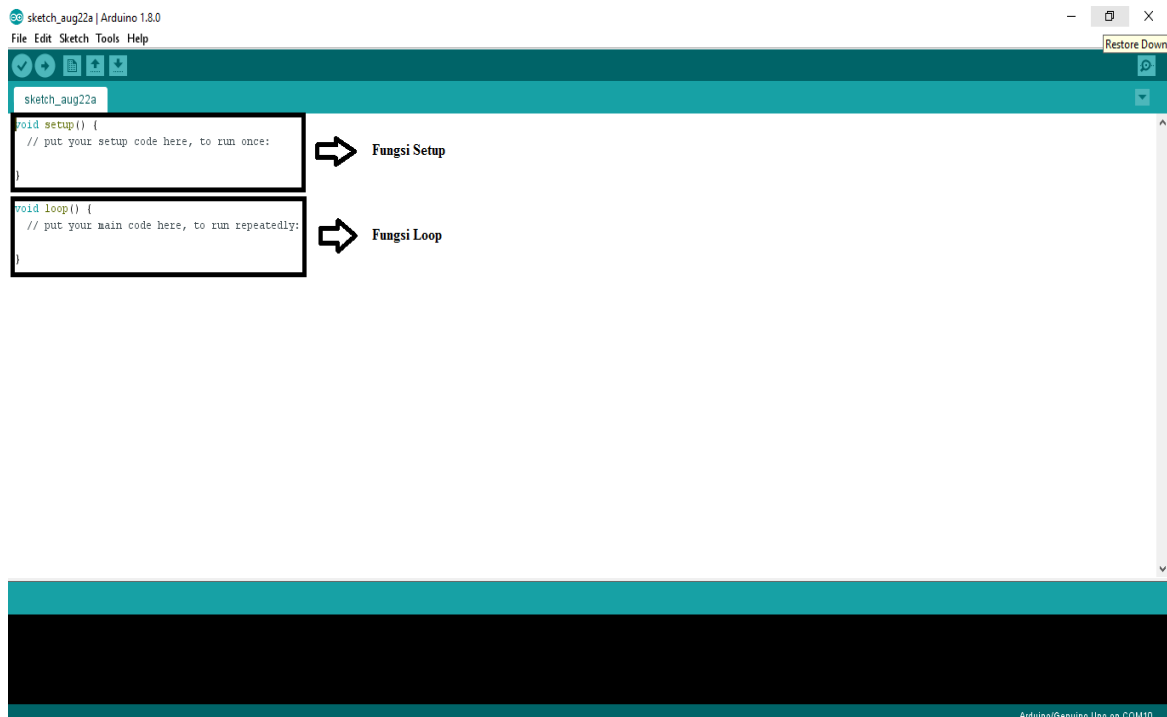


Gambar 12. Tampilan Awal Software Arduino IDE

### 2.10.1 Bagian-bagian Arduino IDE

Arduino IDE dilengkapi dengan fitur *cut* dan *paste*. Pada bagian keterangan akan tampil seperti pesan apakah kode program yang di-*verify* tersebut *error* atau tidak. Dengan adanya *verify* tersebut dapat memudahkan pengguna untuk mengecek kode program yang

*error* berdasarkan barisnya. Pada bagian keterangan juga akan tampil pesan jika kode program yang di-*upload* ke papan **Arduino Uno** sudah berjalan atau belum. Bagian dari struktur pada Arduino IDE dapat dilihat pada Gambar 13 berikut.



**Gambar 13. Software Arduino IDE**

Keterangan:

- Fungsi *setup* dipanggil pertama kali ketika menjalankan *sketch* yang digunakan sebagai tempat, *pin mode*, dan penggunaan *library*. Fungsi ini dijalankan sekali ketika *board* dinyalakan atau di *reset*.
- Fungsi *loop* akan melakukan perulangan berturut-turut, memungkinkan sebuah program dapat diubah. Fungsi ini digunakan untuk mengontrol *board* Arduino.

## Bab III

### Analisis dan Perancangan Sistem

Pada bab ini dijelaskan mengenai analisis kebutuhan *hardware* dan analisis kebutuhan *software* dari sistem.

#### 3.1 Deskripsi Sistem

Pada prototipe sistem peringatan dini kebakaran hutan pada luas lahan sebesar 1x1 meter digunakan 4 buah sensor dimana penyusunan letak sensor akan dibuat sesuai dengan rumus jarak antar tanaman yaitu [15]:

$$\text{Jumlah sensor} = \frac{\text{Luas Lahan}}{\text{Jarak sensor}} \quad (2.3)$$

Prototipe ini disusun dengan rumus tersebut adalah karena jarak maksimum antar sensor adalah 60 cm. Jadi karena luas lahan untuk prototipe adalah 1x1 meter maka jarak antar sensor yang ditentukan adalah 50 cm maka didapatlah hasilnya yaitu empat yang menyatakan sensor yang digunakan adalah empat buah. Prototipe sistem peringatan dini kebakaran hutan dibuat berupa konfigurasi dari sebuah sensor yaitu sensor suhu **LM35**, dimana sensor suhu **LM35** akan melaporkan suhu yang dideteksi kepada arduino kemudian datanya akan dikirim oleh **Arduino Uno** dimana data akan dikirimkan dalam waktu 2 menit setelah data suhu terdeteksi.

Kemudian data suhu dikirim melalui *Ethernet Shield* ke *router*. Dalam pengiriman data, *router* berfungsi untuk menyampaikan data suhu ke *Ethernet Shield* di daerah *server*. Data suhu yang diterima oleh **Arduino Uno** akan diolah dari sinyal analog menjadi sinyal digital.

Setelah data tersebut diolah maka data suhu tersebut akan ditampilkan ke *monitor* pengamat dan *buzzer*. Pada **Arduino Uno** yang ada dilahan prototipe akan ditanamkan *program fuzzy*. *Buzzer* akan berbunyi setelah memenuhi syarat-syarat terjadinya kebakaran. Syarat dinyatakan kebakaran oleh sistem dapat dilihat pada Tabel 1.

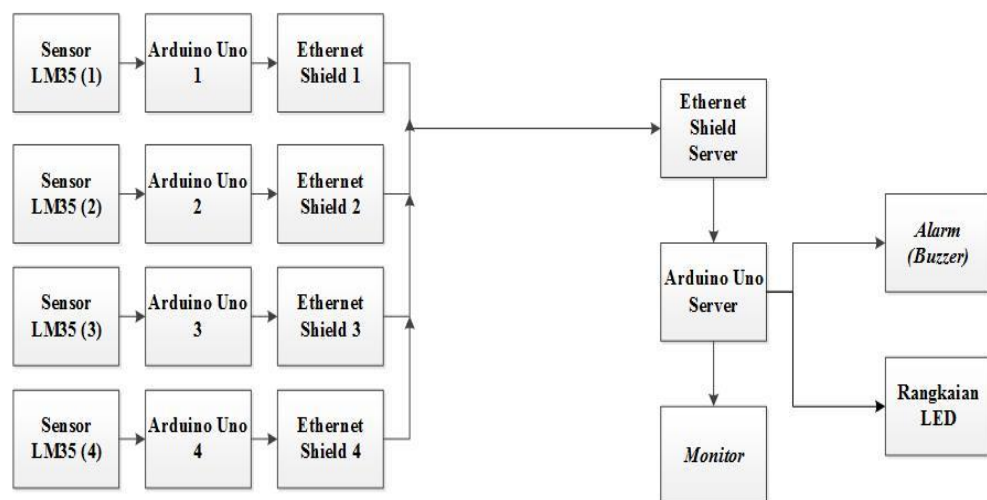
Tabel 1. Syarat Dinyatakan Kebakaran oleh Sistem

Rentang Suhu	Rentang Waktu	Pernyataan
$35^{\circ}\text{C} < x < 75^{\circ}\text{C}$	4 menit	Warning
$> 75^{\circ}\text{C}$	7 menit	Kebakaran

Keterangan:

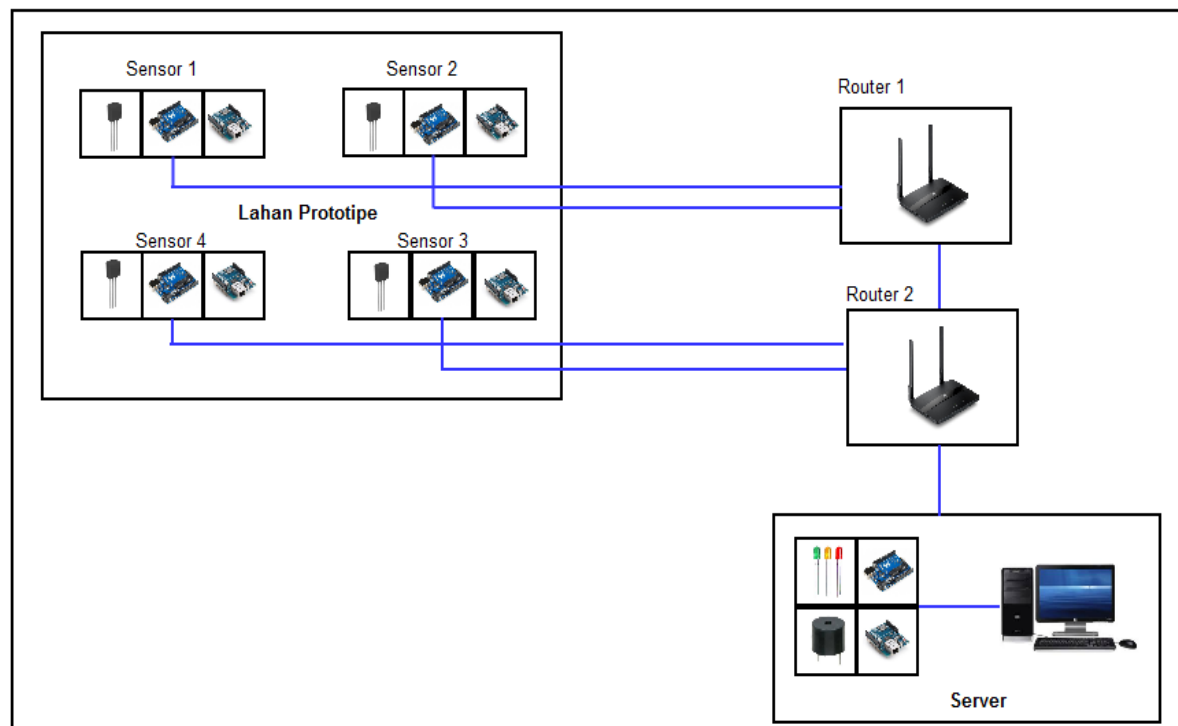
1. Masing-masing sensor akan diberi penamaan yaitu sensor 1, sensor 2, sensor 3, dan sensor 4. Jika minimal dua sensor yang berdekatan mengukur suhu yang meningkat secara drastis yaitu  $>35^{\circ}\text{C}$  hingga mencapai suhu  $75^{\circ}\text{C}$  dalam waktu 4 menit maka dapat dinyatakan sebagai *warning* kebakaran. (Jika suhu tetap bertahan dalam rentang  $>35^{\circ}\text{C}$  sampai  $75^{\circ}\text{C}$  dalam waktu 4 menit maka dapat dinyatakan sebagai *warning* kebakaran).
2. Jika minimal dua sensor membaca data suhu yang meningkat secara drastis yaitu  $>75^{\circ}\text{C}$  hingga mencapai suhu api yaitu  $\leq 150^{\circ}\text{C}$  (berdasarkan *datasheet* sensor suhu **LM35**) dalam waktu 7 menit maka dapat dinyatakan sebagai kebakaran. (Jika suhu tetap bertahan dalam rentang  $>75^{\circ}\text{C}$  sampai  $\leq 150^{\circ}\text{C}$  dalam waktu 7 menit maka dapat dinyatakan sebagai kebakaran).

Diagram blok sistem prototipe ini dapat dilihat pada Gambar 14.



Gambar 14. Diagram Blok Sistem

Rancangan peletakan *hardware* dapat dilihat pada Gambar 15.



**Gambar 15. Rancangan Peletakan Hardware**

Keterangan:

- Pada prototype ini jumlah sensor yang digunakan adalah sebanyak 4 buah.
- Pada prototype ini luas lahan prototype yang telah ditentukan adalah 1x1 meter.
- Pada pengiriman data dari lokasi yang diamati hingga ke *server* digunakan *Ethernet shield* dan *router* yang berfungsi sebagai pengiriman data.
- Rangkaian LED dan *buzzer*, serta monitor diletakkan pada daerah *server*.
- Data suhu yang diukur oleh sensor suhu **LM35** pada lahan prototype akan ditampilkan pada *serial monitor* pada monitor pengamat.

### 3.2 Kebutuhan Software

Kebutuhan *software* ini menjelaskan perangkat lunak yang diperlukan untuk membangun prototype sistem tersebut.

#### 3.2.1 Windows

*Windows* digunakan sebagai sistem untuk menjalankan *software* Arduino IDE dan *software* Netbeans.

#### 3.2.2 Software Arduino

*Software* ini digunakan untuk membuat kode program dan meng-*upload* kode program ke mikrokontroller Arduino agar kode program yang dibuat dapat dijalankan ke sistem.

### 3.3 Kebutuhan *Hardware*

Kebutuhan *hardware* ini menjelaskan perangkat keras yang diperlukan untuk membangun prototipe sistem tersebut.

#### 3.3.1 Arduino Uno

**Arduino Uno** digunakan untuk memproses data dari sensor. Arduino digunakan karena memiliki fitur yang dibutuhkan dan lebih mudah dalam pemakaiannya.

#### 3.3.2 Sensor Suhu LM35

Sensor Suhu **LM35** adalah sensor yang digunakan untuk membaca data suhu karena harganya murah, banyak digunakan, dan data yang dikirimkan adalah akurat.

#### 3.3.3 LED

LED digunakan sebagai penanda apakah terjadi *warning*, kebakaran, atau tidak terjadi hal apapun(normal).

#### 3.3.4 Buzzer

**Buzzer** berfungsi untuk mengirimkan suara dimana jika sensor suhu **LM35** membaca data suhu yang meningkat secara drastis (menandakan terjadinya kebakaran) maka **buzzer** mengeluarkan suara.

#### 3.3.5 Ethernet Shield

*Ethernet Shield* digunakan untuk mengirimkan data dari sensor ke *server* melalui kabel LAN.

#### 3.3.6 Kabel RJ-45

Kabel RJ-45 berfungsi sebagai penghubung atau konektor kabel Ethernet yang digunakan dalam jaringan.

#### 3.3.7 Laptop

Laptop digunakan untuk melihat data yang telah dikirimkan oleh sensor melalui *Ethernet shield*.

#### 3.3.8 Router

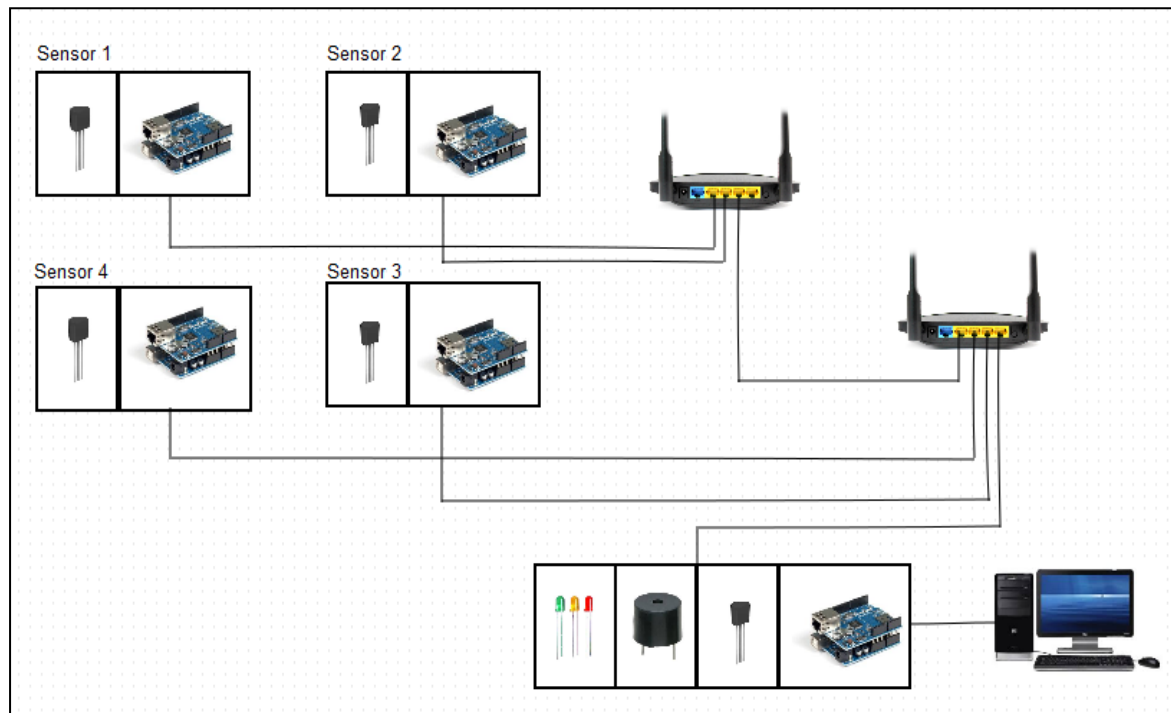
*Router* berfungsi sebagai perantara dalam mentransmisikan data (pengiriman data) dari *Ethernet Shield* yang ada pada lahan prototype ke *Ethernet Shield* yang ada pada *server*.

### 3.4 Rancangan Sistem

Rancangan sistem ini terbagi menjadi 2 bagian, yaitu rancangan pemasangan *hardware* dan rancangan *software* (berbentuk *flowchart*).

### 3.4.1 Rancangan Pemasangan *Hardware*

Prototipe ini membutuhkan rangkaian mikrokontroller yang terdiri dari sensor suhu **LM35**, *buzzer*, *Ethernet Shield* dan *Router* sebagai alat pengiriman data. Pusat operasi dari proses-proses yang dilakukan dalam sistem ini adalah PC (*Monitor Pengamat*). Pada mikrokontroler ditanamkan kode program yang berfungsi untuk mengendalikan kinerja sensor. Rancangan pemasangan *hardware* dapat dilihat pada Gambar 16.

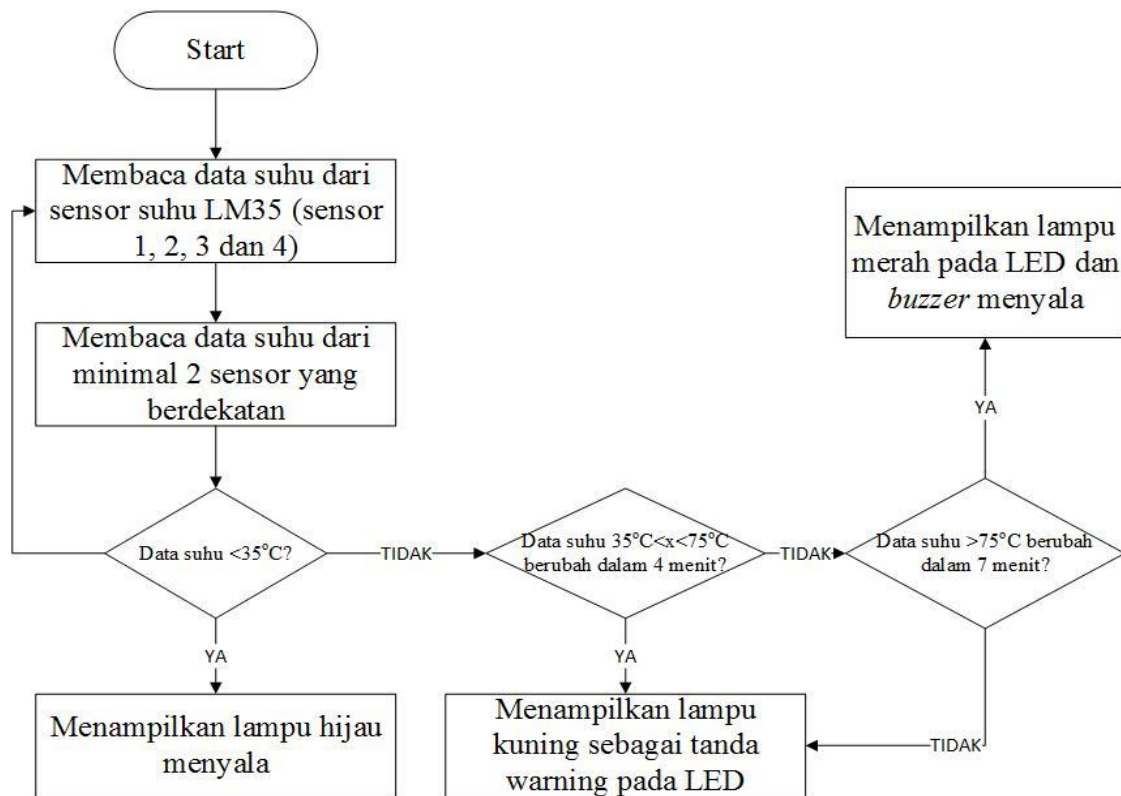


Gambar 16. Rancangan Pemasangan *Hardware*

### 3.4.2 *Flowchart* Peringatan Kebakaran

*Flowchart* peringatan kebakaran yaitu rancangan/sketsa yang berbentuk *flowchart* dari sistem peringatan kebakaran yang telah dibangun. Untuk melihat *flowchart* peringatan kebakaran lebih detail dapat dilihat seperti berikut.





**Gambar 17. Flowchart Peringatan Kebakaran**

Keterangan:

Pada gambar diatas, sensor suhu **LM35** membaca data suhu dan jika sensor suhu **LM35** membaca data suhu  $35 < x < 75^{\circ}\text{C}$  maka data suhu tersebut akan dicek apakah berubah dalam waktu 4 menit jika tidak maka pada LED kuning akan menyala sebagai pertanda *warning* tetapi jika data suhu tersebut berubah dalam waktu 4 menit maka suhu akan dicek lagi sesuai data suhu yaitu  $>75^{\circ}\text{C}$ .

Jika sensor suhu **LM35** membaca data suhu diluar dari suhu  $35 < x < 75^{\circ}\text{C}$  maka data suhu akan dicek lagi yaitu data suhu  $>75^{\circ}\text{C}$  dan jika tidak masuk dalam kedua data suhu tersebut maka akan dicek lagi dengan data suhu yaitu  $<35^{\circ}\text{C}$ .

Jika data suhu yang diukur adalah  $>75^{\circ}\text{C}$  maka akan dicek lagi dalam parameter waktu yaitu dalam waktu 7 menit jika data suhu tidak berubah dalam waktu 7 menit maka LED merah akan menyala dan *buzzer* sebagai *alarm* akan menyala yang menandakan bahwa keadaan tersebut sudah dapat dinyatakan sebagai kebakaran. Tetapi jika data suhu berubah dalam waktu 7 menit maka akan dilanjutkan mengecek data suhu  $<35^{\circ}\text{C}$  dan jika data suhu yang dicek adalah  $<35^{\circ}\text{C}$  maka pada LED hijau menyala sebagai pertanda bahwa keadaan hutan aman. Proses pengecekan ini akan terus berulang secara terus-menerus.



## Bab IV

### Implementasi dan Pengujian

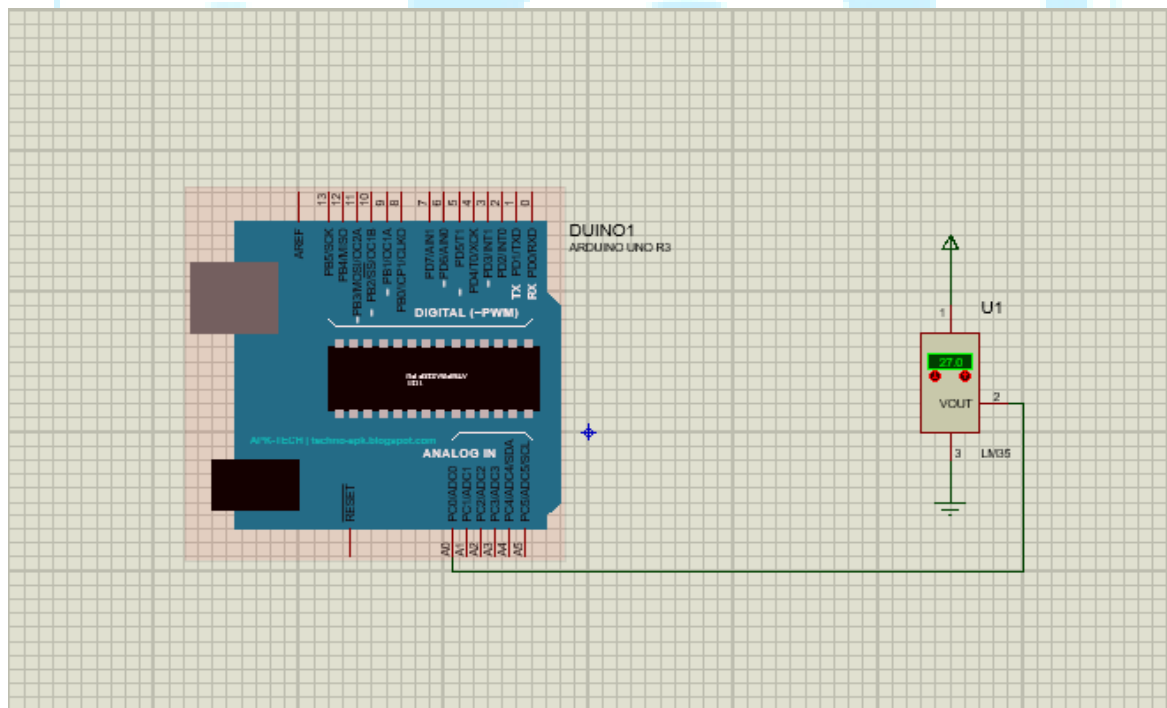
Pada bab ini diuraikan mengenai proses implementasi yang dilakukan pada Tugas Akhir ini, serta pengujian terhadap implementasi yang telah dilakukan. Proses implementasi yang dilakukan berupa perancangan pada sensor suhu **LM35**, konfigurasi pada *router* dan perancangan pada rangkaian untuk *server*. Proses pengujian yang dilakukan berupa pengujian pada sensor suhu **LM35**, pengujian *router* dan pengujian pada rangkaian *server*.

#### 4.1 Implementasi

Pada sub-bab ini menjelaskan tentang implementasi pada setiap bagian sistem yang dirancang. Proses implementasi yang dilakukan berupa perancangan pada sensor suhu **LM35**, konfigurasi untuk *router* dan perancangan pada rangkaian untuk *server*.

##### 4.1.1 Rangkaian Sensor Suhu LM35 dengan Arduino

Pada subbab ini dijelaskan mengenai rangkaian sensor suhu **LM35** dengan arduino. Rangkaian sensor suhu **LM35** dapat dilihat pada Gambar 18 dibawah ini.

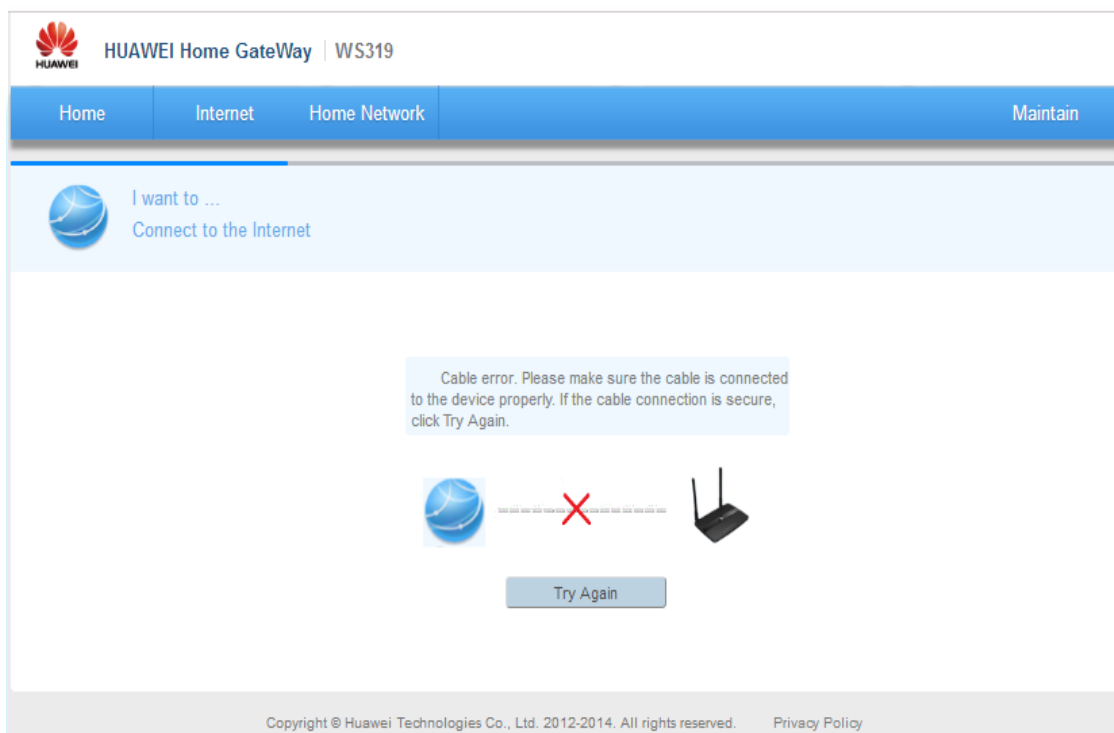


Gambar 18. Rangkaian Sensor Suhu LM35 dengan Arduino

#### 4.1.2 Konfigurasi Router

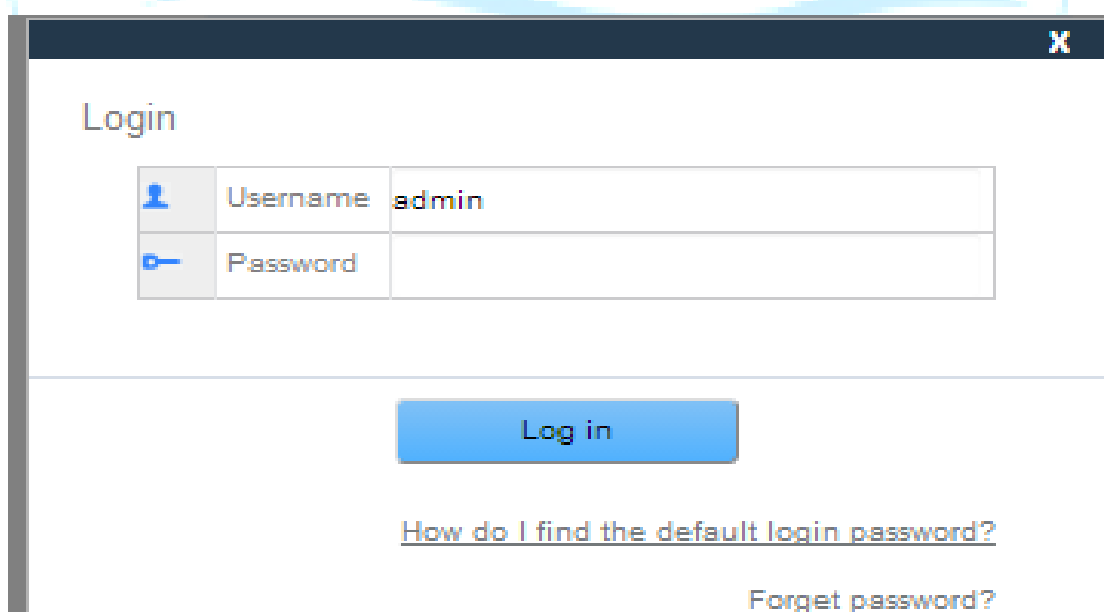
Konfigurasi pada *router* dilakukan dengan cara melakukan konfigurasi IP *address* pada LAN *Interfaces Settings* yang terdapat pada pengaturan *router*. Konfigurasi *router* dapat dilihat seperti berikut.

1. Hubungkan perangkat *router* ke laptop/PC dengan menggunakan kabel LAN. Maka selanjutnya akan tampil halaman untuk pengaturan *router* seperti pada Gambar 19.



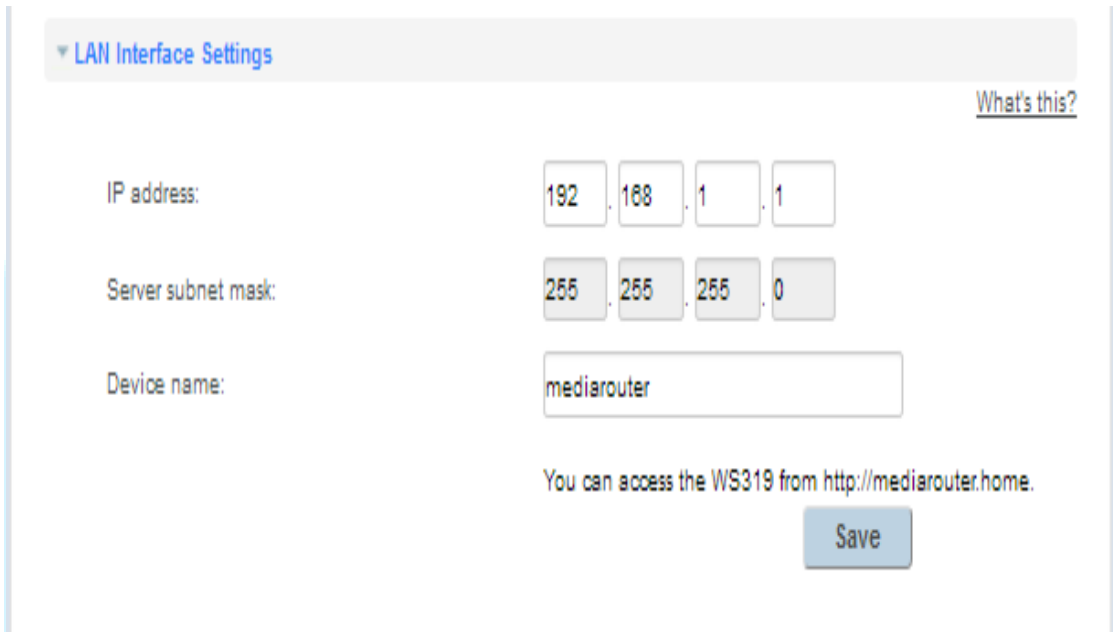
**Gambar 19. Halaman Utama Web untuk Konfigurasi Router**

2. Masukkan *password* yang telah disetting sebelumnya seperti Gambar 20.



**Gambar 20. Halaman Login**

3. Pada bagian LAN *interface* terdapat LAN *Interface Settings*. Maka pada bagian tersebut masukkan IP *address*, *subnet mask*, dan nama *device* yang akan dibuat. Lalu klik “Save”. Berikut adalah tampilannya:



LAN Interface Settings [What's this?](#)

IP address: 192 . 168 . 1 . 1

Server subnet mask: 255 . 255 . 255 . 0

Device name: mediarouter

You can access the WS319 from <http://mediarouter.home>.

Save

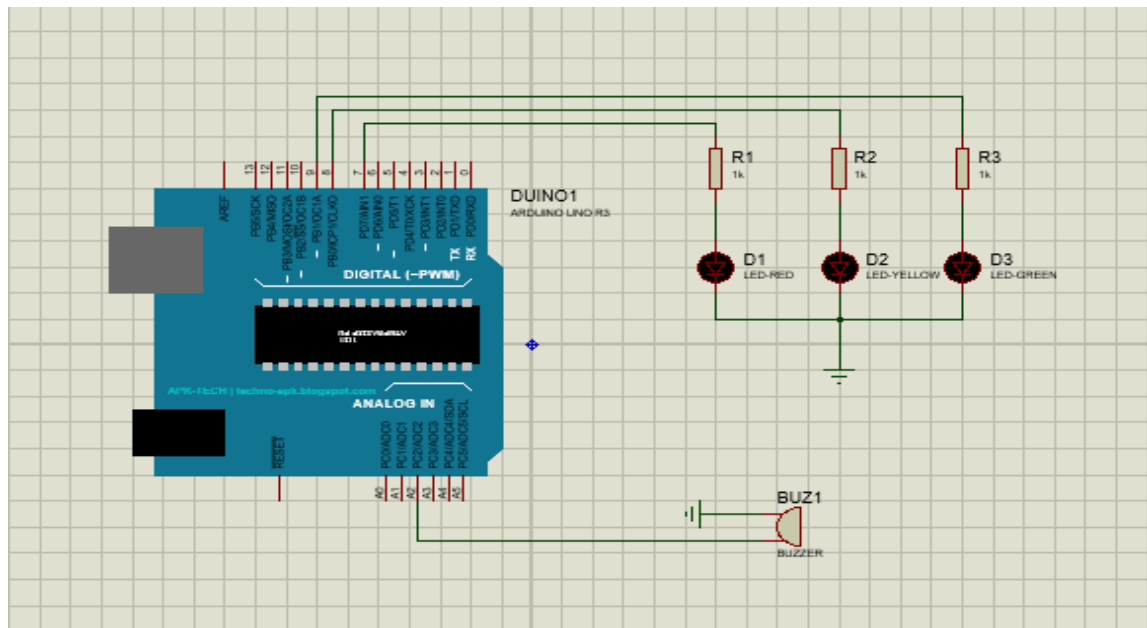
Gambar 21. LAN Interface Setting

#### 4.1.3 Konfigurasi Arduino *Ethernet Shield*

Pada subbab ini dijelaskan mengenai konfigurasi yang dilakukan dalam pembuatan rangkaian *Ethernet shield* dan **Arduino Uno**. Rangkaian *Ethernet shield* dan **Arduino Uno** dipasang dengan cara memasangkan *Ethernet shield* di atas **Arduino Uno** dan menghubungkan laptop dan *Ethernet shield* dengan menggunakan kabel RJ-45. Konfigurasi pada *Ethernet shield* dan **Arduino Uno** juga dilakukan dengan mengubah *IP adress* laptop menjadi *static*. Hal ini dilakukan agar laptop dan Arduino berada pada jaringan yang sama sehingga dapat saling berkomunikasi.

#### 4.1.4 Rangkaian LED dan *Buzzer* pada Server

Pada subbab ini dijelaskan mengenai rangkaian LED dan *buzzer* pada *server*. LED dan *buzzer* dibuat dalam satu rangkaian. Rangkaian dari LED dan *buzzer* dapat dilihat pada Gambar 22.



Gambar 22. Rangkaian LED dan Buzzer

## 4.2 Pengujian

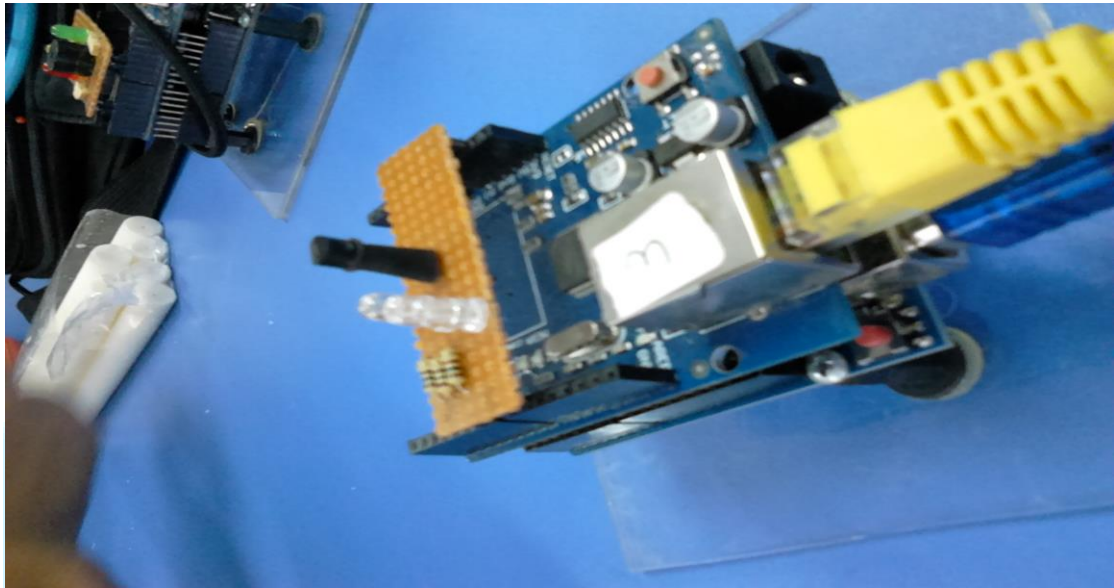
Pengujian dilakukan untuk memastikan apakah sistem peringatan dini kebakaran hutan yang dibangun dapat menjalankan tugasnya dengan baik.

### 4.2.1 Cara Pengujian Sensor Suhu LM35

Pengujian sensor suhu **LM35** dilakukan dengan cara konfigurasi sensor suhu **LM35** dengan **Arduino Uno**. Rangkaian sensor suhu **LM35** dipasang pada PCB bolong dan pada kaki pin sensor telah disolder sehingga data suhu yang dibaca oleh sensor suhu **LM35** lebih akurat. Sensor suhu **LM35** dapat membaca data dan menampilkannya di *serial monitor* pada *software* Arduino.

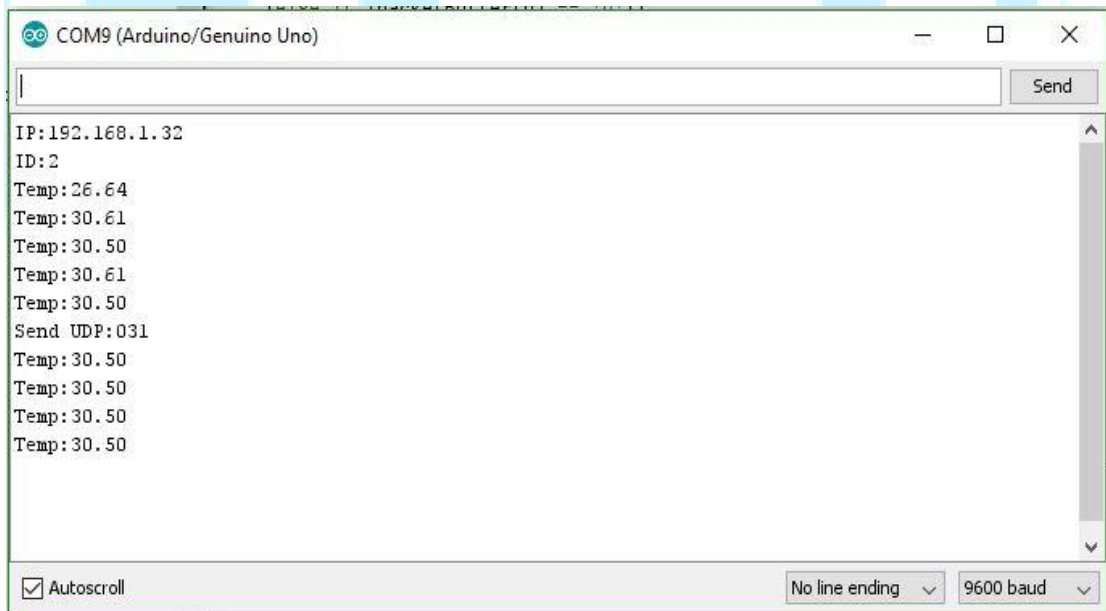
Langkah-langkah yang dilakukan adalah sebagai berikut.

1. Membuat rangkaian sensor suhu **LM35** di PCB bolong. Hasil rangkaian dapat dilihat pada Gambar 23.



**Gambar 23. Pengujian Sensor Suhu LM35 di PCB**

2. Pilih *port Arduino Uno* dan *upload* kode program sensor [1] seperti yang ada pada lampiran. Kode program tersebut di *upload* pada *software* Arduino IDE.
3. Kemudian tekan tombol CTRL+SHIFT+M untuk membuka data suhu yang telah dibaca oleh sensor suhu **LM35** pada *serial monitor*. Maka hasil dari pengujian sensor suhu **LM35** adalah seperti Gambar 24.



**Gambar 24. Hasil Kode Program untuk Sensor Pada Serial Monitor**

Berdasarkan langkah-langkah diatas dapat disimpulkan bahwa sensor dapat membaca data suhu dengan baik.

#### 4.2.2 Cara Pengujian Router

Pengujian *router* dilakukan dengan cara mem-ping IP address setiap *router* melalui *command prompt* yang ada pada *Windows*. Langkah-langkah yang dilakukan adalah sebagai berikut.

1. Sambungkan salah satu kabel LAN dari salah satu *router* ke Laptop. Misalnya dari *router* 2 dengan IP address 192.168.1.7. Kemudian buka *command prompt* melalui perintah WIN+R. Ping *router* 1 dengan IP address 192.168.1.6. Hasilnya adalah seperti berikut.

```
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\Tia Monica>ping 192.168.1.6

Pinging 192.168.1.6 with 32 bytes of data:
Reply from 192.168.1.6: bytes=32 time=20ms TTL=64
Reply from 192.168.1.6: bytes=32 time=1ms TTL=64
Reply from 192.168.1.6: bytes=32 time=1ms TTL=64
Reply from 192.168.1.6: bytes=32 time=1ms TTL=64

Ping statistics for 192.168.1.6:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 20ms, Average = 5ms

C:\Users\Tia Monica>
```

2. Kemudian ping kembali *router* 2 dengan IP address 192.168.1.7 dari *command prompt*, maka jika berhasil akan tampil seperti berikut.

```
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\Tia Monica>ping 192.168.1.7

Pinging 192.168.1.7 with 32 bytes of data:
Reply from 192.168.1.7: bytes=32 time=20ms TTL=64
Reply from 192.168.1.7: bytes=32 time=1ms TTL=64
Reply from 192.168.1.7: bytes=32 time=1ms TTL=64
Reply from 192.168.1.7: bytes=32 time=1ms TTL=64

Ping statistics for 192.168.1.7:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 20ms, Average = 5ms

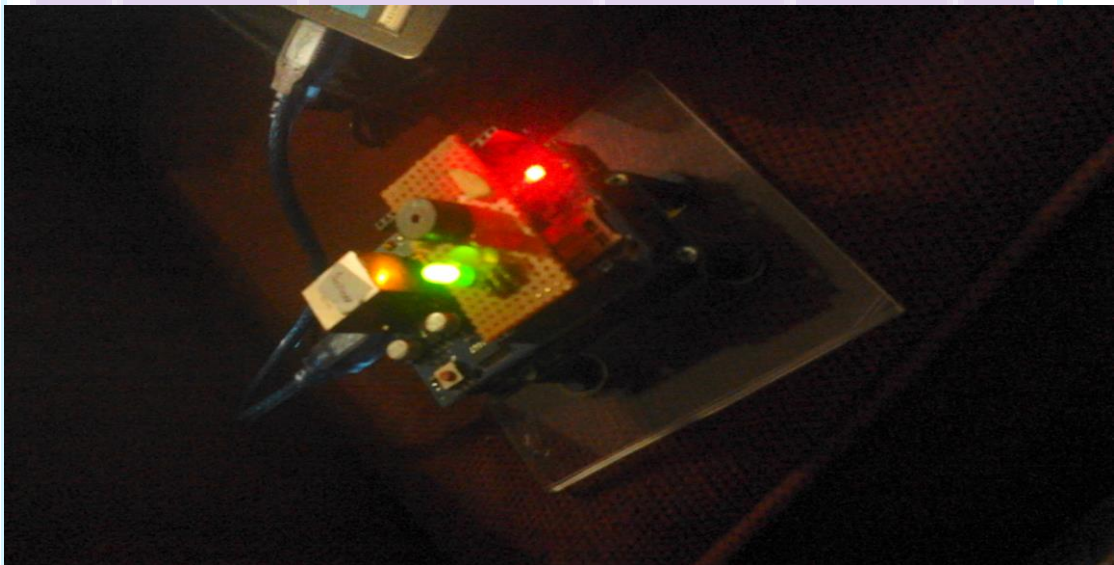
C:\Users\Tia Monica>
```

Berdasarkan langkah-langkah diatas dapat disimpulkan bahwa *router* dapat berkomunikasi satu sama lain.

#### 4.2.3 Cara Pengujian LED dan Buzzer

Pengujian LED dan *buzzer* pada *server* dilakukan dengan cara membuat rangkaian LED dan *buzzer* di PCB bolong. Pin LED dan *buzzer* dibuat saling terhubung ke **Arduino Uno** dengan cara disolder pada PCB bolong sehingga data yang dikirimkan dari lahan prototipe yang diterima oleh *server* adalah akurat. Langkah-langkah yang dilakukan untuk melakukan pengujian LED dan *buzzer* adalah sebagai berikut.

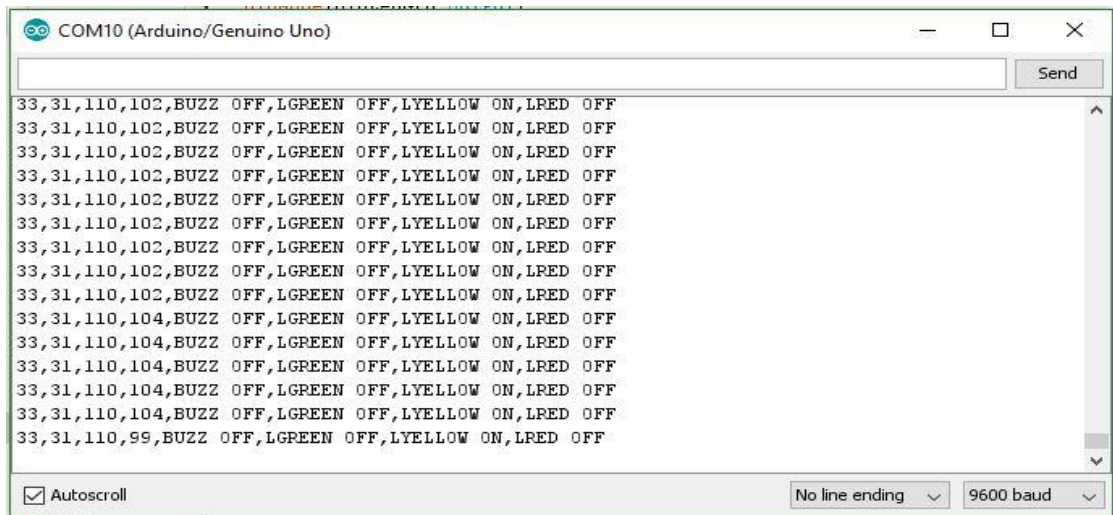
1. Membuat rangkaian LED dan *buzzer* yang berada pada *server* di PCB. Hasil rangkaian LED dan *buzzer* di PCB adalah seperti Gambar 25.



**Gambar 25. Pengujian LED dan Buzzer di PCB**

2. Pilih *port* **Arduino Uno** dan *upload* kode program *server* [2] yang ada pada lampiran pada *software* Arduino IDE. Kode program tersebut di *upload* di *software* Arduino IDE.
3. Kemudian tekan tombol CTRL+SHIFT+M untuk melihat data suhu yang telah dibaca oleh keempat sensor suhu **LM35** pada *serial monitor*. Hasil pembacaan dari keempat sensor suhu **LM35** dapat dilihat pada Gambar 26.





```
33,31,110,102,BUZZ OFF,LGREEN OFF,LYELLOW ON,LRED OFF
33,31,110,102,BUZZ OFF,LGREEN OFF,LYELLOW ON,LRED OFF
33,31,110,102,BUZZ OFF,LGREEN OFF,LYELLOW ON,LRED OFF
33,31,110,102,BUZZ OFF,LGREEN OFF,LYELLOW ON,LRED OFF
33,31,110,102,BUZZ OFF,LGREEN OFF,LYELLOW ON,LRED OFF
33,31,110,102,BUZZ OFF,LGREEN OFF,LYELLOW ON,LRED OFF
33,31,110,102,BUZZ OFF,LGREEN OFF,LYELLOW ON,LRED OFF
33,31,110,102,BUZZ OFF,LGREEN OFF,LYELLOW ON,LRED OFF
33,31,110,102,BUZZ OFF,LGREEN OFF,LYELLOW ON,LRED OFF
33,31,110,104,BUZZ OFF,LGREEN OFF,LYELLOW ON,LRED OFF
33,31,110,104,BUZZ OFF,LGREEN OFF,LYELLOW ON,LRED OFF
33,31,110,104,BUZZ OFF,LGREEN OFF,LYELLOW ON,LRED OFF
33,31,110,104,BUZZ OFF,LGREEN OFF,LYELLOW ON,LRED OFF
33,31,110,104,BUZZ OFF,LGREEN OFF,LYELLOW ON,LRED OFF
33,31,110,104,BUZZ OFF,LGREEN OFF,LYELLOW ON,LRED OFF
33,31,110,99,BUZZ OFF,LGREEN OFF,LYELLOW ON,LRED OFF
```

**Gambar 26. Hasil Kode Program Server Pada Serial Monitor**

Berdasarkan langkah-langkah diatas dapat disimpulkan bahwa LED dan *buzzer* berjalan dengan baik.



## Bab V

### Kesimpulan dan Saran

Pada bab ini dituliskan mengenai kesimpulan dari pengerjaan tugas akhir ini dan saran-saran untuk pengembangan sistem yang dibangun pada penelitian selanjutnya. Kesimpulan dan saran untuk sistem yang dibangun dijelaskan pada subbab berikut.

#### 5.1 Kesimpulan

Kesimpulan yang didapat dari hasil pengujian dan pengamatan yang dilakukan terhadap Prototipe Sistem Peringatan Dini Kebakaran Hutan Menggunakan Sensor Suhu dan Arduino adalah sebagai berikut.

1. Prototipe peringatan dini kebakaran hutan menggunakan **Arduino Uno** dan sensor suhu **LM35** berhasil dilakukan dengan menggunakan *router* sebagai transmisi datanya.
2. Sensor suhu **LM35** dapat membaca data suhu sesuai dengan syarat-syarat dikatakannya sebagai *event* akan terjadinya kebakaran dengan menggunakan metode *fuzzy logic* Tagaki Sugino yaitu apabila minimal 2 sensor mendeteksi suhu  $35^{\circ}\text{C} < x < 75^{\circ}\text{C}$  dalam waktu 4 menit maka dinyatakan sebagai *warning* dan led kuning pada *server* akan menyala dan apabila minimal 2 sensor mendeteksi suhu  $> 75^{\circ}\text{C}$  maka dinyatakan sebagai kebakaran dan led merah pada *server* akan menyala dan *buzzer* akan mengeluarkan bunyi.
3. Sistem peringatan dini kebakaran hutan yang dibangun mampu menampilkan data yang dikirim ke *server* pada *serial monitor software* Arduino IDE.

#### 5.2 Saran

Saran yang dapat diberikan yaitu untuk pengembangan selanjutnya dapat menggunakan modul *wifi* seperti *Zigbee* dan ESP8266 dan membuat aplikasi monitoring berupa *user interface* pada *server*. Akan tetapi, sisi ekonominya harus diperhatikan. Hal ini dikarenakan biaya yang dikeluarkan untuk pembelian modul *wifi* dikategorikan mahal.

## Daftar Pustaka

- [1] Lapanporo, Boni Pahlanop, 2011. "Prototype Sistem Telemetri Berbasis Sensor Suhu dan Sensor Asap untuk Pemantau Kebakaran Lahan". POSITRON. Volume 1, No.2.
- [2] Bagaskara, G., Amri, R. Oktober 2016. "Rancang Bangun Sistem Pendeteksi Kebakaran Lahan Gambut Jenis Kayuan Dengan Memanfaatkan Karakteristik Panas Yang Ditimbulkannya". Jom FTEKNIK. Volume 3, No.2.
- [3] Octavianes, Ferly., dkk. 2012. "Pengukur Kelembaban Tanah dan Suhu Udara sebagai Pendeteksi Dini Kebakaran Hutan Melalui Wireless Sensor Network (WSN) Software".
- [4] Usuman, Ilona., Aardhi, Hasmi. 2010. "Sistem Pendeteksi Suhu dan Asap pada Ruangan Tertutup Memanfaatkan Sensor LM35 dan Sensor AF30". Berkala Fisika. Volume 13, No.2.
- [5] Osman, Smile. 2014. "Jaringan LAN". Tersedia: <http://smileosman.blogspot.co.id/2014/11/makalah-lan.html> [3 November 2014]
- [6] Gerald, Fraiza. 2014. "Media Transmisi pada Jaringan Komputer". Tersedia: <https://fraizageraldi97.blogspot.co.id/2013/10/media-transmisi.html> [23 Mei 2014]
- [7] Utomo, Wiyugo. 2011. "Jaringan Komputer". Tersedia: <http://yugo16.blogspot.co.id/2011/06/jaringan-komputer.html> [13 Juni 2011]
- [8] Stallings, William. 2007. "Data and Computer Communications Eighth Edition". Pearson Education, Inc.
- [9] 3XP'S. 2012. "Makalah Kecerdasan Buatan Fuzzy Logic". Tersedia: <http://3xpdata.blogspot.co.id/2012/04/makalah-kecerdasan-buatan-fuzzy-logic.html> [16 April 2012]
- [10] Muhlis, Nur. 2011. "Logika Fuzzy". Tersedia: <https://nurmuhlis.wordpress.com/2011/03/01/logika-fuzzy/> [1 Maret 2011]
- [11] Prehan, Bagus. 2013. "Sensor Suhu LM35". Tersedia: <http://www.bagusprehan.com/2013/12/sensor-suhu-lm35.html> [26 Desember 2013]
- [12] Marian, P. 2012. "LM35 Datasheet". Tersedia: <http://www.electroschematics.com/6393/lm35-datasheet/> [5 Juli 2012]
- [13] Husni. 2017. "LM35". Tersedia: <http://kl801.ilearning.me/2015/05/27/lm35-3/> [27 Mei 2017]

- [14]Angga, Rida. 2017. “Pengertian Led, Cara Kerja Dan Pemanfaatannya”. Tersedia:  
<http://skemaku.com/pengertian-led-cara-kerja-dan-pemanfaatannya/>
- [15]Asrofi. 2015. “Cara Menghitung Jarak Tanam”. Tersedia:  
<http://www.kampustani.com/cara-menghitung-jarak-tanam/> [30 Maret 2015]



## Lampiran

### Kode program untuk sensor suhu [1]

```
#include <SPI.h>
#include <Ethernet.h>
#include <EthernetUdp.h>
// =====
const int ID_CLIENT = 3; // 1,2,3,4 -> ubah sesuai id client
//=====
byte mac[] = { 0xDE, 0xAD + ID_CLIENT, 0xBC + ID_CLIENT , 0xAF +
ID_CLIENT , 0x5E + ID_CLIENT, 0xCD };
IPAddress ip(192, 168, 1, 30 + ID_CLIENT);
IPAddress msterIp(192, 168, 1, 30);
const char strID[4] = {'A','B','C','D'};
unsigned int localPort = 5000;

const int intervalKirim = 5; // 120 detik = 2 menit

const int pinLedRed = 7;
const int pinLedYel = 8;
const int pinLedGrn = 9;
const int HIJAU = 1;
const int KUNING = 2;
const int MERAH = 3;

long dlyKirim;
int cntKirim;
int reading;
int pinLM35 = A0;
float sendTemp;
String dt;
char *dtChr;
long timerRed;
long timerYel;
int idx;
EthernetUDP Udp;

void setup() {
  analogReference(INTERNAL);
  Ethernet.begin(mac, ip);
  Udp.begin(localPort);

  Serial.begin(9600);
  Serial.print("IP:");
  Serial.println(ip);

  pinMode(pinLedRed, OUTPUT);
  pinMode(pinLedYel, OUTPUT);
  pinMode(pinLedGrn, OUTPUT);
```

```

    Serial.print("ID:");
    Serial.println(ID_CLIENT);
    setLed(HIJAU); delay(300);
    setLed(KUNING); delay(300);
    setLed(MERAH); delay(300);
    setLed(0); delay(300);
}

void loop() {

    if(millis() > dlyKirim + 1000){
        dlyKirim = millis();
        // menyalakan led
        sendTemp = GetTemperature();
        Serial.print("Temp:");
        Serial.println(sendTemp);
        if(sendTemp <= 35){ // jika suhu di bawah 35 derajat maka normal
            setLed(HIJAU);
            timerYel = 0;
            timerRed = 0;
            idx = 1;

            }else if(sendTemp <= 75){ // jika suhu mencapai 35-75 derajat
                timerYel++;
                timerRed = 0;

                if(timerYel > 5){ //240; tunggu hingga 4 menit (60 detik x 4
menit =240 detik)
                    setLed(KUNING);
                    idx = 2;
                }else{
                    if(idx == 1 ) setLed(HIJAU);
                    if(idx == 3 ) setLed(MERAH);
                }
            }else{
                timerRed++;
                timerYel = 0;

                if(timerRed > 7){ // 60 detik x 7 menit =420 detik
                    setLed(MERAH);
                    idx = 3;
                }else{
                    setLed(KUNING);
                }
            }
        }

        // kirim ke server
        cntKirim++;
        if(cntKirim >= intervalKirim) {
            cntKirim = 0;
            //sendTemp = GetTemperature();

            dt = String(sendTemp,0);
            if(dt.length() == 1) {
                dt = "00" + dt;
            }else if(dt.length() == 2) {
                dt = "0" + dt;
            }
        }
    }
}

```

```

//Serial.println(dt);
    char dtSend[10];
    dt.toCharArray(dtSend,10);
    //Serial.println(dtSend);
    Udp.beginPacket(msterIp, localPort);
    Udp.write(strID[ID_CLIENT-1]);
    Udp.write(dtSend);
    Udp.endPacket();
    delay(100);
    Serial.print("Send UDP:");
    Serial.println(dtSend);
}
}
delay(10);
}

float GetTemperature()
{
    float tempC;
    reading = analogRead(pinLM35);
    tempC = reading / 9.31;
    return tempC;
}

void setLed(int cod){
    switch(cod){
        case 0: // semua led mati
            digitalWrite(pinLedGrn,LOW);
            digitalWrite(pinLedYel,LOW);
            digitalWrite(pinLedRed,LOW);
            break;

        case HIJAU: // led hijau nyala
            digitalWrite(pinLedGrn,HIGH);
            digitalWrite(pinLedYel,LOW);
            digitalWrite(pinLedRed,LOW);
            break;

        case KUNING: // led kuning nyala
            digitalWrite(pinLedGrn,LOW);
            digitalWrite(pinLedYel,HIGH);
            digitalWrite(pinLedRed,LOW);
            break;

        case MERAH: // led merah nyala
            digitalWrite(pinLedGrn,LOW);
            digitalWrite(pinLedYel,LOW);
            digitalWrite(pinLedRed,HIGH);
            break;
    }
}
}

```

## Kode program untuk *server* [2]

```
#include <SPI.h>
#include <Ethernet.h>
#include <EthernetUdp.h>
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
IPAddress ip(192, 168, 1, 30);

String inputString = "";
boolean stringComplete = false;

unsigned int localPort = 5000;

char packetBuffer[UDP_TX_PACKET_MAX_SIZE];
char ReplyBuffer[] = "acknowledged";

const int pinBuzz = A2;
const int PinLed [3] = {9,A0,A1};
const int pinLedRed = A1;
const int pinLedYel = A0;
const int pinLedGrn = 9;

long timertBuzz;
bool buzzOnOff;
int ledMode; // 0 : mati semua, 1: Hijau, 2:Yellow, 3:red
bool ledOnOff;
int statusLed[4];
int dataSuhu[4] = {0,0,0,0};
int tmrDiscon[4];
int dlyAktif[4];
long tmrDelay;
long myTime1;
long myTime2;

bool alert[4];

long timerSend;
EthernetUDP Udp;

void setup() {
  Ethernet.begin(mac, ip);
  Udp.begin(localPort);
  pinMode(pinBuzz,OUTPUT);
  pinMode(pinLedRed,OUTPUT);
  pinMode(pinLedYel,OUTPUT);
  pinMode(pinLedGrn,OUTPUT);
  Serial.begin(9600);
}
void loop() {
  if (stringComplete) {
    Serial.print("Ser IN:");
    Serial.println(inputString);
    char dtString[5];
    String dtUdp = inputString;
    String dtSuhu;
    inputString.toCharArray(dtString,5);
```

```

if (dtString[0] == 'A'){
    tmrDiscon[0] = 0;
    //Serial.print("Dev1 Suhu:");
    dtSuhu = dtUdp.substring(1,dtUdp.length());
    dataSuhu[0] = dtSuhu.toInt();
    //Serial.println(dtSuhu);
}else if (dtString[0] == 'B'){
    tmrDiscon[1] = 0;
    //Serial.print("Dev2 Suhu:");
    dtSuhu = dtUdp.substring(1,dtUdp.length());
    dataSuhu[1] = dtSuhu.toInt();
    //Serial.println(dtSuhu);
}else if (dtString[0] == 'C'){
    tmrDiscon[2] = 0;
    //Serial.print("Dev3 Suhu:");
    dtSuhu = dtUdp.substring(1,dtUdp.length());
    dataSuhu[2] = dtSuhu.toInt();
    //Serial.println(dtSuhu);
}else if (dtString[0] == 'D'){
    tmrDiscon[3] = 0;
    //Serial.print("Dev4 Suhu:");
    dtSuhu = dtUdp.substring(1,dtUdp.length());
    dataSuhu[3] = dtSuhu.toInt();
    //Serial.println(dtSuhu);
}
inputString = "";
stringComplete = false;
}

// baca paket data dari jaringan
int packetSize = Udp.parsePacket();
if (packetSize) {
    Udp.read(packetBuffer, UDP_TX_PACKET_MAX_SIZE);
    String dtUdp = packetBuffer;
    String dtSuhu;
    //Serial.print("IN UDP:");
    //Serial.println(packetBuffer);
    dtUdp.trim();
    if (packetBuffer[0] == 'A'){
        tmrDiscon[0] = 0;
        //Serial.print("Dev1 Suhu:");
        dtSuhu = dtUdp.substring(1,dtUdp.length());
        dataSuhu[0] = dtSuhu.toInt();
        //Serial.println(dtSuhu);
    }else if (packetBuffer[0] == 'B'){
        tmrDiscon[1] = 0;
        //Serial.print("Dev2 Suhu:");
        dtSuhu = dtUdp.substring(1,dtUdp.length());
        dataSuhu[1] = dtSuhu.toInt();
        //Serial.println(dtSuhu);
    }else if (packetBuffer[0] == 'C'){
        tmrDiscon[2] = 0;
        //Serial.print("Dev3 Suhu:");
        dtSuhu = dtUdp.substring(1,dtUdp.length());
        dataSuhu[2] = dtSuhu.toInt();
        //Serial.println(dtSuhu);
    }
}

```



```

}else if (packetBuffer[0] == 'D'){
    tmrDiscon[3] = 0;
    //Serial.print("Dev4 Suhu:");
    dtSuhu = dtUdp.substring(1,dtUdp.length());
    dataSuhu[3] = dtSuhu.toInt();
    //Serial.println(dtSuhu);
}

}

// kirim data ke komputer
// format data = SUHU1,SUHU2,SUHU3,SUHU4,BUZZ,LED GREEN,LED
YELLOW, LED RED
if(millis() > timerSend + 1000){ // 1000ms = 1 detik
    timerSend = millis();
    for(int i = 0; i <= 3;i++){
        tmrDiscon[i]++; // hitung timer dosconnect
        if(tmrDiscon[i] > 30) dataSuhu[i] =0; // jika dalam 30 detik
gag ada data mk set 0, anggap diskonek
        Serial.print(dataSuhu[i]);
        Serial.print(",");
        if (dataSuhu[i] <= 35) {
            statusLed[i]=0;
        }else if (dataSuhu[i] <= 75) {
            statusLed[i]=1;
        }else { //if (dataSuhu[i] <= 150) {
            statusLed[i]=2;
        }
    }

    if(buzzOnOff) Serial.print("BUZZ ON,"); else Serial.print("BUZZ
OFF,");
    if(digitalRead(pinLedGrn) == 1 ) Serial.print("LGREEN ON,");
else Serial.print("LGREEN OFF,");
    if(digitalRead(pinLedYel) == 1 ) Serial.print("LYELLOW ON,");
else Serial.print("LYELLOW OFF,");
    if(digitalRead(pinLedRed) == 1 ) Serial.println("LRED ON"); else
Serial.println("LRED OFF");
}

if(millis() > tmrDelay + 500){
    tmrDelay = millis();
    digitalWrite(pinLedGrn,HIGH);
    digitalWrite(pinLedRed,LOW);
    digitalWrite(pinLedYel,LOW);
    buzzOnOff = false;
    //Serial.println(myTime);
    if(statusLed[0] >= 1 && statusLed[1] >= 1){ // jika sensor
1,2 antara 35-75 derajat
        myTime1++;
    }else if(statusLed[1] >= 1 && statusLed[2] >= 1){ // jika
sensor 2,3 antara 35-75 derajat
        myTime1++;
    }else if(statusLed[2] >= 1 && statusLed[3] >= 1){ // jika sensor 3,4
antara 35-75 derajat
        myTime1++;
    }else {
        myTime1 = 0;
    }
}

```

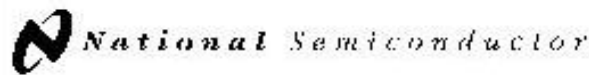
```

if(myTime1 > 10){ //480){ 480 --> 0.5 detik x 480 = 240 detik = 4
menit
    if(myTime2 < 20){ // batalkan kalo led merah aktif
        digitalWrite(pinLedYel,!digitalRead(pinLedYel));
        digitalWrite(pinLedRed,LOW);
        digitalWrite(pinLedGrn,LOW);
    }
}

    if(statusLed[0] == 2 && statusLed[1] == 2){ // jika sensor
1,2 antara > 75 derajat
        myTime2++;
    }else if(statusLed[1] == 2 && statusLed[2] == 2){ // jika
sensor 2,3 antara > 75 derajat
        myTime2++;
    }else if(statusLed[2] == 2 && statusLed[3] == 2){ // jika
sensor 3,4 antara > 75 derajat
        myTime2++;
    }else {
        myTime2 = 0;
    }

    if(myTime2 > 20){ //480){ 480 --> 0.5 detik x 480 = 240 detik
= 4 menit
        digitalWrite(pinLedRed,!digitalRead(pinLedRed));
        digitalWrite(pinLedGrn,LOW);
        digitalWrite(pinLedYel,LOW);
        buzzOnOff = true;
    }
}
// Buzzer ON - OFF
if(buzzOnOff){
    if(millis() > timertBuzz + 200){
        timertBuzz = millis();
        digitalWrite(pinBuzz,!digitalRead(pinBuzz));
    }
}else{
    digitalWrite(pinBuzz,LOW);
}
}

```



November 2000

## LM35 Precision Centigrade Temperature Sensors

### General Description

The LM35 series are precision integrated-circuit temperature sensors, whose output voltage is linearly proportional to the Celsius (Centigrade) temperature. The LM35 thus has an advantage over linear temperature sensors calibrated in ° Kelvin, as the user is not required to subtract a large constant voltage from its output to obtain convenient Centigrade scaling. The LM35 does not require any external calibration or trimming to provide typical accuracies of  $\pm 1/4^\circ\text{C}$  at room temperature and  $\pm 3/4^\circ\text{C}$  over a full  $-55^\circ$  to  $+150^\circ\text{C}$  temperature range. Low cost is assured by trimming and calibration at the wafer level. The LM35's low output impedance, linear output, and precise inherent calibration make interfacing to readout or control circuitry especially easy. It can be used with single power supplies, or with plus and minus supplies. As it draws only  $60\text{ }\mu\text{A}$  from its supply, it has very low self-heating, less than  $0.1^\circ\text{C}$  in still air. The LM35 is rated to operate over a  $-55^\circ$  to  $+150^\circ\text{C}$  temperature range, while the LM35C is rated for a  $-40^\circ$  to  $+110^\circ\text{C}$  range ( $-10^\circ$  with improved accuracy). The LM35 series is available pack-

aged in hermetic TO-48 transistor packages, while the LM35C, LM35CA, and LM35D are also available in the plastic TO-92 transistor package. The LM35D is also available in an 8-lead surface mount small outline package and a plastic TO-220 package.

### Features

- Calibrated directly in ° Celsius (Centigrade)
- Linear  $+10.0\text{ mV}/^\circ\text{C}$  scale factor
- $0.5^\circ\text{C}$  accuracy guaranteeable (at  $+25^\circ\text{C}$ )
- Rated for full  $-55^\circ$  to  $+150^\circ\text{C}$  range
- Suitable for remote applications
- Low cost due to wafer-level trimming
- Operates from 4 to 30 volts
- Less than  $60\text{ }\mu\text{A}$  current drain
- Low self-heating,  $0.08^\circ\text{C}$  in still air
- Nonlinearity only  $\pm 1/4^\circ\text{C}$  typical
- Low impedance output,  $0.1\text{ }\Omega$  for  $1\text{ mA}$  load

### Typical Applications

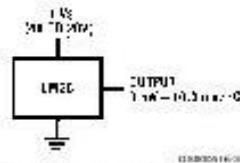
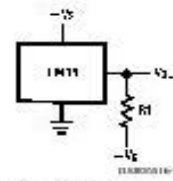


FIGURE 1. Basic Centigrade Temperature Sensor  
( $+2^\circ\text{C}$  to  $+150^\circ\text{C}$ )



Choose  $R_1 = -V_S/50\text{ }\mu\text{A}$   
 $V_{\text{OUT}} = +1,500\text{ mV at } +150^\circ\text{C}$   
 $= +250\text{ mV at } +25^\circ\text{C}$   
 $= -550\text{ mV at } -55^\circ\text{C}$

FIGURE 2. Full-Range Centigrade Temperature Sensor

**Absolute Maximum Ratings** (Note 10)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage	+35V to -0.2V
Output Voltage	+6V to -1.0V
Output Current	10 mA
Storage Temp.:	
TO-46 Package,	-60°C to +180°C
TO-92 Package,	-60°C to +150°C
SO-8 Package,	-65°C to +150°C
TO-220 Package,	-65°C to +150°C
Lead Temp.:	
TO-46 Package,	
(Soldering, 10 seconds)	300°C

TO-92 and TO-220 Package,	
(Soldering, 10 seconds)	260°C
SO Package (Note 12)	
Vapor Phase (60 seconds)	215°C
Infrared (15 seconds)	220°C
ESD Susceptibility (Note 11)	2500V
Specified Operating Temperature Range: $T_{MIN}$ to $T_{MAX}$	
(Note 2)	
LM35, LM35A	-55°C to +150°C
LM35C, LM35CA	-40°C to +110°C
LM35D	0°C to +100°C

**Electrical Characteristics**

(Notes 1, 6)

Parameter	Conditions	LM35A			LM35CA			Units (Max.)
		Typical	Tested Limit (Note 4)	Design Limit (Note 5)	Typical	Tested Limit (Note 4)	Design Limit (Note 5)	
Accuracy (Note 7)	$T_A = +25^\circ\text{C}$	$\pm 0.2$	$\pm 0.5$		$\pm 0.2$	$\pm 0.5$	$\pm 1.0$	$^\circ\text{C}$
	$T_A = -10^\circ\text{C}$	$\pm 0.3$			$\pm 0.3$		$\pm 1.0$	$^\circ\text{C}$
	$T_A = T_{MAX}$	$\pm 0.4$	$\pm 1.0$		$\pm 0.4$	$\pm 1.0$	$\pm 1.5$	$^\circ\text{C}$
	$T_A = T_{MIN}$	$\pm 0.4$	$\pm 1.0$		$\pm 0.4$		$\pm 1.5$	$^\circ\text{C}$
Nonlinearity (Note 8)	$T_{MIN} \leq T_A \leq T_{MAX}$	$\pm 0.18$		$\pm 0.35$	$\pm 0.15$		$\pm 0.3$	$^\circ\text{C}$
Sensor Gain (Average Slope)	$T_{MIN} \leq T_A \leq T_{MAX}$	$\pm 10.0$	$\pm 9.9$ , $\pm 10.1$		$\pm 10.0$		$\pm 9.9$ , $\pm 10.1$	mV/ $^\circ\text{C}$
Load Regulation (Note 3) $0 \leq I_L \leq 1 \text{ mA}$	$T_A = +25^\circ\text{C}$	$\pm 0.4$	$\pm 1.0$		$\pm 0.4$	$\pm 1.0$	$\pm 3.0$	mV/mA
	$T_{MIN} \leq T_A \leq T_{MAX}$	$\pm 0.5$		$\pm 3.0$	$\pm 0.5$		$\pm 3.0$	mV/mA
Line Regulation (Note 3)	$T_A = +25^\circ\text{C}$	$\pm 0.01$	$\pm 0.05$		$\pm 0.01$	$\pm 0.05$	$\pm 0.1$	mV/V
	$4V \leq V_{DS} \leq 30V$	$\pm 0.02$		$\pm 0.1$	$\pm 0.02$		$\pm 0.1$	mV/V
Quiescent Current (Note 9)	$V_{DS} = +5V$ , $+25^\circ\text{C}$	56	67		56	67		$\mu\text{A}$
	$V_{DS} = +5V$	105		131	91		114	$\mu\text{A}$
	$V_{DS} = +30V$ , $+25^\circ\text{C}$	56.2	68		56.2	68		$\mu\text{A}$
	$V_{DS} = +30V$	105.5		133	91.5		116	$\mu\text{A}$
Change of Quiescent Current (Note 3)	$4V \leq V_{DS} \leq 30V$ , $+25^\circ\text{C}$	0.2	1.0		0.2	1.0		$\mu\text{A}$
	$4V \leq V_{DS} \leq 30V$	0.5		2.0	0.5		2.0	$\mu\text{A}$
Temperature Coefficient of Quiescent Current		$\pm 0.39$		$\pm 0.5$	$\pm 0.39$		$\pm 0.5$	$\mu\text{A}/^\circ\text{C}$
Minimum Temperature for Rated Accuracy	In circuit of Figure 1, $I_L = 0$	$\pm 1.5$		$\pm 2.0$	$\pm 1.5$		$\pm 2.0$	$^\circ\text{C}$
Long Term Stability	$T_A = T_{MAX}$ , for 1000 hours	$\pm 0.08$			$\pm 0.08$			$^\circ\text{C}$

## Datasheet Arduino Uno [4]

### ARDUINO UNO Revision 3 Specifications



- Microcontroller: ATmega328
- Operating Voltage: 5V
- Uno Board Recommended Input Voltage: 7 – 12 V
- Uno Board Input Voltage Limits: 6 – 20 V
- Digital I/O Pins: 14 total – 6 of which can be PWM
- Analog Input Pins: 6
- Maximum DC Current per I/O pin at 5VDC: 40ma
- Maximum DC Current per I/O pin at 3.3 VDC: 50ma
- Flash Memory: 32KB (0.5KB used by bootloader)
- SRAM Memory: 2KB
- EEPROM: 1KB
- Clock Speed: 16 MHz

#### Links:

- Arduino specifications and image page: <http://arduino.cc/en/Main/arduinoBoardUno#UxOOLk2YZuH>

### ARDUINO UNO Revision 3 Processor Peripherals (Atmel ATmega 328)



- Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
- One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
- Real Time Counter with Separate Oscillator
- Six PWM channels
- Six channel 10 bit ADC including temperature measurement
- Programmable Serial USART
- Master/Slave SPI Serial Interface
- Byte-oriented 2 wire Serial Interface (Philips I2C compatible)
- Programmable Watchdog Timer with Separate On-chip Oscillator
- On-chip Analog Comparator

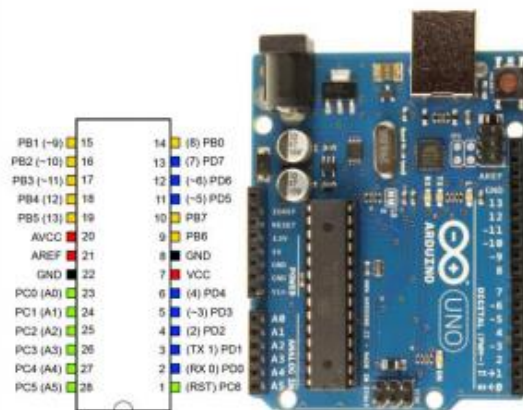
#### Links:

- Source of above diagram: <http://tekkpinov.com/wp-content/uploads/2013/10/1.jpg>
- AT Mega 328 datasheet: <http://www.atmel.com/Images/doc8161.pdf>

MARTUHAN-MARROHA-MARBISUK  
2001



## ARDUINO UNO Revision 3 and ATmega328 processor



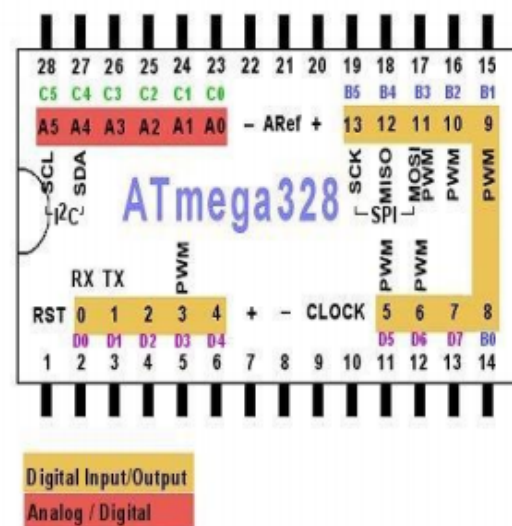
The Arduino board makes it very easy to use the ATmega328 processor by providing easy access to most of the pins via the headers. In addition, it provides:

- 5 VDC regulated power from the 6 – 20 VDC input jack
- 3.3 VDC regulated power available for other electronics
- The crystal oscillator
- A reset switch
- USB access to the serial port
- Headers for connection and for shields

### Links:

- Arduino specifications and image page: <http://arduino.cc/en/Main/arduinoBoardUno#UxOOLk2YzuH>
- ATmega328 processor image modified from image found at: <http://www.protostack.com/microcontrollers/atmega328-pu-atmel-8-bit-32k-avr-microcontroller>

## ARDUINO UNO Revision 3 Processor Pinout (Atmel ATmega 328) – Commonly Used



### Pin Definition

- PORT B (PB0 – PB7) is an 8 bit bidirectional I/O port with internal pull-ups. Processor pins 14 – 17 bring PB0 to PB5 out
  - PB0 – PB5 are also interrupts 0-5 respectively
  - PB1 can also be used as a PWM output
  - PB2 can also be SPI Bus Master Slave Select (\*SS) or PWM output
  - PB3 can also be or SPI Bus Master Out/Slave In (MOSI) or PWM output
  - PB4 can also be SPI Bus Master In/Slave Out (MISO)
  - PB5 can also be SPI Bus Master Clock Input (SCK)
  - PB6 and PB7 are brought out on Processor pins 9 and 10 for the crystal clock oscillator
- PORT C (PC0 – PC5) is a 7 bit bidirectional I/O port with internal pull-up resistors. Processor pins 23 – 28 bring PC0 to PC5 out.
  - PC0 – PC5 are also interrupts 8-13 respectively
  - PC0 – PC5 can also be used as A/D inputs
  - PC4 and PC5 can also be used as SDA and SCL for I2C
  - PC6 is brought out on processor pin 1 as reset
- PORT D (D0 – D7) is an 8 bit bidirectional I/O port with internal pull-ups. Processor pins 2 – 6 and 11 – 13 bring all pins out
  - PD0 can also be USART Input (RXD)
  - PD1 can also be USART Output (TXD)
  - PD3 can also be used as a PWM output
  - PD5 can also be used as a PWM output
  - PD6 can also be used as a PWM output

### Links:

- Source of above diagram: <http://www.hobbytronics.co.uk/arduino-atmega328-pinout>
- AT Mega 328 datasheet: <http://www.atmel.com/Images/doc8161.pdf>