TUGAS AKHIR – EE 184801

RANCANG BANGUN SMART HOME SYSTEM BERBASIS INTERNET OF THINGS

Muhammad Reza Rivandana
NRP 07111640000117

Dosen Pembimbing
Sri Rahayu, S.T., M.Kom

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2020

TUGAS AKHIR – EE 184801

RANCANG BANGUN SMART HOME SYSTEM BERBASIS INTERNET OF THINGS

Muhammad Reza Rivandana
NRP 07111640000117

Dosen Pembimbing
Sri Rahayu, S.T., M.Kom.

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2020

FINAL PROJECT – EE 184801

Design a Smart Home System based Internet of Things

Muhammad Reza Rivandana
NRP 07111640000117

Dosen Pembimbing
Sri Rahayu, S.T., M.Kom.

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2020

PERNYATAAN KEASLIAN
TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul "**Rancang Bangun *Smart Home System* Berbasis *Internet of Things***" adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

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

Surabaya, Juni 2019

Muhammad Reza Rivandana
NRP. 07111640000117

[Halaman Ini Sengaja Dikosongkan]

RANCANG BANGUN SMART HOME SYSTEM BERBASIS INTERNET OF THINGS

TUGAS AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik

Pada

Bidang Studi Telekomunikasi Multimedia
Departemen Teknik Elektro
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember

Menyetujui:

Dosen Pembimbing

Sri Rahayu, S.T., M.Kom.
NIP. 196802281997022001

SURABAYA
JULI, 2020

[Halaman Ini Sengaja Dikosongkan]

Rancang Bangun *Smart Home System* Berbasis *Internet of Things*

Nama : Muhammad Reza Rivandana
NRP : 07111640000117
Pembimbing I : Sri Rahayu, S.T., M.Kom

ABSTRAK

Dalam Tugas Akhir ini akan dibuat contoh implementasi *IoT* dalam pengelolaan jaringan listrik rumah tangga dalam bentuk sebuah prototipe *Smart Home System* (SHS), agar dapat dipantau dan dikendalikan dari jarak jauh lewat jaringan internet. Sistem SHS tersebut diadaptasi dari rumah berukuran $6\times9m^2$ yang terbagi dalam 6 area yang berbeda ,lingkup pengelolaan meliputi penggunaan lampu kipas angin ddan alat masak yang dapat di*monitor* maupun dikendalikan lewat jaringan internet. Lingkup pengelolaan meliputi penggunaan lampu, kipas angin, dan alat masak.

Adapun untuk merealisasikan hal tersebut digunakan sensor cahaya, sensor gas, sensor suhu dan beberapa relay. Selanjutnya perangkat-perangkat tersebut dihubungkan ke NodeMCU ESP8266 sebagai mikrokontroller yang *embedded* dengan modul IoT. Semua data dikirim ke *database/server* via internet, untuk diproses lebih lanjut menjadi tampilan *visual* yang bisa diakses lewat aplikasi *user interface*. Untuk memenuhi kebutuhan tersebut digunakan perangkat lunak (*cloud platform*) Blynk *platform.* Adapun pengujian Fungsional yang meliputi uji akses, uji *monitoring*, dan uji *controlling,* Dan Performnansi QoS (*throughput, jitter, delay* dan *packet loss*) s

Dari hasil fungsinonal SHS alat berfungsi dengan baik dan untuk hasil *Performnansi*(QoS) yang di klasifikasi dengan standar ETSI TIPHON diperoleh delay 1.982667ms, jitter 0.257 ms, 0 Packet loss, Throughput 1.74 bit/milisec yang masuk dalam kategori sangat bagus.

Kata kunci : *Internet of Things, Monitoring, Controlling, Modul IoT, Smart Home System, cloud platform,* Blynk *platform,* QoS, Uji fungsional, uji *controlling throughput,jitter,delay* dan *packet loss.*

[Halaman Ini Sengaja Dikosongkan]

ABSTRACT
DESIGN A SMART HOME SYSTEM BASED INTERNET OF THINGS

Name : Muhammad Reza Rivandana
NRP : 07111640000117
Supervisor I : Sri Rahayu, S.T., M.Kom

ABSTRACT

In this Final Project an example of an IoT implementation will be made in the management of household electricity networks in the form of a prototype Smart Home System (SHS), so that it can be monitored and controlled remotely via the internet network. The SHS system was adapted from a $6 \times 9m2$ house which was divided into 6 different areas, the scope of management included the use of fan lights and cooking utensils that could be monitored or controlled through the internet network. The scope of management includes the use of lamps, fans, and cooking utensils.

As for realizing this, light sensors, gas sensors, temperature sensors and several relays are used. Furthermore, these devices are connected to the NodeMCU ESP8266 as a microcontroller embedded with the IoT module. All data is sent to the database / server via the internet, for further processing into a visual display that can be accessed via the user interface application. To meet these needs the software (cloud platform) Blynk platform. The system testing includes access test, monitoring test, and controlling test, and Performansi QoS (throughput, jitter, delay and packet loss).

From the results of the functional SHS the tool functions properly and for the results of the Performance (QoS) which is classified with the ETSI TIPHON standard, it is obtained 1.982667ms delay, 0.257 ms jitter, 0 Packet loss, 1.74 bit /milisec throughput which is categorized as very good.

Keywords : *Internet of Things, Monitoring, Controlling, Modul IoT, Smart Home System, Cloud platform,* Blynk *platform,* QoS,Functional test, Performanansi tes, *throughput,jitter,delay* dan *packet loss*

[Halaman Ini Sengaja Dikosongkan]

KATA PENGANTAR

Puji syukur penulis panjatkan atas kehadirat Allah SWT. dengan segala rahmat, karunia, petunjuk, serta kekuatan yang berlimpah tak terkira yang telah diberikan kepada penulis sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul :

Rancang Bangung *Smart Home System* Berbasis *Internet of Things*

Tugas Akhir ini ditujukan untuk memenuhi salah satu sebagian persyaratan guna menyelesaikan pendidikan Sarjana pada Bidang Studi Teknik Telekomunikasi Multimedia, Departemen Teknik Elektro, Institut Teknologi Sepuluh Nopember. Penulis menyadari bahwa dalam penyelesaian Tugas Akhir ini tentunya juga tidak terlepas dari bantuan dan dukungan dari berbagai pihak. Untuk itu dengan segala hormat, penulis ingin mengucapkan terima kasih kepada :

1. Allah SWT. atas limpahan rahmat dan karunia-Nya.
2. Kedua orang tua saya tercinta yang sudah memberikan kasih sayang yang luar biasa kepada penulis.

Penulis menyadari dan memohon maaf atas segala kekurangan pada Tugas Akhir ini. Semoga Tugas Akhir ini dapat bermanfaat dalam pengembangan ilmu pengetahuan dan teknologi di kemudian hari.

Surabaya, Juni 2020
Penulis

Muhammad Reza Rivandana
NRP. 07111640000117

[Halaman Ini Sengaja Dikosongkan]

DAFTAR ISI

HALAMAN JUDUL
PERNYATAAN KEASLIAN **6**
ABSTRACT ..... **iii**
KATA PENGANTAR ..... **v**
BAB 1 ..... **1**
PENDAHULUAN **1**
1.1 Latar Belakang **1**
1.2 Rumusan Masalah **1**
1.3 Tujuan Penelitian **2**
1.4 Batasan Masalah **2**
1.5 Metode Penulisan **3**
1.6 Sistematika Penulisan **3**
1.7 Manfaat penelitian **4**
BAB 2 ..... **5**
TINJAUAN PUSTAKA **5**
2.1 Konsep IoT **5**
2.1.1 Sejarah IoT 5
2.1.2 *Definisi IoT* 6
2.1.3 Persyaratan *IoT* 6
2.1.4 Layer IoT 8
2.1.5 KELEBIHAN DAN KEKURANGAN IOT 9
2.1.6 Macam-macam Bidang Penerapan IoT 11
2.7 Elements Of Smart Home **13**

2.9 Quality of Service **22**
2.10 Blynk **24**
2.11 Wireshark **24**
BAB III **29**
METODOLOGI PENELITIAN **29**
3.1 Diagram Alur Penelitian **29**
3.2 Konsep Rumah Pintar Berbasis *Internet of Thing* (IoT) 30
3.3 Pembuatan Prototipe *Smart Home System* (SHS) .. **32**
3.3.1 Perancangan Protoype 32
3.3.2 Penentuan Fasilitas Objek cerdas 33
3.3.3 Penentuan kebutuhan objek cerdas 33
3.3.4 Penentuan Kebutuhan Objek Cerdas 34
3.3.5 Rancangan Sistem Rumah Cerdas 35
3.3.6 Pembuatan rancanganan sistem *Hardware* 35
3.3.7 Pembuatan system perangkat lunak 37
3.3.8 Integrasi Sistem 40
3.3.9 Pengujian Performnansi (*Quality of Service)* 43
3.4.3 Skenario pengkuran 45
3.5 Standar QoS ETSI TIPHON **45**
BAB IV HASIL PENGUJIAN DAN PEMBAHASAN **47**
4.1 Hasil Implementasi **47**
4.1.1 Hasil Implementasi Hardware 47
4.2 Hasil Implementasi Softraware **48**

4.3 Hasil Pengujian Fungsional Sistem 50
4.3.1 Hasil Pengujian Koneksi Modul *IoT-Wifi*..... 50
4.3.2 Pengujian Monitoring Objek Cerdas..... 50
4.3.3 Hasil Pengujian Aktifasi Lampu..... 52
4.3.4 Hasil Pengujian Respond Sensor 53
4.4 Pengukuran Perfomansi (QoS) 54
4.4.1 Hasil Pengukuran Berdasar Lama Pengamatan..... 54
4.4.2 Hasil Pengukuran Berdasarkan Waktu Pengamatan..... 55
4.4.3 Hasil Pengukuran Berdasarkan Jarak 55
4.5 Analisa dan Pembahasan 56
4.6 Perbandingan Performansi Sistem 56
4.6.1 Perbandingan Delay 56
4.6.2 Perbandinga Jitter..... 57
4.6.3 Perbandingan Throughput 58
4.6.4 Perbandingan Packet loss 58
4.6.5 Klasifikasi Hasil Perfomransi (QoS)..... 59
BAB 5..... 118
PENUTUP..... 118
5.1 Kesimpulan..... 118
5.1.1 Hardware..... 118
5.1.2 Delay Throughput dan Jitter 118
5.2 Saran..... 118

Daftar Pustaka..... 120
LAMPIRAN..... 122
Lampiran A 122
LAMPIRAN B 132
***DATASHEET*..... 132**
NodeMCU ESP8266 Specifications & Features..... 133

DAFTAR GAMBAR

Gambar 2. 1 Layer IoT 8
Gambar 2. 2 Arduino Schematic.....
Gambar 2. 3 Pin ESP 8266 17
Gambar 2. 4 Sensor MQ 2 20
Gambar 2. 5 Sensor Suhu 20
Gambar 2. 6 Sensor PIR 21
Gambar 2. 7 Relay 6 Channel 22
Gambar 2. 8 Logo Blynk 24
Gambar 2. 9 Wireshark **25**

Gambar 3. 2 Diagram Blok SHS 30
Gambar 3. 4 Identifikasi Permasalahan 32
Gambar 3. 5 Rancangan Sistem Rumah Cerdas 35
Gambar 3. 6 Skematik Rangkaian 36
Gambar 3. 7 Lay Out PCB 37
Gambar 3. 8 UI Blynk 39
Gambar 3. 9 Intergrasi Sistem 40

DAFTAR TABEL

Tabel 3. 1 Penentuan Fasilitas Objek Cerdas 33
Tabel 3. 2 Penentuan Objek Cerdas..... 34
Tabel 3. 3 Penentuan Kebutuhan Objek..... 34
Tabel 3. 4 Pengujian Koneksi..... 42
Tabel 3. 6 Kriteria Delay..... 46
Tabel 3. 7 Delay 46
Tabel 3. 8 Jitter..... 46
Tabel 4. 1 Hasil Pengujian Modul IoT.....50
Tabel 4. 2 Hasil Pengujian Monitoring..... 51
Tabel 4. 3 Pengujian Aktifasi Lampu 52
Tabel 4. 4 Pengujian Uji Sensor 53
Tabel 4. 5 Hasil Berdasarkan Lama Pengamatan 54
Tabel 4. 6 Hasil Berdasarkan Waktu Pengamatan..... 55
Tabel 4. 7 Tabel Berdasarkan Jarak Pengamatan 55
Tabel 4. 8 Hasil Analisis Qos..... 60
Tabel 4. 9 Hasil Analisis Qos standart TIPHON..... 60

BAB 1
PENDAHULUAN

1.1 Latar Belakang

Gaung revolusi industri 4.0 terus menggema, seiring tekad pemerintah dalam mengadopsi konsep tersebut untuk diemplemantasikan dalam berbagai bidang kehidupan. Hal tersebut bertujuan agar bangsa Indonesia lebih kompetitif dalam menghadapi persaingan dengan bangsa lainnya yang lebih maju. Adapun Revolusi Industri 4.0 sendiri mementingkan integrasi antar alat internet dan pemanfaatan big data. Teknologi yang mengintegrasikan antar alat agar dapat di kelola dari jarak jauh sering disebut dengan *Internet of Things* (IoT), dan hal tersebut menjadi unsur utama dalam revolusi industri 4.0.

Konsep utama *Internet of Things* sesungguhnya memiliki tujuan untuk mengumpulkan berbagai data dari lapangan yang kemudian dilakukan pengolahan atau analisa. Hasil pengolahan data tersebut selanjutnya dapat diakses/ditampilkan kepihak penggunaan yang membutuhkan.

Penggunaan IoT makin hari makin meningkat untuk membantu menyelesaikan masalah sehari-hari. Dan dewasa ini IoT juga berpengaruh di berbagai macam industri seperti manufaktur, logistik, kesehatan, tata kota, rumah, pertanian, bahkan insdustri otomotif. Ada banyak keuntungan dari penggunaan IoT, seperti data yang masuk akan lebih banyak dan akan berakibat kepada proses bisnis yang lebih efisien.

Dalam tugas akhir ini akan dibuat salah satu implementasi IoT untuk mengelola sebuah rumah pintar, yang dapat dimonitor maupun dikendalikan dari jarak jauh via internet. Lingkup pengelolaan meliputi penggunaan lampu penerangan, pendingin ruangan, pemanas, dan sebagainya. Dengan realisasi sistem tersebut diharapkan tujuan efisiensi energi listrik dan faktor keamanan rumah diharapkan bisa menjadi lebih baik.

1.2 Rumusan Masalah

Berdasarkan dari latar belakang tersebut, maka dapat dirumuskan beberapa permasalahan sebagai berikut :

1) Bagaimana membangun perangkat IoT yang mampu mengendalikan pengelolaan penggunaan lampu penerangan, pendingin ruangan, pemanas, dan memberikan peringatan ketika terpantau ada kebocoran gas.
2) Bagaimana mengukur kualitas perangkat yang dibangun dengan melakukan monitoring dan analisis *Quality of Service QoS* pada perangkat yang telah dibangun.

1.3 Tujuan Penelitian

Adapun tujuan yang ingin dicapai dalam pembuatan Tugas Akhir ini adalah:

1) Membuat prototipe Smart Home System (SHS) yang dapat di akses dari jarak jauh leweat (PC/HP).
2) Mengimplemantasikan teknologi Internet of Things pada rancangan SHS yang dibuat.
3) Melakukan pengukuran kinerja pada sistem yang telah dibuat tersebut.
4) Melakukan evaluasi performansi berdasarkan standar *ITU-T* dan *TIPHON (ETSI)*

1.4 Batasan Masalah

Pembahasan ini dapat dilakukan secara terarah dan sesuai dengan yang diharapkan, maka perlu diterapkan batasan-batasan permasalahan yang akan dibahas. Batasan masalah dalam tugas akhir ini antara lain:

1) Rumah yang akan dibuat konsep SHS dalam Tugas Akhir diasumsikan, berukuran $6\times9m^2$ yang terdiri, 2 kamar tidur, 1 ruang tamu, 1kamar mandi, 1ruang makan, 1 dapur dan teras.
2) Objek yang akan di masukkan dalam sistem SHS terdiri dari 4 buah lampu, 1 buah kompor dan 1 buah kipas angin.
3) Sensor yang digunakan meliputi sensor suhu, gas dan PIR yang masing dihubungkan ke sebuah *relay*.
4) Mikrokontroler yang digunakan tipe *Internet of Things* yang sudah *embedded* dengan modul IoT.
5) Bahasa pemrograman yang digunakan adalah Arduino IDE dan Bahasa *C++*.
6) *User interface* menggunakan *Blynk* platfrom.

7) Pengukuran kinerja sistem menggunakan perangkat lunak *network monitoring Wireshark*, meliputi parameter *Throughput*, *Jitter*, dan *Delay*.
8) Standart pengukuran yang dipakai *ITU-T* dan *TIPHON (ETSI)*.

1.5 Metode Penulisan

Metode penulisan yang digunakan selama proses penyusunan dan realisasi tugas akhir ini adalah sebagai berikut:

1) Metode studi pustaka dilakukan dengan mengambil materi-materi terkait dari buku-buku, internet dan sumber lainya sebagai referensi.
2) Penelitian dan eksperimen yaitu dengan melakukan proses penelitian, perancangan dan pengujian aplikasi yang dibuat.
3) Konsultasi yaitu dengan berdiskusi dan bertanya secara langsung kepada dosen pembimbing.

1.6 Sistematika Penulisan

Sistematika penulisan yang digunakan dalam penelitian ini adalah sebagai berikut.

Bab 1

Membahas tentang latar belakang pengambilan judul penelitian, perumusan masalah, tujuan penelitian, batasan masalah, metode penulisan, sistematika penulisan dan manfaat penelitian.

Bab 2

Dijelaskan tentang tinjauan pustaka yaitu penggunaan dan beberapa materi penunjang lain yang digunakan sebagai acuan pada bab berikutnya.

Bab 3

Mengenai variabel penelitian, analisis mengenai sistem yang akan dibangun, dan perancangan perangkat elektronika dan alur program yang berjalan pada mikrokontroler, serta parameter yang digunakan dalam penelitian.

Bab 4

Ada bab ini berisi hasil dari rancangan yang sudah dibuat pada Bab 3. Dari pengujian ini kemudian dianalisis kinerja sistem

smarthome dan ditarik kesimpulan sementara mengenai parameter-parameter yang telah diuji.

Bab 5

Bab ini berisi tentang kesimpulan dan saran dari hasil pengujian yang telah dilakukan.

1.7 Manfaat penelitian

Dengan konsep *Smart Home System* (SHS) berbasis *Internet ofThings (IoT) yang dibuat* dalam tugas akhir ini diharapkan rumah pintar tersebut dapat dipantau(*monitoring*) dan dikendalikan (*controlling*) seperti mematikan atau menghidupkan objek-objek telah dihubingkan dan berada didalam rumah.

BAB 2
TINJAUAN PUSTAKA

Dibuat Landasan teori ini akan membahas Konsep IoT, *Arduino*, *ESP8266*, *sensor PIR*, *sensor MQ2*, *sensor Humidity* dan *QoS* menggunakan *Wireshark*. Semua pembahasan tersebut berguna dalam menunjang sistem yang akan dibuat, sehingga sistem dapat berjalan sesuai dengan yang diharapkan.

2.1 Konsep IoT

Internet of Things (IoT) merupukan sebuah konsep yang bertujuan untuk saat ini menghubungkan benda 1 ke benda lainnya. Pada bab ini akan dibahas tentang landasan teori yang berkaitan dengan sistem yang akan semua benda fisik ke jaringan internet. Kemunculan IoT mampu menjembatani mesin yang memproduksi data dengan memanfaatkan sensor,dalam oerangkat seperti ponsel dan perangkat pintar lainnya untuk mengumpulkan informasi secara realtime. [1]

2.1.1 Sejarah IoT

Mengingat bahwa IoT ini adalah teknologi canggih yang mampu melakukan transfer data lewat jaringan dengan interaksi yang mudah, masa depan dari pengembangannya jadi sangat menjanjikan. Kehidupan manusia sehari-harinya bisa dioptimalkan dan dipermudah dengan sensor cerdas dan peralatan pintar yang berbasis internet ini.Awalnya, internet itu sendiri mulai terkenal di tahun 1989. Lalu pada tahun 1990, seorang peneliti bernama John Romkey membuat suatu perangkat yang kala itu tergolong canggih. Perangkatnya adalah pemanggang roti yang bisa dinyalakan atau juga dimatikan lewat internet.Kemudian di tahun 1994, seseorang bernama Steve Mann menciptakan WearCam, dan pada tahun 1997-nya si Paul Saffo menjelaskan secara singkat mengenai penemuannya soal teknologi sensor dan masa depannya nanti. Barulah di tahun 1999 Kevin Ashton membuat konsep Internet of Things. Kevin ini adalah Direktur Auto IDCentre dari MIT.Di tahun yang sama, yaitu 1999, ditemukan mesin yang sistemnya berbasis Radio Frequency Identification (RFID) secara global. Nah, penemuan inilah yang jadi awal kepopuleran dari konsep IoT. Orang-orang, terutama pakar teknologi jadi berlomba-lomba mengembangkan teknologinya sesuai konsep IoT.Lalu, di tahun

2000, brand ternama LG mengumumkan rencananya untuk membuat dan merilis teknologi IoT yaitu lemari pintar. Lemari pintar ini mampu menentukan apakah ada stok makanan yang perlu diisi ulang dalam lemarinya.Kemudian, di tahun 2003, FRID yang sebelumnya telah disebutkan, mulai ditempatkan pada posisi penting dalam masa pengembangan teknologi di Amerika, melalui Program Savi. Pada tahun yang sama pula, perusahaan ritel raksasa Walmart mulai menyebarkan RFID di semua cabang tokonya yang tersedia di berbagai belahan dunia.IoT kembali terkenal di tahun 2005, yaitu pada saat media-media ternama semacam The Guardian dan Boston Globe mulai mengutip banyak sekali dari artikel ilmiah dan proses pengembangan IoT. Hingga tahun 2008, berbagai macam perusahaan setuju untuk meluncurkan IPSO untuk memasarkan penggunaan IP dalam jaringan bagi “Smart Object” yang juga bertujuan mengaktifkan IoT itu sendiri.[1]

2.1.2 *Definisi IoT*

Menurut wikipedia Definisi Internet of Things telah berkembang karena konvergensi beberapa teknologi, analitik waktu-nyata, pembelajaran mesin, sensor komoditas, dan sistem tertanam. [1] Bidang tradisional sistem tertanam, jaringan sensor nirkabel, sistem kontrol, otomatisasi (termasuk otomatisasi rumah dan bangunan), dan lainnya semuanya berkontribusi untuk memungkinkan Internet of Things. Di pasar konsumen, teknologi IoT paling identik dengan produk yang berkaitan dengan konsep "rumah pintar", meliputi perangkat dan peralatan (seperti perlengkapan pencahayaan, termostat, sistem dan kamera keamanan rumah, dan peralatan rumah lainnya) yang mendukung satu atau ekosistem yang lebih umum, dan dapat dikontrol melalui perangkat yang terkait dengan ekosistem itu, seperti smartphone dan speaker pintar.[1]

2.1.3 Persyaratan *IoT*

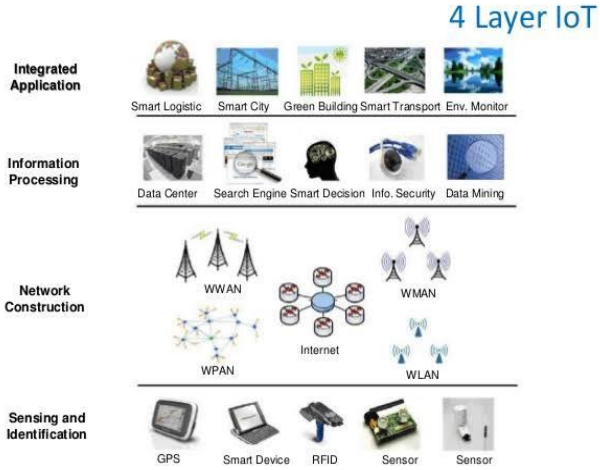
Ada beberapa unsur pembentuk IoT yang mendasar termasuk kecerdasan buatan, konektivitas, sensor, keterlibatan aktif serta pemakaian perangkat berukuran kecil. Berikut, kami akan menjelaskan masing-masing unsur pemberntuk tersebut dengan singkat:

- **Kecerdasan Buatan (*Artificial Intelligence/AI*)** − IoT membuat hampir semua mesin yang ada menjadi “Smart”. Ini berarti IoT bisa

meningkatkan segala aspek kehidupan kita dengan pengembangan teknologi yang didasarkan pada AI. Jadi, pengembangan teknologi yang ada dilakukan dengan pengumpulan data, algoritma kecerdasan buatan, dan jaringan yang tersedia. Sebenarnya ya contohnya bisa jadi mesin yang tergolong sederhana semacam meningkatkan/mengembangkan lemari es/kulkas Anda sehingga bisa mendeteksi jika stok susu dan sereal favorit Anda sudah hampir habis, bahkan bisa juga membuat pesanan ke supermarket secara otomatis jika stok mau habis. Penerapan kecerdasan buatan ini memang sangatlah menarik.[1]

- **Konektivitas** − Dalam IoT, ada kemungkinan untuk membuat/membuka jaringan baru, dan jaringan khusus IoT. Jadi, jaringan ini tak lagi terikat hanya dengan penyedia utamanya saja. Jaringannya tidak harus berskala besar dan mahal, bisa tersedia pada skala yang jauh lebih kecil dan lebih murah. IoT bisa menciptakan jaringan kecil tersebut di antara perangkat sistem.
- **Sensor** − Sensor ini merupakan pembeda yang membuat IoT unik dibanding mesin canggih lainnya. Sensor ini mampu mendefinisikan instrumen, yang mengubah IoT dari jaringan standar dan cenderung pasif dalam perangkat, hingga menjadi suatu sistem aktif yang sanggup diintegrasikan ke dunia nyata sehari-hari kita.
- **Keterlibatan Aktif (Active Engagement)** − Engangement yang sering diterapkan teknologi umumnya yang termasuk pasif. IoT ini mengenalkan paradigma yang baru bagi konten aktif, produk, maupun keterlibatan layanan.

2.1.4 Layer IoT



Gambar 2. 1 *Layer IoT*

- **Lapisan pertama Internet of Things terdiri dari perangkat IOT yang terhubung dengan Sensor:**
  Ini adalah perangkat elektronik kecil, terbatas memori, dan sering dioperasikan dengan baterai dengan sensor dan aktuator terpasang. Tiga kemampuan utama dari perangkat IOT yang khas adalah:
  A. Bisa merasakan dan merekam data
  B. Mampu melakukan komputasi ringan dan akhirnya
  C. Dapat terhubung ke jaringan dan mengkomunikasikan data
  Contohnya termasuk pelacak kebugaran, sensor kelembaban tanah pertanian, sensor medis untuk mengukur kadar glukosa darah dan banyak lagi.
- **Lapisan kedua terdiri dari perangkat gateway IOT:**
  Berbagai perangkat IOT dari lapisan 1 harus terhubung ke internet melalui perangkat komputasi yang lebih kuat yang disebut gateway IOT yang terutama bertindak seperti perangkat jaringan. Jadi, mirip dengan bagaimana router WiFi membantu kita menghubungkan banyak laptop, ponsel, dan tablet ke internet di rumah, gateway IOT mengumpulkan data dari berbagai perangkat penginderaan dan menyampaikannya ke cloud.

Gerbang ini adalah komponen penting dari ekosistem IOT. Biasanya, gateway IOT dilengkapi dengan berbagai kemampuan komunikasi (seperti Bluetooth, Zigbee, LoRa WAN, protokol berpemilik GHz) untuk berbicara dengan perangkat IOT di satu sisi dan koneksi ke jaringan berbasis IP (Internet) di sisi lain (melalui WiFi, Ethernet atau tautan Seluler).

- **Lapisan ketiga IOT adalah Cloud:**
  Semua data sensor yang disampaikan oleh gateway IOT disimpan di server yang di-*hosting cloud*. Server ini menerima, menyimpan, dan memproses data untuk analisis dan pengambilan keputusan. Lapisan ini juga memungkinkan pembuatan dasbor langsung yang dapat dipantau oleh pembuat keputusan dan pengambilan keputusan berdasarkan data yang proaktif. Saat ini, hampir semua perusahaan komputasi awan memiliki penawaran layanan khusus untuk solusi IOT.
- **Dan lapisan terakhir adalah IOT Analytics:**
  Di sinilah keajaiban terjadi dan data mentah yang dikumpulkan diubah menjadi wawasan bisnis yang dapat ditindaklanjuti, yang dapat membantu meningkatkan operasi bisnis, efisiensi, atau bahkan memprediksi peristiwa di masa depan seperti kerusakan mesin. Lapisan ini menggunakan ilmu data dan teknik analisis yang berbeda termasuk algoritma pembelajaran mesin untuk memahami data dan memungkinkan tindakan korektif.[1]

2.1.5 KELEBIHAN DAN KEKURANGAN IOT

Kelebihan dan kekurangan IoT, pada bab ini akan membahas tentang kekurangan dan kelebihan IoT (*Internet of Things*) berikut

Kelebihan *IoT*

1. Data

Semakin banyak informasi yang diperoleh, semakin mudah untuk menentukan tindakan yang tepat berdasar data yang ada. Dengan bantuan komputer dan algoritma program kita tidak perlu mengecek data satu per satu, biarkan mesin yang melakukan nya sesuai algoritma yang kita inginkan, selain cepat juga sangat akurat.

2. Waktu

Dengan bantuan sistem komputer yang telah diprogram sebelumnya untuk mengolah informasi tertentu dan melakukan tindakan sesuai yang telah diprogramkan maka proses analisis dan pengambilan

keputusan berdasar data yang besar akan sangat cepat. Tidak bisa dibayangkan jika hal ini dilakukan secara manual tanpa bantuan mesin.

3. Tracking

Dalam sistem inventory dengan bantuan komputer akan sangat mudah untuk mengecek persedian, lokasi dan kualitas barang sehingga memudahkan kita untuk melakukan pengelolaan sehingga tidak ada kasus kehabisan barang karena lalai dalam pengecekan jika dilakukan secara manual.

4. Biaya

Tidak bisa dipungkiri, penggunaan tenaga manusia yang terbatas kemampuan nya yang berakibat diperlukan banyak tenaga manusia untuk melakukan pekerjaan yang berat. Dengan bantuan mesin yang kemampuannya dapat diatur dan dapat menggantikan pekerjaan manusia, manusia tidak perlu melakukan hal berat dan rumit di zaman sekarang, cukup dengan menjadi operator mesin saja. Dari sini terlihat bahwa biaya untuk menggaji karyawan lebih sedikit karena sudah digantikan oleh mesin.

Dibalik kemudahan dan kecanggihan yang tersaji ketika menggunakan perangkat IoT ada beberapa risiko yang perlu Anda ketahui, berikut kekurangan IoT diantaranya:

1. Harus Sesuai

Kita tau bahwa tidak ada standardisasi penggunaan sensor seperti penggunaan USB, ketika sebuah sistem dengan IoT device mengalami kerusakan maka kita harus membeli di vendor yang sama untuk menggantinya.

2. Dirangkai Secara Kompleks

Dibalik berbagai kemudahan yang telah diberikan kepada para pengguna, di sana ada IoT module yang sudah dirangkai secara komplex untuk menerima dan mengolah informasi dan alat tersebut memerlukan tega ahli untuk merawat secara berkala agar sistem tetap berjalan.

3. Memiliki Celah Keamanan

Semua perangkat dan program komputer pasti memiliki sebuah celah dan rawan akan tindakan hacking, baru-baru ini terjadi serangan DDOS besar-besaran yang disinyalir berasal dari perangkat IoT yang telah di hack dan dijadikan botnet untuk melakukan serangan ini. Oleh karena itu, diperlukan seorang tenaga ahli untuk mengamankan perangkat IoT dari serangan *Hacker*.

2.1.6 Macam-macam Bidang Penerapan IoT

1. Pertanian

Ada berbagai macam pengaplikasani IoT di sektor pertanian. Beberapa diantaranya seperti mengumpulkan data soal suhu, curah hujan, kelembaban, kecepatan angin, serangan hama, dan muatan tanah. Data-data tersebut bisa dipakai buat mengotomatisasi teknik pertanian. Kemudian, bisa juga dipakai untuk mengambil keputusan (decision making) berdasarkan informasi yang ada demi meningkatkan kualitas dan kuantitas, meminimalkan risiko dan limbah, serta mengurangi upaya yang diperlukan dalam mengelola tanaman. Sebagai contoh, petani sekarang sudah bisa memantau suhu dan kelembaban tanah dari jauh, dan bahkan menerapkan data yang diperoleh IoT untuk program pemupukan yang lebih presisi.

2. Energi

Sejumlah besar perangkat yang memakan energi (semacam switch, outlet listrik, lampu, televisi, dll.) kini sudah bisa terintegrasi dengan konektivitas internet. Lalu apa dampaknya? Integrasi itu memungkinkan mesin-mesin ataupun jaringan untuk berkomunikasi dalam menyeimbangkan pembangkitan listrik serta penggunaan energi yang lebih hemat maupun efektif. Perangkat ini juga bisa memungkinkan akses remote control dari pengguna, atau bisa juga manajemen dari satu pusat lewat interface yang berberbasis cloud. Selain itu, bisa juga mengaktifkan fungsi semacam penjadwalan

(misalnya untuk menyalakan/mematikan mesin pemanas, mengendalikan oven, mengubah kondisi pencahayaan dari terang menjadi redup hingga ke gelap, dan lain sebagainya). Jadi dengan IoT di bidang ini, sistem bisa berkumpul dan bertindak berdasarkan informasi yang terkait dengan energi dan daya demi meningkatkan efisiensi produksi dan distribusi listrik.

3. Lingkungan

Aplikasi pemantauan lingkungan dari IOT biasanya pakai sensor dalam membantu terwujunya perlindungan lingkungan. Contohnya seperti apa? Penerapannya misalnya dengan memantau kualitas udara atau air, kondisi atmosfer atau tanah, bahkan juga bisa mencakup

pemantauan teerhadap satwa liar dan habitatnya. Tak hanya itu sebenarnya. Bisa juga IoT ini dimanfaatkan dalam penanggulangan bencana semacam sistem peringatan dini Tsunami atau gempa bumi. Hal ini tentunya bisa sangat membantu. Perangkat IoT dalam hal ini berarti punya jangkauan geografis yang sangat luas serta mampu bergerak.

4. *Smart home*

Perangkat IoT juga bisa dipakai untuk memantau dan mengontrol sistem mekanis, elektrik, dan elektronik yang digunakan di berbagai jenis bangunan (misalnya, industri atau juga rumah Anda sebagai tempat tinggal). Alat atau pengembangan IoT ini juga bisa memantau penggunaan energi secara real-time untuk mengurangi konsumsi energi. Tak hanya itu, bahkan bisa juga melakukan pemantauan terhadap para penghuninya. Contohnya? Begitu Anda masuk ke rumah di malam hari, lampu menyala. Kemudian begitu Anda masuk ke jadwal tidur, lampu akan mati secara otomatis. Pagi hari, taman Anda akan disiram air oleh mesin penyiram otomatis. Begitu juga dengan kulkas Anda yang bisa memesan stok makanan sendiri ketika habis. Semuanya bisa terintegrasi menjadi sistem rumah pintar.

4. Medik dan Kesehatan

Dalam dunia medik dan kesehatan, IoT akan dikembangkan terus. Bahkan, nanti di masa yang akan datang, seluruh rekaman kesehatan Anda bisa ditransfer langsung ke tenaga medis maupun Rumah Sakit. Data-data yang bisa dideteksi dan dikirimkan semacam detak jantung, tingkat gula dalam darah, dan lain sebagainya. Smartphone/ponsel pribadi Anda akan jadi alat pemantau kesehatan yang canggih dan tentunya bisa sangat membantu Anda. Perangkat IoT yang ada bahkan bisa memberikan peringatan saat kesehatan Anda menurun atau memberikan saran pengobatan dan bahkan membuat janji temu dengan dokter. Sebenarnya beberapa teknologi canggih IoT telah dikembangkan dan diterapkan di bidang ini. Contohnya, tempat tidur pintar yang bisa otomatis memberitahukan dokter/perawat ketika pasien hendak bangun dari tempat tidur dll. Menurut laporan dari Goldman Sachs di tahun 2015, perangkat kesehatan semacam ini bisa menyelamatkan negara dari anggaran kesehatan yang berlebihan

.

5. Transportasi

IoT bisa membantu manusia dalam integrasi komunikasi, kontrol, dan pemrosesan informasi pada berbagai sistem transportasi yang ada. Penerapan IoT memang terus-menerus meluas ke berbagai aspek sistem transportasi. Tak hanya teknologi mesinnya yaitu kendaraan, tetapi juga infrastruktur, serta menyinggung fungsi pengemudi/penggunanya. Interaksi dinamis yang terjadi antara komponen-komponen itu berasal dari sebuah sistem transportasi. Sistem tersebut memungkinkan komunikasi antar dan intra kendaraan, kontrol lalu lintas yang lebih efektif karena tergolong cerdas, parkir yang lebih cerdas, manajemen logistik dan armada, kontrol kendaraan, dan juga terkait faktor keselamatan maupun bantuan di jalan.

2.7 Elements Of Smart Home

Sistem Smart Home adalah kontrol dan manajemen yang terintegrasi banyak kecil sistem di rumah. Sistem kecil dapat berupa sakelar lampu, pemantauan suhu, gerakan deteksi, pengawasan rumah dan sensor lainnya. Sensor dalam sistem ini akan dikontrol oleh pengguna yang menggunakan perangkat antarmuka seperti remote control, komputer, dan smartphone. Oleh meningkatkan jenis sensor yang akan dikontrol, sistem utama perlu lebih terspesialisasi mengintegrasikan sub-sistem untuk menjadi sistem Smart Home. Jaringan sistem bisa kabel atau nirkabel tergantung pada aplikasi. Tabel 1 menunjukkan ringkasan Smart Home elemen sistem. Elemen dalam sistem Smart Home Contoh Sensor Pemantauan suhu, deteksi kebakaran, pengawasan rumah Perangkat antarmuka pengguna Remote control, komputer, Smartphone, tablet Jenis jaringan Wired-Fiber optic, kabel coaxial Wireless-Bluetooth, WiFi, ZigBee, RF Kontrol sentralisasi Pengontrol mikro, PLC, komputer, FPGA[2]

1. Arduino

Secara umum *arduino* menggunakan chip mikro kontroler ATmega yang merupakan prosesor yang banyak digunakan dalam membuat aplikasi sistem kendali bidang instrumentasi dan untuk *arduino* sendiri sudah di bentuk dalam prototyping board, mikrokontroler seri AVR pertama kali diperkenalkan ke pasaran sekitar tahun 1997 oleh

perusahaan Atmel, yaitu sebuah perusahaan yang sangat terkenal dengan produk mikrokontroler, stuktur pin *arduino* adalah sebagai berikut :

Gambar 2.2 *Arduino* Schematic [3]

Pada gambar di atas di tunjukan susunan rangkaian kaki-kaki pada board *arduino*, sebagai contoh di atas adalah *arduino* nano, *arduino* nano memiliki spesifikasi sebagai berikut :[3]

a. Mikrokontroller : *Atmel ATmega168*
b. Tegangan kerja : 5 *Volt*
c. Tegangan input:Optimal : 7 – 12 *Volt*
d. Minimum : 6 *Volt*
e. Maksimum : 20 *Volt*
f. Digital pin I/O : pin *D0* sampai pin *D13*
g. 6 pin *PWM*
h. Analog pin : 8 pin yaitu pin *A0* sampai pin *A3*
i. Arus listrik maksimum : 40 *mA*
j. *Flash memori* : 32 *Mbyte* untuk *Arduino* nano 3.x,16 *Mbyte* untuk *Arduino* nano 2.x Besar *flash memori* ini dikurangi 2 *kbyte* yang digunakan untuk menyimpan *file boatloader*.
k. *SRAM* : 1 *kbyte* (ATmega168) dan 2 kbyte (ATmega328)

l. *EEPROM* : 512 *byte* (Atmega168) dan 1 kbyte (Atmega328)
m. Kecepatan *clock* : 16 *MHz*
n. Ukuran *board* : 4,5 mm x 18 mm
o. Berat : 5 gram
p. Daya : catu daya sebesar 5v
q. Memori : Atmega 168 dilengkapi dengan *flash*

Selain dilengkapi dengan flash memori juga dilengkapi dengan *SRAM* dan *EEPROM*. *SRAM* dan *EEPROM* dapat digunakan untuk menyimpan data selama program utama bekerja. Besar *SRAM* untuk ATmega168 adalah 1 kb dan untuk ATmega328 adalah 2 kb sedangkan besar *EEPROM* untuk ATmega168 adalah 512 b dan untuk ATmega328 adalah 1 kb[3].

Input dan Output *Arduino* pro mini mempunyai 14 pin digital yang dapat digunakan sebagai pin input atau output. Pin ini akan mengeluarkan tegangan 5V untuk mode *HIGH* (logika 1) dan 0V untuk mode *LOW* (logika 0) jika dikonfigurasikan sebagai pin output. Jika di konfigurasikan sebagai pin input, maka ke 14 pin ini dapat menerima tegangan 5V untuk mode *HIGH* (logika1) dan 0V untuk mode *LOW* (logika 0). Besar arus listrik yang diijinkan untuk melewati pin digital I/O adalah 40 mA. Pin digital I/O ini juga sudah dilengkapi dengan resistor *pull-up* sebesar 20-50 kΩ. Ke 14 pin digital I/O ini selain berfungsi sebagai pin I/O juga mempunyai fungsi khusus yaitu :

Pin Digital 0 dan pin Digital 1 juga di fungsikan sebagai pin TX dan RX pada komunikasi data serial. Pin Digita 2 dan pin Digital 3 dapat di fungsikan sebagai pin untuk interupsi eksternal. Kedua pin ini dapat dikonfigurasikan untuk pemicu interupsi eksternal. Interupsi dapat terjadi ketika timbul kenaikan atau penurunan tegangan pada pin Digital 2 atau pin Digital 3. Pin Digital 4, pin Digital 5, pin Digital 6, pin Digital 9, pin Digital 10 dan pin Digital 11 dapat digunakan sebagai pin PWM (*pulse width modulator*). *Arduino* nano juga dilengkapi dengan 8 buah pin analog, pin analog ini terhubung ke ADC (*analog to digital converter*) internal, pada kondisi awal, pin analog ini dapat mengukur tegangan dari 0V sampai 5 V DC dengan besar arus maksimum 40 mA range ini dapat diubah dengan memberikan sebuah tegangan referensi dari luar melalui pin Vref. Pin analog selain dapat digunakan untuk input data analog, juga dapat digunakan sebagai pin digital I/O, kecuali pin A6 dan A7- yang hanya dpat digunakan untuk input data analog saja. Pin reset digunakan

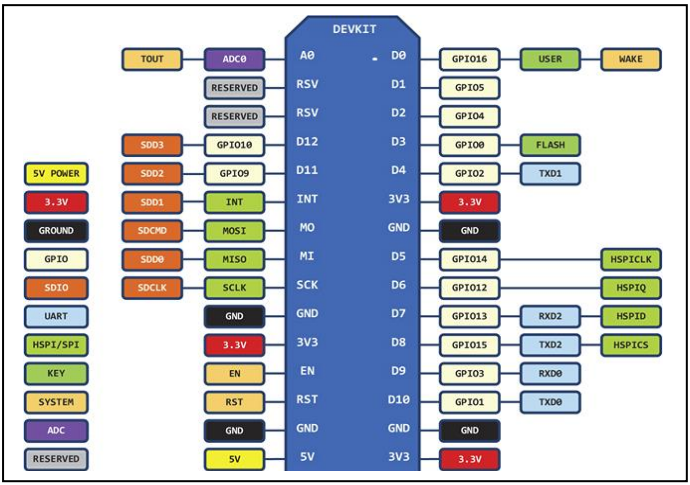
untuk mereset board *Arduino* nano, yaitu dengan menghubungkan pin ini ke ground selama beberapa milidetik, selain dapat direset melalui pin reset juga dapat direset dengan menggunakan tombol reset yang terpasang pada board *Arduino*[4].

Komunikasi *Arduino* sudah dilengkapi dengan beberapa fasilitas untuk komunikasi yang dapat digunakan untuk berkomunikasi dengan komputer (PC atau Laptop), atau dengan board mikrokontroller lainnya. ATmega168 dan ATmega328 dilengkapi dengan komunikasi serial UART TTL (5V), yang terdapat pada pin D0 dan pin D1. Board juga dilengkapi dengan sebuah IC FTDI 232 Rl yang dapat dihubungkan langsung ke komputer untuk menghasilkan sebuah virtual com-port pada operating sistem.[4]

Software *Arduino* (sketch) sebagai IDE *Arduino* juga dilengkapi dengan serial monitor yang memungkinkan programmer untuk menampilkan data serial sederhana yang dapat dikirim atau diterima dari board *Arduino* nano. Led RX dan TX yang terpasang pada board. Selain dapat berkomunikasi dengan menggunakan data serial melalui virtual com-port, *Arduino* pro mini juga dilengkapi dengan mode komunikasi I2C (TWI) dan SPI untuk komunikasi antar hardware. Pemograman *Arduino* nano dapat dengan mudah diprogram dengan menggunakan software *Arduino* (sketch). *Arduino* nano sudah dilengkapi dengan program boatloader, sehingga programmer dapat langsung meng-up-load kode program langsung ke board *Arduino* nano tanpa melalui board perantara atau hardware lain. Programmer juga dapat mem-up-load program ke board *Arduino* pro mini tanpa menggunakan boatloader, tetapi melalui ICSP (in-circuit serial programming) header yang sudah tersedia di board *Arduino* nano. [4]

2. ESP8266

ESP8266 adalah sebuah modul *WiFi* yang akhir-akhir ini semakin digemari para hardware developer, selain karena harganya yang sangat terjangkau, modul *WiFi* serbaguna ini sudah bersifat System on Chip, sehingga kita bisa melakukan programming langsung ke ESP8266 tanpa memerlukan mikrokontroller tambahan. Kelebihan lainnya ESP8266 ini dapat menjalankan peran sebagai adhoc acces point pada jaringan wifi dan dapat digunakan sebagai *Mini Webserver* maupun *client* sekaligus. ESP8266 dikembangkan oleh pengembang asal Tiongkok yang bernama “Espressif”.



Gambar 2.3 merupakan Susunan pin pada ESP8266.[3]

Terdapat pin Tx Rx untuk mengirim dan menerima data dalam bentuk Serial dan dapat berkomunikasi dari mikrocontroller lain seperti *Arduino* atau *microcomputer raspberry pi* atau dari sensor-sensor kemudian terdapat kaki untuk catu daya sebesar 5v serta terdapat pin *general purpose input output* untuk keperluhan pengiriman sinyal *High* atau *Low* yang dapat digunakan juga sebagai pemicu untuk modul *relay*.

2. *Android*

Sistem operasi *Android* merupakan sistem operasi yang berbasis *Linux* untuk *mobile* seperti *smartphone* dan komputer Tablet. Android menyediakan *platform* terbuka bagi pengembang untuk menciptakan aplikasi mereka sendiri agar dapat digunakan oleh banyak orang. Awalnya *Google* membeli *Android.Inc*. yang merupakan pendatang baru yang bergerak dalam pengembangan dan pembuatan piranti lunak untuk ponsel, kemudian untuk mengembangkan sistem operasi Android dibentuklah *Open Handset Alliance* dengan konsorsium dari 34 perusahaan piranti keras, piranti lunak, dan telekomunikasi termasuk *Google*, *HTC*, *intel*, *Motorola*, dan lain-lain[16].

Android.Inc adalah sebuah perusahaan *software* yang didirikan pada bulan Oktober 2003 di Palo Alto California USA dan didirikan oleh beberapa senior dari beberapa perusahaan yang berbasis *IT* & *Communication* yakni Andy Rubin, Rich Miner, Nick Sears dan Chris White. *Android.Inc* didirikan untuk mewujudkan *mobile device* yang lebih peka terhadap lokasi dan preferensi pemilik.

Berikut merupakan macam-macam versi *Android* yang dimulai dengan *Android* versi 1.1 dilengkapi dengan pembaharuan ekstetis pada aplikasi, jam alarm, pencarian suara, pengiriman pesan *Gmail* dan pemberitahuan *email*, Android versi 1.5 (*Cupcake*) dibekali dengan kemampuan merekam dan dukungan *Bluetooth* A2DP serta kemampuan terhubung secara otomatis ke headset *Bluetooth*. Animasi layar dan *keyboard* pada layar yang dapat disesuaikan dengan sistem, *Adroid* versi 1.6 (*Donut*) dapat menampilkan proses pencarian yang lebih baik dibanding sebelumnya, penggunaan baterai indikator dan fitur lainnya adalah gallery yang memungkinkan pengguna untuk memilih foto yang akan dihapus, kamera, *camcorder* dan gallery yang terintegrasi dilengkapi dengan jaringan *CDMA/EVDO*, 802.1x,*VPN*, *Gestures*, dan *Text-to-speech engine*, kemampuan *dial contact*, teknologi *text to change speech* dan pengadaan resolusi VWGA, *Android* versi 2.0/2.1 (*Éclair*) ditambahkan pengoptimalan *hardware*, peningkatan *Google Maps*, perubahan *User Interface* dengan *browser* baru dan dukungan HTML5, daftar kontak yang baru, dukungan *flash* untuk kamera 3,2 MP, *digital Zoom*, dan *Bluetooth* 2.1, *Android* versi 2.2 (*Frozen Yogurt/Froyo*) ditambahkan fasilitas *Adobe Flash* 10.1 dengan kecepatan kinerja dan aplikasi lebih cepat serta integrasi V8 *JavaScript engine* yang dipakai pada *Google Chrome* yang mempercepat kemampuan *rendering* pada *browser*, pemasangan aplikasi dalam SD Card, kemampuan *Wifi Hotspot portable* dan kemampuan *auto update* dalam aplikasi *Android Market*, *Android* versi 2.3/2.3.3 (*Gingerbread*) peningkatan pada versi ini adalah pada kemampuan *game*, peninkatan fungsi *copy paste*, layar antar muka didesain ulang, dukungan format video *VP8* dan *WebM*, efek audio serta dukungan kemampuan *Near Field Communication*, *Android* versi 3.0/3.1 (*Honeycomb*) dengan ukuran layar yang lebih besar *user interface* pada *Honeycomb* juga berbeda karena sudah didesain untuk tablet. *Honeycomb* juga mendukung multi procesor dan juga akselerasi perangkat keras untuk pemrosesan grafis yang lebih baik, *Android* versi 4.0 (*Ice Cream Sandwich*) menyertakan fitur membuka kunci dengan pengenalan wajah, jaringan data pemantauan penggunaan dan control terpadu kontak

jaringan sosial, perangkat tambahan untuk fotografi, mencari *email* secara *offline* dan berbagi informasi dengan menggunakan *Near Field Communication, Android* versi 4.1/4.2/4.3*(Jelly Bean*) pada versi ini memungkinkan melakukan gerakan atau perpindahan yang sangat cepat dan lembut dari menu satu ke menu lainnya tanpa adanya *delay, Android* versi 4.4(*KitKat*) dilengkapi dengan kemampuan mengoptimasi memori dengan meningkatkan kemampuan fitur layar sentuh, selain itu juga terdapat fitur yang secara otomatis dapat menyembunyikan segala macam tampilan kecuali gambar atau teks yang ingin dilihat[4].

Google sebagai pengembang *Android* yang kemudian diasuh oleh *Open Handset Alliance* mengibaratkan Android sebagai sebuah tumpukan *software*. Setiap lapisan dari tumpukan ini menghimpun beberapa program yang mendukung fungsi-fungsi spesifik dari sistem operasi. Lapisan paling bawah adalah kernel, *Google* menggunakan *kernel Linux* versi 2.6 untuk membangun *Android* yang mencakup menejemen memori, keamanan, menejemen daya, dan beberapa *driver hardware*. Bertempat di level yang sama dengan library adalah lapisan *runtime* yang mencakup serangkaian inti *library Java*, dengan ini para *programmer* dapat mengembangkan aplikasi untuk *Android* dengan menggunakan bahasa pemrograman *Java*. Lapisan selanjutnya adalah *application framework*, yang mencakup program untuk mengatur fungsi-fungsi dasar sebuah *smartphone*.[5]

3. Sensor MQ2

Berdasarkan datasheet sensor *LPG MQ2*bekerja untuk mendeteksi kebocoran gas namun untuk tiap sensor memiliki prinsip kerja yang berbeda-beda, berikut merupakan penjelasan untuk masing-masing sensor.[6]

Sensor gas tipe *MQ2*:

a. Mendeteksi material SnO2 pada udara, cara kerja sensor ini ketika kadar gas LPG pada udara rendah nilai konduktifitas pada perangkat ini rendah, pada konsentrasi LPG yang tinggi di udara sekitarnya menyebabkan nilai konduktifitas pada perangkat ini menjadi tinggi, dengan memanfaatkan perubahan inilah dapat di deteksi bila terjadi kebocoran gas
b. Memiliki sensitivitas bagus untuk pendeteksian pada ruang yang luas
c. Sensitive terhadap LPG

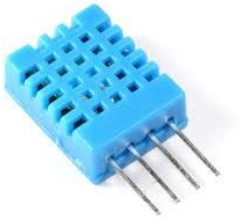
d. Pengaplikasiannya untuk pendeteksi kebocoran LPG *portable* dan pada perusahaan

Gambar 2. 3 *Sensor MQ 2[7]*

Pada gambar diatas merupakan bentuk sensor MQ2 yang dapat digunakan untuk mendeteksi keberadaan gas LPG pada udara. Untuk penerapan sensor ini menggunakan komunikasi di pin serial pada perangkat mikrokontroler.[7]

4. Sensor Suhu DHT 11

Sensor suhu DHT 11 adalah komponen elektronika yang memiliki fungsi untuk mengubah besaran suhu menjadi besaran listrik dalam bentuk tegangan.



Gambar 2. 4 Sensor Suhu DHT11[7]

Sensor suhu DHT 11 adalah komponen elektronika yang memiliki fungsi untuk mengubah besaran suhu menjadi besaran listrik dalam bentuk tegangan. Sensor Suhu LM35 yang dipakai dalam penelitian ini berupa komponen elektronika elektronika yang diproduksi

oleh National Semiconductor.DHT 11 memiliki keakuratan tinggi dan kemudahan perancangan jika dibandingkan dengan sensor suhu yang lain, DHT 11 juga mempunyai keluaran impedansi yang rendah dan linieritas yang tinggi sehingga dapat dengan mudah dihubungkan dengan rangkaian kendali khusus serta tidak memerlukan penyetelan lanjutan.[7]

6. Sensor PIR

Sensor PIR atau disebut juga dengan *Passive Infra Red* merupakan sensor yang digunakan untuk mendeteksi adanya pancaran sinar infra merah dari suatu object, berikut gambar Sensor PIR:

Gambar 2. 5 *Sensor PIR* [7]

Sensor PIR atau disebut juga dengan *Passive Infra Red* merupakan sensor yang digunakan untuk mendeteksi adanya pancaran sinar infra merah dari suatu object. Sesuai dengan namanya sensor PIR bersifat pasif, yang berarti sensor ini tidak memancarkan sinar infra merah melainkan hanya dapat menerima radiasi sinar infra merah dari luar. Sensor PIR dapat mendeteksi radiasi dari berbagai objek dan karena semua objek memancarkan energi radiasi, sebagai contoh ketika terdeteksi sebuah gerakan dari sumber infra merah dengan suhu tertentu yaitu manusia mencoba melewati sumber infra merah yang lain misal dinding, maka sensor akan membandingkan pancaran infra merah yang diterima setiap satuan waktu, sehingga jika ada pergerakan maka akan terjadi perubahan pembacaan pada sensor Sesuai dengan namanya sensor PIR bersifat pasif, yang berarti sensor ini tidak memancarkan sinar infra merah melainkan hanya dapat menerima radiasi sinar infra merah dari luar. Sensor PIR dapat mendeteksi radiasi dari berbagai objek dan karena semua objek memancarkan energi radiasi.[7]

7. *Relay* 6 *Channel*

Modul Arduino Relay memungkinkan berbagai mikrokontroler seperti Arduino, AVR, PIC, ARM dengan output digital untuk mengontrol beban dan perangkat yang lebih besar seperti AC atau Motor DC, elektromagnet, solenoid, dan bola lampu pijar, berikut gambar Relay 6 channel:

Gambar 2. 6 *Relay 6 Channel*

. Modul ini dirancang untuk diintegrasikan dengan 6 relay yang mampu mengendalikan 6 relay. Relai perisai menggunakan satu relai berkualitas tinggi QIANJI JQC-3F dengan beban pengenal 7A / 240VAC, 10A / 125VAC, 10A / 28VDC. Keadaan output relai secara individual ditandai dengan dioda pemancar cahaya.[7]

2.9 Quality of Service

Quality of service adalah pengukuran kinerja dari satu jaringan untuk memberikan layanan yang baik dengan menyediakan bandwidth dan meminimalisir jitter dan delay. Pada tugas akhir ini parameter yang akan diukur adalah throughput, delay, jitter, dan packet loss

a. *Delay, Jitter Thoughput, dan Packet loss*

Delay adalah waktu tunda yang diakibatkan oleh proses transmisi data dari pengirim ke penerima. Oleh karena itu, semakin kecil delay maka proses transmisi data semakin baik. Satuan dari *delay* adalah *second* (s). Delay transmisi dapat dicari dengan menggunakan Persamaan berikut:

Tabel 2.1 Rumus Delay

$Delay\ (sec) Tx\ =\ Received\ Time\ (s)\ -\ Sent\ Time (s)$

Jitter adalah variasi *delay* antar paket yang diakibatkan oleh panjang antrian dalam suatu pengolahan data dan *reassemble* paket-paket data di akhir pengiriman akibat kegagalan sebelumnya. Delay antrian pada *router* dan *switch* dapat menyebabkan *jitter* [6]. Semakin besar beban trafik atau nilai variasi *delay* di dalam jaringan akan menyebabkan semakin besar pula peluang terjadinya tumbukan antar paket, sehingga nilai jitter akan semakin besar dan menyebabkan nilai *QoS* semakin turun. Perhitungan *jitter* dapat dilihat seperti persamaan di bawah ini.[8]

Tabel 2.2 Rumus Jitter

$$Jitter = \frac{total\ variasi\ delay}{total\ paket\ yang\ diterima - 1}$$

Total variasi delay merupakan perbandingan *delay* dengan *delay* rata – rata atau secara statistik bisa disebut standar deviasi atau standar baku standar deviasi ini digunakan untuk memperlihatkan seberapa besar perbedaan data yang ada dibandingkan dari rata-rata data itu sendiri. Standar deviasi merupakan akar kuadrat dari varians dan menunjukan standar penyimpangan data terhadap nilai rata-ratanya. Maka total variasi delay diperoleh dengan persamaan:

Tabel 2.3 Rumus Variasi delay

$$total\ variasi\ delay = \sqrt{s^2} = \left(\frac{n\sum_{i=1}^{n} x_i{}^2 - (n\sum_{i=1}^{n} x_i)^2}{n(n-1)}\right)$$

$s2$ = varians dari delay
n = banyaknya data
$\sum_{i=1}^{n} x_i$ = Jumlah total data
$\sum_{i=1}^{n} x_i{}^2$ = Jumlah total dari data yang dikuadratkan
$(n\sum_{i=1}^{n} x_i)^2$= Jumlah total data dikuadratkan
*data yang dimaksud pada rumus merupakan nilai *delay*

Throughput, adalah *bandwidth* aktual yang terukur pada suatu ukuran waktu tertentu dalam mentransmisikan berkas. Berbeda dengan bandwidth walaupun satuannya sama *bits per second(bps)*, tapi *throughput* lebih menggambarkan *bandwidth* yang sebenarnya pada suatu

waktu dan pada kondisi dan jaringan tertentu yang digunakan untuk mengunduh suatu *file* dengan ukuran tertentu.[8]

Packet loss didefinisikan sebagai kegagalan transmisi paket data mencapai tujuannya. Kegagalan paket tersebut mencapai tujuan, dapat disebabkan oleh beberapa kemungkinkan, di antaranya yaitu: a. Terjadinya overload trafik didalam jaringan, b. Tabrakan (congestion) dalam jaringan, c. Error yang terjadi pada media fisik, d. Kegagalan yang terjadi pada sisi penerima antara lain bisa disebabkan karena overflow yang terjadi pada buffer. Di dalam implementasi jaringan IP, nilai packet loss ini diharapkan mempunyai nilai yang minimum. Secara umum biasanya terdapat pengkategorian performansi jaringan berdasarkan nilai packet loss yaitu sangat bagus, bagus, jelek, dan sedang. Paket loss dapat dihitung dengan rumus[8]

Tabel 2.4 Rumus throughput

$$\frac{\text{Packet transmited} - \text{Packet received}}{\text{Packet transmited}} \times 100\%$$

2.10 Blynk

platform untuk aplikasi OS Mobile (iOS dan Android) yang bertujuan untuk kendali module *Arduino, Raspberry Pi, ESP8266, WEMOS D1*, dan module sejenisnya melalui Internet.

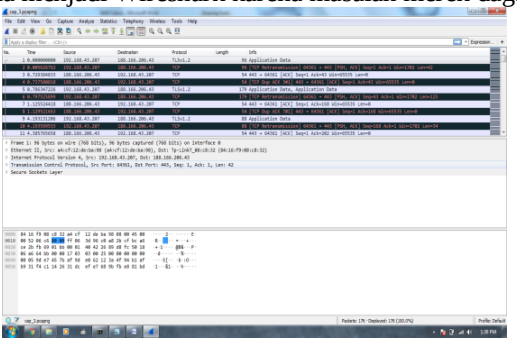
Gambar 2. 7 *Logo Blynk*

Blynk merupakan platform yang menyediakan pengendalian module iot seperti pada mikrokontroler raspberry pi dan arduino dengan memanfaatkan library blynk dapat diintegerasikan perangkat-perangkat yang dikendalikan seperti relay-relay dan sensor.[9]

2.11 Wireshark

Wireshark adalah penganalisis paket bebas dan sumber terbuka. Perangkat ini digunakan untuk pemecahan masalah jaringan, analisis,

perangkat lunak dan pengembangan protokol komunikasi, dan pendidikan. Awalnya bernama Ethereal, pada Mei 2006 proyek ini berganti nama menjadi Wireshark karena masalah merek dagang.[10]



Gambar 2. 8 *Wireshark*

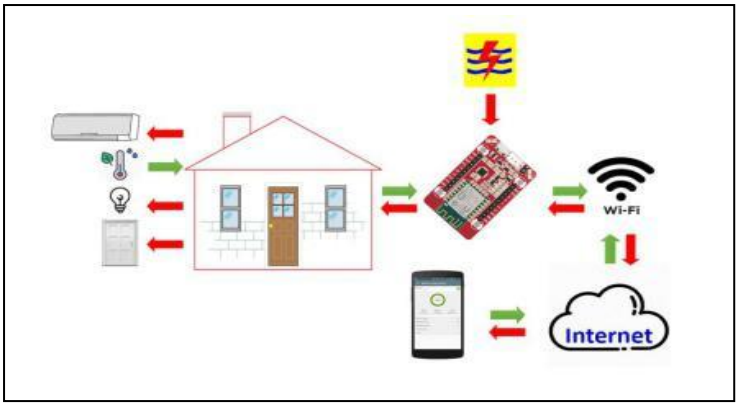
Wireshark merupakan packet analyzer yang bersifat free dan open source. Aplikasi ini digunakan untuk troubleshooting jaringan, analisa paket data dan pengembangan protokol komunikasi. Wireshark merupakan aplikasi cross-platform, menggunakan GTK+ widget toolkit pada user interface-nya dan menggunakan npcap untuk menangkap paket data.[10]

2.12 Penerapan Smarthome

Beberapa Penerapan Konsep *Smarthome* Jika dilihat, beberapa peralatan dapat digunakan dalam penerapan konsep teknologi smarthome ini. Misalnya saja, seperti lampu dan AC yang bisa menyala dan mati menyesuaikan dengan keberadaan orang di area tersebut. Beberapa AC juga bahkan dapat ‘membaca’ berapa orang yang ada di areanya, kemudian menyesuaikan suhu serta daya yang digunaka Pada hal lain, konsep ini juga dapat diterapkan di bagian gerbang depan atau pintu masuk. Nantinya, kedua hal ini akan terhubung pada jaringan internet dan dapat mendeteksi perangkat atau identitas tertentu yang Anda pasangkan.[11]

Ketika Anda akan masuk, pintu atau gerbang dapat dengan otomatis terbuka tanpa Anda harus menyentuhnya Di sisi lain, perangkat seperti CCTV atau alarm juga dapat digunakan secara smart. Kedua perangkat ini berperan besar dalam menjaga keamanan di area hunian.

Pada beberapa perangkat bahkan sudah memiliki fitur panggilan otomatis pada kantor polisi atau keamanan setempat, sehingga ketika ada aktivitas yang mengganggu dapat dengan segera ditindak lanjuti. Hambatan penerapan dalam prakteknya, penerapan teknologi *smarthome* di Indonesia bukan tanpa hambatan. Sederet hambatan mendasar dan bersifat teknis masih dijumpai, bersamaan dengan hambatan sosial dan budaya yang ada.Untuk masalah teknis sendiri, seperti diketahui bersama hingga saat ini terkadang masih ada kendala seperti aliran listrik yang tidak stabil dan kejadian mati listrik. Memang hal ini dapat disikapi dengan pemasangan sumber energi lain seperti misalnya genset, namun demikian hal ini sedikit banyak akan menghambat operasional karena mau tidak mau semua peralatan tetap harus memiliki asupan daya yang cukup. [11]



Gambar 2.10 contoh penerapan smarthome[11]

Dari sisi sosial-budaya, penggunaan jasa asisten rumah tangga juga merupakan salah satu penghambat penerapan smarthome. Pada dasarnya, smarthome akan mengandalkan otomatisasi segala perangkat yang ada di rumah sehingga benar-benar hampir menihilkan partisipasi fisik manusia dalam operasionalnya. Ketika hunian masih menggunakan jasa asisten rumah tangga, operasional smarthome secara umum tentu akan terhambat karena akan ada seorang yang tetap berada di area bangunan sehingga mungkin saja akan mengacaukan sistem yang sudah disiapkan.Jika dilihat secara keseluruhan, sebenarnya konsep ini merupakan suatu konsep yang visioner, di mana nantinya masyarakat

akan banyak menghemat waktu dan dapat meningkatkan efektivitas penggunaan sumber daya energi yang dimiliki. Meski memiliki biaya yang cukup tinggi, prospek penerapan teknologi smarthome di Indonesia dirasa cukup baik. Hal ini juga melihat pertumbuhan ekonomi di Indonesia yang cenderung progresif sehingga segmen pengguna teknologi ini akan meningkat jumlahnya.[11]

2.13 Skenario Pengujian *Smart home System* (SHS)

Di era sekarang ini banyak diantara suami dan istri memiliki karir di luar rumah, dengan kondisi pagi hari sudah berangkat ke kantor dan malam hari baru pulang ke rumah. Hal yang biasa dilakukan adalah sudah menyalakan lampu dari pagi hari, karena takut saat malam hari keadaan rumah gelap karena pemilik rumah belum sampai dirumah. Semua ini akan menyebabkan konsumsi energi yang besar[12]

Beberapa cara memperkecil penggunaan daya pada lampu salah satunya adalah dengan pemakaian lampu HID (natrium tekanan tinggi dan Halid logam), sedangkan untuk lampu natrium tekanan rendah (low pressure sodium lamps) walaupun memiliki efektifitas tinggi, tapi jenis lampu ini menghasilkan pancaran warna yang kurang baik. Untuk pemakaian lampu pijar juga kurang direkomendasikan dalam pemakaian, apalagi jika penggunaan dilakukan selama 24 jam, kelemahan dari lampu pijar ini adalah memiliki watt yang tinggi[12]

Untuk menyelesaikan masalah tersebut, salah satu teknologi yang bisa digunakan adalah dengan memanfaatkan mikrokontroler arduino. Arduino ini merupakan suatu alat yang cerdas seperti komputer, tapi bentuk yang dimiliki kecil dan harganya yang murah. Arduino Uno terdiri dari 14 pin digital masukan/keluaran (6 pin dapat digunakan sebagai keluaran PWM), 6 pin masukan analog, sebuah resonator keramik 16 MHz, sebuah port USB, sebuah power jack, sebuah ICSP header, dan sebuah tombol reset. Dengan spesifikasi yang dimiliki, arduino uno dapat dimanfaatkan untuk mengendalikan beberapa beban. Arduino dapat diaplikasikan pada sistem smart home. Smart home atau rumah pintar merupakan suatu teknologi yang terfokus pada sistem pengendalian peralatan rumah tangga. Dalam perkembangan teknologi smart home ini,

awal mulai sistem dioperasikan dengan memanfaatkan mikroprosesor yang kemudian dengan berkembangnya teknologi, memanfaatkan tenaga listrik menggunakan gelombang radio untuk menghantarkan data. Dimana Sistem rumah cerdas (smart home) adalah sistem aplikasi yang merupakan gabungan antara teknologi dan pelayanan yang dikhususkan pada lingkungan rumah dengan fungsi tertentu yang bertujuan meningkatkan efisiensi, kenyamanan dan keamanan penghuninya‖[12]

Dalam penelitian ini, arduino digunakan untuk otak dalam melakukan pengendalian smart home otomatis. Smart home pada penelitian ini di desain untuk menyalakan dan mematikan lampu secara otomatis dengan memanfaatkan sensor cahaya. Sensor cahaya yang digunakan adalah LDR. LDR merupakan alat yang digunakan untuk mengubah besaran cahaya menjadi besaran listrik. LDR akan diposisikan di luar rumah, yang selanjutnya hasil pembacaan sensor ini akan terintegrasi dengan arduino. Data sensor yang terbaca di Arduino akan digunakan untuk mengaktifkan ataupun memutuskan relay yang dihubungkan ke beban, dimana beban dalam penelitian ini adalah lampu.[12]

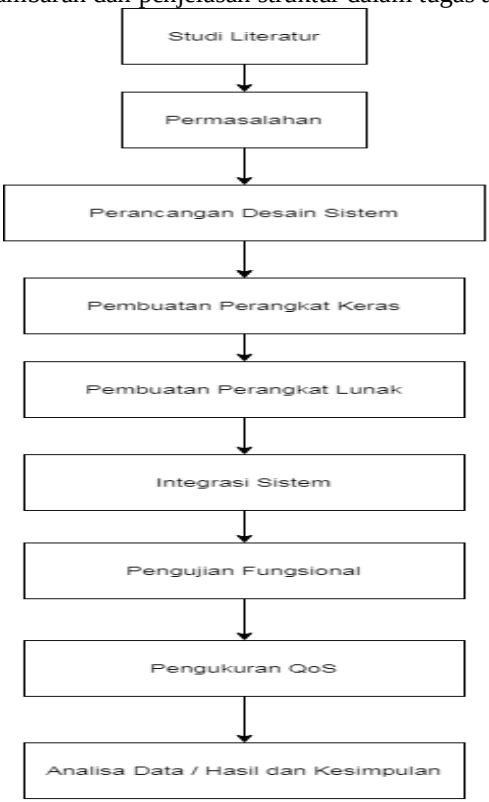
Dengan adanya sistem efisiensi energi smart home berbasis arduino uno ini, dapat bermanfaat bagi masyarakat dalam mengendalikan konsumsi energi dari penggunaan peralatan elektronik yang ada di rumah. Dapat menekan konsumsi daya listrik pada saat rumah kosong sehingga pemilik dalam membayar tagihan listrik juga akan berkuran. Resiko terjadinya konsleting listrik pada rumah kosong juga akan mengalami penurunan.[12]

BAB III

METODOLOGI PENELITIAN

3.1 Diagram Alur Penelitian

Hal yang terpenting harus diperhatikan dalam membuat sesuatu perancangan yaitu alur yang akan ditempuh selama proses penelitian berlangsung dalam diagram alur ini akan mencankup langkah-langkah yang dilalui penulis, keseluruhan diagranalur pada Gambar 3.1 merupakan gambaran dan penjelasan struktur dalam tugas akhir ini.



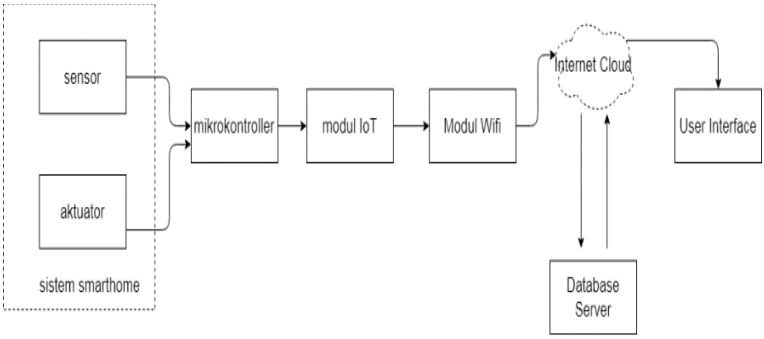
Gambar 3. 1 *Diagram Alur*

3.2 Konsep Rumah Pintar Berbasis *Internet of Thing* (IoT)

Smart Home atau Rumah Pintar merupakan bagian dari inovasi *Internet of Things*, di mana semua benda atau perabotan sehari-hari yang akrab dengan kehidupan masyarakat 'dipintarkan' karena integrasi teknologi dalam bentuk chip serba bisanya. Dalam hal Smart Home, *Internet of Things* akan hadir dalam rupa barang-barang yang biasa ditemui di rumah orang kebanyakan.

Smart Home hadir untuk memudahkan para penghuni rumah dalam mengatur segala hal yang berhubungan dengan kenyamanan diri sebagai penghuni rumah, mulai dari soal keamanan hingga soal akses perabotan yang dibuat lebih interaktif dan bisa 'dikontrol' melalui satu alat saja, yakni aplikasi pada smartphone atau perangkat lainnya.

Smart Home di Indonesia saat ini masih dalam tahap perkenalan dan pembiasaan ke masyarakat. Memasuki ranah komunitas terkecil dan terintim yaitu keluarga pastinya bukanlah hal yang mudah. Adanya gap umur dan budaya dalam menggunakan teknologi dan internet juga bisa mempersulit sistem *Smart Home* untuk dapat dengan mudah diterima dan digunakan oleh keluarga-keluarga di Indonesia.



Gambar 3. 2 Diagram Desian sistem *Smarthome*

Gambar diatas menjelaskan diagram dari sensor dan *actuator* yang menggerakkan sensor di program di mikrokontroller selanjutnya di proses menjadi wifi melalaui modul iot setelah itu disambungkan ke internet cloud untuk mengirim database dan akhirnya sampai ke User Interface.

Sistem yang serba otomatis bisa sangat membingungkan para pengguna teknologi yang terbagi antara yang berpengalaman (generasi milenial yang lebih muda) serta yang belum pernah membiasakan diri dengan internet yang serba praktis (pada rentang usia yang lebih tua). Hal ini, tentunya akan menjadi tantangan tersendiri bagi teknologi *Smart Home* di Indonesia untuk mulai diaktivasikan di dalam target marketnya: para keluarga.Peralatan rumah pintar datang dengan keterampilan belajar mandiri di mana mereka dapat mempelajari jadwal pemilik rumah dan menyesuaikan sesuai kebutuhan. Rumah pintar yang diaktifkan dengan kontrol pencahayaan memungkinkan pemilik rumah untuk mengurangi penggunaan listrik dan karenanya mendapat manfaat dari penghematan biaya terkait energi. Beberapa sistem otomasi rumah memperingatkan pemilik rumah jika ada gerakan yang terdeteksi di rumah ketika sedang pergi, dan beberapa dapat memanggil pemadam kebakaran jika terjadi situasi yang akan terjadi. Setelah peralatan pintar ini terhubung, kami memiliki contoh apa yang kami sebut teknologi *Internet of Things (IoT).*

Rumah pintar dapat menampilkan sistem yang nirkabel atau bawaan. Sistem nirkabel ramah biaya dan lebih mudah untuk dipasang sementara sistem berkabel dianggap lebih andal dan biasanya lebih sulit diretas. Meskipun sistem kabel juga lebih mahal daripada opsi nirkabel, menginstal sistem kabel dapat meningkatkan nilai jual kembali rumah. Menginstal otomatisasi rumah nirkabel dengan fitur-fitur seperti pencahayaan pintar, kontrol iklim, dan keamanan dapat menelan biaya beberapa ribu dolar. Sementara itu, kemewahan dan pilihan hardware dapat menelan biaya puluhan ribu dolar bagi pemilik rumah.

3.2.1Persyaratan System

Untuk dapat menjalankan sistem smarthome berbasis IoT menggunakan *Blynk* yang telah dirancang diperlukan modul-modul sebagai berikut.

1 Arduino Nano
2 ESP826
3 Akses Internet
4 Akun Blynk yang telah memiliki token dan aplikasi Blynk
5 Sensor dan relay

Untuk menggunakan sistem ini perangkat diatas harus diprogram terlebih dahulu dengan listing program yang telah dibuat, token yang telah diperoleh saat melakukan pendaftaran akun Blynk

dimasukan dalam program yang terpasang pada ESP8266, untuk menggunakan token ini harus memiliki library Blynk pada bahasa arduino.

3.3 Pembuatan Prototipe *Smart Home System* (SHS)

Dalam tugas akhir ini akan dilakukan pembuatan prototipe Smart Home System yang akan membantu pada tahap selamjutnya.akan membahas permasalahan penentuan fasilitas objek cerdas, penentuan kode dan cara membuat Prototipe *Smart Home System*(SHS).

3.3.1 Perancangan Protoype

Barikut ini merupakan bentuk perancangan prototype dari tugas akhir, dalam hal ini prototype akan mengambil acuan dari desain rumah nyata meliputi penempatan sensor pada kamar mandi, teras, ruang tamu dan kamar tidur.

Gambar 3. 3 *Identifikasi Permasalahan*

Tujuan penelitian dalam Tugas Akhir ini adalah membuat rancangan rumah cerdas/ *Smart Home System* (SHS) yang bisa dimonitor dan dikendalikan dari jarak jauh dengan menerapkan teknologi berkonsep *Internet of Thing (*IoT), Selanjutnya rancangan tersebut akan dibuat dalam bentuk *prototype* sitstem yang disebut *Smart home System* (SHS).

3.3.2 Penentuan Fasilitas Objek cerdas

Tabel 3. 1 Penentuan Fasilitas Objek Cerdas

No	Lokasi	Objek	Kode
1	Teras	Lampu	L1
2	Ruang tamu	Lampu	L 2
3	Ruang keluarga	Kipas Angin	K 1
4	Ruang tidur 1	Lampu3	L 3
5	Ruang tidur 2	Lampu4	L 4
6	Kamar mandi	Lampu 5	L5
7	Dapur	Kompor	K 2

Pada table diatas menjelaskan penentuan fasilitas objek cerdas yang akan digunakan di prototipe *Smart Home System* (SHS) yang akan menjelaskan mengenai lokasi, objek, dan kode yang akan digunakan dalam peneltian tugas akhir saya, mengenai objek-objek yang dikendalikan melalui *system smarthome*, antara lain lampu teras dengan nama objek lampu dan kode L1, ruang tamu dengan objek lampu kode L2 Ruang keluarga objek kipas angin dengan kode K1, Ruang tidur 1 objek lampu3 dengan kode L3, Ruang Tidur 2 dengan objek Lampu 4 kode L4, kamar mandi dengan objek Lampu 5 dengan kode L5 sampai dengan lokasi dapur dengan objek kompor berkode K2.

3.3.3 Penentuan kebutuhan objek cerdas

Dalam tahap ini akan menjelaskan objek mana saja yang akan dicerdaskan dalam Prototipe *Smart home System* (SHS) yang akan dibahas dalam penelitian Tugas akhir ini antara lain lokasi, objek, kebutuhan di prototipe yang akan dibuat, Berikut tabel penentuan kebutuhan objek:

Tabel 3. 2 Penentuan Objek Cerdas

No	Lokasi	Objek	Kode	Kebutuhan
1	Teras	Lampu	L 1	Menyalakan dan mematikan lampu teras
2	Ruang tamu	Lampu	L 2	Menyalakan dan mematikan Ruang tamu
3	Ruang keluarga	Kipas Angin	K 1	Menyalakan kipas angin bila panas
4	Ruang tidur 1	Lampu3	L3	Menyalakan dan mematikan ruang tidur 1
5	Ruang tidur 2	Lampu4	L4	Menyalakan dan mematikan ruang tidur 2
6	Kamar mandi	Lampu 5	L 5	Menyalakan dan mematikan kamar mandi
7	Dapur	Kompor	K2	Berbunyi alarm bila ada kebakaran

3.3.4 Penentuan Kebutuhan Objek Cerdas

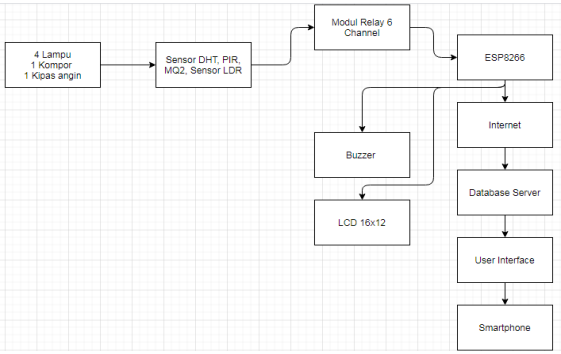
Berikut akan dijelaskan penentuan modul/*divais* untuk mencerdaskan objek.

Tabel 3. 3 Penentuan Kebutuhan Objek

No	Lokasi	Objek	Kode	Modul/*divais*
1	Teras	Lampu	L 1	Lampu+LDR
2	Ruang tamu	Lampu	L 2	Lampu
3	Ruang keluarga	Kipas Angin	K 1	Kipas angin+dht11
4	Ruang tidur 1	Lampu3	L 3	Lampu
5	Ruang tidur 2	Lampu4	L 4	Lampu
6	Kamar mandi	Lampu 5	L 5	Lampu+PIR
7	Dapur	Kompor	K 2	Sensor gas

Pada tabel diatas menjelaskan mengenai konfigurasi sensor dan kode objek, sensor PIR digunakan pada lampu kamar mandi, sensor dht11 atau humidity sensor digunakan pada kipas angin dan sensor LDR light dipendent resistor digunakan untuk lampu teras.

3.3.5 Rancangan Sistem Rumah Cerdas



Gambar 3. 4 Rancangan Sistem Rumah Cerdas

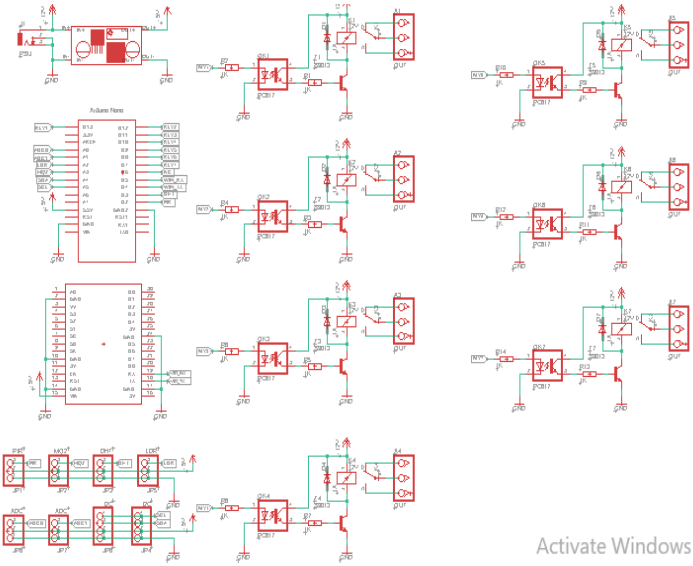
Pada gambar diatas merupakan bentuk perancangan sistem smarthome yang terdapat pengendalian pada kipas dan lampu-lampu dirumah, kemudian dilanjutkan dengan penggunaan sensor-sensor.

3.3.6 Pembuatan rancanganan sistem *Hardware*

Pembuatan rancangan sistem dibawah ini akan meliputi penyusunan skematik rangkaian dan pembuatan PCB. Untuk membuat sistem *Hardware*

1. Skematik Rangkaian

Berikut merupakan gambaran skematik dari rangkaian yang akan dibuat, pada bagian ini akan menjelaskan keseluruhan rangkaian dan modul-modul.

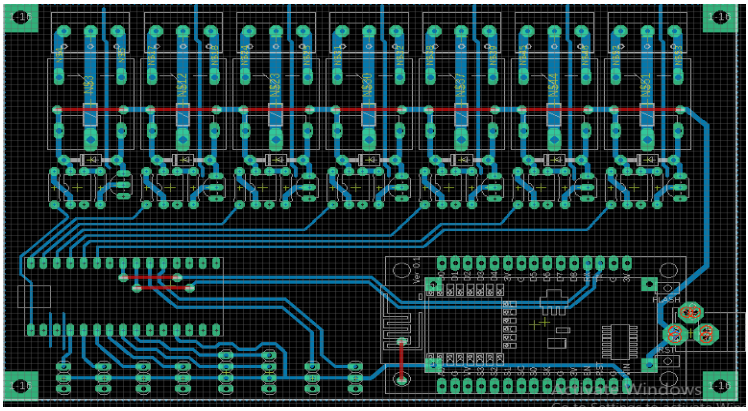


Gambar 3. 5 Skematik Rangkaian

Pada gambar 3.4 dijelaskan bagaimana penghubung ditiap komponen yang digunakan, komunikasi serial digunakan pada modul sementara *relay* dikontrol langsung dengan mikrokontroler. Skema rangkaian pada gambar 3.4 sementara untuk catu daya diperoleh dari *power supply* sebesar 12v.

2. Pembuatan PCB

Pada gambar 3.5 merupakan lay out keseluruhan PCB, Lay out ini dibuat menggunakan software eagle, dengan tujuan untuk proses pencetakanPCB, pada lay out ini menjelaskan konfigurasi relay ke mikrokontroler Arduino



Gambar 3. 6 Lay Out PCB

3. Realisasi perangkat keras

Merupakan bentuk modul-modul yang telah dirangKai dan dipasang pada sistem *smarthome*, alasan peletakan modul *relay* yang menjadi satu papan dengan ESP8266 dan Arduino adalah untuk membentuk suatu struktur yang praktis dan ringkas, untuk penghubung ke modul-modul sensor, kipas, *LCD buzzer* dan lampu menggunakan kabel, peletakan power supply juga menjadi satu bagian dengan papan ini dengan menggunakan catu daya *input* 220 *VAC* dan output *power supply* sebesar 12 V

3.3.7 Pembuatan system perangkat lunak

Pada tahap ini membuat sistem menggunakan Bahasa C, Untuk memogram mikrokontroler dan pemograman modul IoT sedangkan untuk *User Interface* digunakan *Cloud Platform* yang bernama BLYNK.

1. Pemograman Mikrokontroler

Di tahap ini menjelaskan mengenai program mikrokontroler Arduino yang bertugas menerima perintah dari esp8266 berupa action untuk mengaktifkan objek tertentu. Objek yang dimaksud dalam penelitian ini antara lain Teras,Ruang tamu, Kamar tidur1, Kamar tidur 2, Kamar mandi, dan dapur.

```
suhu        = data[1].toInt();
asap        = data[2].toInt();
cahaya      = data[3].toInt();
gerak       = data[4].toInt();
lampu_1     = data[5].toInt();
lampu_2     = data[6].toInt();
lampu_3     = data[7].toInt();
lampu_4     = data[8].toInt();
lampu_5     = data[9].toInt();
kompor      = data[10].toInt();
kipas       = data[11].toInt();
batas_suhu  = data[12].toInt();
batas_asap  = data[13].toInt();
batas_cahaya = data[14].toInt();
otomatis_dht = data[15].toInt();
otomatis_ldr = data[16].toInt();
otomatis_pir = data[17].toInt();
otomatis_mq2 = data[18].toInt();

//Serial.println("-----------------------------");
//Serial.println("SUHU:" + String(suhu) + ", ASAP:" + String(asap) + ", CAHAYA:" + String(cahaya) + ", NOTIFIKASI:" + String(notifikasi));
//Serial.println("LAMPU 1: " + String(lampu_1) + ", LAMPU 2: " + String(lampu_2) + ", LAMPU 3: " + String(lampu_3));
//Serial.println("LAMPU 4: " + String(lampu_4) + ", LAMPU 5: " + String(lampu_5) + ", KOMPOR: " + String(kompor) + ", KIPAS: " + String(kipas));
//Serial.println("BATAS SUHU: " + String(batas_suhu) + ", BATAS ASAP: " + String(batas_asap) + ", BATAS CAHAYA: " + String(batas_cahaya));
//Serial.println("OTOMATIS DHT: " + String(otomatis_dht) + ", OTOMATIS LDR: " + String(otomatis_ldr));
//Serial.println("OTOMATIS PIR: " + String(otomatis_pir) + ", OTOMATIS MQ2: " + String(otomatis_mq2));
```

Gambar 3.7 Source Code Program

Pada gambar diatas merupakan bagian yang menunjukan source code program yang diupload pada arduino yang digunakan untuk mengendalikan relay dan sensor-sensor, serta untuk transmisi data dari sensor ke aplikasi Blynk yang digunakan oleh *user*.

2. Pemograman Modul IoT

Dalam bagian ini menjelaskan mengenai source code pada modul ESP 8266 menggunakan Bahasa C ++ seperti dalam Arduino, pada program berikut menggunakan library esp8266 wifi.h, blynksimpleesp8266_ssl.h. SSID yang digunakan alc, dengan password alcalcalc, agar dapat terhubung ke cloud platform (blynk) maka dibutuhkan token autentikasi seperti pada program yang tertera dibawah:

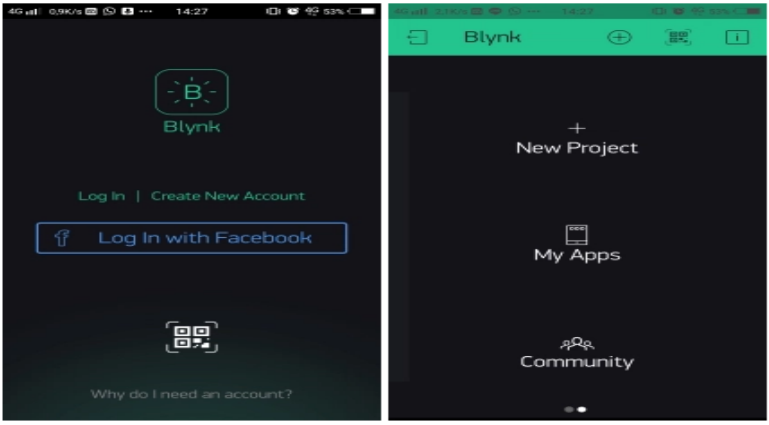
```
// ------------------------------------------------
//#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266_SSL.h>

// ------------------------------------------------
char ssid[] = "alc";
char pass[] = "alcalcalc";
char auth[] = "UCsOEWd3zkvrRe7rJODN1eVGowFxp35o";
```

Gambar 3.8 Source Code Program ESP8266

3. Pembuatan Blynk

Proses instalasi aplikasi Blynk ditunjukan pada gambar 3.8 yang digunakan untuk mengendalikan system *smarthome* dilakukan dengan membuka playsotre pada smarthpone android pengguna kemudian masukan keyword blynk pada kolom pencarian kemudian pilih aplikasi dengan nama, kemudian ikuti petunjuk yang diberikan, berikut merupakan langkah-langkah yang harus dilakukan oleh pengguna dalam mengoperasikan aplikasi ini,

1. Menginstall aplikasi pada *smartphone android*.
2. Melakukan inisialisasi akun untuk memperoleh *token*.
3. Memasukan token pada ESP8266 melalui pemrograman Arduino
4. Untuk mengaplikasikan token ini harus terdapat library Blynk
5. Dibutuhkan editor Arduino IDE
6. Untuk menguji koneksi token Blynk dibutuhkan koneksi internet



Gambar 3. 7 *UI Blynk*

Pada gambar 3.7 merupakan gambaran aplikasi Blynk yang telah terinstall di smartphone user system smarthome, dengan menginstall aplikasi ini user dapat memantau aktifitas dan status sensor dan objek yang dikendalikan, sensor dan objek yang dikendalikan telah

dihubungkan ke mikrokontroler arduino dan library ESP8266 telah terinstall.

3.3.8 Integrasi Sistem

Pada bagian ini akan membahas mengenai integerasi modul-modul yang telah disidakan kedalam sistem smarthome, dalam hal ini integerasi akan berfokus pada perangkat *hardware*.

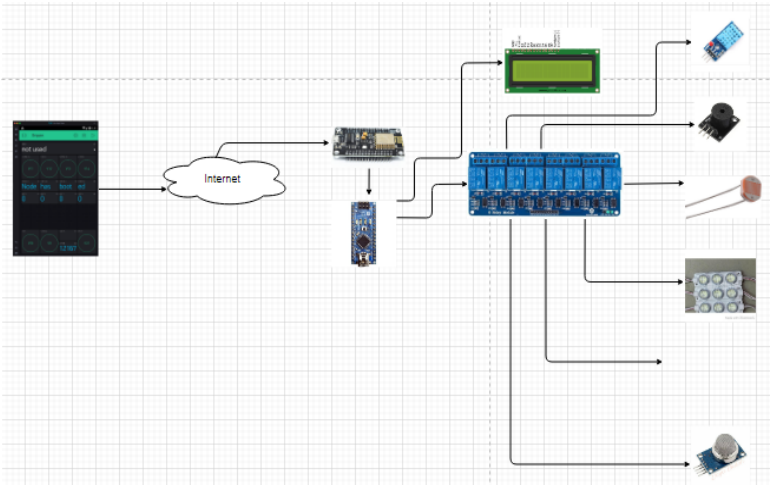
Gambar 3. 8 *Intergrasi Sistem*

Pada tahap ini merupakan penyatuan kesuluruhan modul dan sensor kedalam suatu sistem smart home, dilanjutkan dengan mengaplod program pada Arduino dan esp8266, kemudian proses instalasi blynk pada opengguna SHS. Dari proses ini dapat diketahui apakah kesuluruhan modul berinteraksi dengan baik sehingga dapat dilanjutkan proses pengujian.

1. Diagram integrasi

Dalam tugas akhir ini akan dibangun sistem *smarthome* dengan menggunakan modul *ESP8266, Arduino,* Sensor *Gas Mq-2, Humidity* Sensor dan Sensor *PIR* menggunakan pengendali berupa perangkat *smartphone Android* dan aplikasi *Blynk.* Alur kinerja alat adalah sebagai berikut :

1) *Module ESP8266* bertugas meghubungkan Arduino ke jaringan internet secara *wireless* melalui suatu akses *point hotspot* internet.
2) *Alarm* gas akan dipicu oleh pembacaan sensor *MQ-2* dan bereaksi membunyikan *buzzer*.
3) Mikrokontroler *Arduino Nano* digunakan untuk membaca tiap *input* dari sensor-sensor.
4) Sensor *Humidity* akan membaca perubahan suhu dan memicu kipas angin.



Gambar 3. 9 Hubungan Sensor

Pada Gambar 3.10 menjelaskan hubungan antara sensor-sensor dan perangkat apa saja yang dikendalikan, selain perangkat yang dipicu oleh sensor terdapat lampu yang dapat dikendalikan melalui komunikasi data pada jaringan internet dengan menggunakan aplikasi pada smartphone android menggunakan Blynk, dengan menggunakan aplikasi ini user juga dapat mengaktifkan dan menonaktifkan sensor-sensor serta merubah parameter aksi dari tiap sensor, berikut merupakan parameter-parameter yang dapat diubah dengan menggunakan aplikasi Blynk

2. Pengujian Fungsional

Pada perangkat ini terdapat *modul ESP8266*, *relay*, sensor *MQ-2*, sensor *Passive Infrared*, sensor *humidity* dan sensor cahaya sebagai

perangkat *input sensing*, sementara pada *output* terdapat lampu, *buzzer* dan *LCD*. Pengujian akan dilakukan dengan mengikuti langkah-langkah seperti pada prosedur berikut:

a. Pengujian Koneksi Modul *IoT - WiFi*

menggunakan mode *tethering* pada *smartphone* dengan penggunaan *SSID* yang telah terinput di dalam program *Arduino*.

Tabel 3. 4 Pengujian Koneksi

No	Keterangan Percobaan	Keterangan
1	ID benar password benar	pengujian apakah sistem *smarthome* secara otomatis terhubung ke *smartphone* yang melakukan *tethering* dengan *id* dan *password* tersebut
2	ID benar password salah	saat *password* diubah secara otomatis perang pengujian apakah sistem *smarthome* secara otomatis terhubung ke *smartphone* yang melakukan *tethering* dengan *id* dan *password* tersebut kat terputus
3	ID salah password benar	pengujian apakah sistem *smarthome* secara otomatis terhubung ke *smartphone* yang melakukan *tethering* dengan *id* dan *password* tersebut
4	ID salah password salah	pengujian apakah sistem *smarthome* secara otomatis terhubung ke *smartphone* yang melakukan *tethering* dengan *id* dan *password* tersebut

b. Pengujian Monitoring Objek Cerdas

Pada pengujian ini bertujuan untuk mengetahui kinerja objek cerdas terhadap perintah eksekusi yang diberikan pengguna melalui blynk.

c. **Pengujian Aktifasi Lampu**
Pada pengujian ini akan menggunakan antarmuka aplikasi *Blynk*, pada bagian ini perintah yang dikirim hanya eksekusi mengaktifkan dan mematikan lampu.

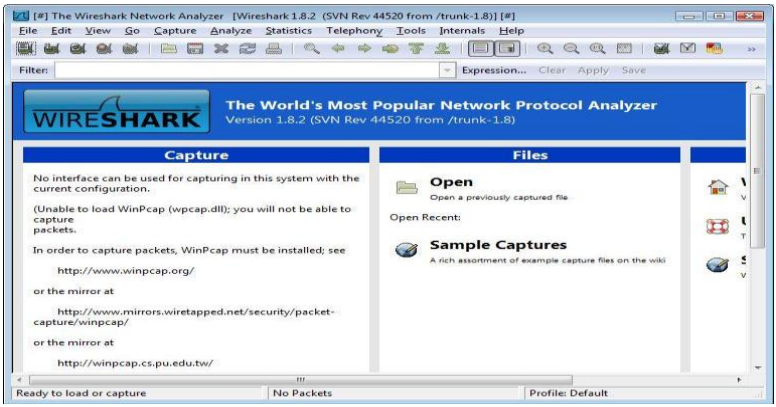
d. **Pengujian Respond Sensor**
Pengujian pengiriman perintah ke sistem *smarthome* menggunakan antarmuka aplikasi *Blynk*, pada bagian ini perintah yang dikirim eksekusi mengaktifkan sensor dan memengkonfigurasi penggunaan sensor *Humidity*, sensor *Passive Infrared*, sensor *MQ-2*, sensor cahaya *LDR*.

3.3.9 Pengujian Performnansi (*Quality of Service)*

Dalam tahap ini akan dilakukan pengujian performnansi pada Prototype Smart Home System menggunakan Software Wireshark, dengan parameter Delay, Throughput, Jitter dan Packet loss. Tujuan pengujian ini adalah untuk mengetahui kapasistas dan kemampuan traffic jaringan dengan mengamati lalu lintas paket data yang berlangsung.

1. Software Tools

Software yang digunakan pada penelitian ini adalah Wireshark. Wireshark merupakan *packet analyzer* yang bersifat *free* dan *open source*. Aplikasi ini digunakan untuk *troubleshooting* jaringan, analisa paket data dan pengembangan protokol komunikasi. Wireshark merupakan aplikasi *cross-platform*, menggunakan GTK+ *widget toolkit* pada *user interface*-nya dan menggunakan npcap untuk menangkap paket data. Wireshark yang digunakan dalam penelitian ini adalah yang berjalan dalam platform linux hal ini memungkinkan untuk melakukan packet sniffing dalam suatu jaringan, dengan ini paket data yang tercapture antara perangkat saling berkomunikasi dalam jaringan.



Gambar 3.10 Wireshark

2. Parameter yang diukur (*Delay, Througput, Jitter, Packet loss)*

Delay adalah waktu tunda yang diakibatkan oleh proses transmisi data dari pengirim ke penerima. Oleh karena itu, semakin kecil delay maka proses transmisi data semakin baik. Satuan dari *delay* adalah *second* (s).

Jitter adalah variasi *delay* antar paket yang diakibatkan oleh panjang antrian dalam suatu pengolahan data dan *reassemble* paket-paket data di akhir pengiriman akibat kegagalan sebelumnya. Delay antrian pada *router* dan *switch* dapat menyebabkan *jitter* [6]. Semakin besar beban trafik atau nilai variasi *delay* di dalam jaringan akan menyebabkan semakin besar pula peluang terjadinya tumbukan antar paket, sehingga nilai jitter akan semakin besar dan menyebabkan nilai *QoS* semakin turun.

Throughput, adalah *bandwidth* aktual yang terukur pada suatu ukuran waktu tertentu dalam mentransmisikan berkas. Berbeda dengan bandwidth walaupun satuannya sama *bits per second(bps)*, tapi *throughput* lebih menggambarkan *bandwidth* yang sebenarnya pada suatu waktu dan pada kondisi dan jaringan tertentu yang digunakan untuk mengunduh suatu *file* dengan ukuran tertentu

3.4.3 Skenario pengkuran

Pada bagian ini merupakan penjelasan mengenai langkah-langkah pengujian, dimulai dengan melakukan pengujian berdasarkan jarak, pengukuran jumlah objek dan kesimpulan. Dengan mengikuti prosedur ini dapat diperoleh hasil pengujian dan kesimpulan penelitian.

1. Pengaruh lama pengukuran

Dalam pengukuran ini dilakukan selama 5 menit pada kurun waktu yang berbeda-beda seperti pada pagi hari,siang hari, dan malam hari. Tujuannya adalah untuk mengetahui perbedaan traffic yang terjadi dengan dugaan traffic paling berat terjadi pada siang hari karena pengguna jaringan sedang banyak aktif dalam waktu ini

2. Pengaruh jarak pengukuran

Dalam pengukuran ini dilaukakan dalam berbagai variasai jawak antara lain 1 meter,1 km, 10 km dalam pengukuran ini digunakan waktu selama 5 menit pada setiap jaraknya. Tujuannya dalah untuk mengetahui apakah terjadi perbedaan traffic berdasarkan pada jarak

3. Pengurah jumlah objek

Pada pengujian ini dilakukan pemberian perintah mengaktifkan objek secara berurutan dimulai dari objek1hingga objek 7 tujuannya adalah untuk mengetahui adanya perbedaan antara eksekusi masing-masing objek. Pengukuran dilakukan 2 menit setiap objek

3.5 Standar QoS ETSI TIPHON

Standar ETSI TIPHON (*Telecommunication and Internet Protocol Harmonization over Network*) merupakan standar yang dipakai untuk mengukur Quality of Service(QoS). Parameter-parameter QoS terdiri atas (TIPHON, 1999):

Throughput yaitu kecepatan (*rate*) transfer data efektif, yang diukur dalam bps (*bit per second*). Kategori throughput ditampilkan pada Tabel 3.5

Tabel 3.5 Peringkat dan kriteria *packet loss*(TIPHON, 1999)

Nilai	Presentase (%)	Indeks
4	100	Sangat Bagus
3	74	Bagus
2	50	Kurang Bagus
1	<25	Jelek

Packet loss merupakan suatu parameter yang menggambarkan suatu kondisi yang menunjukkan jumlah total paket yang hilang, dapat terjadi karena collision dan congestion pada jaringan. Tabel 3.6 menunjukkan peringkat dan kriteria packet loss.

Tabel 3. 5 Kriteria Delay

Nilai	Presentase (%)	Indeks
4	100	Sangat Bagus
3	74	Bagus
2	50	Kurang Bagus
1	<25	Jelek

Delay adalah waktu yang dibutuhkan paket untuk mencapai tujuan, karena adanya antrian, atau mengambil rute yang lain untuk menghindari kemacetan.Peringkat dan kriteria delay diperlihatkan pada Tabel 3.7

Tabel 3. 6 *Delay*

Nilai	Besar Delay (ms)	Indeks
4	<150	Sangat Bagus
3	150 – 300	Bagus
2	300 – 450	Kurang Bagus
1	>450	Jelek

Jitter, diakibatkan oleh variasi-variasi dalam panjang antrian, dalam waktu pengolahan data, dan juga dalam waktu penghimpunan ulang paket-paket di akhir perjalanan jitter. Peringkat dan kriteria jitter diperlihatkan pada Tabel 3.8

Tabel 3. 7 Jitter

Nilai	Besar Jitter (ms)	Indeks
4	0	Sangat Bagus
3	0-75	Bagus
2	75-125	Kurang Bagus
1	125-225	Jelek

Presentase dari Nilai QoS, QoS mengacu pada kemampuan jaringan untuk menyediakan layanan yang lebih baik pada trafik jaringan tertentu QoS menawarkan kemampuan untuk mendefinisikan atribut-atribut layanan jaringan yang disediakan, baik secara kualitatif maupun kuantitatif. Pada Tabel 3.9 merupakan nilai presentase dari QoS

BAB IV

HASIL PENGUJIAN DAN PEMBAHASAN

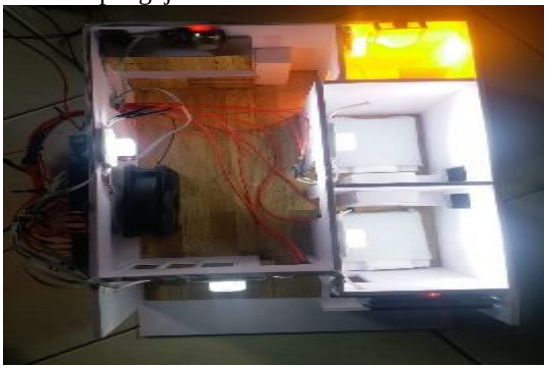
Pada bab ini berisi tentang pembahasan mengenai pengujian alat yang telah dirangkai sesuai dengan perancangan pada bab sebelumnya, pada proses pengujian ini akan dilakukan serangkaian uji coba pada setiap fungsi dan fitur yang telah disediakan, dengan pengujian ini dapat diketahui hasil perancangan dan *error* yang mungkin terjadi selama pengujian, sehingga dapat diperoleh suatu kesimpulan mengenai penelitian yang telah dilakukan.

4.1 Hasil Implementasi

Hasil perancangan yang telah dilakukan adalah sebagai berikut, dijelaskan dari hasil perancangan pada modul sensor yang terpasang pada perangkat *smarhome* aplikasi *Blynk* yang terinstall pada *smartphone android*. Dengan mengamati hasil perancangan yang telah dilakukan maka dapat ditentukan prosedur pengujian yang perlu dilakukan guna memperoleh suatu evaluasi penelitian.

4.1.1 Hasil Implementasi Hardware

Pada bagian ini akan menjelaskan mengenai hasil implementasi perangkat-perangkat keras yang digunakan dalam penelitian ini, dengan tercapaianya implementasi ini menandakan bahwa penelitian ini telah siap memasuki fase pengujian.



Gambar 4. 1 Hasil Implementasi

Pada gambar diatas merupakan bentuk keseluruhan perangkat yang digunakan dan telah disatukan dalam wadah yang mensimulasikan penggunaan pada kondisi nyata, dengan penempatan sensor-sensor seperti sensor LDR di halaman depan, sensor PIR di kamar mandi, sensor DHT di ruang utama, sensor dan sensor MQ2 di dapur untuk mendeteksi kebocoran gas LPG.

4.2 Hasil Implementasi Softraware

Pada bagian ini akan menjelaskan mengenai hasil implementasi pada bagian software meliputi bentuk antar muka Blynk, contoh program pada arduino dan ESP8266.

Gambar 4. 2 Aplikasi Blynk

Pada gambar diatas merupakan bentuk user interface pada Blynk yang telah terinstall dismartphone android, pada user interface tersebut terdapat bagian yang mengindikasikan keaktifan sensor, tombol untuk menyalakan suatu objek, dan setting threshold pada sensor.

```
// ------------------------------------------------
//#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266_SSL.h>

// ------------------------------------------------
char ssid[] = "alc";
char pass[] = "alcalcalc";
char auth[] = "UCsOEWd3zkvrRe7rJODN1eVGowFxp35o";
```

Gambar 4. 3 *Source Code Pada Modul IoT*

Pada Gambar 4.3 merupakan penerapan source code yang ter*upload* pada sistem *smarthome*, pada bagian ini terdapat mikrokontroler *Arduino* dan modul *WiFi ESP8266* berfungsi menerima perintah dari aplikasi *Blynk* yang menggunakan *protocol Tcp*. Pada antar muka aplikasi *Blynk* terdapat tombol kontrol untuk mengaktifkan dan mengkonfigurasi sensor, saat tombol kontrol diaktifkan sebagai contoh mengaktifkan lampu dengan label yang tertera *LED 1* maka setiap aksi yg dieksekusi akan merubah warna tombol sebagai indikasi perintah telah dieksekusi, dengan berubahnya kondisi ini maka *packet tcp* berupa perintah akan dikirim ke *Arduino* melalui jaringan internet dan pada proses eksekusi perintah ini akan dilakukan monitoring paket yang terkirim menggunakan *Wireshark* untuk analisis *QoS*.

```
suhu        = data[1].toInt();
asap        = data[2].toInt();
cahaya      = data[3].toInt();
gerak       = data[4].toInt();
lampu_1     = data[5].toInt();
lampu_2     = data[6].toInt();
lampu_3     = data[7].toInt();
lampu_4     = data[8].toInt();
lampu_5     = data[9].toInt();
kompor      = data[10].toInt();
kipas       = data[11].toInt();
batas_suhu  = data[12].toInt();
batas_asap  = data[13].toInt();
batas_cahaya = data[14].toInt();
otomatis_dht = data[15].toInt();
otomatis_ldr = data[16].toInt();
otomatis_pir = data[17].toInt();
otomatis_mq2 = data[18].toInt();

//Serial.println("-----------------------------");
//Serial.println("SUHU:" + String(suhu) + ", ASAP:" + String(asap) + ", CAHAYA:" + String(cahaya) + ", NOTIFIKASI:" + String(notifikasi));
//Serial.println("LAMPU 1: " + String(lampu_1) + ", LAMPU 2: " + String(lampu_2) + ", LAMPU 3: " + String(lampu_3));
//Serial.println("LAMPU 4: " + String(lampu_4) + ", LAMPU 5: " + String(lampu_5) + ", KOMPOR: " + String(kompor) + ", KIPAS: " + String(kipas));
//Serial.println("BATAS SUHU: " + String(batas_suhu) + ", BATAS ASAP: " + String(batas_asap) + ", BATAS CAHAYA: " + String(batas_cahaya));
//Serial.println("OTOMATIS DHT: " + String(otomatis_dht) + ", OTOMATIS LDR: " + String(otomatis_ldr));
//Serial.println("OTOMATIS PIR: " + String(otomatis_pir) + ", OTOMATIS MQ2: " + String(otomatis_mq2));
```

Gambar 4. 4 Source Code Arduino

Pada Gambar 4.4 merupakan bentuk *source code* pada arduino yang mengendalikan modul-modul yang telah dirangai dan dipasang pada sistem *smarthome*, alasan peletakan modul *relay* yang menjadi satu papan dengan ESP8266 dan Arduino adalah untuk membentuk suatu struktur yang praktis dan ringkas, untuk penghubung ke modul-modul sensor, kipas, *LCD buzzer* dan lampu menggunakan kabel, peletakan power

supply juga menjadi satu bagian dengan papan ini dengan menggunakan catu daya *input* 220 *VAC* dan output *power supply* sebesar

4.3 Hasil Pengujian Fungsional Sistem

Pada perangkat ini terdapat *modul ESP8266*, *relay*, sensor *MQ-2*, sensor *Passive Infrared*, sensor *humidity* dan sensor cahaya sebagai perangkat *input sensing*, sementara pada *output* terdapat lampu, *buzzer* dan *LCD*. Pengujian software akan dilakukan dengan mengikuti langkah-langkah seperti pada tabel berikut:

4.3.1 Hasil Pengujian Koneksi Modul *IoT-Wifi*

Pada pengujian berikut ini akan dilakukan pengujian pada modul ESP8266 dengan parameter uji koneksi otomatis ke SSID yang telah disediakan oleh access point dalam penelitian ini adalah tethering dari handphone. Berikut merupakan tabel hasil pengujian perangkat yang terpasang pada system smarthome, dengan pengujian sebanyak empat kali ditiap modul yang digunakan

Tabel 4. 1 *Hasil Pengujian Modul IoT*

No	Keterangan Percobaan	Status
1	ID benar password benar	terkoneksi
2	ID benar password salah	tidak terkoneksi
3	ID salah password benar	tidak terkoneksi
4	ID salah password salah	tidak terkoneksi

Pada Tabel 4.2 menunjukkan hasil pengujian modul *ESP8266* dengan kondisi pengujian yang berbeda-beda dengan percobaan dengan pola *password* dan *ssid* yang berbeda sehingga diketahui untuk kemungkinan tidak dapat terhubung karena faktor *password* dan *SSID* yang berbeda-beda.

4.3.2 Pengujian Monitoring Objek Cerdas

Berikut merupakan perancangan proses pengujian dan bentuk dari hasil dari pengujian pada setia objek cerdas yang telah diintegerasikan ke dalam sistem *smarthome*.

Berikut merupakan bentuk skenario pengujian sistem smarthome yang meliputi pengujian pada performansi dan pengujian pada setiap objek pintar yang telah diterapkan pada *smarthome system.*

1 Pengujian perfomansi memfokuskan pada pengujian dengan menggunakan wireshark untuk capture paket yang lalu lintas di dalam jaringan.
2 Pengujian menggunakan wireshark dilakukan untuk mengetahui nilai jitter, delay, throughput dan packet loss yang akan digunakan sebagai parameter *Quality of Service.*
3 Pengujian objek cerdas yang telah terintegerasi, dalam hal ini objek cerdas yang di maksud adalah sensor-sensor dan lampu serta kipas, objek ini disebut objek cerdas karena telah terintegerasi dengan sistem yang dapat dikendalikan oleh Blynk, jadi objek cerdas dalam penelitian ini bukan smartlamp seperti produk Philips, bardi ataupun sonoff
4 Rekap hasil pengujian setiap objek cerdas, rekap yang dimaksud adalah kajian mengenai pengujian yang telah dilakukan agar diketahui jika ada objek yang tidak berfungsi sebagaimana mestinya
5 Rekap hasil pengujian performansi menggunakan parameter TIPON

Berikut merupakan hasil dari pengujian monitoring dari stiap objek cerdas dalam penelitian ini, pengujian ini berguna untuk mengisi data dari scenario pengujian modul-modul dan objek cerdas yang ada.

Tabel 4. 2 Hasil Pengujian Monitoring

No.	Objek Blynk	Lokasi	Hasil
1.	Objek 1	Teras	berhasil
2.	Objek 2	Ruang tamu	berhasil
3.	Objek 3	Ruang tamu	berhasil
4.	Objek 4	Kamar 1	berhasil
5.	Objek 5	Kamar 2	berhasil
6.	Objek 6	Kamar mandi	berhasil
7.	Objek 7	Dapur	berhasil

Pada table diatas merupakan bentuk hasil pengujian fungsional pada setiap objek yang terdefinisikan di antar muka aplikasi blynk, tujuan pengujian diatas adalah untuk mengetahui apakah aplikasi ini dapat

bekerja dengan semestinya, hasil yang diperoleh adalah keseluruhan objek dapat bekerja dengan baik.

4.3.3 Hasil Pengujian Aktifasi Lampu

Berikut merupakan hasil dari pengujian aktifasi lampu menggunakan aplikasi blynk ke sistem smarthome

Tabel 4. 3 Pengujian Aktifasi Lampu

Model pengujian	hasil	Keterengan
Nyalakan lampu 1	Berhasil	Berhasil menyalakan lampu, label Led 1 berubah ketika perintah eksekusi diberikan baik untuk k*on*disi *on* dan *off*
Nyalakan lampu 2	Berhasil	Berhasil menyalakan lampu, label Led 2 berubah ketika perintah eksekusi diberikan baik untuk k*on*disi *on* dan *off*
Nyalakan lampu 3	Berhasil	Berhasil menyalakan lampu, label Led 3 berubah ketika perintah eksekusi diberikan baik untuk k*on*disi *on* dan *off*
Nyalakan lampu 4	Berhasil	Berhasil menyalakan lampu, label Led 4 berubah ketika perintah eksekusi diberikan baik untuk k*on*disi *on* dan *off*
Nyalakan lampu 5	Berhasil	Berhasil menyalakan lampu, label Led 5 berubah ketika perintah eksekusi diberikan baik untuk k*on*disi *on* dan *off*

Pada Tabel 4.3 menunjukan hasil pengujian terhadap proses eksekusi perintah menyalakan lampu, dari table tersebut dapat diketahui jika semua fungsi dapat berjalan dengan benar baik dalam perintah eksekusi k*on*disi *on* dan *off*.

4.3.4 Hasil Pengujian Respond Sensor

Berikut merupakan hasil pengujian terhadap setiap sensor yang terintegerasi dalam sistem smarthome.

Tabel 4. 4 Pengujian Uji Sensor

Model Pengujian	Threshold	Hasil dan Keterangan
Aktifkan sensor *Humidity* dan konfigurasi *threshold*	30	Sensor berhasil diaktifkan dan sensor dapat bereaksi dengan perubahan suhu yang telah ditetapkan sebagai *threshold* sehinga memicu penyalaan kipas angin
Aktifkan sensor *MQ-2*	10	Sensor berhasil diaktifkan dan sensor dapat bereaksi dengan deteksi adanya gas *LPG* disekitar area sensor sehinga memicu *buzzer*
Aktifkan sensor *Passive Infrared*	×	Sensor berhasil diaktifkan dan sensor dapat bereaksi dengan perubahan pergerakan objek disekitar area *sensing* yang telah ditetapkan sebagai *threshold* sehinga memicu penyalaan lampu
Aktifkan sensor cahaya	50	Sensor berhasil diaktifkan dan sensor dapat bereaksi dengan perubahan cahaya yang telah ditetapkan sebagai *threshold* sehinga memicu lampu

Berdasarkan pengujian yang ditunjukan pada Tabel 4.4 menunjukan bahwa modul-modul sensor yang telah digunakan dan dipasang dapat bekerja dengan benar yakni menghasilkan perubahan pembacaan ketika nilai sensing berubah dan memberikan aksi eksekusi sesuai dengan yang telah didefinisikan. Pada pengujian yang telah dilakukan setiap sensor perlu diaktifkan terlebih dahulu sebelum dapat melakukan suatu reaksi karena nilai yang terbaca tidak akan diproses oleh system sampai sensor diaktifkan dan terkonfigurasi.

4.4 Pengukuran Perfomansi (QoS)

Pengujian ini ditujukan untuk mengetahui kinerja dari sistem dengan menggunakan analisis *Quality of Services* atau *QoS* dengan parameter pengukuran berupa *jitter, delay, throughput* dan *packet loss*. Tujuan analisis ini adalah untuk mengetahui karakteristik waktu antar *packet* yang melalui jaringan pada sistem *smarthome.*

4.4.1 Hasil Pengukuran Berdasar Lama Pengamatan

Pada bagian ini akan menjelaskan mengenai hasil pengukuran berdasarkan pada lamanya waktu pengamatan data, pengujian ini menggunakan kali linux sebagai sistem operasi yang menjalankan wireshark untuk mengamati packet yang lewat di dalam jaringan, untuk itu sistem ini harus berada dalam satu jaringan lokal berikut merupakan tabel hasil pengukuran berdasarkan lama pengamatan.

Tabel 4. 5 Hasil Berdasarkan Lama Pengamatan

Lama Pengamatan	Delay	Jitter	Throughput	Packet Loss
5 menit	0,2 ms	2,1 ms	2,6 bit/sec	0
10 menit	0,18 ms	2,4 ms	2,9 bit/sec	0
15 menit	0,12 ms	1,3 ms	3,2 bit/sec	0

Pada Gambar 4.5 merupakan bentuk tabel hasil pengukuran berdasarkan lama pengukuran, dengan melihat dari grafik diatas menunjukan hasil yang berbeda pada setiap lama pengamatan hal ini dikarenakan jumlah paket yang lalu lalang selama lima menit dan limabelas menit akan sangat berbeda.

Berdasarkan pengujian selama lima menit, sepuluh menit dan limabelas menit menghasilkan nilai yang bervariasi pada parameter delay, hal ini dipengaruhi oleh kualitas sinyal pada jaringan internet seluler yang pada pengujian selama sepuluh menit kondisi cuaca dalam keadaan hujan mendung sehingga mengalami penurunan kualitas transfer data. Sementara untuk pengujian selama tigapuluh menit menghasilkan empat packet loss, hal ini tidak pernah terjadi pada pengujian sebelumnya, packet loss mungkin dipengaruhi oleh kualitas jaringan selama melakukan pengujian.

4.4.2 Hasil Pengukuran Berdasarkan Waktu Pengamatan

Pada bagian ini akan menjelaskan mengenai hasil pengukuran berdasarkan pada waktu pengamatan data, pengujian ini menggunakan kali linux sebagai sistem operasi yang menjalankan wireshark untuk mengamati packet yang lewat di dalam jaringan, untuk itu sistem ini harus berada dalam satu jaringan lokal berikut merupakan tabel hasil pengukuran berdasarkan waktu pengamatan.\

Tabel 4. 6 Hasil Berdasarkan Waktu Pengamatan

Lama Pengamatan	Delay	Jitter	Throughput	Packet Loss
pagi	2.212 ms	0.286 ms	6.667 bit/sec	0
siang	2.111 ms	0.291 ms	5.611 bit/sec	0
malam	2.076 ms	0.286 ms	3.144 bit/sec	0

Pada tabel 4.6 merupakan bentuk tabel hasil pengukuran berdasarkan waktu pengamatan, dengan melihat dari tabel diatas menunjukan hasil yang berbeda pada setiap lama pengamatan hal ini dikarenakan jumlah paket yang lalu lalang selama proses pengukuran ini berbeda karena beban traffic jaringan akan menyesuaikan dengan jam beraktifitas.

4.4.3 Hasil Pengukuran Berdasarkan Jarak

Pada bagian ini akan menjelaskan mengenai hasil pengukuran berdasarkan pada waktu pengamatan data, pengujian ini menggunakan kali linux sebagai sistem operasi yang menjalankan wireshark untuk mengamati packet yang lewat di dalam jaringan, untuk itu sistem ini harus berada dalam satu jaringan lokal berikut merupakan grafik hasil pengukuran berdasarkan lama pengamatan.

Tabel 4. 7 Tabel Berdasarkan Jarak Pengamatan

Lama Pengamatan	Delay	Jitter	Throughput	Packe t Loss
1 meter	5.119 ms	0.034 ms	0.045 bit/sec	0
5 meter	3.324 ms	0.435 ms	0.098 bit/sec	0
10 meter	3.002 ms	0.987 ms	0.123 bit/sec	0

Pada Gambar 4.5 merupakan bentuk grafik hasil pengukuran berdasarkan jarak pengukuran, dengan melihat dari grafik diatas menunjukan hasil yang berbeda pada setiap jarak pengamatan hal ini dikarenakan kemampuan modul ESP8266 dalam menerima dan terhubung dalam suatu jaringan dipengaruhi oleh jarak ke access point.

4.5 Analisa dan Pembahasan

Pada penelitian ini proses perancangan perangkat keras dimulai dengan pengumpulan modul-modul, perangkaian modul menyesuaikan dengan fungsi-fungsi yang ingin dicapai dalam penelitian ini, persiapan layout PCB dan cross check perangkat menyesuaikan dengan jalur PCB yang telah dibuat.

Komunikasi antara perangkat arduino dan ESP8266 menggunakan komunikasi serial yang ditulis dalam bahasa C arduino, komunikasi ini digunakan untuk mengirimkan input dari perintah blynk ke arduino kemudian diteruskan ke modul relay

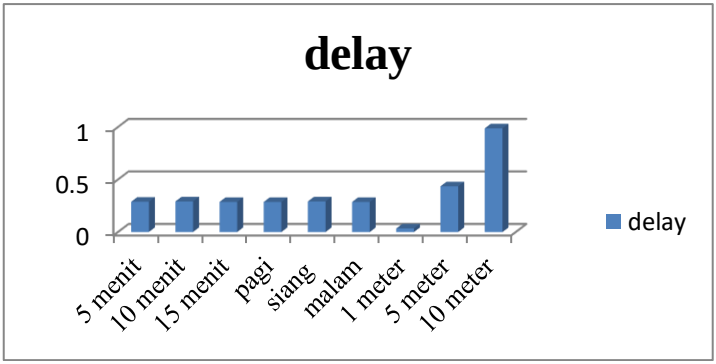
Pengendalian perangkat ini menggunakan blynk yang diinstall pada smartphone user, dari aplikasi blynk ini pengguna dapat memerintahkan objek-objek yang diintegerasikan pada sistem smarthome, komunikasi system smarthome ke blynk menggunakan jaringan internet yang data di berikan user akan dikelola oleh server blynk untuk kemudian diteruskan ke modul ESP8266 yang menggunakan library blynk.

4.6 Perbandingan Performansi Sistem

Pengujian berdasarkan perbandingan perfomansi system bertujuan untuk mengetahui kinerja system dari berbagai aspek meliputi delay, jitter, throughput dan packet loss.

4.6.1 Perbandingan Delay

Perbandingan delay dalam pengujian ini akan merujuk pada nilai delay yang diperoleh dari pengukuran sebelumnya yaitu berdasarkan lama pengamatan, waktu pengamatan dan jarak.

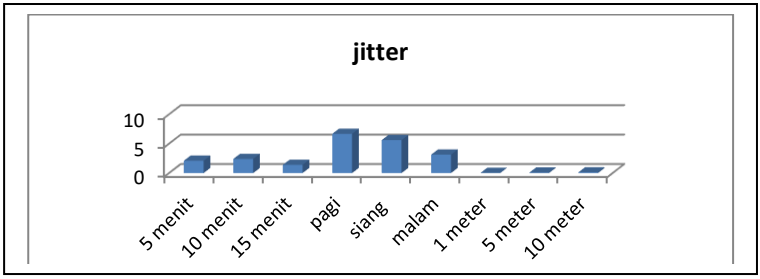


Gambar 4. 5 *Perbandingan Delay*

Pada gambar diatas merupakan perbandingan delay yang diperoleh selama proses pengukuran dengan 3 kriteria pengukuran yang berbeda-beda, hasil yang diperoleh adalah nilai delay yang bervariasi karena setiap proses pengukuran tidak terikata pada suatu variable atau penentuan kondisi khusus.

4.6.2 Perbandinga Jitter

Perbandingan jitter dalam pengujian ini akan merujuk pada nilai delay yang diperoleh dari pengukuran sebelumnya yaitu berdasarkan lama pengamatan, waktu pengamatan dan jarak.

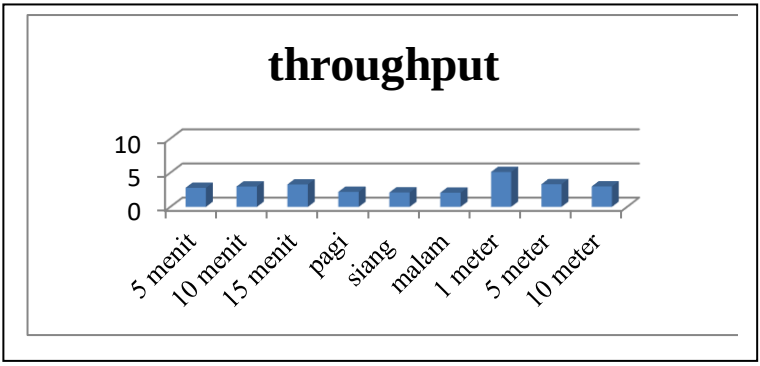


Gambar 4. 6*Perbandingan Jitter*

Pada gambar diatas merupakan perbandingan jitter yang diperoleh selama proses pengukuran dengan 3 kriteria pengukuran yang berbeda-beda, hasil yang diperoleh adalah nilai jitter yang bervariasi karena setiap proses pengukuran tidak terikata pada suatu variable atau penentuan kondisi khusus.

4.6.3 Perbandingan Throughput

Perbandingan delay dalam pengujian ini akan merujuk pada nilai delay yang diperoleh dari pengukuran sebelumnya yaitu berdasarkan lama pengamatan, waktu pengamatan dan jarak.

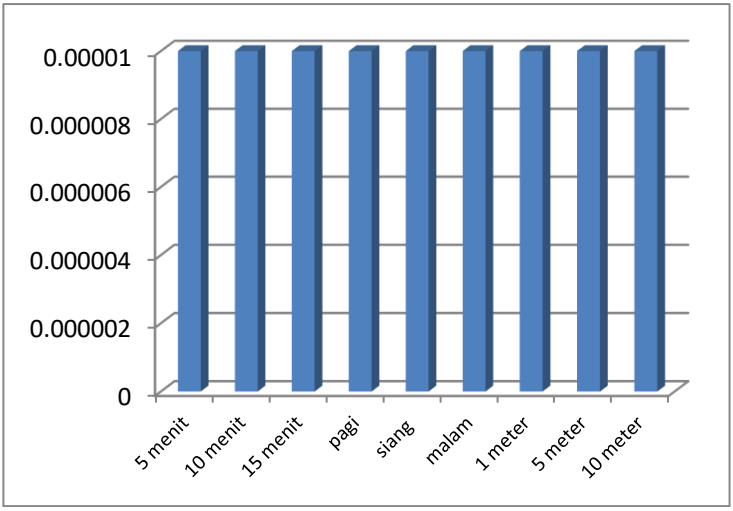


Gambar 4. 7 Perbandingan *Throughput*

Pada gambar diatas merupakan perbandingan throughput yang diperoleh selama proses pengukuran dengan 3 kriteria pengukuran yang berbeda-beda, hasil yang diperoleh adalah nilai throughput yang bervariasi karena setiap proses pengukuran tidak terikata pada suatu variable atau penentuan kondisi khusus.

4.6.4 Perbandingan Packet loss

Perbandingan packet loss dalam pengujian ini akan merujuk pada nilai delay yang diperoleh dari pengukuran sebelumnya yaitu berdasarkan lama pengamatan, waktu pengamatan dan jarak.



Gambar 4. 8 Perbandingan *Packet Loss*

Pada gambar diatas merupakan perbandingan packet losss yang diperoleh selama proses pengukuran dengan 3 kriteria pengukuran yang berbeda-beda, hasil yang diperoleh adalah nilai packet loss yang bervariasi karena setiap proses pengukuran tidak terikata pada suatu variable atau penentuan kondisi khusus.

4.6.5 Klasifikasi Hasil Perfomransi (QoS)

Pada table berikut menjelaskan mengenai rata-rata dari seluruh hasil pengujian yang telah dilakukan, tujuan pencarian nilai rata-rata ini adalah untuk menentukan Berdasarkan table diatas diperoleh nilai rata-rata delay sebesar 1,98 yang pada standart TIPHON menjelaskan bahwa nilai delay tersebut qualitas Berdasarkan table dibawah diperoleh nilai rata-rata delay sebesar 1,98 yang pada standart TIPHON menjelaskan standart berdasarkan *TIPHON*. Berikut merupakan pemaparan hasil analisis QoS pada table berikut:

Tabel 4. 8 Hasil Analisis Qos

Variabel	Delay	Jitter	Throughput	Packet Loss
5 menit	0,2 ms	2,1 ms	2,6 bit/sec	0
10 menit	0,18 ms	2,4 ms	2,9 bit/sec	0
15 menit	0,12 ms	1,3 ms	3,2 bit/sec	0
pagi	2.212 ms	0.286 ms	6.667 bit/sec	0
siang	2.111 ms	0.291 ms	5.611 bit/sec	0
malam	2.076 ms	0.286 ms	3.144 bit/sec	0
1 meter	5.119 ms	0.034 ms	0.045 bit/sec	0
5 meter	3.324 ms	0.435 ms	0.098 bit/sec	0
10 meter	3.002 ms	0.987 ms	0.123 bit/sec	0
Rata-rata	1.982667 ms	0.257667 ms	1.743111 bit/sec	0

Berdasarkan table diatas diperoleh nilai rata-rata delay sebesar 1,98 yang pada standart TIPHON menjelaskan bahwa nilai delay tersebut masuk dalam kategori bagus, parameter jitter memiliki nilai rata-rata 0,25 yang sangat bagus sesuai dengan ketentuan standart dan untuk nilai throughput memiliki rata-rata 1.743 yang masuk dalam kategori sangat bagus.

Tabel 4. 9 Hasil Analisis Qos standart TIPHON

No	Hasil			
	Delay	***Jitter***	***Packet loss***	***Throughput***
1	1.982667 *ms /* sangat bagus	0.257667*ms /* sangat bagus	*0 /* sangat bagus	1.743111 *bit/mili second* sangat bagus

Berdasarkan table diatas diperoleh nilai rata-rata delay sebesar 1,98 yang pada standart TIPHON menjelaskan bahwa nilai delay tersebut masuk dalam kategori bagus, parameter jitter memiliki nilai rata-rata 0,25 yang sangat bagus sesuai dengan ketentuan standart dan untuk nilai throughput memiliki rata-rata 1.743 yang masuk dalam kategori sangat bagus.

4.7 User Manual

Dalam menggunakan *Prototype Smart Home System* ini ada beberapa langkah-langkah seperti berikut:

1. Mendonload aplikasi Blynk pada HP
2. Membuat akun blynk
3. Memprogram untuk mengganti SSID dan password untuk menyambung *Wi-fi*
4. Melihat *User Interface* Blynk di HP apakah sudah terhubung dengan internet
5. *Prototype Smart Home* sudah siap untuk digunakan

BAB 5

PENUTUP

5.1 Kesimpulan

Setelah dilakukan perancangan dan pengujian jaringan komunikasi pada sistem smarthome, maka secara keseluruhan dapat diambil kesimpunglan sebagai berikut:

5.1.1 Hardware

Berdasarkan perancangan yang telah dilakukan maka diperoleh kesimpulan perancangan hardware dengan benar, keseluruhan objek yang dijelaskan dalam perancangan dapat berfungsi semua dimulai dari lampu teras hingga bagian kamar mandi.Sensor-sensor yang diintegerasikan dalam system ini dapat bekerja mengikuti threshold yang telah ditentukan, dimulai dari sensor pir yang ada di teras, sensor humidity pada ruang tamu, sensor gas pada dapur dan passive infrared pada kamar mandi.

5.1.2 Delay Throughput dan Jitter

1. Untuk mengukur *Quality of Service* dilakukan dengan monitoring jaringan menggunakan *wireshark*.
2. Parameter *Quality of Service* yang terpengaruh dari adanya beban *traffic* adalah *delay*, masuk dalam indeks “sangat bagus” pada standar *TIPHON* yaitu <150 *ms*.
3. Hasil pengukuran *throughput* rata rata adalah 1.743111 *bits/sec*.
4. Hasil pengukuran *jitter* pada saat beban *traffic* besar ada dalam kategori “Sangat Bagus” pada standar *TIPHON* dengan nilai 0-75 ms yaitu 2 ms.
5. Delay yang dihasilkan bervariasi dari setiap pengukuran dan masih dalam kategori bagus
6. Untuk nilai throughput yang diperoleh bergantung pada lama pengambilan data atau pengamatan

5.2 Saran

Dari hasil penelitian tugas akhir yang telah dilakukan, berikut merupakan saran yang dapat dilakukan untuk penelitian selanjutnya:

1. Menggunakan teknologi voice command API seperti google assistant atau amazon alexa.
2. Menggunakan perangkat *Narrowband IoT*)NB-IoT untuk di sebuah Runah sungguhan

Daftar Pustaka

[1] Dewaweb, " Internet of Things: Panduan Lengkap ".<URL: https://www.dewaweb.com/blog/internet-of-things/>, Januari 2020

[2] Jimblom, "Using the Arduino Pro Mini 3.3V". <URL: https://learn.sparkfun.com/tutorials/using-the-arduino-pro-mini-33v/all >, Januari 2020

[3] Harry Yuliansyah, "Uji Kinerja Pengiriman Data Secara Wireless Menggunakan Modul ESP8266 Berbasis Rest Architecture", ELECTRICIAN – Jurnal Rekayasa dan Teknologi Elektro, Vol. 10, No. 2, Mei 2016.

[4] Yessy Anggiana, "Lampu Emergency Otomatis Dengan Pengaturan Tingkat Intensitas Cahaya Menggunakan Smartphone Android". Universitas Sriwijaya, Palembang, 2017.

[5] H. Kim Lew, dkk, " Internetworking Technology Overview ". June, 1999.
International Confrence on Applied System Innovation 2018

[6] K.Chooruang ; K.Meekul " Design of an IoT Energy Monitoring System ". 2018 Sixteenth
International Confrence on ICT and Knowladge Engineering

[7] D. Handarly ; J. Lianda " Sistem Monitoring Daya Listrik Berbasis IoT" . Journal of Electrical control and Automotive Engineering (JEECAE)

[8] J. Lianda ; D. Handarly " Sistem Monitoring Konsumsi Daya Listrik Jarak Jauh Berbasis
Internet of things ". JTERA Vol. 4, No 1, Juni 2019, Hal 79-84

[9] H. Kim Lew, dkk, " Internetworking Technology Overview ". June, 1999.

[10] Handi, Hurriyatul Fitriyah, Gembong ES. "Sistem Pemantauan Menggunakan Blynk dan Pengendalian Penyiraman Tanaman Jamur Dengan Metode Logika Fuzzy". Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, Vol 3, No. 4 , April 2016

[11] Bakti Kominfo, " Penerapan Teknologi Smarthome Di Indonesia, Sudah Sejauh Apa ".< URL: https://www.baktikominfo.id/id/informasi/pengetahuan/penerapan_teknologi_smarthome_di_indonesia_sudah_sejauh_apa-1064 >. November 2019

[12] D. Korgut ; D.F Pigatto “ An Internet of Things based House Monitoring System “. 2018 IEEE Symposium on Computers and Communication (ISCC)
[13] Solekhan; “ EFISIENSI BEBAN SMART HOME (RUMAH PINTAR) BERBASIS ARDUINO UNO “. 2017 Prosiding SNATIF Ke -4 Tahun 2017

LAMPIRAN

Lampiran A

Blynk

```
// ------------------------------------------------------------------------------------
//#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266_SSL.h>

// ------------------------------------------------
char ssid[] = "Hari";
char pass[] = "hththari";
char auth[] = "UCsOEWd3zkvrRe7rJODN1eVGowFxp35o";

// ------------------------------------------------
char lcdBuff[20];
String sData, data[20];
unsigned long hold, lcd_hold;
byte otomatis_pir, otomatis_ldr, otomatis_dht, otomatis_mq2;
byte lampu_1, lampu_2, lampu_3, lampu_4, lampu_5, kompor, kipas, buzzer, slider;
int suhu, asap, gerak, cahaya, batas_cahaya, batas_suhu, batas_asap;
bool parsing = false;

WidgetLCD lcd(V0);
BlynkTimer timer;

// ------------------------------------------------------------------------------------
void setup() {
  Serial.begin(9600);
  Blynk.begin(auth, ssid, pass);
  lcd.clear();
  timer.setInterval(500L, parsingMode);
  timer.setInterval(5000L, lcdSlider);
```

Blynk | Arduino 1.8.10

File Edit Sketch Tools Help

Blynk

```
}

// ------------------------------------------------------------------------------------------
void loop() {
  Blynk.run();
  timer.run();
}

// ------------------------------------------------------------------------------------------
void parsingMode() {
  // URUTAN DATA: #lampu1#lampu2#lampu3#lampu4#lampu5#kompor#kipas#batas_suhu#batas_asap -
  // #batas_cahaya#otomatis_dht#otomatis_ldr#otomatis_pir#otomatis_mq2#notifikasi#$

  // #1#2#3#4#5#6#7#8#1#1#0#12#13#14#1#0#1#1#$
  while (Serial.available() > 0) {                                  // Jika ada Serial masuk
    char c = Serial.read();                                         // Baca data serial
    sData += c;                                                     // Tampung data ke variabel String
    if (c == '$') parsing = true;                                   // Parsing aktif jika ada karakter $
    if (parsing) {                                                  // Jika Parsing aktif
      int q = 0;
      for (int i = 0; i < sData.length(); i++) {                    // Perulangan sebanyak jumlah karakter yg masuk
        if (sData[i] == '#') {                                      // Jika ada karakter #
          q++;                                                      // Skip data
          data[q] = "";                                             // Pindah ke array lainnya
        } else {                                                    // Jika tidak ada karakter # (pemisah)
          data[q] = data[q] + sData[i];                             // Tampung data ke dalam Array
        }
      }
      // ------------------------------------------------------------  //
```

Blynk

```
      // ------------------------------------------------------------  //
      suhu        = data[1].toInt();
      asap        = data[2].toInt();
      cahaya      = data[3].toInt();
      gerak       = data[4].toInt();
      lampu_1     = data[5].toInt();
      lampu_2     = data[6].toInt();
      lampu_3     = data[7].toInt();
      lampu_4     = data[8].toInt();
      lampu_5     = data[9].toInt();
      kompor      = data[10].toInt();
      kipas       = data[11].toInt();
      batas_suhu  = data[12].toInt();
      batas_asap  = data[13].toInt();
      batas_cahaya = data[14].toInt();
      otomatis_dht = data[15].toInt();
      otomatis_ldr = data[16].toInt();
      otomatis_pir = data[17].toInt();
      otomatis_mq2 = data[18].toInt();

      //Serial.println("------------------------------");
      //Serial.println("SUHU:" + String(suhu) + ", ASAP:" + String(asap) + ", CAHAYA:" + String(cahaya) + ", NOTIFIKASI:" + String(notifikasi));
      //Serial.println("LAMPU 1: " + String(lampu_1) + ", LAMPU 2: " + String(lampu_2) + ", LAMPU 3: " + String(lampu_3));
      //Serial.println("LAMPU 4: " + String(lampu_4) + ", LAMPU 5: " + String(lampu_5) + ", KOMPOR: " + String(kompor) + ", KIPAS: " + String(kipas));
      //Serial.println("BATAS SUHU: " + String(batas_suhu) + ", BATAS ASAP: " + String(batas_asap) + ", BATAS CAHAYA: " + String(batas_cahaya));
      //Serial.println("OTOMATIS DHT: " + String(otomatis_dht) + ", OTOMATIS LDR: " + String(otomatis_ldr));
      //Serial.println("OTOMATIS PIR: " + String(otomatis_pir) + ", OTOMATIS MQ2: " + String(otomatis_mq2));

      parsing = false;                                              // Matikan Parsing
```

Blynk

```
void lcdSlider() {
  lcd.clear();
  if (slider == 0) {                                                    // Tampilan Suhu
    sprintf(lcdBuff, "SUHU:%02d BATAS:%02d", suhu, batas_suhu);
    lcd.print(0, 0, lcdBuff);
    sprintf(lcdBuff, " OTOMATIS: %s", konversi(otomatis_dht));
    lcd.print(0, 1, lcdBuff);
  }
  else if (slider == 1) {                                               // Tampilan Cahaya
    sprintf(lcdBuff, "CHY:%02d  BATAS:%02d", cahaya, batas_cahaya);
    lcd.print(0, 0, lcdBuff);
    sprintf(lcdBuff, " OTOMATIS: %s", konversi(otomatis_ldr));
    lcd.print(0, 1, lcdBuff);
  }
  else if (slider == 2) {                                               // Tampilan Asap
    sprintf(lcdBuff, "ASAP:%02d BATAS:%02d", asap, batas_asap);
    lcd.print(0, 0, lcdBuff);
    sprintf(lcdBuff, " OTOMATIS: %s", konversi(otomatis_mq2));
    lcd.print(0, 1, lcdBuff);
  }
  else if (slider == 3) {                                               // Tampilan Asap
    sprintf(lcdBuff, "DETEK GERAKAN: %d", gerak);
    lcd.print(0, 0, lcdBuff);
    sprintf(lcdBuff, " OTOMATIS: %s", konversi(otomatis_pir));
    lcd.print(0, 1, lcdBuff);
  }
  else if (slider == 4) {                                               // Tampilan Asap
    sprintf(lcdBuff, "%s|%s  %s|%s", konversi(lampu_1), konversi(lampu_2), konversi(lampu_3), konversi(lampu_4));
    lcd.print(0, 0, lcdBuff);
```

Blynk

```
// ---------------------------------------------------------------------------------------
void lcdSlider() {
  lcd.clear();
  if (slider == 0) {                                                    // Tampilan Suhu
    sprintf(lcdBuff, "SUHU:%02d BATAS:%02d", suhu, batas_suhu);
    lcd.print(0, 0, lcdBuff);
    sprintf(lcdBuff, " OTOMATIS: %s", konversi(otomatis_dht));
    lcd.print(0, 1, lcdBuff);
  }
  else if (slider == 1) {                                               // Tampilan Cahaya
    sprintf(lcdBuff, "CHY:%02d  BATAS:%02d", cahaya, batas_cahaya);
    lcd.print(0, 0, lcdBuff);
    sprintf(lcdBuff, " OTOMATIS: %s", konversi(otomatis_ldr));
    lcd.print(0, 1, lcdBuff);
  }
  else if (slider == 2) {                                               // Tampilan Asap
    sprintf(lcdBuff, "ASAP:%02d BATAS:%02d", asap, batas_asap);
    lcd.print(0, 0, lcdBuff);
    sprintf(lcdBuff, " OTOMATIS: %s", konversi(otomatis_mq2));
    lcd.print(0, 1, lcdBuff);
  }
  else if (slider == 3) {                                               // Tampilan Asap
    sprintf(lcdBuff, "DETEK GERAKAN: %d", gerak);
    lcd.print(0, 0, lcdBuff);
    sprintf(lcdBuff, " OTOMATIS: %s", konversi(otomatis_pir));
    lcd.print(0, 1, lcdBuff);
  }
  else if (slider == 4) {                                               // Tampilan Asap
    sprintf(lcdBuff, "%s|%s  %s|%s", konversi(lampu_1), konversi(lampu_2), konversi(lampu_3), konversi(lampu_4));
```

Blynk

```
  else if (slider == 3) {                                          // Tampilan Asap
    sprintf(lcdBuff, "DETEK GERAKAN: %d", gerak);
    lcd.print(0, 0, lcdBuff);
    sprintf(lcdBuff, " OTOMATIS: %s", konversi(otomatis_pir));
    lcd.print(0, 1, lcdBuff);
  }
  else if (slider == 4) {                                          // Tampilan Asap
    sprintf(lcdBuff, "%s|%s  %s|%s", konversi(lampu_1), konversi(lampu_2), konversi(lampu_3), konversi(lampu_4));
    lcd.print(0, 0, lcdBuff);
    sprintf(lcdBuff, " %s | %s | %s ", konversi(lampu_5), konversi(kompor), konversi(kipas));
    lcd.print(0, 1, lcdBuff);
  }

  if (slider >= 4) {                                               // Update tampilan LCD
    slider = 0;
  } else {
    slider++;
  }
}

// ------------------------------------------------------------------------------------
char* konversi(byte logic) {
  String str;
  if (logic == 1) str = "-ON";
  else str =  "OFF";
  char myBuff[str.length() + 1];
  str.toCharArray(myBuff, str.length() + 1);
  return myBuff;
}

  // ------------------------------------------------------------------------------------
  BLYNK_WRITE(V1) {
    Serial.println("#LAMPU1#" + (String)param.asInt() + "#$");
  }
  BLYNK_WRITE(V2) {
    Serial.println("#LAMPU2#" + (String)param.asInt() + "#$");
  }
  BLYNK_WRITE(V3) {
    Serial.println("#LAMPU3#" + (String)param.asInt() + "#$");
  }
  BLYNK_WRITE(V4) {
    Serial.println("#LAMPU4#" + (String)param.asInt() + "#$");
  }
  BLYNK_WRITE(V5) {
    Serial.println("#LAMPU5#" + (String)param.asInt() + "#$");
  }
  BLYNK_WRITE(V6) {
    Serial.println("#KOMPOR#" + (String)param.asInt() + "#$");
  }
  BLYNK_WRITE(V7) {
    Serial.println("#KIPAS#" + (String)param.asInt() + "#$");
  }
  BLYNK_WRITE(V8) {
    Serial.println("#BATAS_SUHU#" + (String)param.asInt() + "#$");
  }
  BLYNK_WRITE(V9) {
    Serial.println("#BATAS_CAHAYA#" + (String)param.asInt() + "#$");
  }
```

File Edit Sketch Tools Help

Nano

```
// ---------------------------------------------------------------------------------------
// Librarry Arduino
#include <SoftwareSerial.h>                                   // Library Serial, merubah pin I/O menjadi Komu
#include <LiquidCrystal_I2C.h>                                // Library LCD dengan module I2C
#include <DHT.h>                                              // Library Sensor Suhu DHT11
#include <EEPROM.h>                                           // Library EEPROM

// Definisikan Pin sesuai dengan Skemtik, cek skematik EAGLE
#define BUZZER          A0                                    // Pin Buzzer
#define LAMPU1          13                                    // RLY1 Pin Lampu Teras
#define LAMPU2          12                                    // RLY2 Pin Lampu Kamar Mandi
#define LAMPU3          11                                    // RLY3 Pin Lampu Kamar Tidur 1
#define LAMPU4          10                                    // RLY4 Pin Lampu Kamar Tidur 2
#define LAMPU5          9                                     // RLY5 Pin Lampu Ruang Tamu & Makan
#define KOMPOR          8                                     // RLY6 Pin Kompor
#define KIPAS           7                                     // RLY7 Pin Kipas Angin
#define SENSOR_MQ2      A2                                    // Pin Sensor Asap
#define SENSOR_LDR      A3                                    // Pin Sensor Cahaya
#define SENSOR_PIR      2                                     // Pin Sensor Gerakan
#define SENSOR_DHT      3                                     // Pin Sensor Suhu
#define WIFI_RX         4                                     // Pin RX,  terhubung TX Wifi
#define WIFI_TX         5                                     // Pin TX,  terhubung RX Wifi

#define EE_LAMPU1       0                                     // Register EEPROM Lampu 1
#define EE_LAMPU2       1                                     // Register EEPROM Lampu 2
#define EE_LAMPU3       2                                     // Register EEPROM Lampu 3
#define EE_LAMPU4       3                                     // Register EEPROM Lampu 4
#define EE_LAMPU5       4                                     // Register EEPROM Lampu 5
#define EE_KOMPOR       5                                     // Register EEPROM Kompor
```


```
#define EE_LAMPU1       0                                     // Register EEPROM Lampu 1
#define EE_LAMPU2       1                                     // Register EEPROM Lampu 2
#define EE_LAMPU3       2                                     // Register EEPROM Lampu 3
#define EE_LAMPU4       3                                     // Register EEPROM Lampu 4
#define EE_LAMPU5       4                                     // Register EEPROM Lampu 5
#define EE_KOMPOR       5                                     // Register EEPROM Kompor
```


```
#define EE_KIPAS        6                                     // Register EEPROM Kipas
#define EE_BATAS_SUHU   7                                     // Register EEPROM Batas Suhu
#define EE_BATAS_CAHAYA 8                                     // Register EEPROM Batas Cahaya
#define EE_BATAS_ASAP   9                                     // Register EEPROM Batas Cahaya
#define EE_OTOMATIS_DHT 10                                    // Register EEPROM Otomatis Suhu
#define EE_OTOMATIS_PIR 11                                    // Register EEPROM Otomatis Gerakan
#define EE_OTOMATIS_LDR 12                                    // Register EEPROM Otomatis Cahaya
#define EE_OTOMATIS_MQ2 13                                    // Register EEPROM Otomatis Asap

#define PIR_DETEKSI     HIGH                                  // Nilai saat Sensor Deteksi gerakan
#define ON              HIGH                                  // Kondisi ON Relay = HIGH / 1
#define OFF             LOW                                   // Kondisi OFF Relay = LOW / 0

// Variabel Global
byte otomatis_pir, otomatis_ldr, otomatis_dht, otomatis_mq2;
byte lampu_1, lampu_2, lampu_3, lampu_4, lampu_5, kompor, kipas, buzzer, slider;
int suhu, asap, gerak, cahaya, batas_cahaya, batas_suhu, batas_asap;
char lcdBuff[20];
String sData, data[4];
long pir_hold, ldr_hold, dht_hold, mq2_hold, buzzer_hold, lcd_hold, wifi_hold;
bool parsing = false;
bool alarm   = false;
```

```
/ Configurasi Library
oftwareSerial wifi(WIFI_RX, WIFI_TX);                                   // Pin RX TX
iquidCrystal_I2C lcd(0x27, 16, 2);                                      // Alamat I2C, Kolom, Baris
HT dht(SENSOR_DHT, DHT11);

/ ----------------------------------------------------------------------------------------
oid setup() {
  Serial.begin(9600);                                                   // Komunikasi dengan PC
  wifi.begin(9600);                                                     // Komunikasi dengan ESP
  lcd.begin();                                                          // Setup LCD
  dht.begin();                                                          // Setup DHT

  // Konfigurasi Pin
  pinMode(SENSOR_PIR, INPUT);
  pinMode(SENSOR_LDR, INPUT_PULLUP);
  pinMode(SENSOR_MQ2, INPUT_PULLUP);
  pinMode(LAMPU1, OUTPUT);
  pinMode(LAMPU2, OUTPUT);
  pinMode(LAMPU3, OUTPUT);
  pinMode(LAMPU4, OUTPUT);
  pinMode(LAMPU5, OUTPUT);
  pinMode(KIPAS, OUTPUT);
  pinMode(KOMPOR, OUTPUT);
  pinMode(BUZZER, OUTPUT);

  // Baca Kondisi data terakhir di EEPROM
  readEEPROM();

// ----------------------------------------------------------------------------------------
oid loop() {
  parsingMode();                                                    // Cek data Serial dari WIFI
  cahaya = map(analogRead(SENSOR_LDR), 1023, 0 , 0, 99);              // Baca Sensor Cahaya
  asap = map(analogRead(SENSOR_MQ2), 0, 1023 , 0, 99);               // Baca Sensor Asap
  suhu = dht.readTemperature();                                     // Baca Sensor Suhu (Celcius)


  digitalWrite(LAMPU3, lampu_3);                                    // Kondisi Lampu berdasarkan Tombol
  digitalWrite(LAMPU4, lampu_4);                                    // Kondisi Lampu berdasarkan Tombol
  digitalWrite(LAMPU5, lampu_5);                                    // Kondisi Lampu berdasarkan Tombol
  digitalWrite(KOMPOR, kompor);                                     // Kondisi kompor berdasarkan Tombol

  // ------------------------------------------------------------   // KIRIM DATA KE WIFI
  if ((millis() - wifi_hold) > 3000) {                              // Kirim setiap 3 detik sekali
    // URUTAN DATA: #suhu#asap#cahaya#lampu1#lampu2#lampu3#lampu4#lampu5#kompor#kipas#batas_suhu#batas_asap -
    // #batas_cahaya#otomatis_dht#otomatis_ldr#otomatis_pir#otomatis_mq2#notifikasi#$
    String data_wifi = "#" + String(suhu) + "#" + String(asap) + "#" + String(cahaya) + "#" + String(gerak) +
                       "#" + String(lampu_1) + "#" + String(lampu_2) + "#" + String(lampu_3) +
                       "#" + String(lampu_4) + "#" + String(lampu_5) + "#" + String(kompor) + "#" + String(kipas) +
                       "#" + String(batas_suhu) + "#" + String(batas_asap) + "#" + String(batas_cahaya) +
                       "#" + String(otomatis_dht) + "#" + String(otomatis_ldr) + "#" + String(otomatis_pir) +
                       "#" + String(otomatis_mq2) + "#$";
    //Serial.println("KIRIM DATA:" + data_wifi);
    Serial.println("------------------------------");
    Serial.println("SUHU:" + String(suhu) + ", ASAP:" + String(asap) + ", CAHAYA:" + String(cahaya) + ", GERAKAN:" + String(gerak));
    Serial.println("LAMPU 1: " + String(lampu_1) + ", LAMPU 2: " + String(lampu_2) + ", LAMPU 3: " + String(lampu_3));
    Serial.println("LAMPU 4: " + String(lampu_4) + ", LAMPU 5: " + String(lampu_5) + ", KOMPOR: " + String(kompor) + ", KIPAS: " + String(kipas));
    Serial.println("BATAS SUHU: " + String(batas_suhu) + ", BATAS ASAP: " + String(batas_asap) + ", BATAS CAHAYA: " + String(batas_cahaya));
```

```
// ------------------------------------------------------------          // NOTIFIKASI
// Jika tidak ada notif penting
if ((millis() - lcd_hold) > 4000) {
  lcdSlider();                                                        // Tampilkan slider data LCD
  lcd_hold = millis();
}

if (alarm) {
  if ((millis() > (buzzer_hold + 500))) {
    digitalWrite(BUZZER, ON);
  }
  if (( millis() < (buzzer_hold + 1000)) && (millis() > (buzzer_hold + 500))) {
    digitalWrite(BUZZER, OFF);
  }
  if (millis() > (buzzer_hold + 1000) ) {
    buzzer_hold = millis();
  }
} else {
  digitalWrite(BUZZER, OFF);
}


// ------------------------------------------------------------          // SENSOR CAHAYA
if (otomatis_ldr == 1) {                                                // Jika tombol Otomatis LDR ON
  if (cahaya < batas_cahaya) {                                          // Jika Sensor LDR Lebih Batas
    ldr_hold = millis();                                                // Update waktu tahan lampu menyala
  }
  if ((millis() - ldr_hold) > 30000) {                                  // Jika sensor tidak mendeteksi gerakan selama 30 detik
```

```
  // ------------------------------------------------------------          // SENSOR PIR
  if (otomatis_pir == 1) {                                              // Jika tombol Otomatis PIR ON
    if (digitalRead(SENSOR_PIR) == PIR_DETEKSI) {                       // Jika Sensor PIR Deteksi
      pir_hold = millis();                                              // Update waktu tahan lampu menyala
    }
    if ((millis() - pir_hold) > 30000) {                                // Jika sensor tidak mendeteksi cahaya selama 30 detik
      digitalWrite(LAMPU2, OFF);                                        // Lampu Kamar Mandi Mati
      gerak = 0;
    } else {                                                            // Jika masih mendeteksi
      digitalWrite(LAMPU2, ON);                                         // Lampu Kamar Mandi Biarkan menyalah
      gerak = 1;
    }
  } else {                                                              // Jika tombol Otomatis PIR OFF
    digitalWrite(LAMPU2, lampu_2);                                      // Kondisi Lampu berdasarkan Tombol
  }

  // ------------------------------------------------------------          // SENSOR SUHU
  if (otomatis_dht == 1) {                                              // Jika tombol Otomatis DHT ON
    if (suhu > batas_suhu) {                                            // Jika Sensor Suhu Lebih Batas
      dht_hold = millis();                                              // Update waktu tahan lampu menyala
    }
    if ((millis() - dht_hold) > 30000) {                                // Jika sensor tidak melebihi batas selama 30 detik
      kipas = 0;
      digitalWrite(KIPAS, OFF);                                         // Lampu Teras Mati
    } else {                                                            // Jika masih mendeteksi
      kipas = 1;
      digitalWrite(KIPAS, ON);                                          // Lampu Teras Biarkan menyalah
    }
```

```
void lcdSlider() {
  lcd.clear();
  if (slider == 0) {            // Tampilan Suhu
    sprintf(lcdBuff, "SUHU:%02d BATAS:%02d", suhu, batas_suhu);
    lcd.setCursor(0, 0);
    lcd.print(lcdBuff);
    sprintf(lcdBuff, "   OTOMATIS:%d", otomatis_dht);
    lcd.setCursor(0, 1);
    lcd.print(lcdBuff);
  }
  else if (slider == 1) {   // Tampilan Cahaya
    sprintf(lcdBuff, "CHY:%02d  BATAS:%02d", cahaya, batas_cahaya);
    lcd.setCursor(0, 0);
    lcd.print(lcdBuff);
    sprintf(lcdBuff, "   OTOMATIS:%d", otomatis_ldr);
    lcd.setCursor(0, 1);
    lcd.print(lcdBuff);
  }
  else if (slider == 2) {  // Tampilan Asap
    sprintf(lcdBuff, "ASAP:%02d BATAS:%02d", asap, batas_asap);
    lcd.setCursor(0, 0);
    lcd.print(lcdBuff);
    sprintf(lcdBuff, "   OTOMATIS:%d", otomatis_mq2);
    lcd.setCursor(0, 1);
    lcd.print(lcdBuff);
  }
  else if (slider == 3) {  // Tampilan Gerak
    sprintf(lcdBuff, "DETEK GERAKAN: %d", gerak);
```

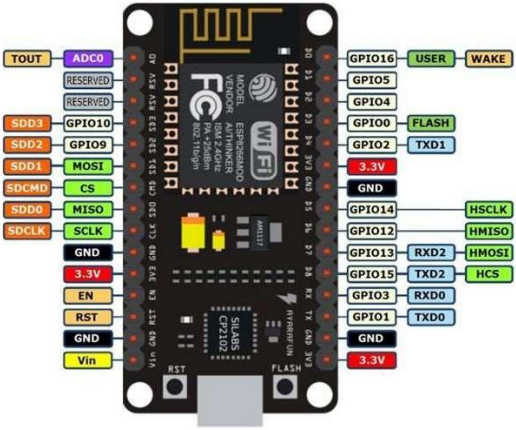
```
/ ---------------------------------------------------------------------------------------
oid updateInput(String respon, int nilai) {
 if (respon == "BATAS_SUHU") batas_suhu = nilai; EEPROM.write(EE_BATAS_SUHU, batas_suhu);
 if (respon == "BATAS_CAHAYA") batas_cahaya = nilai; EEPROM.write(EE_BATAS_CAHAYA, batas_cahaya);
 if (respon == "BATAS_ASAP") batas_asap = nilai; EEPROM.write(EE_BATAS_CAHAYA, batas_asap);
 //-----------------------------------------------
 if (respon == "OTOMATIS_DHT") otomatis_dht = nilai; EEPROM.write(EE_OTOMATIS_DHT, otomatis_dht);
 if (respon == "OTOMATIS_LDR") otomatis_ldr = nilai; EEPROM.write(EE_OTOMATIS_LDR, otomatis_ldr);
 if (respon == "OTOMATIS_MQ2" ) otomatis_mq2 = nilai; EEPROM.write(EE_OTOMATIS_MQ2, otomatis_mq2);
 if (respon == "OTOMATIS_PIR" ) otomatis_pir = nilai; EEPROM.write(EE_OTOMATIS_PIR, otomatis_pir);
 //-----------------------------------------------
 if (respon == "LAMPU1") lampu_1 = nilai; EEPROM.write(EE_LAMPU1, lampu_1);
 if (respon == "LAMPU2") lampu_2 = nilai; EEPROM.write(EE_LAMPU2, lampu_2);
 if (respon == "LAMPU3") lampu_3 = nilai; EEPROM.write(EE_LAMPU3, lampu_3);
 if (respon == "LAMPU4") lampu_4 = nilai; EEPROM.write(EE_LAMPU4, lampu_4);
 if (respon == "LAMPU5") lampu_5 = nilai; EEPROM.write(EE_LAMPU5, lampu_5);
 if (respon == "KOMPOR" ) kompor = nilai; EEPROM.write(EE_KOMPOR, kompor);
 if (respon == "KIPAS" ) kipas = nilai; EEPROM.write(EE_KIPAS, kipas);


/ ---------------------------------------------------------------------------------------
oid readEEPROM() {
 // Baca Kondisi data terakhir di EEPROM
 // Berfungsi jika mati lampu, data akan membaca kondisi terakhir setelah alat menyalah lagi.
 lampu_1      = EEPROM.read(EE_LAMPU1);
 lampu_2      = EEPROM.read(EE_LAMPU2);
 lampu_3      = EEPROM.read(EE_LAMPU3);
 lampu_4      = EEPROM.read(EE_LAMPU4);
```

```
// ---------------------------------------------------------------------------------------
void parsingMode() {
  while (wifi.available() > 0) {                                  // Jika ada Serial masuk
    char c = wifi.read();                                         // Baca data serial
    sData += c;                                                     // Tampung data ke variabel String
    if (c == '$') parsing = true;                                   // Parsing aktif jika ada karakter $
    if (parsing) {                                                  // Jika Parsing aktif
      // Contoh Data Masuk: #LAMPU1#1#$
      int q = 0;
      for (int i = 0; i < sData.length(); i++) {                    // Perulangan sebanyak jumlah karakter yg masuk
        if (sData[i] == '#') {                                      // Jika ada karakter #
          q++;                                                      // Skip data
          data[q] = "";                                             // Pindah ke array lainnya
        } else {                                                    // Jika tidak ada karakter # (pemisah)
          data[q] = data[q] + sData[i];                             // Tampung data ke dalam Array
        }
      }
      String respon = data[1];                                      // Data no 1 sebagai Respon data apa yang dikirim
      String nilai = data[2];                                       // Data no 2 sebagai nilai data yang dikirim
      Serial.println("RESPON: " + respon + ", NILAI: " + nilai);    // Monitoring data diterima ke PC
      updateInput(respon, nilai.toInt());                           // Update kondisi berdasarkan Respon dan Nilai
      parsing = false;                                              // Matikan Parsing
      sData = "";                                                   // Hapus Data dari Wifi
    }
  }
```

LAMPIRAN B

DATASHEET

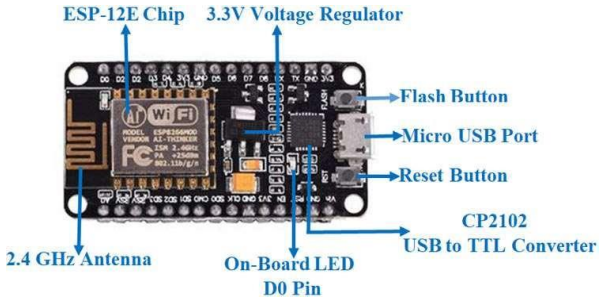
1. NodeMCU ESP8266



NodeMCU is an open-source Lua based firmware and **development board** specially targeted for IoT based Applications. It includes firmware that runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module.

The **NodeMCU ESP8266 development board** comes with the ESP-12E module containing ESP8266 chip having Tensilica Xtensa 32- bit LX106 RISC microprocessor. This microprocessor supports RTOS and operates at 80MHz to 160 MHz adjustable clock frequency. NodeMCU has 128 KB RAM and 4MB of Flash memory to store data and programs. Its high processing power with in-built Wi-Fi / Bluetooth and Deep Sleep Operating features make it ideal for IoT projects.

NodeMCU can be powered using Micro USB jack and VIN pin (External Supply Pin). It supports UART, SPI, and I2C interface



NodeMCU ESP8266 Specifications & Features

- Microcontroller: Tensilica 32-bit RISC CPU Xtensa LX106
- Operating Voltage: 3.3V
- Input Voltage: 7-12V
- Digital I/O Pins (DIO): 16
- Analog Input Pins (ADC): 1
- UARTs: 1
- SPIs: 1
- I2Cs: 1
- Flash Memory: 4 MB
- SRAM: 64 KB
- Clock Speed: 80 MHz
- USB-TTL based on CP2102 is included onboard, Enabling Plug n Play
- PCB Antenna
- Small Sized module to fit smartly inside your IoT projects

NodeMCU Development Board Pinout Configuration

Pin Category	Name	Description
Power	Micro-USB, 3.3V, GND, Vin	**Micro-USB:** NodeMCU can be powered through the USB port **3.3V:** Regulated 3.3V can be supplied to this pin to power the board **GND:** Ground pins **Vin:** External Power Supply
Control Pins	**EN, RST**	The pin and the button resets the microcontroller
Analog Pin	A0	Used to measure analog voltage in the range of 0-3.3V
GPIO Pins	GPIO1 to GPIO16	NodeMCU has 16 general purpose input-output pins on its board
SPI Pins	SD1, CMD, SD0, CLK	NodeMCU has four pins available for SPI communication.
UART Pins	TXD0, RXD0, TXD2, RXD2	NodeMCU has two UART interfaces, UART0 (RXD0 & TXD0) and UART1 (RXD1 & TXD1). UART1 is used to upload the firmware/program.
I2C Pins		NodeMCU has I2C functionality support but due to the internal functionality of these pins, you have to find which pin is I2C.

2. Sensor LDR

SUNROM TECHNOLOGIES
Your Source for Embedded Systems

Email: info@sunrom.com or sunrom@gmail.com
Visit us at http://www.sunrom.com

Document: Datasheet Date: 28-Jul-08 Model #: 3190 Product's Page: www.sunrom.com/p-510.html

Light Dependent Resistor - LDR

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.

Applications

Photoconductive cells are used in many different types of circuits and applications.

Analog Applications

- Camera Exposure Control
- Auto Slide Focus - dual cell
- Photocopy Machines - density of toner
- Colorimetric Test Equipment
- Densitometer
- Electronic Scales - dual cell
- Automatic Gain Control – modulated light source
- Automated Rear View Mirror

Digital Applications

- Automatic Headlight Dimmer
- Night Light Control
- Oil Burner Flame Out
- Street Light Control
- Absence / Presence (beam breaker)
- Position Sensor

Electrical Characteristics

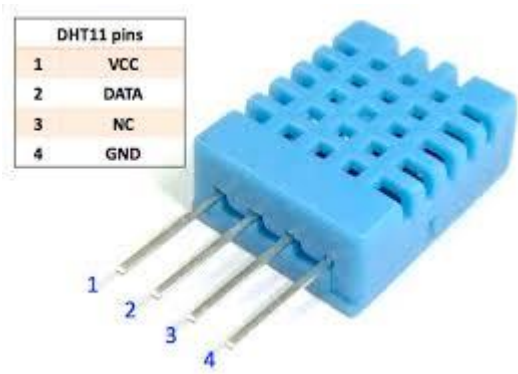
Parameter	Conditions	Min	Typ	Max	Unit
Cell resistance	1000 LUX 10 LUX	- -	400 9	- -	Ohm K Ohm
Dark Resistance	-	-	1	-	M Ohm
Dark Capacitance	-	-	3.5	-	pF
Rise Time	1000 LUX 10 LUX	- -	2.8 18	- -	ms ms
Fall Time	1000 LUX 10 LUX	- -	48 120	- -	ms ms
Voltage AC/DC Peak		-	-	320	V max
Current		-	-	75	mA max
Power Dissipation				100	mW max
Operating Temperature		-60	-	+75	Deg. C

3. Sensor DHT 11
Part Number : DHT11

Function : Temperature and humidity Sensor module

Manufacturers : AOSONG

Image & Pinouts:



Description :

This DFRobot DHT11 Temperature & Humidity Sensor features a temperature & humidity sensor complex with a calibrated digital signal output. By using the exclusive digital-signal-acquisition technique and temperature & humidity sensing technology, it ensures high reliability and excellent long-term stability. This sensor includes a resistive-type humidity measurement component and an NTC temperature measurement component, and connects to a high performance 8-bit microcontroller, offering excellent quality, fast response, anti-interference ability and cost-effectiveness.

Each DHT11 element is strictly calibrated in the laboratory that is extremely accurate on humidity calibration. The calibration coefficients are stored as programmes in the OTP memory, which are used by the sensor's internal signal detecting process. The single-wire serial interface makes system integration quick and easy. Its small size, low power consumption and up-to-20 meter signal transmission making it the best choice for various applications, including those most demanding ones. The component is 4-pin single row pin

package. It is convenient to connect and special packages can be provided according to users' request.

4. Sensor MQ2

HANWEI ELETRONICS CO.,LTD MQ-2 http://www.hwsensor.com

TECHNICAL DATA MQ-2 GAS SENSOR

FEATURES

Wide detecting scope — Fast response and High sensitivity
Stable and long life — Simple drive circuit

APPLICATION

They are used in gas leakage detecting equipments in family and industry, are suitable for detecting of LPG, i-butane, propane, methane ,alcohol, Hydrogen, smoke.

SPECIFICATIONS

A. Standard work condition

Symbol	Parameter name	Technical condition	Remarks
Vc	Circuit voltage	5V±0.1	AC OR DC
V_H	Heating voltage	5V±0.1	ACOR DC
R_L	Load resistance	can adjust	
R_H	Heater resistance	33 Ω ±5%	Room Tem
P_H	Heating consumption	less than 800mw	

B. Environment condition

Symbol	Parameter name	Technical condition	Remarks
Tao	Using Tem	-20℃-50℃	
Tas	Storage Tem	-20℃-70℃	
R_H	Related humidity	less than 95%Rh	
O_2	Oxygen concentration	21%(standard condition)Oxygen concentration can affect sensitivity	minimum value is over 2%

C. Sensitivity characteristic

Symbol	Parameter name	Technical parameter	Remarks
Rs	Sensing Resistance	3K Ω -30K Ω (1000ppm iso-butane)	Detecting concentration scope: 200ppm-5000ppm LPG and propane 300ppm-5000ppm butane 5000ppm-20000ppm methane 300ppm-5000ppm H_2 100ppm-2000ppm Alcohol
α (3000/1000) isobutane	Concentration Slope rate	≤0.6	
Standard Detecting Condition	Temp: 20℃±2℃ Humidity: 65%±5%	Vc:5V±0.1 VH: 5V±0.1	
Preheat time	Over 24 hour		

D. Structure and configuration, basic measuring circuit

	Parts	Materials
1	Gas sensing layer	SnO_2
2	Electrode	Au
3	Electrode line	Pt
4	Heater coil	Ni-Cr alloy
5	Tubular ceramic	Al_2O_3
6	Anti-explosion network	Stainless steel gauze (SUS316 100-mesh)
7	Clamp ring	Copper plating Ni
8	Resin base	Bakelite
9	Tube Pin	Copper plating Ni

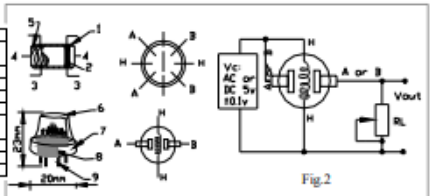
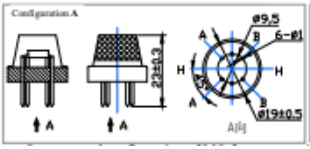
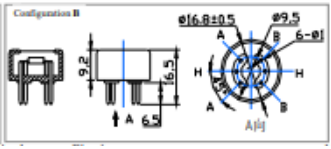


Fig. 1 Fig.2





crust made by plastic and stainless steel net. The heater provides necessary work conditions for work of sensitive components. The enveloped MQ-2 have 6 pin ,4 of them are used to fetch signals, and other 2 are used for providing heating current.

Electric parameter measurement circuit is shown as Fig.2

E. Sensitivity characteristic curve

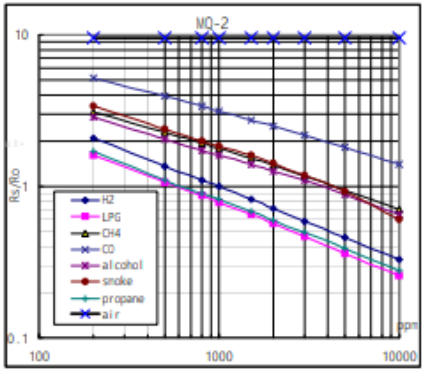


Fig.2 sensitivity characteristics of the MQ-2

Fig.3 is shows the typical sensitivity characteristics of the MQ-2 for several gases.
in their: Temp: 20℃ ,
Humidity: 65% ,
O_2 concentration 21%
RL=5kΩ
Ro: sensor resistance at 1000ppm of H_2 in the clean air.
Rs:sensor resistance at various concentrations of gases.

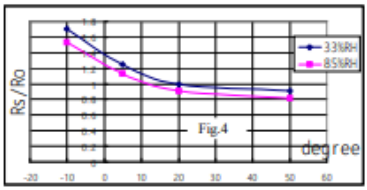


Fig.4 is shows the typical dependence of the MQ-2 on temperature and humidity.
Ro: sensor resistance at 1000ppm of H_2 in air at 33%RH and 20 degree.
Rs: sensor resistance at 1000ppm of H_2 at different temperatures and humidities.

SENSITVITY ADJUSTMENT

Resistance value of MQ-2 is difference to various kinds and various concentration gases. So,When using this components, sensitivity adjustment is very necessary. we recommend that you calibrate the detector for 1000ppm liquified petroleum gas<LPG>,or 1000ppm iso-butane<i-C_4H_{10}>concentration in air and use value of Load resistance that(R_L) about 20 KΩ (5KΩ to 47 KΩ).

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

5. Sensor PIR

HC-SR501 PIR MOTION DETECTOR

Product Discription

HC-SR501 is based on infrared technology, automatic control module, using Germany imported LHI778 probe design, high sensitivity, high reliability, ultra-low-voltage operating mode, widely used in various auto-sensing electrical equipment, especially for battery-powered automatic controlled products.

Specification:

- Voltage: 5V – 20V
- Power Consumption: 65mA
- TTL output: 3.3V, 0V
- Delay time: adjustable (3->5min)
- Lock time: 0.2 sec
- Trigger methods: L – disable repeat trigger, H enable repeat trigger
- Sensing range: less than 120 degree, within 7 meters
- Temperature: – 15 – +70
- Dimension: 32*24 mm, distance between screw 28mm, M2, Lens dimension in diameter: 23mm

Application:

Automatically sensing light for Floor, bathroom, basement, porch, warehouse, Garage, etc, ventilator, alarm, etc.

Features:

- Automatic induction: to enter the sensing range of the output is high, the person leaves the sensing range of the automatic delay off high, output low.
- Photosensitive control (optional, not factory-set) can be set photosensitive control, day or light intensity without induction.
- Temperature compensation (optional, factory reset): In the summer when the ambient temperature rises to 30 ° C to 32 ° C, the detection distance is slightly shorter, temperature compensation can be used for performance compensation.
- Triggered in two ways: (jumper selectable)
  - non-repeatable trigger: the sensor output high, the delay time is over, the output is automatically changed from high level to low level;
  - repeatable trigger: the sensor output high, the delay period, if there is human activity in its sensing range, the output will always remain high until the people left after the delay will be high level goes low (sensor module detects a time delay period will be automatically extended every human activity, and the starting point for the delay time to the last event of the time).
- With induction blocking time (the default setting: 2.5s blocked time): sensor module after each sensor output (high into low), followed by a blockade set period of time, during this time period sensor does not accept any sensor signal. This feature can be achieved sensor output time "and" blocking time "interval between the work can be applied to interval detection products; This function can inhibit a variety of interference in the process of load switching. (This time can be set at zero seconds – a few tens of seconds).
- Wide operating voltage range: default voltage DC4.5V-20V.
- Micropower consumption: static current <50 microamps, particularly suitable for battery-powered automatic control products.
- Output high signal: easy to achieve docking with the various types of circuit.

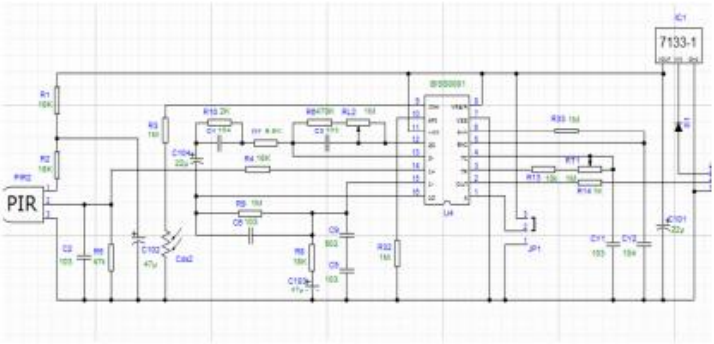
Adjustment:

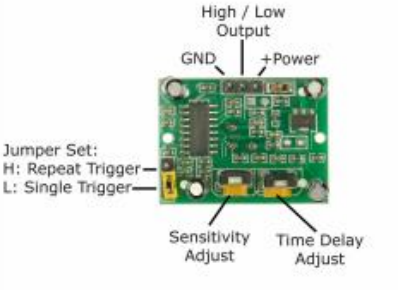
- Adjust the distance potentiometer clockwise rotation, increased sensing distance (about 7 meters), on the contrary, the sensing distance decreases (about 3 meters).
- Adjust the delay potentiometer clockwise rotation sensor the delay lengthened (300S), on the contrary, shorten the induction delay (5S).

Instructions for use:

- Sensor module is powered up after a minute, in this initialization time interval during this module will output 0-3 times, a minute later enters the standby state.
- Should try to avoid the lights and other sources of interference close direct module surface of the lens, in order to avoid the introduction of interference signal malfunction; environment should avoid the wind flow, the wind will cause interference on the sensor.
- Sensor module with dual probe, the probe window is rectangular, dual (A-B) in both ends of the longitudinal direction
  - so when the human body from left to right or right to left through the infrared spectrum to reach dual time, distance difference, the greater the difference, the more sensitive the sensor.
  - when the human body from the front to the probe or from top to bottom or from bottom to top on the direction traveled, double detects changes in the distance of less than infrared spectroscopy, no difference value the sensor insensitive or does not work;
- The dual direction of sensor should be installed parallel as far as possible in inline with human movement. In order to increase the sensor angle range, the module using a circular lens also makes the probe surrounded induction, but the left and right sides still up and down in both directions sensing range, sensitivity, still need to try to install the above requirements.

HC-SR501 PIR MOTION DETECTOR





1 working voltage range :DC 4.5-20V

2 Quiescent Current :50uA

3 high output level 3.3 V / Low 0V

4. Trigger L trigger can not be repeated / H repeated trigger

5. circuit board dimensions :32 * 24 mm

6. maximum 110 ° angle sensor

7. 7 m maximum sensing distance

Product Type	HC--SR501 Body Sensor Module
Operating Voltage Range	5-20VDC
Quiescent Current	<50uA
Level output	High 3.3 V /Low 0V
Trigger	L can not be repeated trigger/H can be repeated trigger(Default repeated trigger)
Delay time	5-300S(adjustable) Range (approximately .3Sec -5Min)
Block time	2.5S(default)Can be made a range(0.xx to tens of seconds
Board Dimensions	32mm*24mm
Angle Sensor	<110 ° cone angle
Operation Temp.	-15-+70 degrees
Lens size sensor	Diameter:23mm(Default)

Application scope

•Security products

•Body induction toys

•Body induction lamps

•Industrial automation control etc

Pyroelectric infrared switch is a passive infrared switch which consists of BISS0001 ,pyroelectric infrared sensors and a few external components. It can au open all kinds of equipments, including incandescent lamp, fluorescent lamp, intercom, automatic, electric fan, dryer and automatic washing machine, etc. It is widely used in enterprises, hotels, stores, and corridor and other sensitive area for automatical lamplight, lighting and alarm system.

Instructions

Induction module needs a minute or so to initialize. During initializing time, it will output 0-3 times. One minute later it comes into standby.

Keep the surface of the lens from close lighting source and wind, which will introduce interference.

Induction module has double -probe whose window is rectangle. The two sub-probe (A and B) is located at the two ends of rectangle. When human body r to right, or from right to left, Time for IR to reach to reach the two sub-probes differs.The lager the time difference is, the more sensitive this module is. Wh body moves face-to probe, or up to down, or down to up, there is no time difference. So it does not work. So instal the module in the direction in which moi activities behaves, to guarantee the induction of human by dual sub-probes. In order to increase the induction range, this module uses round lens which ca from all direction. However, induction from right or left is more sensitivity than from up or down.

6. Buzzer

Buzzer

pro-SIGNAL

Features

- Black in colour
- With internal drive circuit
- Sealed structure
- Wave solderable and washable
- Housing material: Noryl

RoHS Compliant

Applications

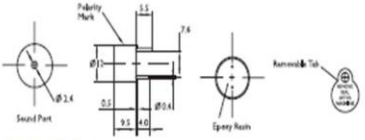
- Computer and peripherals
- Communications equipment
- Portable equipment
- Automobile electronics
- POS system
- Electronic cash register

Specifications:

Rated Voltage	: 6V DC
Operating Voltage	: 4 to 8V DC
Rated Current*	: ≤30mA
Sound Output at 10cm*	: ≥85dB
Resonant Frequency	: 2300 ±300Hz
Tone	: Continuous
Operating Temperature	: -25°C to +80°C
Storage Temperature	: -30°C to +85°C
Weight	: 2g

*Value applying at rated voltage (DC)

Diagram



Dimensions : Millimetres
Tolerance : ±0.5mm

Part Number Table

Description	Part Number
Buzzer, Electromech, 6V DC	ABI-009-RC

Important Notice : This data sheet and its contents (the "Information") belong to the members of the Premier Farnell group of companies (the "Group") or are licensed to it. No licence is granted for the use of it other than for information purposes in connection with the products to which it relates. No licence of any intellectual property rights is granted. The Information is subject to change without notice and replaces all data sheets previously supplied. The Information supplied is believed to be accurate but the Group assumes no responsibility for its accuracy or completeness, any error in or omission from it or for any use made of it. Users of this data sheet should check for themselves the Information and the suitability of the products for their purpose and not make any assumptions based on information included or omitted. Liability for loss or damage resulting from any reliance on the Information or use of it (including liability resulting from negligence or where the Group was aware of the possibility of such loss or damage arising) is excluded. This will not operate to limit or restrict the Group's liability for death or personal injury resulting from its negligence. pro-SIGNAL is the registered trademark of the Group. © Premier Farnell plc 2012.

www.element14.com
www.farnell.com
www.newark.com
www.cpc.co.uk

pro-SIGNAL

LAMPIRAN C

PROPOSAL TUGAS AKHIR

Departemen Teknik Elektro
Fakultas Teknologi Elektro dan Informatika Cerdas - ITS

EE184801 TUGAS AKHIR – 6 SKS

Nama Mahasiswa	: Muhammad Reza Rivandana
Nomer Pokok	: 07111640000117
Bidang Studi	: Telekomunikasi Multimedia
Tugas Diberikan	: Semester Genap Tahun Ajaran 2019/2020
Dosen Pembimbing	: Sri Rahayu, S.T., M.Kom.
Judul Tugas Akhir	: **Rancang Bangun *Smart Home System* Berbasis *Internet of Things* (*Design a Smart Home System based Internet of Things*)**

Uraian Tugas Akhir :

Penggunaan *Internet of Things* (IoT) makin hari makin meningkat Seiring tuntutan revolusi industri 4.0 yang makin masif, implementasi IoT sudah merambah di berbagai bidang seperti manufaktur, logistik, kesehatan, tata kota, rumah, pertanian, dan insdustri otomotif.

Dalam Tugas Akhir ini akan dibuat contoh implementasi *IoT* dalam pengelolaan jaringan listrik rumah tangga dalam bentuk sebuah prototipe *Smart Home System* (SHS), agar dapat dipantau dan dikendalikan dari jarak jauh lewat jaringan internet. Sistem SHS tersebut diadaptasi dari rumah berukuran $6\times9m^2$ yang terbagi dalam 6 area yang berbeda ,lingkup pengelolaan meliputi penggunaan lampu kipas angin ddan alat masak yang dapat dimonitor maupun dikendalikan lewat jaringan internet. Lingkup pengelolaan meliputi penggunaan lampu, kipas angin, dan alat masak. Adapun untuk merealisasikan hal tersebut digunakan sensor cahaya, sensor gas, sensor suhu dan beberapa relay. Selanjutnya perangkat-perangkat tersebut dihubungkan ke NodeMCU ESP8266 sebagai mikrokontroller yang *embedded* dengan modul IoT. Semua data dikirim ke *database/server* via internet, untuk diproses lebih lanjut menjadi tampilan *visual* yang bisa diakses lewat aplikasi *user interface*. Untuk memenuhi kebutuhan tersebut digunakan perangkat lunak (*cloud platform*) *Cayenne platform*. Adapun pengujian sistem yang meliputi uji akses, uji *monitoring*, dan uji *controlling*, Dan QoS (*throughput, jitter* dan *delay*) sementara untuk membantu penyelesaian Tugas Akhir digunakan Bahasa pemograman Arduino IDE dan Bahasa C++.

Kata kunci: *Internet of Things, Monitoring, Controlling, Modul IoT, Smart Home System, cloud platform, Cayenne platform,* QoS, *throughput,jitter* dan *delay*.

Dosen Pembimbing

Sri Rahayu, S.T., M.Kom.
NIP : 196802281997022001

Mengetahui,
Sekretaris Departemen I

Dimas Anton Asfani S.T., M.T.,Ph.D
Nip : 198109052005011002

Menyetujui,
Kepala Lab. Jaringan Telekomunikasi

Eko Setijadi, ST., MT., Ph.D.
NIP : 197210012003121002

BIOGRAFI PENULIS

Penulis buku Tugas Akhir dengan judul "Rancang Bangun *Smart Home System* Berbasis *Internet of Things*" bernama lengkap Muhammad Reza Rivandana. Penulis lahir di Malang, 15 Desember 1997. Penulis menempuh Pendidikan dasar di SD Lowokwawru 3 Malang melanjutkan di SMP 1 Probolinggo, akan tetapi pada kelas 2 SMP penulis pindah ke SMP 18 Malang hingga lulus. Kemudian melanjutkan sekolah di SMAN 1 Malang hingga lulus. Dan melanjutkan kuliah di Institut Teknologi Sepuluh Nopember. Selama masa perkuliahan, penulis aktif di organisasi mahasiswa BEM FTE ITS. Penulis memiliki moto hidup " Bekerja keras dan bersikap baiklah. Hal luar biasa akan terjadi ".